

OPTIMIZAREA CONDUCERII PROCESELOR TEHNOLOGICE PRIN UTILIZAREA DE SUBSISEME INTELIGENTE

Teză destinată obținerii
titlului științific de doctor inginer
la
Universitatea „Politehnica” din Timișoara
în domeniul
INGINERIE ELECTRONICĂ ȘI TELECOMUNICAȚII
de către

Ing. Papazian Petru

Conducător științific:
Referenți științifici:

prof.univ.dr.ing. TIBERIU MUREȘAN
prof.univ.dr.ing. ȘERBAN LUNGU
prof.univ.dr.ing. MIRCEA IVĂNESCU
prof.univ.dr.ing. VIOREL POPESCU

Ziua susținerii tezei: 19 Octombrie 2012.

Seriile Teze de doctorat ale UPT sunt:

- | | |
|------------------------|---|
| 1. Automatică | 7. Inginerie Electronică și Telecomunicații |
| 2. Chimie | 8. Inginerie Industrială |
| 3. Energetică | 9. Inginerie Mecanică |
| 4. Ingineria Chimică | 10. Știința Calculatoarelor |
| 5. Inginerie Civilă | 11. Știința și Ingineria Materialelor |
| 6. Inginerie Electrică | |

Universitatea „Politehnica” din Timișoara a inițiat seriile de mai sus în scopul diseminării expertizei, cunoștințelor și rezultatelor cercetărilor întreprinse în cadrul școlii doctorale a universității. Seriile conțin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susținute în universitate începând cu 1 octombrie 2006.

Copyright © Editura Politehnica – Timișoara, 2012

Această publicație este supusă prevederilor legii dreptului de autor. Multiplicarea acestei publicații, în mod integral sau în parte, traducerea, tipărirea, reutilizarea ilustrațiilor, expunerea, radiodifuzarea, reproducerea pe microfilme sau în orice altă formă este permisă numai cu respectarea prevederilor Legii române a dreptului de autor în vigoare și permisiunea pentru utilizare obținută în scris din partea Universității „Politehnica” din Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300159 Timișoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Teza de doctorat a fost elaborată pe parcursul activității mele în cadrul Departamentului de Electronică Aplicată al Facultății de Electronică și Telecomunicații din cadrul Universității „Politehnica” din Timișoara. În cadrul lucrării sunt reunite atât rezultatele obținute pe baza studiilor și cercetărilor desfășurate pe durata mai multor ani, sub coordonarea intransigentă a conducătorului de doctorat cât și metodele adoptate în soluționarea problemelor specifice domeniului abordat, caracterizate de experiența acumulată ca proiectant și cercetător în domeniul conducerii proceselor tehnologice și a electronicii digitale.

Sunt de menționat deosebitele mulțumiri adresate conducătorului de doctorat, prof.dr.ing. Tiberiu Mureșan, pentru canalizarea interesului către electronica digitală și procesele automatizate și totodată pentru o îndrumare neconținută, perseverentă și de mare valoare, fără de care această teză nu s-ar fi materializat. Sfaturile domnului prof.dr.ing. Tiberiu Mureșan mi-au fost extrem de utile și valoroase pe durata întregii mele activități didactice și de cercetare.

Mulțumesc domnului prof.dr.ing. Aurel Gontean pentru sprijinul acordat pe durata elaborării tezei de doctorat, prin intermediul căruia mi s-a asigurat posibilitatea testării funcționale a anumitor blocuri funcționale a sistemului proiectat, oferindu-mi sisteme de dezvoltare și aparate de măsură moderne.

Mulțumesc colegilor din cadrul colectivelor de Circuite Integrate Digitale și Electronică de Putere, ca și tuturor colegilor din Departamentul Electronică Aplicată din partea cărora am avut un permanent sprijin și imbold pentru finalizarea cu succes a tezei de doctorat.

Se cuvin sincere mulțumiri regretatului prof.dr.ing. Anton Policec, care mi-a călăuzit primii pași în sintetizarea sub formă de articol științific a rezultatelor mele ca și cercetător.

Mulțumiri familiei mele pentru răbdarea, înțelegerea și a numeroaselor încurajări oferite, pe parcursul anilor alocăți elaborării acestei teze de doctorat.

Timișoara, Septembrie, 2012

Ing. Papazian Petru

Pentru Olivia.

Papazian, Petru

Optimizarea conducerii proceselor tehnologice prin utilizarea de subsisteme inteligente

Teze de doctorat ale UPT, Seria 7, Nr. 36, Editura Politehnica, 2012, 244 pagini, 205 figuri, 41 tabele.

ISSN:1842-7014

ISBN: 978-606-554-355-3

Cuvinte cheie: subsistem inteligent, rețea de teren, NCAP, TIM, TEDS, TII, microcontroler, SSIHE, GPMI, proiectare experimentală, FPGA

Rezumat:

Prima parte a tezei prefațează posibilitatea elaborării unei metode de analiză și evaluare a performanțelor rețelelor de teren existente la ora actuală. În acest scop, a fost elaborată o ierarhizare a rețelelor de comunicație care intervin în structura sistemelor de conducere a proceselor tehnologice industriale, cu evidențierea rolului major al rețelei de teren, care asigură comunicația între dispozitivele de teren și unitățile de control ale acestora. S-a realizat o clasificare a acestor rețele pe baza unor parametri considerați a fi semnificativi din punctul de vedere al autorului.

Partea a doua a tezei recurge la metode statistice de proiectare experimentală combinate cu simularea computerizată a comportamentului diverselor structuri de rețele pentru identificarea efectelor diverșilor parametri asupra performanțelor rețelelor de teren. S-a utilizat tehnica „Design of Experiment” (DOE) și s-a aplicat metoda de analiză „D-Optimal”, în mod original, pentru a sintetiza efectele anumitor parametri asupra performanțelor rețelelor de teren. Au rezultat diagrame de influență și modele statistice de predicție a factorilor de răspuns urmăriți.

În ultima parte a tezei a fost conceput un subsistem inteligent de interfațare a dispozitivelor HART la rețeaua Ethernet, asigurând totodată compatibilitatea cu standardul industrial IEEE 1451. S-a propus o metodă originală de înglobare a telegramelor HART în cadrele Ethernet pentru asigurarea comunicării cu dispozitivele de teren HART. Structura propusă, bazată pe noțiuni ca NCAP și TIM, a fost implementată într-un circuit FPGA de tip Virtex 4 și s-a realizat simularea funcțională a noilor blocuri concepute.

CUPRINS

Notații, abrevieri, acronime.....	1
Lista de tabele.....	4
Lista de figuri.....	5
1. Introducere.....	10
1.1. Obiectul tezei.....	10
1.2. Structura tezei.....	11
2. Sisteme de comunicație industriale.....	15
2.1. Introducere.....	15
2.2. Structuri de vehiculare a datelor în sistemele de comunicație industriale.....	15
2.2.1 Ierarhizarea tipurilor de rețele.....	15
2.2.2 Rețeaua de teren.....	16
2.3. Stadiul actual de dezvoltare și utilizare a rețelelor de teren.....	19
2.3.1 Standarde actuale: IEC 61158 și IEC 61784.....	22
2.3.2 Soluții adoptate în implementarea standardului IEC61784-2.....	25
2.3.3 Rețele de teren adaptate la cerințele IEC61784.....	26
2.4. Concluzii și contribuții personale.....	32
3. Analiza și predicția indicatorilor de performanță ale sistemelor bazate pe rețele de teren.....	33
3.1. Introducere.....	33
3.2. Analiza și proiectarea experimentelor.....	34
3.2.1 DOE cu analiză factorială.....	35
3.2.2 DOE cu analiză D-optimal.....	37
3.2.2.1 Criterii pentru cea mai bună proiectare D-optimal.....	40
3.3. Analiza structurală și funcțională a rețelelor de teren de tip IEC 61784 prin metoda D-optimal.....	42
3.3.1 Proiectarea inițială utilizând JMP.....	43
3.3.2 Modelarea rețelelor și simularea experimentelor cu OpNet.....	48
3.3.2.1 Experimentul 1.....	48
3.3.2.2 Experimentul 2.....	53
3.3.2.3 Experimentul 3.....	54
3.3.2.4 Experimentul 4.....	56
3.3.2.5 Experimentul 5.....	58
3.3.2.6 Experimentul 6.....	59
3.3.2.7 Experimentul 7.....	61
3.3.2.8 Experimentul 8.....	63
3.3.2.9 Experimentul 9.....	65
3.3.2.10 Experimentul 10.....	66
3.3.2.11 Experimentul 11.....	68
3.3.2.12 Experimentul 12.....	69
3.3.2.13 Experimentul 13.....	71
3.3.2.14 Experimentul 14.....	72
3.3.3 Evaluarea rezultatelor experimentale.....	74
3.3.3.1 Tipul distribuției.....	75
3.3.3.2 Analiza efectelor individuale.....	82
3.3.3.3 Identificarea efectelor de interacțiune.....	86
3.3.4 Conceperea modelului statistic de predicție.....	89
3.3.4.1 Modelul statistic pentru eficiența transmisiei.....	90
3.3.4.2 Modelul statistic pentru întârziere.....	95

3.4 Concluzii și contribuții personale.....	98
3.4.1 Contribuții personale.....	99
4. Sub sisteme inteligente pentru optimizarea conducerii proceselor tehnologice...	101
4.1. Introducere.....	101
4.2. Standardul industrial IEEE 1451.....	103
4.2.1 IEEE 1451.2 – Interfața TIM – NCAP și formate TEDS.....	105
4.2.2 IEEE 1451.1 – Structura și modelul informațional NCAP.....	109
4.2.3 IEEE 1451.3 Specificații pentru sisteme distribuite, multitransductor.....	111
4.2.4 IEEE 1451.4 Standardul pentru interfața mixtă.....	112
4.2.5 IEEE 1451.5 Specificații pentru comunicarea wireless.....	114
4.2.6 IEEE 1451.0 Specificații pentru unificarea diviziunilor 1451.....	114
4.2.7 Utilizarea microcontrolerelor în implementarea TIM.....	115
4.3. Protocolul de comunicație HART.....	122
4.3.1 Mediul fizic de comunicație HART.....	122
4.3.2 Formatul pachetelor de date HART.....	124
4.3.3 Clase de comenzi HART.....	126
4.3.4 Soluții actuale de implementare HART pe rețele Ethernet.....	127
4.4. Sub sistem inteligent de interfațare HART-Ethernet.....	131
4.4.1 Implementarea modelului informațional NCAP.....	133
4.4.1.1 Elemente de bază UML.....	137
4.4.1.2 Clasa IEEE1451_Block.....	139
4.4.1.3 Clasa IEEE1451_NCAP Block.....	141
4.4.1.4 Clasa IEEE1451_FunctionBlock.....	141
4.4.1.4 Clasele de comunicație Client – Server.....	142
4.4.1.5 Clasa IEEE1451_TransducerBlock.....	143
4.4.1.6 Clasa IEEE1451_ParameterWithUpdate.....	145
4.4.1.8 Modelul UML final al NCAP.....	146
4.4.2 Comunicația NCAP – TIM.....	147
4.4.2.1 Interogarea de către NCAP a TIM.....	148
4.4.2.2 Generarea întreruperilor și registrele de stare.....	149
4.4.3 Transmiterea telegramelor HART pe suport Ethernet.....	149
4.4.3.1 Formatul cadrelor Ethernet II.....	149
4.4.3.2 Înglobarea telegramelor HART.....	150
4.4.3.3 Analiza ratei de transmisie.....	150
4.4.4 Funcționalitatea subsistemului inteligent.....	151
4.5. Concluzii și contribuții personale.....	152
4.5.1 Contribuții personale.....	154
5. Implementarea subsistemului inteligent prin utilizarea FPGA.....	155
5.1. Considerente de implementare.....	155
5.2. Co-proiectarea hardware-software.....	156
5.2.1 Platforma Xilinx ML-401 cu FPGA Virtex 4.....	156
5.2.2 Implementarea blocurilor funcționale hardware-software ale subsistemului inteligent (SSIHE).....	157
5.2.2.1 Magistrala internă OPB.....	158
5.2.2.2 Comunicația pe mediul Ethernet.....	159
5.2.2.3 Accesarea memoriilor externe.....	161
5.2.2.4 Gestionarea întreruperilor.....	162
5.2.2.5 Arbitrarea magistralei OPB.....	163
5.2.2.6 Modulul TIM intern (IPTIM).....	164
5.2.2.7 Blocul UART din componența IPTIM.....	170
5.2.2.8 Modemul HART din componența IPTIM.....	179

5.2.2.9 Logica de comandă a canalului H-Bridge.....	183
5.2.2.10 Structura implementată a SSIHE.....	184
5.2.2.11 Implementarea modulelor software aferente SSIHE.....	187
5.2.3 Integrarea elementelor software/hardware în structura SSIHE.....	188
5.3. Sinteza elementelor SSIHE și testarea funcțională.....	189
5.3.1 Rezultatele sintezei elementelor modulului TIM intern.....	190
5.3.1.1 Sinteza modulului UART și simularea operării.....	190
5.3.1.2 Sinteza modulului modem HART și simularea operării.....	191
5.3.1.3 Blocul de comandă al canalului HBridge (canalul 3).....	192
5.3.1.4 Blocul de gestionare a datelor la nivelul canalelor transductor.....	193
5.3.2 Testarea funcțională a SSIHE.....	195
5.4. Concluzii și contribuții personale.....	198
5.4.1 Contribuții personale.....	199
6. Concluzii finale.....	201
6.1. Sinteza globală a contribuțiilor personale.....	201
6.2. Direcții de cercetare viitoare.....	205
Anexe.....	206
A1.....	206
A2.....	224
A3.....	225
A4.....	228
Bibliografie.....	229

NOTAȚII, ABREVIERI, ACRONIME

APO	Application Process Object
ASIC	Application-Specific Integrated Circuit
BRAM	Block Random Access Memory
CAN	Control Area Network
CBA	Component Based Automation
CiA	CAN in Automation
CIM	Computer Integrated Manufacturing
CIP	Common Interface Protocol
CN	Controlled Nodes
CNC	Computerized Numerically Controlled
CP	Communication Profile
CPF	Communication Profiles Families
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
CSRD	Cyclic Send and Request Data with reply
CST	Composite State Transfer
DCOM	Distributed Component Object Model
DCS	Director-UDP Client/Server
DMIPS	Dhrystones Million Instructions Per Second
DOE	Design of Experiment
DTC	Director-Terminal Client
EDDL	Electronic Device Description Language
EDK	Embedded Development Kit
EMAC	Ethernet MAC
EMC	External Memory Controller
EPL	Ethernet Powerlink
FMS	Fieldbus Message Specification
FPGA	Field Programmable Gate Array
FSK	Frequency Shift Keying
FTP	File Transfer Protocol
GDL	Grade de libertate
GLM	General Linear Model
GPMI	General Purpose Multichannel Interface

HART	Highway Addressable Remote Transducer
Ho	Ipoteza nulă
HTTP	Hyper-Text Transfer Protocol
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineer
IP	Internet Protocol
IPIC	Intellectual Property Interconnect
IPIF	Intellectual Property Interface
IPTIM	Intellectual Property TIM
IRT	Isochronous Real-Time
ISC	Interrupt Service Control
LAN	Local Area Network
LMB	Local Memory Bus
LS	Least Squares Means
MAC	Medium Access Control Layer
MDT	Master Data Telegrams
MII	Media Independent Interface
MMI	Mixed Mode Interface
MMS	Manufacturing Message Service
MMT	Mixed Mode Transducer
MN	Managing Node
NCAP	Network Capable Application Processor
NRT	Non Real-Time
 OCD	Carrier Detect Output
ODVA	Open DeviceNet Vendor Association
OPB	On-Chip Peripheral Bus
OSI	Open Systems Interconnection
PDO	Process Data Objects
PHY	Physical Interface Data Connection
PLC	Programmable Logic Controller
RDCS	Remote DSC
RDTs	Remote Director-Terminal Server
RPC	Remote Procedure Call
RT	Real-Time
RTE	Real-Time Ethernet
RTP	Real-Time and Reliable Datagram Protocol
RTPS	Real-Time Publisher Subscriber

SC	Stație de control
SCNM	Slot Communication Network Management
SDK	Software Development Kit
SDO	Service Data Objects
SDU	Service Data Unit
SoC	Solution on a Chip
SP	Stație de proiectare
SP	Suma Pătratelor
SPI	Serial Peripheral Interface
SSIHE	Subsistem Inteligent de Interfațare HART-Ethernet
TBI	Transducer Bus Interface
TBIM	Transducer Bus Interface Module
TCP/IP	Transfer Control Protocol/Internet Protocol
TEDS	Transducer Electronic Data Sheet
TII	Transducer Independent Interface
TIM	Transducer Interface Module
UART	Universal Asynchronous Receiver Transmitter
UCM	Unitate de control și management
UCS	Unitate centrală de supervizare
UDP	User Datagram Protocol
UIM	Unitate de inspecție și manipulare
UML	Unified Modeling Language
UPC	Unitate de prelucrare și control
VLAN	Virtual bridged LAN
VMC	Viteza mediului de comunicație
VPP	Viteza de prelucrare a pachetelor

LISTA DE FIGURI

- Fig.2.1. Structura ierarhică a unui sistem de comunicație industrial
- Fig.2.2. Timpul de transfer în clasele definite de IEC 61784-2
- Fig.3.1. Distribuția normală a probabilității efectelor
- Fig.3.2. Distribuția efectelor principale
- Fig.3.3. Diagrama de corelare a efectelor
- Fig.3.4. Analiză D-optimal cu factori calitativi multinivel
- Fig.3.5. Fereastra de selecție a factorilor de influență
- Fig.3.6. Factorii de influență definiți în JMP
- Fig.3.7. Fereastra de definire a factorilor de răspuns
- Fig.3.8. Factorii de răspuns definiți în JMP
- Fig.3.9. Setul de candidați generat cu JMP
- Fig.3.10. Profilul varianței de predicție
- Fig.3.11. Profilul varianței de predicție pentru scenariul cel mai defavorabil
- Fig.3.12. Varianța predicției în spațiul experimental
- Fig.3.13. Varianța relativă a coeficienților
- Fig.3.14. Tabelul de proiectare
- Fig.3.15. Rețeaua principală pentru Experimentul 1
- Fig.3.16. Definirea parametrilor aplicației
- Fig.3.17. Definirea profilului aplicației
- Fig.3.18. Structura unei rețele secundare
- Fig.3.19. Atributele semnificative aferente unei stații de lucru
- Fig.3.20. Selectarea rezultatelor simulării
- Fig.3.21. Derularea procesului de simulare
- Fig.3.22. Traficul pe rețea – Experimentul 1
- Fig.3.23. Variația întârzierii – Experimentul 1
- Fig.3.24. Traficul pe rețea – Experimentul 2
- Fig.3.25. Variația întârzierii – Experimentul 2
- Fig.3.26. Rețeaua principală pentru Experimentul 3
- Fig.3.27. Structura unei rețele secundare
- Fig.3.28. Traficul pe rețea – Experimentul 3
- Fig.3.29. Variația întârzierii – Experimentul 3
- Fig.3.30. Rețeaua principală pentru Experimentul 4
- Fig.3.31. Structura unei rețele secundare
- Fig.3.32. Traficul pe rețea – Experimentul 4
- Fig.3.33. Variația întârzierii – Experimentul 4
- Fig.3.34. Traficul pe rețea – Experimentul 5
- Fig.3.35. Variația întârzierii – Experimentul 5
- Fig.3.36. Rețeaua principală pentru Experimentul 6
- Fig.3.37. Traficul pe rețea – Experimentul 6
- Fig.3.38. Variația întârzierii – Experimentul 6
- Fig.3.39. Rețeaua principală pentru Experimentul 7
- Fig.3.40. Structura unei rețele secundare
- Fig.3.41. Traficul pe rețea – Experimentul 7
- Fig.3.42. Variația întârzierii – Experimentul 7

- Fig.3.43. Rețeaua principală pentru Experimentul 8
- Fig.3.44. Structura unei rețele secundare
- Fig.3.45. Traficul pe rețea – Experimentul 8
- Fig.3.46. Variația întârzierii – Experimentul 8
- Fig.3.47. Traficul pe rețea – Experimentul 9
- Fig.3.48. Variația întârzierii – Experimentul 9
- Fig.3.49. Rețeaua principală pentru Experimentul 10
- Fig.3.50. Structura unei rețele secundare
- Fig.3.51. Traficul pe rețea – Experimentul 10
- Fig.3.52. Variația întârzierii – Experimentul 10
- Fig.3.53. Rețeaua principală pentru Experimentul 11
- Fig.3.54. Traficul pe rețea – Experimentul 11
- Fig.3.55. Variația întârzierii – Experimentul 11
- Fig.3.56. Rețeaua principală pentru Experimentul 12
- Fig.3.57. Structura unei rețele secundare
- Fig.3.58. Traficul pe rețea – Experimentul 12
- Fig.3.59. Variația întârzierii – Experimentul 12
- Fig.3.60. Rețeaua principală pentru Experimentul 13
- Fig.3.61. Traficul pe rețea – Experimentul 13
- Fig.3.62. Variația întârzierii – Experimentul 13
- Fig.3.63. Rețeaua principală pentru Experimentul 14
- Fig.3.64. Traficul pe rețea – Experimentul 14
- Fig.3.65. Variația întârzierii – Experimentul 14
- Fig.3.66. Tabelul de proiectare conținând factorii de răspuns
- Fig.3.67. Diagrame de distribuție pentru eficiența transmisiei
- Fig.3.68. Aproximarea prin distribuția Normală și Weibull
- Fig.3.69. Diagramele Q-Q Plot pentru distribuție normală și Weibull
- Fig.3.70. Diagrame de distribuție pentru întârziere
- Fig.3.71. Distribuția întârzierii – 100 Mb/s
- Fig.3.72. Distribuția întârzierii – 10 Mb/s
- Fig.3.73. Q-Q Plot pentru $f_{100}(x)$
- Fig.3.74. Q-Q Plot pentru $f_{10}(x)$
- Fig.3.75. Efecte individuale pentru eficiența transmisiei
- Fig.3.76. Efecte individuale pentru întârziere
- Fig.3.77. Diagrame de influență bazate pe mediile pătratice – eficiența transmisiei
- Fig.3.78. Diagrame de influență bazate pe mediile pătratice – întârziere
- Fig.3.79. Half-normal Plot: a) Eficiența Transmisiei; b) Întârziere
- Fig.3.80. Diagrame de interacțiune pentru eficiența transmisiei
- Fig.3.81. Diagrame de interacțiune pentru întârziere
- Fig.3.82. Specificarea modelului în JMP
- Fig.3.83. Diagrama „Actual by Predicted” pentru eficiența transmisiei
- Fig.3.84. Diagrama „Overlay Plot” pentru eficiența transmisiei
- Fig.3.85. Diagrama „Actual by Predicted” pentru întârziere
- Fig.3.86. Diagrama „Overlay Plot” pentru eficiența transmisiei
- Fig.4.1. Structura bloc a unui sistem de măsurare și control digital
- Fig.4.2. Elementele de bază ale standardului IEEE 1451
- Fig.4.3. Structura TIM cu TEDS incorporat, conform IEEE 1451.2
- Fig.4.4. Structura interfeței TII conform IEEE 1451.2
- Fig.4.5. Structura TEDS conform IEEE 1451.2
- Fig.4.6. Tehnici de calibrare multi-segment
- Fig.4.7. Structura NCAP conform IEEE 1451.1

- Fig.4.8. Modelul informațional NCAP conform IEEE 1451.1
- Fig.4.9. Structura distribuită conform IEEE 1451.3
- Fig.4.10. Structura IEEE 1451.4 mixtă
- Fig.4.11. Structura IEEE 1451.5 Wireless
- Fig.4.12. Structura TIM cu 3 canale de date
- Fig.4.13. Modulul TIM implementat cu PIC18F448 conform IEEE 1451.2
- Fig.4.14. Definierea tipurilor de date TEDS
- Fig.4.15. Definierea structurilor de variabile TEDS
- Fig.4.16. Funcții pentru definirea valorilor aferente TEDS
- Fig.4.17. Linie de comunicație HART clasică
- Fig.4.18. Salvă de semnale HART pe liniile de transmisie analogice
- Fig.4.19. Secvență HART extrasă din semnalul de curent 4 la 20 mA
- Fig.4.20. Tranzacție HART standard
- Fig.4.20. Structura unei telegrame HART
- Fig.4.21. Formate de adresă HART
- Fig.4.22. Durata unei telegrame HART
- Fig.4.23. Clasificarea comenzilor HART
- Fig.4.24. Arhitecturi HART clasice: a) dispozitiv unic; b) multipunct
- Fig.4.25. Multiplexoare HART
- Fig.4.26. Structură HART pe Ethernet cu un singur modem
- Fig.4.27. Structură HART pe Ethernet cu un singur RDTS și multiplexoare
- Fig.4.28. Structură HART pe Ethernet multi-locăție
- Fig.4.29. Structura funcțională a subsistemului inteligent
- Fig.4.30. Structura de referință pentru NCAP
- Fig.4.31. Obiectele din componența modelului informațional NCAP
- Fig.4.32. Declararea claselor în UML
- Fig.4.33. Criterii de vizibilitate în UML
- Fig.4.34. Conceptul de generalizare în UML
- Fig.4.35. Conceptul de agregare în UML
- Fig.4.36. Simboluri UML pentru diagrame de stare
- Fig.4.37. Mediul de modelare Software Ideas Modeler - UML
- Fig.4.38. Descrierea în UML a blocului IEEE1451_Block
- Fig.4.39. Diagrama de stare aferentă clasei IEEE 1451_Block
- Fig.4.40. Descrierea în UML a blocului IEEE1451_NCAP Block
- Fig.4.41. Clasa IEEE1451_FunctionBlock
- Fig.4.42. Diagrama de stare IEEE1451_FunctionBlock
- Fig.4.43. Clasa IEEE1451_Entity
- Fig.4.44 Clasa IEEE1451_ClientPort
- Fig.4.45. Modelul de comunicație Client-Server pentru subsistemul inteligent
- Fig.4.46. Clasa IEEE1451_TransducerBlock
- Fig.4.47. Modelul funcțional al blocului transductor
- Fig.4.48. Clasa IEEE1451_ParameterWithUpdate
- Fig.4.49. Clasa IEEE1451_PhysicalParameter
- Fig.4.50. Modelul informațional al NCAP definit în UML
- Fig.4.51. Operațiile aferente blocului „Control TIM”
- Fig.4.52. Formatul unui cadru Ethernet
- Fig.4.53. Telegramă HART completă
- Fig.4.54. Telegrama HART înglobată într-un cadru Ethernet
- Fig.4.55. Funcționalitatea subsistemului inteligent
- Fig.4.56. Tendința actuală în utilizarea sistemelor de control al producției
- Fig.5.1. Diagrama bloc platforma ML-401

- Fig.5.2. Diagrama bloc aferentă implementării fizice a SSIHE
- Fig.5.3. Structura SSIHE – legendă
- Fig.5.4. Structura modului OPB-IPIF
- Fig.5.5. Structura EMAC-OPB-IPIF aferentă SSIHE
- Fig.5.6. Structura globală a controlerului EMC Xilinx
- Fig.5.7. Controlerul de întreruperi OPB Interrupt Controller
- Fig.5.8. Modulul TIM intern cu interfața OPB IPIF
- Fig.5.9. Ciclul de citire Slave pentru IPTIM
- Fig.5.10. Ciclul de înscriere Slave pentru IPTIM
- Fig.5.11. Controlul registrelor de întrerupere și stare al IPTIM
- Fig.5.12. Controlul operației Trigger
- Fig.5.13. Gestionarea datelor aferente canalelor transductor
- Fig.5.14. Generarea semnalului Reset
- Fig.5.15. Structura canalului Tx HART și logica de control aferentă
- Fig.5.16. Structura canalului Rx HART și logica de control aferentă
- Fig.5.17. Structura canalului H-Bridge și logica de control aferentă
- Fig.5.18. Liniile I/O aferente blocului UART
- Fig.5.19. Structura internă a blocului UART
- Fig.5.20. Definirea blocurilor și semnalelor aferente UART
- Fig.5.21. Secvența VHDL pentru operarea modului TxUnit
- Fig.5.22. Secvența VHDL aferentă operării modului Synchro
- Fig.5.23. Secvența VHDL pentru operarea modului RxUnit
- Fig.5.24. Secvența VHDL pentru implementarea modului Counter
- Fig.5.25. Secvența VHDL a nivelului superior aferent HARTFetcher
- Fig.5.26. Secvența VHDL aferentă modului TxFetcher
- Fig.5.27. Secvența VHDL aferentă modului RxFetcher
- Fig.5.28. Semnalele de intrare/ieșire aferente blocului HARTFetcher
- Fig.5.29. Modulul TxFetcher
- Fig.5.30. Modulul RxFetcher
- Fig.5.31. Liniile de intrare/ieșire ale modemului HART
- Fig.5.32. Structura bloc aferentă modemului HART
- Fig.5.33. Definirea în VHDL a blocurilor funcționale aferente modemului HART
- Fig.5.34. Secvența VHDL aferentă operării blocului Modulator
- Fig.5.35. Secvența VHDL aferentă operării blocului demodulator
- Fig.5.36. Maparea liniilor I/O aferente blocurilor modulator/demodulator
- Fig.5.37. Secvența VHDL aferentă implementării canalului H-Bridge
- Fig.5.38. Structura internă completă a SSIHE implementată în Xilinx EDK/XPS
- Fig.5.39. Semnificația simbolurilor utilizate în structura internă a SSIHE
- Fig.5.40. Conectarea perifericelor interne la magistrală
- Fig.5.41. Liniile de port aferente modului TIM intern
- Fig.5.42. Liniile de port externe implementate în FPGA Virtex 4
- Fig.5.43. Atribuirea adreselor perifericelor din structura internă a SSIHE
- Fig.5.44. Etapele proiectării software a SSIHE în mediul Xilinx SDK
- Fig.5.45. Secvența C++ pentru preluarea adresei de bază a TIM
- Fig.5.46. Secvență C++ aferentă schimbului de date între Microblaze și TIM
- Fig.5.47. Fluxul procesului de co-proiectare hardware/software aferent SSIHE
- Fig.5.48. Diagrame temporale aferente funcționării unității Tx
- Fig.5.49. Diagrame temporale aferente funcționării unității Rx
- Fig.5.50. Operarea modulatorului HART
- Fig.5.51. Operarea demodulatorului HART
- Fig.5.52. Operarea blocului de comandă HBridge

- Fig.5.53. Operarea modulului „TxFetcher”
- Fig.5.54. Operarea modulului „RxFetcher”
- Fig.5.55. Semnale furnizate de către emulatorul HART
- Fig.5.56. Emulator HART bazat pe sistem de dezvoltare cu PIC16F84
- Fig.5.57. Fereastra de transmitere a unei comenzi HART
- Fig.5.58. Fereastra de recepție HART
- Fig.5.59. Sistem bazat pe microcontrolerul PIC18F4550
- Fig.5.60. Topologia HART – Ethernet utilizând SSIHE

LISTA DE TABELE

- Tabelul 2.1. Clasificarea principalelor rețele de teren
- Tabelul 2.2. Elemente noi introduse prin specificații IEC
- Tabelul 2.3. Conținutul standardului IEC 61158
- Tabelul 2.4. Profiluri de comunicație definite prin IEC 61784-1
- Tabelul 2.5. Implementări practice a rețelelor RTE
- Tabelul 3.1. Exemplu cu trei factori de influență
- Tabelul 3.2. Variantele analizei factoriale complete
- Tabelul 3.3. Rezultatele simulărilor aferente analizei factoriale complete
- Tabelul 3.3. Număr de iterații și tipuri de analiză
- Tabelul 3.4. Factori de influență semnificativi
- Tabelul 3.5. Sinteza rezultatelor experimentale
- Tabelul 3.6. Parametrii descriptivi η_T
- Tabelul 3.7. Parametrii caracteristici pentru distribuția normală
- Tabelul 3.8. Parametrii caracteristici pentru distribuția Weibull
- Tabelul 3.9. Parametrii descriptivi Δ_{max}
- Tabelul 3.10. Parametrii descriptivi pentru cele două distribuții
- Tabelul 3.11. Parametrii caracteristici pentru distribuția Weibull
- Tabelul 3.12. Tabelul de influență
- Tabelul 3.13. Efecte ale factorilor de influență
- Tabelul 3.14. Generarea unor variabile suplimentare
- Tabelul 3.15. Set de descriptori pentru η_T
- Tabelul 3.16. Parametrii descriptivi pentru Y
- Tabelul 3.17. Analiza varianței cu parametrii specifici pentru Y
- Tabelul 3.18. Coeficienții modelului de estimare pentru η_T
- Tabelul 3.19. Parametrii descriptivi pentru Δ_{max}
- Tabelul 3.20. Analiza varianței cu parametrii specifici pentru Δ_{max}
- Tabelul 3.21. Coeficienții modelului de estimare pentru Δ_{max}
- Tabelul 4.1. Denumiri standardizate pentru semnalele TII
- Tabelul 4.2. Semnificația funcțională a semnalelor TII
- Tabelul 4.3. Câmpurile Meta-TEDS
- Tabelul 4.4. Channel TEDS pentru canalul 1
- Tabelul 4.5. Channel TEDS pentru canalul 2
- Tabelul 4.6. Channel TEDS pentru canalul 3
- Tabelul 4.7. Antet pentru clasele de obiecte
- Tabelul 4.8. Semnificațiile operației *ClientServerReturnCode*
- Tabelul 4.9. Setul de comenzi mandatorii IEEE 1451
- Tabelul 4.10. Biți de stare asociați canalelor transductor
- Tabelul 5.1. Setul de registre aferent sistemului de întreruperi
- Tabelul 5.2. Semnificațiile liniilor I/O aferente modemului HART
- Tabelul 5.3. Gradul de utilizare al FPGA pentru sinteza UART
- Tabelul 5.4. Gradul de utilizare al FPGA pentru sinteza modemului HART
- Tabelul 5.5. Gradul de utilizare al FPGA pentru sinteza logicii H-Bridge

1. INTRODUCERE

1.1. Obiectul tezei

Stadiul actual de dezvoltare din domeniul industrial impune cerințe stricte asupra sistemelor de conducere a proceselor tehnologice mai ales din punctul de vedere al vitezei de prelucrare a datelor, a posibilităților de reconfigurare precum și încadrarea în anumite standarde. Se conturează astfel o direcție de mare interes în cercetarea posibilităților de îmbunătățire a performanțelor acestor sisteme de conducere, pornind chiar și de la reducerea poluării armonice introduse de către sursele de alimentare monofazate și trifazate în sistemul de distribuție de putere alternativă, așa cum se arată în [1] și [2].

Standardizarea rețelelor de comunicație implicate în vehicularea datelor în sistemele de producție industriale cunoaște la ora actuală un aspect nou, cu o tendință de unificare către un model principal care structurează un sistem de comunicație industrial sub forma ierarhică a trei rețele de tip magistrală, fiecare având atât roluri diferite cât și capacități de comunicare, protocoale și complexități distincte. Rețelele de teren, elemente cheie în asigurarea comunicării între dispozitivele de automatizare, sunt supuse în continuare la diverse îmbunătățiri structurale și funcționale. Un singur tip de rețea nu poate satisface în mod eficient cerințele tuturor aplicațiilor și dispozitivelor conectate. Pentru a acoperi pe o scară cât mai largă necesitățile unui sistem industrial, se conturează o nouă direcție de cercetare în privința unor aspecte legate de parametrii asociați unei rețele de teren.

Tendința generală în conceperea structurilor rețelelor de teren este modelarea acestora pe baza principiilor de funcționare a unei rețele Ethernet, derivând de aici un standard relativ nou, IEC 61874. Se impune astfel o ierarhizare novatoare, corespunzătoare cerințelor acestui standard, deschizându-se calea spre încă o direcție de cercetare.

Monitorizarea proceselor tehnologice aferente sistemelor industriale moderne implică existența unui număr mare de senzori prin intermediul cărora se colectează informația provenită din mediul înconjurător. Cu cât sistemul monitorizat este mai complex, cu atât și inteligența senzorilor și volumul de date prelucrat la nivel local trebuie să crească în mod corespunzător. Tendința actuală de dezvoltare este integrarea acestor blocuri într-un singur chip și aplicarea unui standard universal, agreat fiind standardul IEEE 1451 referitor la interconectarea transductoarelor inteligente. Este introdus noul concept de transductor, care poate fi atât senzor cât și actuator, gradul de inteligență provenind de la anumite blocuri locale de prelucrare a informației, înglobate în structura transductorului.

Dispozitivele utilizate în controlul proceselor tehnologice industriale urmăresc atingerea compatibilității cu standardul IEEE 1451 dar există încă numeroase situații în care trebuie asigurată adaptarea la noul standard al vechilor structuri „moștenite”. Acest aspect necesită cel mai mare efort de cercetare deoarece adaptarea la noul standard trebuie efectuată prin costuri minime de implementare precum și cu păstrarea în anumită măsură a vechilor periferice.

Aceste direcții de cercetare au fost luate în considerare și în cadrul tezei, prin care autorul dorește să aducă anumite contribuții personale la vastul domeniu

al sistemelor de conducere a proceselor tehnologice. Din cele șase capitole ale lucrării, patru tratează problematica dezvoltării cercetării domeniilor expuse mai sus, existând un capitol de noțiuni introductive și unul alocat concluziilor finale și posibilităților de dezvoltare ulterioară.

1.2. Structura tezei

Conținutul tezei, prin intermediul celor șase capitole și patru anexe, abordează un domeniu de cercetare foarte generos, autorul selectând doar aspectele esențiale în descrierea problematicii actuale și în continuare expunând contribuțiile personale legate de domeniul respectiv.

Teza se axează pe introducerea unor contribuții personale privind trei aspecte fundamentale în optimizarea conducerii proceselor tehnologice:

- **ierarhizarea structurilor sistemelor de comunicație industriale după cele mai noi standarde, definirea parametrilor cu semnificație majoră care influențează performanțele rețelelor de teren și analiza soluțiilor actuale adoptate în implementarea standardului IEC 61784-2;**
- **analiza și predicția indicatorilor de performanță a sistemelor bazate pe rețele de teren;**
- **introducerea unor subsisteme inteligente pentru optimizarea conducerii proceselor tehnologice.**

În continuare vom expune elementele esențiale introduse pe parcursul fiecărui capitol.

Capitolul II prefațează posibilitatea elaborării unei metode de analiză și evaluare a performanțelor rețelelor de teren existente la ora actuală. În acest scop, a fost elaborată în primul rând o ierarhizare a rețelelor de comunicație care intervin în structura sistemelor de conducere a proceselor tehnologice industriale. Se evidențiază rolul major al rețelei de teren, care asigură comunicația între dispozitivele de teren și unitățile de control ale acestora. Urmează o clasificare a acestor rețele pe baza unor parametri considerați a fi semnificativi din punctul de vedere al autorului, din care se pot extrage anumiți indicatori utilizați în continuare pentru analiza performanțelor rețelelor de teren.

În ultima parte a acestui capitol s-a evaluat impactul noilor standarde IEC cu privire la caracteristicile rețelelor de teren, efectuându-se o clasificare a acestora pe baza profilurilor de comunicație.

Capitolul III recurge la metode statistice de proiectare experimentală combinate cu simularea computerizată a comportamentului diverselor structuri de rețele pentru identificarea efectelor diversilor parametri asupra performanțelor rețelelor de teren. S-a utilizat tehnica „Design of Experiment” (DOE) și s-a aplicat metoda de analiză „D-Optimal”, în mod original, pentru a sintetiza efectele anumitor parametri asupra performanțelor rețelelor de teren.

Conceperea și configurarea unei rețele eficiente impune identificarea factorilor care influențează în mod deosebit performanțele acesteia. La ora actuală există o diversitate de factori de influență și de aici rezultă problematica determinării prin simulare a efectelor induse, precum și modalitățile corecte de analiză a rezultatelor obținute. Pentru analiza propusă în Capitolul III, parcurgând documentația aferentă, autorul a decis pe baza criteriilor de performanță uzuale în domeniu, a facilităților oferite și nu în ultimul rând a ușurinței de utilizare și de

concepere a unor noi modele descriptive a unor dispozitive sau protocoale de comunicație, selectarea simulatorului OpNet care a condus la rezultate clare, precise, pentru rețelele de teren evaluate.

Comparativ cu analiza factorială, adoptată în câteva lucrări de specialitate, analiza „D-Optimal” a condus la identificarea efectelor introduse de către cinci factori de influență, cu maxim trei niveluri de variație, asupra performanțelor rețelelor de teren parcurgând doar o serie restrânsă de 14 experimente. Factorii de răspuns urmăriți au fost eficiența transmisiei și întârzierea din rețea. Rezultatele celor 14 experimente au fost sintetizate în **Anexa 1**.

Scopul final urmărit în Capitolul III a fost obținerea diagramelor de influență care vin în sprijinul proiectanților rețelelor de teren precum și a unui model matematic unificat de predicție incluzând factorii de influență selectați inițial. Totodată, prin parcurgerea etapelor de analiză, s-au dedus principalele efecte de interacțiune a factorilor de influență asupra performanțelor rețelelor de teren. Utilitatea modelelor obținute constă în posibilitatea proiectantului de a estima pe baza lor, valori ale indicatorilor de performanță cum ar fi eficiența transmisiei sau întârzierea din rețea, pentru orice structură în care intervin combinații ale factorilor de influență stabiliți în analiză, fără a fi necesară derularea experimentului ci doar înlocuirea în model a datelor obținute sub formă matricială.

Capitolul IV tratează posibilitatea introducerii unor dispozitive auxiliare, de tip „**subsisteme inteligente**”, cu o structură hibridă, care pe de o parte mențin compatibilitatea cu vechile elemente din sistem și pe de altă parte respectă cerințele standardului IEEE 1451. Prin introducerea acestora în componența globală a sistemelor de conducere a proceselor tehnologice se poate asigura tranziția la noul standard prin costuri minime datorită posibilității utilizării în paralel atât a vechilor dispozitive cât și a celor noi. Au fost indicate unele modalități de implementare bazate pe microcontrolere de cost redus ale diverselor module funcționale impuse de specificațiile IEEE 1451 în scopul elaborării interfeței standardizate a unui subsistem inteligent capabil să realizeze o comunicație pe o rețea de teren de tip Ethernet industrial.

Prima parte a capitolului IV sintetizează principalele cerințe impuse de către standardul IEEE 1451, cu explicitarea subdiviziunilor importante ale acestuia. Este subliniat rolul fundamental al unității NCAP (Network Capable Application Processor) care poate comunica prin interfața TII (Transducer Independent Interface) cu module TIM (Transducer Interface Module). Sunt descrise prin diagrame sugestive funcțiile fiecărui bloc în parte precum și posibilitățile de alcătuire a modelelor informaționale asociate cu aceste blocuri. Un element novator introdus de către standardul IEEE 1451 este blocul TEDS (Transducer Electronic Data Sheet) cu o structură informațională indicată în **Anexa 2** și **Anexa 3**. În cadrul capitolului IV autorul prezintă o modalitate originală de implementare a informației din blocul TEDS prin intermediul unor structuri de date și secvențe de program scrise în limbajul C și executate cu ajutorul unui microcontroler PIC18F448 care va deține și rolul unui TIM.

Fiind stabilite premisele elaborării unui „subsistem inteligent”, urmează în cadrul capitolului IV o analiză succintă a stadiului actual de utilizare și dezvoltare a dispozitivelor de teren bazate pe protocolul HART. Se constată ca la ora actuală acestea au o răspândire relativ majoră în cadrul sistemelor de conducere a proceselor tehnologice și astfel apare o direcție de cercetare a posibilității compatibilizării acestora cu standardul IEEE 1451. Au fost identificate elementele cheie în funcționarea dispozitivelor HART și a fost propusă o metodă de încapsulare

a telegramelor HART în cadrele Ethernet pentru a asigura transportul acestora pe mediul fizic Ethernet.

Capitolul IV expune bazele elaborării structurii funcționale a **subsistemului inteligent de interfațare HART – Ethernet (SSIHE)**. Punctul de pornire a fost structura de referință a blocului NCAP și s-a indicat o modalitate de implementare a modelului informațional NCAP cu ajutorul limbajului universal UML (Unified Modeling Language). În urma unei analize amănunțite, s-a stabilit modelul de comunicare client – server aferent SSIHE și s-a indicat printr-o diagramă explicativă modelul funcțional al blocului transductor. De aici derivă operațiile aferente logicii de control al TIM care trebuie să gestioneze interogările provenite de la NCAP.

Finalul capitolului IV descrie algoritmul de operare al SSIHE bazat pe organigrama propusă de către autor.

Capitolul V prezintă procesul de proiectare a subsistemului inteligent de interfațare HART – Ethernet (SSIHE), conceput ca o „**punte de legătură**” între vechile dispozitive HART și structurile inteligente bazate pe principiile introduse de către standardul IEEE 1451. Elaborarea SSIHE a impus atât conceperea într-un limbaj de nivel înalt de sinteză hardware (în cazul nostru VHDL) a blocurilor funcționale cât și elaborarea unor rutine software de comandă a acestor blocuri.

Subsistemul inteligent a fost implementat bazându-ne pe conceptul de „Solution on a chip” – SoC (soluție pe un singur cip) care utilizează pe deplin avantajele oferite de circuitele FPGA moderne. În plus, s-au îmbinat atât structuri hardware cât și software care au rezultat în urma procesului de co-proiectare hardware-software. Platforma sistem selectată, ML-401, construită în jurul unui circuit FPGA Virtex 4 (Xilinx), îndeplinește în mod exhaustiv cerințele de implementare ale subsistemului propus în teză.

Diagrama bloc a SSIHE expusă în Capitolul V indică în mod amănunțit legăturile dintre modulele interne și nucleul Microblaze precum și interfața specializată pentru comunicarea cu dispozitivele de teren HART. În plus, a fost inclusă în structura SSIHE o interfață suplimentară, denumită **GPMI (General Purpose Multichannel Interface)** care va permite conectarea unor transductoare suplimentare, compatibile cu standardul IEEE 1451. Au fost descrise pe larg modalitățile de implementare a blocurilor componente ale SSIHE, atât prin adaptarea unor module deja existente și oferite de către firma Xilinx ca module IPIF (Intellectual Property Interface) cât și prin conceperea unor blocuri noi, necesare asigurării compatibilității cu dispozitivele de teren HART. În acest sens a fost necesar studiul semnalelor și informației vehiculate prin intermediul interfeței IPIC (IP Interconnect) al IPIF, reunite în **Anexa 4**.

O deosebită atenție s-a acordat conceperii modulului TIM care a devenit unul intern, IPTIM și care asigură atât stocarea datelor din TEDS cât și logica de control pentru interfațarea la SSIHE a dispozitivelor de teren HART și gestionarea GPMI. A fost efectuată modelarea canalelor transductor, adoptând anumite metode originale de extragere și vehiculare a datelor și au fost indicate principalele secvențe de cod VHDL utilizate în sinteza hardware a blocurilor funcționale.

Co-proiectarea hardware – software a permis elaborarea în paralel atât a unor blocuri funcționale ale SSIHE cât și a unor rutine care vor fi executate de către procesorul Microblaze inclus în circuitul FPGA Virtex 4.

Capitolul V include **sinteza elementelor SSIHE și testarea funcțională (parțială)** atât prin simulare cu ajutorul utilităților oferite de către suita Xilinx ISE cât și prin înglobarea imaginii software în fluxul de biți care descrie structura hardware a sistemului și descărcarea acestor informații în circuitul FPGA. Au fost **proiectate și realizate practic** module suplimentare bazate pe microcontrolere din

familia PIC produse de către Microchip care au permis testarea comunicării pe interfața dedicată dispozitivelor de teren HART.

Capitolul VI reunește concluziile finale și sinteza tuturor contribuțiilor personale ce au rezultat în cadrul tezei. Totodată aici sunt indicate direcțiile de cercetare viitoare în domeniul automatizărilor industriale prin continuarea elaborării unor noi metode de optimizare a comunicării între dispozitivele de teren precum și prin propunerea unui nou standard, denumit de către autor „**Smart HART**”. Acesta va exploata anumite caracteristici ale SSIHE, eliminând însă modulația FSK utilizată în comunicarea cu dispozitivele de teren HART clasice și păstrând în schimb formatul comenzilor și a pachetelor de date HART.

În cadrul tezei s-au utilizat abrevieri și notații care sunt în deplină concordanță cu cele acceptate la nivel mondial, adoptându-se în mare măsură denumirile provenite din limba engleză.

Noțiunile teoretice elaborate pe parcursul capitolelor sunt susținute și validate prin **simulări**.

Elementele hardware au fost testate prin realizarea de **circuite experimentale** și interconectarea modulelor aferente.

Mediile de proiectare/simulare au rulat pe un calculator personal cu sistem de operare Windows 7, recurgându-se la următoarele utilitare: JMP (SAS Company USA), OpNet (OpNet Technologies Inc.), MikroC Compiler (MikroElektronika) și suita Xilinx ISE/EDK/XPS/SDK (Xilinx Inc.).

2. SISTEME DE COMUNICAȚIE INDUSTRIALE

2.1. Introducere

În prezent, sistemele industriale de comunicație asigură legătura atât între instrumente simple cât și cele mai complexe dispozitive de control programabile, întreg ansamblul fiind conectat la rândul său cu una sau mai multe unități centrale de prelucrare a informației, de tip PC. Astfel pentru obținerea unui sistem de conducere industrial robust și cât mai fiabil, intervine în primul rând eficiența structurii de comunicație a datelor adoptată, definită prin diverși factori care contribuie cu o anumită pondere la valoarea globală a acestei eficiențe.

Deși organizațiile internaționale au depus un efort deosebit pentru a dezvolta standarde corespunzătoare pentru rețelele industriale (de teren), se simte încă lipsa unui consens mai larg asupra caracteristicilor fizice pe care trebuie să le întrunească astfel de rețele și mecanismele protocolare aferente [3]. S-a conturat astfel un model principal care structurează un sistem de comunicație industrial sub forma ierarhică a trei rețele de tip magistrală, fiecare având atât roluri diferite cât și capacități de comunicare, protocoale și complexități distincte [4].

2.2. Structuri de vehiculare a datelor în sistemele de comunicație industriale

2.2.1 Ierarhizarea tipurilor de rețele

Prima clasă de rețele este utilizată pentru interconectarea dispozitivelor care supervizează desfășurarea procesului de producție (management, alocarea resurselor, proiectare etc.), pe când a doua categorie de rețele vehiculează datele provenite de la celule de control, inspecția calității etc.

Nivelul cel mai de jos în această ierarhie îl ocupă a treia clasă de rețea, denumită și *rețea de teren* care asigură legătura între unitățile de prelucrare și control (UPC), senzori, actuatori precum și dispozitive simple, mai puțin inteligente.

Figura 2.1 ilustrează structura ierarhică și funcțională a celor trei tipuri de rețele enumerate. Se urmărește o expunere sugestivă, simplă, prin prisma viziunii autorului a celor trei niveluri ale mediului de comunicare precum și a dispozitivelor care pot fi conectate pe cele trei rețele.

Magistrala 1 ocupă nivelul superior și permite interconectarea unităților centrale de supervizare (UCS), a unităților de control și management al procesului de producție (UCM) precum și a unor stații de proiectare (SP) dedicate optimizării conducerii fluxului tehnologic. Ea este în legătură permanentă cu magistralele ierarhic inferioare, având cu acestea, un schimb de date bidirecțional.

Magistrala 2 vehiculează informația provenită de la diverse unități de inspecție și manipulare (UIM), stații de control (SC) etc. și totodată furnizează nivelului superior anumite mesaje legate de starea produsului finit.

Magistrala 3, aparține rețelei de teren, aflată la nivelul cel mai de jos și reprezintă de fapt factorul major în derularea sigură și eficientă a fluxului tehnologic. Dispozitivele de senzare și acționare comunică nemijlocit pe această magistrală cu unitățile locale de prelucrare și control (UPC), preluând informații din mediul fizic și apoi acționând asupra aceluiași mediu, pe baza unui algoritm bine stabilit.

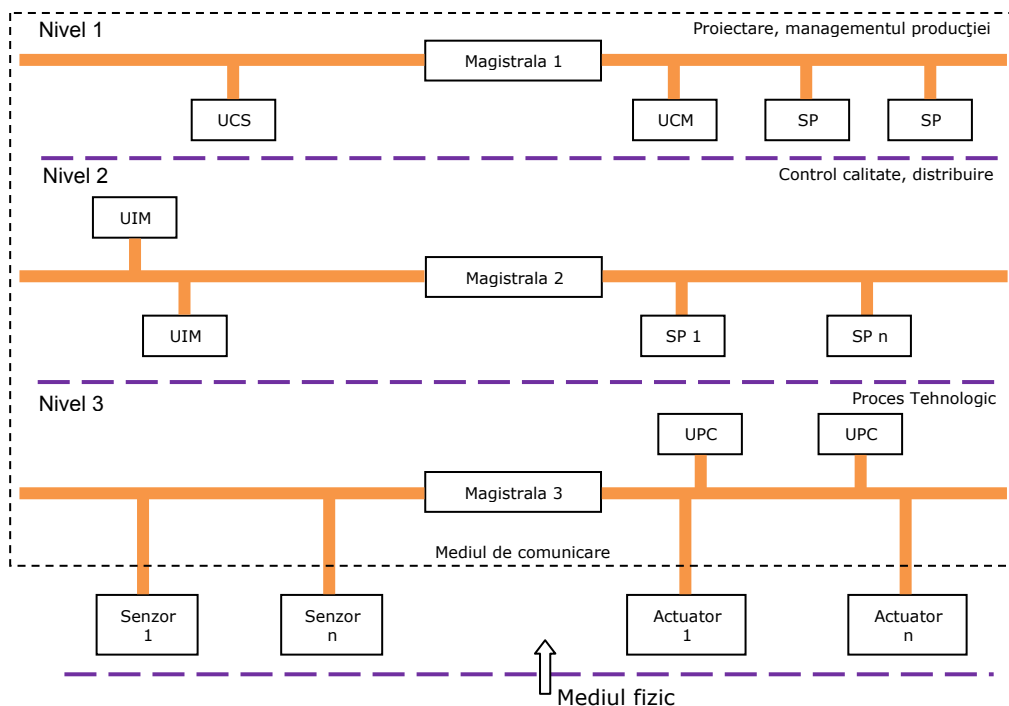


Fig.2.1. Structura ierarhică a unui sistem de comunicație industrial

2.2.2. Rețeaua de teren

Disponând de o structură mai simplă decât a rețelelor aferente nivelurilor 1 și 2, rețelele de teren trebuie să gestioneze un număr considerabil de aspecte care

nu sunt comune celorlalte tipuri de rețele. Se pot astfel sintetiza patru cerințe principale asupra caracteristicilor fundamentale ale acestora și anume:

- i) să asigure suport pentru schimbul de date în timp real;
- ii) să admită operații periodice și aperiodice de citire/scriere a datelor;
- iii) suportul fizic să se bazeze pe tehnici de cost redus de interconectare, permițând astfel atașarea directă pe magistrală a senzorilor sau a actuatorilor;
- iv) conceptul adoptat în proiectarea rețelei să permită înglobarea acesteia în sistemul general de conducere a procesului tehnologic prin posibilitatea unui schimb de informații cu rețelele ierarhic superioare.

Un singur tip de rețea nu poate satisface cerințele tuturor aplicațiilor și dispozitivelor conectate, în mod eficient. Totuși, pentru a acoperi pe o scară cât mai largă necesitățile unui sistem industrial, se pot contura câteva aspecte legate de parametrii asociați unei rețele de teren.

Timpul de răspuns

Problematika majoră a aplicațiilor ce utilizează rețele de teren este limitarea timpului de răspuns. Mai precis, trebuie să existe un interval de timp maxim garantat în generarea unui răspuns la cererea unui transfer de date pe magistrală. Se impune bineînțeles o valoare cât mai redusă a acestui parametru, prin urmărirea reducerii întârzierii maxime care apare de la cererea serviciului lansată pe magistrală de către un utilizator prin intermediul unei stații de lucru (sau a unui periferic conectat la magistrală) și recepționarea mesajului aferent de către nodul destinatar.

Eficiența

Rețeaua de teren are scopul de a vehicula pachetele de date, în mod periodic, între dispozitivele interconectate prin aceasta. Termenul de eficiență, cu o semnificație largă în general, definește pentru cazul de față, raportul dintre cantitatea de informație utilă (valori ale variabilelor de proces etc.) vehiculată pe rețea pe durata unui ciclu de transmisie/recepție și numărul total de biți necesari pentru transportul acestei informații (care include antet, adresă, biți de control etc.). Acest parametru este larg influențat de către mecanismul implementat în nivelul de control al accesului la mediu (Medium Access Control layer) [3].

Prioritatea schimbului de date

Transferurile ciclice de date sunt efectuate întotdeauna într-un mod deterministic. În schimb, cererilor de transfer aciclice le sunt atribuite două niveluri de prioritate: urgent și respectiv normal. Cererile urgente sunt onorate primele de către arbitrul de magistrală care forțează mecanismul de prioritate la elaborarea unei planificări adecvate pentru satisfacerea cererii emise.

Se poate nota deci prezența sau absența unui astfel de mecanism în diversele tipuri de rețele.

Toleranța la erori

Această caracteristică se referă la comportamentul rețelei în cazul defectării dispozitivului de arbitrare al magistralei sau funcționării incorecte al unui Master. În cazul mediilor bazate exclusiv pe existența unui arbitru de magistrală, poate exista o metodă bazată pe generarea unui timeout până la momentul în care alte dispozitive disponibile cu capacitate de arbitrare declanșează mecanismul de alegere a unui nou arbitru. Acest lucru nu va împiedica funcționarea magistralei, dar în schimb anumite variabile de proces pot să devină indisponibile.

Unele topologii simplificate nu au incluse mecanisme de recuperare și este permisă doar o defectare a unui dispozitiv Slave.

Arhitecturi mai robuste gen „token bus”, bazate pe vehicularea circulară a unor markeri sunt practic imune la coruperea sau pierderea informației și permit cuplarea/decuplarea dispozitivelor în cadrul inelului logic astfel creat. Aceste rețele au caracteristici similare din punct de vedere al toleranței la erori cu cele impuse de standardul IEEE 802.4 [3].

Serviciul de legătură a datelor

Uzual există un model stratificat bazat pe trei niveluri de protocolare ce includ nivelul fizic, legătura de date și cel al aplicației, toate acestea fiind incluse în modelul de referință OSI (Open Systems Interconnection). Totuși unele rețele nu au implementate niveluri, iar definiția serviciilor nu este inclusă în standard.

În general, serviciul trebuie să ofere posibilitatea transmiterii mesajelor independent față de tipul conexiunii, cu sau fără confirmarea recepționării corecte la destinatar. O eroare de transmisie ce apare în urma unui algoritm bazat pe confirmare este detectată atunci când semnalul de confirmare a destinatarului nu revine la sursă într-un anumit interval de timp limitat. În acest caz informația se retransmite de câteva ori, fără ca Masterul să elibereze controlul asupra magistralei. Poate exista și posibilitatea efectuării unei cereri de retransmisie a datelor de către receptor spre sursă prin invocarea unui serviciu de actualizare a răspunsului.

Suplimentar, poate exista și un set de servicii cu scopul de a gestiona baza de date a variabilelor distribuite pe magistrală. Astfel, obiectele vehiculate nu reprezintă mesaje ci variabile care au un identificator global de magistrală iar rolul serviciilor este de a citi sau scrie valoarea locală a unei variabile transmise sau recepționate și de a apela la o actualizare aciclică a valorii acestei variabile [3].

Servicii furnizate de către aplicații

Unul dintre obiectivele standardizării rețelelor de teren este de a integra în totalitate subrețelele de instrumentație în cadrul arhitecturii globale a sistemelor de

conducere a proceselor tehnologice, mai precis în contextul magistralelor de tipul al II-lea, expuse anterior.

Există în acest sens specificația asupra serviciului mesajelor de producție MMS (Manufacturing Message Service) – un standard de nivel înalt, definit în mod expres pentru mediul de producție, acceptat pe scară largă în prezent [3]. Astfel, unele tipuri de rețele adoptă o submulțime a acestor servicii MMS, pe când, altele definesc o versiune ușor modificată, cunoscută sub numele de FMS (Fieldbus Message Specification) [5,6].

La modul general, aplicațiile trebuie să beneficieze de următoarele servicii:

- citire/scriere de variabile;
- suport pentru încărcare/descărcare subrutine;
- control de la distanță asupra executării programelor;
- gestionarea evenimentelor speciale cum ar fi cele de alarmare.

Pe de altă parte, există și alte tehnici de exploatare a funcționalităților rețelelor de teren, cum ar fi schimbul periodic automatizat a unui număr de variabile cu timpi de răspuns relativi reduși, în cadrul serviciilor la nivelul aplicației. În acest caz trebuie asigurat accesul utilizatorului către baza de date distribuită a variabilelor, susținută de nivelul imediat inferior al legăturii între date. Astfel acest model de aplicație constă dintr-un set de variabile descrise în termeni ca tip, atribute și valoare și identificate prin nume alcătuite din șiruri de caractere.

Nu în ultimul rând, trebuie să existe posibilitatea de a reîmprospăta valoarea unei variabile și să se poată obține informații despre transmiterea sau recepționarea unei variabile. Magistralele care suportă schimburi de date periodice prin intermediul unui set special de servicii CSR (Cyclic Send and Request Data with reply) trebuie să permită unui dispozitiv Master să inițieze o operație automată de interogare asupra unui set de dispozitive Slave. Interogarea se reia în mod repetat ori de câte ori Masterul devine posesorul fanionului ciclic (tip token), fără vreo intervenție din partea unui proces de aplicație. Acest mecanism poate fi accesat printr-un tip particular de conexiune virtuală denumită conexiune ciclică [7] asigurând astfel o reîmprospătare periodică a valorii variabilelor fără o intervenție explicită din partea utilizatorului.

Parametrii prezenți influențează în mod direct atât funcționalitatea rețelei de teren cât și performanțele acesteia, dat fiind faptul că anumite caracteristici sunt implementate prin diverse modalități, mai mult sau mai puțin eficiente, rezultând necesitatea unei metode de clasificare unitare și de calcul a performanțelor globale a unei rețele de teren.

2.3. Stadiul actual de dezvoltare și utilizare a rețelelor de teren

Se propune în cele ce urmează o metodă originală de analiză și evaluare a performanțelor rețelelor de teren existente în prezent, pornind de la anumite

20 Sisteme de comunicație industriale - 2

particularități deja cunoscute și introducând elemente noi ce vin în sprijinul descrierii unor caracteristici suplimentare.

Ca punct de pornire se vor expune principalele tipuri de rețele de teren utilizate în prezent, clasificate după câteva criterii fundamentale și de importanță majoră din punctul de vedere al autorului, centralizate în Tabelul 2.1.

STANDARD AN	TOPOLOGIE	LUNG. MAX.	NR. DISP.	TIP DE CODARE	METODĂ ACCES	VITEZĂ TRANSM.	MEDIU TRANSM.	DOMENIU UTILIZARE	GRAD UTIL.
ARCNET 1980	STEA	610m	255	BI-PULS RZ	TOKEN	2,5Mb/s	CABLU COAXIAL, TORSADAT, OPTIC	BIROURI, REȚELE LOCALE	5%
ASI 1994	OARECARE	100m	62	MANCHESTER II	CENTRAL, MASTER UNIC	167Kb/s	BIFILAR	PROCESE DE PRODUCȚIE	9%
BITBUS IEEE1118 1991	LINIARĂ	1200m	250	NRZI CU CRC COMPATIBIL RS485	CENTRAL, MASTER UNIC	1,5Mb/s (100m)	CABLU BIFILAR TORSADAT	REȚEA PENTRU ECHIPAMENTE AUXILIARE	5%
CAN 1980	LINIARĂ	500m	110	NRZ	ACCES MULTIPLU CU DETECȚIA COLIZIUNILOR	1Mb/s (40m)	CABLU BIFILAR TORSADAT	AUTOMOTIV CONTROL PROCESE TEHNOLOGICE	25%
CONTROLNET 2000	LINIARĂ STEA ARBORE MIXTĂ	1000m	99	MANCHESTER	MULTI MASTER	5Mb/s	CABLU COAXIAL, OPTIC	PROCESE DE PRODUCȚIE	14%
CCLINK 1997	LINIARĂ	1200m	64	NRZI COMPATIBIL RS485	MASTER UNIC ȘI DISPOZITIVE SLAVE	10Mb/s (100m)	CABLU BIFILAR TORSADAT, OPTIC	PROCESE DE PRODUCȚIE	9%
DEVICENET 2000	LINIARĂ	100m	64	CIP – COMMON INDUSTRIAL PROTOCOL	MASTER/ SLAVE	500Kb/s	CABLU BIFILAR TORSADAT ECRANAT	PROCESE DE PRODUCȚIE	15%
ETHERNET/IP 1980	LINIARĂ STEA	2500m	VAR.	MANCHESTER 4b/5b, MLT3 8b4qt, PAM5	ACCES MULTIPLU CU DETECȚIA COLIZIUNILOR	10Mb/s 100Mb/s 1000Mb/s	CABLU COAXIAL/ TORSADAT NEECRANAT, OPTIC	BIROURI, REȚELE LOCALE, GLOBALE, PRODUCȚIE	50%
FIELDBUS FOUNDATION H1 1996	OARECARE	1900m	32	MANCHESTER	CLIENT/ SERVER	31,25Kb/s	CABLU BIFILAR TORSADAT, OPTIC	PROCESE DE PRODUCȚIE	7%
FIELDBUS FOUNDATION HighSpeed Ethernet HSE 1999	OARECARE	2000m	1024	COMPATIBIL ETHERNET	CLIENT/ SERVER	100Mb/s	CABLU TORSADAT NEECRANAT, OPTIC	PROCESE DE PRODUCȚIE	7%
FIP/WorldFIP 1980/1993	LINIARĂ	1000m	64	MANCHESTER	PRODUCĂTOR CONSUMATOR	31,25Kb/s 2,5Mb/s 5Mb/s-optic	CABLU BIFILAR TORSADAT, OPTIC	PROCESE DE PRODUCȚIE/ CERN	9%
HART 1989	INEL	3000m	1000	FSK	MASTER/ SLAVE	1,2Kb/s	CABLU TORSADAT ECRANAT 2/4 FIRE	PROCESE DE PRODUCȚIE	26%

2.3 – Stadiul actual de dezvoltare și utilizare a rețelelor de teren 21

INTERBUS-S 1987	INEL	500m 13Km	215	NRZ	MASTER/ SLAVE	500Kb/s 2Mb/s	CABLU TORSADAT ECRANAT/ OPTIC	PROCESE DE PRODUCȚIE	37%
LON 1999	OARECARE	2000m	64	P-PERSISTENT CSMA	PRODUCĂTOR CONSUMATOR	78Kb/s 1,25 Mb/s	COAXIAL TORSADAT OPTIC WIRELESS	PROCESE DE PRODUCȚIE	9%
MODBUS 1979	OARECARE	1000m	247	COMPATIBIL RS232/485 CU CRC	MASTER/ SLAVE CLIENT/ SERVER	9,6Kb/s	CABLU 2/4 FIRE	PROCESE DE PRODUCȚIE	22%
PROFIBUS PROFIBUS PA 1989	OARECARE	1200m 1900m	127 32	COMPATIBIL RS485 MANCHESTER	MASTER/ SLAVE	12Mb/s 31,25Kb/s	CABLU BIFILAR TORSADAT, OPTIC	PROCESE DE PRODUCȚIE	26%
SERCOS I/1991 II/1999 III/2005	INEL INEL INEL/ LINIARĂ	50000m	254 254 511	TDMA TDMA v. ETHERNET	MASTER/ SLAVE	4Mb/s 16Mb/s 100Mb/s	OPTIC OPTIC v. ETHERNET	PROCESE DE PRODUCȚIE	22%

Tabelul 2.1. Clasificarea principalelor rețele de teren

După cum rezultă, la ora actuală există numeroase rețele de teren industriale, care se bazează pe diverse tehnologii și metode de comunicație. Din punctul de vedere al utilizatorului distincția dintre aceste rețele se rezumă la viteză, numărul maxim de dispozitive (noduri), lungimea maximă a segmentului de rețea, topologia tipul mediului de transmisie și domeniul de aplicabilitate. Pe de altă parte, un rol important în modul de operare al unei rețele de teren îl joacă modul de arbitrar, tipul accesului la informație a participanților precum și protocolul de comunicație utilizat.

Din Tabelul 2.1 rezultă în principiu că fiecare rețea prezintă caracteristici individuale și a fost concepută strict pentru aplicații concrete, după necesitatea proceselor de producție particulare.

În momentul proiectării unei rețele, un factor important îl reprezintă în mod uzual viteza de vehiculare a datelor. Deși, în anumite studii de specialitate [6], autorii arată că viteza pentru Lonwork este relativ redusă (1,25 Mb/s), acest lucru nu reprezintă un factor critic. În lucrarea [8] „Evolution potentials for fieldbus system” autorul introduce rețeaua multi-nivel pentru procesele CIM (Computer Integrated Manufacturing), împreună cu aspectele istorice ale rețelei de teren. Rețeaua de teren este cea care permite unui dispozitiv de control de tip PLC (Programmable Logic Controller) să comunice la nivelul cel mai de jos în procesul de producție.

Se poate remarca o evoluție importantă în ultima vreme din punctul de vedere al tehnologiei de comunicație din domeniul industrial, cu toate că perioada este relativ scurtă de la apariția primei rețele până în prezent. Consultând Tabelul 2.1 se poate observa o trecere treptată de la interfețe simple paralele, seriale gen RS485, spre configurații mai sofisticate și tipuri de comunicații foarte diversificate, rezultând necesitatea unor standardizări cum ar fi IEC (International Electrotechnical Commission) 61158, 61784 sau Foundation Fieldbus.

Standardizarea a fost o încercare crucială asupra tehnologiei rețelelor de teren. În contrast cu multitudinea soluțiilor tehnice disponibile, adesea chiar redundante, doar o standardizare internațională ar fi reușit să asigure interoperabilitatea și continuitatea din punctul de vedere al utilizatorilor acestor rețele. De aici derivă și numeroase surse de conflict între diverse grupuri de interes,

influențate în principal de către principalii producători din domeniu. Astfel aceștia au influențat într-un sens sau altul procesul de standardizare, înclinând balanța spre propriile lor soluții tehnice deja existente. Urmările au fost pe măsură, evenimentele recente din domeniu conducând la o satisfacere în prea mică măsură a utilizatorilor finali.

2.3.1. Standarde actuale: IEC 61158 și IEC 61784

La momentul de față o rețea de teren este privită ca o rețea de comunicație digitală care este utilizată în locul celei clasice analogice de 4-20mA din domeniul proceselor industriale.

IEC definește rețeaua de teren ca „o magistrală de date digitală, serială, multipunct pentru comunicarea cu dispozitive industriale de instrumentare și control – dar nu numai – cum ar fi traductoare, actuatoare și controlere locale” [9]. În literatura de specialitate, această definiție este regăsită într-un mod mai extins, reprezentativ fiind formularea de către Fieldbus Foundation: „O rețea de teren este o legătură de comunicare digitală, bidirecțională, multipunct între dispozitive inteligente de măsurare și control. Ea joacă rolul de rețea locală (LAN) pentru controlul avansat al proceselor, a executării de la distanță a operațiilor de citire/scriere precum și a aplicațiilor industriale de viteză ridicată” [9].

Specificațiile IEC furnizează un singur document care stă la baza comparației între tehnologiile existente deși nu asigură utilizatorului final alegerea celei mai bune opțiuni în selectarea rețelei de teren. Cu toate acestea, standardizarea aduce beneficii atât utilizatorilor cât și proiectanților și diverselor organizații internaționale din domeniu. Tabelul 2.2 sintetizează principalele aporturi ale standardului IEC 61158 pentru cele trei categorii enumerate.

Semnificație utilizator	Semnificație proiectant	Semnificație organizații
Format unic	Format unic	Aprobare internațională
Însumarea tehnologiilor	Întrepătrunderea tehnologiilor	Recunoaștere internațională

Tabelul 2.2. Elemente noi introduse prin specificații IEC

De asemenea, s-a urmărit lărgirea setului de operații aplicabile asupra proceselor cum ar fi: diagnosticul în timp real, calibrarea la distanță, introducerea dispozitivelor adaptabile precum și reducerea deconectărilor nedorite ale sistemului.

Nu în ultimul rând, o standardizare conduce în mod implicit la reducerea costurilor de operare și întreținere, sprijină integrarea în normele de securitate și îmbunătățește eficiența conducerii procesului tehnologic.

Standardul IEC 61158, adoptat în Ianuarie 2000, conține opt protocoale, fiecare susținut de un anumit grup de producători și de fapt total incompatibile între ele. A urmat depunerea unui efort deosebit din partea IEC de a armoniza aceste protocoale care duc lipsă de o arhitectură și o terminologie comună. Tabelul 2.3 expune conținutul standardului IEC 61158.

După cum se observă, caracteristicile tehnice nu sunt singurii factori care diferențiază diversele rețele de teren existente la ora actuală. Chiar și în urma consultării datelor din Tabelul 2.3, rezultatul acestei standardizări conduce și în continuare la coexistența unor soluții incompatibile.

Niveluri	Referință	Semnificații/Observații
Generalități	IEC 61158-1	Raport tehnic
Nivelul fizic	IEC 61158-2	8 tipuri de transmitere a datelor, aprobat în 1993, bazat pe Foundation Fieldbus, WorldFIP și Profibus PA
Nivel de legătură al datelor -servicii -protocele	IEC 61158-3 IEC 61158-4	Aprobat în 2000. Fiecare standard aparține unui alt protocol.
Nivel de aplicație -servicii -protocele	IEC 61158-5 IEC 61158-6	Tip1: FF-H1, Tip2: ControlNet, Tip3: Profibus, Tip4: P-Net, Tip5: FF-HSE, Tip6: SwiftNet, Tip7: WorldFIP, Tip8: Interbus-S
Administrare rețea	IEC 61158-7	Pregătit pentru revizuire
Testarea conformității	IEC 61158-8	În așteptare

Tabelul 2.3. Conținutul standardului IEC 61158

Anii 2008 – 2010 aduc la completarea standardului cu noi tipuri de rețele de teren, la renunțarea la SwiftNet precum și la reorganizarea tipurilor sub forma familiilor de profiluri de comunicație – CPF (Communication Profiles Families) reunite sub standardul IEC 61784-1 [10]. Există 16 astfel de profiluri, fiecare din ele cuprinzând un număr de rețele de teren, asociate după anumite criterii, așa cum este indicat în tabelul 2.4.

CPF	PROTOCOL DE BAZĂ	REȚELE DE TEREN ASOCIATE
1	Foundation Fieldbus	H1, H2, HSE
2	CIP	ControlNet, EtherNet/IP, DeviceNet
3	Profibus	DP, PA, Profinet (4 tipuri)
4	P-NET	P-NET RS485, P-NET RS232, P-NET/IP
5	WorldFIP	3 tipuri
6	InterBus	diverse tipuri
7	SwiftNet	în curs de eliminare/revizuire
8	CC-Link	3 tipuri
9	HART	4-20 mA/HART
10	Vnet/IP	Vnet
11	TCNet	2 tipuri
12	EtherCat	Industrial Ethernet
13	Ethernet Powerlink	Ethernet
14	EPA	Ethernet
15	MODBUS-RTPS	MODBUS-RTPS, MODBUS-TCP
16	SERCOS	Tipurile I, II și III

Tabelul 2.4. Profiluri de comunicație definite prin IEC 61784-1

Utilizarea anumitor aplicații conduce în mod implicit la existența diferitelor cerințe caracterizate prin indicatori de performanță. O listă a indicatorilor de performanță alcătuiește cerințele globale a unor clase de aplicații dar totodată acești

indicatori au anumite limitări sau domenii de influență restrânse. Astfel, fiecare profil de comunicație enumerat în Tabelul 2.4 are rolul de a specifica indicatorii de performanță adecvați unei anumite clase de aplicații.

Pentru a profita de avantajele tehnologiei Internet și a protocoalelor de tip HTTP (Hyper-Text Transfer Protocol) în utilizarea serverelor web la gestionarea și controlul dispozitivelor de teren sau a protocoalelor FTP (File Transfer Protocol) care permit încărcarea directă a fișierelor de date în aceste dispozitive, s-a introdus specificația IEC 61784-2. Aceasta reunește toate soluțiile de comunicație în timp real pe suport Ethernet adecvate RTE (Real-Time Ethernet) [11] care trebuie să mențină compatibilitatea cu protocolul TCP/IP și cu infrastructura Ethernet existentă.

IEC 61784-2 propune în noua listă de profiluri de comunicație pentru RTE, următorii indicatori de performanță:

- **timpul de transfer:** durata de timp necesară de a transporta un pachet de date SDU (Service Data Unit) sau un mesaj de la un nod sursă la un nod destinație al rețelei. Acest timp este măsurat în cadrul interfeței spre nivelul de aplicație (Application Layer). Valoarea maximă trebuie specificată pentru cazul transmisiei fără erori și pentru o transmisie cu un cadru pierdut cu recuperare;
- **numărul maxim de noduri:** indică numărul maxim de dispozitive RTE suportat de către un profil de comunicație (CP – Communication Profile);
- **topologia de bază a rețelei:** sunt acceptate configurații stea, inel (buclă) și cu înlănțuire (Daisy-chain) sau combinații între acestea;
- **numărul maxim de comutatoare (switch) între dispozitive (noduri):** definește de fapt structura globală a rețelei suportată de profilul de comunicație selectat;
- **rata de transfer:** reprezintă cantitatea totală a datelor vehiculate de un obiect al procesului unei aplicații (APO – Application Process Object) considerată la nivel de octet, pe un segment, pe o durată de o secundă;
- **lățimea de bandă non RTE:** procentul lățimii de bandă care se poate utiliza pentru comunicații care nu necesită timp real;
- **acuratețea sincronizării temporale:** are rolul de a indica valoarea maximă a deviației între semnalele de tact provenite de la oricare două noduri din rețea;
- **timpul de recuperare de redundanță:** specifică timpul maxim de recuperare în cazul apariției unei singure avarii permanente. În acest caz, timpul de transfer cu avarii permanente, dar nu având avarii temporare este înlocuit de acest timp de recuperare.

Pe de altă parte, din punctul de vedere al aplicațiilor utilizator, se pot evidenția trei clase de cerințe asupra timpului de transfer în rețelele RTE [11], ilustrate grafic în figura 2.2.

Prima categorie, de viteză redusă, se aplică în cazurile în care un operator uman monitorizează evoluția unor procese tehnologice, pe un ecran, câteva imagini pe secundă. Acest tip de operare nu necesită timpi de transfer mai mici de 100 ms și comunicația se poate implementa utilizând protocolul TCP/IP clasic pe mediul fizic Ethernet deja existent.

A doua clasă, cu timpi de transfer sub 10 ms, include controlul proceselor în care intervine schimbul de date între dispozitive PLC (Programmable Logic Controller) sau PC în vederea supervizării unor unelte ce acționează în procesul tehnologic.

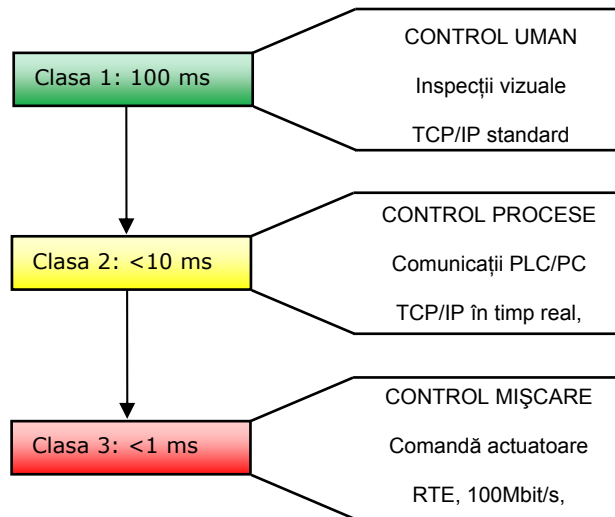


Fig.2.2. Timpul de transfer în clasele definite de IEC 61784-2

Pentru a atinge astfel de performanțe, sunt necesare echipamente de calcul performante care pot prelucra protocolul TCP/IP în timp real sau se impune simplificarea acestuia în vederea obținerii acelorași rezultate cu dispozitive de calcul simple și mai puțin costisitoare.

Clasa a treia implică utilizarea unei infrastructuri Ethernet cu o rată de transfer de minim 100 Mbit/s și necesită modificarea atât a protocolului de acces la mediu cât și a structurii hardware. În această categorie se regăsesc sistemele de comandă a mișcării unor actuatoare conectate la rețele de teren, cu timpi de transfer sub 1 ms, pentru asigurarea sincronizării corespunzătoare între aceste dispozitive.

2.3.2. Soluții adoptate în implementarea standardului IEC 61784-2

Global, interfețele de comunicație sunt structurate pe diverse nivele. Comună oricărui tip de rețea Ethernet este infrastructura fizică. Aplicațiile non real-time utilizează protocoalele definite în standardul ISO 8802-3 precum și suita TCP/UDP/IP [11].

La ora actuală există trei abordări în implementarea unei rețele RTE, adoptate funcție de categoria în care se încadrează domeniul de utilizare, conform figurii 2.2.

Prima metodă menține neschimbate protocoalele TCP/UDP/IP și acționează asupra nivelului superior pentru a atinge o comunicație în timp real, de unde denumirea soluției – „deasupra TCP/IP” (on top of TCP/UDP/IP)[11].

O altă abordare constă în renunțarea la protocoalele TCP/UDP/IP și accesarea directă a funcțiilor Ethernet, cunoscută sub numele „deasupra Ethernet” (on top of Ethernet) [11].

Pentru a întruni a treia clasă de cerințe ilustrate în figura 2.2 este necesară o înlocuire și modificare a infrastructurii și a mecanismului Ethernet în ideea obținerii unui comportament în timp real adecvat. Metoda este cunoscută sub numele „Ethernet modificat” (Modified Ethernet) [11].

2.3.3. Rețele de teren adaptate la cerințele IEC 61784

Pe baza soluțiilor expuse mai sus, la ora actuală există numeroase implementări practice a rețelelor RTE, cele mai semnificative fiind expuse în tabelul 2.5.

Mod de implementare	deasupra TCP/IP	deasupra Ethernet	Ethernet modificat
REJEA/ TIP CPF- IEC 61158	MODBUS/TCP-CPF 15	ETHERNET POWERLINK-CPF 13	SERCOS-CPF 16
	ETHERNET/IP-CPF 2	TCNET-CPF 11	ETHERCAT-CPF 12
	P-NET-CPF 4	EPA-CPF 14	PROFINET IO-CPF 3
	VNET/IP-CPF 10	PROFINET CBA-CPF 3	

Tabelul 2.5. Implementări practice a rețelelor RTE

Vom analiza în continuare câteva aspecte specifice fiecărui tip de rețea enumerată în tabelul 2.5.

1. **CPF 2 – ETHERNET/IP** (IP – Industrial Protocol) este sprijinit de către ODVA (Open DeviceNet Vendor Association) și ControlNet International. Elaborat de către Rockwell, utilizează protocolul CIP (Common Interface Protocol) care se poate regăsi și în cazul rețelelor ControlNet și DeviceNet [12]. Tehnologia de comunicație adoptată, deja asigură o compatibilitate ISO/IEC 8802-3 de operare în timp real. Doar în modul full – duplex, comutat cu „switch-uri”, apar coliziuni și de aici și imposibilitatea deducerii întârzierilor de pe rețea. Dispozitivul de comutare poate întârzia diferitele cadre Ethernet dacă porturile de ieșire sunt în prealabil ocupate cu transmiterea altor cadre. Rezultă de aici caracterul non – deterministic al întârzierilor, de neacceptat în cazul aplicațiilor în timp real [11]. Pentru eliminarea acestui comportament, s-a implementat un algoritm care permite emițătorului atribuirea de priorități cadrelor Ethernet, fapt prevăzut în specificația IEEE 802.3 [13]. Se remarcă introducerea în cadrul standard Ethernet al unui câmp denumit VLAN (Virtual bridged Local Area Network) compus dintr-un identificator (VLAN-ID) și un indice de prioritate al mesajului cuprins între 0 și 7. Astfel mesajelor Ethernet/IP de timp real li se va atribui indicele de prioritate cel mai ridicat pentru a putea fi transmise prin dispozitivele de comutare înaintea cadrelor clasice, obținându-se performanțe care se încadrează în cerințele unei rețele de timp real [13].

Protocolul CIP utilizat necesită sincronizarea semnalelor de tact provenite de la toate dispozitivele conectate pe rețea, specificația IEEE 1588 [14] asigurând sincronizarea cu o acuratețe de 0,5 μ s. Prin această modalitate se asigură executarea tuturor operațiilor sau acțiunilor pe baza unei planificări temporale elaborate în prealabil conducând la independența distribuțiilor temporale a

aplicațiilor față de anumitele întârzieri introduse în rețeaua de comunicație. Respectând cu strictețe aceste cerințe legate de sincronizare, Ethernet/IP devine o soluție viabilă pentru aplicațiile în timp real, deși în sine, ca rețea de comunicare este tot de tip non-deterministic.

2. **CPF 4 – P-NET**, încapsulat în pachete UDP/IP, a fost propus de către comitetul național Danez [15]. Rezultă de aici avantajul posibilității de rutare a pachetelor P-NET atât prin rețele bazate pe IP cât și prin cele non IP. Pentru acest procedeu există două elemente cheie: adresele sursă și tabela de destinație. În cazul unei aplicații simple, cele două elemente pot fi chiar adresele celor două noduri ce alcătuiesc rețeaua de teren [11]. Prin includerea în acest tabel a unor adrese IP se va obține posibilitatea rutării în rețelele bazate pe IP, aceste adrese fiind tratate de către nodurile unei rețele de teren P-NET în mod obișnuit, ca și celelalte aparținând protocolului original P-NET. Acest lucru va permite unui client P-NET să acceseze servere de pe o rețea bazată pe IP aparent fără necesitatea de a cunoaște detalii despre aceste adrese [11].

La ora actuală, specificațiile P-NET/IP nu indică modul în care se asigură comportamentul în timp real al acestui protocol pe rețele de tip Ethernet.

3. **CPF 10 – VNET/IP** este un protocol introdus de către Yokogawa Electric Corporation și utilizează protocolul standard TCP/IP pentru a beneficia de facilitățile oferite pe rețea cum ar fi HTTP (Hyper Text Transfer Protocol) precum și alte extensii de timp real denumite RTP (Real-time and Reliable Datagram Protocol) [11].

Utilizând de fapt doar protocolul UDP/IP pentru a transporta cadrele RTP, VNET-IP în sine nu este un protocol Ethernet în timp real. Rețeaua VNET-IP constă din unul sau mai multe domenii interconectate prin rutere. Totuși se respectă valoarea limită de 10 ms a timpului minim de răspuns pentru aplicațiile de control a proceselor industriale, deși traficul generat de alte transferuri gen HTTP sau TCP pe aceeași rețea, ar putea altera această valoare de 10 ms [16].

La nivelul de aplicație, variabilele, evenimentele, anumite domenii de timp, sunt toate tratate ca obiecte și li se atribuie servicii corespunzătoare. Astfel o variabilă poate fi accesată atât prin metoda „Client – Server”, prin operații de citire/scriere cât și prin metoda „Publicare – Subscriere” cu moduri de operare „push/pull” [11]. În modul „pull”, dispozitivul „Publisher” distribuie datele în mod periodic pe rețea, la cererea unui așa zis dispozitiv subscris. În modul „push”, cererea este generată local doar de către „Publisher”.

4. **CPF 15 – MODBUS/TCP**, definit de către Schneider Electric și sprijinit de Modbus-IDA, utilizează standardul clasic industrial MODBUS, din 1979, dar suprapus pe o rețea TCP/IP, folosind portul 502 [17]. Aceasta este o soluție larg adoptată în aplicațiile industriale de viteză redusă, care necesită control uman.

Din punctul de vedere al protocolului, MODBUS este de tip cerere/răspuns și oferă servicii specificate prin coduri de funcții pentru citirea sau scrierea datelor care pot proveni de la diverse dispozitive conectate la rețeaua de teren. Fiind foarte simplu, acest protocol necesită completarea cu definiții de servicii noi pentru posibilitatea integrării în standardele internaționale [11]. Conform cerințelor IEC 61874, au fost definite extensii pentru operarea în timp real, bazate pe protocolul RTPS (Real-Time Publisher Subscriber) [18]. Acesta furnizează două modele de comunicație: „publish – subscribe” (publicare – subscriere) prin care se pun la dispoziție date pe magistrală fără specificarea unui anume destinatar, acestea fiind preluate de către oricare din dispozitivele conectate la rețea care le consideră de utilitate și protocolul CST (Composite State Transfer) prin care se transferă

informații de stare de la un dispozitiv „writer” (de scriere) la unul de tip „reader” (de citire).

Protocolul RTPS este proiectat să funcționeze pe suport UDP/IP care nu asigură un transport garantat al datelor, lipsind dialogul de confirmare („handshake”) sau corecția erorilor. Un mesaj reprezintă exact conținutul („payload”) unei singure datagrame („datagram”) UDP/IP. În literatura de specialitate, datagrama este definită ca un bloc de date care poate fi transmis fără stabilirea în prealabil a unui dialog de comunicare, așa cum rezultă și din [19], [20].

La ora actuală, protocolul RTPS nu are o răspândire prea largă în domeniul industrial și astfel nu se cunosc performanțele reale oferite de către acesta. Totuși există simulări descrise în [21] care confirmă încadrarea performanțelor MODBUS/TCP în clasa de cerințe definite de către IEC 61784-2.

5. **CPF 3 – PROFINET CBA** este sprijinit de către PROFIBUS International și definit de către mai mulți producători, prima versiune fiind bazată pe protocolul CBA (Component Based Automation) [22]. Valorile variabilelor de intrare/ieșire ale tuturor dispozitivelor conectate pe rețea sunt transmise prin conexiune standard TCP/IP cu utilizarea protocoalelor RPC (Remote Procedure Call) și DCOM (Distributed Component Object Model) regăsite în aplicații de birou [11]. Datorită acestui fapt, acest profil de comunicație se poate utiliza în cazul aplicațiilor ce nu necesită timpi de transfer mai mici de 100 ms (control uman). Totuși, pentru a obține timpi apropiați de 10 ms, PROFINET CBA comută în modul de exploatare a protocolului în timp real RT (Real Time), bazat pe un Ethertype special, cu prioritizarea cadrelor și ocolirea stivei TCP/IP [23].

Avantajul PROFINET CBA îl constituie posibilitatea utilizatorului de a configura în mod clasic dispozitivele PLC (Programmable Logic Controller) atașate rețelei, prin definirea conexiunilor logice între diferite componente. Astfel se poate specifica tipul datelor transmise precum și timpul necesar unei transmisii (acesta din urmă datorită coexistenței protocoalelor RT sau non RT).

6. **CPF 11 – TCnet (Time-critical Control Network)** este propus de către Toshiba [24]. Se bazează pe un protocol de acces CSMA/CD modificat față de standardul Ethernet uzual, fiind alocată o perioadă de transmisie cu viteză ridicată prin introducerea unui serviciu de date ciclic în timp real și a unui serviciu de mesaje asincron. Serviciul de date reprezintă un transfer orientat pe conexiuni punct – multipunct prestabilite pe aceeași legătură locală, separate prin rutere. Serviciul asincron permite mesajelor sporadice să străbată ruterele și să ajungă prin legături extinse la destinație [24].

TCnet permite gestionarea mediilor cu redundanță, stiva în timp real gestionând selecția a două intrări pentru recepția cadrelor și a două ieșiri către două medii de transmisie redundante. Astfel la apariția unei coliziuni pe unul din medii, transmisia este continuată pe celălalt, totodată acceptându-se și un prim cadru TCnet fără erori de transmisie să provină de pe un astfel de mediu redundant [24].

Procesele declanșate de aplicațiile care rulează pe fiecare nod de rețea folosesc în comun o memorie virtuală denumită „Common Memory”, definită de către nivelul de aplicație RTE-Tcnet.

Aceasta este împărțită în blocuri de diferite mărimi. Un anumit nod poate publica un bloc de date și îl poate transmite tuturor celorlalți participanți la rețea prin intermediul serviciului ciclic de date. Fiecare nod recepționează blocul de date (pe baza principiului de „abonat”) și își actualizează propria memorie comună cu aceste date, astfel asigurându-se un acces rapid la date prin citirea propriilor copii a memoriei comune [24].

7. **CPF 13 – Ethernet Powerlink (EPL)** a fost definit de către Bernecker și Rainer (B&R) și la ora actuală este susținut de către un grup de standardizare Powerlink Standardisation Group (EPSG) [25]. Schimbul de date se realizează pe baza unui sistem master – slave într-un segment Ethernet comun denumit "Slot Communication Network Management" (SCNM) [25]. Accesul în timp real la datele ciclice se asigură de către masterul cunoscut sub numele de „Managing Node” (MN) care totodată permite trecerea cadrelor TCP/IP standard doar în anumite ferestre temporale (time slots) atribuite acestor operații. Toate celelalte dispozitive conectate în rețea sunt considerate slave și se numesc „Controlled Nodes” (CN) și transmit date doar în urma unei cereri generate de către MN. Începutul unui ciclu se semnalează prin emiterea de către MN a unui cadru „Start of Cycle” (SoC) destinat tuturor CN iar durata de transmitere și recepționare a acestui cadru va dicta modul de sincronizare în comun a tuturor nodurilor (dispozitivelor) din rețea. În cadrul unui ciclu se pot distinge patru perioade distincte: Start, Isocronă, Asincronă și de latență. Durata fiecăreia poate să fie variabilă în limitele prestabilite ale ciclului EPL dar o deosebită importanță prezintă menținerea cât mai exactă a momentului de începere a fiecărui ciclu EPL [25].

Rețeaua EPL este privită ca un Ethernet „protejat” având un controler master (MN) și un anumit număr de dispozitive periferice slave, conectate la rețea (CN). Pentru a garanta funcționarea mecanismului de acces SCNM aferent MN, nodurile non EPL nu au permisiunea de a accesa această rețea. Mesajele vehiculate pe EPL sunt sincronizate cu ajutorul unui semnal de tact distribuit, conform protocolului IEEE 1588, având ca referință nodul MN [25]. Masterul gestionează și operațiile de rutare precum și alocarea sau translatarea adreselor IP ale dispozitivelor conectate, asigurându-se astfel și posibilitatea comunicării între diferite rețele de teren bazate pe EPL.

Standardul CANopen dezvoltat de către CAN in Automation (CiA) stă la baza nivelului de aplicație al EPL [26]. Acesta definește profiluri pentru comunicație, dispozitive și aplicații. Prin înglobarea CANopen în standardul EPL s-a atins posibilitatea coexistenței unui schimb de date performant cu protocoalele TCP/UDP/IP deja familiare rețelei clasice Ethernet. Aceste profiluri implementează controlul proceselor fizice prin definirea obiectelor de proces „Process Data Objects” (PDO) și indică comportamentul dispozitivelor din rețea prin intermediul unor parametri sau date de configurare denumite „Service Data Objects” (SDO). PDO sunt transmise prin comunicația EPL isocronă iar obiectele SDO cu ajutorul protocolului UDP/IP.

8. **CPF 14 – EPA (Ethernet for Plant Automation)**, elaborat în China, de către Zhejiang Supcon Co. Ltd. [27], reprezintă o abordare distributivă pentru a obține o comunicație deterministică bazată pe un mecanism de divizare temporală înglobat în nivelul de acces la mediu Media Access Control (MAC). Durata finalizării unei proceduri de comunicație este cunoscută sub numele de macrociclu. Acesta decurge în două etape: transferul periodic al mesajelor și transferul non – periodic. Ultimul fragment al transferului periodic va conține un anunț care specifică dacă dispozitivul în cauză urmează să transmită și un mesaj non – periodic și doar în această situație transferul menționat este permis [27].

Modelul protocolului la nivelul aplicației cuprinde procese EPA cu blocuri de funcții (algoritmi cu memorie statică asociată proprie) și procese non timp real ce pot rula în paralel într-o rețea EPA. Acestea din urmă sunt cele care utilizează protocoale Ethernet și TCP/IP standard. Comunicarea între două blocuri de funcții este privită ca o conectare a parametrilor de intrare/ieșire a serviciilor de aplicație EPA.

9. **CPF 3 – PROFINET IO**, sprijinit de către PROFIBUS International, este rezultatul definirii unui model de aplicație bazat pe cunoscutul PROFIBUS DP adoptat în standardul IEC 61158 [22]. Comunicația este ciclică iar un dispozitiv supervisor I/O (In/Out) gestionează toate celelalte echipamente periferice și controlere conectate la rețea.

Dispozitivele conectate realizează schimbul de date prin trei tipuri de servicii: isocron în timp real (Isochronous Real-Time – IRT), în timp real (Real-Time – RT) și non timp real (Non Real-Time – NRT). Acesta din urmă corespunde protocoalelor standard TCP/UDP/IP sau poate fi identic PROFIBUS CBA. Comunicarea Ethernet este împărțită în două cicluri, fiecare cu faze temporale diferite. În prima fază, denumită isocronă, sunt transmise toate cadrele IRT care trec prin comutatoarele Ethernet fără interpretarea informației legate de adresă. Aceste comutatoare operează conform unei tabele temporale prestabilite pe baza timpilor de așteptare definiți. Următoarea fază a transmisiei (etapa RT) constă în trecerea comutatoarelor la modul de comunicare bazat pe adrese și funcționarea clasică tip Ethernet. Totodată, cadrele RT sunt urmate de secvențele Ethernet NRT. Pentru a asigura mecanismul de comunicație descris, toate dispozitivele de comutare PROFINET sunt sincronizate prin intermediul unui algoritm IEEE 1588 particularizat, cu o corelare „ad-hoc”, impunându-se o deviere temporală de maximum o microsecundă între impulsurile de tact ale fiecărui dispozitiv [28].

Comunicația în timp real (RT) pentru PROFINET IO se poate realiza în condiții optime pe o structură fizică de rețea Ethernet full duplex de 100 Mb/s. Totuși, pentru schimbul de date în modul IRT este necesar un comutator dedicat PROFINET – Ethernet care trebuie de fapt integrat în fiecare dispozitiv conectat la rețea.

Conceptul fiind introdus la început în cazul INTERBUS (IEC 61158), modelul protocolului la nivelul aplicației implică existența unui dispozitiv proxy pentru posibilitatea integrării dispozitivelor bazate pe proxy în rețelele de teren existente. Astfel utilizatorul percepe în mod unitar atât dispozitivele conectate la Ethernet cât și cele la o anumită rețea de teren, acest lucru înlesnind înlocuirea treptată a sistemelor clasice bazate pe rețele de teren cu soluții moderne bazate exclusiv pe Ethernet.

10. **CPF 12 – ETHERCAT**, introdus de către Beckhoff și susținut de către grupul EtherCAT Technology Group (ETG), utilizează pachete Ethernet vehiculate într-o topologie inelară [29].

Un segment EtherCAT reprezintă de fapt un singur dispozitiv Ethernet care vehiculează cadre standard ISO/IEC 8802-3 (Ethernet). Totuși acest dispozitiv poate fi alcătuit dintr-un număr mare de periferice slave EtherCAT. Controlul accesului la mediu implică existența unui Master care trimite cadrele la dispozitivele Slave care pot executa operații de extragere sau inserare a datelor în aceste cadre. Pentru realizarea structurii în inel, ultimul dispozitiv Slave EtherCAT din segmentul de rețea trimite înapoi cadrul astfel procesat pentru ca apoi primul dispozitiv Slave să îl returneze nodului Master sub forma cadrului de răspuns. Standardul permite interconectarea nodului Master cu dispozitivele Slave fie direct, fie prin comutatoare Ethernet clasice, exploatând avantajele comunicării full duplex. Deși topologia fizică este una liniară, cadrele sunt transferate ciclic de la un dispozitiv către altul, ultimul nod returnând informația către Master tot cu ajutorul dispozitivelor (nodurilor) intermediare, formându-se astfel un inel logic. Procesarea și emiterea informației spre nodul următor din cadrele Ethernet se face gradual, pe parcursul recepționării treptate. Totodată, se pot identifica și executa comenzi adresate de către Master, chiar în timpul transmiterii cadrelor de la un nod la altul. Pentru a realiza acest

lucru, fiecare dispozitiv trebuie să dispună de un comutator integrat, cu două porturi de comunicație, implementat printr-un ASIC (Application-Specific Integrated Circuit) dedicat [29].

Mesajele EtherCAT pot fi dirijate direct spre locația alocată datelor din cadrul Ethernet sau integrate în datagrame UDP transportate prin adrese IP. Ele sunt fragmentate și apoi reconstituite la nodul receptor, această procedură neafectând timpilor de completare a unui ciclu de transmisie datorită disponibilității unei lățimi de bandă acoperitoare (algoritm EtherCAT în loc de fragmentare clasică IP) [29]. Prin acest algoritm se garantează participarea oricărui dispozitiv EtherCAT la traficul normal Ethernet fără a depăși durate de transfer de 100 μ s impuse de comunicația în timp real. În cazul eliminării traficului standard Ethernet se pot atinge și durate de aproximativ 30 μ s.

În cazul nivelului de aplicație, EtherCAT utilizează similar cu EPL specificațiile CANopen. Excepție fac obiectele SDO (vezi CPF 13 – EPL) care sunt dirijate către un mecanism tip casuță poștală de gestionare a mesajelor față de cel bazat pe protocol IP folosit în cazul EPL.

11. CPF 16 – SERCOS (Seriell Real time Communication System Interface) este bine cunoscut datorită interfeței sale optice de tip inel controlată computerizat (CNC – Computerized Numerically Controlled). Încă la ora actuală are propriul său standard, IEC 61491 [30]. Pe viitor se preconizează divizarea acestuia în partea de aplicație și în partea de comunicație cea de pe urmă fiind integrată în setul de specificații IEC 61158/IEC 61784 [31]. Totodată se urmărește compatibilizarea cu Ethernet de unde va deriva profilul SERCOS III [32].

SERCOS presupune existența unui singur Master și până la 254 dispozitive Slave cu rolul de controlere de mișcare pe diverse axe, care dispun de câte două porturi Ethernet independente, topologia fiind liniară sau în inel. Ultimul Slave dintr-o topologie liniară poate fi conectat la un comutator pentru comunicarea cu alte dispozitive prin protocoale TCP/IP sau UDP.

SERCOS III exploatează două canale logice de comunicație: canalul de timp real (RT) și canalul IP (standard Ethernet care nu este în timp real) [32]. Ciclul de comunicare este declanșat de către Master și constă din până la patru mesaje Master Data Telegrams (MDT) și patru Device Telegrams (AT – acknowledge telegrams) plasate în canalul RT respectiv IP. Scopul lor este obținerea sincronizării și transmiterea informației utile către dispozitivele Slave (comenzi, depanare etc.). Mesajele AT sunt transmise de către Master sub forma de cadre goale dar existând câmpuri predefinite fără informație în care fiecare Slave introduce datele necesare. Numărul și mărimea mesajelor RT sunt fixate printr-o procedură de inițializare. Totodată și durata de existență a canalului IP este fixată și determină numărul maxim de mesaje IP standard ce pot fi transmise pe această durată. Secvențele de sincronizare, RT și IP sunt reluate cu fiecare ciclu de comunicație care au durate specifice începând cu valoarea 31,25 μ s care se dublează succesiv până la maxim 65000 μ s. Aceste valori sunt transmise tuturor dispozitivelor Slave în timpul inițializării iar pentru atingerea unor performanțe relativ ridicate se impune implementarea fizică a algoritmilor cu ajutorul unor circuite FPGA (Field-Programmable Gate Array) care sunt capabile să separe canalul RT de canalul IP.

SERCOS a adoptat ca model al protocolului la nivelul aplicației algoritmul utilizat în conducerea sistemelor de acționare unde se regăsește un controller și mai multe actuatoare (servomotoare, relee etc.) între care are loc un schimb de date ciclic. În principiu, actuatoarele furnizează informație referitoare la stare și variabile ce descriu o anumită poziție iar Masterul emite comenzi către acestea. Astfel prin

stabilirea unor diverși parametri ai modelului se poate obține o anumită funcționalitate a întregului sistem.

În urma trecerii în revistă a caracteristicilor rețelelor de teren standardizate IEC, se poate constata diversitatea metodelor și specificitatea modului individual de implementare a acestor cerințe din standard (IEC 61784). Există diverse topologii în conjunctură cu diverse protocoale și servicii de transport a datelor care conduc practic la aceleași caracteristici funcționale dar la o totală incompatibilitate între dispozitivele specifice fiecărei rețele. Mai mult, unele protocoale propuse se află doar în stadiul de definire, neexistând dispozitive fizice de implementare. Se simte astfel lipsa unor soluții de reducere a multitudinii variantelor propuse pentru înlesnirea alegerii de către proiectant a rețelei de teren optimale în conducerea unui anumit proces tehnologic.

2.4. Concluzii și contribuții personale

Pentru conceperea unei viziuni de ansamblu asupra structurilor fundamentale de vehiculare a datelor în sistemele de comunicație industriale, este necesară în primul rând existența unei ierarhizări a tipurilor de rețele. În momentul de față se depun eforturi deosebite din partea organizațiilor internaționale de specialitate pentru dezvoltarea standardelor corespunzătoare rețelelor de teren, conturându-se doar un model principal care definește structura, modul de comunicare și tipul de protocoale ale unei astfel de rețele de teren.

Venind în sprijinul proiectantului, autorul tezei, în urma unui studiu amănunțit al literaturii de specialitate, a standardelor enunțate până în prezent de către organizațiile internaționale, pe parcursul acestui capitol, urmărește introducerea unor contribuții personale prin:

1. **Identificarea problematicii** actuale în alegerea și utilizarea sistemelor de comunicație industriale;
2. **Ierarhizarea** după cele mai noi criterii a tipurilor de rețele de teren;
3. **Sintetizarea** principalelor cerințe asupra caracteristicilor fundamentale ale rețelelor de teren;
4. **Clasificarea generală** a principalelor tipuri de rețele de teren după **criterii** de importanță majoră din **punctul de vedere** al autorului;
5. **Expunerea soluțiilor** adoptate în prezent în implementarea standardului IEC61784-2.

Prin aceste aspecte expuse în Capitolul II, se prefațează posibilitatea elaborării unei metode de analiză și evaluare a performanțelor rețelelor de teren existente la ora actuală. Se vor utiliza metode statistice de proiectare experimentală combinate cu simularea computerizată a comportamentului diverselor structuri de rețele pentru identificarea efectelor diverșilor parametri asupra eficienței rețelelor de teren.

3. ANALIZA ȘI PREDICȚIA INDICATORILOR DE PERFORMANȚĂ ALE SISTEMELOR BAZATE PE REȚELE DE TEREN

3.1. Introducere

„Design of Experiment” (DOE), întâlnită sub denumirea de proiectarea experimentelor sau proiectare experimentală [33], furnizează o metodă eficientă, având un cost redus, pentru analiza unei rețele de teren din punctul de vedere al influenței diversilor factori cum ar fi: topologia, banda de trecere, tipul protocolului de comunicație, viteza de transfer a datelor, redundanța etc. Acest procedeu de analiză, combinat cu un simulator software de rețele, poate veni în sprijinul proiectanților de sisteme de conducere a proceselor tehnologice, în alegerea rețelei de teren adecvate aplicației în cauză. În cele ce urmează, autorul propune o astfel de analiză dintr-o perspectivă originală, considerând-o contribuție proprie, utilă în selectarea rețelei de teren optimale pentru controlul unui anumit proces tehnologic.

Pentru a concepe și configura o rețea cât mai eficientă, este necesară identificarea factorilor care influențează în mod deosebit performanțele acesteia. La ora actuală există o diversitate de factori de influență și de aici rezultă problematica determinării prin simulare a efectelor induse, precum și modalitățile corecte de analiză a rezultatelor obținute. În literatura de specialitate, se pot identifica următoarele repere: determinarea metodei propice pentru proiectarea rețelei, modul de analiză a rezultatelor precum și procedura de determinare a principalilor factori de influență [34].

Predicția modului de operare a unei rețele de teren se poate realiza utilizând programe complexe de simulare, care conduc la reducerea substanțială a costurilor de instalare și testare funcțională. Datele furnizate de astfel de simulatoare vin în sprijinul proiectanților și permit evaluarea performanțelor înainte de o implementare fizică, așa cum este subliniat și în lucrarea [35]: „Obiectivul principal al simulării unei rețele de teren este de a prezice comportarea sistemului și de a identifica soluțiile optime pentru a minimiza costurile de proiectare cât și cele hardware”.

La ora actuală există nenumărate programe de simulare a rețelelor de teren, iar diverși cercetători sau proiectanți utilizează diferite tipuri de simulatoare, justificând în variate moduri alegerea celui mai potrivit. Printre cele mai cunoscute și performante simulatoare se pot menționa: NS2 Network Simulator, Real, GtnetS (The Georgia Tech Network Simulator), SSF (The Scalable Simulation Framework), OMNET++, HEGONS (HEterogeneous Grooming Optical Network Simulator), GloMoSim Simulation Engine, M5 Simulator, OpNeT Network Simulator, NetScale, NetSim, COMNET etc. Prin studierea unor lucrări de specialitate dedicate comparației unor astfel de simulatoare, se poate remarca afirmația din [36]: „Venerabilul simulator NS2 (McCanne și Floyd 1997) este cu siguranță cel mai popular și cel mai utilizat mediu de simulare pentru analiza rețelelor. Include modele detaliate a unor versiuni TCP, un număr mare de algoritmi de ordonare a traficului, diverse modele de aplicație (gen trafic web HTTP) precum și suport de urmărire și memorare a evenimentelor.” Totodată, despre OpNet se afirmă: „Simulatorul OpNet (Bertolotti și Dunand 1993) este un produs software comercial, cu o scară de utilizare foarte mare, elaborat de către OpNet Technologies Inc. Acest simulator conține modele

foarte detaliate a unui număr mare de dispozitive de rețea, incluzând routere, switch-uri și hub-uri precum și dispozitive wireless și protocoale MAC”.

O evaluare comparativă strictă între NS2 și OpNet se regăsește în lucrarea „OpNet Modeler and NS2: Comparing the Accuracy of Network Simulators for Packet-Level Analysis using a Network Testbed” [37]. Autorii evidențiază diferențele de performanță, acuratețe în furnizarea rezultatelor, viteza de simulare precum și varietatea ustensilelor puse la dispoziția utilizatorului.

Pentru analiza propusă în lucrarea de față, parcurgând documentația aferentă, autorul a decis pe baza criteriilor de performanță uzuale în domeniu, a facilităților oferite și nu în ultimul rând a ușurinței de utilizare și de concepere a unor noi modele descriptive a unor dispozitive sau protocoale de comunicație, selectarea simulatorului OpNet care va conduce la rezultate clare, precise, pentru rețelele de teren evaluate.

3.2. Analiza și proiectarea experimentelor

Modalitatea derulării unei proiectări experimentale este puternic influențată de alegerea algoritmului de investigare și de modul de colectare a datelor. Uzual, criteriile de selecție sunt îndreptate spre obținerea informațiilor din efectuarea a cât mai puține experimente precum și reținerea datelor realmente folositoare. Sunt cunoscute patru modalități de colectare a datelor pentru efectuarea unei analize: extragerea dintr-o bază de date, generarea unor date noi, efectuarea unor experimente specifice pentru a obține un anumit răspuns de la un sistem sau derularea experimentelor într-un mod structurat matematic.

DOE reprezintă o metodă alternativă foarte utilă la analiza experimentală a datelor [38] deoarece generează tabele de date structurate (conțin informație despre variație structurată) care pot sta la baza generării de modele robuste.

Elaborarea unei proiectări experimentale necesită alegerea unui număr relativ redus de experimente care vor fi derulate în anumite condiții și situații controlate, rezultate în urma parcurgerii a patru etape primordiale:

- definirea obiectivului investigației, prin identificarea variabilelor de importanță maximă și stabilirea condițiilor optime;
- definirea variabilelor care vor fi monitorizate pentru a descrie rezultatele experimentelor (variabilele de răspuns);
- definirea variabilelor care vor fi modificate (controlate) pe parcursul experimentelor precum și stabilirea domeniilor de variație ale acestora;
- selectarea unei metode standard de analiză (de obicei statistică) care este adecvată din punctul de vedere al numărului de variabile, a preciziei în furnizarea unui rezultat optim precum și a costului de implementare.

Rezultatele furnizate de către DOE vor consta într-o înșiruire de experimente bine definite, care trebuie efectuate de către proiectant pentru a obține informațiile necesare atingerii obiectivelor prestabilite prin analiza finală a datelor cu ajutorul unor metode statistice implementate prin software specializat.

Utilitarul JMP, proiectat de către compania americană SAS (denumire provenită inițial de la „Statistical Analysis System”), oferă posibilitatea de analiză avansată a datelor obținute în urma rulării experimentelor. Printre utilizatori se numără atât proiectanții din industria aeronautică, farmaceutică, de larg consum cât și manageri și operatori din industria alimentară.

În cazul DOE, programul JMP permite definirea în totalitate de către utilizator a modului de analiză a datelor, cu posibilitatea introducerii a diverse

scenarii, modele, criterii de optimizare precum și simulări accelerate pentru a evidenția anumite situații critice ce pot apărea în realitate.

Combinat cu simulatorul de rețele OpNet, JMP va furniza informații extrem de precise și utile în selectarea parametrilor optimi pentru alcătuirea unei rețele de teren specifice unei anumite aplicații și va permite corelarea de efecte introduse prin selectarea anumitor configurații.

3.2.1 DOE cu analiză factorială

La momentul de față există în literatura de specialitate, abordări ale acestei problematice prin metode statistice cum ar fi analiza factorială [34], dar care necesită un volum foarte mare de calcule și experimente pentru un număr relativ redus de factori. Considerând doar trei factori de influență, cu două niveluri de variație pentru fiecare, conform analizei factoriale din [34], rezultă un număr de 8 experimente obligatorii pentru a deduce rezultate concrete privind influența acestor factori asupra rețelei de teren.

Un exemplu concret, care ia în considerare trei factori, se regăsește în [34]. Tabelul 3.1 centralizează acești factori împreună cu nivelurile de variație pentru fiecare în parte.

Factori	Nivel inferior	Nivel superior
Viteza de comunicare	1Mb/s	2,5Mb/s
Topologia	Stea	Inel
Tipul dispozitivului de distribuire	Hub	Switch

Tabelul 3.1. Exemplu cu trei factori de influență

Criteriul de selectare a nivelurilor și a celor trei factori de influență rezultă dintr-o analiză în prealabil a modului de proiectare și configurare a rețelelor de teren uzuale precum și a principalelor caracteristici constructive și funcționale ale acestora.

Metoda de analiză adoptată este factorială completă, ceea ce conduce la 8 experimente (8 simulări ale diverselor combinații rezultate din tabelul 3.1). Tabelul 3.2 ilustrează ordinea și modalitatea de corelare a celor 8 variante. Pentru simplitate, nivelul inferior s-a notat cu „-1” iar cel superior cu „+1” iar cei trei factori cu A, B respectiv C.

Ordinea rulării experimentului	A	B	C
1	1	-1	1
2	-1	-1	-1
3	1	1	1
4	-1	1	-1
5	1	-1	-1
6	-1	1	1
7	1	1	-1
8	-1	-1	1

Tabelul 3.2. Variantele analizei factoriale complete

În urma simulării celor opt situații s-au obținut 8 pachete de date recepționate ca rezultat al generării a 8 pachete de date emise, pentru configurațiile

36 Analiza și predicția indicatorilor de performanță ale sistemelor bazate pe rețele de teren-3

de rețea ce rezultă din tabelul 3.2. Raportul dintre pachetele recepționate și cele emise reprezintă unitatea de răspuns analizată, care indică de fapt eficiența rețelei simulate. Tabelul 3.3 sintetizează rezultatele celor opt simulări.

Pachete de date	1	2	3	4	5	6	7	8
Recepționate	4798,3	3355,8	4849,2	4850	3355,8	4849,2	4851,7	4841,7
Trimise	4799,2	4844,2	4850,8	4852,5	4844,2	4850,8	4852,5	4843,3
Raport recepționat/trimis	0,9998	0,6928	0,9997	0,9995	0,6928	0,9997	0,9998	0,9997

Tabelul 3.3. Rezultatele simulărilor aferente analizei factoriale complete

Cu ajutorul unui utilitar software uzual („mini-tab”) au fost analizate aceste rezultate și s-au trasat câteva diagrame de probabilitate care ilustrează efectele și interacțiunea factorilor considerați în experiment.

Diagrama normală de distribuție de probabilitate a efectelor obținută pe baza rezultatelor celor opt simulări, prezentată în figura 3.1, definește factorul B (topologia) și factorul C (tipul distribuitorului) ca principalele efecte cu semnificație statistică de estimare a eficienței rețelei.

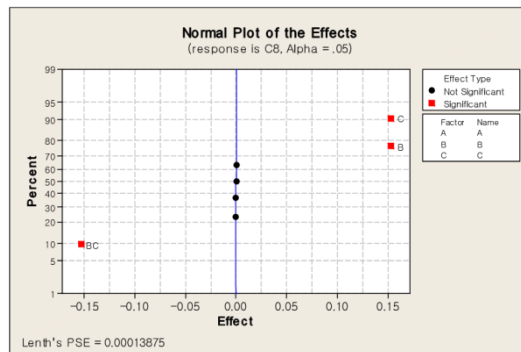


Fig. 3.1. Distribuția normală a probabilității efectelor

Totodată se observă o interacțiune între factorii B și C ceea ce indică o influență compusă a acestora asupra eficienței rețelei. Pe de altă parte, viteza de comunicare (factorul A) împreună cu alte interacțiuni care se regăsesc simbolizate de-a lungul liniei verticale din figura 3.1, se pot neglija.

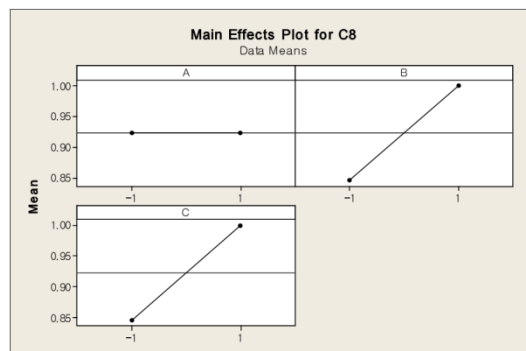


Fig. 3.2. Distribuția efectelor principale

Efectele principale semnificative ale factorilor considerați se pot deduce urmărind diagrama din figura 3.2. Topologia rețelei (factorul B) precum și tipul dispozitivului de distribuire (factorul C) influențează puternic eficiența rețelei: diagrama indică o eficiență mai mare a unei rețele cu topologie stea și dispozitiv hub față de o rețea cu topologie în inel și același dispozitiv hub. Figura 3.2 ne ilustrează că și în urma acestei analize, efectul vitezei de comunicare (factorul A) este neglijabilă din punctul de vedere al eficienței rețelei.

În experimentul analizat, pentru a observa interacțiunile dintre factori, mai precis dintre B și C, s-a generat diagrama de corelare a efectelor prezentată în figura 3.3.

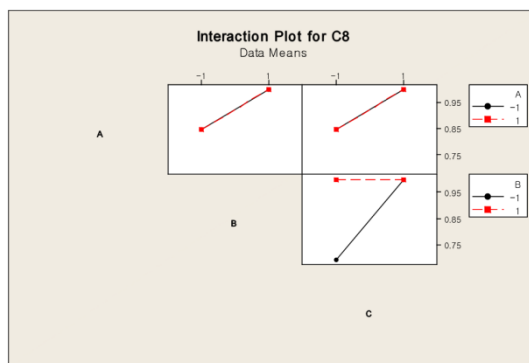


Fig. 3.3. Diagrama de corelare a efectelor

Se poate observa că o semnificație deosebită o reprezintă interacțiunea dintre topologia rețelei (factorul B) și tipul distribuitorului (factorul C). Eficiența unei rețele va fi întotdeauna relativ scăzută dacă va exista în proiect combinarea unei topologii de tip inel cu un distribuitor de tip hub (factorul B cu nivelul -1 și factorul C cu nivelul -1) [34].

3.2.2 DOE cu analiză D-optimal

Necesitatea utilizării analizei D-optimal apare în situațiile în care metodele clasice statistice nu satisfac în totalitate cerințele asupra obținerii unor rezultate suficient de clare sau conduc la efectuarea unor calcule voluminoase. Astfel, se poate recurge la această analiză dacă:

- factorii calitativi au mai mult de două niveluri;
- este necesară reducerea numărului de experimentări (simulări);
- se dorește includerea a unor rezultate experimentale deja obținute;
- există factori de proces compuși (interdependenți).

Pentru analiza rețelelor de teren este clară existența a mai mulți factori calitativi cu două sau chiar trei niveluri de variație și deci urmărirea reducerii numărului de experimente precum și a volumului de calcul rezultat de aici. În unele cazuri, deja există rezultate experimentale anterioare, care se doresc a fi incluse în noile analize – lucru imposibil pentru abordările clasice [39].

Factorii calitativi pot lua doar valori discrete neexistând un domeniu de variație continuu. Figura 3.4 ilustrează o analiză care necesită doi factori calitativi și unul cantitativ. Existând trei respectiv patru valori discrete pentru factorii calitativi, o analiză factorială completă ar necesita $3 \cdot 4 \cdot 2 = 24$ iterări pentru obținerea

rezultatelor. Abordarea D-optimal reduce numărul acestor iterații la 12, marcate în figura 3.4 cu cercuri negre, observându-se asigurarea unei analize distribuite în mod echilibrat pe întreg spațiul experimental („O analiză echilibrată are același număr de iterații pentru fiecare nivel al unui factor calitativ” [39]).

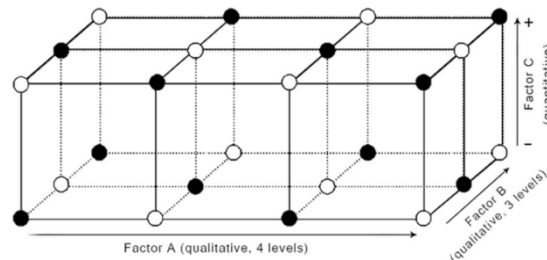


Fig. 3.4. Analiză D-optimal cu factori calitativi multinivel

În figura 3.4, factorii A și B sunt calitativi și au patru respectiv trei niveluri de variație discrete. Factorul C este cantitativ, cu valori cuprinse între -1 și +1. Punctele marcate cu negru reprezintă elementele de analiză selectate prin metoda D-optimal. Pentru o comparație și mai concretă, Tabelul 3.3 prezintă modificarea numărului de iterații funcție de numărul factorilor considerați în experiment și tipul analizei adoptate.

Factori	Factorial complet	Factorial fracțional	D-optimal
5	32	16	14
6	64	32	28
7	128	64	35
8	256	64	43
9	512	128	52

Tabelul 3.3. Număr de iterații și tipuri de analiză

Se poate observa că numărul de iterații utilizat în cazul analizei D-optimal este cel mai redus în comparație cu celelalte abordări și totodată acesta crește relativ mai lent odată cu mărirea numărului de factori [40].

Analiza D-optimal reprezintă în esență o proiectare asistată de calculator, conducând la un subset optim de experimente care trebuie efectuate.

Punctul de plecare reprezintă matricea setului de candidați, adică totalitatea experimentelor realizabile teoretic sau practic, în care „fiecare rând desemnează un experiment și fiecare coloană cite o variabilă” [40]. Aceasta va avea N rânduri (N reprezentând numărul experimentelor) și va fi notată ξ_N .

Vom considera un exemplu simplu, cu doi factori de influență, a_1 și a_2 . Matricea setului de candidați va fi ξ_4 , având patru rânduri și două coloane. Fiecare factor poate avea doar o valoare minimă sau maximă, notate pentru simplitate -1 respectiv 1. Relația (3.1) indică forma matricii aferente experimentului.

$$\xi_4 = \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix} \quad (3.1)$$

Următorul pas constă în alcătuirea matricii de proiectare, X , cu dimensiunile $n \times p$, ce derivă dintr-un model cu p coeficienți. Numărul rândurilor, n , se alege de către proiectant și reprezintă numărul experimentelor incluse în procesul de analiză. Urmează selecția unui model (liniar, patratic etc.) care utilizat împreună cu matricea ξ_4 va conduce la obținerea ușoară a matricii de proiectare X . Fiecare coloană va conține o combinație a factorilor din setul de candidați, dependentă de elementele modelului ales. Această matrice mai este întâlnită sub denumirea de matrice model [40].

Relația (3.2) exemplifică un model de regresie special, care poate fi utilizat în cazul analizei D-optimal, având în compoziție termenul de interacțiune adițional ε (variabilă aleatoare, independentă, cu valoarea medie 0 și varianță constantă).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \varepsilon \quad (3.2)$$

Considerând matricea din (3.1) și ecuația (3.2) vom obține matricea model X având patru rânduri și patru coloane, cu toate elementele din ξ_4 incluse în proiectare. De obicei, matricea candidaților conține un număr mult mai mare de elemente (mai multe experimente) iar matricea model va conține un set mult mai restrâns (mai puține rânduri și coloane decât în matricea ξ).

Astfel, vom avea matricea X :

$$X = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (3.3)$$

Alcătuirea matricii prezentate în (3.3) se bazează pe următorul principiu:

- prima coloană reflectă termenul constant β_0 , deci va conține doar cifra 1;
- următoarele două coloane conțin elementele matricii ξ_4 , descrise în (3.1);
- interacțiunea ambilor factori (x_1 și x_2), regăsită în ecuația (3.2) prin termenul $\beta_{12} x_1 x_2$, este inclusă în ultima coloană a matricii X , fiind produsul elementelor individuale.

Având la dispoziție un set mai mare de candidați, posibilitățile de grupare a elementelor ce descriu experimentele sunt mult mai vaste, ca urmare alcătuirea matricii X se va baza pe anumite criterii de optimizare speciale, cea mai bună combinație dintre acestea conducând la matricea de proiectare optimală, notată X^* [40].

Pentru a putea utiliza procedeul descris mai sus, este necesară definirea a încă două matrici și anume: matricea de informație ($X'X$) și matricea de dispersie $(X'X)^{-1}$. Se poate observa că ambele utilizează produsul dintre matricea X și transpusa sa X' . Utilizând ca bază teoretică metoda celor mai mici pătrate, pentru un model liniar de forma:

$$y = X\beta + \varepsilon \quad (3.4)$$

vom obține un set optimal de coeficienți dat de estimatorul $\hat{\beta}$:

$$\hat{\beta} = (X'X)^{-1}X'y \quad (3.5)$$

O analiză aprofundată a elementelor teoretice ale acestei metode se poate regăsi în literatura de specialitate uzuală, recomandat fiind volumul „Experiments: Planning, Analysis, and Parameter Design Optimization”, autori Wu & Hamada [41].

3.2.2.1 Criterii pentru cea mai bună proiectare D-optimal

În majoritatea cazurilor, problematica proiectării unui dispozitiv sau sistem se rezumă la stabilirea variantei optime a structurii care va fi apoi implementată. La fel și metoda D-optimal presupune stabilirea unor criterii de optimalitate, ele fiind axate pe maximizarea matricii informaționale ($X'X$). Se vor expune pe scurt aceste criterii, după care vom selecta metoda reprezentativă pentru determinarea numărului minim de experimente ce trebuie efectuate și care asigură informația necesară generării unui model statistic cu erori de predicție minimale.

D-optimalitatea (Determinantul)

Cel mai uzual criteriu urmărește maximizarea determinantului matricii informaționale ($X'X$) ce caracterizează experimentul. Se urmărește de fapt construirea variantei optime a matricii de proiectare, notate X^* , care va conține doar acele n experimente pentru care determinantul $|X'X|$ ia valoarea maximă. Semnificația acestui fapt este că „acele n experimente se extind peste cea mai mare regiune a spațiului experimental” [39]. Relația (3.6) indică acest lucru:

$$|X^{*'}X^*| = \max_{\xi_{n \rightarrow N}} (|X'X|) \quad (3.6)$$

Algebric, maximizarea determinantului în (3.6) este echivalentă cu minimizarea determinantului matricii de dispersie $(X'X)^{-1}$, deoarece:

$$|X'X| = \frac{1}{|(X'X)^{-1}|} \quad (3.7)$$

Corelarea exprimată prin relația (3.7) reduce semnificativ volumul de calcul aferent acestei analize [40].

A-optimalitatea

Acest criteriu implică operații asupra matricii de dispersie $(X'X)^{-1}$. Matricea de proiectare este considerată optimală atunci când urma matricii de dispersie are valoarea minimă, așa cum indică relația (3.8).

$$Tr(X^{*'}X^*)^{-1} = \min_{\xi_{n \rightarrow N}} (Tr(X'X)^{-1}) \quad (3.8)$$

unde

$$Tr(X'X)^{-1} = \sum_{i=1}^p a_{ii} \quad (3.9)$$

reprezintă urma matricii de dispersie $(X'X)^{-1}$, adică suma elementelor a_{ii} de pe diagonala principală, $p \times p$ fiind dimensiunea matricii. Minimizarea matricii semnifică în esență aflarea valorii minime a varianței medii a coeficienților estimați [40].

După cum indică și relațiile (3.8), (3.9), procesul de selecție a matricii de proiectare optimale implică un volum relativ mare de calcule și este mai rar adoptat ca și criteriu de optimizare.

V-optimalitatea (Varianța medie a predicției)

Considerând vectorul ζ_i care descrie un singur experiment i și ζ_i' transpusa acestuia, putem exprima prin relația (3.10) varianța predicției unui singur candidat, cu alte cuvinte putem afla incertitudinea răspunsului estimat [40].

$$d(\zeta_i) = \zeta_i' * (X'X)^{-1} * \zeta_i \quad (3.10)$$

În urma aplicării criteriului V-optimal, candidații selectați vor avea cea mai mică valoare a varianței medii de predicție, exprimată prin relația (3.11).

$$\frac{1}{n} \sum_{i=1}^n \zeta_i' * (X'^*X^*)^{-1} * \zeta_i = \min_{\xi_{n \rightarrow N}} \left(\frac{1}{n} \sum_{i=1}^n \zeta_i' * (X'X)^{-1} * \zeta_i \right) \quad (3.11)$$

Analizând relațiile de mai sus și parcurgând literatura de specialitate [38-41], putem incadra și acest criteriu în categoria „dificil de implementat”, necesitând atât un volum de calcul ridicat cât și algoritmi sofisticăți în cazul actualizării cu noi parametri de intrare.

G-optimalitatea

Un ultim criteriu de optimizare ia în considerare tot varianța predicției setului de candidați, matricea de proiectare optimală fiind selectată în vederea minimizării varianței maxime a predicției din cadrul analizei. Relația (3.12) reflectă acest lucru.

$$\max(\zeta_i' * (X'^*X^*)^{-1} * \zeta_i) = \min_{\xi_{n \rightarrow N}} (\max(\zeta_i' * (X'X)^{-1} * \zeta_i)) \quad (3.12)$$

În majoritatea cazurilor însă, criteriul de G-optimalitate nu este utilizat pentru determinarea celui mai bun proiect, dar în schimb este util pentru a identifica cel mai bun proiect dintr-un set obținut prin aplicarea altor criterii, ca de exemplu D-optimal [40]. Astfel, s-a introdus noțiunea de G-eficiență, definită prin:

$$G_{eff} = 100\% * \left(\frac{p}{n * d_{max}(\zeta)} \right) \quad (3.13)$$

Pentru relația (3.13) avem: p numărul coeficienților din modelul ales, n numărul de iterații (de experimente care trebuie efectuate) și $d_{max}(\zeta)$ valoarea maximă a varianței de predicție a matricii model X . Principial, G-eficiența reflectă o comparație între o proiectare D-optimal și o proiectare factorial fracțional, exprimată în procente [40].

Numărul de condiție

Pentru o posibilă clasificare a unei proiectări D-optimal, se utilizează un criteriu de evaluare asemănător G-eficienței, cunoscut sub numele de număr de condiție. Acesta indică sfericitatea și simetria proiectării D-optimal prin calculul „raportului dintre cea mai mare și cea mai mică valoare singulară” a matricii X [39]. Parametrul $Cond(X)$ este limitat inferior la valoarea 1 ceea ce semnifică ortogonalitate iar o creștere a valorii va indica un proiect mai puțin ortogonal.

3.3. Analiza structurală și funcțională a rețelelor de teren de tip IEC 61784 prin metoda D-optimal

Vom propune în continuare o metodă de analiză și predicție a valorilor unor indicatori de performanță a rețelelor de teren bazată pe proiectarea D-optimal, descrisă în paragraful precedent, adaptată cerințelor impuse de conceperea rețelelor de teren industriale. Metoda reprezintă o abordare nouă a problematicii existente la ora actuală în domeniu, urmărindu-se obținerea unor rezultate utile atât din punctul de vedere al proiectantului, cât și al utilizatorului final.

Pe baza sintezei din Capitolul II, am identificat cinci potențiali factori de influență semnificativi în funcționarea rețelelor de teren, prezentați în Tabelul 3.4.

Factori	Simbol	Nivel -1	Nivel 0	Nivel 1
Topologia	A	Stea	Liniară	Inel
Tipul dispozitivului de distribuire	B	Switch	---	Hub
Viteza de maximă de comunicare	C	10Mb/s	---	100Mb/s
Viteza de procesare a pachetelor	D	250 μ s	25 μ s	2,5 μ s
Protocol	E	TCP	---	UDP

Tabelul 3.4. Factori de influență semnificativi

Vom urmări două criterii de răspuns, considerate semnificative în literatura de specialitate [19], [30], [34], eficiența transmisiei definită ca raportul între numărul total de pachete recepționate și numărul total de pachete trimise precum și întârzierea maximă introdusă de rețea, Δ_{max} , exprimată în milisecunde.

Relația (3.14) definește eficiența transmisiei, η_T :

$$\eta_T = \frac{R_p}{S_p} \quad (3.14)$$

unde:

R_p reprezintă numărul total de pachete recepționate pe durata simulării experimentului;

S_p reprezintă numărul total de pachete trimise pe durata simulării experimentului.

Analiza globală a structurii rețelei precum și modelul de predicție va lua în considerare ambele criterii de răspuns menționate.

3.3.1 Proiectarea inițială utilizând JMP

Pasul următor constă în definirea factorilor din tabelul 3.4 prin intermediul utilitarului JMP și generarea matricii setului de candidați care va conduce la stabilirea numărului de experimente necesar a fi efectuat (a numărului de iterații).

Proiectul JMP va fi denumit „Analiza” și va fi de tipul DOE, Custom Design. Fereastra inițială permite introducerea atât a factorilor cât și a criteriilor de răspuns, într-un mod intuitiv, cu diverse opțiuni de clasificare și integrare. Figura 3.5 ilustrează punctul de plecare în definirea factorilor de influență, explicitând modul de utilizare al meniurilor aferente.

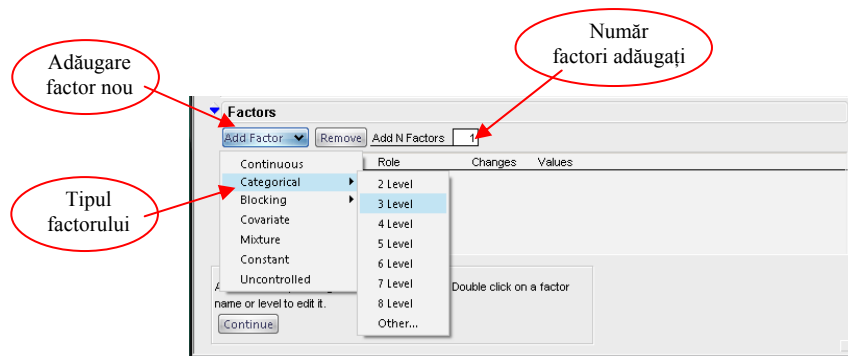


Fig. 3.5. Fereastra de selecție a factorilor de influență

În cazul nostru considerat, toți factorii sunt de tip „Categorical”, cu două sau trei niveluri de variație. Introducând elementele expuse în Tabelul 3.4, vom obține forma finală reprezentată în figura 3.6.

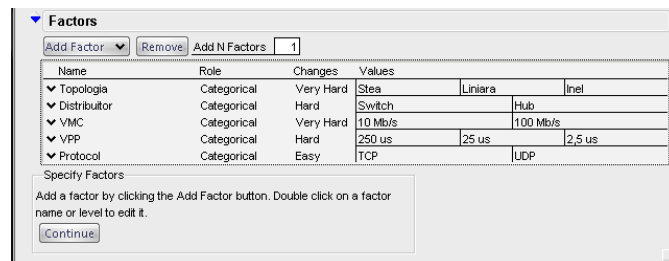


Fig. 3.6. Factorii de influență definiți în JMP

Se observă în figura 3.6 gruparea sintetică a factorilor considerați, precum și existența unui parametru suplimentar, reflectat în coloana denumită „Changes” care permite specificarea ușurinței de modificare în cadrul experimentului a parametrului în cauză. Există trei niveluri: „Easy” (ușor), „Hard” (greu) și „Very Hard” (foarte greu). Am considerat o modificare a topologiei și a vitezei de comunicare a mediului de transmisie foarte greu de realizat în cazul rețelelor fizice.

Urmează definirea factorilor de răspuns, fereastra inițială fiind cea din figura 3.7.

44 Analiza și predicția indicatorilor de performanță ale sistemelor bazate pe rețele de teren-3

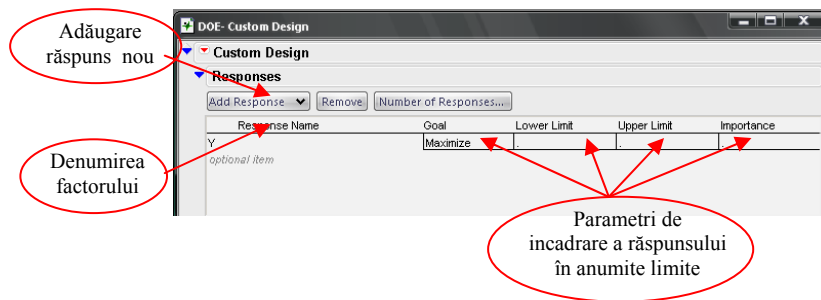


Fig. 3.7. Fereastra de definire a factorilor de răspuns

Pentru optimizarea analizei, există posibilitatea precizării limitelor de variație a factorilor de răspuns, precum și ținta către care se dorește să tindă aceste valori, așa cum se observă în figura 3.7.

Pe baza criteriilor de răspuns stabilite în procesul de optimizare urmărit, vom obține factorii grupați în fereastra JMP din figura 3.8.

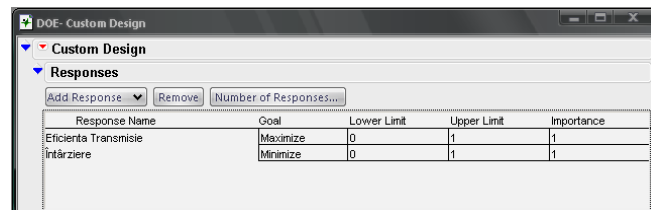


Fig. 3.8. Factorii de răspuns definiți în JMP

În urma generării automate a matricii setului de candidați (conform exemplificării prin relația (3.1)), se obține rezultatul din figura 3.9.

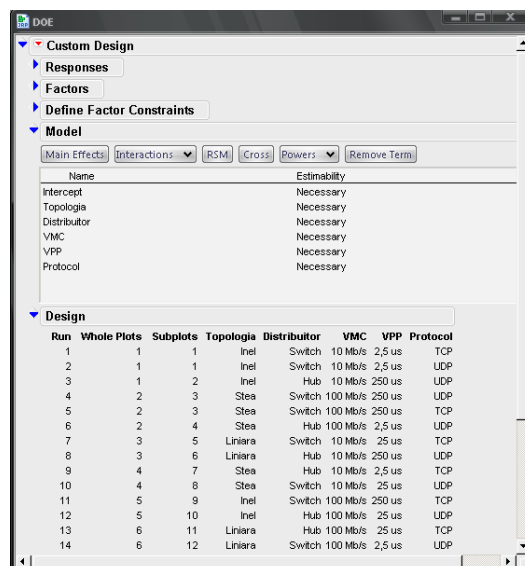


Fig. 3.9. Setul de candidați generat cu JMP

Modelul selectat va reflecta efectele principale introduse de factorii de influență, așa cum se poate remarca și în figura 3.9, urmărind rubrica „Model”.

După cum rezultă din cele de mai sus, vom avea o matrice ξ_{14} , deci un număr total de 14 experimente de realizat, față de varianta clasică a analizei factoriale complete, pentru care ar fi rezultat un număr de $3*2*2*3*2*5=360$ de experimente.

Expresia matricii ξ_{14} este redată prin relația (3.15).

$$\xi_{14} = \begin{bmatrix} 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & -1 & -1 \\ -1 & 1 & 1 & 1 & 1 \\ 0 & -1 & -1 & 0 & -1 \\ 0 & 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 0 & 1 \\ 1 & -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & -1 & 1 & 1 & 1 \end{bmatrix} \quad (3.15)$$

Suplimentar, JMP ne oferă câteva informații referitoare la profilul varianței predicției, asupra curbei de variație relativă a varianței predicției, precum și a varianței relative a coeficienților modelului.

Varianța predicției reprezintă produsul între varianța erorii și o mărime ce depinde de metoda de proiectare și factorii selectați. Înaintea începerii colectării datelor, atât varianța erorii cât și varianța predicției sunt necunoscute. Totuși, raportul dintre varianța predicției și varianța erorii nu depinde de varianța erorii. Acest raport, denumit varianța relativă a predicției, este influențat doar de particularitățile proiectării și a selecției factorilor de influență și poate fi calculat înaintea colectării datelor.

Profilul varianței de predicție ilustrează grafic varianța relativă a predicției funcție de fiecare factor în parte, cu valori fixe pentru ceilalți factori de influență.

După efectuarea (simularea) tuturor experimentelor indicate de matricea (3.15) și colectarea datelor, se va putea estima varianța reală a predicției prin înmulțirea varianței relative cu eroarea medie patrată rezultată.

Vom urmări ca varianța predicției să fie cât mai redusă ca valoare și să se încadreze în limitele admise de către factorii de influență. Comparând profilurile a două proiectări putem obține informații despre gradul de optimalitate a acestora, fiind preferabil proiectul care prezintă o varianță cât mai redusă.

În urma analizei cu ajutorul utilitarului JMP, am obținut profilul prezentat în figura 3.10.

Figura 3.10 ilustrează profilul pentru valorile implicite, rezultat fără aplicarea criteriului de maximizare a varianței.

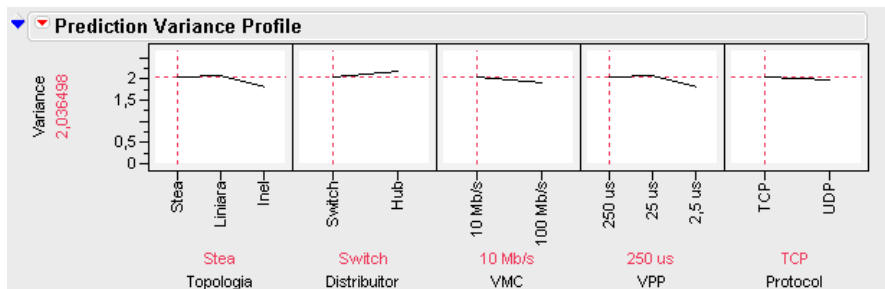


Fig. 3.10. Profilul varianței de predicție

Analizorul inclus în JMP permite vizualizarea varianței maxime a predicției pentru identificarea cazului (scenariului) cel mai defavorabil rezultat din combinația anumitor factori, prin apelarea opțiunii „Maximize Desirability”. Prin urmare, obținem profilul pentru cazul cel mai defavorabil, expus în figura 3.11.

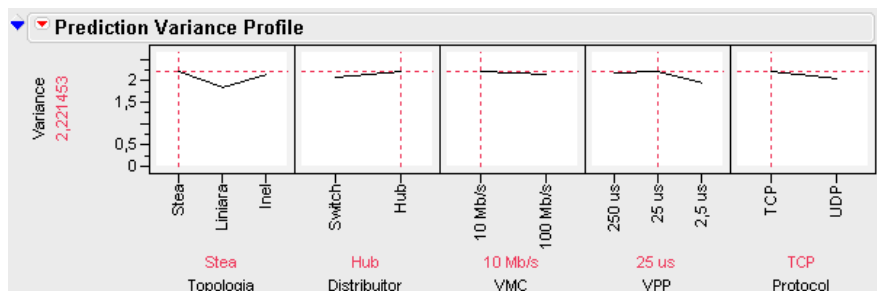


Fig. 3.11. Profilul varianței de predicție pentru scenariul cel mai defavorabil

O modalitate simplă de evaluare a limitelor în care se încadrează această varianță a predicției este de a urmări curba de variație ilustrată în figura 3.12, cu raportare procentuală la întreg spațiul experimental (0 – 100%).

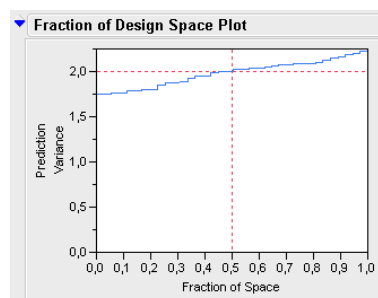


Fig. 3.12. Varianța predicției în spațiul experimental

Curba din figura 3.12 este utilă în cazul existenței factorilor de influență multipli, ea exprimând influența cumulată a acestora.

Varianța relativă a coeficienților este raportată la varianța erorii care fiind necunoscută înaintea efectuării experimentelor, ia valoarea 1. În urma efectuării

tuturor experimentelor, având un estimat pentru varianța erorii, acesta se poate înmulți apoi cu valoarea varianței relative pentru a obține în final varianța estimată a coeficienților în parte.

În momentul validării modelului (prin opțiunea „Analyze-Fit Model”), eroarea standard de predicție a coeficientului în cauză trebuie să fie egală cu rădăcina patrată a valorii varianței estimate.

Un alt parametru util se găsește în coloana „Power”, indicând puterea testului (în cazul nostru capacitatea proiectării selectate de a identifica anumite efecte introduse de factorii de influență). Intervin aici două elemente importante: *nivelul de semnificație* (definit apriori de către proiectant) și *mărimea efectului* (denumit raport semnal-zgomot în JMP) care oferă o informație suplimentară în interpretarea rezultatului proiectării, cu apropiere majoră de semnificația practică. Ambii termeni sunt bine cunoscuți în literatura de specialitate din domeniul statisticii [42], [43].

Pentru nivelul de semnificație (α) se acceptă în mod uzual valoarea 0,05 care conduce la obținerea unor rezultate satisfăcătoare în urma rulării testului (a proiectării) [42], [43].

Puterea testului va prezenta valori relativ mai ridicate în cazul în care vom stabili o valoare mai mare pentru mărimea efectului. Aceasta din urmă reprezintă raportul dintre valoarea absolută a parametrului de regresie și σ (deviația standard). Cu alte cuvinte, mărimea efectului permite identificarea oricărui coeficient de regresie care ar fi de m ori mai mare decât deviația standard, m exprimând valoarea mărimii efectului introdusă în rubrica adecvată în JMP.

Figura 3.13 redă tabelul varianței relative a coeficienților pentru valorile impuse ale nivelului de semnificație respectiv a mărimii efectului.

Relative Variance of Coefficients		
Significance Level	0,050	
Signal to Noise Ratio	1,500	
Effect	Variance	Power
Intercept	0,328	0,592
Topologia 1	0,326	0,594
Topologia 2	0,344	0,573
Distribuitor	0,163	0,870
VMC	0,326	0,594
VPP 1	0,178	0,839
VPP 2	0,197	0,803
Protocol	0,118	0,950

Fig. 3.13. Varianța relativă a coeficienților

Observăm ca valoarea puterii testului este mai mare de 0,5 pentru toți parametrii, asigurându-se astfel o putere de discriminare suficient de mare între decizia corectă și eroare [43].

Următorul pas îl reprezintă generarea tabelului de proiectare în care în urma rulării celor 14 experimente (prin simularea cu OpNet) se vor înscrie valorile factorilor de răspuns definiți. Figura 3.14 ilustrează forma acestui tabel generat cu ajutorul utilitarului JMP.

Modelul proiectării este descris printr-un „Script” JMP, pe care îl putem rula („Run Script”) pentru a vizualiza apoi fereastra de dialog „Fit Model” care va permite validarea și optimizarea modelului selectat.

	Whole Plots	Subplots	Topologia	Distribuitor	VMC	VPP	Protocol	Eficienta Transmisie	Întârziere
1	1	1	Inel	Switch	10 Mb/s	2,5 us	TCP	*	*
2	1	1	Inel	Switch	10 Mb/s	2,5 us	UDP	*	*
3	1	2	Inel	Hub	10 Mb/s	250 us	UDP	*	*
4	2	3	Stea	Switch	100 Mb/s	250 us	UDP	*	*
5	2	3	Stea	Switch	100 Mb/s	250 us	TCP	*	*
6	2	4	Stea	Hub	100 Mb/s	2,5 us	UDP	*	*
7	3	5	Linara	Switch	10 Mb/s	25 us	TCP	*	*
8	3	6	Linara	Hub	10 Mb/s	250 us	UDP	*	*
9	4	7	Stea	Hub	10 Mb/s	2,5 us	TCP	*	*
10	4	8	Stea	Switch	10 Mb/s	25 us	UDP	*	*
11	5	9	Inel	Switch	100 Mb/s	250 us	TCP	*	*
12	5	10	Inel	Hub	100 Mb/s	25 us	UDP	*	*
13	6	11	Linara	Hub	100 Mb/s	25 us	TCP	*	*
14	6	12	Linara	Switch	100 Mb/s	2,5 us	UDP	*	*

Fig. 3.14. Tabelul de proiectare

3.3.2 Modelarea rețelelor și simularea experimentelor cu OpNet

Cele 14 experimente vor consta în elaborarea celor 14 configurații de rețele din tabelul de proiectare și simularea funcționării acestora cu ajutorul mediului OpNet, amintit în paragraful 3.2. O justificare a utilizării OpNet se poate regăsi în lucrarea [37], unde este prezentată o metodă de verificare practică și teoretică a capacității acestui mediu de simulare de a furniza rezultate corecte pentru rețele industriale de tip Ethernet.

Experimentele se vor modela și simula fiecare în parte, prin gruparea pe proiecte diferite, gestionabile cu mediul OpNet. Rezultatele se vor obține atât grafic cât și sub formă tabelară de unde vom extrage prin calcul factorii de răspuns definiți anterior.

Structura rețelelor se va întinde pe o suprafață de 250m x 150m, cu cinci noduri principale la care sunt conectate câte zece stații de lucru (acestea se pot echivala cu diverse periferice gen transductor inteligent).

Modelele de tehnologii incluse în fiecare proiect sunt: Client – Server, Ethernet, Ethernet_Advanced, Links, Token Ring și Utilities. Pentru o analiză mai aprofundată a acestor elemente se recomandă parcurgerea materialului bibliografic [34], [35], [36], [37].

3.3.2.1 Experimentul 1

Structura rețelei va fi în acest caz: topologie inel, distribuitor switch, viteza maximă a mediului de comunicare 10 Mb/s, viteza de prelucrare a pachetelor 2,5 μs, protocol TCP.

Figura 3.15 ilustrează nodurile rețelei principale precum și elementele suplimentare care definesc aplicația și modul de vehiculare a datelor.

OpNet notează componentele din rețea cu denumirea generică „node” (nodurile rețelei) iar calea de acces spre nivelurile inferioare cu „subnet”.

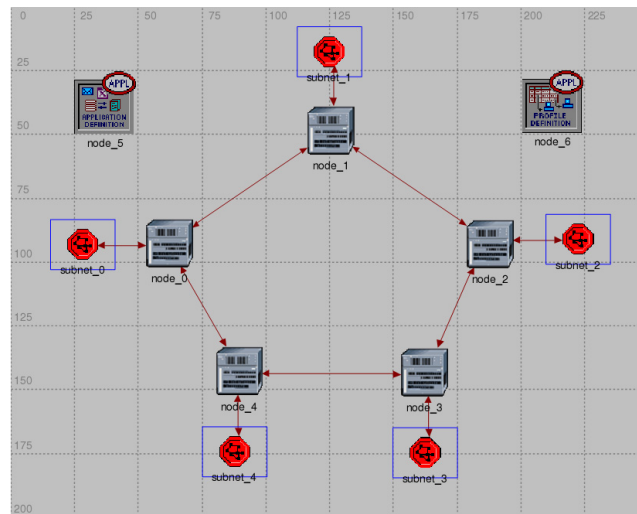


Fig. 3.15. Rețeaua principală pentru Experimentul 1

Distingem în figura 3.15 următoarele elemente:

- node_0...node_4: distribuitorul (factorul B), de tip Ethernet Switch, cu 16 porturi I/O;
- subnet_0...subnet_4: rețele secundare conectate la câte un distribuitor;
- node_5: element special, de tip utilitar, care definește caracteristicile aplicației;
- node_6: element special, de tip utilitar, care definește profilul aplicației preluat apoi de către stațiile de lucru;
- legăturile între noduri de realizează prin intermediul liniilor de rețea marcate cu săgeți (de tip 10BaseT pentru Experimentul 1).

Caracteristicile aplicației se definesc în prima instanță, prin utilitarul „Application Definition”, figura 3.16 expunând atributele specifice care intervin în simularea în cauză.

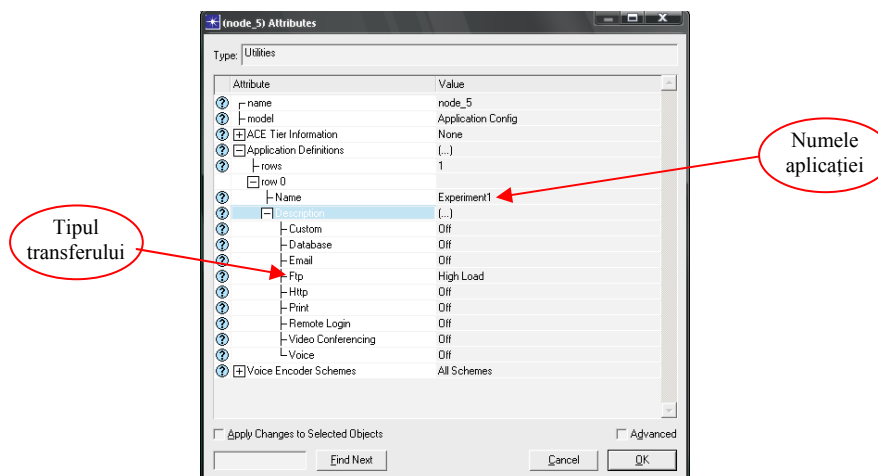


Fig. 3.16. Definirea parametrilor aplicației

Denumirea aplicației va fi „Experiment 1” iar tipul transferului s-a ales „Ftp” (File Transfer Protocol) care permite acces de tip „client – server” necesar pentru simularea rețelelor de teren dorite (vezi Capitolul II).

„Profile Definition” ajută la alcătuirrea profilurilor utilizator pentru a fi folosite de către nodurile rețelei de a genera traficul la nivelul aplicației. Elementele definite în „Application Definition” sunt utilizate pentru configurarea profilurilor. Pot exista în cadrul aceluiași proiect mai multe profiluri selectabile la nivel de nod de rețea. Atributele importante ale „Profile Definition” care definesc primul experiment sunt redată în figura 3.17.

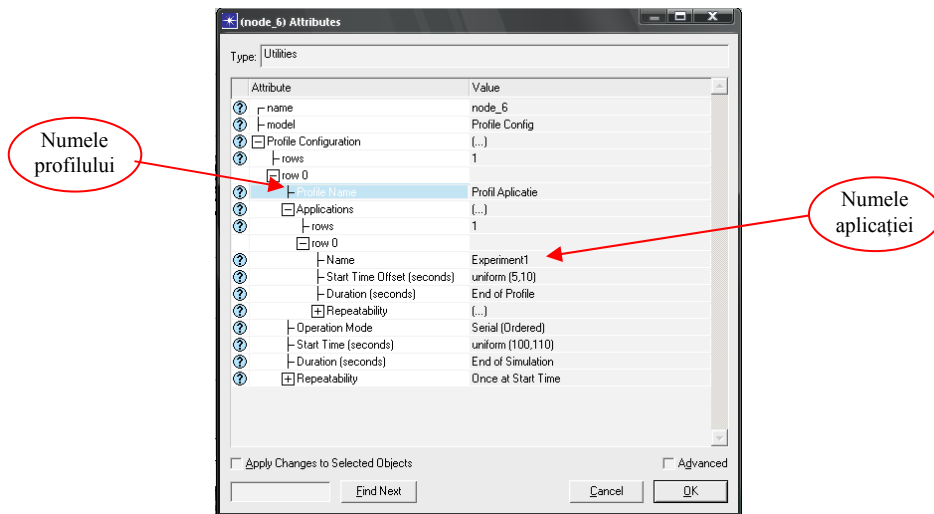


Fig. 3.17. Definirea profilului aplicației

Fiecare distribuitor de tip switch permite accesul la câte o rețea secundară (subnet_0,..., subnet_4) cuprinzând zece stații de lucru (node_0,...,node_9) interconectate tot prin intermediul unui switch, așa cum se observă în figura 3.18.

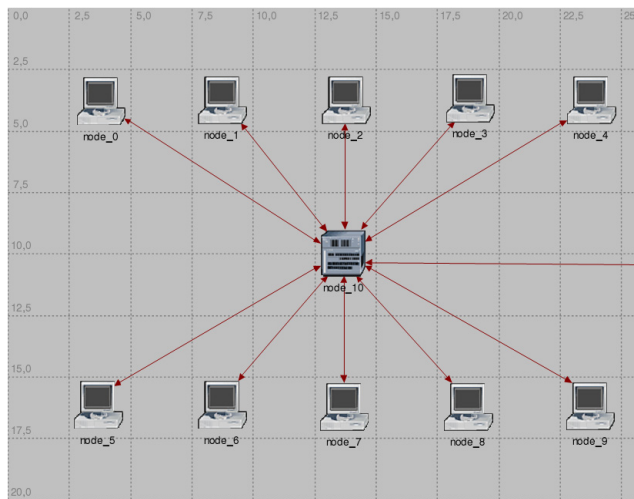


Fig. 3.18. Structura unei rețele secundare

Fiecare stație de lucru poate reprezenta un PLC (Programmable Logic Controller) sau chiar un transductor inteligent (Smart Transducer) potrivit conceptelor actuale referitoare la prelucrarea informației în cadrul sistemelor de conducere a proceselor tehnologice [44]. Factorii de influență D și E se stabilesc la nivelul acestor noduri, prin elementele „Processing Speed Multiplier” și respectiv „Application: Transport Protocol Specification”. Totodată aici se stabilește și profilul utilizat de către stație prin „Application: Supported Profiles”. Figura 3.19 ilustrează fereastra de atribute pentru una din cele zece stații de lucru (toate au atribute identice).

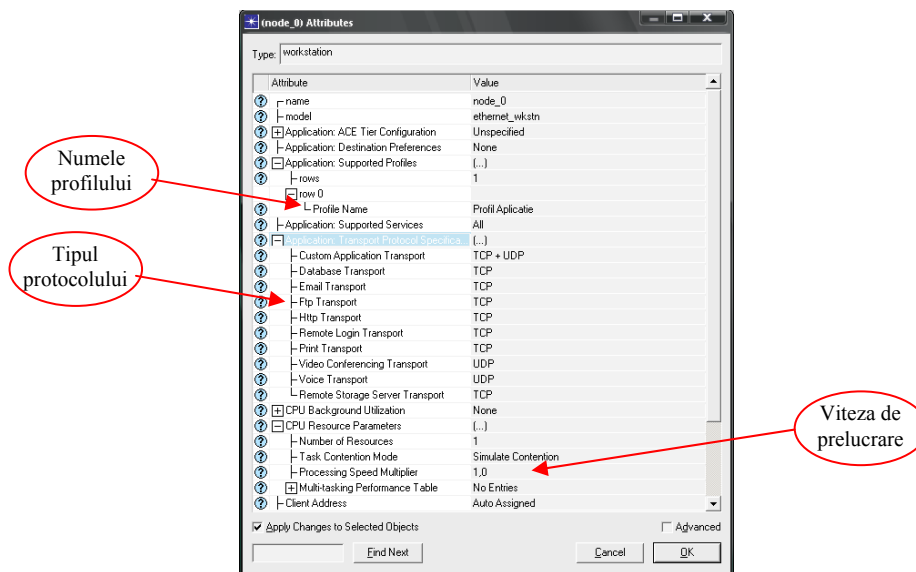


Fig. 3.19. Atributele semnificative aferente unei stații de lucru

Ca rezultate ale simulării vom urmări graficul variației pachetelor de date trimise, a pachetelor de date recepționate (pachete/s) și întârzierea introdusă de configurația selectată.

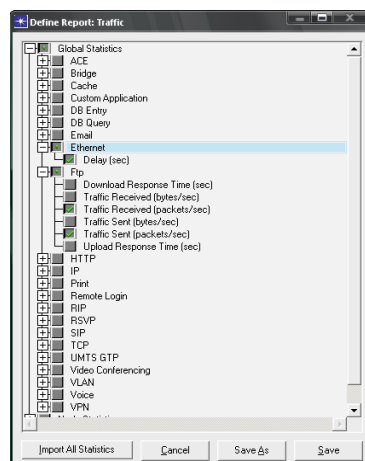


Fig. 3.20. Selectarea rezultatelor simulării

Toate aceste elemente se pot alege prin intermediul unui meniu intuitiv („Simulation-Define Report”), conform ferestrei prezentate în figura 3.20.

Ultimul pas înainte de rularea simulării proiectului este de a stabili în fereastra „Simulation-Configure Discrete Event Simulation” durata de simulare (s-a adoptat valoarea 60 minute), numărul inițial pentru generarea valorilor aleatoare („Seed” =128), numărul de eșantioane pentru durata simulării („Values per statistic” =100) precum și numărul evenimentelor după care se va face reactualizarea ferestrei de urmărire a desfășurării simulării („Update Interval” =100000).

Opțiunea „Simulation-Run Discrete Event Simulation” permite începerea simulării proiectului întocmit prin parcurgerea etapelor prezentate anterior, aferente cerințelor pentru Experimentul 1.

Mediul OpNet ne oferă posibilitatea urmăririi desfășurării procesului de simulare, în mod grafic, ilustrând numărul de evenimente simulate pe secundă, atât în valoare momentană cât și în valoare medie, așa cum se poate observa și în figura 3.21.

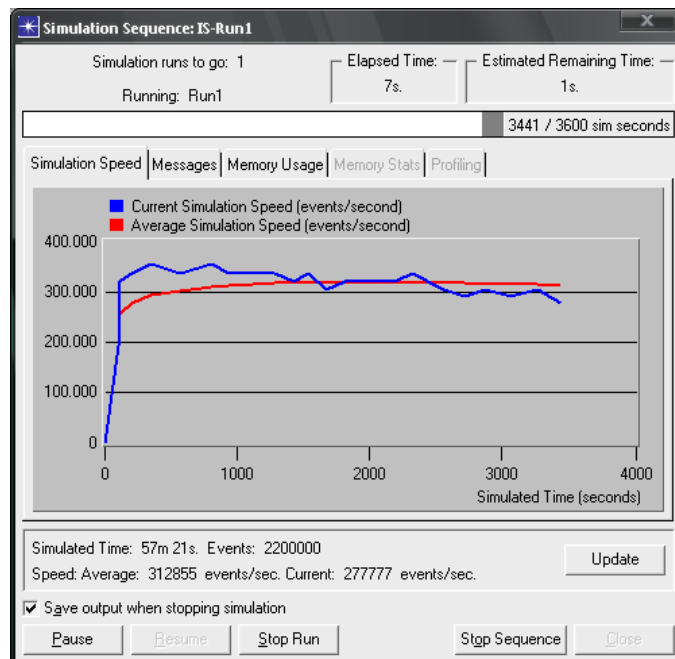


Fig. 3.21. Derularea procesului de simulare

Rezultatele le vom vizualiza în mod grafic, iar apoi vom genera tabele cu valori ale factorilor de interes.

Anexa 1 va reuni întreg setul de valori pentru cele 14 experimente derulate.

Figura 3.22 redă variația traficului pe rețea, prin ilustrarea pachetelor trimise și a celor recepționate.

Întârzierea introdusă de structura rețelei Ethernet definită se poate urmări în figura 3.23.

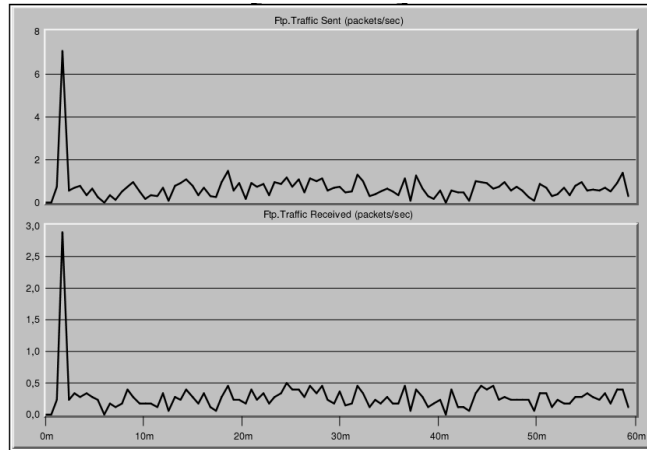


Fig. 3.22. Traficul pe rețea – Experimentul 1

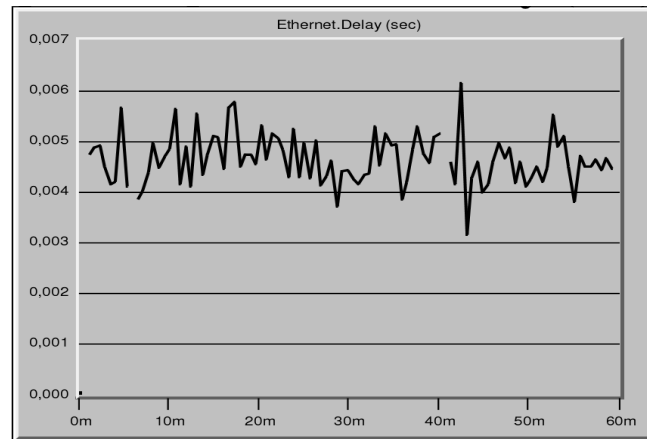


Fig. 3.23. Variația întârzierii – Experimentul 1

Factorii de răspuns care ne interesează au fost extrași din tabelul prezentat în Anexa 1.

Pentru Experimentul 1 am obținut $S_p=66,5$, $R_p=27$ iar valoarea maximă a întârzierii $\Delta_{1max}=6,16$ ms. Eficiența transmisiei, η_{T1} , conform relației (3.14) va fi:

$$\eta_{T1} = \frac{27}{66,5} = 0,406$$

3.3.2.2 Experimentul 2

Se va păstra aceeași structură ca și la primul experiment, cu modificarea factorului E (protocolul), astfel încât pentru transportul de tip Ftp vom avea protocol UDP.

În urma simulării, graficul variației traficului pe rețea are forma expusă în figura 3.24.

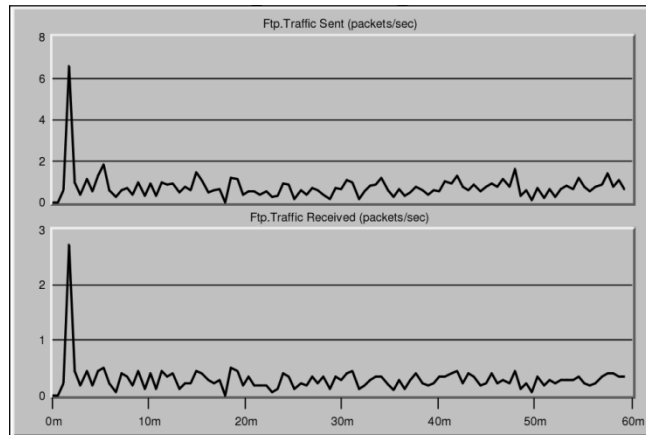


Fig. 3.24. Traficul pe rețea – Experimentul 2

Întârzierea introdusă de structura rețelei Ethernet, cu protocol de transport UDP, se poate urmări în figura 3.25.

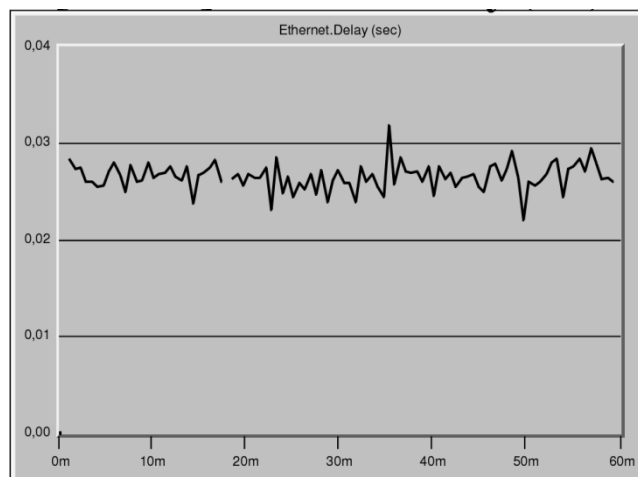


Fig. 3.25. Variația întârzierii – Experimentul 2

Consultând Anexa 1, pentru Experimentul 2 am obținut $S_p=70,94$, $R_p=28,94$ iar valoarea maximă a întârzierii $\Delta_{2max}=31,9$ ms. Eficiența transmisiei, η_{T2} , conform relației (3.14) va fi:

$$\eta_{T2} = \frac{28,94}{70,94} = 0,4079$$

3.3.2.3 Experimentul 3

Structura analizată are topologie inel, distribuitor hub, viteza maximă a mediului de comunicare 10 Mb/s, viteza de prelucrare a pachetelor 250 μ s și

protocol de comunicație UDP. Datorită faptului că nu este permisă interconectarea directă a două hub-uri Ethernet, este necesară introducerea în rețeaua principală a unor Bridge Ethernet care apoi au conexiuni către un hub în parte. Figura 3.26 ilustrează structura principală a rețelei pentru al treilea experiment.

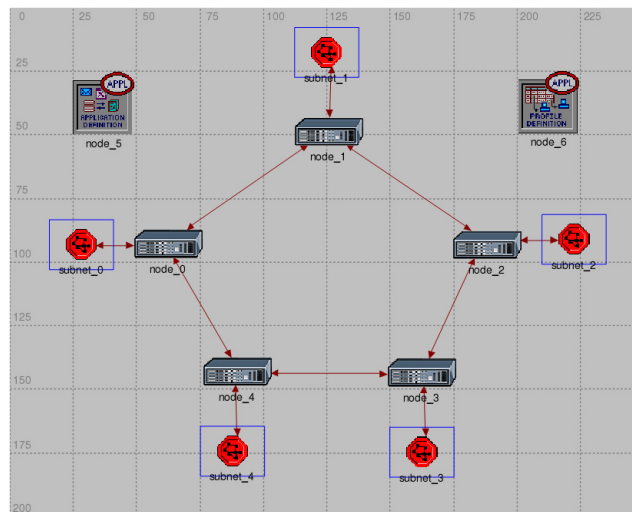


Fig. 3.26. Rețeaua principală pentru Experimentul 3

Elementele noi introduse sunt nodurile node_0,...,node_4 care sunt de tip Ethernet Bridge. Rețelele secundare conțin zece stații de lucru interconectate prin intermediul unui hub Ethernet, așa cum se distinge în figura 3.27.

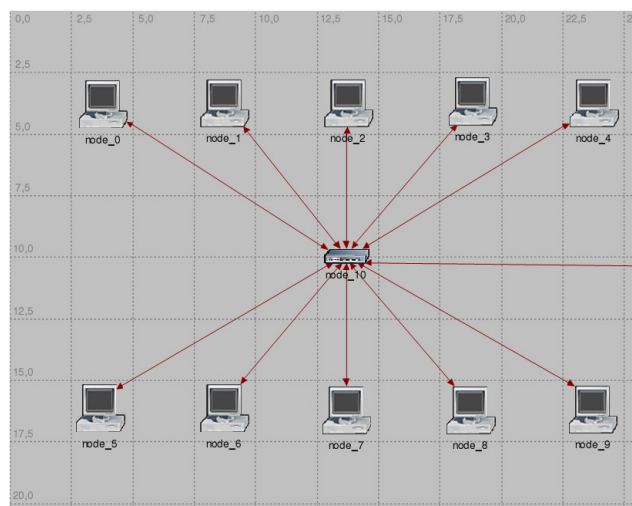


Fig. 3.27. Structura unei rețele secundare

Graficul variației traficului pe rețea este redat în figura 3.28 iar variația întârzierii în figura 3.29.

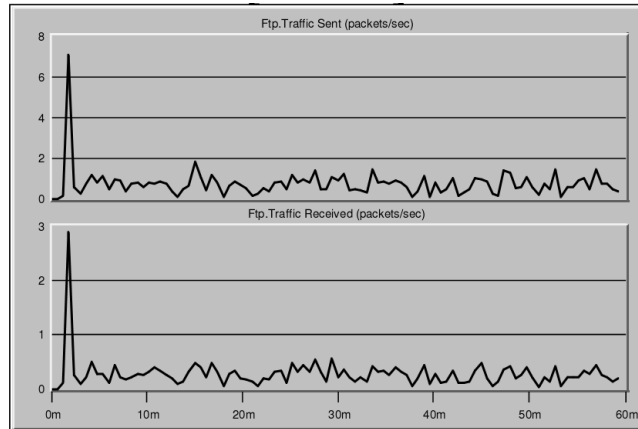


Fig. 3.28. Traficul pe rețea – Experimentul 3

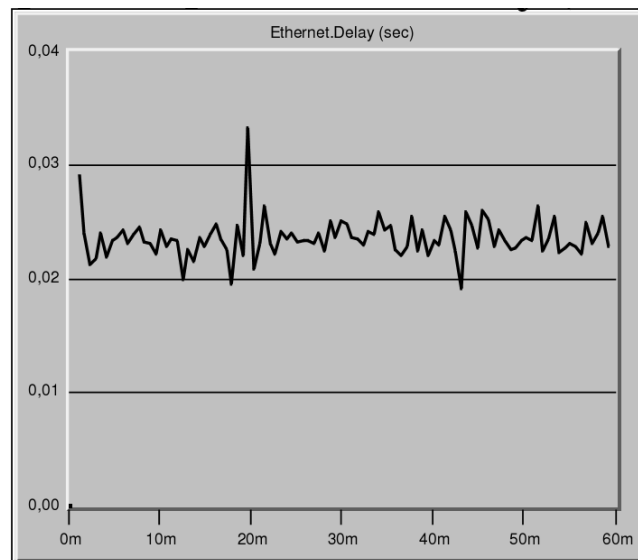


Fig. 3.29. Variația întârzierii – Experimentul 3

Obținem $S_p=72,69$, $R_p=27,52$ iar întârzierea maximă $\Delta_{2max}=33,26$ ms. Eficiența transmisiei pentru acest experiment va fi:

$$\eta_{T3} = \frac{27,52}{72,69} = 0,3786$$

3.3.2.4 Experimentul 4

Vom analiza o structură cu topologie stea, distribuitor switch, viteza maximă a mediului de comunicare 100 Mb/s, viteza de prelucrare a pachetelor 250 μ s și protocol UDP. Topologia rețelei principale este ilustrată în figura 3.30.

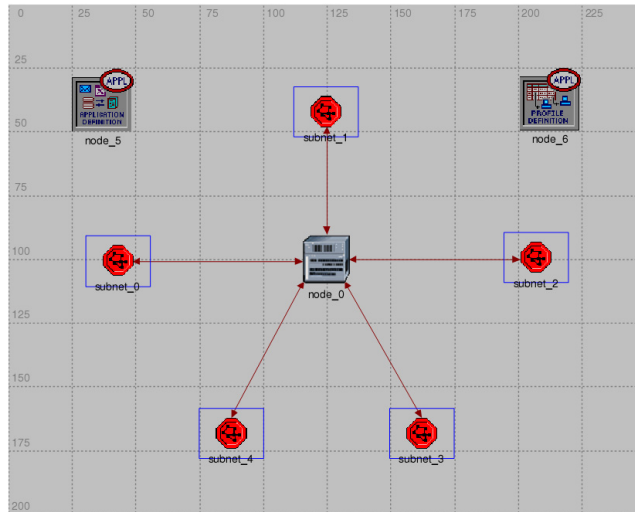


Fig. 3.30. Rețeaua principală pentru Experimentul 4

Rețelele secundare vor conține zece stații de lucru interconectate printr-un distribuitor de tip switch, așa cum se poate observa în figura 3.31.

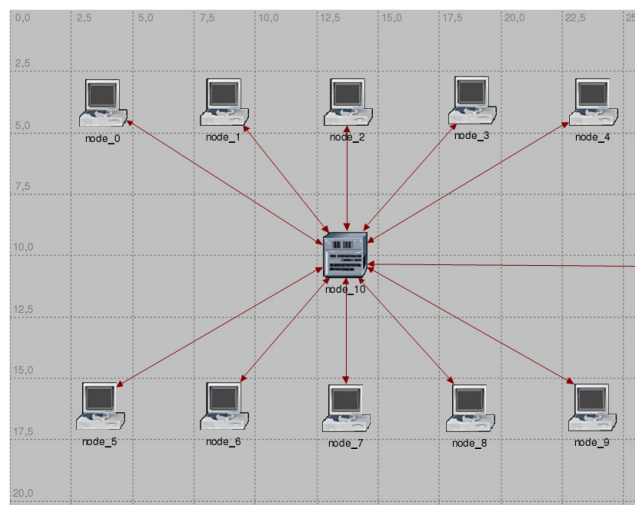


Fig. 3.31. Structura unei rețele secundare

Rulând simularea, obținem valorile prezentate în Anexa 1, din care putem extrage $S_p=73,47$, $R_p=30,14$ și valoarea maximă a întârzierii $\Delta_{4max}=0,754$ ms. Eficiența transmisiei, η_{T4} , conform relației (3.14) va fi:

$$\eta_{T4} = \frac{30,14}{73,47} = 0,4102$$

Forma de variație a traficului pe rețea este redat în figura 3.32 iar variația întârzierii în figura 3.33.

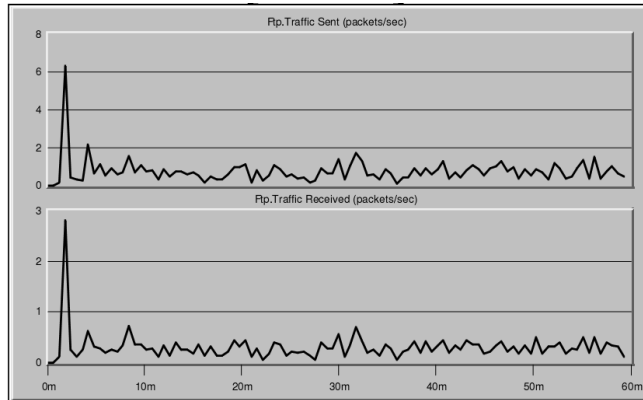


Fig. 3.32. Traficul pe rețea – Experimentul 4

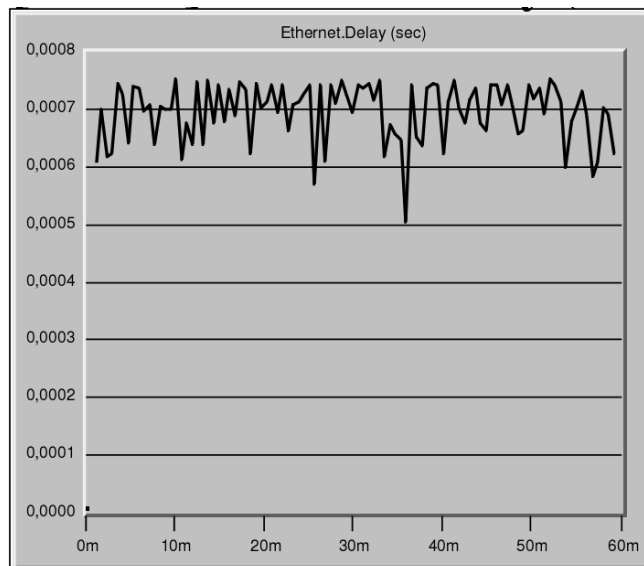


Fig. 3.33. Variația întârzierii – Experimentul 4

3.3.2.5 Experimentul 5

Acest experiment diferă de cel precedent prin tipul protocolului de comunicație (factorul E), acesta fiind TCP. În urma simulării cu OpNet, obținem graficul variației traficului pe rețea conform figurii 3.34. Variația întârzierii este redată în figura 3.35.

Valorile numerice din Anexa 1 conduc la $S_p=72,88$, $R_p=29,22$ și valoarea maximă a întârzierii $\Delta_{5max}=0,297$ ms. Eficiența transmisiei, η_{T5} , va fi:

$$\eta_{T5} = \frac{29,22}{72,88} = 0,4009$$

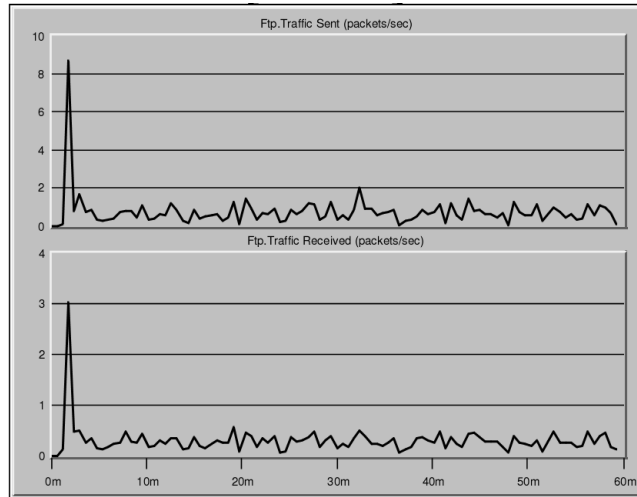


Fig. 3.34. Traficul pe rețea – Experimentul 5

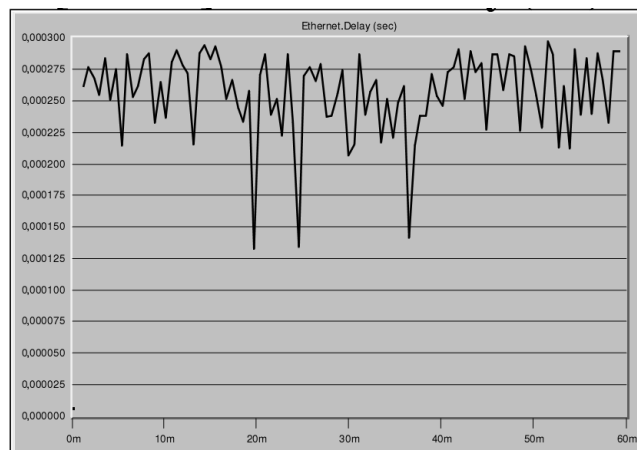


Fig. 3.35. Variația întârzierii – Experimentul 5

3.3.2.6 Experimentul 6

Vom păstra topologia stea și viteza maximă a mediului de comunicare, dar distribuitorul va fi de tip hub, viteza de prelucrare a pachetelor 2,5 μ s și protocolul de comunicație UDP. Figura 3.36 indică modificările structurale ale rețelei principale pentru efectuarea acestui experiment.

Cele cinci rețele secundare vor conține fiecare câte zece stații de lucru, interconectate prin câte un distribuitor de tip hub, cu structura identică celei din figura 3.28.

După parcurgerea simulării, din Anexa 1 extragem $S_p=76,11$, $R_p=30,61$ și valoarea maximă a întârzierii $\Delta_{6max}=2,37$ ms. Eficiența transmisiei, η_{T6} , va fi:

$$\eta_{T6} = \frac{30,61}{76,11} = 0,4022$$

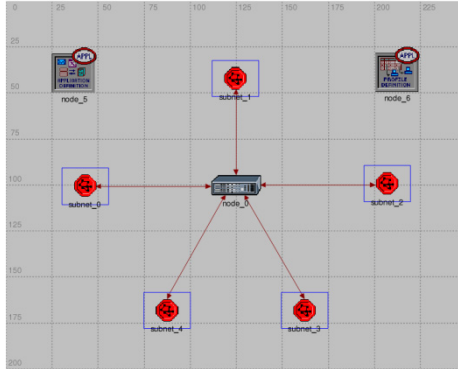


Fig. 3.36. Rețeaua principală pentru Experimentul 6

Graficul de variație al traficului pe rețea este ilustrat în figura 3.37 iar întârzierea este redată în figura 3.38.

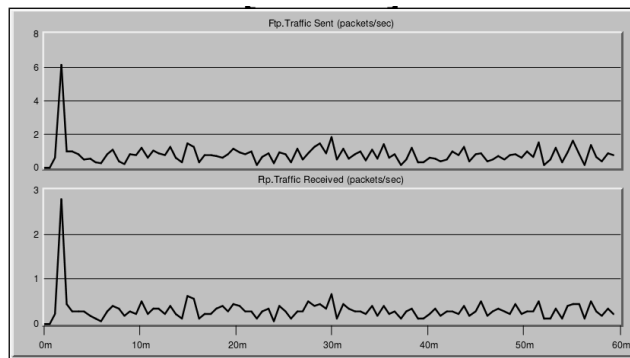


Fig. 3.37. Traficul pe rețea – Experimentul 6

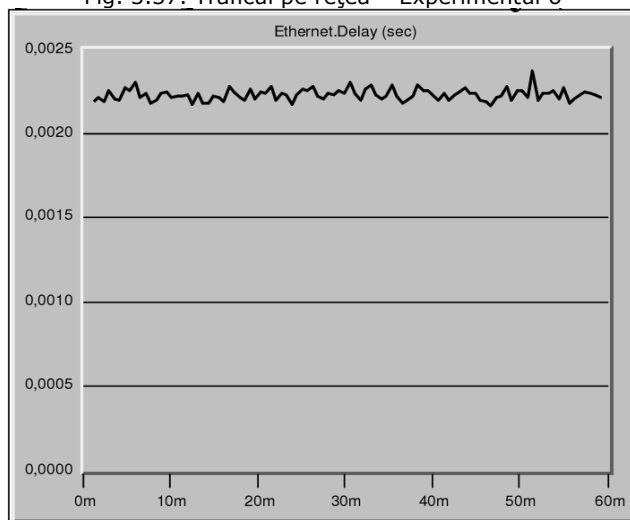


Fig. 3.38. Variația întârzierii – Experimentul 6

3.3.2.7 Experimentul 7

Particularitatea acestei structuri rezultă din topologia liniară, cu distribuitor switch, așa cum se poate observa în figura 3.39.

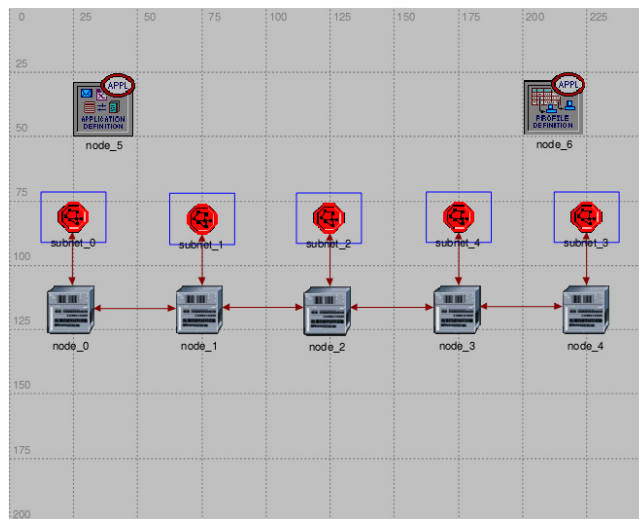


Fig. 3.39. Rețeaua principală pentru Experimentul 7

Viteza maximă a mediului de comunicare este 10 Mb/s, viteza de prelucrare a pachetelor 25 μ s iar protocolul de comunicație de tip TCP.

Structura unei rețele secundare cuprinde zece stații de lucru interconectate prin câte un switch, așa cum este ilustrat în figura 3.40.

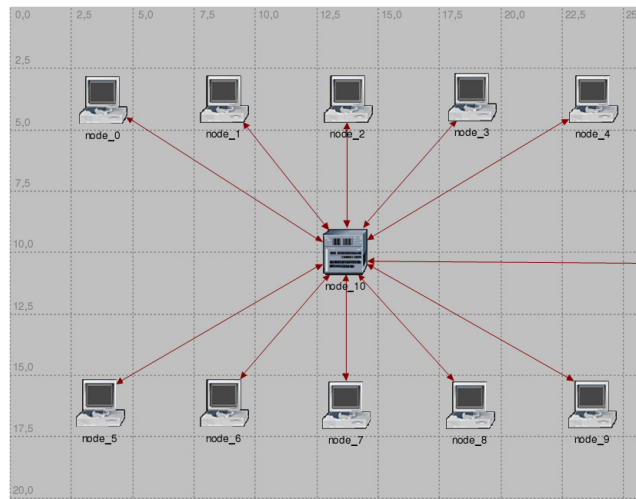


Fig. 3.40. Structura unei rețele secundare

Simularea conduce la $S_p=71,33$, $R_p=29$ și valoarea maximă a întârzierii $\Delta_{7max}=5,06$ ms. (vezi Anexa 1). Eficiența transmisiei, η_{T7} , va fi:

$$\eta_{T7} = \frac{29}{71,33} = 0,4066$$

Traficul pe rețea are forma de variație prezentată în figura 3.41 iar întârzierea introdusă de structura aleasă în figura 3.42.

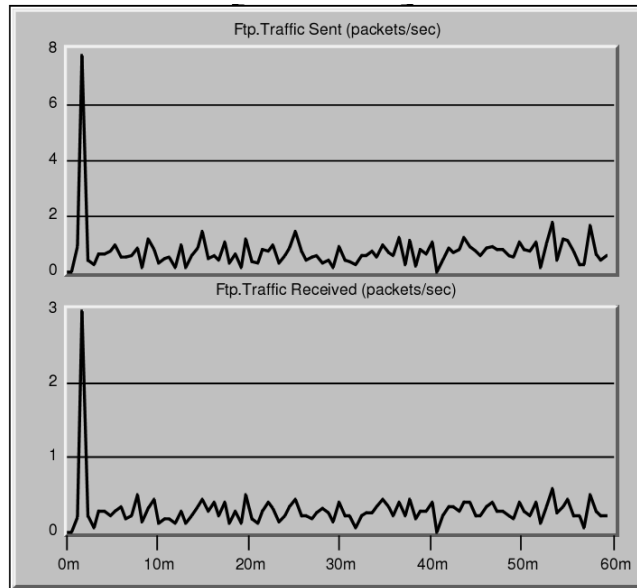


Fig. 3.41. Traficul pe rețea – Experimentul 7

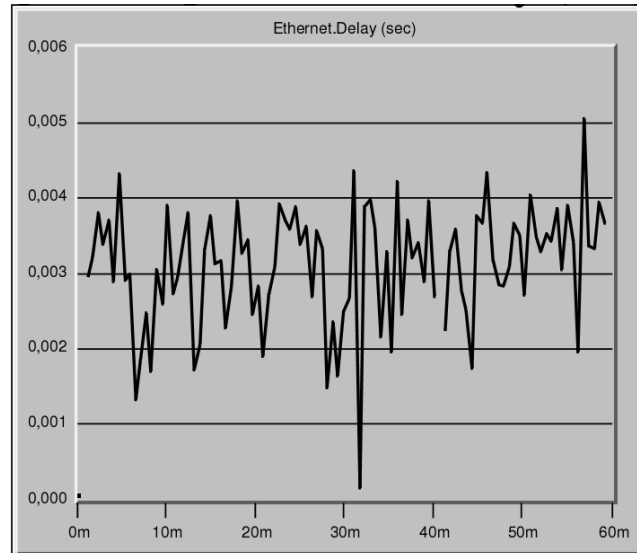


Fig. 3.42. Variația întârzierii – Experimentul 7

3.3.2.8 Experimentul 8

Rețeaua prezintă tot o topologie liniară dar distribuitorul va fi de tip hub, așa cum se distinge în figura 3.43. Viteza maximă a mediului de comunicare este 10 Mb/s, viteza de prelucrare a pachetelor 250 μ s iar protocolul de comunicație de tip UDP.

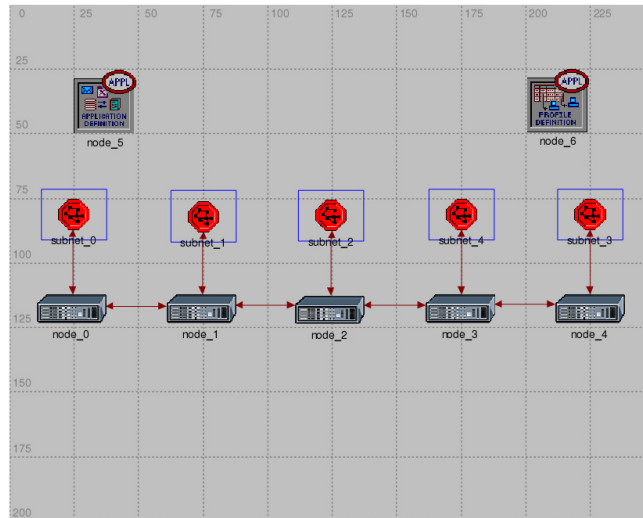


Fig. 3.43. Rețeaua principală pentru Experimentul 8

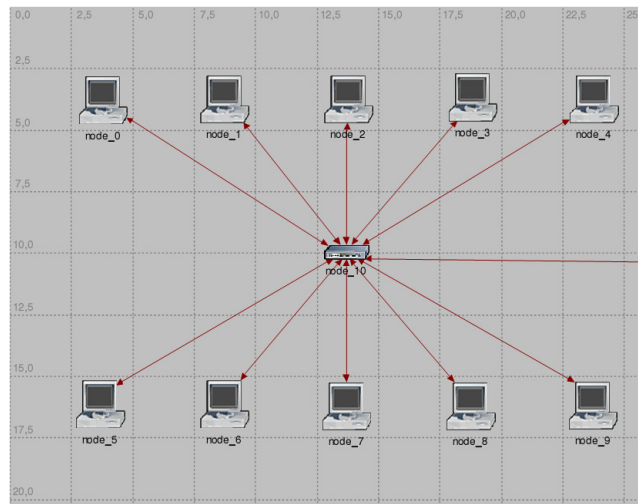


Fig. 3.44. Structura unei rețele secundare

Rețelele secundare vor conține zece stații de lucru interconectate prin intermediul unui hub, așa cum este ilustrat în figura 3.44.

În urma simulării cu OpNet, obținem graficul variației traficului pe rețea conform figurii 3.45. Variația întârzierii este redată în figura 3.46.

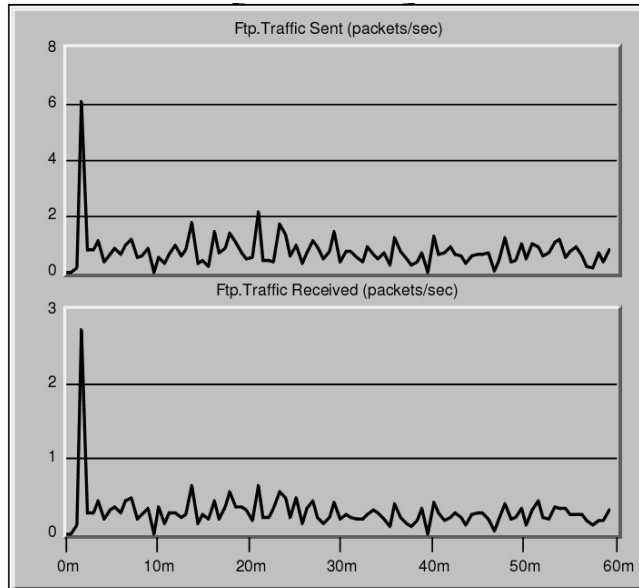


Fig. 3.45. Traficul pe rețea – Experimentul 8

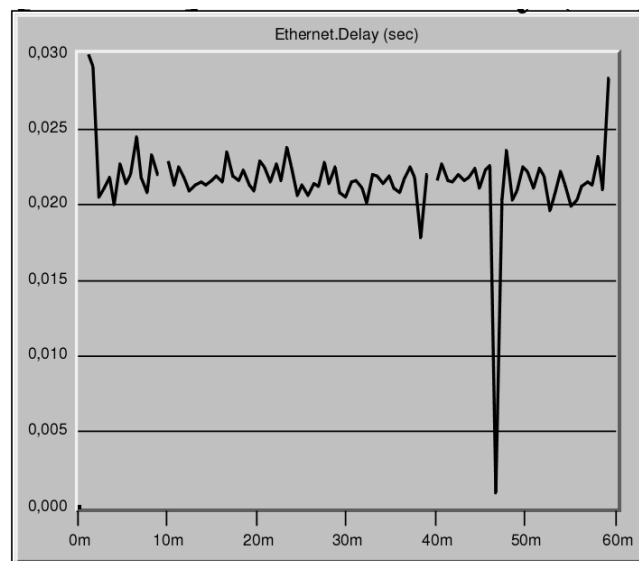


Fig. 3.46. Variația întârzierii – Experimentul 8

Din Anexa 1 extragem $S_p=72,92$, $R_p=28,75$ și valoarea maximă a întârzierii $\Delta_{8max}=29,91$ ms. Eficiența transmisiei, η_{T8} , va fi:

$$\eta_{T8} = \frac{28,75}{72,92} = 0,3943$$

3.3.2.9 Experimentul 9

Structura fizică a rețelei este similară celei din Experimentul 6, ilustrată în figura 3.36, cu stațiile de lucru interconectate conform figurii 3.27. Viteza maximă a mediului de comunicare va fi 10 Mb/s, viteza de prelucrare a pachetelor 2,5 μ s iar protocolul de comunicație de tip TCP.

Obținem pentru traficul pe rețea un grafic de variație redat în figura 3.47 iar pentru întârziere o variație conform figurii 3.48.

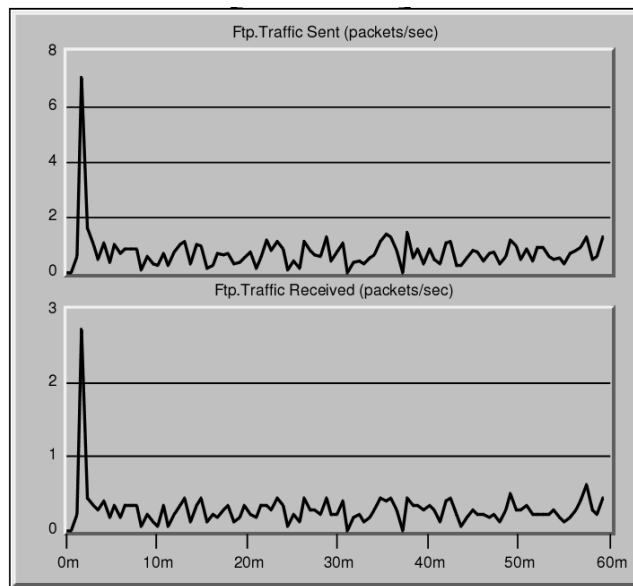


Fig. 3.47. Traficul pe rețea – Experimentul 9

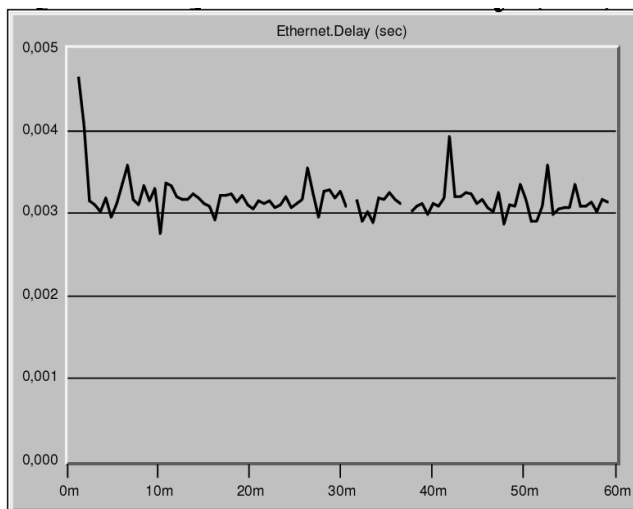


Fig. 3.48. Variația întârzierii – Experimentul 9

Valorile extrase pe baza Anexei 2 sunt: $S_p=70,77$, $R_p=28,11$ și valoarea maximă a întârzierii $\Delta_{9\max}=4,65$ ms. Eficiența transmisiei, η_{T9} , va fi:

$$\eta_{T9} = \frac{28,11}{70,77} = 0,3972$$

3.3.2.10 Experimentul 10

Vom analiza în acest caz o rețea cu topologie stea, distribuitor switch, similară celei din figura 3.49. Cele zece stații de lucru sunt interconectate tot printr-un distribuitor switch, așa cum se poate observa în figura 3.50. Viteza maximă a mediului de comunicare va fi 10 Mb/s, viteza de prelucrare a pachetelor 25 μ s iar protocolul de comunicație de tip UDP.

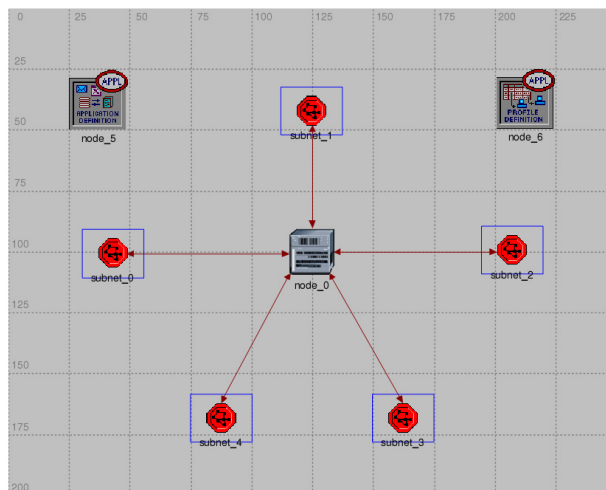


Fig. 3.49. Rețeaua principală pentru Experimentul 10

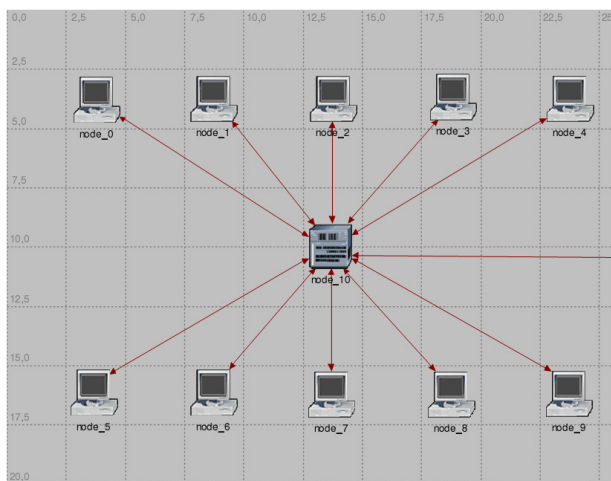


Fig. 3.50. Structura unei rețele secundare

Valorile obținute în urma simulării sunt: $S_p=73,11$, $R_p=29,78$ și valoarea maximă a întârzierii $\Delta_{10\max}=25,45$ ms. Eficiența transmisiei, η_{T10} , va fi:

$$\eta_{T10} = \frac{29,78}{73,11} = 0,4073$$

Figura 3.51 indică variația traficului iar figura 3.52 graficul întârzierii.

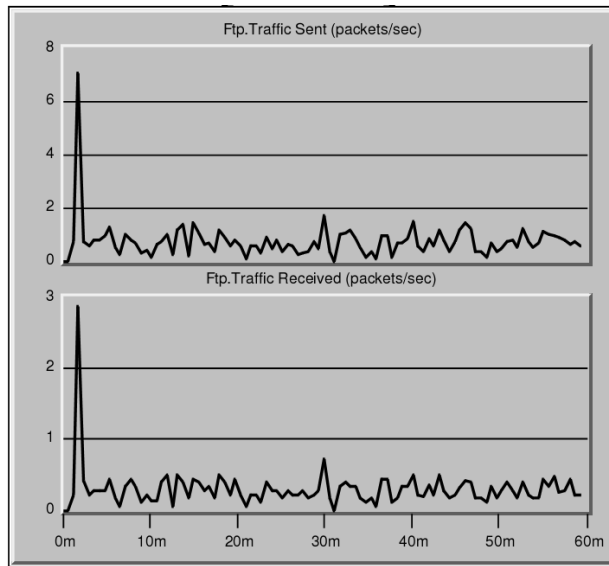


Fig. 3.51. Traficul pe rețea – Experimentul 10

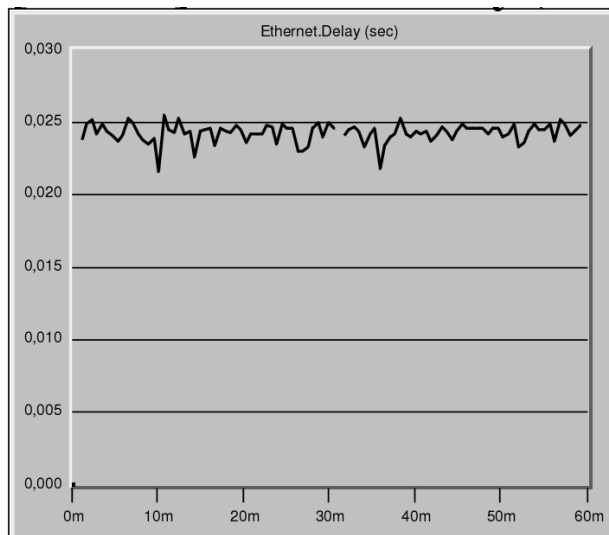


Fig. 3.52. Variația întârzierii – Experimentul 10

3.3.2.11 Experimentul 11

Caracteristic acestui experiment este topologia inel, distribuitor switch, viteza maximă a mediului de comunicare 100 Mb/s, viteza de prelucrare a pachetelor 250 μ s iar protocolul de comunicație de tip TCP. Figura 3.53 redă structura rețelei principale. Cele cinci rețele secundare vor conține zece stații de lucru interconectate prin intermediul distribuitorilor de tip switch, similar celor prezentate în figura 3.50.

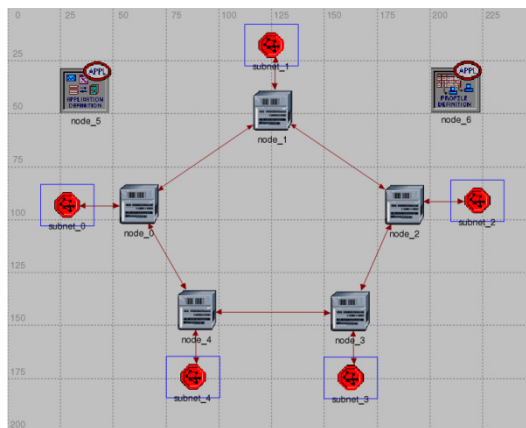


Fig. 3.53. Rețeaua principală pentru Experimentul 11

Simularea acestui experiment ne conduce la valorile $S_p=74,64$, $R_p=28,47$ și valoarea maximă a întârzierii $\Delta_{11max}=0,538$ ms (vezi Anexa 1). Eficiența transmisiei, η_{T11} , va fi:

$$\eta_{T11} = \frac{28,47}{74,64} = 0,3814$$

Traficul pe rețea va avea o variație conform figurii 3.54 iar întârzierea conform figurii 3.55.

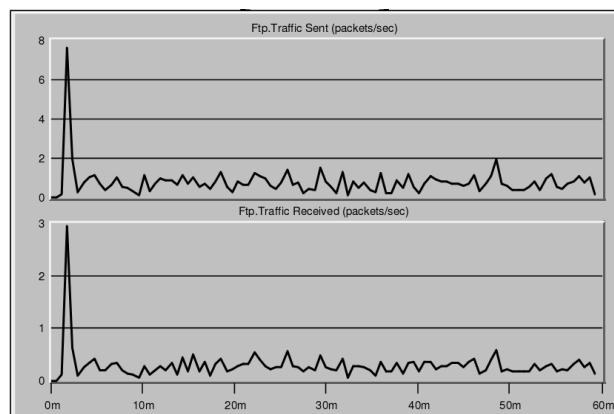


Fig. 3.54. Traficul pe rețea – Experimentul 11

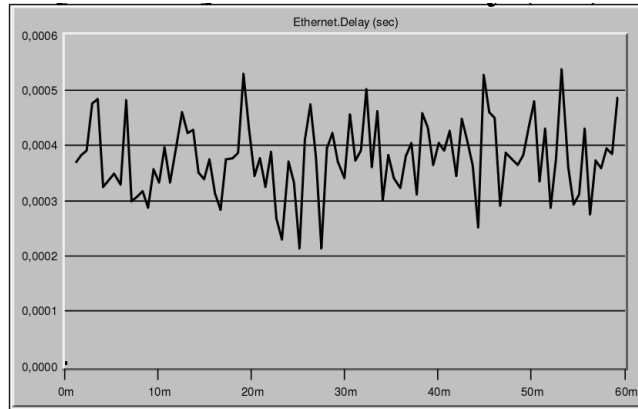


Fig. 3.55. Variația întârzierii – Experimentul 11

3.3.2.12 Experimentul 12

Se păstrează topologia inel dar cu distribuitor de tip hub, așa cum este redat în figura 3.56. Viteza maximă a mediului de comunicare va fi 100 Mb/s, viteza de prelucrare a pachetelor 25 μs iar protocolul de comunicație de tip UDP.

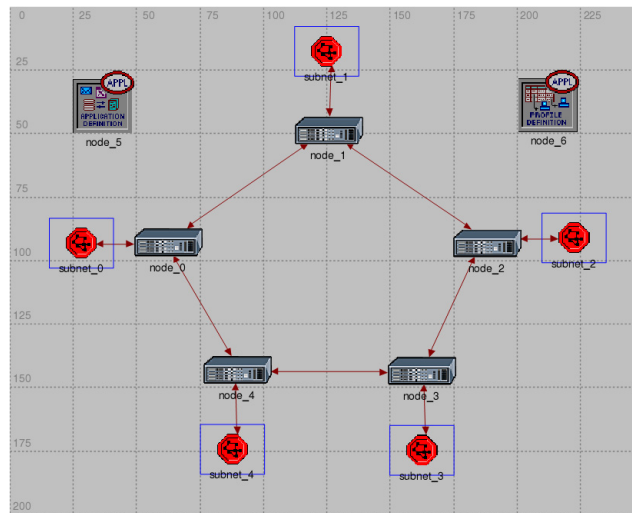


Fig. 3.56. Rețeaua principală pentru Experimentul 12

Fiecare rețea secundară va conține zece stații de lucru interconectate cu ajutorul a câte unui hub, așa cum este ilustrat în figura 3.57.

Traficul pe rețea este reprezentat în figura 3.58 iar întârzierea este redată în figura 3.59.

Obținem următoarele valori (vezi Anexa 1): $S_p=69,44$, $R_p=28,61$ și valoarea maximă a întârzierii $\Delta_{12max}=2,78$ ms (vezi Anexa 1). Eficiența transmisiei, η_{T12} , va fi:

$$\eta_{T12} = \frac{28,61}{69,44} = 0,4120$$

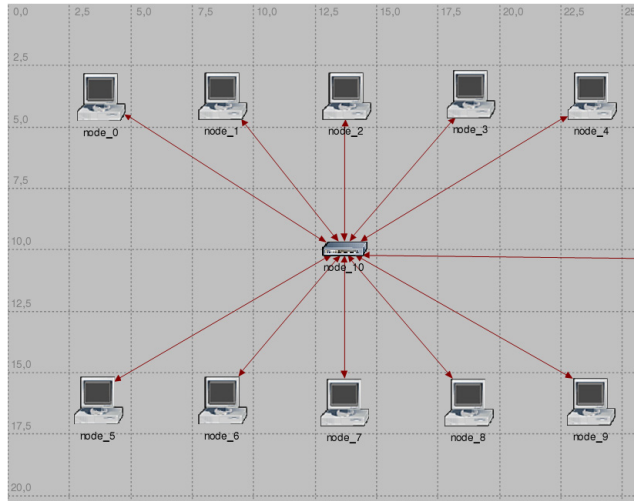


Fig. 3.57. Structura unei rețele secundare

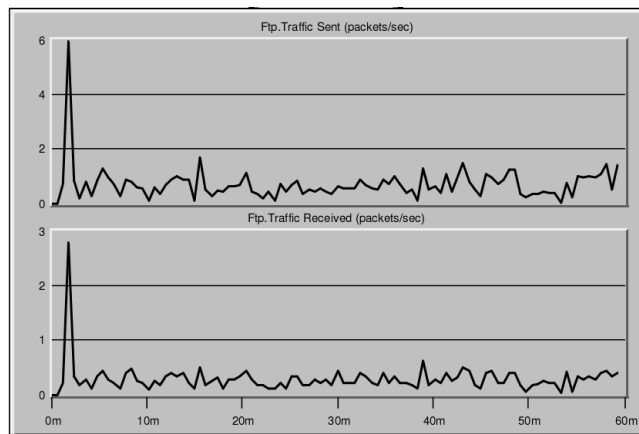


Fig. 3.58. Traficul pe rețea – Experimentul 12

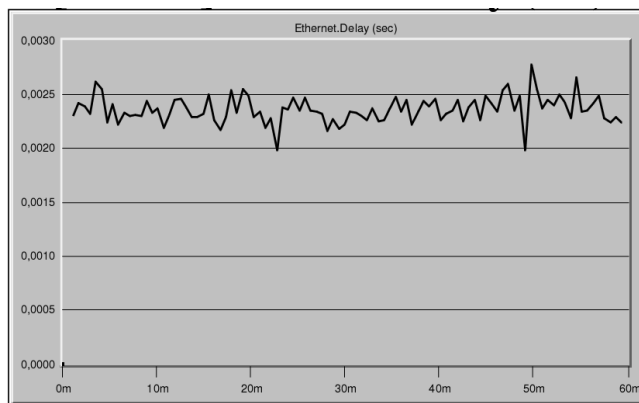


Fig. 3.59. Variația întârzierii – Experimentul 12

3.3.2.13 Experimentul 13

Rețeaua analizată va dispune de o topologie liniară, cu distribuitor hub. Viteza maximă a mediului de comunicare va fi 100 Mb/s, viteza de prelucrare a pachetelor 25 μs iar protocolul de comunicație TCP. Figura 3.60 redă structura rețelei principale.

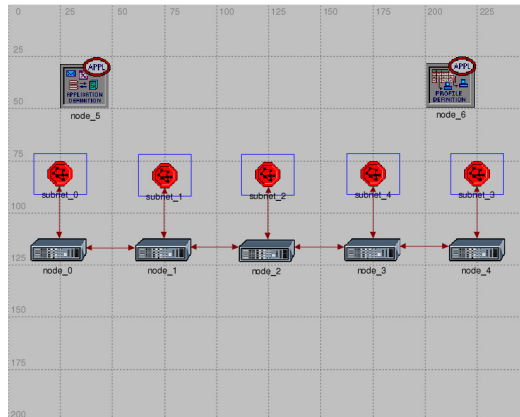


Fig. 3.60. Rețeaua principală pentru Experimentul 13

Cele cinci rețele secundare conțin zece stații de lucru interconectate prin intermediul a câte unui hub, așa cum este vizibil în figura 3.57.

Rezultatele simulării ne conduc la valorile $S_p=74,67$, $R_p=29,67$ și valoarea maximă a întârzierii $\Delta_{13max}=0,418$ ms (vezi Anexa 1). Eficiența transmisiei, η_{T13} , va fi:

$$\eta_{T13} = \frac{29,67}{74,67} = 0,3973$$

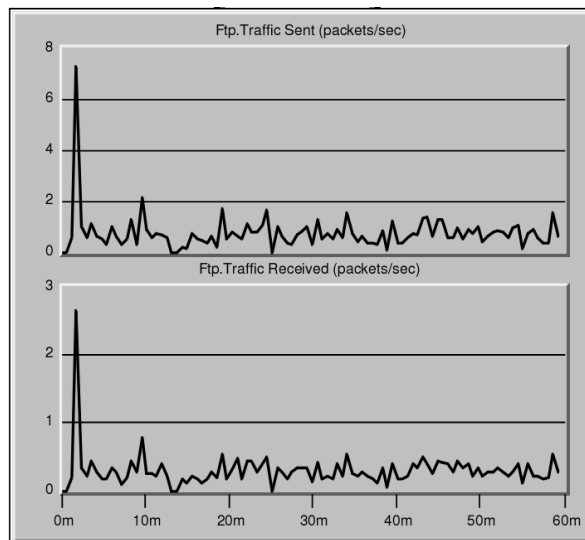


Fig. 3.61. Traficul pe rețea – Experimentul 13

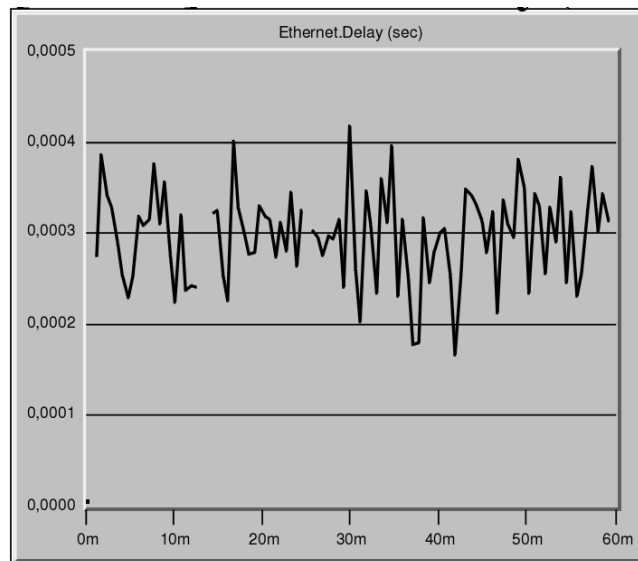


Fig. 3.62. Variația întârzierii – Experimentul 13

Figura 3.61 ilustrează variația traficului pe rețea iar figura 3.62 indică graficul întârzierii introduse de către structura selectată.

3.3.2.14 Experimentul 14

Ultima structură de rețea analizată va prezenta o topologie liniară cu distribuitor switch și viteza maximă a mediului de comunicare 100 Mb/s. Viteza de prelucrare a pachetelor de date va fi de 2,5 μ s iar protocolul de comunicație UDP. Similar celorlalte experimente, o rețea secundară va conține zece stații de lucru care vor comunica tot prin intermediul unui switch.

În figura 3.63 este prezentată structura rețelei principale aferentă acestui experiment.

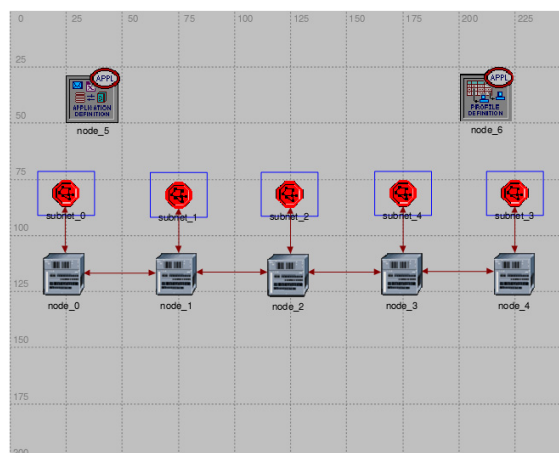


Fig. 3.63. Rețeaua principală pentru Experimentul 14

Structura unei rețele secundare se poate urmări în figura 3.50.
 Conform Anexei 2, obținem în urma simulării valorile: $S_p=69,22$, $R_p=28,72$
 și $\Delta_{14max}=2,9$ ms. Eficiența transmisiei, η_{T14} , va fi:

$$\eta_{T14} = \frac{28,72}{69,22} = 0,4149$$

Variația traficului pe rețea se poate urmări în figura 3.64.

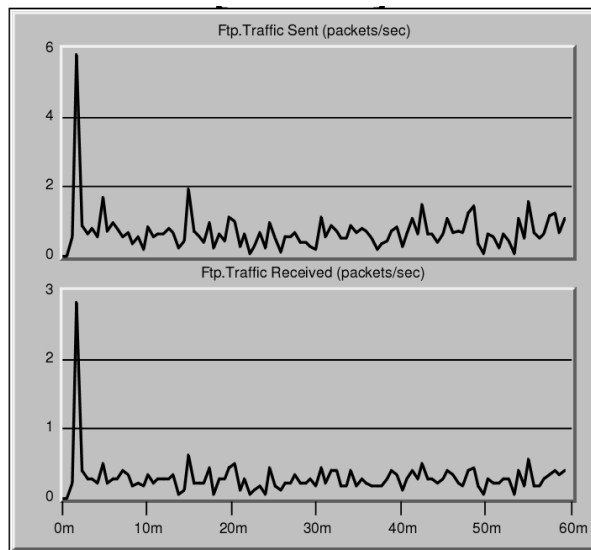


Fig. 3.64. Traficul pe rețea – Experimentul 14

Întârzierea introdusă are o variație conform graficului din figura 3.65.

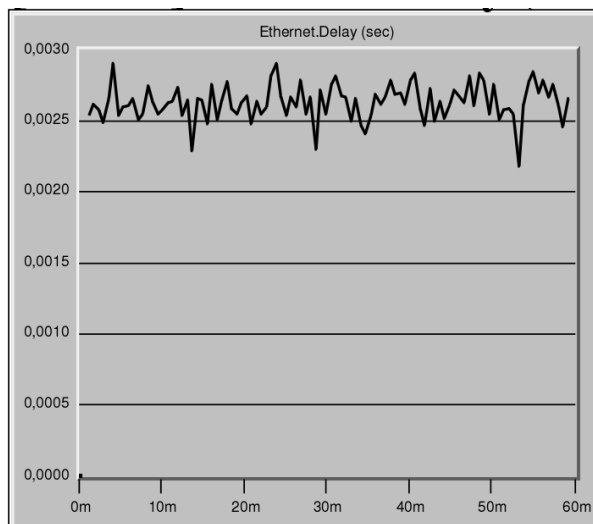


Fig. 3.65. Variația întârzierii – Experimentul 14

3.3.3 Evaluarea rezultatelor experimentale

În urma celor 14 experimente, putem sintetiza sub forma tabelului 3.5 rezultatele de interes pentru continuarea analizei și optimizării rețelelor de teren.

Experiment	Sp	Rp	η_T	Δ_{\max} (ms)
1	66,5	27	0,406	6,16
2	70,94	28,94	0,4079	31,9
3	72,69	27,52	0,3786	33,26
4	73,47	30,14	0,4102	0,754
5	72,88	29,22	0,4009	0,297
6	76,11	30,61	0,4022	2,37
7	71,33	29	0,4066	5,06
8	72,92	28,75	0,3943	29,91
9	70,77	28,11	0,3972	4,65
10	73,11	29,78	0,4073	25,45
11	74,64	28,47	0,3814	0,538
12	69,44	28,61	0,412	2,78
13	74,67	29,67	0,3973	0,418
14	69,22	28,72	0,4149	2,9

Tabelul 3.5. Sinteza rezultatelor experimentale

Factorii de răspuns, η_T (eficiența transmisiei) și Δ_{\max} (întârzierea maximă), vor fi introduși în tabelul principal de proiectare, expus în figura 3.14, rezultând configurația prezentată în figura 3.66. Urmează ca utilitarul JMP, pe baza acestor date să genereze diverse diagrame preliminare care ne vor permite să analizăm tendința de variație a factorilor de răspuns.

Whole Plots	Subplots	Topologia	Distribuitor	VMC	VPP	Protocol	Eficienta Transmisie	Întârziere	
1	1	1	Inel	Switch	10 Mb/s	2,5 us	TCP	0,406	6,16
2	1	1	Inel	Switch	10 Mb/s	2,5 us	UDP	0,4079	31,9
3	1	2	Inel	Hub	10 Mb/s	250 us	UDP	0,3786	33,26
4	2	3	Stea	Switch	100 Mb/s	250 us	UDP	0,4102	0,754
5	2	3	Stea	Switch	100 Mb/s	250 us	TCP	0,4009	0,297
6	2	4	Stea	Hub	100 Mb/s	2,5 us	UDP	0,4022	2,37
7	3	5	Linara	Switch	10 Mb/s	25 us	TCP	0,4066	5,06
8	3	6	Linara	Hub	10 Mb/s	250 us	UDP	0,3943	29,91
9	4	7	Stea	Hub	10 Mb/s	2,5 us	TCP	0,3972	4,65
10	4	8	Stea	Switch	10 Mb/s	25 us	UDP	0,4073	25,45
11	5	9	Inel	Switch	100 Mb/s	250 us	TCP	0,3814	0,538
12	5	10	Inel	Hub	100 Mb/s	25 us	UDP	0,412	2,78
13	6	11	Linara	Hub	100 Mb/s	25 us	TCP	0,3973	0,418
14	6	12	Linara	Switch	100 Mb/s	2,5 us	UDP	0,4149	2,9

Fig. 3.66. Tabelul de proiectare conținând factorii de răspuns

3.3.3.1 Tipul distribuției

Prin trasarea a trei diagrame pentru fiecare factor de răspuns în parte, vom determina tipul distribuției de variație a acestora.

Histograma factorului de răspuns este cea mai simplă reprezentare și ne ajută să observăm anumite tendințe de grupare în jurul unor valori centrale. Având o variabilă cantitativă continuă, această diagramă permite vizualizarea formei distribuției statistice aferente variabilei.

Diagrama normală a probabilității trasată pentru factorul de răspuns permite identificarea tipului de distribuție după care variază factorul în cauză. Linia roșie continuă reprezintă o distribuție normală iar cele două linii curbe întrerupte formează un interval de încredere de 95% pentru valorile reprezentate.

„Box plot” (o diagramă de tip „cutie cu mustăți”) care permite vizualizarea distribuției răspunsului pe domenii de variație, extremitatea inferioară respectiv superioară a dreptunghiului indicând limita de 25% respectiv 75% din valoarea răspunsului. Dreapta din interiorul patrulaterului ilustrează valoarea mediană a eșantionului de răspuns iar rombul reprezintă valoarea medie precum și încadrarea în intervalul de încredere de 95%. Diagrama este folosită atunci când numărul eșantioanelor de care dispunem este relativ redus [42], [43].

Pentru eficiența transmisiei obținem cele trei diagrame ilustrate în figura 3.67 a, b și c.

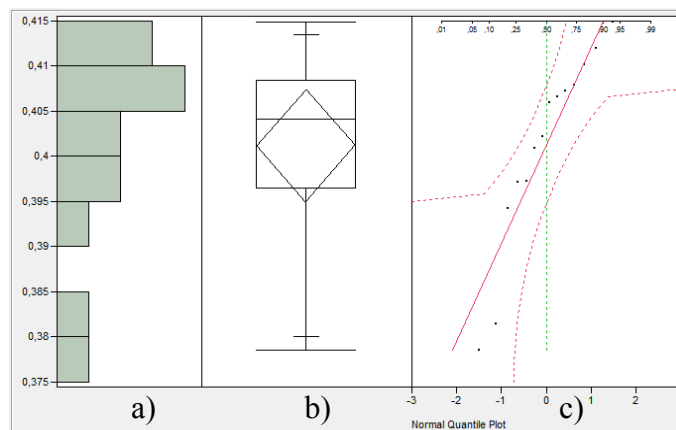


Fig. 3.67. Diagrame de distribuție pentru eficiența transmisiei

Figura 3.67a ne indică o distribuire a eșantioanelor în jurul unei valori centrale. Figura 3.67b este mai ilustrativă, fiind marcate prin două linii orizontale limitele de variație ale eșantioanelor. Quartilele marcate pe „box plot” cu segmente scurte sunt așezate echidistant unele față de altele ceea ce ar putea conduce la ipoteza unei distribuții normale. În schimb curba normală a quantilelor reprezentată în figura 3.67c indică unele abateri de la caracterul distribuției normale pentru acest factor.

Vom urmări în continuare determinarea funcției densitate de probabilitate ce caracterizează distribuția variabilei aleatoare „eficiența transmisiei”. În urma analizei datelor experimentale cu ajutorul utilitarului JMP, obținem câțiva parametri descriptivi, expuși în tabelul 3.6.

Parametru	Valoare
Mean (μ)	0,4012
Std Dev (σ)	0,010750098
Std Err Mean	0,002873085
upper 95% Mean	0,407406922
lower 95% Mean	0,394993078
N	14
Sum Wgt	14
Sum	5,6168
Variance (σ^2)	0,000115565
Skewness	-1,04026848
Kurtosis (χ)	0,53457451
CV	2,679486139
N Missing	0

Tabelul 3.6. Parametrii descriptivi η_T

Se pot recunoaște câteva elemente uzuale în statistică: media μ , abaterea standard σ (eroarea medie pătratică), varianța σ^2 și coeficientul de variație în procente. În plus, avem informații despre suma tuturor eșantioanelor (Sum), ponderea (Sum Wgt), limitele intervalului de încredere asupra mediei (upper 95% Mean, lower 95% Mean) precum și numărul eșantioanelor excluse din analiză (N Missing).

Coeficientul de asimetrie, denumit „skewness”, exprimă gradul de dezechilibru al unei distribuții. Dacă se află între valorile -1 și 0, atunci distribuția este asimetrică negativă, cu abatere spre stânga iar dacă se încadrează între 0 și 1 vom avea o distribuție cu abatere spre dreapta. Distribuția perfect simetrică va fi caracterizată de un coeficient skewness egal cu 0 [42], [43].

Coeficientul de boltire sau aplatizare, „kurtosis”, (regăsit și sub notația χ) este o măsură a răspândirii fiecărei observații empirice în jurul valorii centrale. Pentru o distribuție strict normală, valoarea kurtosisului statistic este 0 și atunci ea se numește distribuție mezocurtică [43]. Pentru valori $\chi > 0$ avem o grupare mai puternică a valorilor în jurul valorii centrale, rezultând o curbă mai boltită decât la o distribuție normală, cunoscută sub numele de distribuție leptocurtică. Dacă $\chi < 0$ vom regăsi o grupare mai slabă în jurul valorii centrale, rezultând o curbă mai aplatizată, denumită platicurtică [43].

Consultând tabelul 3.6 putem deduce că pentru „eficiența transmisiei” avem o distribuție leptocurtică asimetrică negativă, cu abatere spre stânga. Ne vom aștepta deci la o funcție densitate de probabilitate care va conduce la o curbă de distribuție care se apropie cel mai mult de aceste caracteristici.

O deosebită atenție trebuie să acordăm și valorii coeficientului de variație, CV, pe baza căruia putem determina cât de reprezentativă este media. Fiind o valoare procentuală, acest coeficient se poate situa între 0 și 100 și cu cât are o valoare mai mică, cu atât seria statistică este mai omogenă și astfel media este mai reprezentativă [42], [43]. Literatura de specialitate ne furnizează interval de valori pentru CV, cu următoarele semnificații:

- $CV \in (0,17\%)$: media este strict reprezentativă;
- $CV \in (17\%, 35\%)$: media este moderat reprezentativă;
- $CV \in (35\%, 50\%)$: media este reprezentativă în sens larg;
- $CV > 50\%$: media nu este reprezentativă, seria fiind eterogenă.

Tabelul 3.6 ne furnizează valoarea $CV = 2,679486139$ și astfel rezultă că seria statistică este omogenă iar media (μ) este strict reprezentativă [42], [43].

Utilitarul JMP permite determinarea tipului de distribuție a unei variabile de răspuns prin modelarea funcției densitate de probabilitate asociate distribuției. Avem două funcții ce pot candida în cazul nostru: cea aferentă unei distribuții normale și cea aferentă unei distribuții Weibull cu 2 parametri. Figura 3.68 ilustrează cele două curbe de repartiție suprapuse peste histograma factorului de răspuns „eficiența transmisiei”.

Culoarea roșie corespunde distribuției normale iar cea albastră distribuției Weibull.

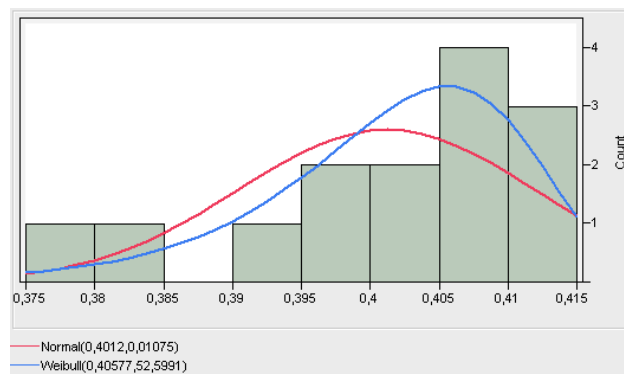


Fig. 3.68 Aproximarea prin distribuția Normală și Weibull

Aproximarea prin distribuția normală va conduce la parametrii caracteristici (locația μ și deviația standard σ) prezentați în tabelul 3.7.

Type	Parameter	Estimate	Lower 95%	Upper 95%
Location	μ	0,4012	0,394993078	0,407406922
Std. Dev.	σ	0,010750098	0,007793326	0,017318865

Tabelul 3.7. Parametrii caracteristici pentru distribuția normală

Pentru a putea lua o decizie corectă asupra „gradului de potrivire” [45] a acestei distribuții cu modul de variație real al eșantioanelor vom recurge la teste care verifică anumite proprietăți ale variabilelor iar pentru stabilirea normalității unei distribuții utilitarul JMP recurge la testul Shapiro – Wilk [45]. Ipoteza nulă pentru acest teste este că datele empirice au o distribuție normală. Se obțin valorile $W = 0,901435$ și $P = 0,1183$. Consultând literatura de specialitate, constatăm că pentru pragul alfa ales inițial, $\alpha = 0,05$ (vezi figura 3.13 – semnificanță nivel), valoarea minimă a lui W pentru a accepta ipoteza nulă este 0,874. Constatăm că și valoarea $P > \alpha$ (P reprezintă probabilitatea calculată reprezentând estimarea probabilității de a rejecta ipoteza nulă). Adoptând notație din literatura de specialitate, H_0 pentru ipoteza nulă [45], vom avea pentru testul de normalitate $H_{01} =$ „distribuția este normală” și putem afirma cel mai corect: „nu putem respinge H_{01} ”.

Considerăm în cele ce urmează, ipoteza nulă H_0 = „distribuția este de tip Weibull”. Funcția densitate de probabilitate aferentă este redată prin curba de culoare albastră ilustrată în figura 3.68. Parametrii caracteristici furnizați de către utilitarul JMP sunt redați în tabelul 3.8.

Type	Parameter	Estimate	Lower 95%	Upper 95%
Scale	α	0,405771	0,40104498	0,4102694
Shape	β	52,59911	32,7203525	77,797134

Tabelul 3.8. Parametrii caracteristici pentru distribuția Weibull

Distribuția Weibull cu doi parametri este caracterizată prin parametrul de scară α și parametrul de formă β [45]. Valoarea $\beta > 1$ confirmă asimetria negativă cu abatere spre stânga a curbei din figura 3.68. Vom verifica gradul de potrivire al acestei distribuții, prin aplicarea testului Cramer – von Mises modificat după Watson, executat cu ajutorul utilitarului JMP. Criteriul permite compararea unei distribuții teoretice descrise prin funcția F^* (în cazul nostru Weibull cu doi parametri) cu o distribuție obținută empiric, descrisă prin F_n , cu n numărul de eșantioane [43], [45]. Indicele calculat va fi W^2 și valoarea acestuia se va compara cu o valoare extrasă din tabelul de valori critice Cramer – von Mises, pentru nivelul de semnificație $\alpha = 0,05$ și un număr de $n = 14$ eșantioane (pe care o vom nota W_{CM}). Ipoteza nulă este confirmată în cazul în care $W^2 < W_{CM}$ [43], [45]. Totodată vom evalua și valoarea calculată a probabilității, P , care va fi comparată cu nivelul $\alpha = 0,05$.

Utilitarul JMP ne furnizează valorile: $W^2 = 0,030774$ și $P = 0,25$. Identificăm în tabelul de valori critice, $W_{CM} = 0,214$. Rezultă $W^2 < W_{CM}$ iar $P > \alpha$ ceea ce va conduce la afirmația „nu putem respinge H_0 ”.

Având rezultate pozitive în urma testelor parcurse pentru ambele tipuri de distribuție, suntem nevoiți să recurgem la un criteriu suplimentar pentru stabilirea cu exactitate a funcției densitate de probabilitate adecvate.

Diagrama Q-Q Plot reprezintă curba quantilelor distribuției teoretice versus eșantioanelor empirice. În cazul în care punctele reprezentate se aliniază dealungul unei drepte înclinată și valorile sunt strict crescătoare de la stânga spre dreapta, rezultă validarea ipotezei nule [45].

Curbele quantilelor (normal și Weibull) obținute cu ajutorul utilitarului JMP sunt redată în figura 3.69.

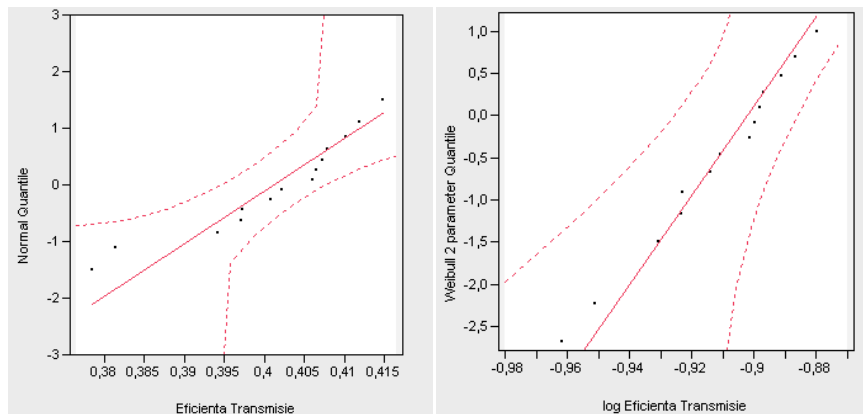


Fig. 3.69. Diagramele Q-Q Plot pentru distribuție normală și Weibull

Analizând cele două reprezentări grafice, putem distinge o mai bună aliniere a punctelor (eșantioanelor) pe dreaptă în cazul unei distribuții Weibull. Evaluând și valorile furnizate de către testul Cramer – von Mises, vom adopta pentru factorul de răspuns „eficiența transmisiei” o distribuție de tip Weibull. Funcția densitate de probabilitate aferentă, cu parametrii din tabelul 3.8, va avea forma:

$$f(x) = \frac{52,59}{0,4057} \left(\frac{x}{0,4057}\right)^{51,59} e^{-\left(\frac{x}{0,4057}\right)^{52,59}} \quad (3.16)$$

Integrând funcția $f(x)$ dată de relația (3.16) pe un anumit interval, vom putea afla probabilitatea ca variabila aleatoare x (în cazul nostru eficiența transmisiei) să se afle în cadrul acestui interval.

Pentru întârzierea introdusă de rețea obținem distribuțiile expuse în figura 3.70 a, b și c.

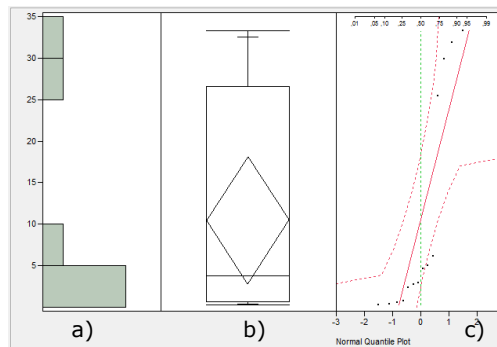


Fig. 3.70. Diagrame de distribuție pentru întârziere

Se remarcă în figura 3.70a gruparea eșantioanelor de răspuns în jurul a două intervale, așa cum și era de așteptat datorită celor două valori ale vitezei mediului de comunicare.

Parametru	Valoare
Mean (μ)	10,4605
Std Dev (σ)	13,1333025
Std Err Mean	3,510022736
upper 95% Mean	18,043443103
lower 95% Mean	2,877556896
N	14
Sum Wgt	14
Sum	146,447
Variance (σ^2)	172,483634576
Skewness	1,054980506
Kurtosis (χ)	-0,846945501
CV	125,551383784
N Missing	0

Tabelul 3.9. Parametrii descriptivi Δ_{\max}

Diagramele 3.70b și 3.70c indică o deviere clară față de o distribuție normală a valorilor.

Parametrii descriptivi aferenți distribuției obținute sunt redați în tabelul 3.9.

Valoarea obținută $CV = 125,55$ indică foarte clar caracterul eterogen al seriei, media nefiind reprezentativă. În acest caz este necesară separarea datelor în serii componente, pe grupuri.

Vom urmări în continuare determinarea funcției densitate de probabilitate ce caracterizează distribuția variabilei aleatoare „întârziere”. În urma analizei datelor experimentale cu ajutorul utilitarului JMP, observând și distribuția eșantioanelor în figura 3.69a, putem deduce existența practic a unei funcții densitate de probabilitate compuse, definite doar pe anumite domenii de variație a eșantioanelor. Distingem două distribuții, aferente celor două valori a vitezei mediului de comunicare și anume 10 Mb/s respectiv 100 Mb/s, ilustrate separat în figura 3.71 și 3.72.

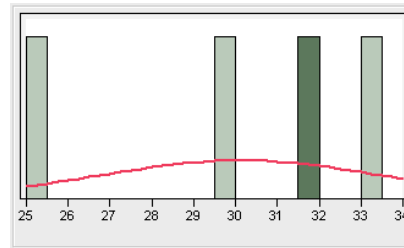
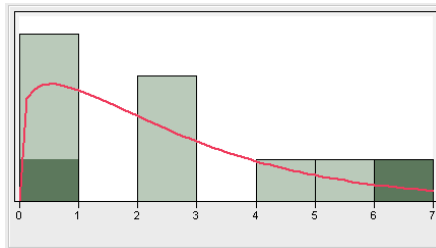


Fig. 3.71 Distribuția întârzierii – 100 Mb/s Fig. 3.72 Distribuția întârzierii – 10 Mb/s

Pentru cele două seturi de distribuții prezentate mai sus, obținem parametrii descriptivi reuniți în tabelul 3.10.

Consultând tabelul 3.10 putem deduce că pentru „întârziere – 100 Mb/s” avem o distribuție platycurtică asimetrică pozitivă, cu abatere spre dreapta, deoarece coeficientul de asimetrie, „skewness” este mai mare ca 0, iar coeficientul de aplatizare „kurtosis” este negativ.

Parametru	Valoare – 10 Mb/s	Valoare – 100 Mb/s
Mean (μ)	30,13	2,5927
Std Dev (σ)	3,40981915	0,12986593
Std Err Mean	1,70490958	0,00410968
upper 95% Mean	35,5557832	4,11631431
lower 95% Mean	24,7042168	1,06908569
N	4	10
Sum Wgt	4	10
Sum	120,52	25,927
Variance (σ^2)	11,6268667	0,00016865
Skewness	-1,1149534	0,45818407
Kurtosis (χ)	0,93740031	-1,196484
CV	11,3170234	5,008
N Missing	10	4

Tabelul 3.10. Parametrii descriptivi pentru cele două distribuții

Totodată, tabelul 3.10 indică pentru „întârziere – 10 Mb/s” o distribuție leptocurtică asimetrică negativă, cu abatere spre stânga, stabilită de valorile „skewness” negativ și „kurtosis” pozitiv.

Pentru cele două serii grupate, obținem valori satisfăcătoare pentru coeficientul de variație CV: 11,31 respectiv 5,008 astfel cele două medii statistice fiind acum reprezentative.

Utilitarul JMP ne permite să aproximăm funcțiile densitate de probabilitate aferente celor două cazuri, acestea fiind marcate cu roșu așa cum se poate observa în figura 3.71 respectiv 3.72.

Opțiunile de modelare a funcțiilor incluse în JMP ne conduc la alegerea unei distribuții de tip Weibull cu doi parametri pentru „întârziere – 100 Mb/s” precum și a unei distribuții normale pentru „întârziere – 10 Mb/s”.

Notăm cu $f_{100}(x)$ funcția densitate de probabilitate pentru „întârziere – 100 Mb/s”, de tip Weibull, de forma generală:

$$f(x) = \frac{\beta}{\alpha} \left(\frac{x}{\alpha}\right)^{\beta-1} e^{-\left(\frac{x}{\alpha}\right)^{\beta}} \quad (3.17)$$

unde α reprezintă factorul de scară iar β este factorul de formă.

Aproximarea printr-o distribuție Weibull cu doi parametri pentru „întârziere – 100 Mb/s” conduce la valorile incluse în tabelul 3.11.

Type	Parameter	Estimate	Lower 95%	Upper 95%
Scale	α	2,73428745	1,42713523	5,00727636
Shape	β	1,17129929	0,66110284	1,87048008

Tabelul 3.11. Parametrii caracteristici pentru distribuția Weibull

Conform relației (3.17), vom avea funcția densitate de probabilitate:

$$f_{100}(x) = 0,4284 \left(\frac{x}{2,7343}\right)^{0,1713} e^{-\left(\frac{x}{2,7343}\right)^{1,1713}} \quad (3.18)$$

Notăm cu $f_{10}(x)$ funcția densitate de probabilitate pentru „întârziere – 10 Mb/s”, de tip normală, de forma generală:

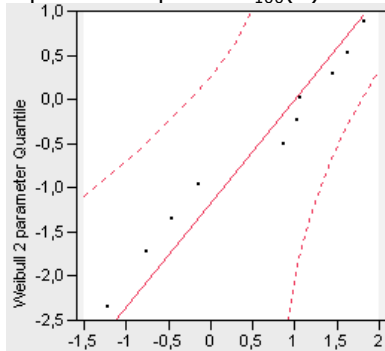
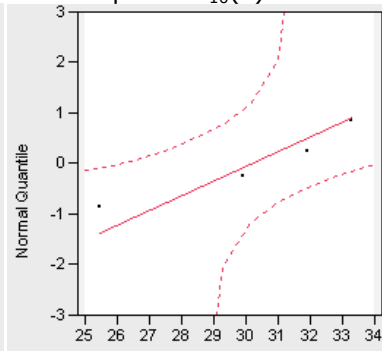
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.19)$$

unde σ reprezintă deviația standard iar μ media eșantioanelor.

Disponând de aceste valori, prezentate în tabelul 3.10, putem deduce expresia matematică a funcției densitate de probabilitate pentru „întârziere – 10 Mb/s”, ca fiind:

$$f_{10}(x) = 0,117e^{-\frac{(x-30,13)^2}{23,256}} \quad (3.20)$$

Ultimul pas pentru validarea funcțiilor este testarea ipotezei nule [45] prin trasarea curbei quantilelor și efectuarea testelor de potrivire adecvate. Figura 3.73 redă curba quantilelor pentru $f_{100}(x)$ iar figura 3.74 pentru $f_{10}(x)$.

Fig. 3.73. Q-Q Plot pentru $f_{100}(x)$ Fig. 3.74. Q-Q Plot pentru $f_{10}(x)$

Analizând cele două reprezentări, putem observa o aliniere satisfăcătoare a eșantioanelor dealungul dreptelor de referință.

Suplimentar, aplicăm testul Crammer – von Mises pentru evaluarea gradului de potrivire a distribuției selectate pentru „întârziere – 100 Mb/s”. Obținem valorile $W^2 = 0,07897$ și $p = 0,1932$. Identificăm în tabelul de valori critice, $W_{CM} = 0,214$. Rezultă $W^2 < W_{CM}$ iar $P > \alpha$ ceea ce va conduce la afirmația „nu putem respinge H_0 ”.

În cazul distribuției normale selectate pentru „întârziere – 10 Mb/s”, aplicând testul Shapiro – Wilk [45], obținem $W = 0,9309$ și $P = 0,6$. Rezultatul $P > \alpha$ ne conduce la afirmația „nu putem respinge H_0 ”.

Relațiile (3.19) și (3.20) ne ajută să definim expresia generală a funcției densitate de probabilitate ce caracterizează factorul de răspuns „întârziere”, expresia acesteia fiind compusă din $f_{100}(x)$ și $f_{10}(x)$, domeniile de existență fiind indicate de parametrii din tabelul 3.9.

Obținem:

$$f(x) = \begin{cases} 0,4284 \left(\frac{x}{2,7343}\right)^{0,1713} e^{-\left(\frac{x}{2,7343}\right)^{1,1713}}, & 0 \leq x \leq \mu \\ 0,117e^{-\frac{(x-30,13)^2}{23,256}}, & -\sigma\sqrt{3} \leq x - \mu \leq \sigma\sqrt{3} \end{cases} \quad (3.21)$$

3.3.3.2 Analiza efectelor individuale

Vom urmări în continuare determinarea principalelor efecte introduse de către factorii de influență asupra valorilor medii ale răspunsurilor „eficiența transmisiei” și „întârziere”, pentru fiecare factor în parte.

Utilitarul JMP ne permite reprezentarea grafică a eșantioanelor ce definesc factorii de răspuns, sub forma de „box – plot”, cu marcarea liniilor mediane și ilustrarea tendinței de variație a acestora în funcție de nivelurile stabilite ale factorilor de influență.

Figura 3.75 reunește reprezentările grafice aferente influenței exercitate asupra eficienței transmisiei de către: a) topologie; b) tipul distribuitorului; c) viteza mediului de comunicație; d) viteza de prelucrare a pachetelor; e) protocol.

Pentru o ilustrare cât mai sugestivă, valorile medii ale eșantioanelor urmărite sunt conectate prin segmente de dreaptă, marcate pe figură cu albastru, acestea indicând în mod explicit influența fiecărui factor în mod particular.

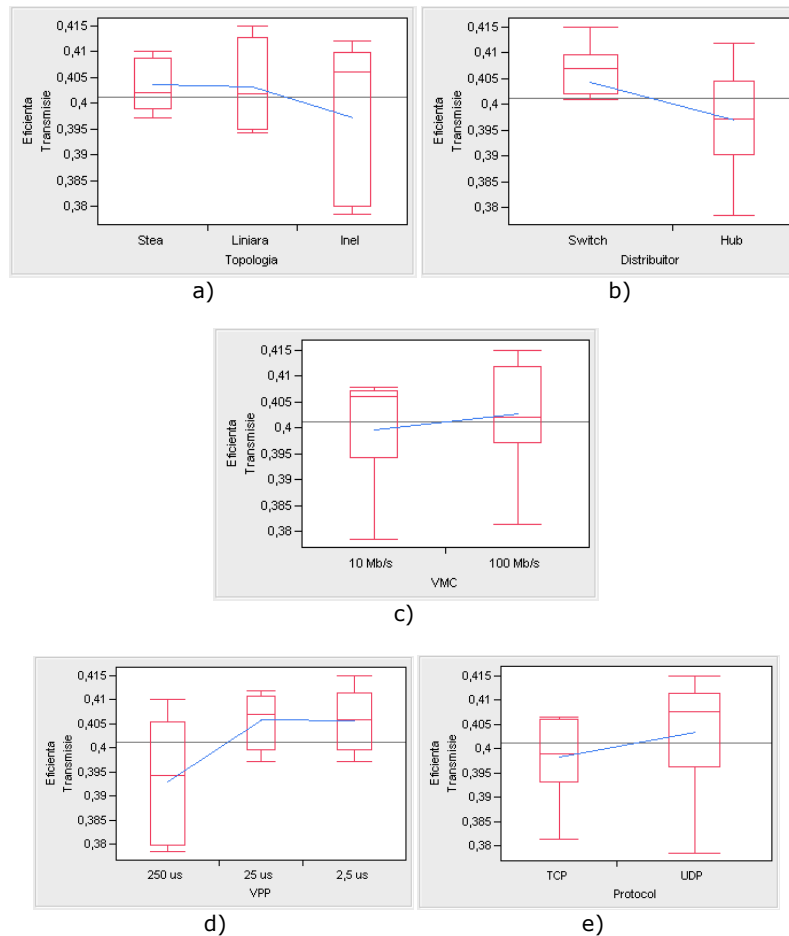


Fig. 3.75. Efecte individuale pentru eficiența transmisiei

În figura 3.76 sunt redate reprezentările grafice aferente influenței exercitate asupra întârzierii introduse în rețea de către: a) topologie; b) tipul distribuitorului; c) viteza mediului de comunicație; d) viteza de prelucrare a pachetelor; e) protocol.

Pe baza acestor reprezentări putem deduce modul de influență pozitiv sau negativ a factorilor de răspuns de către parametrii stabiliți inițial, funcție de dispunerea segmentelor care unesc mediile eșantioanelor. Prezența segmentelor orizontale indică exercitarea unei influențe minore sau chiar deloc a acelor parametri asupra factorilor de răspuns.

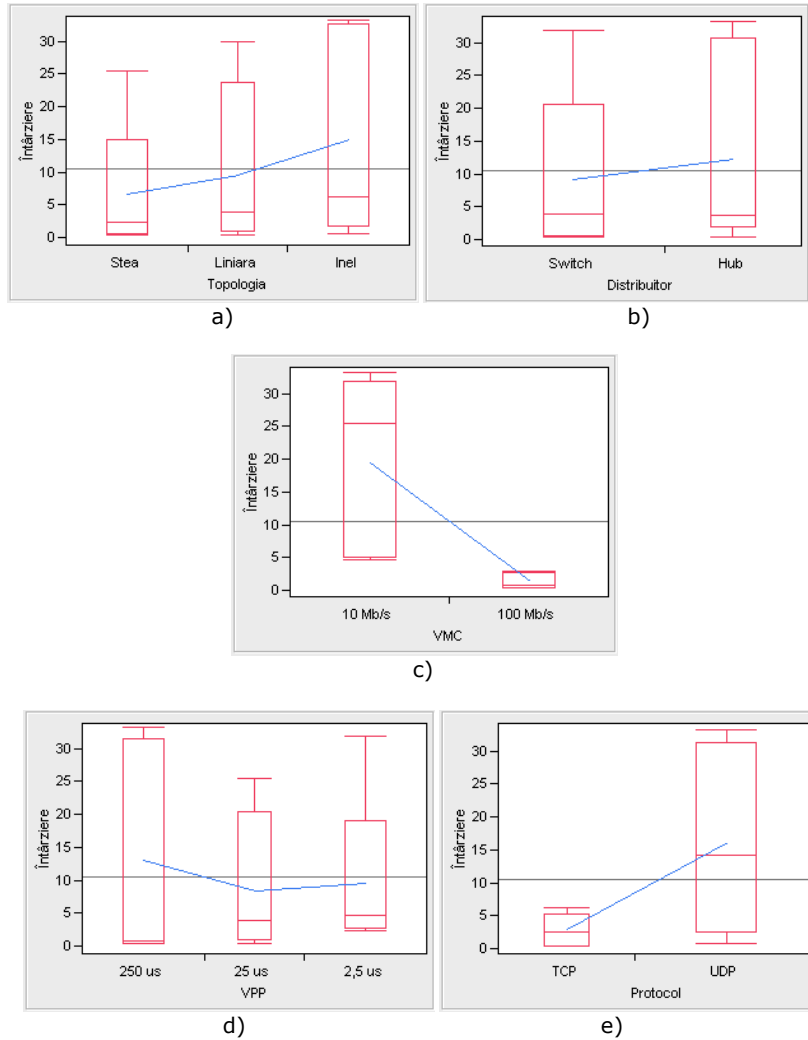


Fig. 3.76. Efecte individuale pentru întârziere

Pentru o interpretare mai bună, am sintetizat sub forma tabelului 3.12 principalele efecte exercitate în mod individual asupra eșantioanelor de către factorii de influență selectați.

	Topologia			Distribuitor		VMC		VPP			Protocol	
	Stea	Linia	Inel	Switch	Hub	10Mb/s	100Mb/s	250μs	25μs	2,5μs	TCP	UDP
Eficiența transmisiei	X	X	-	+	-	X	X	-	X	X	-	+
Întârzierea	+	-	-	X	X	-	+	-	X	X	+	-

Tabelul 3.12. Tabelul de influență

Consultând tabelul 3.12 putem observa că anumiți parametri sau niveluri de variație ale acestora nu influențează factorii de răspuns (rubrici marcate cu X), unii îmbunătățesc performanțele (rubrici marcate cu „+”) iar anumite configurații reduc aceste performanțe (rubrici marcate cu „-”).

Utilitatea informațiilor obținute și reunite în tabelul 3.12 constă în posibilitatea proiectantului de a elimina factorii care conduc la scăderea performanțelor rețelei de teren concepute, prin considerarea efectelor cumulate ai acestor parametri atât asupra eficienței transmisiei cât și asupra întârzierii introduse în rețea. Spre exemplu putem concluziona că o topologie în inel pentru o astfel de rețea va fi defavorabilă, reducând eficiența transmisiei și măbind valoarea întârzierii. Similar, un distribuitor de tip Hub va fi de asemenea defavorabil. Totodată se vor evita mediile de comunicație ce asigură viteze mai mici sau egale cu 10Mb/s și se va urmări eficientizarea subsistemelor care prelucrează informația într-un interval de timp mai mare sau egal cu 250 μ s. În cazul protocolului de comunicație adoptat se pot observa influențe opuse asupra eficienței transmisiei față de întârziere, iar alegerea finală va fi impusă de specificul aplicației.

Un alt aspect al analizei constă în identificarea factorului cu influența cea mai mare asupra răspunsului urmărit.

Pentru eficiența transmisiei obținem ca un prim rezultat reprezentările grafice ale mediilor pătratice (notate pe diagrame cu LS – Least Squares Means) în raport cu fiecare factor de influență, așa cum este ilustrat în figura 3.77. Această medie ne ajută să identificăm mai bine influența individuală a termenilor asupra formării nivelului mediu [46], [47].

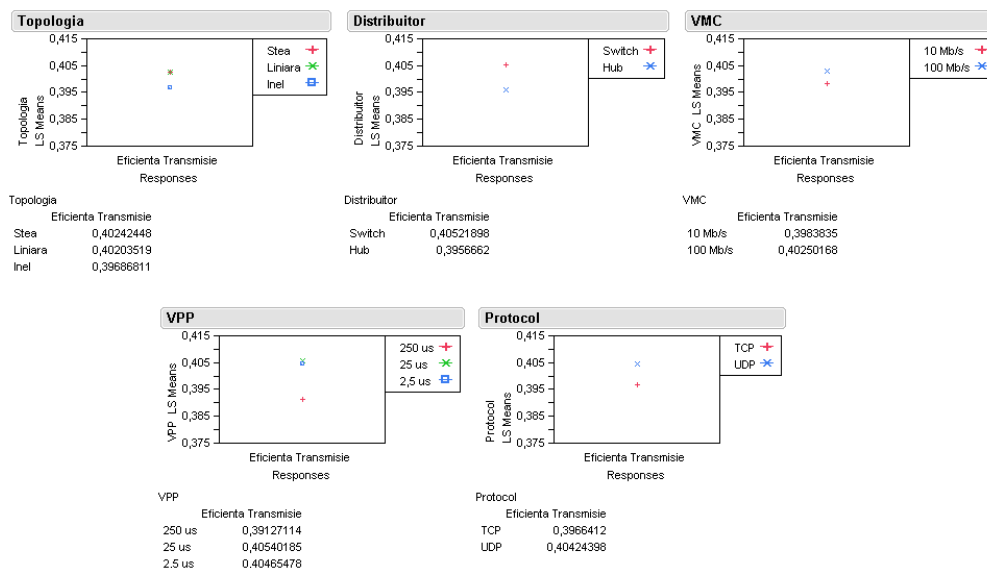


Fig. 3.77. Diagrame de influență bazate pe mediile pătratice – eficiența transmisiei

Consultând figura 3.77, pe baza distanței dintre punctele marcate aferente fiecărui factor de influență în parte, putem determina care dintre aceștia exercită cel mai mare efect asupra răspunsului eficiența transmisiei.

Rezultă cel mai important factor de influență asupra eficienței transmisiei ca fiind parametrul VPP (viteza de prelucrare a pachetelor).

Pentru factorul de răspuns întârziere, vom efectua același set de analize. Figura 3.78 ilustrează diagramele de influență bazate pe mediile pătratice.

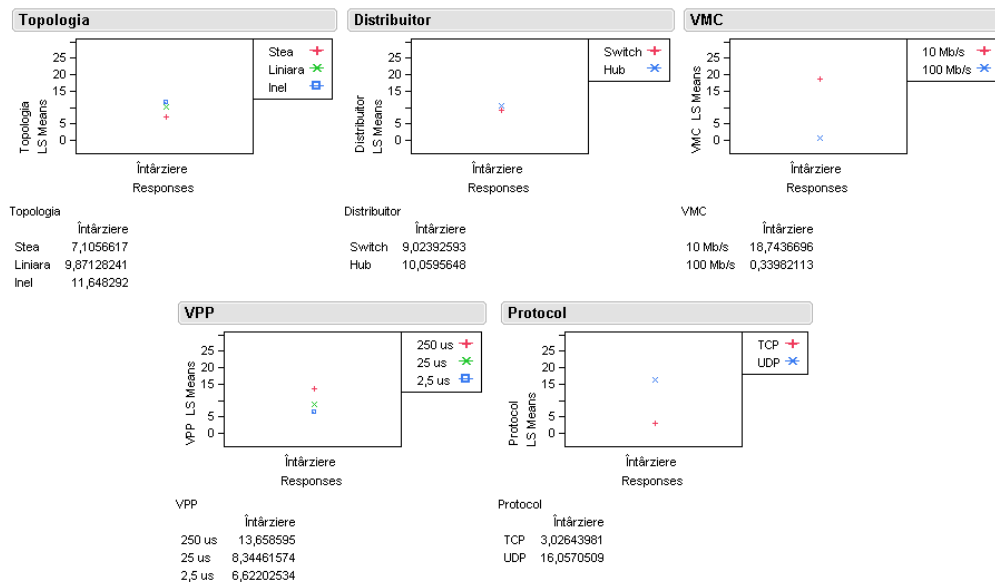


Fig. 3.78. Diagrame de influență bazate pe mediile pătratice – întârziere

Diagramele expuse mai sus conduc la cel mai important factor de influență asupra întârzierii ca fiind parametrul VMC (viteza mediului de comunicație).

3.3.3.3 Identificarea efectelor de interacțiune

Pentru efectuarea unei analize complete precum și pentru conceperea unui model statistic, un rol important îl joacă și determinarea posibilelor efecte introduse de către interacțiunile dintre factorii de influență asupra eșantioanelor de răspuns. În acest sens vom genera pentru fiecare factor de răspuns în parte câte o diagramă de probabilitate „half – normal” [42], [45], [47] care ne permite separarea efectelor importante de cele nesemnificative asupra unei variabile de răspuns.

Apelând opțiunea „Analyze – Modeling – Screening” din JMP și specificând factorii de influență împreună cu răspunsul urmărit, obținem pentru eficiența transmisiei diagrama din figura 3.79a respectiv diagrama din figura 3.79b în cazul întârzierii.

Pe baza acestor diagrame putem extrage informații referitoare la factorii ce vor fi incluși în modelul statistic de predicție a celor două răspunsuri analizate. Observăm existența unor efecte de interacțiune semnificative la nivel de doi factori (evidențiate prin termeni produs).

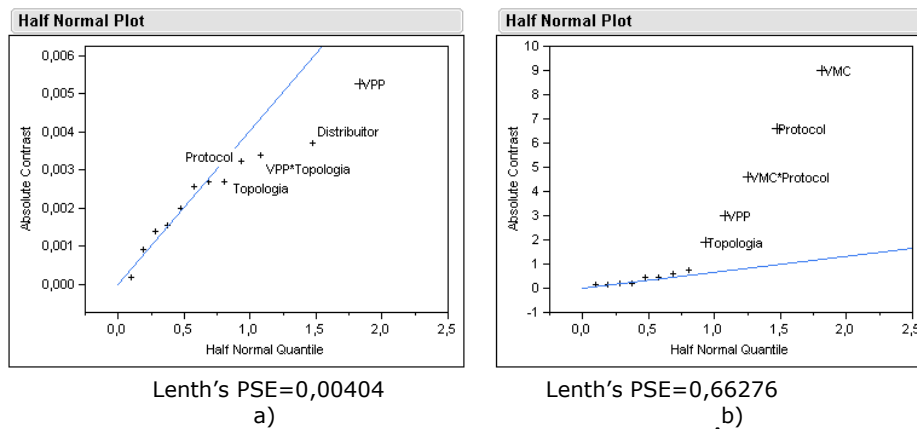


Fig. 3.79. Half-normal Plot: a) Eficiența Transmisiei; b) Întârziere

Reperete marcate pe diagrame care sunt aliniate de-a lungul drepte (care corespunde valorii Lenth's PSE specifice) au influență neglijabilă asupra răspunsului urmărit [42], [45].

Pentru a veni în sprijinul proiectantului, am considerat utilitatea expunerii unor diagrame de interacțiune sugestive referitoare la efectele factorilor de influență asupra variabilelor de răspuns selectate, construite pe baza informațiilor colectate apriori și conform literaturii de specialitate [46], [48].

Consultând aceste diagrame putem confirma existența unor efecte de interacțiune la nivel de doi termeni, așa cum rezultă și din figura 3.79. Deducem de asemenea și factorii care nu influențează variabilele de răspuns și care vor fi excluși din modelul statistic. Se pot urmări cel mai favorabil precum și cel mai defavorabil caz în selectarea combinațiilor între topologie, distribuitor, protocol etc. asupra parametrilor de răspuns eficiența transmisiei și întârziere.

Figura 3.80 ilustrează diagramele de interacțiune pentru eficiența transmisiei, iar figura 3.81 pentru răspunsul întârziere.

Pentru factorul de răspuns eficiența transmisiei observăm din diagramele de interacțiune aferente lipsa unei influențe semnificative provenind de la elementul VMC (segmentele sunt suprapuse) precum și prezența unei influențe datorate interacțiunii elementelor VPP și Topologia (VPP · Topologia).

Pentru factorul de răspuns întârziere, figura 3.80 ne indică o lipsă a influenței din partea elementului Distribuitor precum și existența unui efect de interacțiune rezultat din produsul termenilor VMC și Protocol (VMC · Protocol).

Rezultatele obținute pe baza acestor două diagrame vin în completarea informațiilor furnizate de către reprezentările grafice din figura 3.79 (Half – Normal Plot), ambele conducând la aceleași concluzii.

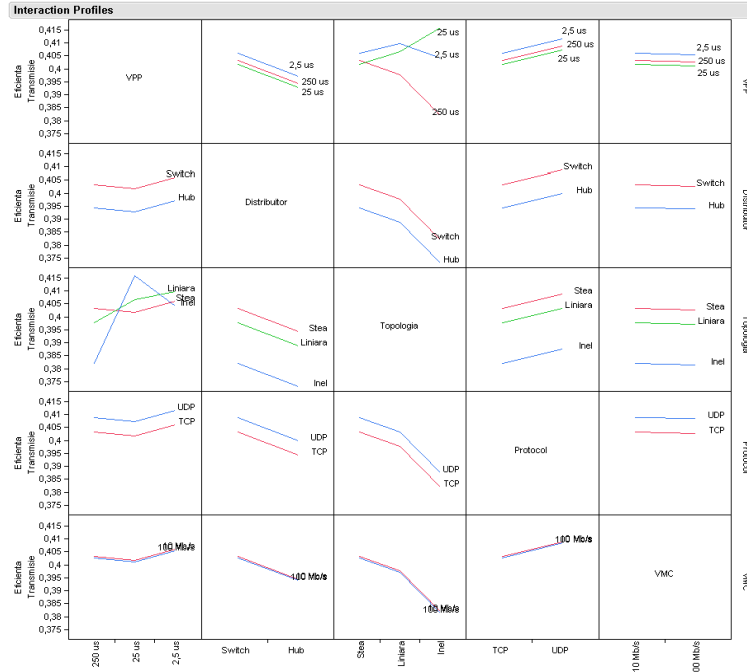


Fig. 3.80. Diagrame de interacțiune pentru eficiența transmisiei

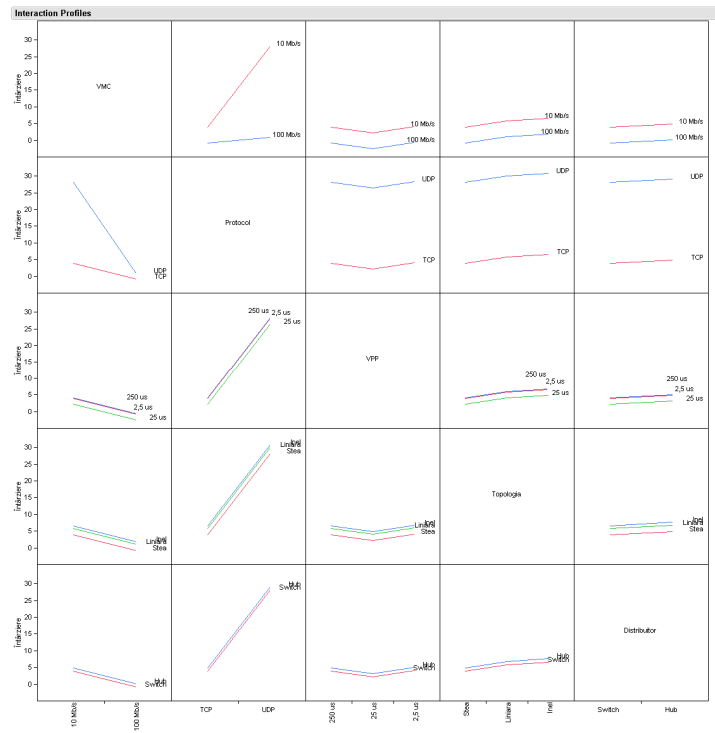


Fig. 3.81. Diagrame de interacțiune pentru întârziere

3.3.4 Conceperea modelului statistic de predicție

Pe baza rezultatelor obținute în paragrafele anterioare, tabelul 3.13 sintetizează pentru fiecare răspuns în parte efectele principale precum și cele de interacțiune ale factorilor de influență considerați în analiză.

	Răspuns	
	Eficiența transmisiei	Întârziere
Factori incluși	VPP Distribuitor VPP*Topologia Protocol Topologia	VMC Protocol VMC*Protocol VPP Topologia
Factori excluși	VMC	Distribuitor

Tabelul 3.13. Efecte ale factorilor de influență

Analiza derulată se bazează pe un model liniar, uzual în astfel de situații [41], [45]. Totuși, o variabilă independentă discretă având mai mult de două niveluri de variație, așa cum se utilizează și în lucrarea de față, nu poate fi introdusă direct într-un model liniar [49]. Informația inițială va trebui reconstituită din combinațiile unor variabile suplimentare, dihotomice, ce derivă din variabila primară. O variabilă discretă, cu m niveluri de variație va genera $m-1$ variabile dihotomice care vor reflecta printr-o codificare 0/1 sau -1/1 prezența sau lipsa unei anumite categorii [49]. Un exemplu simplu este regăsit în literatura de specialitate, [49], expunând cazul unei variabile cu trei niveluri care va genera două variabile noi, cu două niveluri, așa cum este ilustrat în tabelul 3.14.

În cazul nostru, variabilele aferente sunt factorii de influență topologie și VPP, având fiecare trei niveluri de variație.

Nivel	Codificare inițială	Variabila fictivă 1	Variabila fictivă 2
N_1	1	1	0
N_2	2	0	1
N_3	3	0	0

Tabelul 3.14. Generarea unor variabile suplimentare

Avantajul utilizării programelor moderne dedicate prelucrărilor statistice constă în posibilitatea utilizării unui concept integrant – modelul liniar general (GLM) care permite automatizarea generării variabilelor fictive astfel încât la nivel de utilizator această problemă devine transparentă [49]. Utilitarul JMP înglobează aceste caracteristici și ne va sprijini în continuarea analizei și conceperii modelului statistic aferent factorilor de răspuns selectați în experimente.

Datorită naturii diferite a celor doi factori de răspuns precum și a combinațiilor între termenii care stabilesc valoarea răspunsurilor (așa cum rezultă și din tabelul 3.13), vom elabora modele de predicție distincte pentru fiecare răspuns în parte. Punctul de plecare comun este modelul liniar general, GLM, precum și

metoda celor mai mici pătrate ca algoritm de regresie pentru determinarea coeficienților modelelor.

Expresia matematică descrie variabila dependentă, Y , ca sumă a trei componente: (1) termenul liber denumit și intercept [42], [43]; (2) suma ponderată a variabilelor independente (factorii de influență) și (3) eroarea. Pentru k variabile independente, ecuația fundamentală aferentă GLM este redată prin relația:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k + \varepsilon \quad (3.22)$$

unde:

α reprezintă termenul liber (o constantă matematică);

$\beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k$ este suma ponderată a variabilelor independente;

ε indică eroarea de predicție.

Relația (3.22) poate fi echivalată cu expresia descrisă de ecuația (3.2) deoarece variabilele independente incluse în modelul GLM pot conține transformări (sau combinații) neliniare ale variabilelor inițiale, determinate experimental, precum și sume sau produse ale acestora [43]. Rezultă avantajul principal al GLM în posibilitatea introducerii în ecuația (3.22) a efectelor de interacțiune identificate prin generarea unor noi variabile.

Considerând factorii de influență ca variabilele independente ale experimentelor derulate, vom atribui acestora, conform tabelului 3.15 un set de descriptori care ne va permite punerea în ecuație conform relației (3.22).

Descriptor	Factor de influență
X_1	Topologia
X_2	Distribuitor
X_3	VMC
X_4	VPP
X_5	Protocol

Tabelul 3.15. Set de descriptori pentru η_T

În cele ce urmează, pe baza relației (3.22), a tabelului 3.14 și a tabelului 3.15 vom parcurge etapele proiectării modelului statistic atât pentru eficiența transmisiei cât și pentru întârziere.

Pentru modelul obținut prezintă interes următoarele:

- variabilele care intră în model;
- coeficientul de corelație multiplă (R^2) care indică gradul total de potrivire al modelului;
- coeficientul de corelație multiplă pătratic ajustat la numărul de termeni (R^2 adj.);
- eroarea standard a estimării;
- reziduurile, care specifică abaterile modelului de la realitate.

3.3.4.1 Modelul statistic pentru eficiența transmisiei

În expresia matematică a modelului liniar vom folosi ca notație comună pentru variabile litera Z pentru a evita eventualele confuzii cu factorii inițiali X .

Consultând tabelul 3.13 deducem existența pentru eficiența transmisiei a cinci variabile în model și obținem următoarele echivalări:

$$\begin{aligned} Z_1 &= X_1 \\ Z_2 &= X_2 \\ Z_3 &= X_1 \cdot X_4 \\ Z_4 &= X_4 \\ Z_5 &= X_5 \end{aligned}$$

Elaborăm modelul liniar general de predicție aferent răspunsului eficiența transmisiei, \hat{Y}_η , pe baza relației (3.22), conținând variabilele Z , și termenul liber.

$$\hat{Y}_\eta = \alpha + \beta_1 Z_1 + \beta_2 Z_2 + \beta_3 Z_3 + \beta_4 Z_4 + \beta_5 Z_5 \quad (3.23)$$

Trebuie avut în vedere faptul că factorii de influență (variabilele independente) sunt multicategoriali, cu două sau trei niveluri de variație. Acest lucru conduce la o formă matriceală pentru variabilele Z , având o coloană și două sau trei rânduri. Similar, coeficienții β vor fi descriși de matrici linie, cu mai multe coloane.

Apelăm în continuare la opțiunea „Fit Model – Model Specification” în utilitarul JMP, specificând variabilele independente precum și răspunsul Y urmărit. Alegem metoda celor mai mici pătrate pentru regresia liniară, fereastra interactivă fiind ilustrată în figura 3.82.

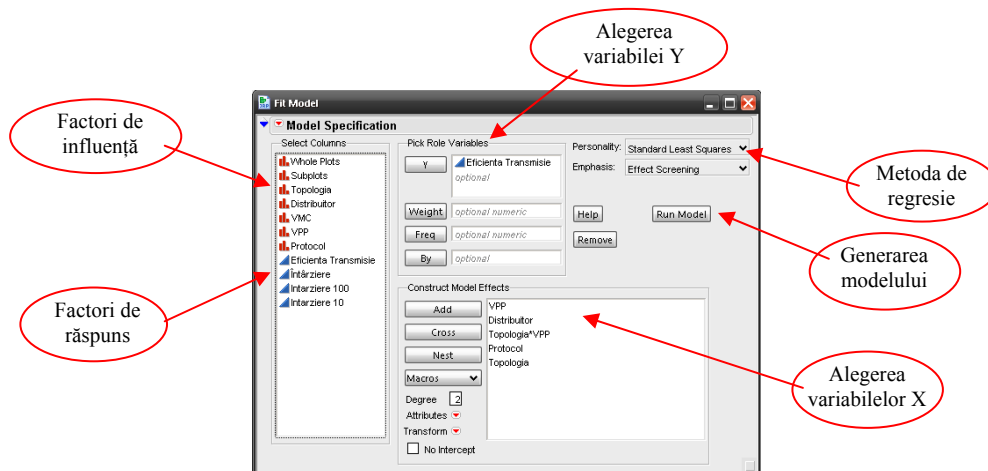


Fig. 3.82. Specificarea modelului în JMP

Generăm modelul, prin selectarea opțiunii „Run Model” și vom obține rapoarte de analiză, diagrame de potrivire și coeficienții modelului.

Ne oprim la principalii parametri descriptivi, prezentați în tabelul 3.16.

Pornind de la semnificația statistică a lui R^2 , putem obține în procente gradul total de potrivire al modelului ca fiind $R^2 \cdot 100$ (%), în cazul nostru obținând o valoare satisfăcătoare de 99%.

Informații utile obținem și prin analiza varianței, cu parametri specifici unui model liniar [46], [48], sintetizați în tabelul 3.17.

Parametrul	Valoare
Rsquare (R^2)	0,99063509
RSquare Adj (R^2 adj)	0,9594187
Root Mean Square Error	0,00216559
Mean of Response	0,4012
Observations (Sum Wgts)	14

Tabelul 3.16. Parametrii descriptivi pentru Y

Sursa	GDL	Suma pătratelor	Media pătratică	Valoarea F
Model	10	0,001488	0,000149	31,73446
Error	3	1,41E-05	4,69E-06	Prob > F
C. Total	13	0,001502	---	0,007996

Tabelul 3.17. Analiza varianței cu parametrii specifici pentru Y

Tabelul 3.17 aduce informații comparative între modelul statistic propus și un model standard, raportat doar la o singură valoare medie, procedeu uzual în domeniul statisticii [46], [48]. Distingem următoarele rubrici:

- sursa, semnificând cele trei surse de variație: Model, Error respectiv C.Total;
- GDL (grade de libertate), parametru asociat fiecărei surse de variație, după cum urmează:
 - i) pentru Model, GDL semnifică numărul parametrilor (fără intercept) utilizați în model;
 - ii) pentru Error, GDL semnifică diferența dintre totalul gradelor de libertate, C. Total și cele atribuite pentru Model. Error reprezintă estimația dispersiei pentru repetiția erorilor și este pătratul erorii standard a estimației [49];
- suma pătratelor (SP), indicând o sumă de pătrate asociată fiecărei surse de variație, fiind o măsură a variabilității detectate (măsurate) în răspuns. Este de fapt suma pătratelor diferențelor între răspunsul estimat și cel măsurat (obținut experimental). Valoarea totală a SP (pentru C.Total) reprezintă suma pătratelor diferențelor între valorile estimate și cele actuale, corespunzând practic erorii rezultate (Error) în urma potrivirii modelului de regresie. Valoarea SP atribuită pentru Model va fi SP C.Total – SP Error;
- media sumelor de pătrate, definită ca o sumă de pătrate divizată cu numărul gradelor de libertate asociate fiecărei surse de variație. Acest raport transformă suma pătratelor într-o medie – media sumelor de pătrate;
- valoarea F, obținută ca rezultatul raportului dintre media pătratică a modelului (Model) și cea a erorii (Error). Această valoare prezintă o distribuție F [45], [49] iar în cazul în care termenii modelului introduc efecte semnificative, valoarea F va fi mai mare decât cea rezultată din pură întâmplare (asociată cu pragul de semnificație $\alpha = 0,05$);
- Prob>F indică probabilitatea de a obține valori mari pentru F doar din pură întâmplare decât din modelul de regresie specificat. Valori Prob>F mai mici sau egale cu 0,05 sunt considerate a fi evidența existenței a cel puțin unui factor de regresie semnificativ în modelul analizat [49].

O observație importantă referitoare la rezultatele obținute în tabelul 3.17, exprimată pe baza informațiilor din literatura de specialitate [45], [49] constă în definirea obiectivului țintă a acestei analize a varianței. Valori mari ale SP Model raportate la valori relativ reduse ale SP Error conduc la valori F ridicate și valori p foarte reduse, ceea ce reprezintă și scopul în sine a acestui test de varianță – determinarea certitudinii ca termenii modelului să fie semnificativ diferiți de zero.

Pentru modelul de estimare aferent variabilei de răspuns eficiența transmisiei, tabelul 3.17 ne indică valori satisfăcătoare în cazul parametrilor Prob>F respectiv Valoarea F care se încadrează în tiparul cerințelor expuse mai sus.

Coeficienții modelului de estimare pentru eficiența transmisiei, obținuți în urma regresiei, precum și eroarea standard asociată fiecăruia, sunt sintetizați în tabelul 3.18.

Termen	Coeficient	Eroare std.
Intercept	0,4011	0,00070152
VPP[250 us]	-0,0087429	0,00086039
VPP[25 us]	0,00505714	0,00090946
VPP[2,5 us]	0,00368571	0,00088876
Distribuitoar[Switch]	0,00441429	0,00115755
Distribuitoar[Hub]	-0,0044143	0,00115755
Topologia[Stea]*VPP[250 us]	0,00809286	0,00142933
Topologia[Stea]*VPP[25 us]	-0,0067357	0,00177053
Topologia[Stea]*VPP[2,5 us]	-0,0013571	0,00223513
Topologia[Liniara]*VPP[250 us]	0,0018881	0,00162435
Topologia[Liniara]*VPP[25 us]	-0,003119	0,00161669
Topologia[Liniara]*VPP[2,5 us]	0,00123095	0,00177053
Topologia[Inel]*VPP[250 us]	-0,009981	0,0011823
Topologia[Inel]*VPP[25 us]	0,00985476	0,00162435
Topologia[Inel]*VPP[2,5 us]	0,00012619	0,00142933
Protocol[TCP]	-0,0027786	0,00081851
Protocol[UDP]	0,00277857	0,00081851
Topologia[Stea]	0,00068571	0,00088876
Topologia[Liniara]	0,00169048	0,00090946
Topologia[Inel]	-0,0023762	0,00086039

Tabelul 3.18. Coeficienții modelului de estimare pentru η_T

Tabelul 3.18 ne va permite în cele ce urmează, extragerea matricilor de coeficienți β și apoi construirea modelului statistic în forma finală.

Obținem:

$$\begin{aligned} \beta_1 &= [0,000685 \quad 0,00169 \quad -0,00237] \\ \beta_2 &= [0,00441 \quad -0,00441] \\ \beta_3 &= [0,00809 \quad -0,00673 \quad -0,00135 \quad 0,00188 \quad -0,00311 \quad 0,00123 \quad -0,00998 \quad 0,00985 \quad 0,000126] \quad (3.24) \\ \beta_4 &= [-0,00874 \quad 0,00505 \quad 0,00368] \\ \beta_5 &= [-0,00277 \quad 0,00277] \end{aligned}$$

Matricile asociate variabilelor independente (factorii de influență) sunt construite pe baza nivelurilor de variație corespunzătoare fiecărei variabile, conținând un singur element 1 în poziția nivelului de variație selectat și 0 în rest. Vor rezulta matrici coloană, având după caz două sau trei linii, după cum urmează:

$$Z_1 = \begin{bmatrix} A(-1) \\ A(0) \\ A(1) \end{bmatrix} \quad Z_2 = \begin{bmatrix} B(-1) \\ B(1) \end{bmatrix} \quad Z_3 = \begin{bmatrix} A(-1) \cdot D(-1) \\ A(-1) \cdot D(0) \\ A(-1) \cdot D(1) \\ A(0) \cdot D(-1) \\ A(0) \cdot D(0) \\ A(0) \cdot D(1) \\ A(1) \cdot D(-1) \\ A(1) \cdot D(0) \\ A(1) \cdot D(1) \end{bmatrix} \quad Z_4 = \begin{bmatrix} D(-1) \\ D(0) \\ D(1) \end{bmatrix} \quad Z_5 = \begin{bmatrix} E(-1) \\ E(1) \end{bmatrix} \quad (3.25)$$

Elementele matricilor Z corespund notațiilor din tabelul 3.4. Selectarea unui anumit nivel de variație pentru un factor de influență (A, B, D sau E) se realizează prin atribuirea valorii 1 pentru acel nivel și 0 în rest.

De exemplu, pentru o topologie de tip inel vom avea $A(1) = 1$, $A(-1) = 0$ și $A(0) = 0$ iar pe baza expresiilor din (3.25) vom avea pentru matricea Z_1 forma:

$$Z_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Corelând relația (3.23) cu expresiile din (3.25) și cunoscând matricile β ale coeficienților, cu valorile elementelor precizate în (3.24), putem scrie forma finală a modelului de predicție pentru eficiența transmisiei:

$$\hat{Y}_\eta = 0,4011 + \beta_1 \begin{bmatrix} A(-1) \\ A(0) \\ A(1) \end{bmatrix} + \beta_2 \begin{bmatrix} B(-1) \\ B(1) \end{bmatrix} + \beta_3 \begin{bmatrix} A(-1) \cdot D(-1) \\ A(-1) \cdot D(0) \\ A(-1) \cdot D(1) \\ A(0) \cdot D(-1) \\ A(0) \cdot D(0) \\ A(0) \cdot D(1) \\ A(1) \cdot D(-1) \\ A(1) \cdot D(0) \\ A(1) \cdot D(1) \end{bmatrix} + \beta_4 \begin{bmatrix} D(-1) \\ D(0) \\ D(1) \end{bmatrix} + \beta_5 \begin{bmatrix} E(-1) \\ E(1) \end{bmatrix} \quad (3.26)$$

Modelul dat de relația (3.26) poate fi utilizat de către proiectant în predicția unei valori pentru eficiența transmisiei, așa cum este definită aceasta prin expresia (3.14), pentru diverse configurații ale rețelelor de teren, prin simpla completare cu 0 sau 1 a matricilor Z ce descriu factorii de influență.

Pentru o analiză suplimentară a calității modelului stabilit, vom apela și la o metodă grafică de reprezentare a valorilor actuale versus celor prognozate ale variabilei de răspuns eficiența transmisiei. Figura 3.83 ilustrează diagrama „Actual by Predicted”, având marcate dreapta de regresie, media eșantioanelor (linia orizontală), intervalul de încredere de 95% precum și punctele de interes pentru cele 14 eșantioane.

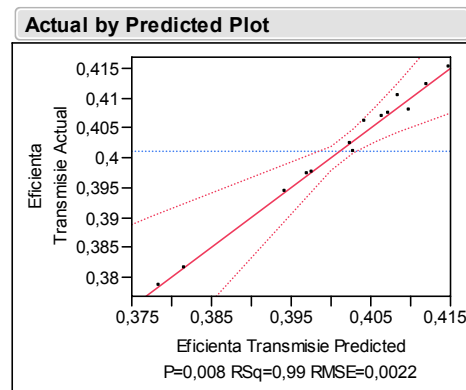


Fig. 3.83. Diagrama „Actual by Predicted” pentru eficiența transmisiei

Modelul de predicție este satisfăcător dacă toate punctele marcate în diagrama din figura 3.83 sunt aliniat de-a lungul dreptei de regresie (trasată cu linia continuă roșie) și se găsesc în interiorul zonei separate de liniile punctate care delimitează intervalul de încredere de 95%. Observăm ca acest lucru este îndeplinit în totalitate și confirmă astfel încă odată validitatea modelului.

Disponând de expresia modelului, putem trasa și o diagramă suprapusă a celor două mărimi: variabila Y_{η} actuală și cea estimată, \hat{Y}_{η} , reprezentată în figura 3.84 (Overlay Plot). Observăm și pe această cale gradul ridicat de potrivire al modelului generat.

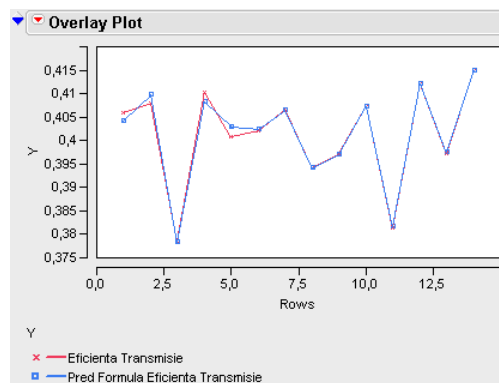


Fig. 3.84. Diagrama „Overlay Plot” pentru eficiența transmisiei

3.3.4.2 Modelul statistic pentru întârziere

Pe baza informației din tabelul 3.13, similar paragrafului 3.3.4.1, vom defini variabilele auxiliare Z, pentru construirea modelului aferent răspunsului întârziere.

Obținem următoarele echivalări:

$$Z_1 = X_1$$

$$Z_2 = X_3$$

$$Z_3 = X_3 \cdot X_5$$

$$Z_4 = X_4$$

$$Z_5 = X_5$$

Expresia modelului liniar general (3.22) ne ajută să construim modelul de estimare a variabilei întârziere, care va avea forma:

$$\widehat{\Delta}_{max} = \alpha + \beta_1 Z_1 + \beta_2 Z_2 + \beta_3 X_3 + \beta_4 Z_4 + \beta_5 X_5 \quad (3.27)$$

Revenind la opțiunea „Fit Model – Model Specification” în utilitarul JMP, specificând variabilele independente conform tabelului 3.13 precum și răspunsul Y urmărit, în cazul nostru $\widehat{\Delta}_{max}$, generăm modelul prin „Run Model” și vom obține informații similare celor din paragraful 3.3.4.1.

Parametrii ce descriu statisticile generale sunt redați în tabelul 3.19.

Parametrul	Valoare
Rsquare (R^2)	0,99412311
RSquare Adj (R^2 adj)	0,98726674
Root Mean Square Error	1,48198473
Mean of Response	10,4605
Observations (Sum weights)	14

Tabelul 3.19. Parametrii descriptivi pentru Δ_{max}

Gradul total de potrivire al modelului rezultă a fi în proporție de 99,4%, indicat de către parametrul R^2 , așa cum rezultă din tabelul 3.19.

Sintetizăm în tabelul 3.20 informațiile obținute și prin analiza varianței, cu parametri specifici modelului liniar [46], [48], cu semnificațiile descrise în paragraful 3.3.4.1.

Sursa	GDL	Suma pătratelor	Media pătratică	Valoarea F
Model	8	2229,10958	318,444225	144,992628
Error	5	13,1776724	2,19627873	Prob > F
C. Total	13	2242,28725	---	2,89827e-6

Tabelul 3.20. Analiza varianței cu parametrii specifici pentru Δ_{max}

Examinând valorile parametrilor obținuți în tabelul 3.20, pentru modelul de estimare aferent variabilei de răspuns întârziere (Δ_{max}), obținem valori satisfăcătoare în cazul indicatorilor Prob>F respectiv Valoarea F care se încadrează în tiparul cerințelor expuse mai sus.

Coefficienții modelului de estimare pentru întârziere, obținuți în urma regresiei, precum și eroarea standard asociată fiecăruia, se regăsesc în tabelul 3.21.

Urmează extragerea matricilor de coeficienți β și apoi construirea modelului statistic în forma finală.

Termen	Coeficient	Eroare std.
Intercept	9,46328788	0,40277181
VMC[10 Mb/s]	8,0057268	0,42058412
VMC[100 Mb/s]	-8,0057268	0,42058412
Protocol[TCP]	-6,6094545	0,40277181
Protocol[UDP]	6,60945455	0,40277181
VPP[250 us]	0,46269728	0,64570418
VPP[25 us]	-1,1343636	0,61411625
VPP[2,5 us]	0,67166635	0,64570418
Topologia[Stea]	-1,5788629	0,57128425
Topologia[Liniara]	0,39230303	0,61411625
Topologia[Inel]	1,18655983	0,57128425
VMC[10 Mb/s]*Protocol[TCP]	-5,6392165	0,48321437
VMC[10 Mb/s]*Protocol[UDP]	5,63921649	0,48321437
VMC[100 Mb/s]*Protocol[TCP]	5,63921649	0,48321437
VMC[100 Mb/s]*Protocol[UDP]	-5,6392165	0,48321437

Tabelul 3.21. Coeficienții modelului de estimare pentru Δ_{max}

Avem:

$$\begin{aligned}
 \beta_1 &= [-1,5788629 \quad 0,39230303 \quad 1,18655983] \\
 \beta_2 &= [8,0057268 \quad -8,0057268] \\
 \beta_3 &= [-5,6392165 \quad 5,63921649 \quad 5,63921649 \quad -5,6392165] \\
 \beta_4 &= [0,46269728 \quad -1,1343636 \quad 0,67166635] \\
 \beta_5 &= [-6,6094545 \quad 6,60945455]
 \end{aligned} \tag{3.28}$$

Matricile Z asociate variabilelor independente X vor avea următoarea structură:

$$Z_1 = \begin{bmatrix} A(-1) \\ A(0) \\ A(1) \end{bmatrix} \quad Z_2 = \begin{bmatrix} C(-1) \\ C(1) \end{bmatrix} \quad Z_3 = \begin{bmatrix} C(-1) \cdot E(-1) \\ C(-1) \cdot E(1) \\ C(1) \cdot E(-1) \\ C(1) \cdot E(1) \end{bmatrix} \quad Z_4 = \begin{bmatrix} D(-1) \\ D(0) \\ D(1) \end{bmatrix} \quad Z_5 = \begin{bmatrix} E(-1) \\ E(1) \end{bmatrix} \tag{3.29}$$

Pe baza expresiilor (3.27), (3.28) și (3.29) putem concepe forma finală a modelului de predicție a răspunsului întârziere, care va avea forma:

$$\widehat{\Delta}_{max} = 9,4632 + \beta_1 \begin{bmatrix} A(-1) \\ A(0) \\ A(1) \end{bmatrix} + \beta_2 \begin{bmatrix} C(-1) \\ C(1) \end{bmatrix} + \beta_3 \begin{bmatrix} C(-1) \cdot E(-1) \\ C(-1) \cdot E(1) \\ C(1) \cdot E(-1) \\ C(1) \cdot E(1) \end{bmatrix} + \beta_4 \begin{bmatrix} D(-1) \\ D(0) \\ D(1) \end{bmatrix} + \beta_5 \begin{bmatrix} E(-1) \\ E(1) \end{bmatrix} \tag{3.30}$$

Figura 3.85 ilustrează diagrama „Actual by Predicted”, având marcate dreapta de regresie, media eşantioanelor (linia orizontală), intervalul de încredere de 95% precum și punctele de interes pentru cele 14 eşantioane, prin care putem evalua și grafic gradul de potrivire al modelului.

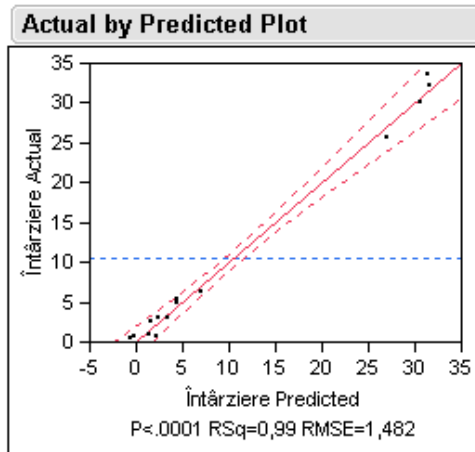


Fig. 3.85. Diagrama „Actual by Predicted” pentru întârziere

Modelul stabilit prin relația (3.30) ne permite generarea diagramei „Overlay Plot” (figura 3.86) prin care reprezentăm atât variația eșantioanelor actuale (experimentale) cât și a celor rezultate pe baza predicției.

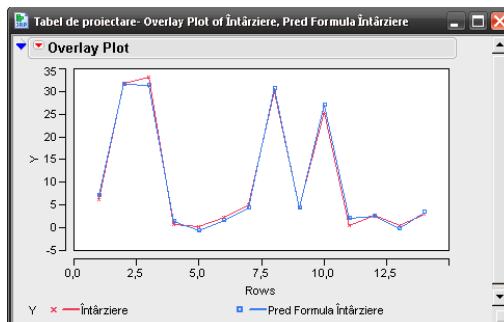


Fig. 3.86. Diagrama „Overlay Plot” pentru eficiența transmisiei

3.4 Concluzii și contribuții personale

Diversitatea factorilor care influențează în mod deosebit performanțele rețelelor de teren conduce la necesitatea identificării și elaborării unor noi metode de analiză și predicție comportamentală adecvate. Tendința actuală este urmărirea unor obiective cum ar fi:

- determinarea metodei corespunzătoare de alcătuire a structurilor și a combinațiilor funcționale între diverse dispozitive din componența rețelelor de teren în scopul evitării pierderii unor informații vitale ce descriu efectele factorilor de influență asupra indicatorilor de performanță stabiliți;
- modul în care se analizează rezultatele experimentale;
- stabilirea unor modele care țin cont de mărimea efectelor introduse de factorii de influență selectați.

În urma acestor analize rezultă diverși indicatori care stau la baza obținerii unor soluții optime pentru îmbunătățirea performanțelor globale ale rețelelor de teren.

În capitolul de față am urmărit în linii mari elaborarea unei proiectări experimentale, bazate pe DOE (Design of Experiment), cu un algoritm de selecție de tip D-Optimal care conduce la obținerea unor tabele de date structurate și permite generarea de modele robuste, pe baza unui număr relativ redus de experimente.

În acest sens, am parcurs câteva etape primordiale și anume:

- i) identificarea variabilelor de maximă importanță și a valorilor optime ale acestora, definind astfel obiectivul investigației;
- ii) selectarea variabilelor care vor fi monitorizate pentru a descrie rezultatele experimentelor, acestea alcătuind variabilele de răspuns;
- iii) aplicarea unei metode standard de analiză adecvată din punct de vedere al numărului de variabile precum și a preciziei în furnizarea unui rezultat (model) optim;
- iv) verificarea rezultatelor obținute prin simulare și comparare cu cele existente din literatura de specialitate.

În urma realizării experimentelor și a finalizării analizei statistice, am obținut modele de predicție a indicatorilor de performanță a rețelelor de teren care înglobează numeroși factori de influență considerați importanți în literatura de specialitate și care conduc la rezultate cu o abatere relativ redusă față de comportarea reală.

Utilitatea modelelor constă în posibilitatea proiectantului de a estima pe baza lor, valori ale indicatorilor de performanță cum ar fi eficiența transmisiei sau întârzierea din rețea, pentru orice structură în care intervin combinații ale factorilor de influență stabiliți în analiză, fără a fi necesară derularea experimentului ci doar înlocuirea în model a datelor obținute sub formă matricială.

3.4.1 Contribuții personale

Pe parcursul întregului capitol am urmărit o expunere gradată a elementelor ce intervin în procesul de analiză derulat cu scopul modelării statistice a rețelelor de teren și a obținerii unei estimări pentru parametrii selectați, rezultând anumite contribuții personale sintetizate în cele ce urmează.

1. **Identificarea**, pe baza unei analize anterioare (Capitolul II), a **principalilor factori de influență și indicatori de performanță** corespunzători rețelelor de teren conforme standardului IEC 61784;
2. **Aplicarea proiectării experimentale (DOE)** bazate pe metoda **D-Optimal** la analiza rețelelor de teren, **fapt nemenționat la ora actuală** în literatura de specialitate;
3. **Selectarea și utilizarea** unui software de analiză și modelare statistică pe calculator, care s-a dovedit a fi **suficient de performant** pentru scopul urmărit: SAS JMP;
4. **Transpunerea** datelor inițiale în tabele aferente JMP;
5. **Efectuarea experimentelor** impuse în urma aplicării metodei D-Optimal cu ajutorul mediului de simulare **OpNet**;
6. **Urmărirea simultană a doi indicatori de performanță** definiți prin **eficiența transmisiei și întârzierea din rețea**;

7. **Stabilirea legii de distribuție și determinarea expresiei matematice** a funcției densitate de probabilitate asociată celor doi factori de răspuns concretizate prin relațiile (3.16) și (3.21);
8. **Generarea și interpretarea** diagramelor care indică atât efectele principale cât și cele de interacțiune ale factorilor de influență, asupra variabilelor de răspuns, expuse în figurile 3.75, 3.76, 3.77, 3.78, 3.80 și 3.81;
9. **Conceperea** unui „tabel de influență”, cu notații intuitive, care descrie caracterul efectelor introduse de către factorii de influență (tabelul 3.12);
10. **Elaborarea expresiilor modelelor** pentru factorii de răspuns eficiența transmisiei și întârziere, descrise de relațiile (3.26) și (3.30), prin **introducerea** unor **descriptori matriciali** ai variabilelor independente (factorii de influență). Prin utilizarea acestor forme matriciale, proiectantul are posibilitatea de a stabili extrem de simplu structura rețelei pentru care dorește predicția indicatorilor de performanță;
11. **Verificarea** gradului de **potrivire** al acestor modele, atât cu ajutorul a două **metode grafice** cât și prin **interpretarea** parametrilor statistici ce exprimă erorile de predicție.

Prin elementele parcurse în Capitolul III am evidențiat unele aspecte noi legate de modalitățile de analiză globală a rețelelor de teren, privite din perspectiva a două elemente cheie: DOE cu metoda D-Optimal și analiza statistică, **neutilizate până în prezent în acest domeniu**. S-a deschis calea spre dezvoltări ulterioare a acestui concept în privința aplicării metodei combinate și pentru identificarea, analiza și predicția a numeroși alți factori de influență/răspuns, specifici fiecărei aplicații în parte.

4. SUBSISTEME INTELIGENTE PENTRU OPTIMIZAREA CONDUCERII PROCESELOR TEHNOLOGICE

4.1. Introducere

Pentru monitorizarea proceselor tehnologice aferente sistemelor industriale moderne este necesară existența unui număr mare de senzori prin intermediul cărora se colectează informația provenită din mediul înconjurător. Cu cât sistemul monitorizat este mai complex, cu atât și inteligența senzorilor și volumul de date prelucrat la nivel local trebuie să crească în mod corespunzător.

Un sistem modern de măsurare și control poate fi conceput având ca bază structura bloc prezentată în figura 4.1 [8]. Pe magistrala digitală, o unitate centrală comunică în mod direct cu un anumit număr de periferice (dispozitive de intrare sau ieșire). Această unitate preia informații de la diverse tipuri de senzori și în urma unei prelucrări corespunzătoare, comandă dispozitivele actuator.

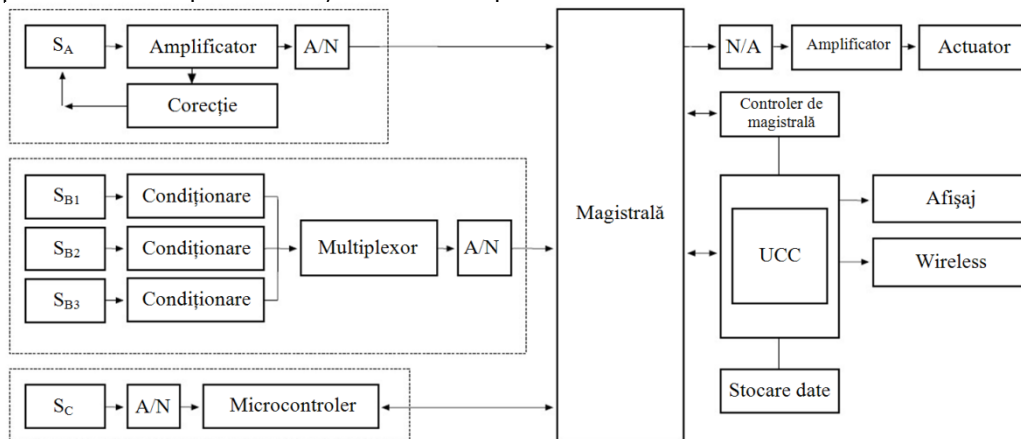


Fig.4.1. Structura bloc a unui sistem de măsurare și control digital

Sistemul prezentat cuprinde trei blocuri de preluare de date, prin intermediul unor diverși senzori, S_A , S_B și S_C care prelucrează informația prin diverse modalități și apoi o transmit la o magistrală de date. De exemplu, senzorul S_A furnizează un semnal de ieșire care este amplificat și apoi supus unor corecții (de offset, liniarizare etc.) urmând să fie transformat într-un flux de date numeric cu ajutorul unui convertor A/N. Grupul de senzori S_{B1}, \dots, S_{B3} este cuplat la circuite de condiționare a semnalului independente cu ieșirile conectate la un singur multiplexor analogic, care apoi generează semnalul de intrare a blocului de conversie A/N. Ultimul modul cuprinde senzorul S_C conectat la blocul de conversie A/N obținându-se astfel un semnal numeric preluat de către un microcontroler care aplică corecțiile necesare pe baza datelor de calibrare stocate în memoria proprie ROM. O caracteristică majoră a acestei configurații, este posibilitatea de a prelua informații de pe magistrală prin intermediul microcontrolerului local și de a le retransmite

senzorului pentru diverse ajustări necesare în timpul procesului de măsurare (de exemplu modificarea domeniului de măsură, recalibrare, ajustarea offsetului etc.). Totodată, un astfel de subsistem este capabil să recunoască un anumit format de adresare, având posibilitatea de a detecta comenzi provenite de la o unitate centrală.

În mod similar, actuatorul conectat la sistem este comandat prin preluarea de pe magistrală a datelor numerice și conversia acestora prin intermediul blocului N/A care furnizează un semnal analogic amplificat ce se aplică actuatorului. Cu ajutorul unor microcontrollere pe 8 biți se pot implementa sisteme de comandă pentru actuatore bazate pe motoare de curent continuu sau motoare pas cu pas, o aplicație fiind descrisă în lucrarea „**Using the PIC16F84 Microcontroller in Intelligent Stepper Motor Control**”, elaborată de către autor pe parcursul conducerii tezei de doctorat [51].

În prezent, majoritatea componentelor unui sistem bazat pe magistrală, sunt încă separate și au un înveliș fizic propriu. Tendința de dezvoltare pe viitor este integrarea acestor blocuri într-un singur chip și aplicarea unui standard universal [8]. Această concepție este marcată în figura 4.1 prin liniile întrerupte.

Totuși, doar integrarea pe un singur chip a acestor componente nu rezolvă în totalitate problematica interconectării acestor subsisteme. Existența unei vaste diversități de producători de senzori și actuatore, cu standarde și principii de comunicare proprii, împiedică la ora actuală atingerea scopului expus. O modalitate eficientă de cvasi-unificare o reprezintă înglobarea unei așa numite „inteligente” proprii ale dispozitivului care cuprinde un senzor sau actuator, rezultând astfel noțiunea de transductor inteligent, conectabil direct la magistrală și adresabil de către o unitate centrală de comandă și control. În acest sens la ora actuală este elaborat și utilizat standardul IEEE 1451, cu diverse diviziuni în curs de adoptare [50]. Odată cu acest standard sunt introduse dispozitive și denumiri specifice: NCAP (Network Capable Application Processor), Smart Transducer, TIM (Transducer Interface Module), TII (Transducer Independent Interface) și TEDS (Transducer Electronic Data Sheet), elemente descrise pe larg de către autor în „**Soluții actuale de prelucrare a informației în sistemele de conducere a proceselor tehnologice**” [44]. Dispozitivul cheie al acestui standard îl constituie „transductorul” inteligent care nu reprezintă doar elementul care transformă o formă de energie în alta, ci dobândește și proprietatea de a prelucra local informația și de a comunica bidirecțional cu o unitate de control ierarhic superioară. Astfel, transductorul inteligent poate fi atât senzor cât și actuator [44].

Dispozitivele utilizate în controlul proceselor tehnologice industriale urmăresc atingerea compatibilității cu standardul IEEE 1451 dar există încă numeroase situații în care trebuie asigurată adaptarea la noul standard al vechilor structuri „moștenite”. Producătorii de senzori și actuatore furnizează o gamă extrem de variată de dispozitive bazate încă pe standarde vechi printre care MODBUS, PROFIBUS, WordLIP, Hart etc. nefiind interschimbabile și nici compatibile în mod direct cu cerințele IEEE 1451. Există în schimb posibilitatea introducerii unor dispozitive auxiliare, cu denumirea de „subsisteme inteligente”, cu o structură hibridă, care pe de o parte mențin compatibilitatea cu vechile elemente și pe de altă parte respectă cerințele standardului IEEE 1451. Prin introducerea acestora în componența globală a sistemelor de conducere a proceselor tehnologice se poate asigura tranziția la noul standard prin costuri minime datorită posibilității utilizării în paralel atât a vechilor dispozitive cât și a celor noi.

Pe parcursul acestui capitol vom evidenția cerințele actuale ale standardului IEEE 1451, cu expunerea unor modalități de implementare bazate pe

microcontrolere de cost redus ale diverselor module funcționale impuse de specificațiile IEEE 1451 în scopul elaborării interfeței standardizate a unui subsistem inteligent capabil să realizeze o comunicație pe o rețea de teren de tip Ethernet industrial.

În prezent, la nivel mondial, protocolul HART (Highway Addressable Remote Transducer) este încă prezent în structura sistemelor de producție industriale de importanță majoră cum ar fi: industria farmaceutică, chimică, alimentară, de prelucrare a petrolului, electronică etc. [52]. Fiind de o importanță majoră, am sintetizat pe parcursul capitolului, câteva aspecte fundamentale aferente acestui protocol cu scopul ulterior de a concepe un subsistem inteligent, bazat pe FPGA (Field Programmable Gate Array) care va realiza compatibilizarea cu standardul IEEE 1451 și va îngloba concepte noi, originale, expuse în paragrafele ulterioare.

În încheiere vom stabili prin simulare și implementare pe o placă ML-401 conținând un FPGA Virtex – 4, produs de Xilinx, parametrii de funcționare și performanțele subsistemului elaborat.

4.2. Standardul industrial IEEE 1451

Obiectivele proiectelor IEEE P1451 urmăresc definirea unui set comun de interfețe de comunicare pentru interconectarea transductorilor cu sisteme bazate pe microprocesoare, instrumente de măsurare și rețele de teren, într-un sistem independent de magistrală. Ca urmare, nu vor exista cerințe restrictive asupra utilizării diferitelor dispozitive cum ar fi convertoare analog – numerice, microprocesoare, circuite de condiționare etc. Țelul final este de a crea mijloacele pentru atingerea unei compatibilități universale transductor – magistrală [50]. Comitetul Tehnic al Tehnologiilor Senzoriale al IEEE (Institute of Electrical and Electronics Engineer) derulează o serie de proiecte noi, sub denumirea de IEEE P1451, care au ca principal scop dezvoltarea unei familii de standarde pentru conectarea transductorilor inteligenți la magistrale industriale [53]. Aceste standarde se împart în subfamilii, într-o continuă extindere și îmbunătățire la ora actuală. Odată cu finalizarea adoptării acestor noi standarde, producătorii de transductori inteligenți vor avea posibilitatea proiectării sub un singur set de specificații universale a dispozitivelor de monitorizare.

Standardul IEEE 1451 pentru interconectarea transductorilor inteligenți este format din următoarele subdiviziuni:

- IEEE 1451.0 Unificarea formatelor și a protocoalelor, 2005
- IEEE 1451.1 Definirea modelelor informaționale, 1999
- IEEE 1451.2 Interfața, 1997 – revizuit în prezent
- IEEE 1451.3 Rețea locală pentru transductori, 2004
- IEEE 1451.4 Mod mixt (analog/digital) și TEDS, 2005
- IEEE 1451.5 Wireless, propus din 2004, sub revizuire în prezent
- IEEE 1451.6 CAN pentru rețeaua de teren, propus din 2004
- IEEE 1451.7 RFID, actual doar propus, fără specificații concrete

Pe parcursul elaborării standardului, grupul de lucru a aderat la câteva principii fundamentale printre care: adoptarea unei abordări independente față de tipul de rețea de comunicație în conceperea standardului aferent interconectării transductorilor; considerarea unor soluții scalabile care utilizează concepte adaptive pentru furnizarea unor soluții simple pentru anumite clase de aplicații și totodată o largă varietate de opțiuni de implementare pentru alte clase; independența față de limbajul de programare utilizat în implementarea standardului; utilizarea exclusivă a

foii de catalog electronice, TEDS (Transducer Electronic Data Sheet), pentru specificarea caracteristicilor transductorilor; implementarea unui algoritm de corecție pentru compensarea caracteristicilor non – ideale ale transductorilor [50].

Figura 4.2 prezintă elementele conceptuale ale standardului IEEE 1451.

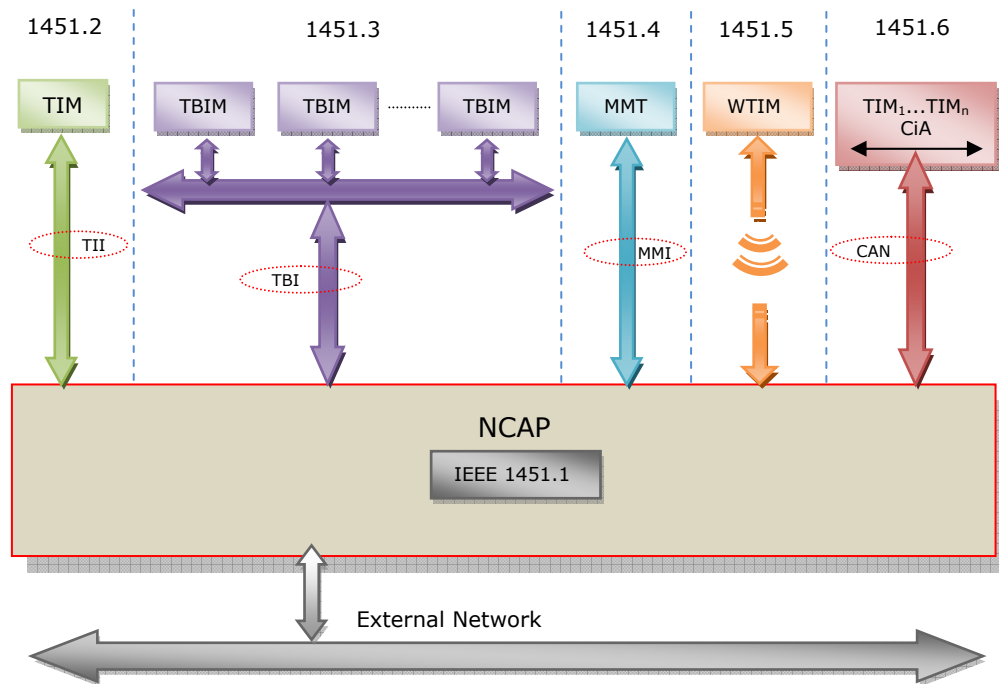


Fig.4.2. Elementele de bază ale standardului IEEE 1451

Esența conceptului IEEE 1451 constă în existența modului NCAP (Network Capable Application Processor) căruia îi este asociat un model informațional de tip obiect (Object Model), definit prin diviziunea IEEE 1451.1. Rolul NCAP este de a asigura comunicarea dispozitivelor atașate acestuia cu o rețea de teren oarecare (External Network), preferabil de tip Ethernet industrial (vezi Capitolul 3). Totodată modulul controlează local toate perifericele conectate (transductorii) prin intermediul unor interfețe de comunicare simplificate, specifice fiecărei subdiviziuni a standardului.

Distingem în figura 4.2 subdiviziunile standardului IEEE 1451, care au un rol bine definit și în principiu nu pot coexista în cazul aceluiași NCAP.

IEEE 1451.2 definește interfața clasică TII (Transducer Independent Interface), prin care un bloc TIM (Transducer Interface Module) se conectează la modulul NCAP. Specificația 1451.2 prevede posibilitatea conectării doar a unui singur TIM la NCAP printr-o astfel de interfață TII. Scopul TII este de a asigura independența tipului de transductor ales față de caracteristicile mediului de comunicare precum și posibilitatea conectării la un NCAP a transductorilor provenind de la diverși fabricanți [53]. În urma acestui standard, apare necesitatea existenței a câte unui NCAP pentru fiecare modul TIM, fapt ce mărește costurile de implementare hardware.

Pentru a satisface nevoia conectării mai multor TIM la un singur NCAP, s-a adoptat subdiviziunea IEEE 1451.3 care se referă la o modalitate de interconectare

pe o rețea locală a mai multor transductori prin blocurile TBIM (Transducer Bus Interface Module) care dispun de circuite specializate (transceivere) prin intermediul cărora se face posibilă această conectare la mediul fizic al rețelei, denumit TBI (Transducer Bus Interface). Sunt definite în acest sens semnale de control speciale iar NCAP trebuie să dispună de circuite de intrare/ieșire adecvate și de o logică de comandă și arbitraj specifică modului de comunicare adoptat.

Existența și la ora actuală a unui număr impresionant de sisteme de conducere bazate pe senzori analogici a impus necesitatea elaborării subdiviziunii IEEE 1451.4, mixte, ale acestui standard, care prevede posibilitatea conectării la NCAP a unor dispozitive denumite MMT (Mixed Mode Transducers) prin intermediul interfeței MMI (Mixed Mode Interface). Pentru IEEE 1451.4 s-a definit și un format special pentru TEDS [44], [50].

Dezvoltarea rapidă a sistemelor de comunicare fără fir (wireless) a condus la propunerea adoptării specificației IEEE 1451.5 dar care în momentul de față este sub revizuire datorită inexistenței unui acord asupra tipului de protocol utilizat. Astfel, în prezent, este la latitudinea utilizatorului modul de implementare a comunicării wireless, prin intermediul a unor standarde deja cunoscute: Bluetooth, ZigBee, WiFi etc. În cazul implementării unui sistem de comandă bazat pe comunicare wireless, în unele situații modulul NCAP poate să fie exclus din structura globală, așa cum rezultă și din aplicațiile elaborate de către autor în „**Using PIC18F448 In Wireless DC Motor Control**” [54] și „**Wireless Power Supply Using PIC18F448**” [55].

O tendință de înglobare a structurilor de rețele deja existente și larg răspândite în industrie se distinge prin introducerea recentă a subdiviziunii IEEE 1451.6, care prevede utilizarea rețelei CiA (CAN in Automation) pentru interconectarea la nivel de TIM și CAN (Control Area Network)/ CAN Open pentru conectarea la modulul NCAP. Această structură este utilizată cu precădere în industria automotive [50].

Vom parcurge în cele ce urmează, principalele caracteristici ale fiecărei subdiviziuni ale familiei de standarde IEEE 1451, expuse în ordinea adoptării și a importanței funcționale.

4.2.1. IEEE 1451.2 – Interfața TIM – NCAP și formate TEDS

Adoptată prima din suita IEEE 1451, în 1997, această specificație definește structura fizică a interfeței TIM, protocoalele de comunicație, formatul blocului TEDS și a datelor vehiculate între TIM și NCAP.

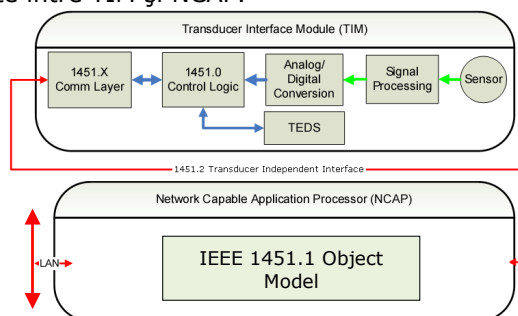


Fig.4.3. Structura TIM cu TEDS incorporat, conform IEEE 1451.2

Transducer Interface Modul, TIM, conține circuitele de condiționare a semnalului provenit de la un senzor sau circuite de ieșire pentru comanda unui actuator, un bloc de conversie analog/numerică, logica de control implementată conform specificațiilor IEEE 1451.0 împreună cu blocul TEDS și interfața fizică de conectare la TII, așa cum este prezentat în figura 4.3 [56]. La un TIM se pot conecta simultan atât senzori cât și actuatoare, în număr până la 255, care alcătuiesc fiecare în parte câte un canal de transmisie/recepție local.

TEDS (Transducer Electronic Data Sheet), stocat într-o memorie nonvolatilă atașată unui transductor, conține câmpuri ce descriu tipul, caracteristicile, modul de operare și posibilitățile de calibrare a transductorului. Informația este memorată pe maxim 256 bytes. Un transductor cu un astfel de TEDS integrat este caracterizat prin calități unice prin care este posibilă auto-identificarea față de sistemul de magistrală la care este conectat. Informația este transferată către blocul NCAP în mod automat, eliminându-se astfel necesitatea intervenției operatorului uman în desfășurarea procesului de identificare.

Interfața TII este alcătuită din zece semnale, așa cum este indicat în figura 4.4.

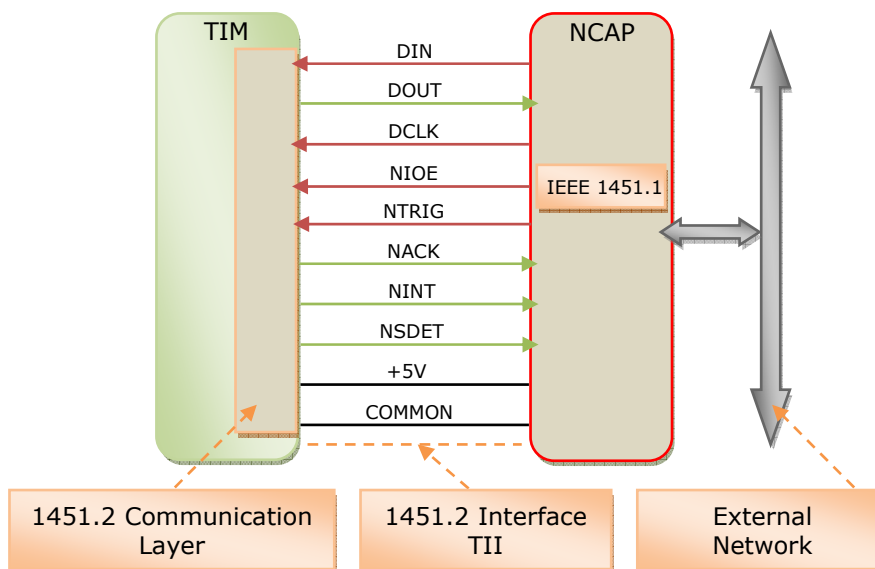


Fig.4.4. Structura interfeței TII conform IEEE 1451.2

Interfața digitală propusă [56] dispune de linii de comandă hardware pentru a permite NCAP inițierea măsurătorilor de către senzori și declanșarea efectuării acțiunilor fizice de către actuatoare. NCAP are posibilitatea de a controla un canal individual sau toate canalele transductorilor în mod simultan. Linia de tact hardware, DCLK, este comandată de către NCAP. Transferul de date este de tip sincron iar în blocul TEDS există un câmp care specifică rata maximă de transfer a datelor suportată de către TIM. Acest lucru asigură un mecanism flexibil pentru compatibilitatea a diverși NCAP cu TIM.

Atribuirea standardizată a denumirilor pinilor (menționând și culorile) aferente semnalelor vehiculate pe liniile interfeței TII este descrisă în tabelul 4.1. Transferul de date propriu-zis se realizează printr-o interfață clasică SPI (Serial Peripheral Interface), înglobată în TII prin semnalele DIN, DOUT și DCLK.

Număr Pin	Denumire Semnal	Culoare asociată	Direcție față de NCAP	Direcție față de TIM
1	DCLK	Maro	OUT	IN
2	DIN	Roșu	OUT	IN
3	DOUT	Portocaliu	IN	OUT
4	NACK	Galben	IN	OUT
5	COMMON	Verde	POWER	POWER
6	NIOE	Albastru	OUT	IN
7	NINT	Violet	IN	OUT
8	NTRIG	Gri	OUT	IN
9	POWER (+5V)	Alb	POWER	POWER
10	NSDET	Negru	IN	OUT

Tabelul 4.1. Denumiri standardizate pentru semnalele TII

Semnificația funcțională a semnalelor TII este prezentată prin intermediul tabelului 4.2.

Linia	Logică	Comandă	Funcție
DIN	Pozitivă	NCAP	Transport de adrese și date de la NCAP la TIM
DOUT	Pozitivă	TIM	Transport de date de la TIM la NCAP
DCLK	Front Crescător	NCAP	Frontul crescător validează datele DIN și DOUT
NIOE	Activ pe 0	NCAP	Semnaleză transportul activ de date și delimitează pachetele
NTRIG	Front Descrescător	NCAP	Realizează acțiunea de declanșare/interogare
NACK	Front Descrescător	TIM	Deservește două funcții: 1. Acceptarea trigerării (declanșării) 2. Acceptarea transferului de date
NINT	Front Descrescător	TIM	Cerere de servicii de la NCAP de către TIM
NSDET	Activ pe 0	TIM	Detectează prezența unui TIM
POWER	---	NCAP	Tensiunea de alimentare +5V
COMMON	---	NCAP	Referință semnal/masă

Tabelul 4.2. Semnificația funcțională a semnalelor TII

Putem observa dezavantajul introdus de numărul mare de linii a interfeței TII, care astfel concepută nu reprezintă o soluție optimă pentru conectarea la distanță mare a modulelor TIM. Datorită acestui neajuns, există tendințe de a înlocui porțiuni ale acestora cu standarde mai simple, cum ar fi RS232, RS485 [53] sau prin module wireless cu diverse standarde proprietare. Un exemplu aplicativ îl reprezintă posibilitatea utilizării modulului MRF24J40MA produs de firma Microchip în combinație cu un microcontroler PIC18F448, expus de către autor în lucrarea „**Wireless Power Supply Using PIC18F448**” [55].

Comunicația pe interfața TII este controlată de către NCAP, care poate iniția o măsurare sau o acționare prin interogarea (trigger) TIM și acesta la rândul său răspunde cu un semnal de acceptare (aknowledge). Modulul TIM poate întrerupe operațiile derulate de către NCAP doar în cazul apariției unei situații de excepție sau erori. Standardul IEEE 1451.2 definește un set de registre de stare pentru semnalarea excepțiilor standard cum ar fi: erori hardware, canale ocupate, erori de calibrare, erori de autotest [50], [53].

Schimbul de date între NCAP și TIM necesită patru linii: DCLK, DOUT, DIN și NIOE. DCLK este comandată de către NCAP. Transferul de date este bazat pe protocoale de transfer pe bit de tipul SPI (Serial Peripheral Interface).

NCAP comandă linia NTRIG pentru a iniția un proces de măsurare sau o acționare iar TIM utilizează linia NTRACK pentru a verifica dacă funcția solicitată a fost realizată cu succes. Modulul TIM notifică NCAP asupra apariției unei condiții de excepție (eroare) prin linia NIO_INT.

Tensiunea de alimentare de +5V pentru TIM se asigură prin linia POWER.

Transducer Electronic Data Sheet (TEDS) reprezintă una din inovațiile tehnice introduse de standardul IEEE 1451.2. Un TEDS, care conține informații asupra caracteristicilor și performanțelor unui transductor nu reprezintă în sine o noutate. Pe parcursul a mai mulți ani, deja diferiți producători înglobează structuri de date memorate în dispozitive diverse. Factorul inovativ constă în modelul general al unui transductor bazat pe informația din blocul TEDS care suportă o largă varietate de senzori și actuatoare [56].

TEDS conțin câmpuri care descriu în totalitate tipul, modul de operare și caracteristicile transductorului. Dacă transductorul este mutat într-o nouă locație, este deplasat împreună cu TEDS. În acest mod, informația necesară exploataării dispozitivului este întotdeauna atașată de transductor.

Figura 4.5 indică secțiunile principale adresabile ale unui TEDS împreună cu exemplificarea conținutului fiecărei secțiuni. Secțiunile punctate sunt opționale dar contribuie substanțial la îmbunătățirea exploataării dispozitivului [57].

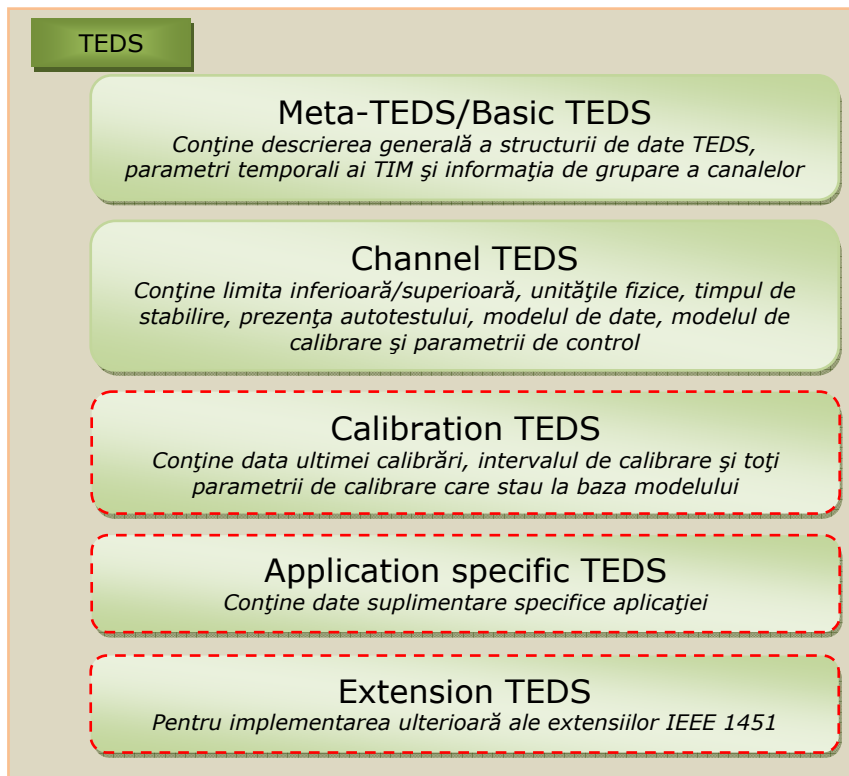


Fig.4.5. Structura TEDS conform IEEE 1451.2

Elementele TEDS definite anterior suportă o diversitate de transductori cu o structură de uz general a blocului TEDS. Această abordare conduce la o implementare foarte ușoară a sistemului în ansamblu. Dacă anumite câmpuri nu sunt necesare pentru descrierea unui transductor dat, acestea au lungimea zero în blocul de memorie TEDS, astfel neocupând spațiu de memorare inutil.

Standardul IEEE 1451.2 prevede o metodă generală pentru descrierea unităților fizice senzate sau acțiunilor efectuate de către un transductor. Metoda, exemplificată în Anexa 2, utilizează o secvență binară de 10 biți pentru a coda unitățile fizice. O unitate este reprezentată ca un produs al celor 7 unități de bază din SI (Sistemul Internațional de unități) și al celor 2 unități suplimentare din SI, fiecare ridicat la un exponent de putere rațional. Structura codează doar exponenții, produsul rezultând implicit [57]. Anexa 2 conține exemple pentru reprezentarea distanței, presiunii, accelerației etc.

Specificațiile referitoare la calibrarea din cadrul TEDS permit producătorului de transductoare să descrie o calibrare multi-dimensională pentru fiecare canal în parte. Pentru a elimina polinoame de ordin ridicat, este posibilă specificarea unei calibrări segmentate unde fiecare porțiune (segment) are o lățime și un offset variabil. Este necesară în acest caz prezența unui bloc de corecție generală în interiorul NCAP pentru interpretarea unei astfel de scheme de calibrare pentru asigurarea funcționării indiferent de tipul de transductor atașat.

O curbă de calibrare multi-segment este ilustrată în figura 4.6 [57].

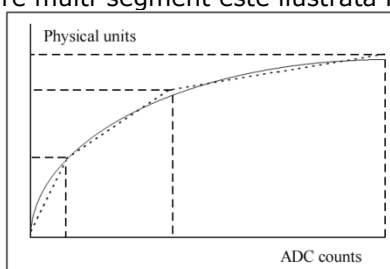


Fig.4.6. Tehnici de calibrare multi-segment

Anexa 3 conține un exemplu complet al unui TEDS compatibil IEEE 1451.2 pentru un senzor de presiune cu un singur canal. Acesta este un senzor ceramic cu o ieșire analogică între 0V și 5V DC corespunzător unei variații a presiunii de la 0 la 20.684.190 Pa. Senzorul prezintă un timp de răspuns de 10 ms iar nelinearitatea maximă este de 0,56% din V_{alim} . Componentele primare ale TIM cu un singur canal sunt: un ADC pe 12 biți (ciclul de conversie de 75 μ s) și un microcontroler PIC pe 8 biți, cu memorie EEPROM de 4K. Calibrarea este specificată prin 5 segmente egale cu offset diferit de 0 pentru fiecare segment în parte. Aceasta conduce la reducerea nelinearității de la 0,56% la aproximativ 0,03%.

Conținutul TEDS din Anexa 3 a fost creat de către firma Texas Instruments pentru a demonstra modul în care un transductor real poate fi descris prin standardul IEEE 1451.2 [57].

4.2.2. IEEE 1451.1 – Structura și modelul informațional NCAP

Subdiviziunea IEEE 1451.1, introdusă în 1999, urmărește elaborarea definiției unui obiect model comun pentru componentele unui transductor inteligent interfațabil pe magistrală atât din punct de vedere hardware cât și software [58]. Elementul de bază este blocul NCAP (Network Capable Application Processor) care

furnizează caracteristica de compatibilitate a transductorului cu magistrala. Structura simplificată a NCAP este prezentată în figura 4.7.

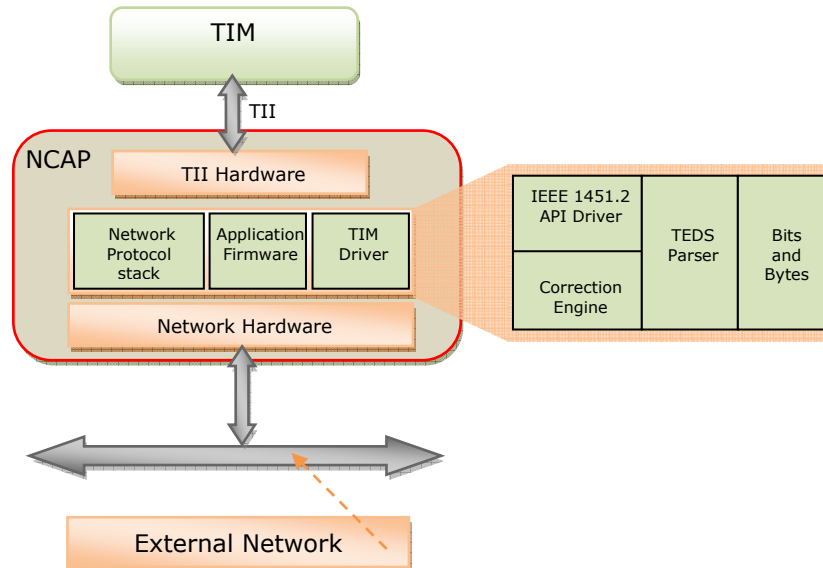


Fig.4.7. Structura NCAP conform IEEE 1451.1

Modelul obiectului transductor inteligent furnizează două interfețe:

- interfața spre blocul TIM care cuprinde detalii asupra implementării hardware ale acestuia în cadrul unui model informațional simplu. În consecință, interfața hardware a transductorului va fi privită ca un driver I/O (TIM Driver);
- interfața spre blocul NCAP și porturi care conține detalii despre diferitele implementări de protocoale de rețea susținută de un set redus de metode de comunicare.

Comportamentul specific aplicației este modelat prin blocuri funcționale. Pentru a produce comportamentul dorit, blocurile funcționale comunică și cu alte blocuri din interiorul sau exteriorul transductorului. Acest model aplicațional, independent de rețea prezintă următoarele avantaje:

- asigură un grad ridicat de interoperabilitate între transductori și rețele diverse, astfel realizând funcția "plug and play";
- simplificarea suportului pentru protocoale multiple de control al transductorilor inteligenți.

Diviziunea IEEE 1451.1 prevede elaborarea experimentală în limbajul C++ a rutinelor de implementare a specificațiilor prezentate anterior [58].

Structura software a unui NCAP conține trei blocuri: driverul TIM (TIM Driver), codul aplicației (Application Firmware) și stiva cu registre ce implementează protocolul de rețea (Network Protocol Stack). Driverul TIM este compus din patru blocuri funcționale:

- "Bits and bytes": responsabil pentru transferul bidirecțional de date pe interfață;
- "TEDS parser": conferă datelor citite din blocul TEDS al TIM conectat o interpretare logică în conformitate cu specificațiile IEEE 1451.2;
- "Correction engine": reprezintă algoritmul de conversie a șirului de biți citit din TIM în unități specificate în TEDS pentru senzori sau în unități specificate în TEDS pentru comanda unui TIM ce aparține unui actuator;

- "IEEE 1451.2 API driver": asigură accesul către blocurile TEDS, citirea informației de la senzori, controlul actuatorilor și supraveghează apariția unor cereri de întrerupere sau erori survenite în timpul funcționării sistemului.

În principiu, este necesar doar un singur driver TIM pentru fiecare tip de NCAP pentru a implementa specificațiile IEEE 1451.2. În consecință, pentru fiecare familie de microcontrolere ce suportă acest standard ar putea exista un singur driver software pentru interfața IEEE 1451.2, un singur parser TEDS și un singur conversion engine.

Blocul NCAP se poate implementa cu ajutorul unui microcontroler care dispune de un modul de comunicație pe o rețea de uz general și execută un soft adecvat, liniile de port ale acestuia constituind interfața către modulul TIM.

Modelul informațional este reprezentat printr-un set de clase de obiecte, atribute metode și comportamente care formează o noțiune abstractă ce descrie caracteristicile dispozitivelor prin termeni specifici programării orientate pe obiecte. Figura 4.8 indică structura modelului informațional, așa cum se regăsește și în literatura de specialitate [58].

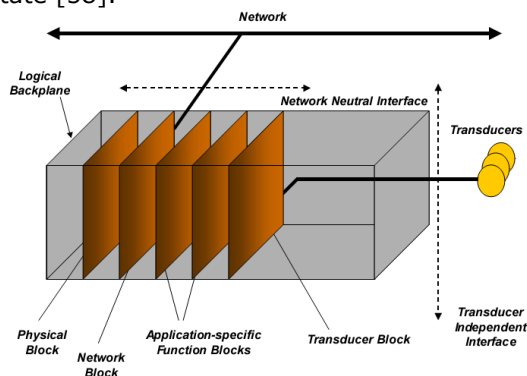


Fig.4.8. Modelul informațional NCAP conform IEEE 1451.1 [58]

Se pot distinge: blocul fizic (Physical Block), blocuri pentru unul sau mai mulți transductori, blocul funcțiilor specifice aplicației (Application – specific Function Blocks) și blocul de rețea (Network Block).

Blocul fizic asigură fundalul structurii stratificate și conține atât resursele hardware cât și elementele software ale modelului. Caracteristicile fiecărui transductor care poate fi conectat la un NCAP sunt descrise de către Transducer Block în etapa de inițializare și configurare a sistemului. Blocul funcțiilor, furnizează în linii mari tiparul pentru metode și date specifice aplicației și care pot fi apelate ori de câte ori este nevoie. Totalitatea operațiilor referitoare la comunicarea pe rețea sunt înglobate și izolate de restul funcțiilor în NCAP prin intermediul blocului de rețea (Network Block) și utilizează o interfață neutră față de tipul rețelei la care se face conectarea. Acest bloc se bazează pe principiul RPC (Remote Procedure Calls), utilizată cu precădere în sistemele de conducere distribuite [59].

4.2.3. IEEE 1451.3 Specificații pentru sisteme distribuite, multitransductor

IEEE 1451.3 urmărește elaborarea unei interfețe digitale standard pentru a conecta multiple transductoare separate fizic, într-o configurație multinod.

Există câteva probleme fundamentale care trebuie rezolvate atunci când se dorește conectarea unor noduri la o magistrală auto-identificabilă. Cea mai evidentă problemă este de a defini un protocol care permite tuturor dispozitivelor de pe magistrală o autoidentificare corectă. Totodată apare necesitatea asigurării unei viteze de operare ridicate în timpul procesului de conectare la tensiunea de alimentare a dispozitivelor, astfel încât autoidentificarea să se desfășoare în parametrii prestabiliți.

Standardul propune utilizarea unei singure linii de transmisie atât pentru alimentarea modulelor TIM atașate cât și pentru schimbul de date, acestea devenind module TBIM (Transducer Bus Interface Module). În figura 4.9 este prezentat modul de conectare a mai multor blocuri TBIM la un NCAP precum și liniile aferente interfeței TBI (Transducer Bus Interface).

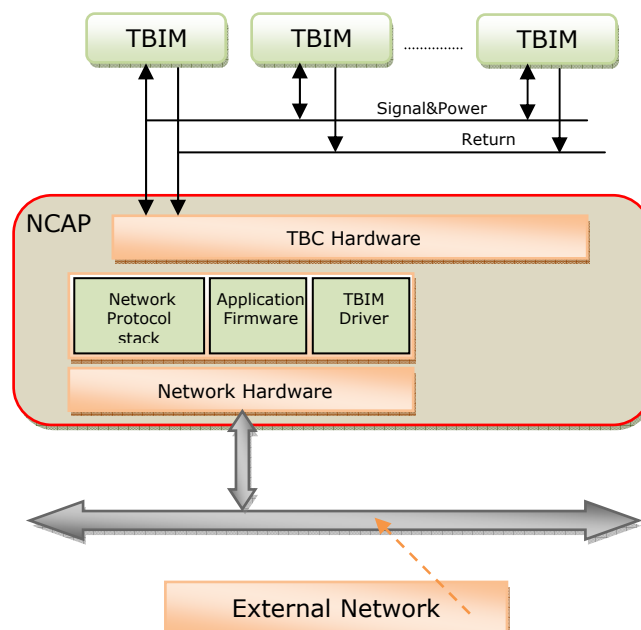


Fig.4.9. Structura distribuită conform IEEE 1451.3

Topologia magistralei TBI (Transducer Bus Interface) este liniară și se vehiculează prin intermediul acestora trei tipuri de semnale: tact pentru sincronizare, la frecvența de 2 MHz, pachetele de date și tensiunea de alimentare. Fizic, comunicația se realizează pe cablu ecranat, torsadat, pe două conductoare. Datele sunt transmise în banda între 4,75 și 9,25 MHz cu sincronizare la 2 MHz pe principiul multiplexării domeniilor de frecvență. Modul de operare poate fi de tip comandă/răspuns sau flux de date [59].

O variantă simplă de implementare a TBI este utilizarea standardului de comunicare RS 485 pentru interconectarea modulelor TBIM.

4.2.4. IEEE 1451.4 Standardul pentru interfața mixtă

În sistemele de conducere a proceselor tehnologice industriale, transductorii analogici sunt utilizați încă pe scară largă în combinație cu instrumentele electronice de măsurare a diverselor mărimi neelectrice.

De obicei, informația este transmisă unui calculator pentru evaluare. Astfel, transductorilor analogici li s-au atașat blocuri TEDS și a fost oferită posibilitatea conectării directe la magistrală, rezultând diviziunea 1451.4, așa cum se ilustrează în figura 4.10. Acesta definește o interfață pentru transductori ce vehiculează informație mixtă (analogică și numerică), cu o comunicare bidirecțională: transmiterea datelor TEDS spre calculatorul gazdă, apoi comutarea în modul de lucru analogic.

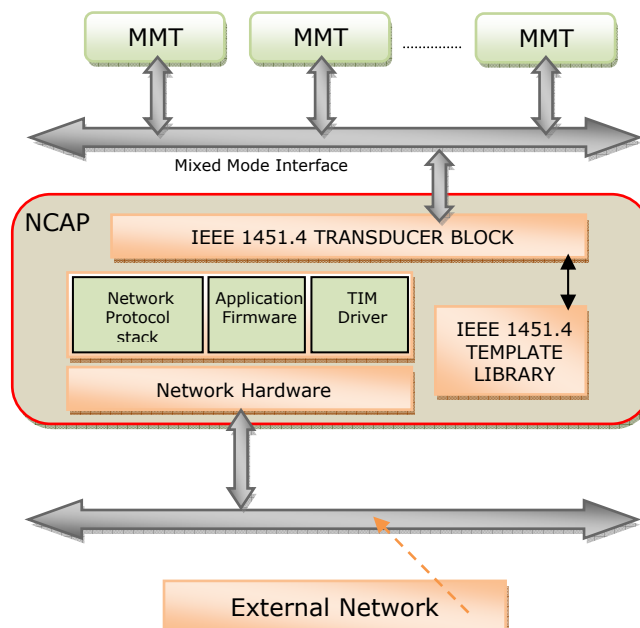


Fig.4.10. Structura IEEE 1451.4 mixtă

Transductorul MMT (Mixed Mode Transducer) conține o interfață MMI (Mixed Mode Interface) și un bloc TEDS. Fizic, interfața MMI este formată din cel puțin puțin două conductoare și respectă o conexiune serială, multipunct de tip Master – Slave. Pentru extinderea compatibilității cu diverși transductori deja existenți în domeniul analogic, specificația IEEE 1451.4 indică două clase pentru MMI: Clasa 1 cu două fire de conexiune și Clasa 2 cu multiple conexiuni [59].

Blocul TEDS are o structură fixă și pot exista mai multe noduri de memorie pe MMI care implementează câte un TEDS. Acestea conțin câmpuri care descriu tipul, operațiile și atributele unuia sau mai multor elemente transductoare (senzori sau actuatoare). Un model TEDS (Template) reunește toate obiectele software ce descriu structurile de date din TEDS. Sunt elaborate într-un limbaj denumit EDDL (Electronic Device Description Language) și rezidă într-un Transducer Block care apelează la o bibliotecă de modele, IEEE 1451.4 Template Library, așa cum rezultă și din figura 4.10. Transducer Block este un obiect software care modelează transductorul compatibil IEEE 1451.4, este conținut în NCAP și utilizat pentru accesarea, decodarea și codarea informației din TEDS în limbajul EDDL.

Pentru separarea comunicației în timp real față de Transducer Block (T-Block), este specificat un protocol IEEE 1451.4 care atribuie interpretarea informațiilor stocate în TEDS blocului T-Block iar procesarea globală a datelor revine NCAP [59].

Specificațiile introduse de către IEEE 1451.4 vin în sprijinul conceperii în cadrul lucrării de față a unui subsistem inteligent, bazat pe un NCAP, care va fi capabil să implementeze o „punte de legătură” între dispozitivele HART încă larg răspândite în domeniul industrial și noile sisteme și concepte bazate pe standardul industrial IEEE 1451.

4.2.5. IEEE 1451.5 Specificații pentru comunicarea wireless

Pentru asigurarea posibilității unei conexiuni wireless între NCAP și modulele TIM, s-a introdus recent diviziunea IEEE 1451.5 – încă sub revizuire, fiind precizate în prezent trei interfețe agreate: 802.11 (WiFi), 802.15.1 (Bluetooth) și 802.15.4 (ZigBee) [60].

O vedere de ansamblu asupra acestor specificații este prezentată în figura 4.11.

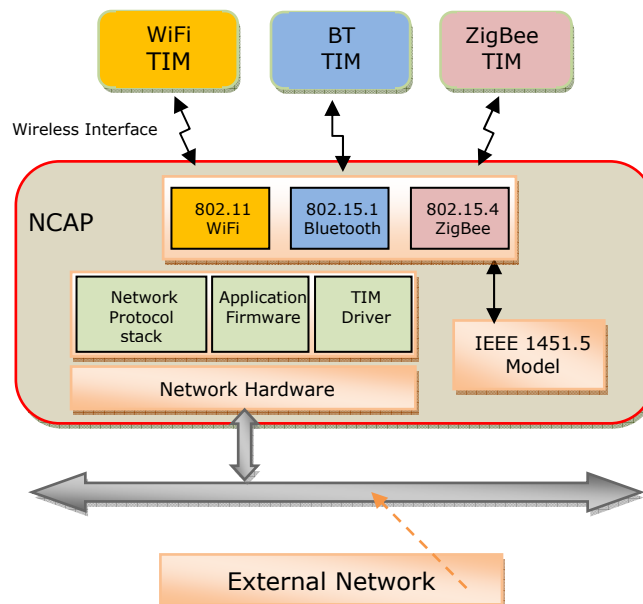


Fig.4.11. Structura IEEE 1451.5 Wireless

Operarea sistemului indicat de către specificațiile IEEE 1451.5 se bazează pe blocurile TEDS clasice, definite prin IEEE 1451.2, cu un plus de informație referitoare la caracteristicile mediului fizic de acces (PHY – Physical Interface Data Connection) pentru fiecare standard wireless în parte: 802.11 PHY TEDS, Bluetooth PHY TEDS respectiv ZigBee PHY TEDS [60].

4.2.6. IEEE 1451.0 Specificații pentru unificarea diviziunilor 1451

Principalul obiectiv al specificației IEEE 1451.0, propusă în anul 2005, este obținerea unei interoperabilități la nivelul datelor vehiculate în cazul în care sunt interconectate multiple rețele de transductori, fiecare aderând la o diviziune IEEE 1451 diferită. Acest lucru a fost realizat prin furnizarea unui set comun de comenzi, formate TEDS și protocoale de comunicație în combinație cu utilizarea unui software

partajat care implementează diversele caracteristici ale fiecărei diviziuni în parte [61]. Nivelul fizic (PHY), prin care se asigură conexiunile între TIM și NCAP, este implementat prin diverse mecanisme aferente diviziunilor standardului IEEE 1451. De exemplu, specificația IEEE 1451.2 indică utilizarea conexiunilor electrice bazate pe interfețe clasice UART (Universal Asynchronous Receiver Transmitter) sau SPI (Serial Peripheral Interface) între TIM și NCAP. Pe de altă parte, recentul IEEE 1451.5 recurge la protocoale wireless uzuale (vezi paragraful 4.2.4.) pentru realizarea acestei conexiuni. Unificarea acestor diviziuni, prin utilizarea aceluiași set de comenzi, structuri de date și protocoale de comunicație, se finalizează prin specificația IEEE 1451.0 [61].

4.2.7. Utilizarea microcontrolerelor în implementarea TIM

Având în vedere faptul că la momentul actual există o serie de microcontrolere de cost redus și performanțe relativ ridicate, înglobând memorii program și de date de tip EEPROM sau flash, convertoare A/N, module PWM și blocuri de comunicare serială UART sau SPI, se poate lua în considerare utilizarea acestora ca unitate de comandă și control în cadrul unui modul TIM. În plus, blocul TEDS se poate implementa direct în memoria de date a microcontrolerului, fără a mai fi necesară și o memorie externă.

O modalitate concretă de implementare a unui TIM, propusă de către autor și descrisă pe larg în „**Hardware implementation of a PIC18F448 based TIM for IEEE1451.2 compliant actuator control**” [62], se bazează pe utilizarea microcontrolerului PIC18F448 produs de firma Microchip. Modulul TIM elaborat permite controlul unui actuator format dintr-un motor de curent continuu comandat prin intermediul unei punți H, de tip MOSFET. Pentru controlul poziției și al gradului de încărcare au fost prevăzuți atât senzori mecanici cât și traductori de curent. Astfel modulul TIM va dispune de trei canale: unul de ieșire, pentru actuator și două de intrare pentru monitorizare, cu arhitectura expusă în figura 4.12.

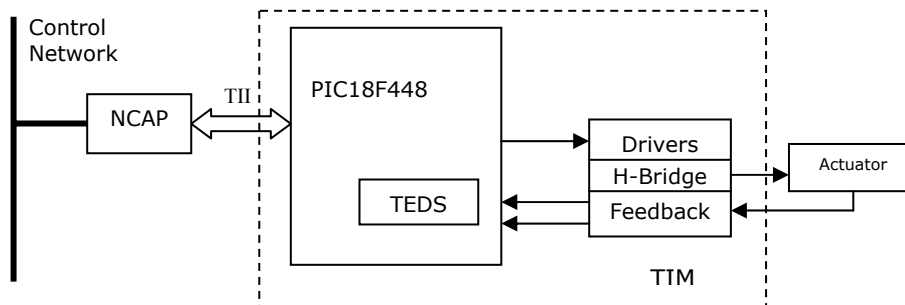


Fig. 4.12. Structura TIM cu 3 canale de date

Structura hardware a modului TIM trebuie să fie astfel concepută încât să permită implementarea tuturor funcțiilor de bază prevăzute de către specificațiile IEEE 1451.2 referitoare la interfața TII pe 10 linii, blocul TEDS și modul de comunicare cu NCAP. Analizând aceste considerente, am selectat microcontrolerul PIC18F448 care dispune de suficiente linii de port pentru implementarea TII și a celor trei canale de date, oferă un spațiu de 16Ko memorie program și 256 octeți memorie date EEPROM. Totodată, prezintă caracteristici avansate pentru comanda PWM a motoarelor de curent continuu materializate prin prezența unui modul ECCP (Enhanced Capture/Compare) cu patru ieșiri PWM [62].

Prezența interfeței SPI, înglobată în structura internă a microcontrolerului PIC18F448 permite implementarea directă a interfeței TII către NCAP. Controlul transductorului se realizează prin patru linii de port dedicate (ieșirile PWM ale modului ECCP) care prin intermediul unui bloc driver (MOSFET Driver) comandă puntea H (MOSFET H Bridge). Aceste linii formează primul canal de date al TIM (de ieșire). Pentru urmărirea poziției actuatorului și a gradului de încărcare s-au implementat în modulul TIM două canale de date, de intrare, denumite „analog feedback” (răspunsul reacție analogic) – pentru măsurarea curentului consumat de către actuator, respectiv „digital feedback” (reacția numerică) pentru stabilirea poziției cu ajutorul unor senzori mecanici. În figura 4.13 sunt prezentate liniile de port dedicate conexiunilor electrice aferente modulului TIM implementat [62].

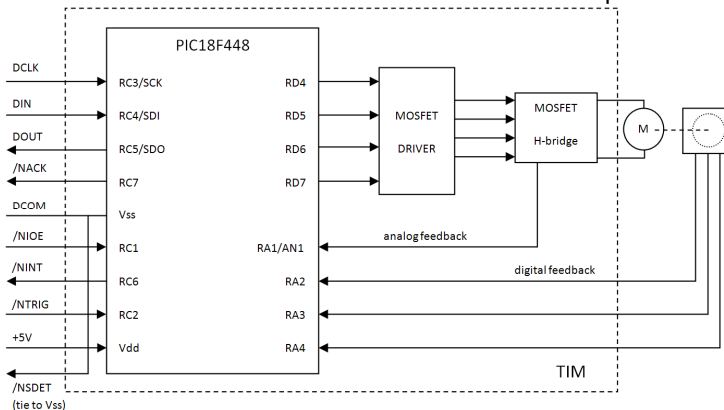


Fig. 4.13. Modulul TIM implementat cu PIC18F448 conform IEEE 1451.2

Blocul TEDS (IEEE 1451.2) va conține rubricile „Channel TEDS” – câte unul pentru fiecare canal și „Meta-TEDS” – care descrie sistemul global. Valorile câmpurilor aferente TEDS se pot înscrie în spațiul alocat din memoria de date EEPROM al microcontrolerului prin funcții speciale, scrise sub formă de cod program în limbaj C adecvat familiei PIC18F4xx. Tabelul 4.3 reunește câmpurile Meta-TEDS incluse în blocul TEDS implementat în memoria de date a microcontrolerului PIC18F448.

Zonă	Câmp	Semnificație	Bytes	Tip	Valoare
0x00	1	Lungime Meta TEDS	4	U32L	42
	2	Număr diviziune standard IEEE 1451	1	U8E	2
	3	Număr versiune TEDS	2	U16E	1
	4	Cheie de extensie ulterioară	1	U8E	0
	5	Cheie CHANNEL_ZERO	1	U8E	0
	6	Cheie TEDS aplicație utilizator	1	U8E	0
	7	Număr canale implementate	1	U8C	3
	8	Lungime model canal de date	1	U8C	2
	9	Repetiții date pentru canal	2	U16C	1
	10	Durată maximă actualizare canal	4	F32	2e-5
	11	Durată maximă inițializare scriere canal	4	F32	0
	12	Durată maximă inițializare citire canal	4	F32	8e-5
	13	Timp de răspuns intrare/ieșire	4	F32	5e-4
	14	Durată perioadă de eșantionare canal	4	F32	2e-4
	15	Rata de transfer maximă	4	U32C	2e+5
0x29	16	Lungime blocuri de date canale grupate	2	U16L	0
	17	Sumă de control (checksum)	2	U16C	0 (inițial)

Tabelul 4.3. Câmpurile Meta-TEDS

Tabelul 4.4 conține câmpurile implementate din spațiul Channel TEDS pentru canalul 1, ce descrie actuatorul.

Zonă	Câmp	Semnificație	Bytes	Tip	Valoare	
0x2A	1	Lungime Channel TEDS	4	U32L	71	
	2	Cheie de calibrare	1	U8E	0	
	3	Cheie extensie TEDS	1	U8E	0	
	4	Cheie extensie TEDS industrială	1	U8E	0	
	5	Cheie tip canal	1	U8E	ACTUATOR	
	6	Unități fizice	10	UNITS	SI	
	7	Limita inferioară	4	F32	0	
	8	Limita superioară	4	F32	255	
	9	Incertitudine caz cel mai defavorabil	4	F32	0	
	10	Cheie autotest	1	U8E	0	
	Channel TEDS 1	11	Model date canal	1	U8E	0
		12	Lungime model canal de date	1	U8C	1
		13	Biți semnificativi model	2	U16C	4
		14	Repetiții date pentru canal	2	U16C	0
		15	Durată maximă actualizare canal	4	F32	25e-4
		16	Durată maximă inițializare scriere canal	4	F32	1e-3
		17	Durată maximă inițializare citire canal	4	F32	1e-3
		18	Durată perioadă de eșantionare canal	4	F32	5e-3
		19	Durată inițializare canal	4	F32	5e-3
		20	Timp mort canal	4	F32	0.5
	0x70	21	Corecție temporală	4	F32	1
		22	Acuratețe declanșare (trigger)	4	F32	0.5
23		Sumă de control (checksum)	2	U16C	0 (inițial)	

Tabelul 4.4. Channel TEDS pentru canalul 1

Pentru fiecare senzor atașat (cele două canale Ch.2 și Ch.3) se definește câte un Channel TEDS specific, prezentat în tabelul 4.5 respectiv tabelul 4.6.

Zonă	Câmp	Semnificație	Bytes	Tip	Valoare	
0x71	1	Lungime Channel TEDS	4	U32L	71	
	2	Cheie de calibrare	1	U8E	0	
	3	Cheie extensie TEDS	1	U8E	0	
	4	Cheie extensie TEDS industrială	1	U8E	0	
	5	Cheie tip canal	1	U8E	SENSOR	
	6	Unități fizice	10	UNITS	SI	
	7	Limita inferioară	4	F32	1	
	8	Limita superioară	4	F32	1023	
	9	Incertitudine caz cel mai defavorabil	4	F32	10	
	10	Cheie autotest	1	U8E	0	
	Channel TEDS 2	11	Model date canal	1	U8E	0
		12	Lungime model canal de date	1	U8C	2
		13	Biți semnificativi model	2	U16C	10
		14	Repetiții date pentru canal	2	U16C	0
		15	Durată maximă actualizare canal	4	F32	25e-4
		16	Durată maximă inițializare scriere canal	4	F32	1e-3
		17	Durată maximă inițializare citire canal	4	F32	1e-3
		18	Durată perioadă de eșantionare canal	4	F32	5e-3
		19	Durată inițializare canal	4	F32	5e-3
		20	Timp mort canal	4	F32	0.5
	0xB7	21	Corecție temporală	4	F32	1
		22	Acuratețe declanșare (trigger)	4	F32	0.5
23		Sumă de control (checksum)	2	U16C	0 (inițial)	

Tabelul 4.5. Channel TEDS pentru canalul 2

Zonă	Câmp	Semnificație	Bytes	Tip	Valoare	
0xB8	1	Lungime Channel TEDS	4	U32L	71	
	2	Cheie de calibrare	1	U8E	0	
	3	Cheie extensie TEDS	1	U8E	0	
	4	Cheie extensie TEDS industrială	1	U8E	0	
	5	Cheie tip canal	1	U8E	SENSOR	
	6	Unități fizice	10	UNITS	SI	
	7	Limita inferioară	4	F32	0	
	8	Limita superioară	4	F32	7	
	9	Incertitudine caz cel mai defavorabil	4	F32	0	
	10	Cheie autotest	1	U8E	0	
	Channel TEDS 3	11	Model date canal	1	U8E	0
		12	Lungime model canal de date	1	U8C	1
		13	Biți semnificativi model	2	U16C	3
		14	Repetiții date pentru canal	2	U16C	0
		15	Durată maximă actualizare canal	4	F32	25e-4
		16	Durată maximă inițializare scriere canal	4	F32	1e-3
		17	Durată maximă inițializare citire canal	4	F32	1e-3
		18	Durată perioadă de eșantionare canal	4	F32	5e-3
		19	Durată inițializare canal	4	F32	5e-3
		20	Timp mort canal	4	F32	0.5
	0xFE	21	Corecție temporală	4	F32	1
		22	Acuratețe declanșare (trigger)	4	F32	0.5
23		Sumă de control (checksum)	2	U16C	0 (inițial)	

Tabelul 4.6. Channel TEDS pentru canalul 3

Fiecare secțiune din TEDS este înglobată în spațiul memoriei de date EEPROM a microcontrolerului PIC18F448 conform adreselor specificate în rubrica „Zona” din cele patru tabele indicate mai sus. Acestea sunt plasate ierarhic și se supun structurii generale TEDS, prezentate în figura 4.5. Astfel, în spațiul de adresare 0x00-0x29 se regăsesc cei 42 bytes ai Meta-TEDS, în spațiul 0x2A-0x70 cei 71 de bytes ai Channel 1 TEDS, în spațiul 0x71-0xB7 cei 71 de bytes ai Channel 2 TEDS și în spațiul 0xB8-0xFE cei 71 de bytes aferenți Channel 3 TEDS.

Tipurile de date la care se apelează în descrierea câmpurilor sunt specificate în standardul IEEE 1451 [57], [61] și pentru a putea fi utilizate în rutine și funcții scrise în limbaj C, acestea trebuie predefinite în fișiere „header” (antet). Figura 4.14 prezintă conținutul fișierului antet „IEEE1451_data.h” care reprezintă codul sursă, în limbaj MikroC (pentru PIC18F448), aferent definirii tipurilor de date conform specificației IEEE 1451.2.

Câmpul „Cheie tip canal” joacă un rol important în determinarea tipului de dispozitiv atașat modulului TIM și care în cazul nostru este atât senzor cât și actuator. Valorile „SENSOR” și „ACTUATOR” sunt definite în cadrul fișierului antet „IEEE1451_data.h” și li se atribuie valoarea 0 respectiv 1, așa cum indică standardul IEEE 1451.2.

Pe baza acestor tipuri de date am definit în continuare structura unor variabile compuse care alcătuiesc suportul virtual pentru câmpurile din componența blocului TEDS, pentru fiecare secțiune în parte. Codul sursă aferent este inclus în fișierul „IEEE1451_TEDS.h”, prezentat în figura 4.15.

Urmează alcătuirea unor biblioteci de funcții prin intermediul cărora se realizează înscrierea valorilor efective în spațiul de memorie de date EEPROM alocat TEDS. Structura TEDS a fost optimizată, păstrându-se informațiile mandatorii și excluzând pe cele redundante sau neutilizate în aplicație [62]. Blocul de date

obținut a fost redus astfel la o lungime de 254 bytes care se încadrează în cei 256 bytes de memorie de date EEPROM ai microcontrolerului PIC18F448.

```
// -----
// Definitii tipuri de date IEEE 1451.2
// -----

// Intreg fara semn pe 8 biti

typedef unsigned char U8C; // ... contorizare
typedef unsigned char U8E; // ... enumerare
typedef unsigned char U8L; // ... lungime camp

// Intreg fara semn pe 16 biti

typedef unsigned int U16C;
typedef unsigned int U16E;
typedef unsigned int U16L;

// Intreg fara semn pe 32 biti

typedef unsigned long U32C;
typedef unsigned long U32L;

// Numar real, precizie simpla, 32 biti

typedef float F32;

// Sir de caractere

typedef unsigned char STRING;

// Unitati fizice, 10 bytes

#define U_exponents 9 // numarul de exponenti

// tipuri de unitati de masura din matrice

enum IEEE1451_units {RADIANS,STERADIANS,METERS,KILOGRAMS,SECONDS,AMPERES,
                    KELVINS,MOLES,CANDELAS};

// Tipuri de enumeratii din Enumeration Field:

enum IEEE1451_enumeration {PRODUCT,U_OVER_U,LOG10_U,LOG10_U_OVER_U,
                           DIGITAL_DATA,ARBITRARY};

typedef struct {
    U8E EnumField;
    U8C UExponents[U_exponents];
} UNITS ;

// Tipuri de transductori

enum IEEE1451_trans {SENSOR,ACTUATOR,EVENT_SEQUENCE_SENSOR,DATA_SEQUENCE_SENSOR,
                    GENERAL_TRANSDUCER,BUFFERED_SENSOR,BUFFERED_DATA_SEQUENCE_SENSOR,
                    GLOBAL};
```

Fig.4.14. Definirea tipurilor de date TEDS

Metoda de generare software a datelor din TEDS, propusă în lucrarea de față vine în completarea implementării hardware realizate în [62] și reprezintă o abordare nouă, originală care permite cu ușurință adaptarea la structuri TEDS specifice altor aplicații, prin completarea fișierelor antet „IEEE1451_data.h” respectiv „IEEE1451_TEDS.h”.

Pentru a prelua valorile câmpurilor din tabelele 4.3 – 4.6, am creat structurile „MetaTeds_s” și „ChTeds_s” care se bazează pe tipurile de date definite în codul prezentat în figura 4.14.

Declarăm în continuare variabilele „Meta”, „Ch1Teds”, „Ch2Teds” și „Ch3Teds” de tipul „MetaTeds_s” respectiv „ChTeds_s” și le atribuim valorile indicate

din tabelele prezentate mai sus în cadrul unor funcții scrise tot în limbajul C, reunite în fișierul „*FuncțiiTEDS.h*”, prezentat în figura 4.16.

```

// -----
// Definitii structuri de variabile IEEE 1451.2
// -----

#include "IEEE1451_data.h"

// MetaTEDS

typedef struct {
    U32L  MLength;           // campul 1
    U8E   WGNNumber;        // campul 2
    U16E  VNNumber;         // campul 3
    U8E   FEKey;            // campul 4
    U8E   ChZeroKey;        // campul 5
    U8E   UASTedsKey;       // campul 6
    U8C   NumOfCh;          // campul 7
    U8C   DModLength        // campul 8
    U16C  DRep;             // campul 9
    F32   WCChUpdateTime;   // campul 10
    F32   WCCWriteSetupTime; // campul 11
    F32   WCCReadSetupTime; // campul 12
    F32   IOResponseTime;   // campul 13
    F32   WCChSampling;     // campul 14
    U32C  MaxDRate;         // campul 15
    U16L  GrSBLength;       // campul 16
    U16C  Checksum;         // campul 17
} MetaTeds_s;

// Channel TEDS

typedef struct {
    U32L  ChLength;         // campul 1
    U8E   CalKey;           // campul 2
    U8E   TExtKey;          // campul 3
    U8E   IExtKey;          // campul 4
    U8E   ChType;           // campul 5
    UNITS PHYUnits;        // campul 6
    F32   LowRangeLimit;    // campul 7
    F32   HiRangeLimit;     // campul 8
    F32   Uncertainty;      // campul 9
    U8E   SelfTestKey;      // campul 10
    U8E   ChDataModel;      // campul 11
    U8C   ChDataModelLength; // campul 12
    U16C  ModelSigBits;     // campul 13
    U16C  DataRep;          // campul 14
    F32   ChUpdateTime;     // campul 15
    F32   ChWriteSetupTime; // campul 16
    F32   ChReadSetupTime;  // campul 17
    F32   ChSamplingPeriod; // campul 18
    F32   ChWarmupPeriod;   // campul 19
    F32   ChHoldOffTime;    // campul 20
    F32   ChTimingCorr;     // campul 21
    F32   ChTriggerAcc;     // campul 22
    U16C  Checksum;         // campul 23
} ChTeds_s;

// Unitati Sistemul International

UNITS S;

```

Fig.4.15. Definirea structurilor de variabile TEDS

Fișierul „*FuncțiiTEDS.h*” a fost conceput astfel încât să poată fi utilizat cu ușurință și în aplicații care necesită un bloc TEDS cu mai multe canale, prin completarea codului sursă cu funcții suplimentare, pe baza modelului inițial. Pentru structura TEDS curentă am definit funcțiile „*MetaTEDS_init*”, „*Ch1TEDS_init*”, „*Ch2TEDS_init*” și „*Ch3TEDS_init*” care pot fi apelate ulterior în rutina principală a codului program.

Valorile variabilelor definite prin aceste funcții sunt înscrise apoi în memoria de date EEPROM a microcontrolerului PIC18F448.

```

#include "IEEE1451_TEDS.h"
void MetaTEDS_init(void)
{
    // Definire continut Meta TEDS
    MetaTeds_s Meta = {
        (U32L)sizeof(MetaTeds_s)- sizeof(U32L), // Lungime Meta TEDS
        2, // Numar diviziune IEEE 1451
        1, // Numar versiune TEDS
        0, // Cheie de extensie ulterioara
        0, // Cheie CHANNEL_ZERO
        0, // Cheie TEDS aplicatie utilizator
        3, // Numar canale implementate
        2, // Lungime model canal de date
        1, // Repetitii date pentru canal
        0.0025, // Durata maxima actualizare canal
        0.001, // Durata maxima initializare scriere canal
        0.001, // Durata maxima initializare citire canal
        0.5, // Timp de raspuns intrare/iesire
        0.0005, // Durata perioada de esantionare canal
        2000000, // Rata de transfer maxima
        0, // Lungime blocuri de date canale grupate
        0 // Suma de control
    };
}

void Ch1TEDS_init(void)
{
    // Canal 1: Actuator MCC
    ChTeds_s Ch1TEDS = {
        (U32L)sizeof(ChTeds_s)- sizeof(U32L), // Lungime Channel TEDS
        0, // Cheie de calibrare
        0, // Cheie extensie TEDS
        0, // Cheie extensie TEDS industrialia
        1, // Cheie tip canal
        // Unitati fizice
        0x04,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
        0, // Limita inferioara
        255, // Limita superioara
        0, // Incertitudine caz cel mai defavorabil
        0, // Cheie autotest
        0, // Model date canal
        1, // Lungime model canal de date
        4, // Biti semnificativi model
        0, // Repetitii date pentru canal
        0.0025, // Durata maxima actualizare canal
        0.001, // Durata maxima initializare scriere canal
        0.001, // Durata maxima initializare citire canal
        0.005, // Durata perioada de esantionare canal
        0.005, // Durata initializare canal
        0.5, // Timp mort canal
        1.0, // Corectie temporala
        0.5, // Acuratete declansare (trigger)
        0 // Suma de control
    };
}

void Ch2TEDS_init(void)
{
    // Canal 2: Current Feedback Sensor
    ChTeds_s Ch2TEDS = {
        (U32L)sizeof(ChTeds_s)- sizeof(U32L), // Lungime Channel TEDS
        0, // Cheie de calibrare
        0, // Cheie extensie TEDS
        0, // Cheie extensie TEDS industrialia
        0, // Cheie tip canal
        // Unitati fizice
        0x04,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
        1, // Limita inferioara
        1023, // Limita superioara
        10, // Incertitudine caz cel mai defavorabil
        0, // Cheie autotest
        0, // Model date canal
        2, // Lungime model canal de date
        10, // Biti semnificativi model
        0, // Repetitii date pentru canal
        0.0025, // Durata maxima actualizare canal
        0.001, // Durata maxima initializare scriere canal
        0.001, // Durata maxima initializare citire canal
        0.005, // Durata perioada de esantionare canal
        0.005, // Durata initializare canal
        0.5, // Timp mort canal
        1.0, // Corectie temporala
        0.5, // Acuratete declansare (trigger)
        0 // Suma de control
    };
}

void Ch3TEDS_init(void)
{
    // Canal 3: Rotary Encoder
    ChTeds_s Ch3TEDS = {
        (U32L)sizeof(ChTeds_s)- sizeof(U32L), // Lungime Channel TEDS
        0, // Cheie de calibrare
        0, // Cheie extensie TEDS
        0, // Cheie extensie TEDS industrialia
        0, // Cheie tip canal
        // Unitati fizice
        0x04,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
        0, // Limita inferioara
        7, // Limita superioara
        0, // Incertitudine caz cel mai defavorabil
        0, // Cheie autotest
        0, // Model date canal
        1, // Lungime model canal de date
        3, // Biti semnificativi model
        0, // Repetitii date pentru canal
        0.0025, // Durata maxima actualizare canal
        0.001, // Durata maxima initializare scriere canal
        0.001, // Durata maxima initializare citire canal
        0.005, // Durata perioada de esantionare canal
        0.005, // Durata initializare canal
        0.5, // Timp mort canal
        1.0, // Corectie temporala
        0.5, // Acuratete declansare (trigger)
        0 // Suma de control
    };
}

```

Fig.4.16. Funcții pentru definirea valorilor aferente TEDS

Elaborarea acestor rutine a fost necesară pentru a crea baza software pentru implementarea funcțiilor aferente blocului TIM intern care va intra în componența subsistemului inteligent propus în lucrare. Testarea în prealabil a acestora a fost efectuată prin sistemul bazat pe microcontrolerul PIC18F4550 [62].

4.3. Protocolul de comunicație HART

Highway Addressable Remote Transducer (HART), introdus pentru prima dată în anul 1989, furnizează o modalitate de implementare a comunicației digitale între dispozitive de control ale proceselor tehnologice bazate pe microprocesoare dar cu transmiterea analogică a informației prin intermediul unei magistrale care funcționează conform standardului cunoscut sub denumirea „tehnica 4 la 20 mA” [63]. Reprezintă prima metodă de comunicație digitală, bidirecțională, care nu perturbă semnalul analogic coexistent pe aceeași linie de transmisie. Adoptat în decursul anilor la nivel mondial, la ora actuală există peste 5 milioane de dispozitive HART instalate în diverse sisteme de producție începând cu industria petrolului până la cea alimentară [63]. Astfel există un interes justificat în a adopta o soluție viabilă de a păstra dispozitivele de teren HART deja instalate în procesul de tranziție la sisteme de comunicație complet digitale și care se supun standardelor actuale cum ar fi IEC 61784 (vezi Capitolul 2) și IEEE 1451. Tendința de utilizare a rețelelor de tip Ethernet industrial, compatibile Ethernet clasic, este evidentă și aduce o multitudine de avantaje, descrise pe larg în Capitolul 2.

În lucrarea de față am urmărit conceperea unei „punți de legătură” între dispozitivele HART clasice și cerințele standardului IEEE 1451 referitoare la interconectarea transductorilor inteligenți. Originalitatea constă în elaborarea unui subsistem inteligent, compatibil IEEE 1451, care va permite atașarea directă a perifericelor HART la acesta și asigurarea accesului la o rețea de teren tip Ethernet industrial.

Considerăm necesară parcurgerea sumară a anumitor caracteristici ale protocolului HART referitoare atât la formatul datelor vehiculate cât și la anumite modalități de conectare la mediul fizic (modemul HART) pentru utilizarea acestora pe parcursul conceperii subsistemului inteligent.

4.3.1. Mediul fizic de comunicație HART

Principiul de funcționare al comunicației HART se bazează pe infrastructura clasică tip „bucă de curent” utilizată în transmiterea informației de către dispozitivele de conducere a proceselor tehnologice la unitățile de control de tip PLC (Programmable Logic Controller) [63]. Pornind de la modul de conectare al perifericelor care utilizează tehnica analogică „4 la 20 mA”, protocolul HART presupune suprapunerea unui semnal modulat în frecvență, de tip FSK (Frequency Shift Keying), pe liniile de transmisie analogică. Rezultă necesitatea înglobării atât în controlere cât și în dispozitivele de proces (senzori sau actuatoare) a unui modem HART cu interfață adecvată liniilor de transmisie analogice [63]. Figura 4.17 prezintă structura liniei de comunicație adecvată protocolului HART.

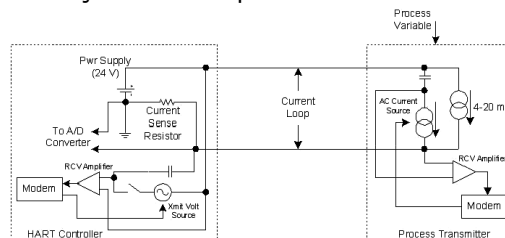


Fig.4.17. Linie de comunicație HART clasică [63]

Interfața către liniile de transmisie include în ambele părți câte un modem și un amplificator (RCV Amplifier) cu o impedanță de intrare relativ ridicată pentru a nu încărca bucla de curent analogică. Dispozitivul de proces (Process Transmitter) conține în structura sa o sursă de curent alternativ (AC Current Source) iar controlerul (HART Controller) o sursă de tensiune alternativă (Xmit Volt Source). În mod normal, comutatorul în serie cu sursa de tensiune este în poziția „deschis”.

Pentru a trimite un mesaj HART, dispozitivul de proces activează sursa de curent alternativ, cuplată capacitiv, suprapunând pe linia de transmisie un curent de valoare aproximativă de 1 mA, de frecvență ridicată. Rezistorul de sesizare din controler (Current Sense Resistor) transformă această variație într-o tensiune care apare între cele două terminale ale liniei. Această tensiune este sesizată de către amplificatorul de recepție (RCV Amplifier) și furnizată blocului demodulator al controlerului HART. În practică, cele două surse de curent din dispozitivul de proces (AC current Source și „4 – 20 mA”) sunt implementate sub forma unui singur regulator de curent iar semnalele analogice sunt mixate cu cele digitale HART înaintea regulatorului [63].

Controlerul HART poate trimite un mesaj spre dispozitivul de proces prin închiderea comutatorului de transmisie (vezi figura 4.17). Acest lucru conduce la conectarea sursei de tensiune (Xmit Volt Source) la liniile de transmisie și suprapunerea unei tensiuni alternative de frecvență ridicată și amplitudine de 500 mV vârf la vârf care este sesizată de către amplificatorul de recepție al dispozitivului de proces (Process Transmitter) și furnizată mai departe la intrarea demodulatorului (modem), așa cum se poate deduce din figura 4.17.

Forma tensiunii suprapuse pe liniile de transmisie se poate observa în figura 4.18 [63].

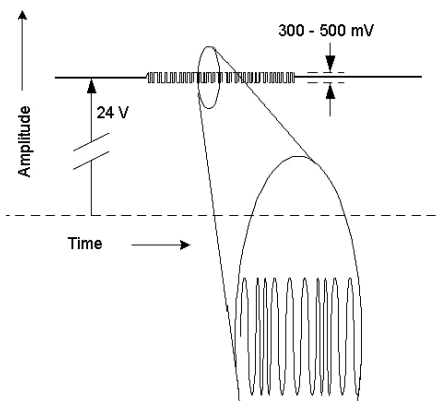


Fig.4.18. Salvă de semnale HART pe liniile de transmisie analogice [63]

Amplitudinea tensiunii suprapuse este relativ redusă față de valoarea tensiunii de alimentare continuă, de 24 V și va avea o gamă de variație cuprinsă aproximativ între 300 și 500 mV care depinde de mai mulți parametri ai dispozitivului receptor, cum ar fi valoarea rezistenței de sesizare a curentului, sarcina capacitivă și anumite pierderi datorate altor periferice de pe rețea [63].

Din punctul de vedere al formei curentului suprapus, putem observa un domeniu de variație de 1 mA, vârf la vârf, care de fapt modulează semnalul analogic existent pe liniile de transmisie care are o frecvență de variație redusă, de aproximativ 10 Hz și poate lua valori între 4 și 20 mA.

Figura 4.19 prezintă forma semnalului HART suprapus peste transmisia analogică în curent (tehnica „4 la 20 mA”) [63].

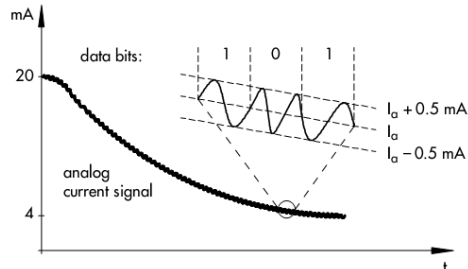


Fig.4.19. Secvență HART extrasă din semnalul de curent 4 la 20 mA [63]

Pentru codarea biților de date este utilizată modulația FSK, bazată pe standardul de comunicație Bell 202, evidențiată în figura 4.19 [63]. Nivelului logic „0” i s-a atribuit un semnal de frecvență 2200 Hz iar nivelului „1” o valoare a frecvenței de 1200 Hz. Semnalul HART este uzual extras cu ajutorul unui filtru trece – sus având frecvența de tăiere în domeniul 400 la 800 Hz.

4.3.2. Formatul pachetelor de date HART

Protocolul HART operează conform principiului Master – Slave și orice comunicație poate fi inițiată de către Master care poate fi atât o stație de control cât și un dispozitiv periferic. Sunt acceptați doi Masteri, cel primar fiind sistemul de control (un PC) și cel secundar un terminal mobil dedicat sau un laptop.

Dispozitivele de teren HART (HART Field Device) sunt considerate Slave și nu pot transmite date decât în momentul în care acestea sunt interogate de către un Master.

Serviciile de comunicație HART furnizează atât comenzi standard (schimb de date Master/Slave) cât și comenzi globale (recepționate de către toate dispozitivele) [63]. O tranzacție simplă constă în emiterea unei „telegramme” Master urmată în mod direct de o „telegrammă” de răspuns din partea unui Slave, conform reprezentării din figura 4.20 [63].

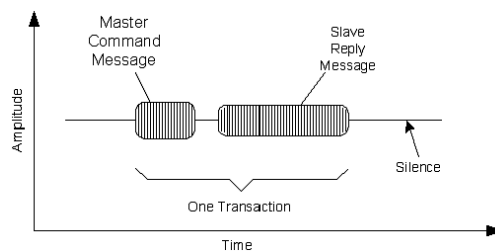


Fig.4.20. Tranzacție HART standard [63]

În urma stabilirii unei conexiuni, prin intermediul apelării comenzii 11 HART, se poate emite un mesaj global, către toate dispozitivele, pentru verificarea de exemplu a configurației sistemului [63].

Anumite dispozitive HART suportă modul opțional de comunicație în salvă, prin care un singur dispozitiv de teren trimite ciclic telegrame, separate de pauze cu o durată de 75 ms, către controlerul Master. Prin acest mod se poate atinge

transmiterea până la patru telegrame pe secundă față de doar două în modul standard [64].

Structura unei telegrame HART este prezentată în figura 4.21. Fiecare cuvânt este transmis sub forma unui caracter UART, pe 11 biți, prevăzut cu un bit de start, de paritate și de stop [64].

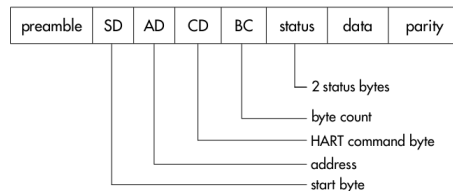


Fig.4.20. Structura unei telegrame HART [64]

Preambulul (preamble) poate conține trei sau mai multe caractere hexazecimale de valoare FF cu rolul de a sincroniza semnalele participanților.

Octetul de start (SD) indică participantul care începe o transmisie de date (poate fi Master sau Slave) și tipul de adresare utilizat (formatul scurt sau cel lung).

Câmpul de adresă al formatului scurt conține un singur octet cu rolul de a face distincția între cei doi masteri precum și pentru stabilirea modului de transmisie în salve de date. Dispozitivele de teren sunt adresate pe patru biți, rezultând spațiul de adrese 0 – 15. În cazul formatului lung, se utilizează cinci octeți în care avem 36 biți de adresă rezervați pentru dispozitivele de teren, extinzând astfel numărul maxim de participanți. Cele două formate de adresă sunt indicate în figura 4.21.

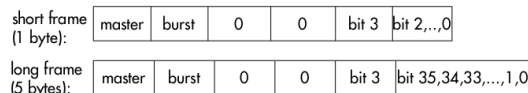


Fig.4.21. Formate de adresă HART [64]

Octetul de comandă (CD) înglobează trei categorii de comenzi HART: universale, de uz comun și specifice dispozitivelor de teren [63], [64]. Semnificația acestora va fi detaliată pe parcursul implementării comunicației HART în subsistemul inteligent conceput de către autor.

Câmpul BC indică lungimea mesajului, care este variabilă deoarece o telegramă HART poate conține între 0 și 25 octeți de date. Contorul BC este unicul identificator pe baza căruia se poate decoda conținutul mesajului și calcula suma de control.

Cei doi octeți de stare (status) sunt incluși doar în mesajele de răspuns provenite de la dispozitivele Slave și conțin informație codată pe mai mulți biți care au valoarea 0 în cazul în care mesajul a fost trimis cu succes și dispozitivul se află în stare operațională.

Biții efectivi de date pot reprezenta numere întregi fără semn, reale sau șiruri de caractere codate ASCII. Formatul datelor este indicat de către tipul comenzii HART, nu toate comenzile necesită și câmp de date.

Suma de control reprezentată prin biții „parity” conține informația de paritate referitoare la toți octeții din componența telegramei HART.

Timpul necesar transmiterii unei telegrame HART este corelat cu rata de transfer cea mai scăzută (afărentă frecvenței de 1200 Hz) și cu numărul de biți din telegramă [64].

Lungimea unei telegrame depinde de numărul caracterelor din mesajul propriu-zis, care poate fi cuprins între 0 și 25. Pentru formatul de adresare scurt (figura 4.21), la un mesaj de 25 caractere îi revine o telegramă conținând în totalitate 35 de caractere, transmise sub formă de cuvinte UART, pe 11 biți fiecare [64]. Putem calcula pe baza acestor considerente, timpul necesar transmiterii unei astfel de telegrame HART. Procedura de calcul este prezentată în figura 4.22.

TELEGRAMA:	$35 \text{ caractere} * 11 \text{ biți} = 385 \text{ biți}$
DURATĂ 1 BIT:	$1/1200 = 0,83 \text{ ms}$
DURATĂ TRANZACȚIE:	$385 * 0,83 = 0,32 \text{ s}$
DURATĂ OCTET DATE:	$0,32/25 = 13 \text{ ms}$

Fig.4.22. Durata unei telegrame HART

În practică se consideră o durată de 500 ms pentru finalizarea unei tranzacții HART, rezultând un număr de două tranzacții pe secundă. Aceste valori indică o comunicație relativ lentă, nepotrivită controlului în timp real a proceselor critice din punct de vedere temporal. Astfel, apare necesitatea reducerii considerabile a timpului de tranzacție. Acest lucru va fi urmărit și pe parcursul lucrării, prin elaborarea adecvată a subsistemului inteligent compatibil IEEE 1451.

4.3.3. Clase de comenzi HART

Nivelul de aplicație al protocolului HART stabilește clasele de comenzi pe baza cărora operează rutinele de comunicație în dispozitivele HART. Comenzile predefinite permit unui Master HART să furnizeze instrucțiuni către un dispozitiv de teren și să transmită sau să recepționeze pachete de date [64]. Pentru asigurarea unei comunicații universale, comenzile HART sunt clasificate din punct de vedere funcțional în comenzi pentru dispozitive Master respectiv pentru dispozitive de teren, așa cum se prezintă în figura 4.23.

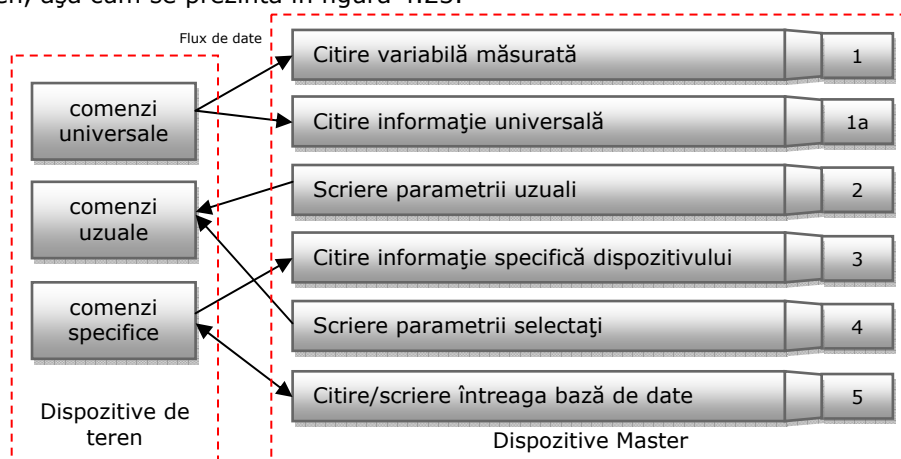


Fig.4.23. Clasificarea comenzilor HART

În funcție de etapa care urmează a fi executată în cadrul procesului de conducere, dispozitivul Master poate utiliza o comandă aparținând uneia din cele șase clase de conformanță expuse în figura 4.23 (incadrate la dispozitive Master). Fiecare astfel de clasă conține o submulțime de comenzi HART acoperind o gamă variată de sarcini de conducere și de mentenanță [63], [64].

Dispozitivele de teren interpretează și procesează doar acele comenzi HART care li se adresează în mod direct sau pe cele globale transmise către toate dispozitivele. Aceste comenzi aparțin uneia din cele trei clase aferente dispozitivelor de teren (figura 4.23). Ele se delimitează în:

- comenzi universale, interpretate și utilizate de către toate dispozitivele de teren care se conformează protocolului HART (parametrii hardware, software etc.);
- comenzi uzuale, adresate majorității dispozitivelor HART dar nu în mod obligatoriu tuturor participanților la magistrală (citirea unor variabile, stabilirea unor parametri etc.);
- comenzi specifice, prin care se asigură atât accesul la date ce descriu tipul sau structura unui dispozitiv cât și operațiile referitoare la inițializarea dispozitivelor de teren.

Clasele de comenzi prezentate vor fi implementate cu ajutorul unor rutine software în cadrul subsistemului inteligent, la nivelul blocului NCAP pentru a asigura compatibilitatea cu dispozitivele de teren HART conectate la acesta.

4.3.4. Soluții actuale de implementare HART pe rețele Ethernet

Tendința actuală, așa cum a fost expus și în Capitolul 2, este de a utiliza rețeaua Ethernet ca suport pentru comunicația între dispozitivele de conducere a proceselor tehnologice. Deoarece protocolul HART este bazat pe comunicația serială pe infrastructura 4 la 20 mA, dispozitivele HART nu pot beneficia în mod direct de tehnologia și arhitectura Ethernet, fără intercalarea unor „punți de legătură” (dispozitive gateway) [65].

Există câteva variante de arhitecturi tipice propuse în literatura de specialitate referitoare la conectarea dispozitivelor HART la o structură de rețea tip Ethernet, expuse pe scurt în continuare, împreună cu avantajele și dezavantajele ce le implică.

Vom considera ca punct de pornire cele două structuri HART clasice, cu un singur dispozitiv de teren respectiv multipunct cu mai multe dispozitive conectate la un Master, expuse în figura 4.24.a și 4.24.b [65].

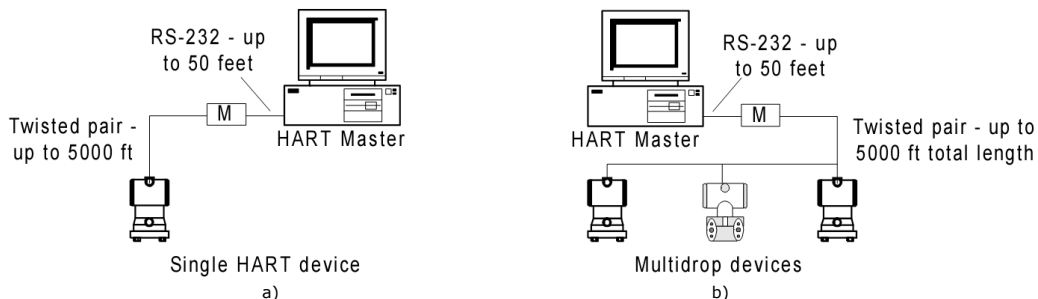


Fig.4.24. Arhitecturi HART clasice: a) dispozitiv unic; b) multipunct [65]

Soluțiile clasice pentru conectarea dispozitivelor HART, impun utilizarea unui modem HART (notat cu M în figura 4.24 a, b) care este conectat la o unitate centrală de comandă (HART Master) prin intermediul unei interfețe RS232. Modemul HART transformă semnalele și formatul de date RS232 în standardul de comunicație Bell 202 cu modulație FSK [63] și permite conectarea a maxim 15 dispozitive de teren. În cazul în care este necesară conectarea a mai mult de 15 dispozitive, se recurge la utilizarea unor multiplexoare HART care uzual comunică pe o interfață RS485 cu unitatea centrală, așa cum este prezentat în figura 4.25 [65].

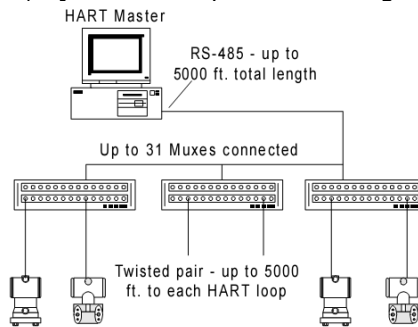


Fig.4.25. Multiplexoare HART [65]

Configurația din figura 4.25 permite conectarea a maxim 31 de multiplexoare HART, care înglobează și modemul necesar comunicării pe linia analogică 4 la 20 mA.

Structurile HART clasice, expuse mai sus, nu permit în mod direct conectarea la o rețea de tip Ethernet și nici modemurile existente nu au capacitatea de a comunica direct pe o astfel de rețea.

Masterul HART, conectat pe o rețea Ethernet, emite comenzi codate în cadre Ethernet, conform protocolului TCP/IP sau UDP. În funcție de tipul protocolului și al topologiei adoptate, la ora actuală putem distinge trei abordări a problematicii de implementare HART pe Ethernet, prezentate pe scurt în continuare.

Prima soluție, prezentată în figura 4.26, constă în utilizarea unui singur modem HART plasat la distanță și conectat la un convertor RS232 – Ethernet, de exemplu de tip Arcom Director [65]. Un convertor similar este intercalat și între unitatea centrală (HART Master) și rețeaua de teren Ethernet.

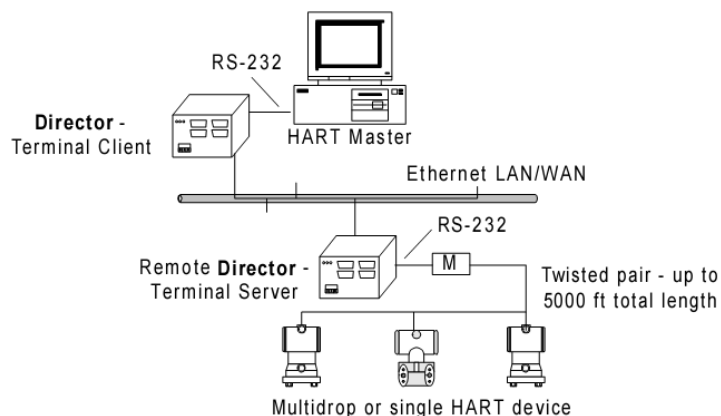


Fig.4.26. Structură HART pe Ethernet cu un singur modem [65]

Masterul HART, conectat printr-o interfață RS-232 la convertorul Director – Terminal Client (DTC), comunică prin intermediul protocolului TCP/IP pe rețeaua Ethernet cu Remote Director – Terminal Server (RDTS) care furnizează tot prin intermediul interfeței RS-232 semnale corespunzătoare modemului HART. Portul serial al RDTS este configurat pentru operații HART standard, cu rată de transmisie de 1200 baud, fără paritate, 8 biți de date și un bit de stop [65]. În esență, DTC și RDTS acționează ca un simplu gateway (punte de legătură) între rețeaua Ethernet și cea serială, vehiculând mesaje HART între dispozitivele conectate. Prin această configurație, masterul HART detectează virtual modemul ca fiind direct conectat pe portul serial, fără sesizarea celor două convertoare DTC și RDTS.

Pentru a mări numărul de dispozitive de teren ce se pot conecta în sistem, există a doua variantă de implementare HART pe Ethernet care se bazează pe principiul prezentat în figura 4.25. În acest caz, RDTS va dispune de un port serial RS-485 care permite conectarea a maxim 31 de multiplexoare HART, plasate la o distanță maximă de 1500 m. Figura 4.27 indică o structură HART pe Ethernet cu un singur RDTS și multiplexoare [65].

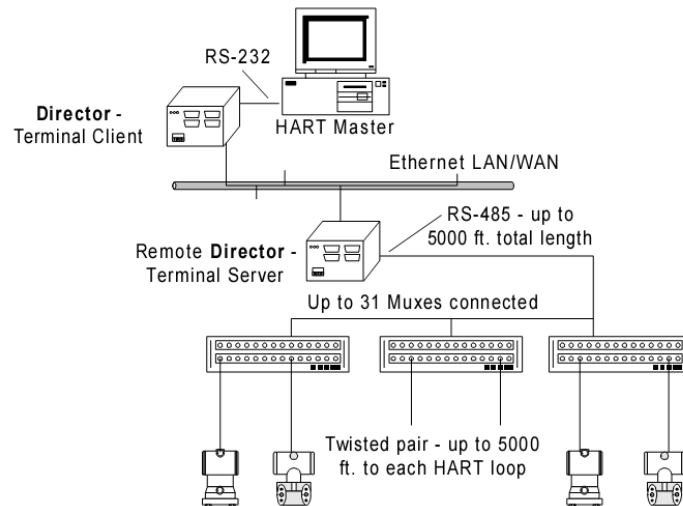


Fig.4.27. Structură HART pe Ethernet cu un singur RDTS și multiplexoare [65]

Configurația din figura 4.27 este adoptată în cazurile în care există o locație fizică de unde se prelevează valori ce descriu un anumit proces tehnologic, aflată la o distanță relativ ridicată față de unitatea centrală de comandă. Similar primei abordări, prezența rețelei Ethernet este transparentă dispozitivului HART Master datorită punții de legătură IP la serial asigurate de către convertoarele DTC și RDTS. Ratele de transfer a datelor în acest caz însă pot fi diferite la cele două convertoare pentru a se adapta caracteristicilor dispozitivului Master respectiv a multiplexoarelor HART conectate în sistem [65].

Primele două soluții de implementare au luat în considerare existența unui singur Master HART care poate comunica doar cu dispozitive aflate la o singură locație, prin intermediul unei perechi de convertoare DTC și RDTS. A treia variantă abordează situația în care dispozitivele de teren HART sunt amplasate în diferite puncte de lucru aflate la distanță unele față de altele și există o rețea de teren Ethernet comună acoperind întreg spațiul fizic alocat sistemului de conducere. Acest

lucru implică existența a mai multor convertoare dedicate fiecărei locații în parte, așa cum se indică în figura 4.28 [65].

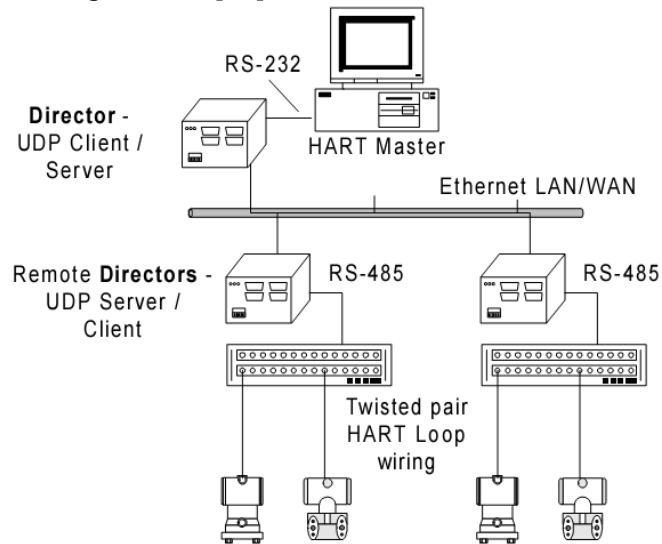


Fig.4.28. Structură HART pe Ethernet multi-locație [65]

Pentru a asigura comunicația cu multiple noduri din rețea (multiplexoare și dispozitive de teren HART), convertoarele utilizează în acest caz protocolul UDP (User Datagram Protocol) care este independent față de conexiune și suportă moduri de lucru cu adresare globală sau multiplă [65]. Converterul atașat la HART Master prin interfața RS-232, denumit Director-UDP Client/Server (DCS), preia mesajele de la acesta din urmă și le transmite către o adresă globală sau multiplă. Convertoarele Remote Directors-UDP Server/Client (RDSC) sunt configurate astfel încât să recepționeze aceste mesaje și să le translateze către propriile porturi seriale de tip RS-485. La aceste porturi pot fi conectate modemuri HART simple sau de tip multiplexor. Răspunsul de la dispozitivul de teren este preluat pe portul serial al RDSC și retransmis la o adresă unică, specifică IP, către DCS [64], [65].

Analizând soluțiile prezentate anterior, existente la ora actuală, deducem existența a numeroase dezavantaje ce rezultă în urma structurilor adoptate.

În primul rând există un lanț considerabil de conversii a formatului pachetelor de date precum și a semnalelor vehiculate, datorită diverselor tipuri de interfețe utilizate de către blocurile interconectate. În figura 4.28 distingem conversiile: RS-232 – Ethernet; Ethernet – RS-485; RS-485 – HART (prin modemul HART). Aceasta conduce în mod indirect la pierderea facilităților și a performanțelor oferite de rețeaua de tip Ethernet.

Convertoarele DTC, RDTs (figura 4.26, 4.27) și DCS, RDSC (figura 4.28) sunt de tip proprietar (de exemplu Arcocom Directors), nestandardizate și implică uzual costuri ridicate de instalare și sunt relativ dificil de configurat [63], [64], [65].

Instalarea celui de-al doilea Master HART va conduce în mod implicit la amplasarea unui convertor DTC sau DCS suplimentar.

Pe parcursul lucrării vom urmări optimizarea implementării protocolului HART pe suport Ethernet prin elaborarea unui subsistem inteligent care va îngloba concepte inovatoare, în conformitate cu standardul IEEE 1451 și va păstra funcționalitatea dispozitivelor de teren HART.

4.4. Subsistem inteligent de interfațare HART-Ethernet

Pentru a veni în sprijinul proiectanților și a utilizatorilor de sisteme de conducere a proceselor tehnologice, vom parcurge în continuare etapele conceperii unui subsistem inteligent (vezi paragraful 4.1) menit să implementeze o punte de legătură între mediul fizic Ethernet și dispozitivele de teren HART. În esență, structura sa va conține un bloc NCAP ce implementează funcțiile de bază prescrise de către standardul IEEE 1451, un modul de comunicare Ethernet și o interfață specializată care va îngloba modemul HART și care derivă din specificațiile IEEE 1451.4 pentru realizarea comunicației NCAP – dispozitive de teren HART.

Soluția propusă urmărește implementarea într-un singur circuit integrat a modulelor NCAP/TIM/modem HART/ pentru a reduce costurile de fabricație și de a îmbunătăți performanțele sistemului din punct de vedere al vitezei de procesare a informației și al ratei de transfer a datelor.

Modulul TIM, va avea anumite blocuri înglobate în NCAP și va prezenta o structură adaptată cerințelor dispozitivelor de teren HART.

Blocul TEDS, inexistent la dispozitivele de teren HART clasice, poate fi atașat sistemului prin implementarea sub formă de noduri de rețea suplimentare care se pot accesa de către NCAP prin interfața mixtă sau poate fi incorporat în structura NCAP sub forma unei baze de date actualizate în mod dinamic.

Pentru asigurarea comunicării pe linia analogică, subsistemul inteligent va cuprinde în structura sa un modem HART și circuite de intrare/ieșire specifice tehnicii de transmisie 4 la 20 mA.

Structura funcțională a subsistemului propus este prezentată în figura 4.29.

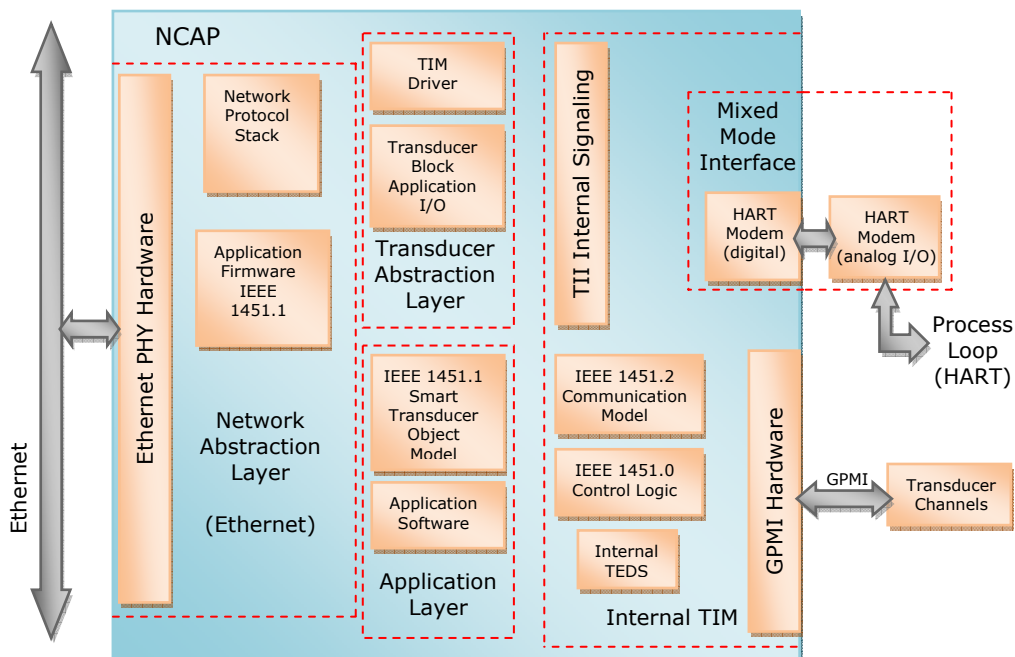


Fig.4.29. Structura funcțională a subsistemului inteligent

Blocul NCAP, principala componentă a subsistemului, este adaptat cerințelor specifice ale comunicației cu dispozitivele de teren HART, dar în același timp permite și interfațarea prin GPI (General Purpose Multichannel Interface) a transductoarelor compatibile IEEE 1451.2. Astfel, am propus pentru NCAP o structură multiport, așa cum se poate observa în figura 4.29.

Rolul NCAP în cadrul subsistemului inteligent este de a centraliza și închea toate funcțiile de control și comunicare, atât la nivelul rețelei Ethernet cât și pentru dispozitivele de teren conectate la acesta.

Comunicația pe rețea este percepută ca o operare cu porturi I/O, blocurile funcționale sunt implementate prin metode software iar caracteristicile transductoarelor sunt translatate în NCAP prin modulele de aplicație Transducer Block.

Implementarea modelului global NCAP se va baza pe conceptul expus în paragraful 4.2.2, cu respectarea claselor de obiecte ce alcătuiesc modelul informațional indicat în figura 4.8.

Obiectele ce descriu structura hardware de interfațare la Ethernet, biblioteca de rutine Ethernet și porturile de intrare/ieșire nu sunt definite în mod explicit prin standardul IEEE 1451. Acestea sunt specifice aplicației (în cazul nostru comunicația pe Ethernet) și ca urmare vor fi utilizate resursele proprii ale plăcii de dezvoltare cu FPGA, selectată pentru implementarea subsistemului. Network Abstraction Layer va asigura schimbul de date între mediul Ethernet și nivelul de aplicație Application Layer din structura subsistemului propus în figura 4.29.

Modulul NCAP conceput în lucrarea de față conține blocuri hardware și software care trebuie implementate astfel încât să își mențină compatibilitatea cu standardul IEEE 1451.1. Distingem în acest sens modelul de date (necesar descrierii diverselor tipuri de date utilizate), un model obiect (pentru clasele de obiecte impuse) și două modele de comunicație (aferele interfeței spre Ethernet respectiv transductor).

Structura de referință pentru NCAP [58], pe baza căreia s-a realizat implementarea blocurilor funcționale, este divizată pe trei niveluri: de rețea, de aplicație și de transductor, așa cum este indicat în figura 4.30.

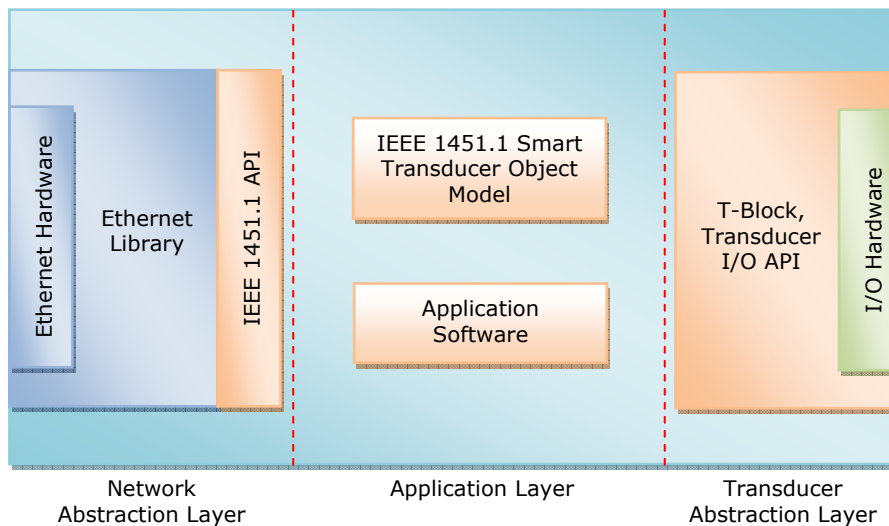


Fig.4.30. Structura de referință pentru NCAP

Nivelul pentru rețea (Network Abstraction Layer) reunește atât interfața hardware specifică Ethernet cât și modulele software ce implementează comunicația și transferul de date.

Nivelul de aplicație (Application Layer) conține doar blocuri software standard, impuse de specificațiile IEEE 1451 și rutine elaborate exclusiv pentru aplicația propusă.

Nivelul de abstractizare al transductorului (Transducer Abstraction Layer) cuprinde atât o structură hardware aferentă comunicării cu modulul TIM și dispozitivele de teren HART atașate cât și rutinele software din cadrul T-Block.

4.4.1. Implementarea modelului informațional NCAP

Pentru implementarea modelului informațional vom lua în considerare diferitele proprietăți ale obiectelor ce alcătuiesc NCAP [66], expuse în figura 4.31. Aceste proprietăți indică ierarhizarea claselor și relațiile de apartenență dintre ele și la ora actuală sunt descrise adecvat printr-un limbaj unificat de modelare UML (Unified Modeling Language) în literatura de specialitate [66], [67].

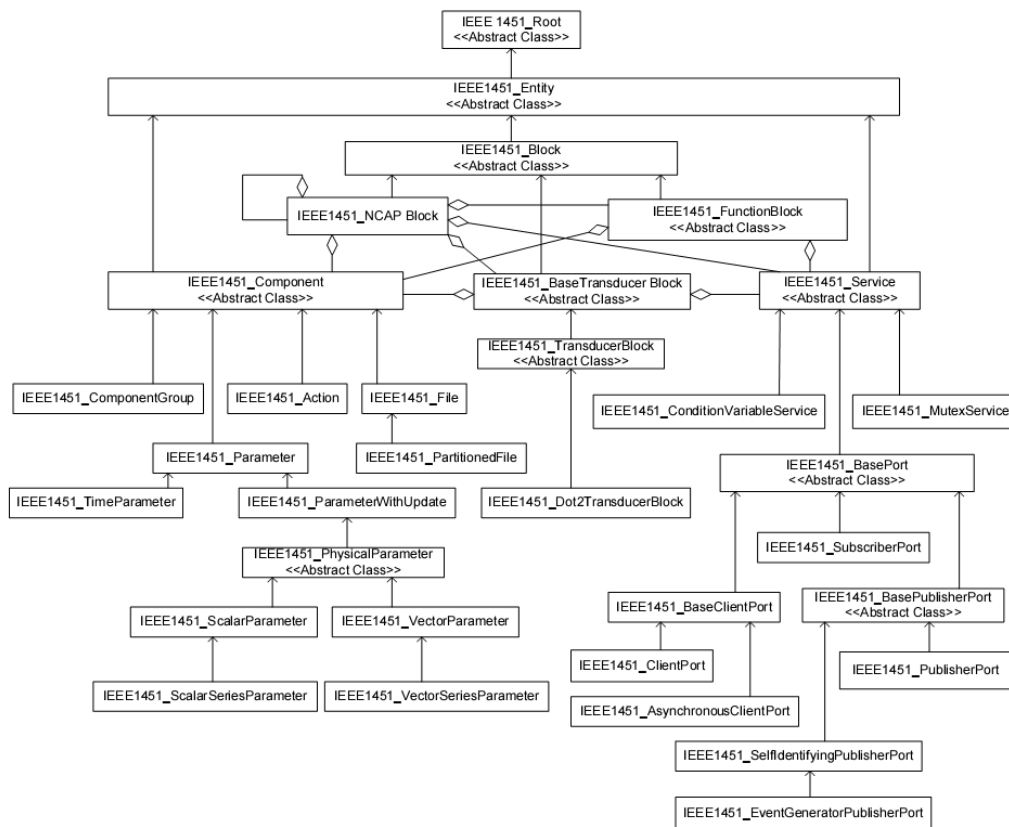


Fig.4.31. Obiectele din componența modelului informațional NCAP [67]

În figura 4.31 sunt sintetizate legăturile relaționale între obiecte prin marcarea acestora cu două tipuri de săgeți. Cele în formă de romb sunt utilizate pentru a evidenția relația de apartenență la blocul înspre care indică săgeata. De

exemplu, blocul NCAP deține Component Block, Function Block și așa mai departe [67].

Săgețile clasice indică relația de moștenire între clasele părinte și subclase. Astfel, IEEE1451_Entity este părintele clasei IEEE1451_Block care va moșteni funcționalitatea clasei superioare (părinte).

Vom descrie pe scurt semnificația fiecărui element reprezentat în figura 4.31 cu scopul de a crea o vedere de ansamblu asupra nivelului de complexitate a standardului IEEE 1451. Urmează apoi o optimizare a structurii existente prin identificarea și selectarea elementelor strict necesare atingerii funcționalității subsistemului elaborat și astfel degrevarea unității centrale de anumite operații redundante.

Distingem următoarele elemente:

- **IEEE1451 Root Class**, constituind punctul de plecare a ierarhizării claselor de obiecte definite prin standardul IEEE 1451;
- **IEEE1451 Entity**, care reprezintă rădăcina ierarhiei claselor tuturor obiectelor definite prin acest standard și care sunt vizibile pe rețeaua de comunicație;
- **IEEE1451 Block**, fiind rădăcina ierarhiei claselor tuturor obiectelor de tip Block;
- **IEEE1451 NCAP Block**, furnizând resursele și operațiile ce intervin într-un proces NCAP. Implementează gestionarea blocurilor, componentelor și serviciilor din interiorul NCAP (înregistrarea, de-registrarea, inițializarea și pornirea respectiv oprirea sistemului);
- **IEEE1451 Function Block**, rădăcina ierarhiei claselor tuturor obiectelor care abstractizează funcționalitatea aplicațiilor;
- **IEEE1451 Base Transducer Block**, rădăcina ierarhiei claselor tuturor obiectelor Transducer Block;
- **IEEE1451 Transducer Block**, constituind rădăcina ierarhiei de clase a tuturor obiectelor Transducer Block specificate prin standardul IEEE 1451.X;
- **IEEE1451 Component**, rădăcina ierarhiei de clase a tuturor obiectelor Component;
- **IEEE1451 Parameter**, utilizat pentru modelarea variabilelor vizibile în rețea și de a furniza o modalitate de accesare a acestora;
- **IEEE1451 Parameter With Update**, idem cu cel anterior, având în plus asociat un mecanism ce permite actualizarea acestor variabile;
- **IEEE1451 Physical Parameter**, împreună cu subclasele asociate, utilizat pentru reprezentarea variabilelor, modelate de către clasa superioară Parameter With Update (care reprezintă în mod direct sau indirect realitatea fizică). Această clasă furnizează concret informații necesare pentru interpretarea unei măsurări sau a unei acționări;
- **IEEE1451 Scalar Parameter**, pentru modelarea elementelor cantitative care nu au asociate dimensiuni sau orientări și au caracter matematic pur scalar;
- **IEEE1451 Scalar Series Parameter**, utilizat în modelarea elementelor ce descriu o succesiune de valori scalare distribuite în mod uniform de-a lungul unei dimensiuni fizice;
- **IEEE1451 Vector Parameter**, necesar în cazul modelării variabilelor ce descriu cantități fizice cu dimensiuni multiple sau orientări asociate, având caracter matematic pur vectorial;
- **IEEE1451 Vector Series Parameter**, care modelează o serie uniformă de mărimi vectoriale;
- **IEEE1451 Time Parameter**, utilizat în reprezentarea mărimilor cu parametrii temporali. Această clasă, împreună cu subclasele asociate, are rolul de a modela variabile care reprezintă în mod direct sau indirect momentul de timp al apariției

- unui eveniment, durata dintre două evenimente sau anumite caracteristici exclusiv temporale ale unui eveniment;
- **IEEE1451 Action**, care furnizează un model pentru reprezentarea activităților care modifică starea sistemului și pentru a fi executate necesită o durată relativ ridicată față de alte proceduri;
 - **IEEE1451 File**, reprezentând o abstractizare a resurselor de date. Fișierele reprezintă un bloc de memorie care poate fi accesat prin toate metodele (deschidere, închidere, citire, scriere);
 - **IEEE1451 Partitioned File**, pentru fișiere divizate în mai multe partiții;
 - **IEEE1451 Component Group**, utilizat pentru specificarea unor clase de apartenență între obiectele unui sistem;
 - **IEEE1451 Service Abstract**, care va fi rădăcina pentru ierarhia claselor tuturor obiectelor de tip Service. Clasele Service reprezintă tipuri de obiecte care implementează comunicația între blocuri și diverse aspecte funcționale ale acestora;
 - **IEEE1451 Base Port**, fiind rădăcina ierarhiei claselor tuturor obiectelor de tip port de comunicație utilizate pentru transmiterea informației pe rețeaua fizică;
 - **IEEE1451 Base Client Port**, reprezentând rădăcina ierarhiei claselor tuturor obiectelor care modelează comunicația de tip client – server;
 - **IEEE1451 Client Port**, furnizând interfața de comunicare client – server pentru aplicațiile aferente părții client. Abstractizează caracteristicile particulare ale rețelei, fiind furnizate două modele pentru acest tip de comunicație: „blocking” (cu blocare) și „send and forget” (cu trimitere și ignorare);
 - **IEEE1451 Asynchronous Client Port**, asigurând funcționalitatea părții client pentru un model de comunicație asincron, de tip client – server;
 - **IEEE1451 Base Publisher Port**, furnizând funcționalitatea de bază a componentei publisher și a subclaselor asociate;
 - **IEEE1451 Publisher Port**, asigurând funcționalitatea completă la nivelul părții publisher pentru modelul de comunicație publisher – subscriber;
 - **IEEE1451 Self Identifying Publisher Port**, constituind un model ce implementează funcționalitatea la nivelul comunicației publish – subscribe care permite unui „subscriber” să stabilească legătura cu un „publisher” și posibilitatea celui din urmă să notifice un „subscriber” în cazul modificării procedurii de publicare;
 - **IEEE1451 Event Generator Publisher Port**, care permite unor evenimente la nivelul intern al unui bloc funcțional să publice prin sistemul publisher – subscriber;
 - **IEEE1451 Subscriber Port**, furnizând obiecte care dispun de un mecanism de aderare (subscribing) la operații de publicare;
 - **IEEE1451 Mutex Service**, asigurând capacitatea de excludere mutuală;
 - **IEEE1451 Condition Variable Service**, pentru implementarea capacității de gestionare a activităților concurente.

Pentru identificarea obiectelor descrise mai sus, vom utiliza proprietățile prescrise în specificațiile standardului IEEE 1451, care definesc aceste obiecte prin:

- **Class ID**, care furnizează clasa obiectului, poziția ierarhică, cu valoare unică;
- **Class Name**, conținând o descriere textuală a semanticilor clasei în cauză;
- **Object ID**, fiind unic în cadrul unui sistem și realizând distincția între obiecte;
- **Object Tag**, prin care se definește o destinație logică pentru partea server în cazul comunicației server – client, având valoare unică;
- **Object Name**, furnizând o descriere textuală a semanticilor instanței unei clase și care va fi definit la generarea obiectului;

- **Object Dispatch Address**, reprezentând adresa specifică rețelei, utilizată de către infrastructura rețelei de comunicație pentru a adresa obiectul. În cazul subsistemului inteligent propus, pentru rețeaua Ethernet, valoarea acestui parametru va fi adresa IP.

Modelul de date prevede definirea unor tipuri variate de primitive și derivate care vor interveni în funcționalitatea claselor de obiecte. Acestea sunt implementate direct prin limbajul de programare C utilizat și au fost descrise în paragraful 4.2.6 și prezentate în figura 4.14, 4.15 și 4.16.

Un tip derivat de date, utilizat în comunicarea pe rețea, este tipul *Argument*, reprezentând recipientul care poate conține oricare din celelalte tipuri definite prin specificațiile IEEE 1451.1 [66]. Datele aferente tuturor aplicațiilor care permit comunicarea pe rețea vor fi transportate prin intermediul unor matrici de argumente rezultând tipul de date *Argument Array*.

Clasele de obiecte necesită definirea unui antet cu un format comun, specific standardului IEEE 1451, prezentat în tabelul 4.7.

Denumire	Descriere
Class ID	Matrice de octeți (unsigned char), utilizat pentru marcarea poziției clasei în cadrul ierarhiei
Class Description	Denumirea formală a clasei, interpretată ca un șir de caractere IEEE 1451
Parent Class Name	Valoarea descriptorului de clasă a clasei părinte de unde derivă subclasa în cauză

Tabelul 4.7. Antet pentru clasele de obiecte

Operațiile implementate pentru clasele de obiecte în cazul comunicației de tip client – server au ca rezultat tipurile de date *OpReturnCode* și *ClientServerReturnCode*.

OpReturnCode este declarat ca un întreg, pe 16 biți, fără semn, structurat pe două câmpuri (minor și major) și este utilizat ca rezultat al majorității operațiilor de tip IEEE 1451.1 [66], [67].

Octetul superior descrie câmpul minor iar cel inferior, câmpul major. De exemplu, o valoare hexazecimală de 0x109 pentru *OpReturnCode*, va semnifica un câmp minor de valoare 9 și câmpul major de valoare 1. Rezultatul operației (valoarea returnată) va fi combinația celor două câmpuri.

ClientServerReturnCode este reprezentat pe 32 de biți (unsigned int) și se poate interpreta ca o succesiune de patru câmpuri: *portCode*, *performCode*, *operationMinorCode* și *operationMajorCode*, având semnificațiile expuse în tabelul 4.8.

Câmp	Descriere
portCode	Rezultatul funcției client, fiind octetul superior
performCode	Rezultatul funcției server, fiind următorul octet
operationMinorCode	OpReturnCode minor pentru aceste operații
operationMajorCode	OpReturnCode major pentru aceste operații

Tabelul 4.8. Semnificațiile operației *ClientServerReturnCode*

Rezultatul 0x05020109 în hexazecimal pentru *ClientServerReturnCode*, va însemna următoarele valori: portCode=5, performCode=2, operationMinorCode=1 și operationMajorCode=9.

Interacțiunea claselor de obiecte este reprezentată pe baza conceptului „card cage” (o placă de bază la care se pot atașa module), expus și în paragraful 4.2.2, figura 4.8. Clasele formează blocurile funcționale principale, percepute ca module software, care se pot conecta (virtual) la placa de bază („backplane”), constituind structura globală a NCAP [67].

4.4.1.1 Elemente de bază UML

Pentru descrierea funcționalității claselor de obiecte aferente NCAP, vom recurge la utilizarea limbajului UML care se caracterizează prin elemente unice de modele și se bazează pe descrierea la nivel înalt a comportamentului unui sistem prin operații și metode.

O clasă se definește prin specificarea numelui, a atributelor și a operațiilor prin care se caracterizează, așa cum se prezintă în figura 4.32.

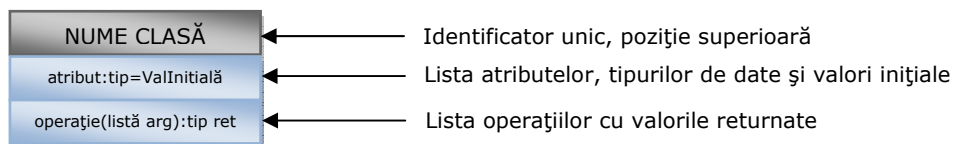


Fig.4.32. Declararea claselor în UML

În UML, o operație semnifică o acțiune sau funcție impusă unui obiect al unei clase cu scopul de a modifica comportamentul sistemului. Metoda reprezintă implementarea unei operații [68].

Pentru a indica modul de acces la informația conținută în cadrul unei clase, limbajul de programare oferă indicatori de vizibilitate după cum urmează:

- *private* (+), care maschează informația față de orice element din afara partiției clasei în cauză;
- *public* (-), care permite tuturor claselor să acceseze informația astfel marcată;
- *protected* (#), care oferă posibilitatea subclasselor de a accesa informația moștenită de la o clasă părinte (superioară).

Conceptul de vizibilitate, similar cu încapsularea (ascunderea) informației în Programarea Orientată pe Obiecte (limbaje uzuale C++, Java etc.), este expus sub formă grafică în figura 4.33.

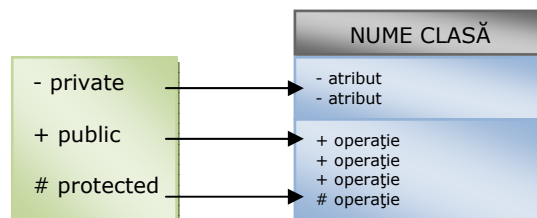


Fig.4.33. Criterii de vizibilitate în UML

Generalizarea este similară cu moștenirea proprietăților în cadrul claselor, rezultând clasele derivate sau subclassele unei clase de bază.

Legătura relațională (ierarhia) între mai multe clase este simbolizată sub forma prezentată în figura 4.34.

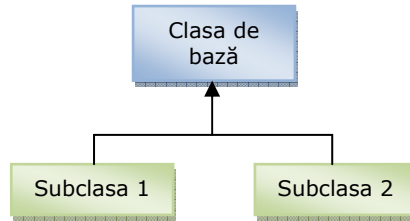


Fig.4.34. Conceptul de generalizare în UML

Pentru a permite definirea apartenenței unei clase față de alta, UML oferă conceptul de agregare, simbolizat printr-o terminație în formă de romb a segmentului ce unește două obiecte. Astfel se semnalează că un obiect conține logic sau fizic alte obiecte. Figura 4.35 indică un exemplu în care un (1) Obiect1 deține zero sau mai multe (0..*) Obiect2.

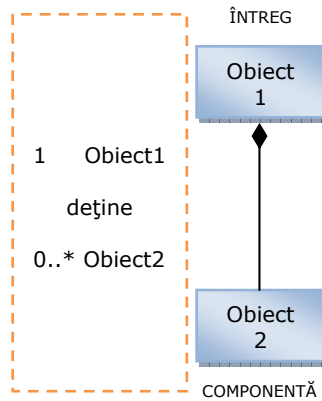


Fig.4.35 Conceptul de agregare în UML

Clasele agregate trebuie să fie independente între ele iar clasa superioară va deține întotdeauna un rol mai important decât clasa componentă (înglobată) [68].

Termenii și conceptele utilizate de către UML prevăd și modul de specificare a diagramelor de stare pe baza cărora se descrie comportamentul claselor definite prin metodele expuse anterior. Vom expune în continuare doar modalitățile de reprezentare în UML a stărilor unui automat secvențial, teoria generală și aplicațiile moderne ale acestora fiind larg dezbătute în literatura de specialitate, semnificativă fiind lucrarea „Automata Theory with Modern Applications” elaborată de către J. A. Anderson și T. J. Head editată la Cambridge University Press în 2006 [69].

Reprezentarea stărilor se realizează cu ajutorul unor simboluri dedicate care indică starea inițială, starea finală, stările intermediare precum și tranzițiile între acestea, așa cum este prezentat în figura 4.36.



Fig.4.36. Simboluri UML pentru diagrame de stare

4.4.1.2 Clasa IEEE1451_Block

Pentru descrierea claselor din componența modelului NCAP, vom utiliza mediul „Software Ideas Modeler” (modelare software prin diagrame UML), prezentat exemplificativ în figura 4.37, care dispune de o versiune cu licență gratuită pentru cercetare și oferă un cadru simplu și eficient de elaborare a proiectelor în UML.

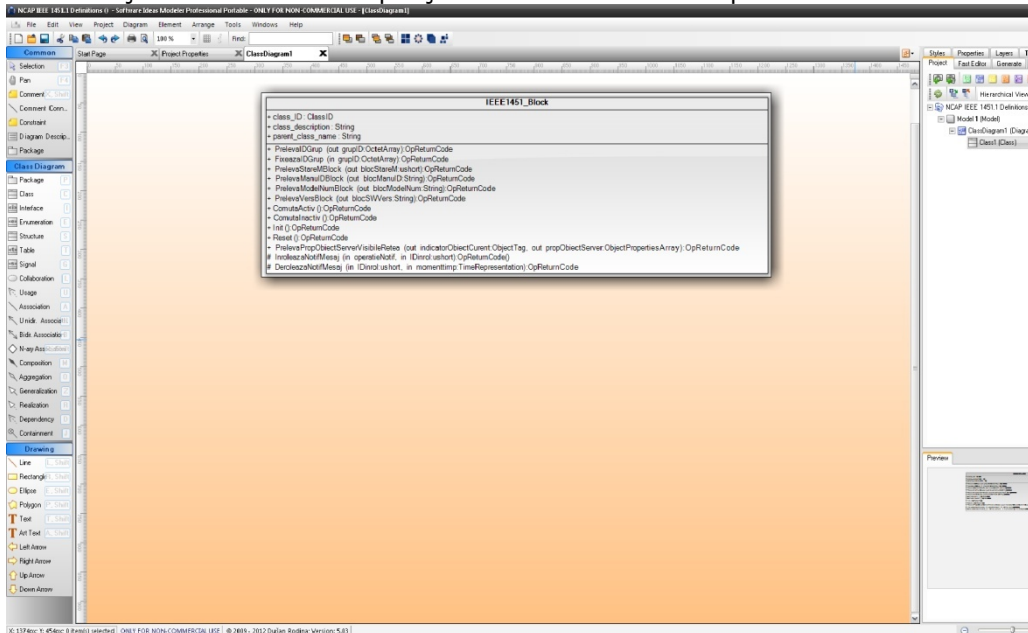


Fig.4.37. Mediul de modelare Software Ideas Modeler - UML

Nivelul ierarhic superior al tuturor obiectelor componente ale modelului îl reprezintă IEEE 1451_Block, care prin limbajul UML și pe baza [66], [67] poate fi definit și structurat așa cum se indică în figura 4.38.

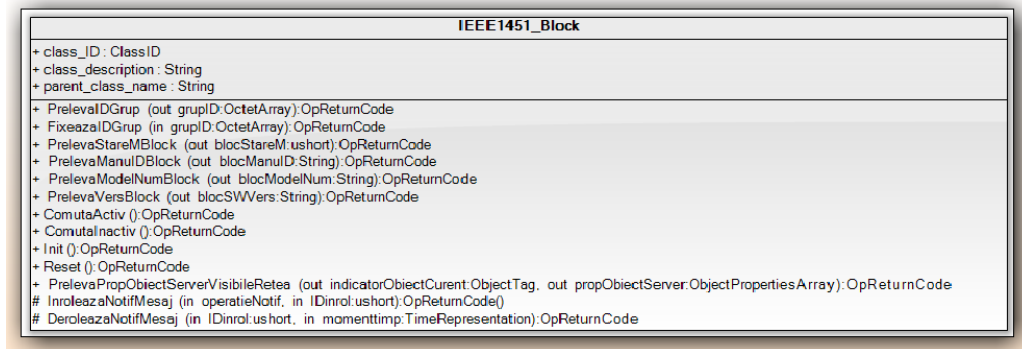


Fig.4.38. Descrierea în UML a blocului IEEE1451_Block

Atributele și operațiile incluse în acest bloc respectă specificațiile IEEE 1451.1 pentru modelul NCAP și nu vor fi descrise pe larg în cadrul lucrării [66], [67]. În schimb, comportamentul acestei clase se supune regulilor de funcționare a

unui automat secvențial cu trei stări principale, indicate în standardul IEEE 1451.1, prezentate în figura 4.39 [67], [69].

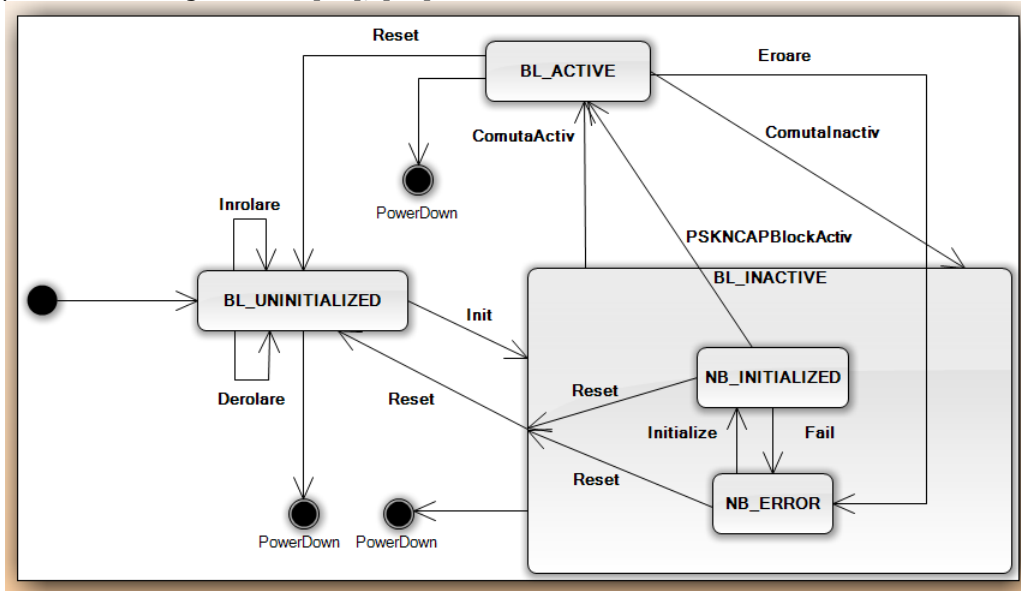


Fig.4.39. Diagrama de stare aferentă clasei IEEE 1451_Block

Blocul NCAP este controlat prin trei stări: BL_UNINITIALIZED, BL_INACTIVE și BL_ACTIVE.

BL_UNINITIALIZED este rezervată pentru operații locale (înrolare/derolare) care conduc la crearea obiectului „Block” și la pregătirea tuturor funcțiilor acestuia. Tot în cadrul acestei stări se înrolează toate celelalte obiecte care implementează comunicația pe rețea și care sunt deținute de NCAP Block.

BL_INACTIVE, o stare compusă, este destinată activităților de configurare a proprietăților de comunicare pe rețea a blocului NCAP Block și a blocurilor deținute de către acesta. Totodată are și rolul de inițializare, diagnosticare și mentenanță a blocului NCAP Block. Se poate descompune în stările *NB_INITIALIZED* și *NB_ERROR* care apelează la mecanisme de tranziție din cadrul clasei IEEE1451_NCAP Block. Tranziția inițială din starea BL_UNINITIALIZED către starea NB_INITIALIZED va avea loc la apelarea operației **Init** al clasei IEEE1451_Block (figura 4.38). Intrarea NCAP în starea NB_ERROR se va realiza atunci când se generează intern tranziția **Fail** sau dacă modulul NCAP detectează o eroare și apelează la tranziția **ComutăInactiv**.

BL_ACTIVE reprezintă starea în care au loc activitățile rezultate prin rularea normală a aplicațiilor aferente blocului NCAP. Se poate ajunge în această stare prin tranziția **ComutăActiv** sau în urma trecerii prin sub – starea NB_INITIALIZED și efectuării tranziției **PSKNCAPBlockActiv**.

Fiecare obiect „Block” definit prin standardul IEEE 1451 va moșteni proprietățile automatului secvențial prezentat în figura 4.39 precum și particularitățile clasei IEEE1451_Block din figura 4.38.

Stările și tranzițiile aferente diagramei din figura 4.39 se pot implementa prin obiecte și funcții scrise în limbajul de programare C++ [67].

4.4.1.3 Clasa IEEE1451_NCAP Block

Elementul cheie pentru implementarea comunicației pe rețea și a configurării întregului sistem îl constituie clasa IEEE1451_NCAP Block. Această clasă deține toate celelalte obiecte „Block” din cadrul aceleiași ierarhii și în plus are proprietatea de a se auto-deține (așa cum se observă și în figura 4.31) [66], [67].

Modul de definire și structura acestei clase sunt prezentate în figura 4.40.

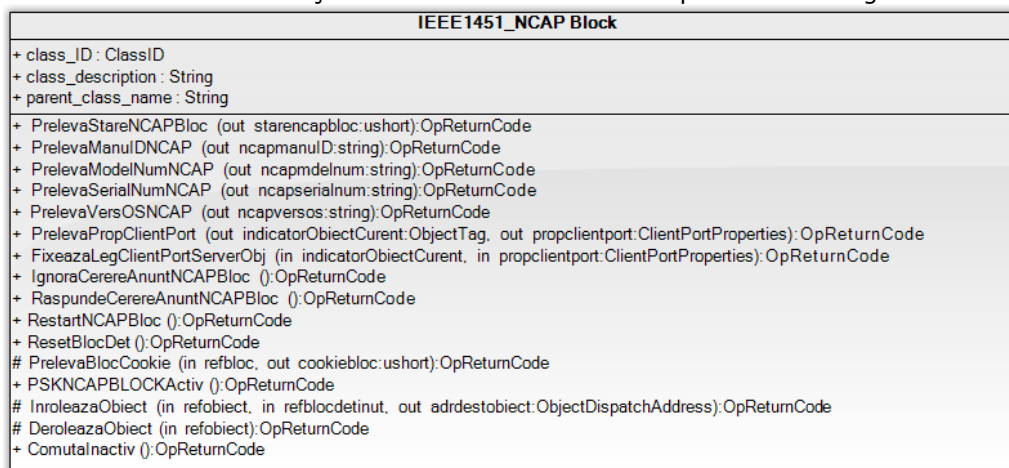


Fig.4.40. Descrierea în UML a blocului IEEE1451_NCAP Block

Comportamentul acestei clase este controlat de către automatul secvențial asociat clasei IEEE1451_Block care include sub-stările NB_INITIALIZED respectiv NB_ERROR, expuse în figura 4.39 (sub-stări ale NCAP Block).

4.4.1.4 Clasa IEEE1451_FunctionBlock

Mecanismul principal pentru abstractizarea și încapsularea funcționalității aplicațiilor îl constituie clasa de funcții IEEE1451_FunctionBlock [67], definită în UML așa cum se prezintă în figura 4.40.

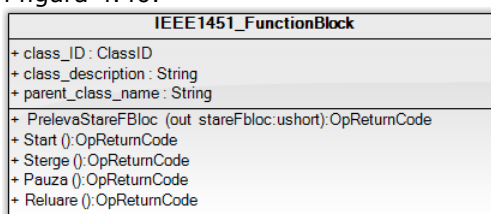


Fig.4.41. Clasa IEEE1451_FunctionBlock

Obiectele specifice aplicațiilor vor fi controlate și deținute de către clasa FunctionBlock. Comportamentul acestora este impus de către stările moștenite de la IEEE1451_Block, starea BL_ACTIVE fiind compusă din trei sub-stări: *FB_STOPPED*, *FB_RUNNING* și *FB_IDLE*, [67], [69], expuse în figura 4.42. Tranzițiile între aceste stări se realizează pe baza operațiilor definite în clasa IEEE1451_FunctionBlock, prezentată în figura 4.41.

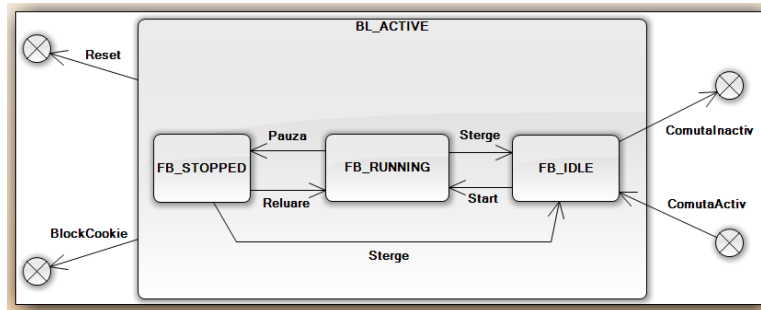


Fig.4.42. Diagrama de stare IEEE1451_FunctionBlock

4.4.1.4 Clasele de comunicație Client – Server

Standardul IEEE1451 specifică două modele de comunicație suportate de către NCAP: Client – Server și Publish – Subscribe [66]. Subsistemul inteligent propus în lucrare va utiliza doar modelul Client – Server pentru comunicarea pe rețeaua Ethernet. Ca urmare, vom defini în continuare doar clasele IEEE1451_Entity și IEEE1451_ClientPort.

Modelul Client – Server se utilizează cu precădere în comunicația pe rețea a două noduri. Nivelul „Server” al NCAP este furnizat de către clasa Entity (figura 4.43), prin intermediul operației „ExeCS” [67].

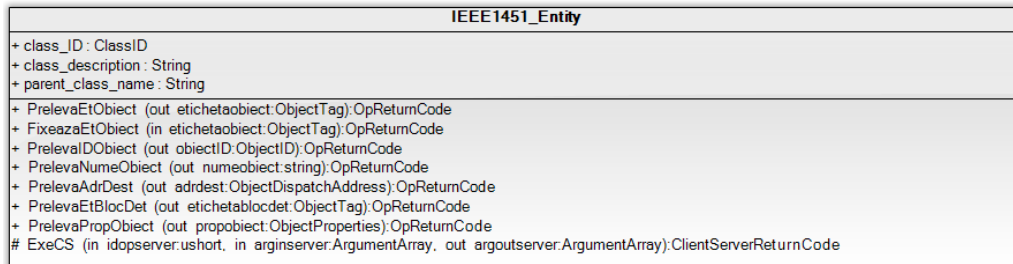


Fig.4.43. Clasa IEEE1451_Entity

Funcționalitatea nivelului „Client” al NCAP este implementată prin intermediul clasei IEEE1451_ClientPort [67], prezentată în figura 4.44, în care este definită operația „ExeC”.

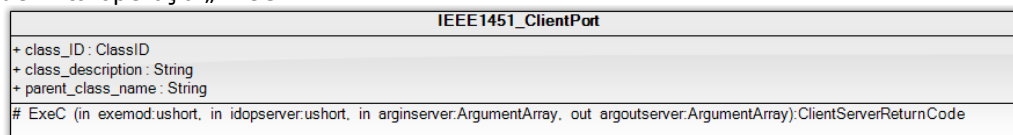


Fig.4.44 Clasa IEEE1451_ClientPort

Informațiile referitoare la portul către rețea vor fi stabilite prin rutinele software și astfel în momentul în care NCAP va fi operațional, obiectele server și client vor putea comunica în mod direct pe rețeaua Ethernet.

În cadrul subsistemului propus vom adapta și implementa comunicația client – server cerințelor aplicației, pe baza modelului de interacțiune uzual [67], [70], cu operațiile ilustrate în mod simplificat, așa cum este prezentat în figura 4.45.

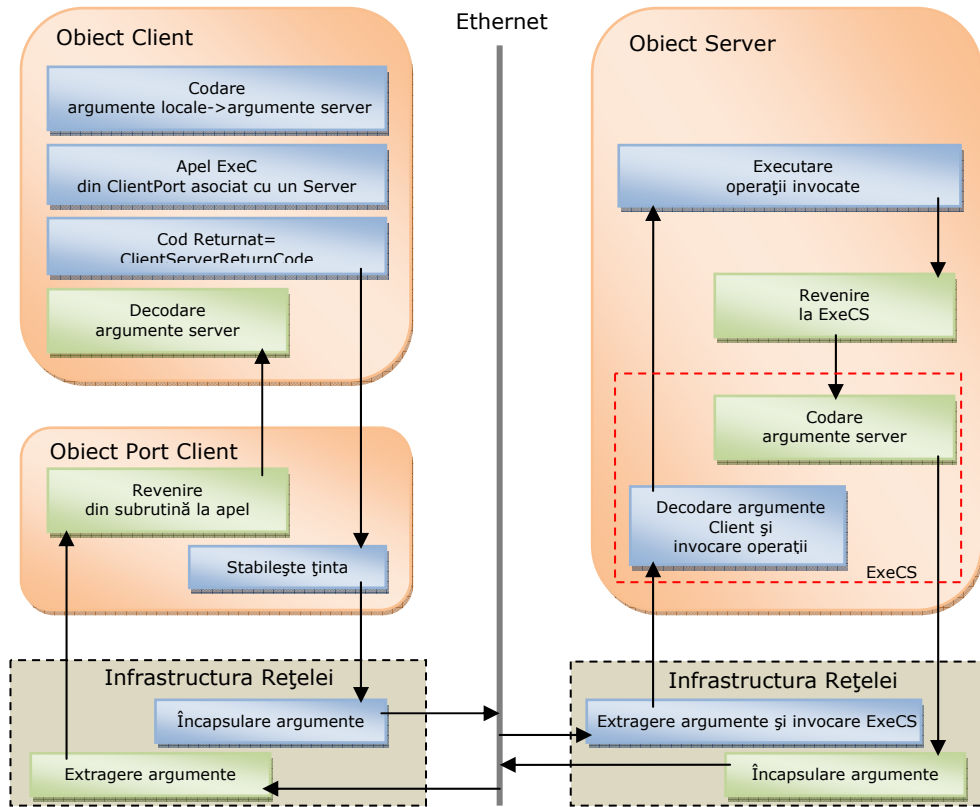


Fig.4.45. Modelul de comunicație Client-Server pentru subsistemul inteligent

Codarea argumentelor locale cu scopul de a obține formatul de intrare acceptat de către Server are loc la nivelul obiectului Client. Operația „ExeC”, definită în clasa IEEE1451_ClientPort, este apelată cu parametri de intrare „argument intrare Server” precum și cu informațiile referitoare la adresa acestuia. Obiectul „Port Client” determină în continuare destinația mesajului iar încapsularea datelor pentru forma adecvată vehiculării pe rețea este realizată la nivelul infrastructurii rețelei Ethernet.

Infrastructura rețelei, aferentă obiectului Server, extrage argumentele și invocă operația „ExeCS” definită în clasa IEEE1451_Entity. Urmează decodarea informației și executarea operațiilor cerute de către Client. Rezultatele sunt retransmise părții Client prin reapelarea „ExeCS”, codarea informației, încapsularea argumentelor pentru vehicularea pe infrastructura rețelei, extragerea datelor și în final decodarea argumentelor recepționate la nivelul obiectului Client.

4.4.1.5 Clasa IEEE1451_TransducerBlock

Principalele operații efectuate la nivelul NCAP referitoare la modulele TIM și transductoare sunt accesarea parametrilor din zona Meta-TEDS și interogarea globală a tuturor canalelor individuale. Legătura dintre transductoarele fizice atașate TIM și cele generic definite în cadrul blocului funcțional Transducer Block din NCAP

este realizată cu ajutorul clasei *IEEE1451_TransducerBlock* [67]. Structura sa este prezentată în figura 4.46.

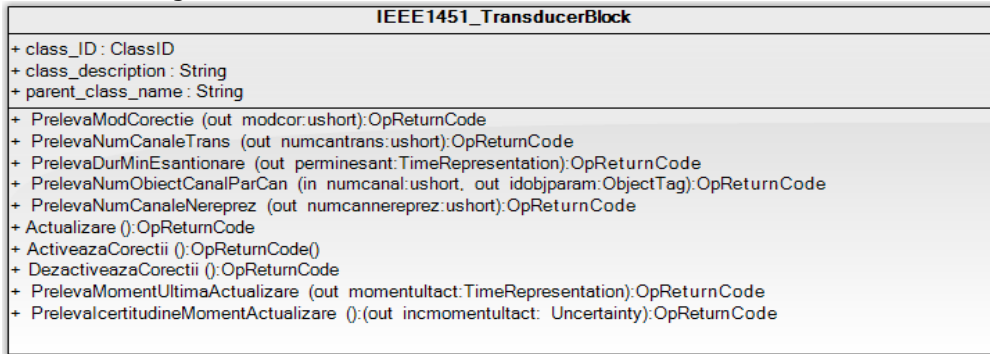


Fig.4.46. Clasa IEEE1451_TransducerBlock

Diagramele de stare care descriu comportamentul blocului Transducer sunt cele definite pentru toate obiectele bloc din *IEEE1451_Block* (vezi figura 4.39).

Clasa *TransducerBlock* tratează canalele transductor ca instanțe ale unei subclase a clasei *Component* asociată unui anumit canal.

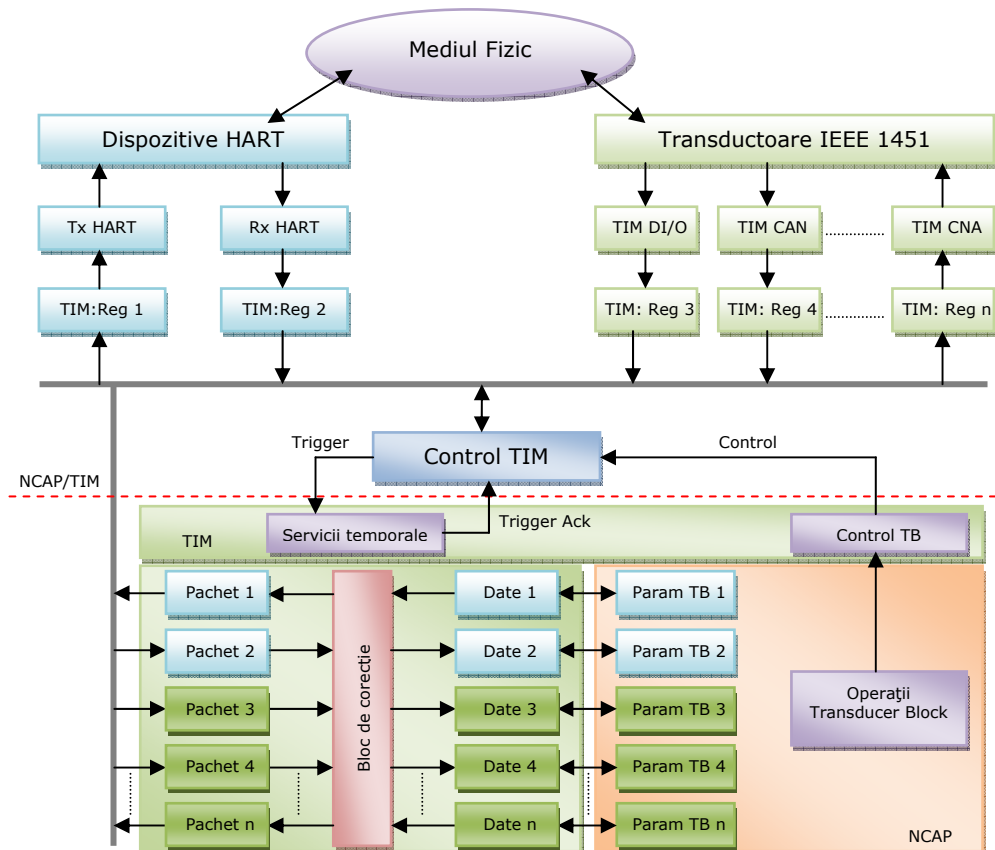


Fig.4.47. Modelul funcțional al blocului transductor

Modelul funcțional elaborat pentru subsistemul inteligent, propus în lucrare, va conține elementele indicate în figura 4.47. De exemplu, pentru un senzor conectat la un convertor analog – numeric vom avea în urma acestei modelări o reprezentare de tip parametru scalar la nivelul NCAP. Fiecare registru din cadrul Transducer Block (care va conține Pachet 1,...,Pachet n) are un corespondent la nivelul TIM (TIM: Reg 1,...,TIM: Reg n) iar în urma citirii conținutului din TEDS aceste registre vor avea același conținut și va fi stabilit cu exactitate și numărul acestora (în funcție de numărul de canale transductor). Următorul set de registre (Date 1,...,Date n) este necesar stocării datelor rezultate în urma aplicării algoritmilor de corecție indicați de către TEDS. În cazul în care modulul TIM nu necesită astfel de corecții, cele două seturi de registre menționate pot fi modelate ca unul singur (Pachet și Date se transformă în Date și dispăre blocul de corecție).

Pentru a implementa comunicația cu dispozitivele HART, recurgem la alocarea a două canale transductor reprezentate în figura 4.47 prin „Tx HART” și „Rx HART” cărora le-am atribuit registrele „TIM: Reg 1” respectiv „TIM: Reg 2”. Valorile acestora vor fi încărcate sau citite din blocul UART necesar transmiterii/recepționării pachetelor de date HART. Structura propusă în figura 4.47 va permite totodată conectarea unor transductoare suplimentare, compatibile IEEE 1451, prin alocarea unui număr variabil de canale transductor, reprezentate prin setul de registre „TIM: Reg 3,..., TIM: Reg n”, cu n maxim 255.

La nivelul NCAP, canalele transductor vor fi sesizate și tratate sub forma unor parametri, stocați în registrele indicate în model sub denumirea „Param TB” (Parametri Transducer Block). Astfel va fi necesară definirea unei clase „Parametri” care va modela în mod individual toate transductoarele conectate la subsistem.

4.4.1.6 Clasa IEEE1451_ParameterWithUpdate

Standardul IEEE 1451 prevede posibilitatea interogării de către NCAP, în mod individual, a canalelor transductor precum și accesarea informației din Meta – TEDS. Vom crea în acest sens o clasă „Parametru” cu subclase atașate pentru citirea informației din Channel – TEDS așa cum se indică și în [66], [67].

Clasa *IEEE1451_ParameterWithUpdate* va conține doar operațiile majore efectuate asupra unui canal transductor: citirea valorii furnizate de către un senzor și scrierea unei valori pentru comanda unui actuator. Similar, pentru comunicarea cu dispozitivele HART vom implementa aceleași două moduri de interogare pentru a asigura o tratare unitară la nivelul NCAP a informației. Figura 4.48 prezintă structura clasei *IEEE1451_ParameterWithUpdate*.

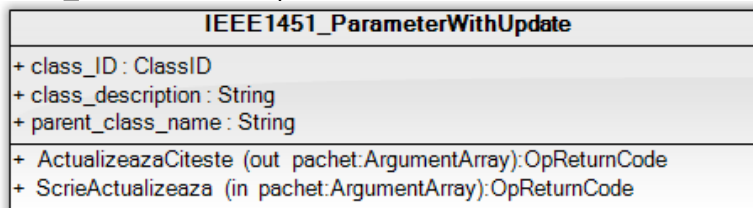


Fig.4.48. Clasa IEEE1451_ParameterWithUpdate

Operația „ActualizeazaCiteste” va avea ca parametru de ieșire pachetul de date furnizat de către canalul „Rx HART” sau de către un senzor conectat la oricare alt canal.

Operația „ScrieActualizeaza” preia ca parametru de intrare pachetul de date aferent canalului „Tx HART” sau a oricărui alt canal la care este conectat un actuator.

Subclasa care moștenește proprietățile *IEEE1451_Parameter* și care va permite accesarea Channel – TEDS va fi *IEEE1451_PhysicalParameter*, având structura indicată în figura 4.49.

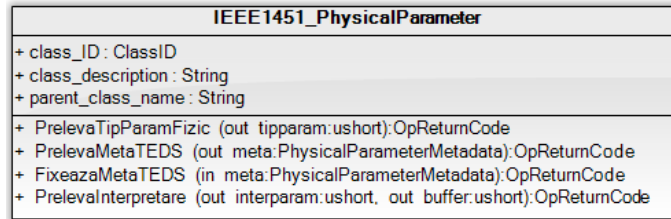


Fig.4.49. Clasa IEEE1451_PhysicalParameter

Pentru determinarea tipului de transductor conectat la un canal am definit operația „PrelevaTipParamFizic”, cu valoarea de ieșire un număr întreg.

Operația „PrelevaMetaTEDS” va permite accesarea Channel – TEDS aferent fiecărui transductor atașat.

În cazul canalelor dedicate comunicației HART, va fi necesară în prealabil obținerea informațiilor referitoare la natura dispozitivelor conectate (senzor sau actuator) și înscrierea acestora în blocul TEDS intern, dedicat. Urmează redirecționarea prelevării datelor suplimentare către această memorie internă care conține datele TEDS. Operația „FixeazaMetaTEDS” va fi utilizată în acest scop, ea fiind opțională în specificațiile standardului IEEE 1451.

Tipul datelor vehiculate prin canalele transductor va fi determinat prin apelarea operației „PrelevaInterpretare”, realizându-se diferențierea între valorile citite de la un senzor și cele transmise către un actuator.

4.4.1.8 Modelul UML final al NCAP

În vederea implementării modelului informațional aferent subsistemului propus, blocul NCAP a fost restructurat și a fost concepută ierarhia principalelor clase de obiecte, bazându-ne pe [66], [67], obținându-se modelul prezentat în figura 4.50.

Pe baza acestor clase și a relațiilor dintre ele, se pot elabora codul sursă în limbajul de programare C++. Rutinele vor rula pe o platformă de microprocesor de tip MicroBlaze, oferită de firma Xilinx sub formă de implementare software într-un FPGA.

Avantajul utilizării acestei configurații constă în vehicularea informației pe o magistrală comună și posibilitatea coexistenței blocurilor constructive în cadrul aceluiași circuit integrat.

În plus, metoda propusă oferă o flexibilitate ridicată și permite o restructurare a sistemului prin simpla rescriere a anumitor rutine software.

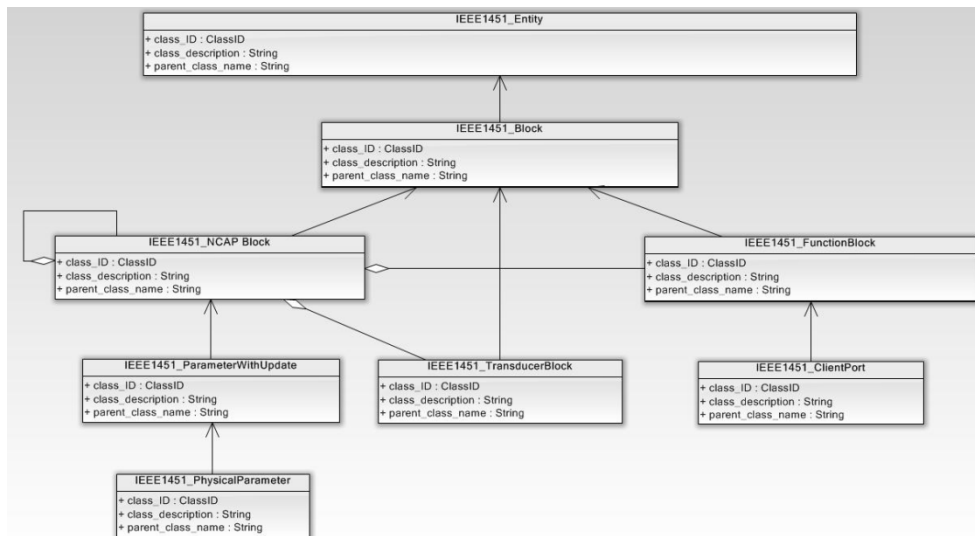


Fig.4.50. Modelul informațional al NCAP definit în UML

4.4.2. Comunicația NCAP – TIM

Standardul IEEE 1451 prevede posibilitatea conectării unui singur modul TIM la un NCAP, prin intermediul interfeței specializate TII. Subsistemul inteligent propus va îngloba o structură NCAP/TIM combinată, interfața TII fiind implementată la nivelul intern al circuitului integrat. Structura comenzilor va conține atât adresa canalului cât și codul efectiv al comenzii asociate (le vom denumi adresă canal respectiv adresă funcțională). Ambele adrese vor fi reprezentate pe 8 biți, astfel existând posibilitatea definirii unor comenzi suplimentare în cazul unor dezvoltări ulterioare. Setul de comenzi mandatorii, prescrise de către standardul IEEE 1451, este prezentat în tabelul 4.9.

Adresa funcțională	Comandă canal 0	Comandă canal singular
0	Citire Meta-TEDS	Citire Channel TEDS
1	Scriere mască întrerupere canal 0	Scriere mască întrerupere standard
2	Citire mască întrerupere canal 0	Citire mască întrerupere standard
3	Citire stare canal 0	Citire stare canal
4	Interogare canal 0	Interogare canal transductor
5	Scriere date globale	Scriere date canal
6	Citire date globale	Citire date canal
7	RESET	Rezervat
8...255	Dezvoltări ulterioare	Dezvoltări ulterioare

Tabelul 4.9. Setul de comenzi mandatorii IEEE 1451

Spațiul de adrese 8...255 rămâne disponibil pentru definirea unor comenzi specifice anumitor aplicații. Se pot implementa de exemplu comenzi aferente controlului exclusiv al unor dispozitive de teren HART conectate la subsistem.

4.4.2.1 Interogarea de către NCAP a TIM

În momentul interogării TIM, canalul aferent unui senzor va fi eșantionat sau cel aferent unui actuator va fi actualizat cu o anumită valoare. Vor fi implementate atât interogările individuale cât și cele globale. Unitatea de control „Control TIM”, prezentată în figura 4.47 va genera semnalul „Trigger” și va gestiona controlul registrelor aferente canalelor transductor. Secvența operațiilor efectuate de către unitatea de control este prezentată în figura 4.51.

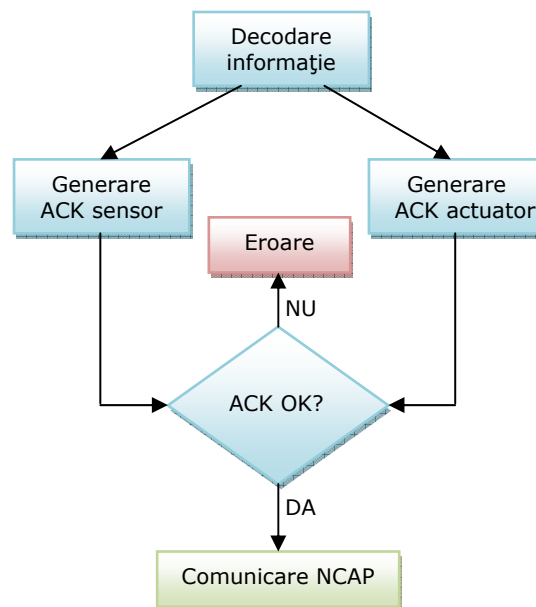


Fig.4.51. Operațiile aferente blocului „Control TIM”

Unitatea de control va decoda comenzile emise de către NCAP și le transmite canalului adresat. Semnalul de acceptare (Acknowledge – ACK) va fi generat în moduri diferite, funcție de tipul dispozitivului în cauză: senzor sau actuator.

În cazul unui senzor, semnalul ACK se emite atunci când se încheie conversia analog – numerică iar rezultatul se memorează în registrele aferente din TIM. Pentru un senzor cu ieșire digitală acest semnal va fi generat în urma memorării rezultatului în registrele TIM.

Pentru un actuator având comandă numerică, semnalul ACK va fi generat în momentul recepționării cererii de interogare (trigger). Dacă informația este trimisă la un convertor numeric – analogic, semnalul de acceptare va fi generat în urma încheierii conversiei.

La o interogare globală (comandă aplicată la Canalul 0), semnalul de acceptare va fi generat de către transductoare în mod individual, pe măsură ce operațiile au fost executate cu succes.

4.4.2.2 Generarea întreruperilor și registrele de stare

Standardul IEEE 14511 prevede patru biți de stare, definiți în mod obligatoriu, pentru Canalul 0 (Channel Zero) și trei biți pentru fiecare canal independent. Aceștia vor fi implementați și utilizați în cazul subsistemului propus și sunt prezentați în tabelul 4.10.

Canal 0	Canale individuale
Trigger ACK	Trigger ACK
Invalid Command	Channel Operational
TIM Operational	---
Corrections En/Dis	Corrections En/Dis

Tabelul 4.10. Biți de stare asociați canalelor transductor

Implementarea biților de stare se va realiza prin asocierea registrelor aferente cu zone de memorie RAM disponibile în cadrul subsistemului propus.

Sistemul de generare al întreruperilor trebuie să țină cont de valorile biților de stare și gestionarea acestora se va realiza cu ajutorul unui controler dedicat care va funcționa pe principiul unui codificator prioritar.

4.4.3. Transmiterea telegramelor HART pe suport Ethernet

Aplicațiile de control în timp real trebuie să satisfacă diverse cerințe, dezbătute amănunțit pe parcursul Capitolului II. Pentru vehicularea datelor remarcăm două condiții fundamentale: determinismul și robustețea. Rețelele industriale exploatează la ora actuală avantajele oferite de infrastructura Ethernet pentru comunicarea între dispozitivele de teren datorită costului redus de implementare și performanțelor ridicate din punctul de vedere al ratei de transfer.

Subsistemul inteligent de legătură HART - Ethernet va recepționa telegrame HART înglobate în cadre Ethernet după o metodă propusă în urma studiului efectuat asupra structurii pachetelor de date vehiculate pe rețelele Ethernet.

4.4.3.1 Formatul cadrelor Ethernet II

Standardul Ethernet II prevede utilizarea unui așa – numit „cadru” Ethernet, având structura prezentată în figura 4.52 [11].

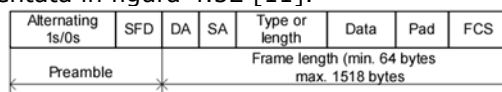


Fig.4.52. Formatul unui cadru Ethernet [11]

Cadrul de mai sus este format din opt câmpuri:

- Preamble, care conține un șir alternant de biți 1 și 0 care se extind pe lungimea a 7 octeți, urmat de SFD (Start Frame Delimiter) de lungime 1 octet;
- DA (Destination Address), adresa destinației, reprezentată pe 6 octeți;
- SA (Source Address), adresa sursei care generează datele, pe 6 octeți;
- Type or Length, care specifică tipul sau lungimea câmpului de date, pe 2 octeți;
- Data, reprezentând informația utilă (câmpul de date) având lungime variabilă, cuprinsă între 0 și 1500 octeți;

- PAD, care este un câmp de compensare a lungimii cadrului în cazul în care pachetul de date utile nu depășește 46 de octeți, având astfel o lungime variabilă între 0 și 46 octeți;
- FCS (Frame Check Sequence), utilizat ca sumă de control pentru detecția și corecția eventualelor erori, fiind reprezentat pe 4 octeți.

Este cunoscut faptul că datorită algoritmului CSMA/CD (Carrier Sense Multiple Access/Collision Detection), cadrele cu o lungime mai mică de 64 octeți pot provoca erori grave de transmisie și blocarea rețelei. De aceea, standardul prevede existența unui câmp de compensare a lungimii, PAD, care inserează biți de 0 în pachetul de date până la completarea lungimii totale de 64 octeți [11], [23].

4.4.3.2 Înglobarea telegramelor HART

Analizând formatul unui cadru Ethernet, indicat în figura 4.52, putem utiliza câmpul „Data” pentru a include aici informațiile referitoare la telegramele HART.

Vom lua în considerare structura completă a unei telegrami HART, prezentată și în paragraful 4.3.2, incluzând în structura prezentată în figura 4.53 și câmpul opțional „EXP” (0 până la 3 octeți) care a fost prevăzut în ultimele revizii ale standardului, pentru dezvoltări ulterioare. Vom utiliza acest câmp ca identificatorul telegrami HART.

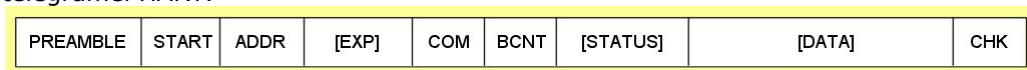


Fig.4.53. Telegramă HART completă

Urmează alcătuirea structurii noi al cadrului Ethernet – HART, care este prezentată în figura 4.54.

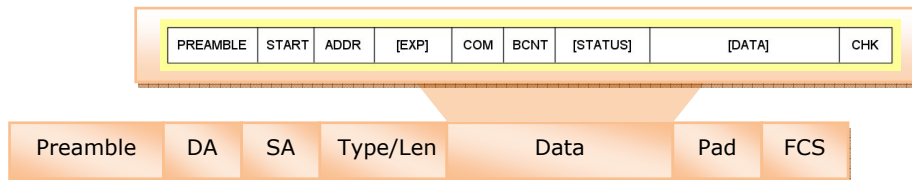


Fig.4.54. Telegrama HART înglobată într-un cadru Ethernet

Subsistemul inteligent, conectat la rețeaua Ethernet prin intermediul interfeței furnizate de către blocul NCAP din structura sa, va putea accesa telegramele HART puse la dispoziție de către un calculator gazdă. Prin citirea valorii înscrise în câmpul „EXP”, NCAP va putea identifica existența telegrami HART în cadrul Ethernet recepționat.

Noua structură obținută permite totodată includerea a mai multor telegrame HART în același cadru Ethernet, astfel mărind eficiența transmisiei în sensul raportului dintre informația utilă și cea de identificare și control.

4.4.3.3 Analiza ratei de transmisie

Protocolul HART necesită asigurarea unei rate minime de transmisie care rezultă din modul de codare a informației și numărul biților din telegramă.

Timpul necesar transmiterii unei telegrame HART clasice fiind corelat cu rata de transfer cea mai scăzută (afereantă frecvenței de 1200 Hz) și cu numărul de biți din telegramă (paragraful 4.3.2), rezultă o valoare de 0,83 ms pentru transmiterea unui bit și 13 ms pentru un octet de date. Prin implementarea noii structuri prouise a pachetului de date, se pot obține durate pentru un bit de 51,2 μ s în cazul mediului Ethernet de 10 Mb/s și de 512 ns în cazul mediului Ethernet de 1 Gb/s.

4.4.4. Funcționalitatea subsistemului inteligent

La nivel global, subsistemul inteligent, bazat pe principiile prescrise de standardul IEEE 1451, implementează operațiile necesare comunicării pe rețeaua Ethernet cu o unitate de comandă centrală sau cu alte dispozitive de control (gen NCAP), preia ca date de intrare informațiile provenite de la dispozitivele de teren conectate, trimite comenzi către actuatori și prelucrează la nivel anumite informații.

Organigrama prezentată în figura 4.55 descrie funcționalitatea subsistemului proiectat.

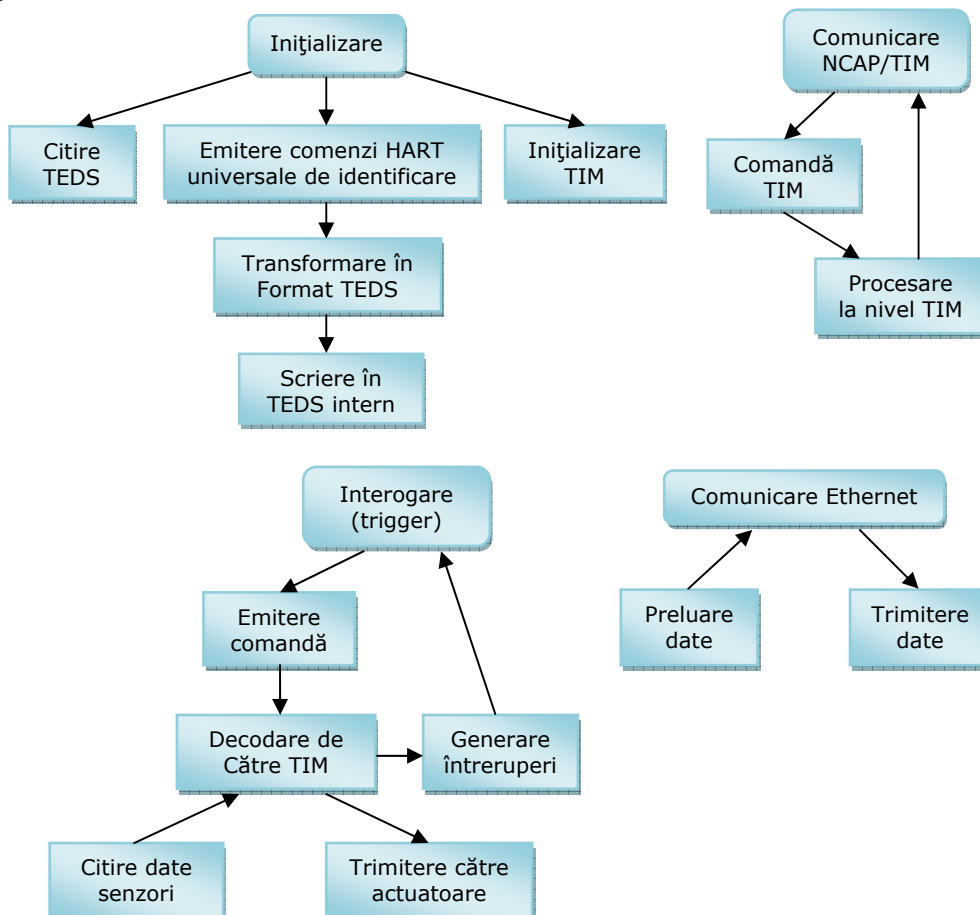


Fig.4.55. Funcționalitatea subsistemului inteligent

Inițializarea sistemului include operații atât la nivelul NCAP cât și al modului TIM înglobat. Sunt prevăzute două moduri de tratare a datelor din TEDS. În modul standard, are loc imediat citirea informației aflate în blocul TEDS, conform specificațiilor IEEE 1451. Pentru operarea cu dispozitivele de teren HART, în scopul tratării unitare a informației, la inițializare se emit comenzile HART universale de identificare 0, 1, 2, 3, 14, 15, 16 și apoi datele recepționate se transformă în format specific standardului IEEE 1451 și se înscriu într-o zonă TEDS implementată în memoria internă a sistemului.

Urmează instanțierea diverselor obiecte din modelul informațional NCAP referitoare la transductoarele identificate și trimiterea unei comenzi de resetare către TIM. În cazul în care nu apar erori în accesarea și tratarea datelor din TEDS, se va trece la inițializarea efectivă a blocului NCAP. Are loc apelarea operațiilor „InIt” și „ComutaActiv” definite în clasa IEEE1451_Block (paragraful 4.4.1.2) și evaluarea valorilor returnate (pentru detectarea unei posibile erori de execuție). În continuare se execută comanda TIM cu adresa funcțională 1, „Scriere mască întrerupere” (vezi tabelul 4.8) pentru Canalul 0 și pentru canalele individuale.

Etapă finală constă în inițializarea blocului de comunicare pe rețeaua Ethernet și detectarea subsistemului de către unitatea de control centrală.

Interogarea TIM (trigger) se realizează de către blocul NCAP prin trimiterea unei comenzi corespunzătoare către TIM care o decodifică și inițiază operația de citire a datelor provenite de la un senzor sau de transmitere a unor comenzi către actuator.

Întreruperile generate de către TIM sunt tratate de către blocul NCAP în funcție de specificul fiecăreia.

Comunicarea pe rețeaua Ethernet, bazată pe principiul Client – Server descrisă în paragraful 4.4.1.4, se poate iniția atât de către calculatorul central de comandă cât și de către subsistemul inteligent, prin intermediul interfeței specializate incluse în blocul NCAP.

4.5. Concluzii și contribuții personale

Capitolul IV tratează noțiunile fundamentale necesare conceperii unui sistem performant, de cost redus, compatibil cu specificațiile și cerințele standardelor actuale. **Principiile expuse au fost sintetizate și aduse la o formă de prezentare caracteristică** punctului de vedere al autorului, noțiunile utilizate fiind grupate și prezentate într-un **mod grafic original**.

- Etapele aferente conceperii subsistemului propus au fost următoarele:
- i) studiul principiilor de structurare și funcționare a celor mai noi sisteme de conducere a proceselor tehnologice;
 - ii) identificarea elementelor care suportă o anumită restructurare în scopul îmbunătățirii performanțelor funcționale;
 - iii) parcurgerea specificațiilor standardului IEEE 1451, primordial în conceperea și realizarea unui sistem de control modern;
 - iv) analiza modului de operare a dispozitivelor de teren HART existente la ora actuală;
 - v) expunerea structurii globale aferente subsistemului propus;
 - vi) modelarea într-un limbaj de nivel înalt (UML) a blocurilor funcționale;
 - vii) prezentarea noului mod de transmitere a datelor către subsistemul propus;
 - viii) descrierea funcțională a subsistemului.

Conceptul actual al transductorului inteligent (smart transducer) a condus la perceperea acestuia ca mai mult decât elementul simplu care transformă o formă de energie în alta. Cu ajutorul acestuia se poate comunica bidirecțional cu o unitate de control ierarhic superioară.

Fiind introdus prin standardul IEEE 1451, transductorul inteligent, împreună cu noțiuni ca TIM (Transducer Interface Module), TII (Transducer Independent Interface), TEDS (Transducer Electronic Data Sheet) și NCAP (Network Capable Application Processor), acesta constituie o abordare inovatoare a întregii problematice referitoare la conducerea proceselor tehnologice.

Migrarea către noul standard este preocuparea curentă în domeniu, iar în capitolul de față **am expus o modalitate originală**, prin conceperea unui „subsistem inteligent” care oferă o cale de tranziție către noul standard prin costuri minime datorită posibilității utilizării în paralel atât a vechilor dispozitive cât și a celor noi.

În acest sens, am selectat sistemul clasic HART de comunicare cu dispozitivele de control al proceselor tehnologice, larg răspândit la ora actuală în diverse arii de activitate, așa cum rezultă și în urma unei clasificări extrase din literatura de specialitate [44], [52], [64], expuse în figura 4.56.

Se poate observa o ușoară tendință de creștere a utilizării dispozitivelor HART, pe parcursul anilor 2002 – 2010.

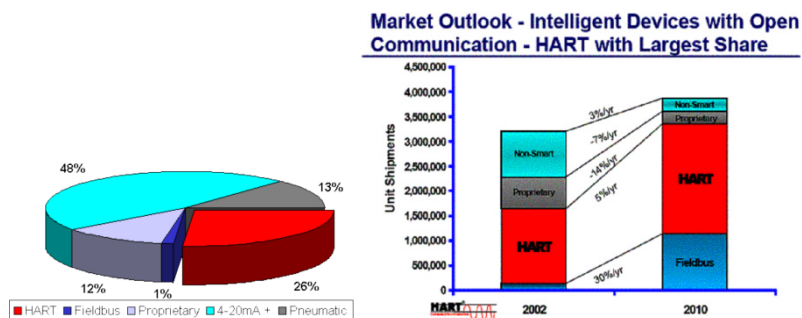


Fig.4.56. Tendința actuală în utilizarea sistemelor de control al producției

Modalitatea propusă urmărește compatibilizarea dispozitivelor de teren HART existente cu cerințele noului standard IEEE 1451, fără o modificare constructivă a acestora. Astfel **am conceput o „punte de legătură”**, materializată prin intermediul unui subsistem inteligent bazat pe principiile introduse de către standardul IEEE 1451. Nucleul subsistemului îl constituie un bloc NCAP în combinație cu un modul TIM înglobat în structura generală împreună cu circuitele necesare comunicării cu dispozitivele de teren HART. În plus, prin intermediul NCAP, noul sistem va avea capacitatea de a comunica pe rețeaua Ethernet cu alte dispozitive de control compatibile IEEE 1451.

Pe parcursul capitolului următor, voi expune modul de implementare efectivă a structurilor bloc prezentate, cu ajutorul unui circuit FPGA de tip Virtex 4, produs de către Xilinx, inclus pe o placă de dezvoltare ML-401 precum și testarea funcțională a blocurilor componente.

4.5.1. Contribuții personale

Capitolul IV reunește atât noțiuni sintetizate din literatura de specialitate și expuse prin prisma unei noi abordări cât și următoarele contribuții personale:

1. **Sintetizarea principalelor concepte actuale** referitoare la structura și cerințele impuse sistemelor de conducere a proceselor tehnologice;
2. **Identificarea elementelor** care necesită o anumită restructurare în scopul îmbunătățirii performanțelor funcționale;
3. **Expunerea printr-o viziune proprie**, ilustrativă și concentrată, a specificațiilor standardului IEEE 1451;
4. **Propunerea** unui „subsistem inteligent”, destinat compatibilizării dispozitivelor de teren HART cu cerințele standardului IEEE 1451;
5. **Elaborarea unor metode software originale** de modelare și implementare a informațiilor conținute în blocul TEDS;
6. **Utilizarea microcontrolerelor din familia PIC18F44x** ca suport hardware și metodele software de la pct.5, pentru conceperea modulelor TIM;
7. **Studiul soluțiilor actuale de implementare HART** pe o rețea Ethernet;
8. **Conceperea structurii funcționale a unui subsistem inteligent** bazat pe un bloc NCAP și un modul TIM;
9. **Introducerea interfeței GPMI** (General Purpose Multichannel Interface), cu denumirea propusă de către autor;
10. **Elaborarea modelului informațional NCAP** cu ajutorul limbajului UML și pe baza specificațiilor existente în literatura de specialitate;
11. **Identificarea operațiilor necesare implementării comunicației Client – Server** în cadrul subsistemului inteligent;
12. **Definirea modelului funcțional al blocului transductor**, necesar implementării comunicației cu dispozitivele de teren HART la nivelul NCAP;
13. **Propunerea unei metode de înglobare a comenzilor HART** în cadrele Ethernet, care nu periclitează compatibilitatea față de protocoalele Ethernet existente;
14. **Conceperea organigramei** care descrie funcționalitatea subsistemului inteligent la nivel global.

Soluțiile actuale de prelucrare a informației în sistemele de control a proceselor tehnologice au fost prezentate de către autor în lucrarea [44]. Numeroase metode de implementare a anumitor concepte vehiculate pe parcursul capitolului de față au fost valorificate și expuse în cadrul unor lucrări proprii, având referirile [51], [54], [55], [62].

5. IMPLEMENTAREA SUBSISTEMULUI INTELIGENT PRIN UTILIZAREA FPGA

5.1. Considerente de implementare

Capacitatea de reconfigurare simplă și rapidă a FPGA (Field-Programmable Gate Array), a condus pe parcursul anilor, la o largă utilizare de către proiectanți a acestora pentru elaborarea sistemelor prototip. În prezent, circuitele FPGA dispun de resurse suficiente pentru a putea fi competitive la nivelul cerințelor impuse de sistemele de calcul de performanță ridicată, HPC (High Performance Computing) [71].

Utilizarea FPGA în sistemele HPC oferă trei avantaje majore față de dispozitivele bazate pe microprocesoare clasice:

- i) consumul relativ redus de energie;
- ii) accelerarea procesului de prelucrare a informației prin concentrarea pe un singur cip a diverselor blocuri funcționale;
- iii) îmbunătățirea performanțelor sistemului prin axarea directă pe anumite cerințe specifice aplicației.

În prezent, cel mai însemnat producător global de circuite FPGA este compania Xilinx care în anul 1984 a conceput primul FPGA. Acesta conținea 64 de blocuri logice, fiind fabricat în tehnologia de 2 μm [72].

Seria Virtex (producător Xilinx) a fost adoptată pe o scară largă în implementarea aplicațiilor performante care necesită volum relativ ridicat de operații [71], [73].

Momentul cheie în adoptarea soluțiilor FPGA a fost apariția în anul 2002 a seriei Virtex II Pro FPGA, care conține până la o sută de mii de celule logice împreună cu blocuri aritmetice pe 18 x 18 biți și suportă frecvențe de lucru de maximum 400 MHz [71], [72].

Virtex 4, cuprinde până la două sute de mii de celule logice și permite frecvențe de lucru de maxim 500 MHz, fiind adecvat aplicațiilor de control al proceselor tehnologice care necesită diverse blocuri funcționale, inclusiv comunicația pe rețea [72].

Primul circuit FPGA cu o tehnologie de 65 nm este Virtex 5 și prezintă noi optimizări funcționale față de celelalte serii.

Pe parcursul anilor 2000 - 2007 se remarcă o tendință de acumulare a dispozitivelor FPGA, având loc integrarea unor blocuri complexe: microprocesoare și circuite de comunicație rapidă. Tot în această perioadă apare și primul FPGA cu procesor hardware încorporat, bazat pe arhitectura ARM (Advanced RISC Machines) dar și tendința producătorilor de a dezvolta procesoare soft, implementabile eficient în blocurile logice ale FPGA [73].

Familiiile Virtex 6 de la Xilinx și Stratix IV de la Altera, au devenit în prezent platforme sistem, capabile să implementeze pe un singur cip un întreg sistem numeric de control [72], [73].

Implementarea subsistemului inteligent de interfațare HART - Ethernet, al cărui principiu funcțional a fost prezentat în Capitolul IV, va ține cont de tendințele

actuale de cercetare în domeniu și va avea ca suport fizic un dispozitiv FPGA, de tip Virtex 4, produs de către Xilinx.

5.2. Co-proiectarea hardware-software

Subsistemul inteligent va fi implementat bazându-ne pe conceptul de „Solution on a chip” – SoC (soluție pe un singur cip) care va utiliza pe deplin avantajele oferite de circuitele FPGA moderne [74]. În plus, vor fi îmbinate atât structuri hardware cât și software, care au rezultat în urma procesului de co-proiectare hardware-software.

5.2.1 Platforma Xilinx ML-401 cu FPGA Virtex 4

Proiectanții sistemelor de control au posibilitatea de a beneficia de facilitățile și caracteristicile oferite de către circuitul FPGA Virtex 4 prin intermediul platformei sistem ML-401, a cărei diagramă bloc este prezentată în figura 5.1 [75].

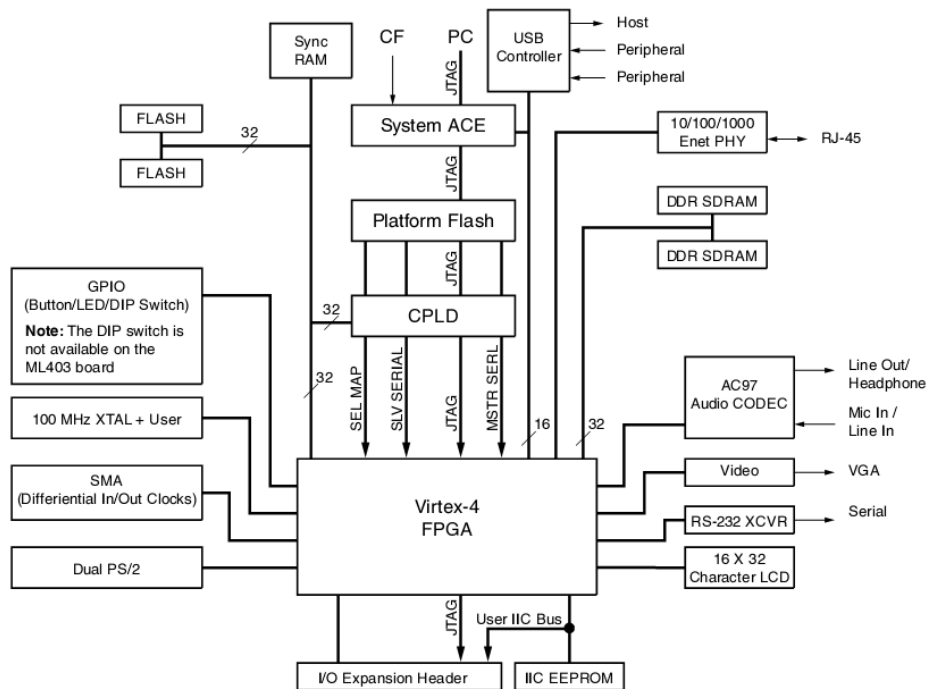


Fig.5.1. Diagrama bloc platforma ML-401 [75]

Sistemul prezentat în figura 5.1 se axează pe circuitul FPGA Virtex 4, XC4VLX25-FF668-10, produs în tehnologia de 90 nm, care permite operarea la frecvențe de tact până la maxim 500 MHz și conține 200000 blocuri logice. Viteza de transfer a datelor pe liniile I/O se ridică la 10 Gb/s. Consumul de energie este redus cu aproape 50% față de seriile anterioare (29 mW la 100 MHz) iar performanțele de

calcul se ridică la 480 DMIPS (Dhrystones Million Instructions Per Second) [72], [73].

Platforma ML-401 prezintă următoarele caracteristici de interes major în implementarea subsistemului inteligent propus în cadrul lucrării:

- SDRAM DDR, 64 MB, cu magistrala pe 32 de biți, cu tact până la 266 MHz;
- SRAM sincron de tip ZBT, 9 Mb, cu magistrala pe 32 biți, 4 biți de paritate;
- Memorie flash, 8 MB;
- Memorie EEPROM I²C, 4 Kb;
- Interfață Ethernet PHY cu trei viteze de operare: 10/100/1000 Mb;
- Interfață USB Cypress CY7C67300;
- Interfață serială RS-232;
- 448 linii de port I/O reconfigurabile;
- Oscilator de tact local, cu frecvența de 100 MHz;
- Intrări de tact diferențiale care acceptă semnal cu frecvența până la valoarea de 500 MHz.

O particularitate importantă este posibilitatea implementării în Virtex 4 a unor procesoare soft care permit rularea codurilor utilizator concepute și elaborate în limbaje de nivel înalt (C/C++). În acest sens, există posibilitatea utilizării procesorului Microblaze, oferit de către compania Xilinx și inclus în mediul de dezvoltare XPS (Xilinx Platform Studio) [75].

5.2.2. Implementarea blocurilor funcționale hardware-software ale subsistemului inteligent (SSIHE)

În Capitolul IV au fost tratate aspectele legate de blocurile funcționale ale subsistemului inteligent de interfațare HART – Ethernet (SSIHE), cu accentuarea algoritmilor și a principiilor care stau la baza modelului informațional global al acestuia.

În cele ce urmează, vom expune în figura 5.2 diagrama bloc aferentă implementării fizice a subsistemului, cu delimitarea modulelor software și hardware.

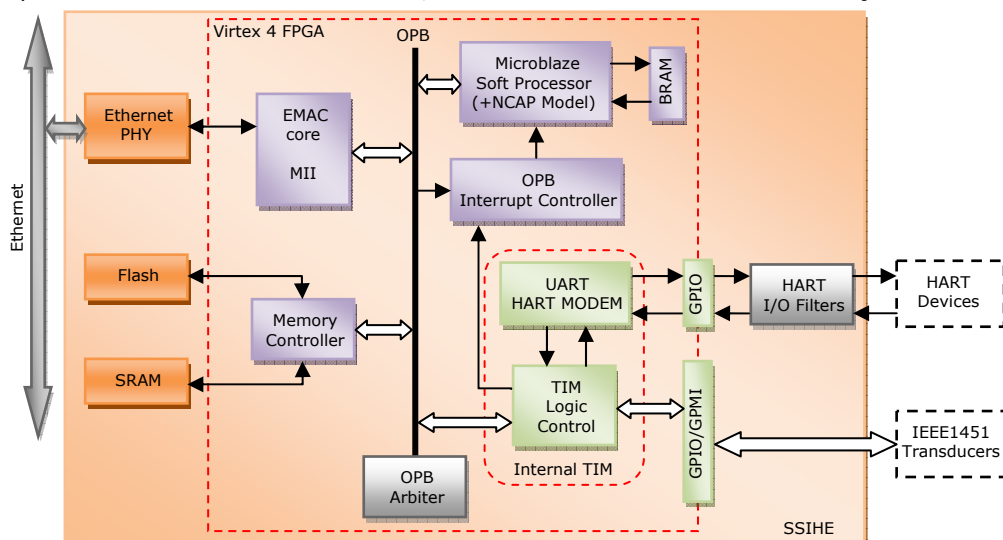


Fig.5.2. Diagrama bloc aferentă implementării fizice a SSIHE

Clasificarea modulelor componente ale structurii prezentate în figura 5.2 se poate realiza conform legendei indicate în figura 5.3.

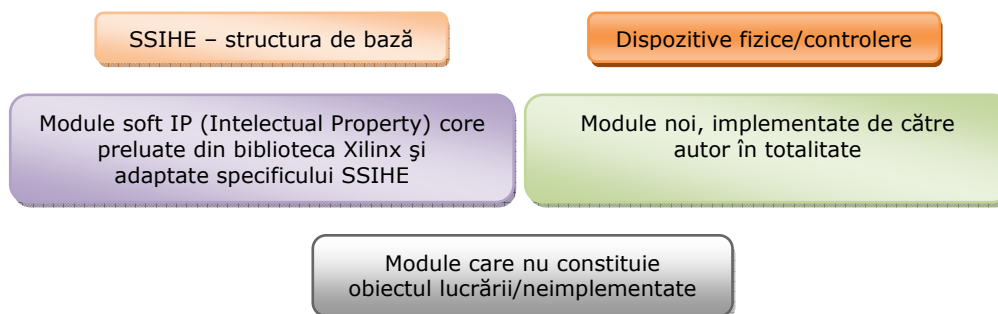


Fig.5.3. Structura SSIHE - legendă

Platforma sistem ML-401, cu structura expusă în figura 5.1, ne permite să implementăm și să testăm funcțional structura de bază propusă pentru SSIHE.

Soluția originală în conceperea SSIHE constă în integrarea pe un singur circuit FPGA (Virtex 4) a modulelor de control (NCAP, TIM) și a circuitelor de interfațare cu transductoarele (HART, IEEE 1451). În acest sens am recurs la integrarea unui procesor Microblaze, pe 32 de biți, oferit de către Xilinx. Avantajul soluției propuse constă în posibilitatea modelării perifericelor incluse în jurul procesorului în funcție de cerințele aplicației. În cazul unui procesor hard, inclus deja pe o platformă sistem, adeseori rămân neutilizate resurse hardware numeroase care ridică doar costurile de fabricație și măresc consumul de energie al întregului sistem.

5.2.2.1 Magistrala internă OPB

Procesorul soft Microblaze va comunica pe magistrala internă OPB (On-Chip Peripheral Bus) – parte componentă a sistemului de magistrale CoreConnect introduse de către IBM și agreeate de către Xilinx în platformele sistem cu FPGA [75]. Magistrala OPB permite interconectarea tuturor perifericelor interne printr-un set de semnale standardizate cu posibilitatea adaptării acestora la specificul aplicației. Totuși, o utilizare în mod direct a acestor semnale implică parcurgerea integrală a protocolului OPB și conceperea modulelor proprii astfel încât să gestioneze întreaga suită de conexiuni și semnale aferente OPB (un număr de peste 50) [76].

În cazul SSIHE, alegem o soluție eficientă de comunicare pe OPB prin includerea unui modul IPIF (Intellectual Property Interface) oferit de către compania Xilinx. Toate operațiile impuse de celelalte periferice integrate vor apela la modulul OPB-IPIF, atașat în mod unic la acestea și configurat în mod corespunzător.

Adoptarea acestei soluții conduce în mod implicit și la reducerea duratei ciclului de proiectare și sinteză a circuitelor pentru perifericele utilizator, în mediul de dezvoltare Xilinx EDK (Embedded Development Kit) fiind furnizate bibliotecile aferente implementării unui modul OPB-IPIF personalizat (adaptat cerințelor proiectării SSIHE).

Având la dispoziție structura generală a modulului OPB-IPIF, proiectantul are posibilitatea de a selecta din aceasta doar serviciile necesare propriei aplicații pentru interfațarea unui anumit periferic la magistrala locală OPB

Structura generală a modulului OPB-IPIF este prezentată în figura 5.4.

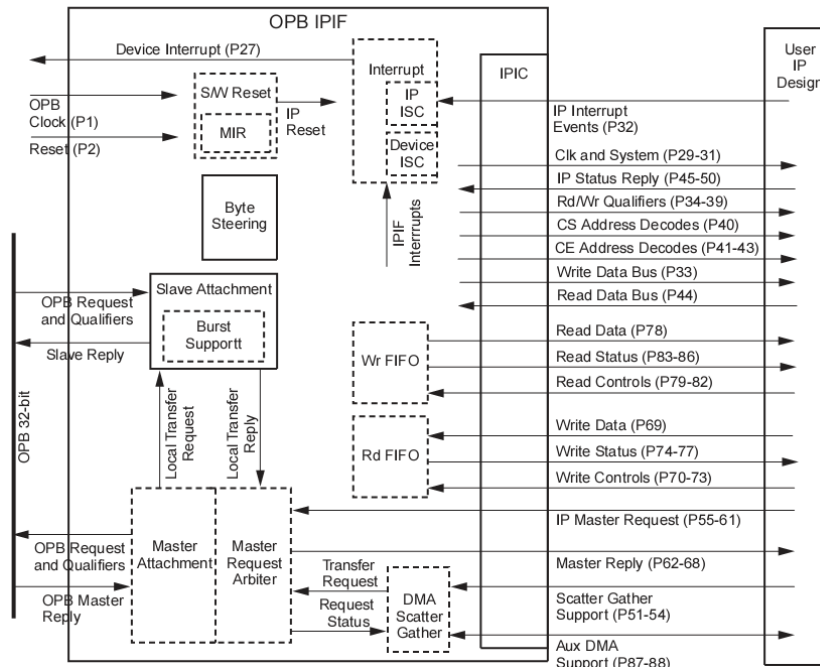


Fig.5.4. Structura modului OPB-IPIF [77]

Blocurile din structura IPIF, marcate cu linie întreruptă, sunt opționale funcție de specificul aplicației utilizator. Comunicarea între modulul IP (Intellectual Property) al utilizatorului și IPIF se realizează efectiv cu ajutorul interfeței IPIC (IP Interconnect). Pentru un dispozitiv Master, proiectantul va selecta doar semnalele aferente interfațării IPIC – Master iar pentru un dispozitiv Slave doar semnalele aferente interfațării IPIC – Slave [77]. Anexa 4 expune semnificația tuturor semnalelor aferente interfeței IPIC.

Pe baza structurii prezentate în figura 5.4, vom selecta blocurile și semnalele necesare pentru fiecare IPIF aferent perifericelor incluse în componența SSIHE.

5.2.2.2 Comunicația pe mediul Ethernet

Controlerul Ethernet PHY, Marvell Alaska 88E1111 permite moduri de operare cu rate de transfer 10/100/1000 Mb/s și va fi utilizat pentru conectarea SSIHE la mediul fizic Ethernet. Comunicația între circuitul FPGA Virtex 4 și Ethernet PHY se va realiza pe liniile aferente MII (Media Independent Interface), conform standardului IEEE 802.3, care asigură rate de transfer de 100 Mb/s la o frecvență a oscilatorului de tact local de 25 MHz. În continuare, blocul „EMAC core” va comunica pe magistrala OPB cu nucleul soft Microblaze prin intermediul unui modul OPB-IPIF specializat cu funcții de control pentru un dispozitiv Slave.

Xilinx furnizează un modul EMAC, parametrizabil de către utilizator, care va fi utilizat pentru implementarea comunicării SSIHE pe rețeaua Ethernet. Interfața IPIF a acestui bloc o vom adapta pentru realizarea unei conexiuni cât mai eficiente

cu nucleul EMAC și pentru a permite adresarea acestuia sub formă de periferic Slave de către procesorul Microblaze.

Structura combinată EMAC - OPB-IPIF, adaptată cerințelor SSIHE, este prezentată în figura 5.5.

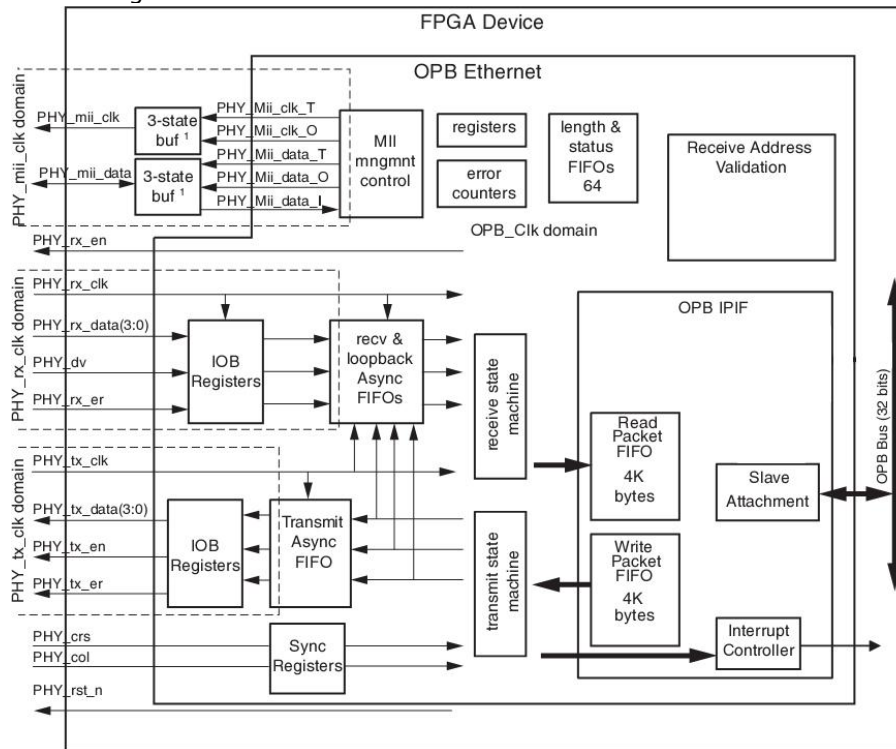


Fig.5.5. Structura EMAC-OPB-IPIF aferentă SSIHE

Interfațarea cu circuitul Ethernet PHY extern se realizează prin semnalele marcate cu inițialele „PHY” iar controlul acestora este efectuat de către blocul „MII mngmnt control” (aferent Media Independent Interface).

Modul de lucru va fi de tip Full Duplex, atribuind valoarea 0 parametrului C_HALF_DUPLEX_EXIST, iar pachetele de date se vehiculează prin registrele „Read Packet FIFO” respectiv „Write Packet FIFO”, care au fost dimensionate la o capacitate de stocare de 4KO. Parametrii care stabilesc mărimea acestor registre sunt C_IPIF_READ_FIFO_DEPTH și C_IPIF_WRITE_FIFO_DEPTH și indică numărul blocurilor BRAM (Block RAM), din interiorul FPGA, alocate pentru stocarea cadrelor Ethernet. O valoare mai mare pentru acești parametri va implica utilizarea a mai multor resurse hardware din FPGA dar în schimb va îmbunătăți consistent rata de transfer pe rețea și va reduce gradul de încărcare al procesorului. Valoarea minimă de 2KO pentru aceste memorii FIFO este suficientă pentru a stoca un cadru Ethernet standard, de 1518 octeți. Totuși, mărirea la 4KO va conduce la posibilitatea memorării a două astfel de cadre și la o îmbunătățire semnificativă a performanțelor sistemului.

Registrele de stare incluse în blocul „length & status FIFOs” memorează informația referitoare la starea transmisiei și lungimea cadrelor Ethernet transmise sau recepționate. Mărimea acestor registre reflectă numărul maxim de intrări din

lista de așteptare și este preferabil să alegem valoarea maximă posibilă, prin atribuirea valorii 64 la parametrul C_MAC_FIFO_DEPTH [77].

Validarea adresei de proveniență a pachetului de date are loc la nivelul blocului „Receive Address Validation” care a fost optimizat pentru reducerea resurselor hardware în urma implementării (parametrul C_CAM_EXIST=0), prin eliminarea modulului CAM (Contents Addressable Memory). Acesta ar permite stocarea a până la 64 adrese MAC ale dispozitivelor conectate pe rețea pentru o validare a accesului pe baza filtrării după aceste adrese. Sistemul SSIHE propus nu necesită o astfel de filtrare.

Conectarea modulului EMAC la magistrala locală OPB se va realiza în modul Slave, prin semnalele gestionate de blocul „Slave Attachment”.

La nivel fizic, modulul EMAC va fi conectat la pinii circuitului Ethernet PHY Marvell Alaska 88E1111 și în continuare la rețeaua Ethernet printr-un conector RJ45.

5.2.2.3 Accesarea memoriilor externe

În scopul implementării blocului TEDS specificat de standardul IEEE 1451, în structura SSIHE am inclus o memorie non-volatilă de tip flash, care este încorporată în platforma sistem ML-401 și are capacitatea 8 MB.

Pentru stocarea unor informații temporare, în general este nevoie și de blocuri de memorie RAM, identificate în SSIHE prin circuitele de memorare ZBT incluse în platforma ML-401, de capacitate 9 Mb.

Accesul la aceste memorii se va realiza cu ajutorul blocului „Memory Controller” care va comunica pe magistrala locală OPB. Implementarea o efectuăm apelând la modulul OPB EMC Memory Controller (External Memory Controller) oferit de către Xilinx și integrat în mod automat în structura finală a FPGA [77]. Structura globală a EMC Xilinx este prezentată în figura 5.6.

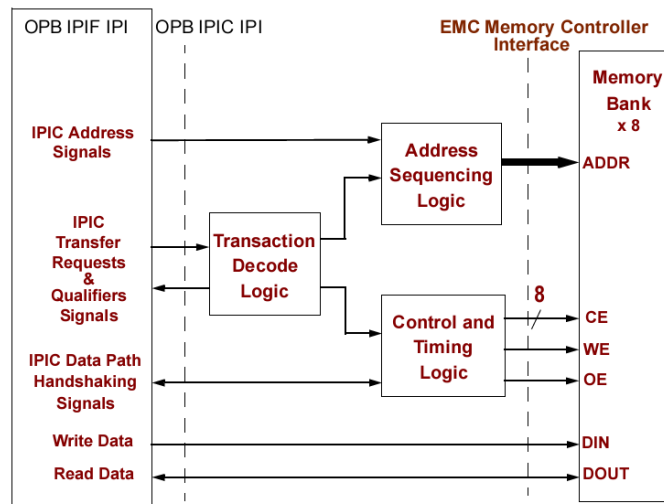


Fig.5.6. Structura globală a controlerului EMC Xilinx [77]

Memoria externă este gestionată sub forma de bancuri continue (maxim 8) și se pot distinge semnalele clasice aferente operării cu memorii SRAM/flash.

5.2.2.4 Gestionarea întreruperilor

Pentru parametrizarea controlerului de întreruperi în interiorul FPGA, trebuie să identificăm în primul rând toate sursele de întrerupere provenite de la blocurile componente ale SSIHE.

Pe baza structurii SSIHE din figura 5.2 și a modului de funcționare a blocurilor incluse deducem existența a două surse de întrerupere interne: EMAC și TIM Control. Definim acest număr prin atribuirea `C_NUM_INTR_INPUTS=2`. Tipul, frontul activ și nivelul se stabilesc prin parametrii `C_KIND_OF_INTR`, `C_KIND_OF_EDGE` respectiv `C_KIND_OF_LVL` [78].

Blocul „OPB Interrupt Controller” va fi implementat apelând la modulul OPB IntC oferit de către Xilinx [77], [78]. Setul de registre pus la dispoziția programatorului este prezentat în tabelul 5.1.

Denumire registru	Referire	Offset
Interrupt Status Register	ISR	00h
Interrupt Pending Register	IPR	04h
Interrupt Enable Register	IER	08h
Interrupt Acknowledge Register	IAR	0Ch
Set Interrupt Enable Bits	SIE	10h
Clear Interrupt Enable Bits	CIE	14h
Interrupt Vector Register	IVR	18h
Master Enable Register	MER	1Ch

Tabelul 5.1. Setul de registre aferent sistemului de întreruperi

Prin citirea sau scrierea valorilor individuale ale biților din registrele expuse în tabelul 5.1, vom avea posibilitatea de a monitoriza și controla sistemul de întreruperi astfel implementat. Pentru o aprofundare a semnificațiilor acestor registre recomandăm parcurgerea specificațiilor tehnice oferite de către Xilinx, cu referire la [77], [78].

Organizarea internă a modului IntC este indicată în figura 5.7.

Toate registrele aferente IntC sunt accesate prin intermediul interfeței OPB/DCR. Adresele de bază ale acestor registre sunt furnizate de către un parametru de configurare. Față de această adresă există o deplasare (Offset), așa cum rezultă din tabelul 5.1 cu scopul de a respecta convenția de localizare a registrelor în interfețele standard OPB-IPIF [76], [77].

Deoarece adresele tip OPB sunt reprezentate la nivel de octet, deplasările față de adresa de bază vor fi localizate la multiplii de 4.

Modulul IntC va fi configurat ca un periferic OPB, pe 32 de biți, conform dimensiunii standard a magistralei interne OPB.

Semnala cerere de întrerupere, furnizat la ieșirea unică Irq va fi generat pe baza operațiilor logice SAU efectuate asupra tuturor surselor de întrerupere (în cazul nostru EMAC și TIM Control), la nivelul blocului funcțional IRQ Generation.

Ieșirea Irq, precum și intrările (Int_inputs) pentru sursele de întrerupere pot fi activate sau inactivate (mascate) în mod individual.

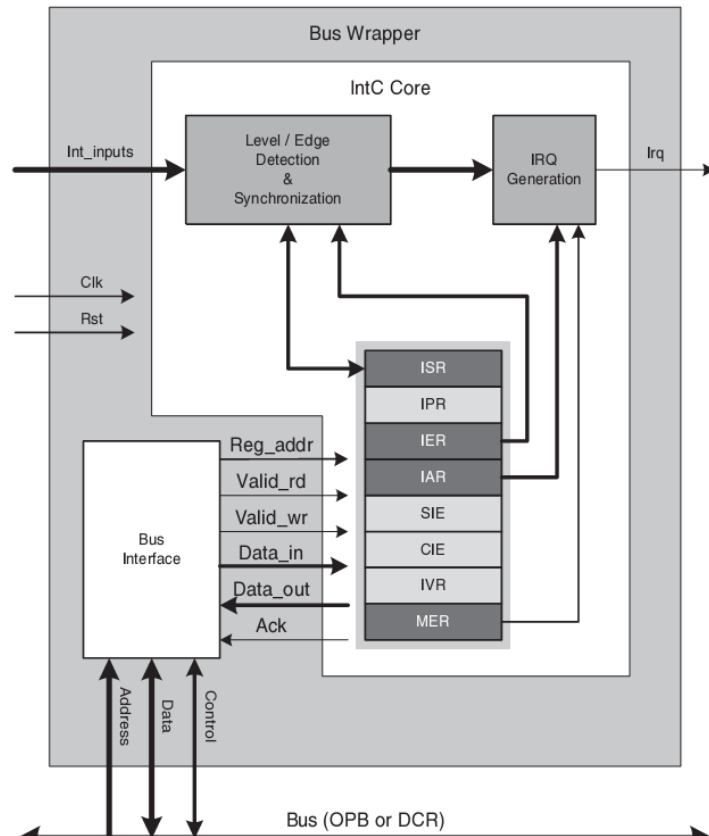


Fig.5.7. Controlerul de întreruperi OPB Interrupt Controller [78]

Prezența registrelor IPR, SIE, CIE și IVR se poate indica prin atribuirea valorii 1 parametrilor C_HAS_IPR, C_HAS_SIE, C_HAS_CIE respectiv C_HAS_IVR. Această opțiune este utilă pentru reducerea resurselor hardware necesare implementării fizice a controlerului IntC în cazul în care anumite registre nu sunt utilizate.

Prin citirea biților registrului ISR vom putea identifica proveniența cererii de întrerupere din sistem.

5.2.2.5 Arbitrarea magistralei OPB

În cazul existenței a mai multe dispozitive Master, conectate pe aceeași magistrală locală OPB, va fi necesară implementarea unui bloc de arbitrare, conform specificațiilor IBM pentru CoreConnect [76] și Xilinx [79] pentru platforma sistem ML-401. Pentru structura SSIHE, în cadrul sistemului elaborat, având doar un singur Master (procesorul Microblaze), nu vom recurge la includerea blocului de arbitrare. Acesta este doar evidențiat în diagrama bloc din figura 5.2 și se poate implementa în dezvoltări ulterioare.

5.2.2.6 Modulul TIM intern (IPTIM)

Soluția novatoare propusă în conceperea SSIHE se referă și la integrarea modulului TIM, împreună cu blocul UART și modem-ul HART, în structura internă a FPGA Virtex 4. Modulul „Internal TIM” va comunica prin magistrala locală OPB cu procesorul Microblaze (nucleul propriu-zis al NCAP). Semnalele interfeței clasice TII, dintre NCAP și TIM, vor fi înlocuite cu semnalele specifice OPB. Mai precis, vom structura un modul IPIF aferent conectării TIM la OPB și vom implementa porturile I/O necesare conectării dispozitivelor de teren HART și a transductoarelor la SSIHE.

Structura generală a modulului TIM intern (denumit IPTIM) este prezentată în figura 5.8.

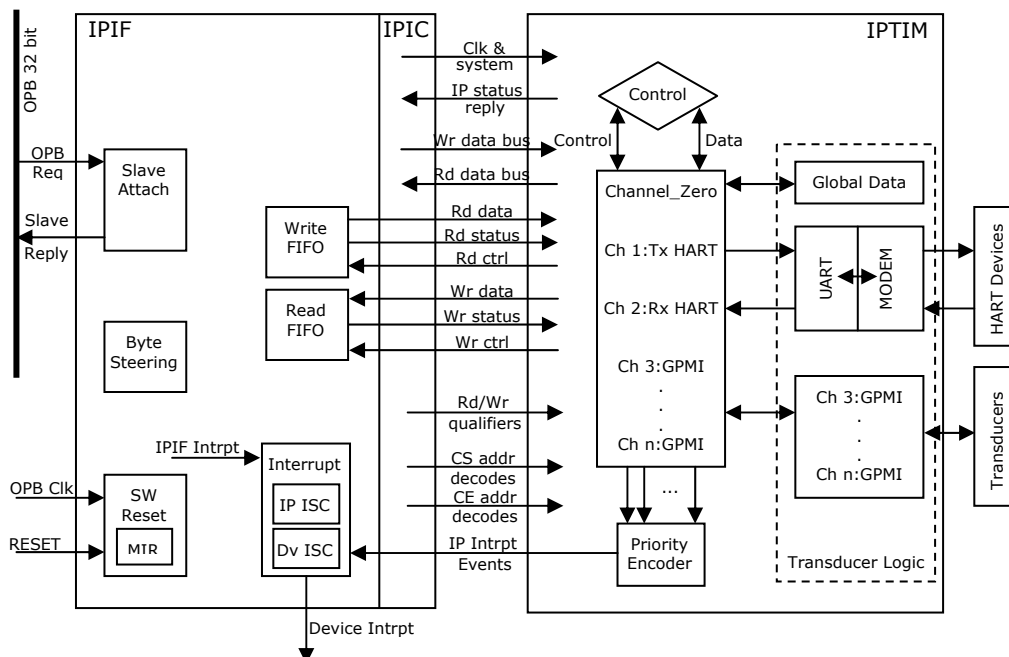


Fig.5.8. Modulul TIM intern cu interfața OPB IPIF

Modulul TIM intern va fi interfațat pe magistrala OPB prin intermediul blocului Slave attach din componența OPB IPIF. Conexiunile aferente IPIC, evidențiate în figura 5.8, înlocuiesc semnalele clasice impuse de către interfața TII. Astfel, modulul TIM va avea posibilitatea transferului de date direct pe OPB, în mod paralel, pe 32 de biți, mărind în acest fel ratele de transfer în mod considerabil. Registrele intermediare Write FIFO și Read FIFO vor stoca temporar datele transmise sau recepționate de către TIM.

Gestionarea sistemului global de întreruperi generate de către TIM va fi realizat la nivelul OPB IPIF, prin intermediul blocului Interrupt format din două componente ISC: Dv ISC (Device Interrupt Service Control) și IP ISC (Intellectual Property ISC). În acest sens sunt comparate semnalele IPIF Intrpt (de proveniență internă IPIF) și IP Intrpt Events (de proveniență TIM) și este generat semnalul compus Device Intrpt furnizat mai departe modulului „OPB Interrupt Controller” (figura 5.2).

La nivelul TIM, cererile locale de întrerupere vor fi gestionate în mod prioritar de către blocul Priority Encoder, bazat pe o structură logică specifică.

Semnalele aferente interfeței IPIC sunt preluate de către blocul Control din cadrul TIM, la nivelul căruia are loc separarea informației în date și control. Acest bloc prezintă de fapt două interfețe: la nivelul IPIC și la nivelul canalelor transductor (incluzând liniile de date și de control). La nivelul IPIC avem semnalele expuse în figura 5.8, definite conform specificațiilor Xilinx pentru conexiunile magistralei OPB, acestea fiind implementate în limbajul VHDL și respectă diagramele temporale impuse de tranzacțiile pe OPB [76], [77]. În schimb, conexiunile aferente canalelor transductor, în afară de cele aferente comunicației cu dispozitivele de teren HART, vor fi definite în funcție de specificul aplicațiilor ulterioare, de la caz la caz. Principalele operații care trebuie asigurate sunt cele de citire/scriere a valorilor stocate în fiecare registru asociat unui canal transductor.

Blocul canalelor transductor conține un set de registre, câte trei pentru fiecare canal după cum urmează: registrul de date, registrul de stare și registrul de întrerupere. În plus, în cadrul acestui bloc vom genera semnalele de întrerupere furnizate către modulul Priority Encoder. Aceste semnale se generează prin efectuarea operației ȘI logic între conținutul registrelor de stare și de întrerupere.

Canalul 1 și Canalul 2 sunt rezervate pentru implementarea transferului de informații între TIM și dispozitivele de teren HART. Aici intervine modulul combinat UART/HART Modem, care se poate distinge în structura prezentată în figura 5.8.

Canalele 3,...,n sunt puse la dispoziția proiectantului pentru implementarea a maxim (n-3) conexiuni aferente unor transductoare de uz general, compatibile IEEE 1451. Acest număr este corelat cu numărul liniilor I/O disponibile ale circuitului FPGA (în cazul nostru Virtex 4 pe platforma sistem ML-401).

Prin accesarea canalului Channel_Zero, se va permite o adresare globală a tuturor transductoarelor conectate la sistem, așa cum prevede standardul IEEE 1451 [61].

Identificarea modului de către procesor se va realiza prin citirea valorilor din registrul MIR (Module Identification Register) inclus în blocul SW Reset. La acest nivel se preiau și semnalele RESET și OPB Clk.

Tranzacțiile între blocul IPTIM și magistrala OPB se vor realiza pe baza diagramelor temporale impuse de protocolul OPB, prezentate în figura 5.9 (citire Slave) și 5.10 (înscriere Slave).

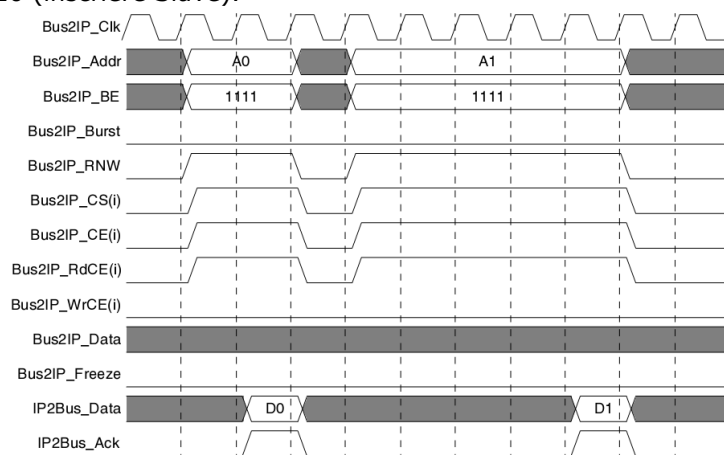


Fig.5.9. Ciclul de citire Slave pentru IPTIM [77]

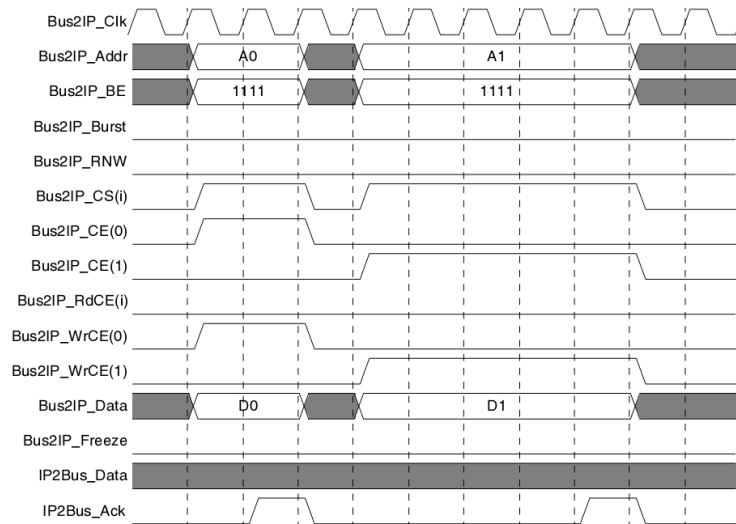


Fig.5.10. Ciclul de înscriere Slave pentru IPTIM [77]

Se recomandă consultarea semnificațiilor semnalelor OPB IPIF/IPIC expuse în Anexa 4.

Nucleul NCAP, implementat prin procesorul Microblaze, va putea interoga IPTIM cu ajutorul semnalelor OPB Req (Requests). Citirea datelor provenite de la un canal transductor va fi realizată prin magistrala OPB, preluând informația de pe liniile IP2Bus_Data <0:31>, iar transmiterea datelor către un canal transductor prin liniile Bus2IP_Data <0:31>. Direcția transferului de date va fi stabilită de valoarea logică a semnalului Bus2IP_RNW, așa cum este indicat și în figurile 5.9 și 5.10.

Vom implementa în cadrul SSIHE comenzile obligatorii impuse de către standardul IEEE 1451: Citire/Scriere registru de întreruperi, Citire registru de stare, Interogare individuală/globală, Citire/Scriere canal transductor și Reset TIM. În acest sens, vom atribui adrese pe 16 biți acestor comenzi, având un offset al adresei pentru fiecare canal transductor în parte.

Etapa următoare constă în definirea structurii unității de control (Control) al IPTIM. Luăm în considerare necesitatea existenței a trei canale: Channel_Zero, Channel 1 (Tx HART) și Channel 2 (Rx HART). Suplimentar, pentru exemplificarea funcționalității sistemului și a modului de definire a canalelor transductor suplimentare (non HART) vom include un canal de date (Channel 3) pentru controlul numeric al unui transductor extern, sub forma unei punți H-MOSFET prin care apoi se pot genera diverse valori de tensiune de comandă, așa cum este propus de către autor și în lucrarea „**Industrial Smart Power Supply**” [80] printr-o structură clasică NCAP – TIM. Dezvoltarea în lucrarea de față a conceptului indicat în [80] presupune eliminarea modului TIM extern și interfațarea driverelor punții H în mod direct la liniile GPMI ale SSIHE.

Expunem în continuare modurile de operare ale unității de control și semnalele vehiculate de către aceasta, pentru cele patru cazuri specifice funcționării IPTIM.

Controlul registrelor de întrerupere și stare, pentru fiecare canal transductor în parte, presupune generarea de către blocul Control al unui semnal de validare al canalului precum și existența liniilor de transmitere a valorii efective a datelor stocate în acel registru, pe 32 de biți.

Figura 5.11 exemplifică modul de implementare al controlului întreruperilor și a registrelor de stare pentru canalele transductor din componența IPTIM.

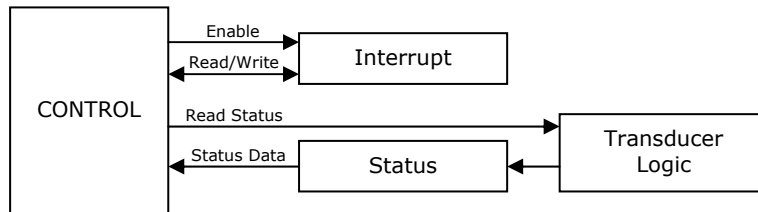


Fig.5.11 Controlul registrelor de întrerupere și stare al IPTIM

Interogarea/Declanșarea (Trigger) unui canal transductor, cu scopul de a prelua valorile transmise de către senzori sau de a trimite comenzi către un actuator, va fi realizată prin conectarea blocului Control în mod direct la circuitele modului Transducer Logic, așa cum este indicat în figura 5.12.

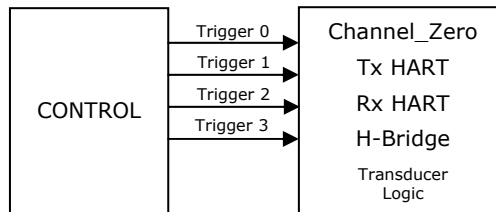


Fig.5.12. Controlul operației Trigger

Gestioarea datelor (citire/scriere) aferente canalelor transductor se realizează tot de către blocul Control, prin semnalele de validare Enable și cele 32 de linii de date (Data Read sau Data Write). Validarea se transmite direct la modulul Transducer Logic iar datele se citesc sau se înscriu prin registrele dedicate canalelor transductor 0, 1, 2 sau 3, modul de implementare fiind indicat în figura 5.13.

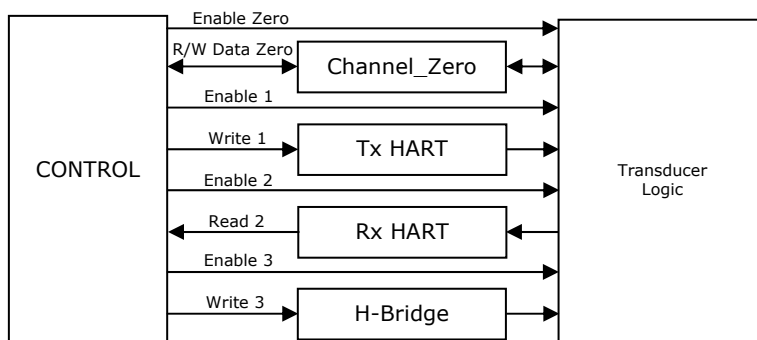


Fig.5.13. Gestionarea datelor aferente canalelor transductor

Comanda Reset este necesară pentru a readuce conținutul tuturor registrelor atribuite canalelor transductor la valorile inițiale (cele implicite la momentul conectării tensiunii de alimentare). Se aplică simultan la toate canalele transductor implementate și este generat de către blocul Control din cadrul structurii IPTIM.

Figura 5.14 prezintă modul de implementare al comenzii Reset.

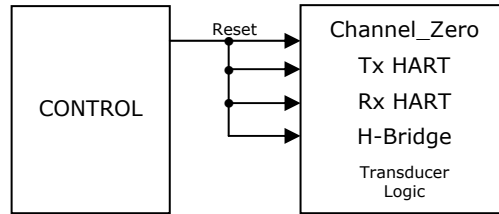


Fig.5.14 Generarea semnalului Reset

Blocul canalelor transductor va fi structurat astfel încât să existe posibilitatea conectării atât la semnalele de control cât și la logica de interfațare a transductorilor externe. În plus, fiecare canal transductor va genera semnalele de întrerupere, valorile logice fiind stocate în registrul de întrerupere, pe 32 de biți. Pentru Channel 1 (Tx HART) și Channel 2 (Rx HART) vom implementa canalul transductor sub forma unui registru de date pe 32 de biți, a unui registru de stare pe 32 de biți și a unui registru de întrerupere pe 32 de biți. Registrul de date va fi accesat în mod secvențial cu extragerea unor pachete de date pe 8 biți, datorită interfațării cu circuitul UART. Este necesară astfel multiplexarea informației provenite de la registrul de date, existând patru cicluri de citire a câte 8 biți. Ieșirea multiplexorului va furniza datele efective pentru modulul UART în cazul canalului Tx HART și datele de intrare pentru registrul canalului Rx HART, cu selecția prin RxAv.

Channel 3 va dispune de o structură simplificată, datele fiind unidirecționale (înscrisere în registrul de date).

Valoarea pe 32 de biți obținută prin efectuarea la nivel de bit a operației ȘI logic între registrele stare și de întrerupere va constitui parametrul de intrare în blocul Priority Encoder (pentru fiecare canal în parte).

Structura globală propusă pentru canalul Tx HART și configurația blocului Transducer Logic aferentă acestui canal este prezentată în figura 5.15.

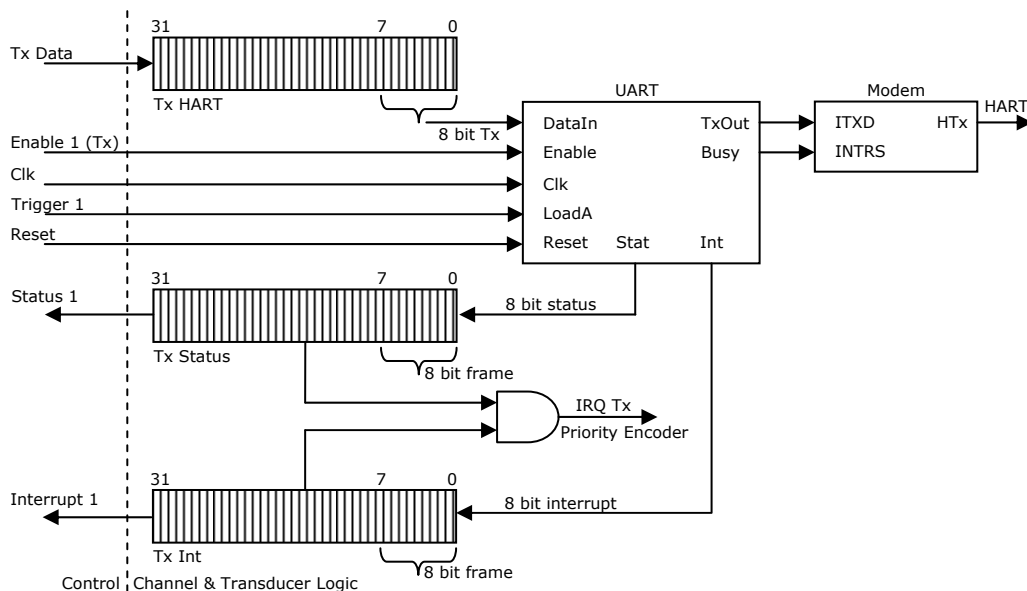


Fig.5.15. Structura canalului Tx HART și logica de control aferentă

Similar structurii prezentate în figura 5.15, vom propune în cazul canalului Rx HART configurația expusă în figura 5.16.

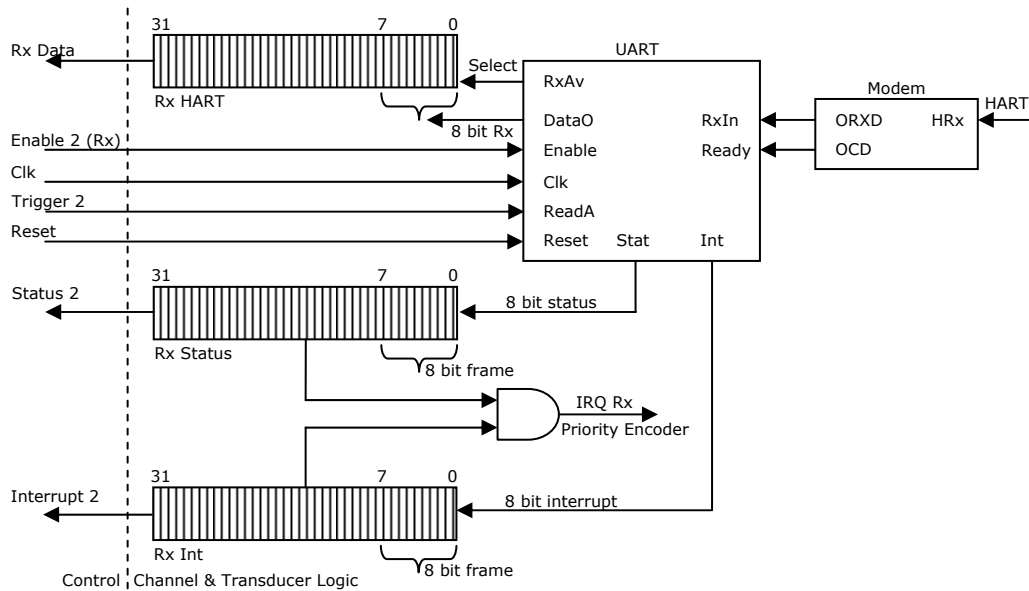


Fig.5.16. Structura canalului Rx HART și logica de control aferentă

Distingem în ambele cazuri semnalele aferente blocului Control, modul de preluare a datelor din cele trei registre asociate canalelor precum și modul de generare al întreruperilor furnizate către blocul Priority Encoder.

Canalul 3 necesită o structură relativ simplă față de celelalte două canale anterioare, având în componența sa cele trei registre: de date, de stare și de întrerupere și o logică simplă de validare și transfer a datelor pe 4 biți la patru linii de port externe. Vom păstra registrul de stare și de întrerupere pentru dezvoltări ulterioare. Structura canalului Channel 3 (H-Bridge) este indicată în figura 5.17.

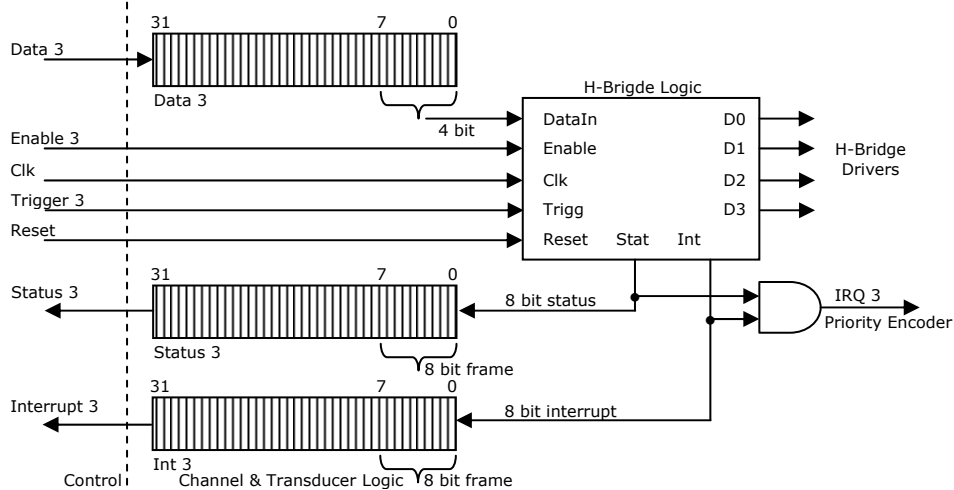


Fig.5.17. Structura canalului H-Bridge și logica de control aferentă

5.2.2.7 Blocul UART din componența IPTIM

Vom parcurge în continuare etapele elaborării structurii funcționale ale blocului UART. În acest sens trebuie definite porturile și semnalele de intrare/ieșire, conform simbolului din figura 5.18.

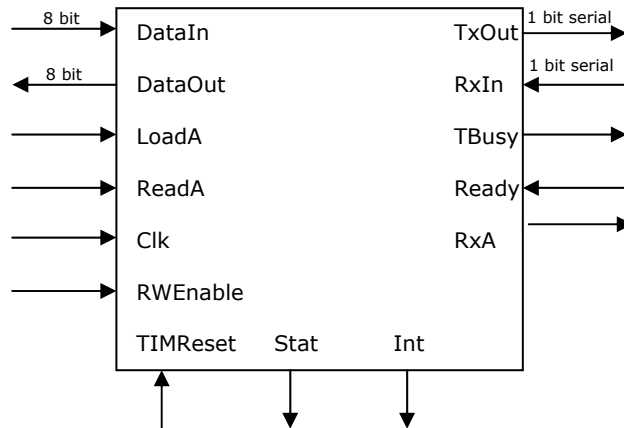


Fig.5.18. Liniile I/O aferente blocului UART

Componențele interne ale blocului UART vor asigura transmiterea serială a valorii pe 8 biți, preluate prin liniile DataIn, către linia de ieșire TxOut, recepția serială pe linia de intrare RxIn și furnizarea valorii pe 8 biți pe liniile DataOut precum și generarea biților de stare și de întrerupere (prin registrele asociate Stat și Int).

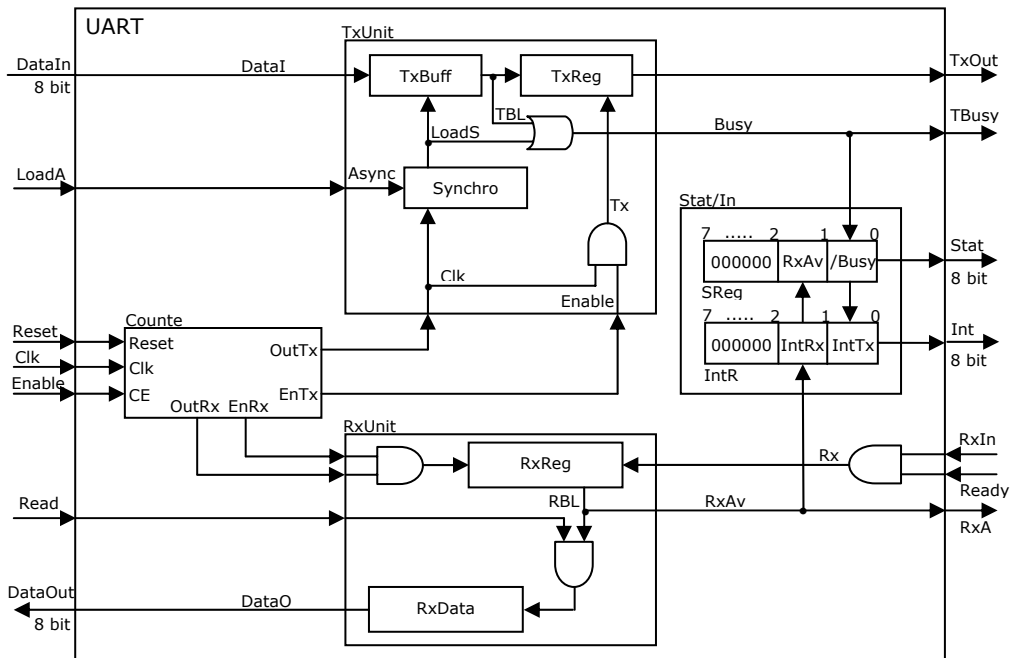


Fig.5.19 Structura internă a blocului UART

Structura internă propusă pentru modulul UART este prezentată în figura 5.19. Distingem modulul de transmisie TxUnit, cel de recepție RxUnit, un numărător/divizor al impulsurilor de tact provenite de la logica OPB, registrele de stare Stat/Int precum și logica de control auxiliară pentru anumite semnale de intrare/ieșire.

O etapă esențială în proiectarea blocului UART este stabilirea ratei de divizare pentru obținerea transmisiei/recepției seriale la 1200 bit/s impusă de către dispozitivele de teren HART. În acest sens vom stabili în continuare valoarea registrului de numărare, BaudCount, care va determina factorul de divizare al semnalului de tact, conform relației:

$$\text{BaudCount} = \text{FPGA_Clk} / \text{HART_Baud} \quad (5.1)$$

unde FPGA_Clk reprezintă frecvența semnalului de tact aferent magistralei OPB (implicit și al sistemului global) iar HART_Baud rata de transmisie impusă de către standardul HART. Pentru modulația FSK (standardul HART), un bit/s reprezintă un baud, astfel HART_Baud va avea valoarea 1200. Platforma sistem ML-401 dispune de un oscilator de tact la frecvența de 100 MHz. În aceste condiții obținem:

$$\text{BaudCount} = 10^8 / 1200 = 83333,33 \quad (5.2)$$

Putem considera partea întreagă a acestei valori, stabilind BaudCount la valoarea 83333 ceea ce va conduce la obținerea ratei de 1200,004 b/s care se încadrează în cerințele protocolului de comunicație HART.

Modulul Counter va furniza semnalul intern de tact pentru serializarea conținutului registrului de transmisie TxReg precum și pentru încărcarea serială a registrului RxReg. În plus, generează semnalele de validare ale modulelor TxUnit și RxUnit.

Transmiterea datelor de către TxUnit se realizează în mai multe etape. Inițial, la activarea liniei LoadA, în cadrul blocului Synchro, se declanșează procesul de sincronizare cu semnalul de tact Clk și se generează impulsul LoadS, de încărcare a celor 8 biți de date de pe DataIn în registrul TxBuff. La încheierea acestui proces, se furnizează un „1” logic la ieșirea TBL (TxBuff Loaded) pe baza căruia se încarcă registrul de transmisie serială TxReg. Pe parcursul încărcării și a procesului de transmisie, semnalul Busy va fi menținut activ, pe „1” logic. Serializarea datelor stocate în TxReg se va realiza pe baza impulsurilor Tx obținute din semnalul de tact divizat, OutTx și validate prin ieșirea EnTx.

Recepția serială, corelată cu detectarea purtătoarei de către modemul HART (ieșirea OCD a acestuia) prin intrarea Ready, se bazează pe stocarea biților de date în registrul RxReg. Încărcarea acestuia are loc în mod secvențial, tactul fiind furnizat de către ieșirea OutRx a blocului Counter. Semnalul RBL (RxReg Buffer Loaded) indică încheierea ciclului de încărcare a RxReg precum și disponibilitatea datelor. Citirea datelor din RxData are loc în momentul activării liniei ReadA, cu testarea bitului RBL (prin operația ȘI logic).

Completarea operațiilor de transmisie/recepție serială va fi monitorizată prin registrul de stare SReg și cel de întreruperi IntR, pe 8 biți fiecare. Vom utiliza doar primii doi biți ai fiecărui registru, următorii 6 fiind rezervați pentru dezvoltări ulterioare. Pe durata transmisiei, bitul SReg.0 va avea valoarea logică „0” (/Busy) iar la încheierea transmisiei, bitul IntR.0 va trece pe „1” logic. În cazul încheierii

recepției, bitul SReg.1 va deveni „1” și bitul IntR.1 va deveni „1”. Aceste registre sunt puse la dispoziția blocului de control al IPTIM prin liniile Stat și Int (pe 8 biți).

Modelarea blocului UART s-a realizat în limbajul VHDL.

```
entity UART is
  generic(BaudCount: INTEGER range 0 to 128000 := 83333); -- divizor tact
  port (
    -- semnale globale si TIM Control
    TIM_Reset : in std_logic; -- Reset
    DataIn : in std_logic_vector(7 downto 0); -- date de intrare
    DataOut : out std_logic_vector(7 downto 0); -- date de iesire
    RWEnable : in std_logic; -- Enable operare UART
    Ready : in std_logic; -- activare Rx
    RxA : out std_logic; -- octet disponibil
    TBusy : out std_logic;
    Int : out std_logic_vector(7 downto 0); -- registrul de intrerupere;
    Stat : out std_logic_vector(7 downto 0); -- registrul de stare;
    Clk_In : in std_logic; -- tact sistem (de la TIM)
    TxOut : out std_logic; -- iesirea seriala
    RxIn : in std_logic; -- intrarea seriala
  );
end UART;
architecture Behaviour of UART is
  component Counter
    generic(COUNT: INTEGER range 0 to 128000); -- contor baud
    port (
      Clk : in std_logic; -- Clock
      Reset : in std_logic; -- Reset
      CE : in std_logic; -- Chip Enable
      O : out std_logic; -- Output
    );
  end component;
  component RxUnit
    port (
      Clk : in std_logic; -- tact sistem
      Reset : in std_logic; -- Reset
      Enable : in std_logic; -- Enable
      ReadA : in std_logic; -- comanda asincrona citire octet
      RxD : in std_logic; -- intrare seriala
      Rdy : in std_logic; -- intrare detectie OCD
      RxAv : out std_logic; -- octet disponibil
      DataO : out std_logic_vector(7 downto 0); -- octet receptionat
    );
  end component;
  component TxUnit
    port (
      Clk : in std_logic; -- tact sistem
      Reset : in std_logic; -- Reset
      Enable : in std_logic; -- Enable
      LoadA : in std_logic; -- comanda asincrona incarcare octet
      TxD : out std_logic; -- iesire seriala
      Busy : out std_logic; -- fanion transmitere (Busy)
      DataI : in std_logic_vector(7 downto 0); -- octet de transmis
    );
  end component;
  signal RxData : std_logic_vector(7 downto 0); -- octetul receptionat
  signal TxData : std_logic_vector(7 downto 0); -- octetul de transmis
  signal SReg : std_logic_vector(7 downto 0); -- registrul de stare
  signal IntR : std_logic_vector(7 downto 0); -- registrul de intrerupere
  signal EnRx : std_logic; -- Enable RXUnit
  signal EnTx : std_logic; -- Enable TXUnit
  signal RxAv : std_logic; -- fanion date receptionate
  signal TxBusy : std_logic; -- fanion date in curs de transmisie
  signal ReadA : std_logic; -- comanda citire octet
  signal LoadA : std_logic; -- comanda incarcare octet
  signal Sig0 : std_logic; -- gnd
  signal Sig1 : std_logic; -- vcc
end architecture;
```

Fig.5.20. Definirea blocurilor și semnalelor aferente UART

Modulele interne și semnalele de intrare/ieșire și interne aferente blocurilor componente ale UART sunt definite conform secvenței de cod prezentate în figura 5.20.

Modulul TxUnit va opera conform secvenței de cod VHDL indicate în figura 5.21, conținând și blocul de sincronizare (componenta Synchro) care este definită la acest nivel (în TxUnit).

```

architecture Behaviour of TxUnit is
    component synchro
    port (
        C1 : in std_logic; -- comanda asincrona
        C : in std_logic; -- tact sistem
        O : out Std_logic);-- tact sincronizat
    end component;

    signal TxBuff : std_logic_vector(7 downto 0); -- registrul de incarcare
    signal TxReg : std_logic_vector(7 downto 0); -- registrul de transmisie
    signal TBL : std_logic; -- fanion incarcare
    signal LoadS : std_logic; -- comanda incarcare sincronizata

begin
    -- sincronizarea cu tactul sistem
    SyncLoad : Synchro port map (LoadA, Clk, LoadS);
    Busy <= LoadS or TBL;

    -- procesul de transmisie seriala
    TxProc : process(Clk, Reset, Enable, DataI, TxBuff, TxReg, TBL)
    variable BitPosition : INTEGER range 0 to 10; -- pozitia bitului in pachet
    begin
        if Reset = '1' then
            TBL <= '0';
            BitPosition := 0;
            TxD <= '1';
        elsif Rising_Edge(Clk) then
            if LoadS = '1' then
                TxBuff <= DataI;
                TBL <= '1';
            end if;
            if Enable = '1' then
                case BitPosition is
                    when 0 => -- bitul de stop sau idle
                        TxD <= '1';
                        if TBL = '1' then -- inceput transmisie
                            TxReg <= TxBuff;
                            TBL <= '0';
                            BitPosition := 1;
                        end if;
                    when 1 => -- bitul de start
                        TxD <= '0';
                        BitPosition := 2;
                    when others =>
                        TxD <= TxReg(BitPosition-2); -- secventa seriala de iesire
                        BitPosition := BitPosition + 1;
                    end case;
                if BitPosition = 10 then -- testare ultimul bit din pachet
                    BitPosition := 0;
                end if;
            end if;
        end process;
    end Behaviour;

```

Fig.5.21. Secvența VHDL pentru operarea modulului TxUnit

Procesul de sincronizare ține cont de semnalul de încărcare asincron, LoadA, care este asociat cu Trigger1 furnizat de către TIM Control și cu momentul apariției impulsului de tact (Clk) imediat următor. La ieșirea modulului Synchro vom obține impulsurile de încărcare sincronă, LoadS.

Codul VHDL aferent implementării proceselor asociate modulului Synchro este prezentat în figura 5.22.

```

architecture Behaviour of synchro is
  signal AsyncL : std_logic;
  signal SysClk : std_logic;
  signal SyncL : std_logic;
begin
  RiseAsyncL : process(C1,SyncL)
  begin
    if Rising_Edge(C1) then
      AsyncL <= '1';
    end if;
    if (SyncL = '1') then
      AsyncL <= '0';
    end if;
  end process;

  SyncP : process(C,SyncL)
  begin
    if Rising_Edge(C) then
      if (AsyncL = '1') then
        SysClk <= '1';
      else SysClk <= '0';
      end if;
      if (SysClk = '1') then
        SyncL <= '1';
      else SyncL <= '0';
      end if;
    end if;
    if (SyncL = '1') then
      SysClk <= '0';
    end if;
  end process;
  O <= SysClk;
end Behaviour;

```

Fig.5.22. Secvența VHDL aferentă operării modulului Synchro

Recepția serială, asigurată de modulul RxUnit, presupune eșantionarea liniei RxIn în momentul în care linia Ready este adusă la nivelul „1” logic de către ieșirea OCD (Carrier Detect Output) a modemului HART. OCD este menținut pe „1” logic atâta timp cât există transmisie prin modulație FSK pe liniile dedicate dispozitivelor de teren HART. Semnalul efectiv, purtător de informație, va fi furnizat la ieșirea porții ȘI din structura UART prezentată în figura 5.19 (denumit Rx).

În scopul asigurării unei rate de eșantionare adecvate, în general, dispozitivele UART utilizează circuite de tip PLL (Phase-Locked Loop) [81]. Pentru eliminarea blocului PLL am recurs la implementarea unui numărator de eșantioane care va detecta și contoriza tranzițiile succesive în semnalul RxIn pe parcursul a patru stări (reprezintă aproximativ 50% din lungimea totală a pachetului), durată suficient de mare pentru determinarea momentelor în care se prelevă biții de pe linia RxIn. Variabila locală „SmpCnt” va contoriza aceste tranziții.

Fanionul RBL indică încărcarea completă a registrului tampon RxReg și permite în continuare transferul datelor în registrul RxData, la activarea liniei ReadA

asociată cu semnalul Trigger2 furnizat de către IPTIM. Pasul final constă în obținerea pachetului de date, pe 8 biți, la liniile DataOut ale blocului UART.

Secvența VHDL prin care am implementat recepția serială este indicată în figura 5.23.

```

architecture Behaviour of RxUnit is
  signal RxReg  : std_logic_vector(7 downto 0); -- registrul de receptie seriala
  signal RBL   : std_logic;                -- fanion receptie octet
begin
  -- proces octet disponibil
  RxAvProc : process(RBL,Reset,ReadA)
  begin
    if ReadA = '1' or Reset = '1' then
      RxAv <= '0'; -- RxReg a fost citit
    elsif Rising_Edge(RBL) then
      RxAv <= '1'; -- RxReg a fost transferat in RxData
    end if;
  end process;

  -- procesul de receptie seriala
  RxProc : process(Clk,Reset,Enable,RxD,Rdy,RxReg)
  variable BitPos : INTEGER range 0 to 10; -- pozitia bitului in pachet
  variable SmpCnt : INTEGER range 0 to 3; -- esantionare patru tranzitii
  begin
    if Reset = '1' then -- Reset
      RBL <= '0';
      BitPos := 0;
    elsif Rising_Edge(Clk) and Rdy='1' then
      if Enable = '1' then
        case BitPos is
          when 0 => -- idle
            RBL <= '0';
            if RxD = '0' then -- bit de start
              SmpCnt := 0;
              BitPos := 1;
            end if;
          when 10 => -- bit de stop
            BitPos := 0;
            RBL <= '1'; -- octet receptionat
            DataO <= RxReg; -- incarcare octet receptionat
          when others =>
            if (SmpCnt = 1 and BitPos >= 2) then -- contorizare tranzitii "0->1" pe RxD
              RxReg(BitPos-2) <= RxD; -- incarcare secventiala in RxReg
            end if;
            if SmpCnt = 3 then
              BitPos := BitPos + 1;
            end if;
          end case;
          if SmpCnt = 3 then
            SmpCnt := 0;
          else
            SmpCnt := SmpCnt + 1;
          end if;
        end if;
      end if;
    end process;
  end Behaviour;

```

Fig.5.23. Secvența VHDL pentru operarea modulului RxUnit

Divizarea frecvenței semnalului de tact Clk al sistemului se realizează prin intermediul blocului Counter, pe baza relației (5.2) și a considerențelor expuse apriori. Secvența de cod VHDL prin care se implementează operarea blocului Counter este prezentată în figura 5.24.

```
architecture Behaviour of Counter is
begin
  counter : process(Clk,Reset)
    variable Cnt : INTEGER range 0 to Count-1;
  begin
    if Reset = '1' then
      Cnt := Count - 1;
      O <= '0';
    elsif Rising_Edge(Clk) then
      if CE = '1' then
        if Cnt = 0 then
          O <= '1';
          Cnt := Count - 1;
        else
          O <= '0';
          Cnt := Cnt - 1;
        end if;
      else O <= '0';
      end if;
    end if;
  end process;
end Behaviour;
```

Fig.5.24. Secvența VHDL pentru implementarea modulului Counter

Schimbul de date între modulul UART și registrele asociate canalelor transductor va fi realizat în mod secvențial datorită numărului diferit de biți pe care acestea sunt reprezentate (canalul transductor – 32 biți, DataIn/Out pe 8 biți). Vom implementa aceste operații printr-un bloc suplimentar, denumit HARTFetcher, de înscriere/extragere de date care va face conversia 32:8 și 8:32 pe principiul încărcării cu patru pachete de 8 biți a registrelor TxHART și RxHART aferente canalelor transductor Channel 1 respectiv Channel 2.

```
architecture Behavioral of HARTFetcher is
  component FetchTxData
    generic (cnt : integer range 0 to 3);
    Port ( Trigger1 : in STD_LOGIC;
           TxDone : in STD_LOGIC;
           DataOTx : out STD_LOGIC_VECTOR (7 downto 0);
           TxData : in STD_LOGIC_VECTOR (31 downto 0));
  end component FetchTxData;
  component FetchRxData
    Port ( Trigger2 : in STD_LOGIC;
           RxAvIn : in STD_LOGIC;
           DataORx : in STD_LOGIC_VECTOR (7 downto 0);
           RxData : out STD_LOGIC_VECTOR (31 downto 0));
  end component FetchRxData;
begin
  TxFetcher: FetchTxData
    generic map (cnt => cc)
    port map (Trigg1,TxDn,DOTx,TxDat);
  RxFetcher: FetchRxData port map (Trigg2,RxAvail,DORx,RxDat);
end Behavioral;
```

Fig.5.25. Secvența VHDL a nivelului superior aferent HARTFetcher

Blocul HARTFetcher conține două module: TxFetcher, pentru extragerea secvențială din registrul TxHART (pe 32 biți) a 8 biți de date pentru UART (la DataIn) și RxFetcher, pentru preluarea pachetelor de date pe 8 biți furnizate de către UART la ieșirea DataOut și înscrierea secvențială a acestora în RxHART (pe 32 biți). Descrierea în VHDL a entității HARTFetcher este expusă în figura 5.25.

Modulul TxFetcher este implementat conform secvenței din figura 5.26.

```

entity FetchTxData is
  generic (cnt : integer range 0 to 3);
  Port (   Trigger1 : in  STD_LOGIC;
          TxDone   : in  STD_LOGIC;
          DataOTx  : out STD_LOGIC_VECTOR (7 downto 0);
          TxData   : in  STD_LOGIC_VECTOR (31 downto 0));
end FetchTxData;
architecture Behavioral of FetchTxData is
begin
  WriteHART:process(Trigger1,TxDone,TxData)
    variable cnt : integer range 0 to 3;
    begin
      if cnt = 4 then cnt := 0;
      end if;
      if Trigger1 = '1' and TxDone = '1' then
        MUX: case cnt is
          when 0 => DataOTx <= TxData(7 downto 0);
          when 1 => DataOTx <= TxData(15 downto 8);
          when 2 => DataOTx <= TxData(23 downto 16);
          when 3 => DataOTx <= TxData(31 downto 24);
        end case MUX;
        cnt := cnt+1;
      end if;
    end process WriteHART;
end Behavioral;

```

Fig.5.26. Secvența VHDL aferentă modulului TxFetcher

Modulul RxFetcher operează conform secvenței VHDL din figura 5.27.

```

entity FetchRxData is
  Port (   Trigger2 : in  STD_LOGIC;
          RxAvIn   : in  STD_LOGIC;
          DataORx  : in  STD_LOGIC_VECTOR (7 downto 0);
          RxData   : out STD_LOGIC_VECTOR (31 downto 0));
end FetchRxData;
architecture Behavioral of FetchRxData is
  signal RxHART : STD_LOGIC_VECTOR (31 downto 0);
  signal ReadA  : STD_LOGIC;
begin
  ReadHART:process(Trigger2,RxAvIn,DataORx,RxHART)
    variable c:integer;
    begin
      c:=0;
      ReadA <= Trigger2;
      RXL: while (c<=31) loop
        if (ReadA='1' and RxAvIn='1') then
          RxHART(c+7 downto c) <= DataORx;
        end if;
        c:=c+8;
      end loop RXL;
      RxData <= RxHART;
    end process ReadHART;
end Behavioral;

```

Fig.5.27. Secvența VHDL aferentă modulului RxFetcher

În urma sintezei aferente implementării în circuitul FPGA Virtex 4, obținem simbolurile pentru cele două module din structura blocului HARTFetcher, așa cum am expus în figura 5.28.

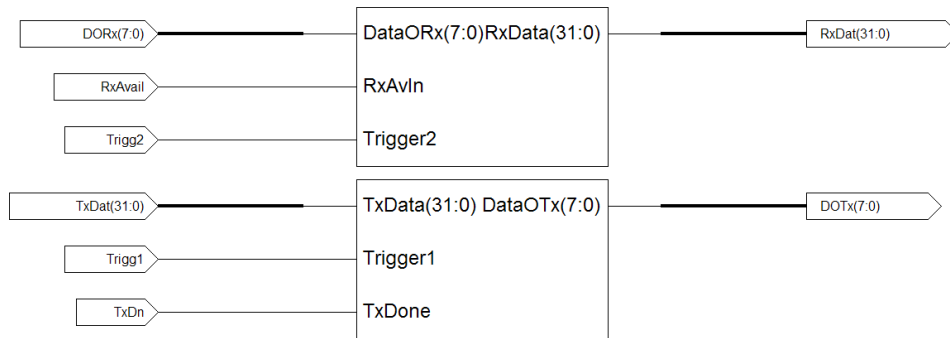


Fig.5.28. Semnalele de intrare/ieșire aferente blocului HARTFetcher

Modulul TxFetcher va dispune de structura internă din figura 5.29.

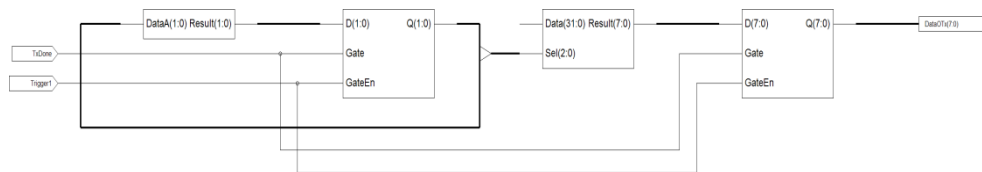


Fig.5.29. Modulul TxFetcher

Modulul RxFetcher, cu o structură relativ simplă, va conține circuite bistabile de tip D, conform figurii 5.30.

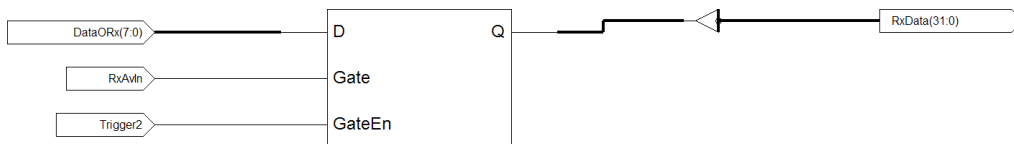


Fig.5.30. Modulul RxFetcher

Am conceput codul VHDL sub formă modulară, cu definirea distinctă a entităților FetchTxData și FetchRxData și apoi instanțierea acestora în blocul programului principal, sub formă de componente (component). Legătura între liniile de intrare/ieșire ale blocului HARTFetcher și componentele astfel definite s-a realizat prin maparea porturilor individuale către cele globale (instrucțiunea „port map”). În acest fel s-a oferit posibilitatea dezvoltării ulterioare a structurii inițiale propuse pentru blocul HARTFetcher.

Structura nivelului funcțional superior al blocului UART, obținută în urma procesului de sinteză, va cuprinde și modulul HARTFetcher.

5.2.2.8 Modemul HART din componența IPTIM

Un rol important în asigurarea compatibilității cu dispozitivele de teren HART îl deține modemul HART, interfațat pe de o parte cu blocul UART și pe de altă parte cu circuitele de adaptare la bucla de curent 4...20 mA. Vom implementa în VHDL o structură de modem corespunzătoare cerințelor ce rezultă din figura 5.15 și 5.16. Totodată vom respecta notațiile standard utilizate în domeniu pentru semnalele de intrare și ieșire ale acestui modem HART. Sintetizând totalitatea semnalelor vehiculate, blocul Modem va dispune de o structură I/O prezentată în figura 5.31.

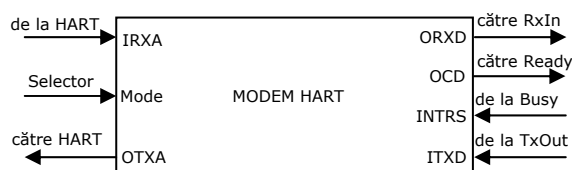


Fig.5.31. Liniile de intrare/ieșire ale modemului HART

Pentru identificarea ușoară a funcțiilor implementate, am păstrat pentru liniile de intrare/ieșire a acestui bloc denumirile standard utilizate în literatura de specialitate în cazul circuitelor dedicate modem HART. În schimb, structura internă va fi adaptată cerințelor impuse de modul de funcționare al IPTIM și al blocului intern UART. Liniile IRXA și OTXA vor accepta și vor furniza semnal numeric (niveluri standard CMOS), urmând ca anumite circuite externe de interfațare și formare a semnalului modulat trapezoidal (impus de comunicația HART) să fie atașate în cadrul unei dezvoltări ulterioare (vezi figura 5.2).

Semnificațiile liniilor I/O aferente modemului HART propus în figura 5.31 sunt indicate în tabelul 5.2.

Denumire	Direcție	Semnificație
IRXA	In	Intrarea de date, frecvența 1200Hz/2200Hz
OTXA	Out	Ieșirea de date, FSK, frecvența 1200Hz/2200Hz
ORXD	Out	Ieșirea digitală, demodulată, serială
OCD	Out	Semnalare apariție purtătoare (validează ORXD)
INTRTS	In	Cerere transmisie (validează OTXA)
ITXD	In	Intrare digitală, serială
Mode	In	Selector mod de lucru (analog/digital)
SysClk	In	Semnalul de tact global al sistemului

Tabelul 5.2. Semnificațiile liniilor I/O aferente modemului HART

În plus față de liniile I/O standard, am prevăzut o intrare Mode, de selecție a modului de lucru (utilizat cu precădere în testarea funcțională a sistemului). În cazul în care dorim preluarea sau furnizarea de semnale digitale cu niveluri logice standard (CMOS), stabilim valoarea bitului Selector la „1” logic (în lipsa circuitelor de adaptare și formare a semnalului trapezoidal și a filtrelor de intrare/ieșire).

Pentru comutarea modemului în modul de operare standard, stabilim bitul Selector pe „0” logic. În acest caz se va ține cont de generarea semnalului OCD în funcție de nivelurile de referință HART pentru semnalul FSK de intrare (neimplementat în cazul nostru).

Structura bloc internă a modemului HART este prezentată în figura 5.32.

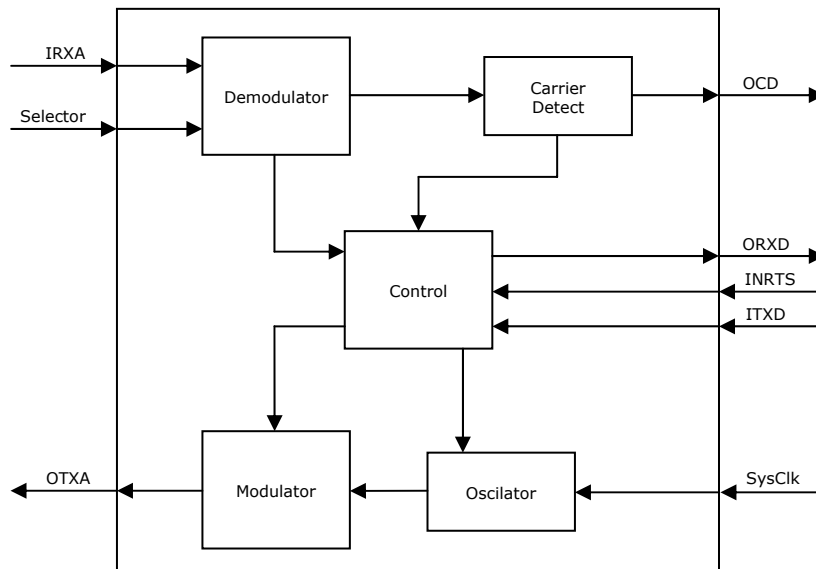


Fig.5.32. Structura bloc aferentă modemului HART

Implementarea blocurilor funcționale va ține cont de asigurarea comunicării modemului pe două linii cu dispozitivele de teren HART (IRXA și OTXA). Liniile dedicate interfațării cu blocul UART vor fi reprezentate prin semnale interne în cadrul modului IPTIM.

```
entity HARTModem is
  Port (
    IRXA : in STD_LOGIC;
    OTXA : out STD_LOGIC;
    Mode : in STD_LOGIC;
    ORXD : out STD_LOGIC;
    OCD : out STD_LOGIC;
    INRTS : in STD_LOGIC;
    ITXD : in STD_LOGIC;
    SysClk : in STD_LOGIC);
end HARTModem;
architecture Behavioral of HARTModem is
  component Modulador
    Port (
      TxDMod : in STD_LOGIC;
      ReqMod : in STD_LOGIC;
      OTxMod : out STD_LOGIC;
      ClkMod : in STD_LOGIC);
  end component;
  component Demodulator
    Port (
      IRxMod : in STD_LOGIC;
      RxDMod : out STD_LOGIC;
      CDMod : out STD_LOGIC;
      SelMod : in STD_LOGIC;
      ClkDMod : in STD_LOGIC);
  end component;
end Behavioral;
```

Fig.5.33. Definierea în VHDL a blocurilor funcționale aferente modemului HART

Secvența VHDL prin care se definesc blocurile modemului HART este expusă în figura 5.33.

Modulatorul va accepta ca intrare de date ieșirea TxOut furnizată de către UART iar cererea de transmisie se va realiza prin intrarea INRTS unde este aplicat semnalul Busy provenind de la UART.

În componența modulatorului vor interveni două oscilatoare locale, sincronizate cu semnalul Clk al sistemului, cu factori de divizare corespunzători pentru obținerea frecvențelor de 1200 Hz respectiv 2200 Hz („1” logic respectiv „0” logic). Putem utiliza ca principiu de funcționare, modulul Counter din componența UART, unde se realizează deja o divizare a frecvenței semnalului de tact la 1200 Hz. Vom implementa, în mod similar, un oscilator care va furniza și semnalul cu frecvența de 2200 Hz aferent nivelului „0” logic, factorul de divizare având valoarea 45454. Ambele oscilatoare vor fi sincrone cu tactul sistemului (100 MHz). Datele aduse pe linia ITXD vor comanda un bloc multiplexor care va permite comutarea ieșirii validate prin semnalul Busy (de la UART) pe una dintre cele două semnale provenite de la oscilatoare. Sincronismul se va păstra datorită utilizării semnalului de tact global, activ pe frontul crescător. Secvența VHDL prin care implementăm operarea blocului modulator este prezentată în figura 5.34.

```

architecture Behavioral of Modulator is
signal   OutLow  : out STD_LOGIC;
signal   OutHigh : out STD_Logic;
signal   ENLow   : in  STD_LOGIC;
signal   ENHigh  : in  STD_LOGIC;
signal   Reset   : in  STD_LOGIC;
component Counter
  generic(COUNT: INTEGER range 0 to 128000); -- contor baud
  port ( Clk      : in  std_logic; -- Clock
        Reset    : in  std_logic; -- Reset
        CE       : in  std_logic; -- Chip Enable
        O        : out std_logic); -- Output
end component;
begin
  Reset <= '0';
  ENLow  <= '0';
  ENHigh <= '0';
  OSCLow : Counter
    Generic map (COUNT => DIVLow)
    port map (ClkMod,Reset,ENLow,OutLow);
  OSCHigh : Counter
    Generic map (COUNT => DIVHigh)
    port map (ClkMod,Reset,ENHigh,OutHigh);
  ModGen : process (TxDMod,ReqMod,ClkMod,OutLow,OutHigh)
  begin
    Transmit : while (ReqMod = '1') loop
      ModSel : case TxDMod is
        when '0' => ENLow <= '1';
                    ENHigh <= '0';
                    OTxMod <= OutLow;
        when '1' => ENHigh <= '1';
                    ENLow <= '0';
                    OTxMod <= OutHigh;
        when others => ENLow <= '0';
                    ENHigh <= '0';
      end case ModSel;
    end loop Transmit;
  end process;
end Behavioral;

```

Fig.5.34. Secvența VHDL aferentă operării blocului Modulator

Procesul de demodulare se bazează pe contorizarea duratei între două fronturi crescătoare ale semnalului furnizat la intrarea IRXA a blocului demodulator. Valoarea duratei se va compara cu factorul de divizare DivLow (având valoarea 45454) aferent frecvenței de 2200 Hz, obținându-se astfel un criteriu de decizie în privința valorilor logice furnizate la ieșirea ORXD. În cazul în care nu se recepționează semnal provenind de la dispozitivele HART, ieșirea demodulată ORXD va fi menținută pe „1” logic (OCD pe „0” logic).

Secvența VHDL prin care implementăm operarea demodulatorului este prezentată în figura 5.35.

```

architecture Behavioral of Demodulator is
  signal timecount : std_logic_vector(15 downto 0);
begin
  process(ClkDMod)
  begin
    if rising_edge(ClkDMod) then
      timecount <= timecount + 1;
    end if;
  end process;
  process(IrxMod,SelMod,timecount)
  variable edge : integer range 0 to 6;
  begin
  Demod: case SelMod is
    when '0' => CDMod <= 'Z';
                RxDMod <= 'Z';
    when '1' =>
      if timecount > 45454 then
        if edge >= 2 then
          CDMod <= '1';
          RxDMod <= '1';
        else
          CDMod <= '1';
          RxDMod <= '0';
        end if;
      else
        CDMod <= '0';
      end if;
    elsif rising_edge(IRxMod) then
      edge := edge + 1;
    elsif IRxMod = '0' then
      RxDMod <= '1';
    end if;
    when others => CDMod <= 'Z';
                  RxDMod <= 'Z';
  end case Demod;
  end process;
end Behavioral;

```

Fig.5.35. Secvența VHDL aferentă operării blocului demodulator

Liniile I/O ale celor două blocuri componente ale modemului HART vor fi mapate către liniile definite în cadrul entității „HARTModem” (figura 5.33) prin intermediul secvenței VHDL din figura 5.36.

```

ModHART : Modulator
  port map (ITXD,INRTS,OTXA,SysClk);
DemHART:Demodulator
  port map (IRXA,ORXD,OCD,Mode,SysClk);

```

Fig.5.36. Maparea liniilor I/O aferente blocurilor modulator/demodulator

Modulul IPTIM va gestiona semnalele asociate blocurilor UART și Modem.

5.2.2.9 Logica de comandă a canalului H-Bridge

Pe baza structurii propuse în figura 5.17, vom elabora în VHDL secvența de cod care implementează operarea modului logic aferent comenzii tranzistoarelor MOS din puntea H (H-Bridge) externă (canalul transductor Channel 3). În acest sens, trebuie asigurată posibilitatea declanșării operării acestui canal, prin evaluarea semnalului Trigger 3 provenit de la blocul Control al modului TIM. Se vor lua în considerare și semnalele Reset și Enable 3 (de inițializare și validare). Datele se vor extrage din registrul Data 3 (asociat canalului Channel 3), pe patru biți, fiecare front crescător al impulsului de tact sistem (Clk) validând furnizarea la ieșirile D0,...,D3 (liniile DHOut ale modului logic H-Bridge) a semnalului de comandă pentru actuatorul asociat canalului 3. Figura 5.37 prezintă secvența VHDL prin care implementăm logica de comandă aferentă canalului 3 (H-Bridge).

```

entity HbridgeLogic is
  Port ( DataIn : in  STD_LOGIC_VECTOR (3 downto 0);
        Enable : in  STD_LOGIC;
        Clk : in  STD_LOGIC;
        Trigg : in  STD_LOGIC;
        Reset : in  STD_LOGIC;
        Stat : out  STD_LOGIC_VECTOR (7 downto 0);
        Int : out  STD_LOGIC_VECTOR (7 downto 0);
        DHOut : out  STD_LOGIC_VECTOR (3 downto 0));
end HbridgeLogic;
architecture Behavioral of HbridgeLogic is
  signal driveH : STD_LOGIC_VECTOR (3 downto 0);
  signal statv : STD_LOGIC_VECTOR (7 downto 0);
  signal intv : STD_LOGIC_VECTOR (7 downto 0);
  begin
  DRV : process(Reset,Enable,Trigg)
    begin
      if Reset='1' then
        driveH <= "0000";
        statv <= "00000000";
        intv <= "00000000";
      end if;
      if Enable='1' and Trigg='1' and Reset = '0' then
        driveH <= DataIn;
      else
        driveH <= "0000";
      end if;
    end process DRV;
  EXE : process(Clk)
    begin
      if Rising_edge(Clk) then
        DHout <= driveH;
      end if;
    end process EXE;
  --Stat,Int momentan neutilizate
end Behavioral;

```

Fig.5.37 Secvența VHDL aferentă implementării canalului H-Bridge

5.2.2.10 Structura implementată a SSIHE

Pentru a permite comunicarea între modulul IPTIM și IPIF, este necesară definirea semnalelor de intrare/ieșire ale acestui modul, impuse de către interfața IPIC (vezi figura 5.8). Suplimentar, vom declara semnalele interne ale IPTIM prin

care se vehiculează informația între unitatea de control și canalele transductor (pe baza considerentelor expuse în figurile 5.11, 5.12, 5.13 și 5.14). În acest scop, vom utiliza mediul de proiectare Xilinx EDK (Embedded Development Kit) care conține utilitarele XPS (Platform Studio) și SDK (Software Development Kit) și permite integrarea într-o structură generală IPIF a modulelor proiectate de către utilizator pentru anumite aplicații specifice. Pe baza diagramei prezentate în figura 5.2, vom elabora configurația internă a SSIHE în mediul EDK/XPS și vom obține structura completă expusă în figura 5.38.

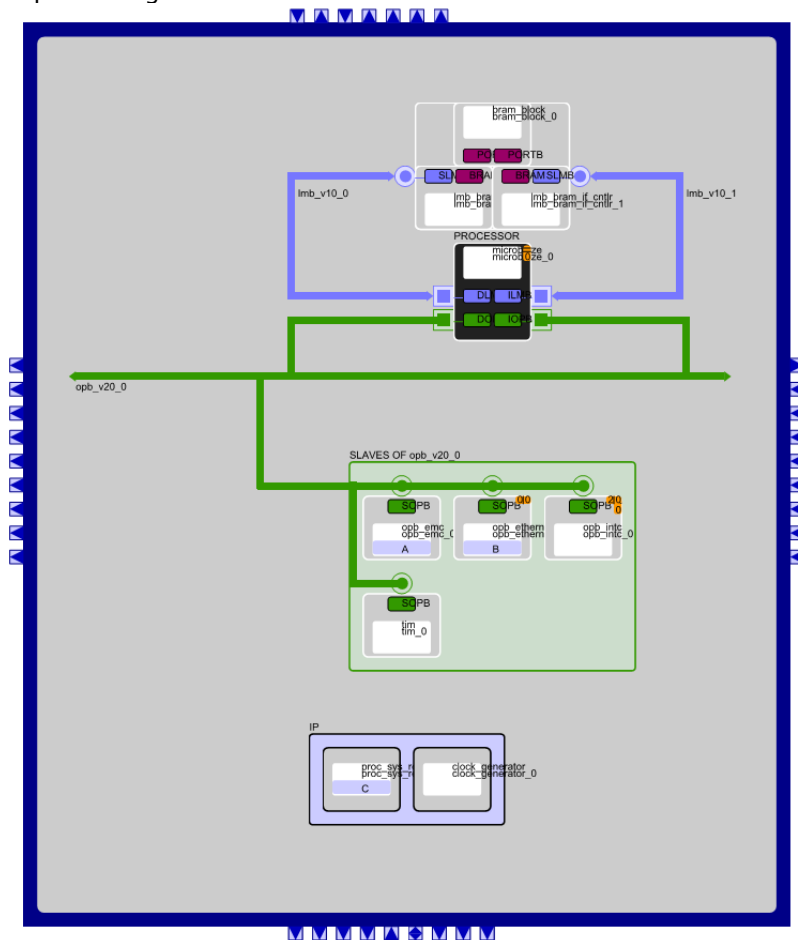


Fig.5.38 Structura internă completă a SSIHE implementată în Xilinx EDK/XPS

Recunoaștem în diagrama expusă în figura 5.38 elementele incluse în structura bloc inițială a SSIHE (figura 5.2): procesorul Microblaze împreună cu magistrala locală LMB (Local Memory Bus) care permite accesarea memoriei interne BRAM, controlerul pentru memoriile externe, nucleul Ethernet, controlerul pentru întreruperi și modulul TIM intern. Magistrala internă OPB asigură legătura între blocurile interne și microprocesorul Microblaze. Legătura cu perifericele externe este realizată prin linii de port bidirecționale, marcate în figura 5.38 în mod corespunzător. Semnificația simbolurilor generice incluse în diagramă sunt indicate în figura 5.39.

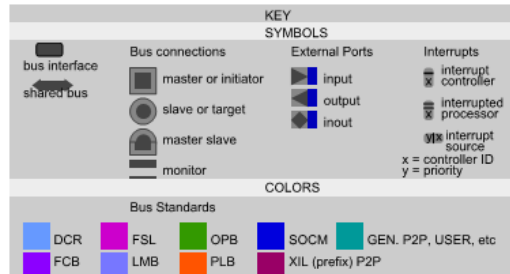


Fig.5.39 Semnificația simbolurilor utilizate în structura internă a SSIHE

În cadrul procesului de proiectare, am parcurs etapele specifice integrării perifericelor în cadrul circuitului FPGA Virtex 4. Am selectat tipul magistralei interne (OPB) și am stabilit modul de conectare a acestor periferice la OPB. Figura 5.40 indică modul de conectare la magistrală a dispozitivelor interne.

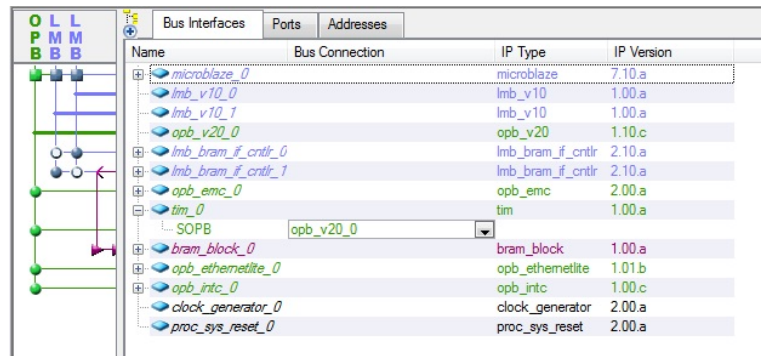


Fig.5.40. Conectarea perifericelor interne la magistrală

Modulul TIM intern, regăsit în figura 5.40 sub denumirea tim_0, este conectat la magistrala OPB (opb_v20_0) ca dispozitiv slave (SOPB).

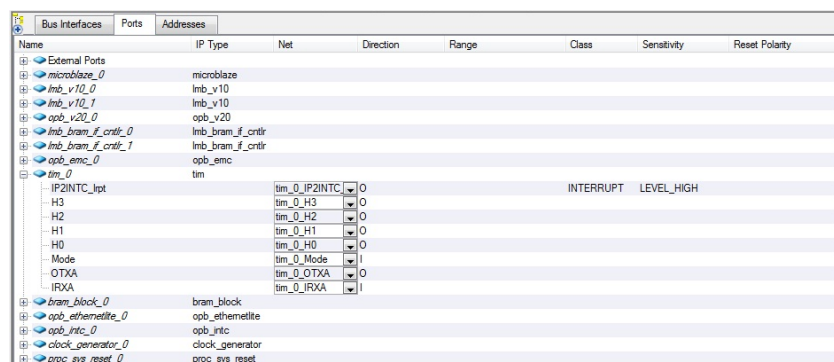


Fig.5.41. Liniile de port aferente modulului TIM intern

Liniile de port definite pentru tim_0 (figura 5.41) sunt cele aferente conectării la OPB, linia de întrerupere și cele externe pentru conectarea transductoarelor.

Centralizarea tuturor liniilor de port externe este realizată în figura 5.42.

Name	IP Type	Net	Direction	Range	Class	Sensitivity	Reset Polarity
tim_0_H3_pin		tim_0_H3	O				
tim_0_H2_pin		tim_0_H2	O				
tim_0_H1_pin		tim_0_H1	O				
tim_0_H0_pin		tim_0_H0	O				
tim_0_Mode_pin		tim_0_Mode	I				
tim_0_OTXA_pin		tim_0_OTXA	O				
tim_0_IRXA_pin		tim_0_IRXA	I				
proc_sys_reset_0_Ext_Reset...		proc_sys_reset	I		RST		
opb_ethernetlite_0_PHY_bc_d...		opb_ethernetlite	I				
opb_ethernetlite_0_PHY_rc_d...		opb_ethernetlite	I				
opb_ethernetlite_0_PHY_crs_pin		opb_ethernetlite	I				
opb_ethernetlite_0_PHY_dv_pin		opb_ethernetlite	I				
opb_ethernetlite_0_PHY_rc_d...		opb_ethernetlite	I	[3:0]			
opb_ethernetlite_0_PHY_col_pin		opb_ethernetlite	I				
opb_ethernetlite_0_PHY_rc_er...		opb_ethernetlite	I				
opb_ethernetlite_0_PHY_rst_n...		opb_ethernetlite	O				
opb_ethernetlite_0_PHY_bc_e...		opb_ethernetlite	O				
opb_ethernetlite_0_PHY_bc_d...		opb_ethernetlite	O	[3:0]			
opb_emc_0_Mem_A_pin		opb_emc_0_Me	O	[0:31]			
opb_emc_0_Mem_DQ		opb_emc_0_Me	IO	[0:31]			
opb_emc_0_Mem_DQ_Lpin		opb_emc_0_Me	I	[0:31]			
opb_emc_0_Mem_DQ_Q_pin		opb_emc_0_Me	O	[0:31]			
opb_emc_0_Mem_DQ_T_pin		opb_emc_0_Me	O	[0:31]			
opb_emc_0_Mem_CEN_pin		opb_emc_0_Me	O	[0:1]			
opb_emc_0_Mem_OEN_pin		opb_emc_0_Me	O	[0:1]			
opb_emc_0_Mem_WEN_pin		opb_emc_0_Me	O				
opb_emc_0_Mem_QWEN_pin		opb_emc_0_Me	O	[0:3]			
opb_emc_0_Mem_BEN_pin		opb_emc_0_Me	O	[0:3]			
opb_emc_0_Mem_RFN_pin		opb_emc_0_Me	O				
opb_emc_0_Mem_CE_pin		opb_emc_0_Me	O	[0:1]			
opb_emc_0_Mem_ADV_LDN...		opb_emc_0_Me	O				
opb_emc_0_Mem_LBON_pin		opb_emc_0_Me	O				
opb_emc_0_Mem_CKEN_pin		opb_emc_0_Me	O				
opb_emc_0_Mem_RNW_pin		opb_emc_0_Me	O				

Fig.5.42. Liniile de port externe implementate în FPGA Virtex 4

Etapa următoare este atribuirea adreselor prin care se poate accesa fiecare periferic de către procesorul Microblaze. Rezultatul procesului de atribuire al adreselor este prezentat în figura 5.43.

Instance	Name	Base Address	High Address	Size	Bus Interface(s)	Bus Connection	Lock	ICache	DCache	IP Type	IP Version
tim_0	C_AR0_BASEADDR	0x10000000	0x10ffff	16M	SOPB	opb_v20_0				tim	1.00 a
lmb_bram_*_cntrl_0	C_BASEADDR	0x00000000	0x0000ffff	64K	SLMB	lmb_v10_0				lmb_bram_*_cntrl	2.10 a
lmb_bram_*_cntrl_1	C_BASEADDR	0x00000000	0x0000ffff	64K	SLMB	lmb_v10_1				lmb_bram_*_cntrl	2.10 a
opb_ethernetlite_0	C_BASEADDR	0x40e00000	0x40e0ffff	64K	SOPB	opb_v20_0				opb_ethernetlite	1.01 b
opb_jrtc_0	C_BASEADDR	0x41200000	0x4120ffff	64K	SOPB	opb_v20_0				opb_jrtc	1.00 c
opb_v20_0	C_BASEADDR	0x7c218000	0x7c21ffff	32K	Not Applicable					opb_v20	1.10 c
tim_0	C_BASEADDR	0x7c200000	0x7c20ffff	64K	SOPB	opb_v20_0				tim	1.00 a
opb_emc_0	C_MEM0_BASEADDR	0x7c230000	0x7c23ffff	64K	SOPB	opb_v20_0				opb_emc	2.00 a
opb_emc_0	C_MEM1_BASEADDR	0x7c220000	0x7c22ffff	64K	SOPB	opb_v20_0				opb_emc	2.00 a

Fig.5.43. Atribuirea adreselor perifericelor din structura internă a SSIHE

Elementul suplimentar introdus în structura internă a SSIHE este magistrala locală LMB, prin care asigurăm accesul direct la memoria BRAM locală prin intermediul controlerelor lmb_bram_cntrl. Această memorie este împărțită în două zone: date și instrucțiuni, fiecare zonă dispunând de controler dedicat. Controlerul BRAM pot accesa un spațiu de memorare de 64 KO, dar din considerente de optimizare a implementării fizice a sistemului, memoria BRAM de 64 KO este împărțită în două zone de 32 KO atribuite instrucțiunilor respectiv datelor gestionate de către procesorul Microblaze.

Prin intermediul platformei Xilinx SDK (Software Development Kit) vom importa structura creată în etapele anterioare pentru a avea posibilitatea elaborării rutinelor de comandă în limbajul C/C++ aferente testării funcționale a SSIHE.

5.2.2.11 Implementarea modulelor software aferente SSIHE

Platforma hardware a SSIHE elaborată în paragraful 5.2.2 va fi utilizată ca suport pentru executarea rutinelor software care vor implementa și valorifica principalele caracteristici expuse ale sistemului inteligent. Vom parcurge în acest sens etapele de proiectare software indicate în figura 5.44.

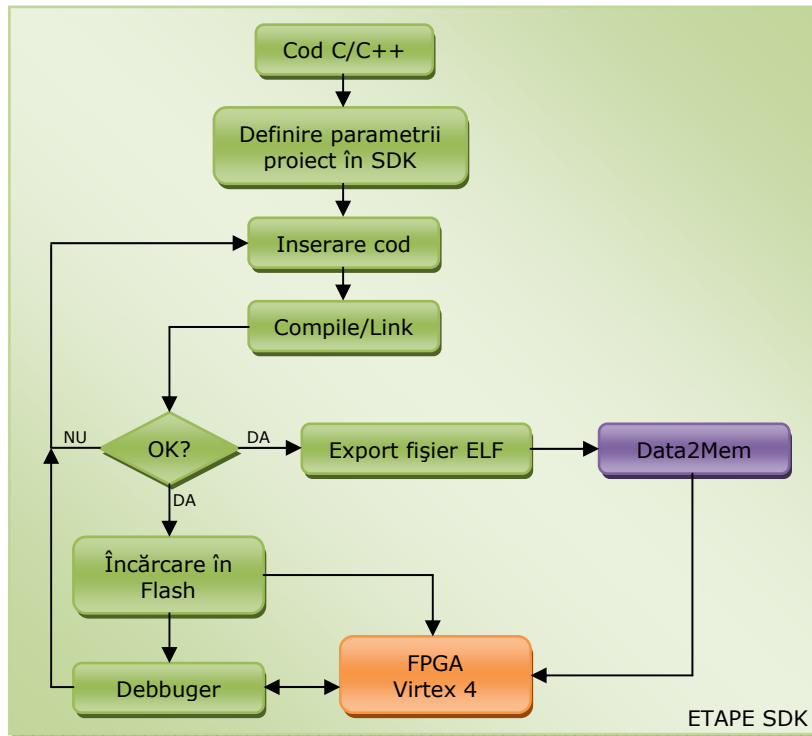


Fig.5.44. Etapele proiectării software a SSIHE în mediul Xilinx SDK

Rutinele elaborate în limbajul C++ vor cuprinde atât funcții predefinite pentru accesarea de către procesorul Microblaze a registrelor asociate perifericelor preluate din biblioteca IP Xilinx și adaptate la cerințele SSIHE (EMAC, EMC, controlerul de întreruperi și BRAM) conectate la OPB cât și secvențe de cod dedicate pentru controlul modulului TIM intern. Vom analiza în continuare doar aspecte legate de rutinele de control aferente TIM.

Principiul de bază al schimbului de date între modulul intern TIM și procesorul Microblaze constă în adresarea prin pointeri a registrelor asociate canalelor transductor, pornind de la adresa de bază a TIM. Reprezentarea acestor adrese se realizează prin tipul de date Xuint32 (pe 32 de biți) iar adresa de bază a modulului TIM (TIMLogic) se poate prelua prin secvența prezentată în figura 5.45.

```

#include "xparameters_ml401.h"
..... secvențe omise.....
Xuint32 *TIMLogic;
TIMLogic=(Xuint32*)XPAR_tim_0_BASEADDR;
  
```

Fig.5.45. Secvența C++ pentru preluarea adresei de bază a TIM

Se poate observa în figura 5.45 utilizarea fișierului „header C++” denumit „xparameters_ml401.h” care conține în prealabil parametrii importanți care descriu proprietățile fiecărui periferic inclus în structura internă a SSIHE. Astfel, apelăm la noțiunea „XPAR_tim_0_BASEADDR” care specifică adresa de bază a modului TIM intern și este definit în „xparameters_ml401.h”.

Urmează implementarea schimbului de date efectiv între Microblaze și TIM prin intermediu magistrale OPB. În acest sens, față de adresa de bază vom introduce un decalaj (offset) pentru fiecare canal transductor în parte și vom declara variabilele generice data1, data2 și data3 de tip Xuint32 care vor stoca datele aferente celor trei canale. Secvența de program C++ prin care transmitem către canalul TxHART valoarea data1, citim valoarea data2 de la canalul RxHART și în final controlăm canalul HBridge prin valoarea înscrisă în data3 este expusă în figura 5.46.

```
Xuint32 data1,data2;
int data3;
data1=HARTCMD_0; // emitere comanda HART numarul 0 (citire identificatori unici)
data3=0x0A; //comandă HBridge
*(TIMLogic+0x1)=data1; // canal1
data2=*(TIMLogic+0x2); // canal2
*(TIMLogic+0x3)=data3; // canal3
```

Fig.5.46. Secvență C++ aferentă schimbului de date între Microblaze și TIM

Pentru referirea cât mai ușoară în program asupra comenzilor HART am conceput un fișier „header C++”, denumit „HARTCommands.h” prin care definim prin constante numerice fiecare comandă în parte, conform specificațiilor standardului HART așa cum este indicat în [63], [64].

Blocul TEDS (intern în cazul implementării în SSIHE) impus de către standardul IEEE 1451, este alocat în cadrul memoriei Flash accesată de către procesorul Microblaze prin intermediul controlerului EMC (External Memory Controller), conectat la magistrala OPB și având adresa specificată în figura 5.43. Prin utilizarea unor rutine oferite de către Xilinx SDK, vom înscrie la inițializarea sistemului parametrii dispozitivelor de teren HART conectate la sistem, extrași prin canalul RxHART. Datele astfel stocate se vor putea utiliza ulterior prin citirea zonelor de memorie Flash alocate.

Tipurile de date specifice TEDS au fost definite în limbajul C în Capitolul 4, paragraful 4.2 și vor fi incluse în structura generală a softului de comandă.

În mod similar, realizăm comunicarea pe rețeaua Ethernet precum și gestionarea întreruperilor, prin rutine furnizate de către Xilinx SDK și adaptate la cerințele sistemului. Aceste rutine sunt utilizate doar pentru testarea funcțională a SSIHE și nu constituie obiectul tezei.

5.2.3. Integrarea elementelor software/hardware în structura SSIHE

Prin parcurgerea etapelor prezentate în paragrafele precedente am obținut un set de elemente atât hardware cât și software care contribuie la implementarea funcționalității sistemului inteligent propus în lucrare. Acestea constituie rezultatul co-proiectării hardware/software utilizată la ora actuală în elaborarea sistemelor de tip „embedded” (integrate) și oferă avantajul posibilității dezvoltării simultane a celor două componente (hardware și software). Procesul de reunire a acestor elemente în cadrul circuitului FPGA Virtex 4 se poate realiza sub mediul Xilinx EDK amintit și apriori, având la dispoziție unelte specifice metodei în cauză.

Ilustrarea procesului de integrare, sub formă grafică, este realizată prin organigrama expusă în figura 5.47.

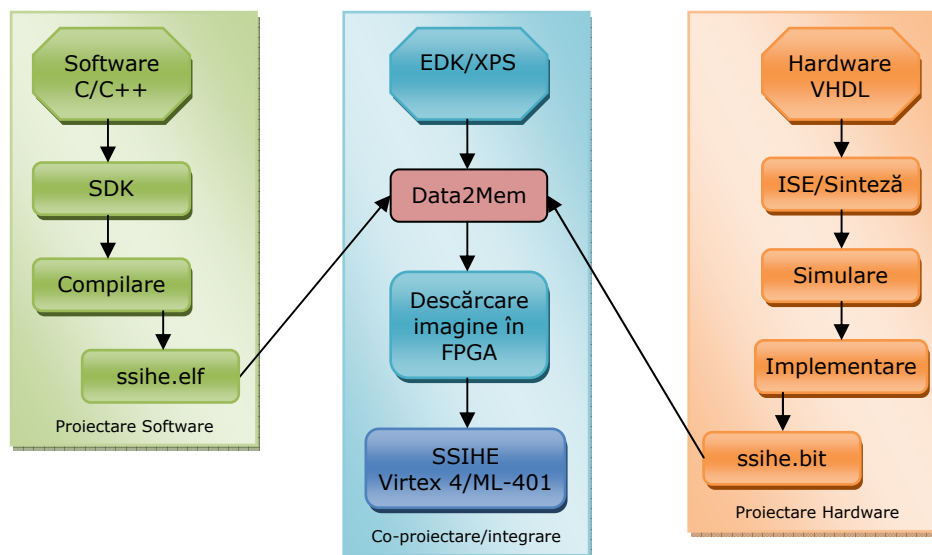


Fig.5.47. Fluxul procesului de co-proiectare hardware/software aferent SSIHE

Elementul central al fluxului de proiectare îl constituie mediul EDK/XPS prin care definim și structurăm componentele de bază ale sistemului propus (procesor, controlere, modul TIM etc.). În urma acestui prim proces, rezultă fișiere drivere care descriu sintetic componentele incluse în sistem.

Prin limbajul de descriere hardware, VHDL, indicăm structura blocurilor incluse în sistem prin EDK. Rezultatul sintezei codului VHDL este pregătit pentru implementare prin generarea fișierului „ssihe.bit”.

Având la dispoziție platforma fizică, în mediul SDK am elaborat rutinele software în limbajul C/C++ care rulează pe platforma concepută în EDK/XPS iar prin procesul de compilare va rezulta în final fișierul executabil „ssihe.elf”.

Cele două fișiere, „ssihe.bit” și „ssihe.elf” au ca destinație unealta de traducere specifică mediului EDK, „Data2Mem” care ne permite înglobarea unei imagini software în fluxul de biți care descrie structura hardware a aceluiași sistem. Urmează descărcarea acestor informații în circuitul FPGA și testarea funcțională a sistemului.

5.3. Sinteza elementelor SSIHE și testarea funcțională

Structura SSIHE descrisă în paragraful 5.2.2 prin diagrame bloc și secvențe de cod VHDL a fost implementată într-un circuit FPGA Virtex 4 inclus pe platforma sistem ML-401. Totodată am utilizat unele resurse externe circuitului FPGA oferite de către perifericele amplasate pe platforma sistem. Procesul de sinteză la nivel de unități logice a condus la rezultate semnificative atât din punct de vedere teoretic cât și practic, anumite aspecte fiind reliefate în continuare.

5.3.1. Rezultatele sintezei elementelor modulului TIM intern

Conceput ca element Slave, conectat pe magistrala internă OPB, modulul TIM intern conține blocul IPIF, interfața IPIC și blocul IPTIM. Acesta din urmă este alcătuit din blocul UART, modemul HART, logica de comandă H-Bridge, arbitrul de întreruperi și blocul de control. Vom parcurge în continuare rezultatele obținute în urma sintezei anumitor blocuri componente din structura IPTIM.

5.3.1.1 Sinteza modulului UART și simularea operării

Blocul UART preia informația furnizată de către modemul HART, pe linia RxIn, o transferă în registrul RxData, extrage informația din registrul TxData, o serializează și în final o furnizează către modemul HART pe linia TxOut. Operațiile prezentate sunt corelate cu semnalul de tact global al sistemului și cu semnalele de interogare/declanșare și validare Trigger, respectiv Enable, gestionate de către logica de control a modulului TIM.

Procesul de sinteză al UART, pe baza codului VHDL, ne furnizează parametrii prezenți în tabelul 5.3.

Device utilization summary: 4vlx25ff668-10			
Logic utilization	Used	Available	Utilization
Number of Slices	62	10752	0%
Number of Slice Flip Flops	63	21504	0%
Number of 4 input LUTs	107	21504	0%
Number of bonded IOBs	28	448	6%
Number of GCLKs	2	32	6%

Tabelul 5.3. Gradul de utilizare al FPGA pentru sinteza UART

Consultând tabelul 5.3 observăm gradul relativ redus de ocupare a resurselor FPGA în cazul implementării modulului UART.

În vederea testării funcționale a acestui modul intern, am generat vectorii de intrare corespunzători operării normale ale acestuia, considerând cazuri tipice pentru semnalele de control provenite de la TIM, atât pentru unitatea de transmisie TX cât și pentru cea de recepție Rx. Figura 5.48 indică funcționarea unității de transmisie serială (Tx).

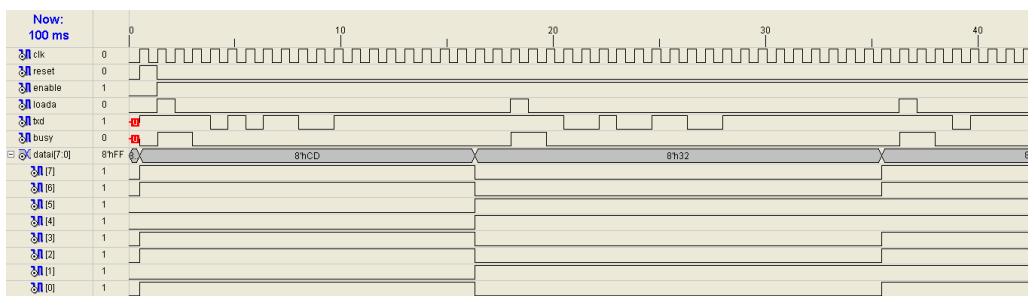


Fig.5.48. Diagrame temporale aferente funcționării unității Tx

În cazul transmisiei seriale avem ca parametru de intrare valoarea pe 8 biți „datai” și ca parametru de ieșire valoarea serializată obținută pe linia „txd”.

Observăm în figura 5.48 preluarea valorii pe 8 biți, „datai” în urma emiterii de către TIM a comenzii de încărcare, „loada” (mapată către semnalul Trigger1). Distingem totodată și semnalul „busy” care este activ pe durata prelucrării datelor de către unitatea Tx. Diagramele din figura 5.48 indică un ciclu repetitiv de preluare a datelor pentru trei impulsuri de încărcare „loada”.

Figura 5.49 expune diagramele temporale aferente operării unității de recepție serială (Rx).

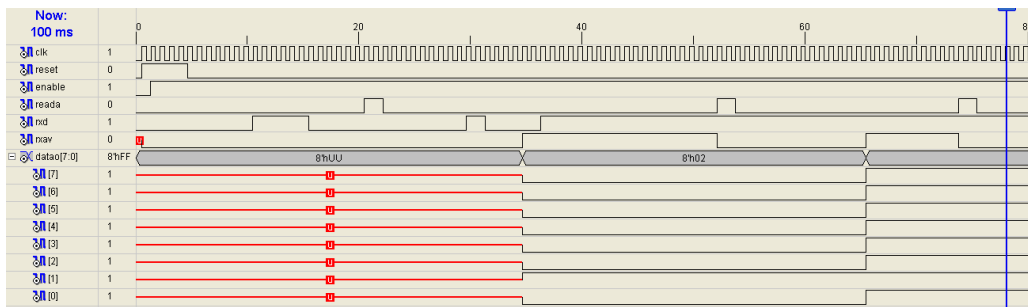


Fig.5.49. Diagrame temporale aferente funcționării unității Rx

Parametrul de intrare este reprezentat prin valorile de pe linia „rxd” recepționate în mod serial. Ciclul de citire se declanșează odată cu emiterea comenzii „reada” de către TIM (mapată către Trigger3). La încheierea deserializării, unitatea Rx activează linia „rxav”, așa cum se poate observa în figura 5.49, în acest moment fiind disponibile datele pe 8 biți înscrise în „datao”. Exemplificăm în figura 5.49 două cicluri de deserializare și citire paralelă a datelor.

5.3.1.2 Sinteza modului modem HART și simularea operării

Elementul cheie în asigurarea comunicării la nivel fizic între SSIHE și dispozitivele de teren HART, modemul HART intern conlucrează cu modulul UART. Procesul de sinteză al modemului HART, pe baza codului VHDL, ne furnizează parametrii prezentați în tabelul 5.4.

Device utilization summary: 4vlx25ff668-10			
Logic utilization	Used	Available	Utilization
Number of Slices	58	10752	0%
Number of Slice Flip Flops	56	21504	0%
Number of 4 input LUTs	111	21504	0%
Number of bonded IOBs	8	448	1%
Number of GCLKs	3	32	9%

Tabelul 5.4. Gradul de utilizare al FPGA pentru sinteza modemului HART

Structura propusă a modemului permite operarea cu ajutorul semnalului de tact global al sistemului, păstrând astfel sincronismul datelor. Pe baza semnalului „Busy” provenit de la UART, se va lua decizia emiterii semnalului modulat FSK la ieșirea OTXA. Similar, la detectarea unui semnal FSK pe linia IRXA, modemul va genera o cerere de transmisie către UART prin activarea liniei OCD și va furniza biții de date pe linia ORXD.

Modul de obținere a semnalului modulat la ieșirea OTXA, corelat cu semnalele de comandă și cu tactul sistemului este indicat în figura 5.50.

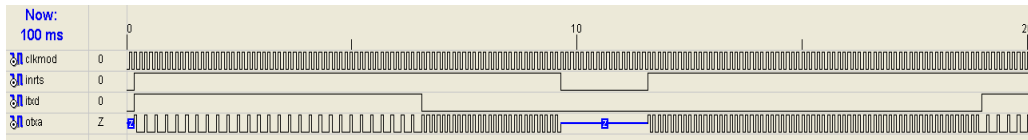


Fig.5.50. Operarea modulatorului HART

La activarea liniei „inrts” de către modulul UART, modulatorul HART va furniza la ieșirea OTXA (semnalul „otxa” din figura 5.50) semnalul FSK având frecvență aferentă nivelului logic de intrare stabilit pe linia „itxd”. Așa cum am mai specificat, pentru nivelul „0” logic avem frecvență de 2200 Hz iar pentru „1” logic frecvență de 1200 Hz. Putem observa pe diagrama din figura 5.50 faptul că la inactivarea liniei „inrts” („0” logic) ieșirea „otxa” va fi trecută în starea de impedanță ridicată, ignorând datele furnizate pe „itxd”.

Demodulatorul, care preia semnalul FSK provenit de la dispozitivele de teren HART (semnalul digital), va opera conform diagramelor prezentate în figura 5.51.

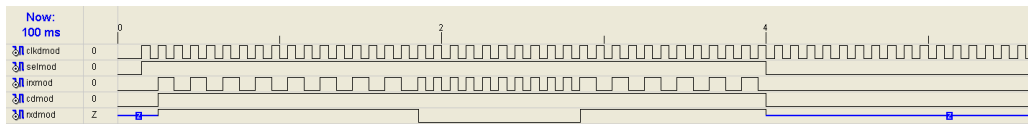


Fig.5.51. Operarea demodulatorului HART

La intrarea „irxmod” se aduce semnalul modulat provenit de la dispozitivele de teren HART. La detectarea transmisiei, demodulatorul va activa linia „cdmod” (Carrier Detect) semnalând existența unor date valide. Putem observa în diagrama din figura 5.51 extragerea nivelului „1” logic pentru frecvența de 1200 Hz a semnalului de intrare și a nivelului „0” logic pentru o frecvență de 2200 Hz la intrare.

Trebuie menționat faptul că pentru generarea diagramelor aferente funcționării blocului modem HART, prezentate în figura 5.50 respectiv 5.51, am considerat ca semnal de tact de referință (2200 Hz) valoarea divizată în mod corespunzător a semnalului de tact a sistemului (100 MHz) și care s-a adus la liniile de intrare „clkmod” respectiv „clckmod”. În acest fel se poate urmări în mod direct funcționarea corectă a acestui bloc.

5.3.1.3 Blocul de comandă al canalului HBridge (canalul 3)

Pentru exemplificarea funcționalității SSIHE ca un NCAP multiport (prin atașarea liniilor GPMI), am implementat canalul transductor HBridge, care permite comanda unei punți H cu tranzistoare MOS la care se poate atașa un motor de curent continuu (actuador). Structura acestui canal a fost expusă în paragraful 5.2.2.6, figura 5.17 iar secvența de cod VHDL aferentă în paragraful 5.2.2.9, figura 5.37.

În urma procesului de sinteză am obținut parametrii specifici implementării în FPGA a blocului de comandă HBridge, prezentați în cadrul tabelului 5.5.

Device utilization summary: 4vlx25ff668-10			
Logic utilization	Used	Available	Utilization
Number of Slices	2	10752	0%
Number of Slice Flip Flops	4	21504	0%
Number of 4 input LUTs	4	21504	0%
Number of bonded IOBs	12	448	2%
Number of GCLKs	1	32	3%

Tabelul 5.5. Gradul de utilizare al FPGA pentru sinteza logicii H-Bridge

Simularea funcționării blocului de comandă HBridge ne furnizează diagramele temporale prezentate în figura 5.52.

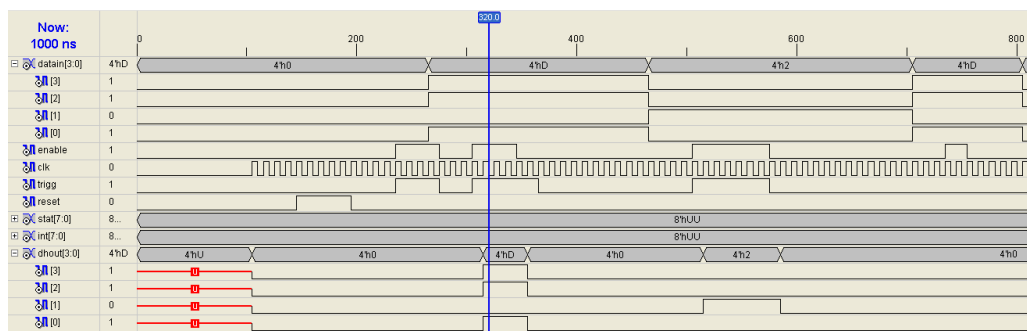


Fig.5.52. Operarea blocului de comandă HBridge

Registrul asociat canalului HBridge permite vehicularea datelor pe 4 biți (datain [3:0]). Acestea se transmit la cele patru linii de ieșire (dhout[3:0]) la momentul activării semnalului de declanșare/interogare „trigg” de către IPTIM, la fiecare front crescător al impulsului de tact al sistemului. Sunt luate în considerare și semnalele „enable „ de validare și „reset” de inițializare generate tot de către IPTIM.

Pentru exemplificarea principiului de operare prin diagramele expuse în figura 5.52, am ales valori logice arbitrare pentru datele de intrare datain[3:0].

5.3.1.4 Blocul de gestionare a datelor la nivelul canalelor transductor

Conform structurii IPTIM prezentate în paragraful 5.2.2.6, figura 5.8, am elaborat în VHDL blocul pentru extragerea și înscrierea datelor prin intermediul registrelor asociate canalelor transductor (TxHART, RxHART, HBridge). Am implementat două module, „TxFetcher” respectiv „RxFetcher” reunite în cadrul blocului „HARTFetcher”.

Registrul TxData va conține informația pe 32 de biți care este furnizată de către NCAP modulului IPTIM și urmează să fie serializată pe cadre de câte 8 biți de date. Această extragere pe cadre de 8 biți este realizată de către „TxFetcher”, la fiecare front crescător al impulsului de declanșare Trigger1 generat de către IPTIM, așa cum se poate deduce și din diagrama prezentată în figura 5.53. De exemplu, în cadrul acestei diagrame, putem surprinde momentul extragerii biților txdata[16:23] declanșat de către al treilea impuls al semnalului Trigger1 și înscris în registrul dataotx[0:7] care urmează a fi apoi serializat (reperul la poziția 27,6 ns). Urmează apoi extragerea ultimului cadru, txdata[24:31] așa cum se poate observa în figura 5.53 după care procesul se poate relua.

Operarea este corelată cu activarea semnalului „txdone” pentru a nu omite cadre din registrul TxData asociat canalului TxHART.

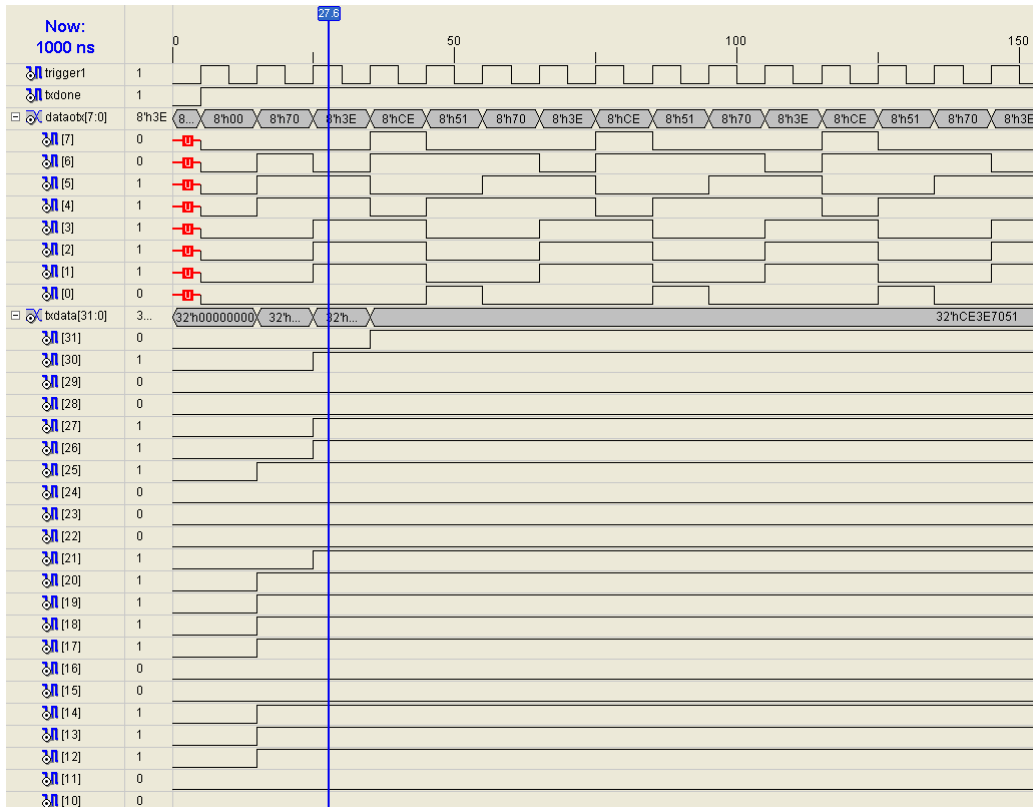


Fig.5.53. Operarea modulului „TxFetcher”

Registrul RxData va stoca cele patru cadre de câte 8 biți recepționate și deserializate de către blocul UART. Dispunând de datele pe 8 biți furnizate prin registrul „dataorx[7:0]”, modulul „RxFetcher” are rolul de a prelua acești biți și de ai repartiza în cadrul registrului „rxdata[31:0]” pe secvențe de câte 8 biți. Acest lucru are loc la fiecare front crescător al semnalului de interogare Trigger2 generat de către blocul IPTIM corelat cu momentul în care este activă și linia „rxavin” care marchează disponibilitatea datelor deserializate de către UART.

Pentru exemplificarea operării, s-a generat diagrama temporală expusă în figura 5.54, în cadrul căreia am impus un semnal de interogare (Trigger2) repetitiv, cu menținerea pe „1” logic a semnalului de validare a datelor „rxavin” și a unor valori constante pe o anumită perioadă ale registrului „dataorx[7:0]”. Astfel putem observa cu ușurință momentul înscrierii cadrelor pe 8 biți în registrul „rxdata[31:0]” prin urmărirea reperului de pe diagramă (la 137,3 ns) care indică preluarea cadrului aferent ciclului al patrulea de înscriere la pozițiile „rxdata[24:31]”.

În operarea reală, IPTIM va genera impulsurile de interogare „Trigger2” doar pe parcursul a patru cicluri pentru o activare a liniei „rxavin” după care se așteaptă o tranziție „1->0” pe această linie și apoi ciclul se reia.

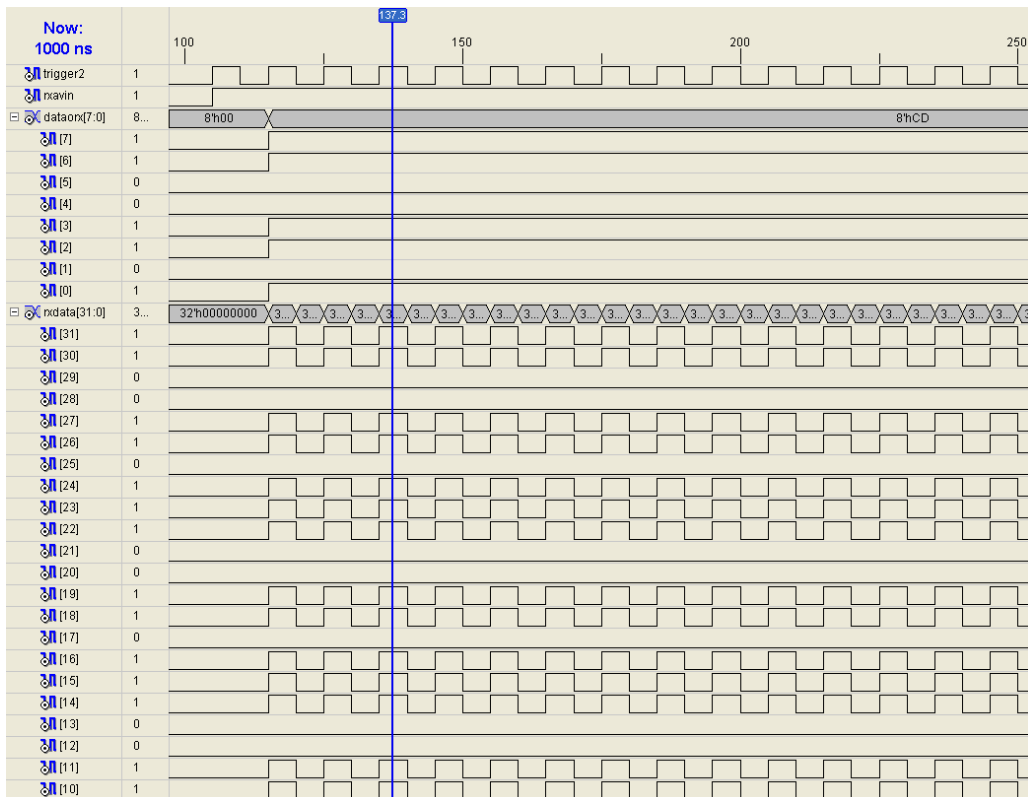


Fig.5.54. Operarea modului „RxFetcher”

Similar operării blocului „HARTFetcher” extragem datele pe patru biți din cadrul registrului Data3 aferente comenzii punții H implementate sub forma canalului transductor HBridge.

5.3.2. Testarea funcțională a SSIHE

În vederea determinării caracteristicilor funcționale ale SSIHE, este necesară parcurgerea a mai multor etape. Primordială este testarea posibilităților de comunicare cu dispozitivele de teren HART. În acest sens, am elaborat un emulator HART bazat pe un sistem flexibil, dezvoltat în jurul unui microcontroler PIC16F84, prezentat pe larg în lucrarea „**Signal Generating Techniques using PIC16F84 Microcontrollers**” [82].

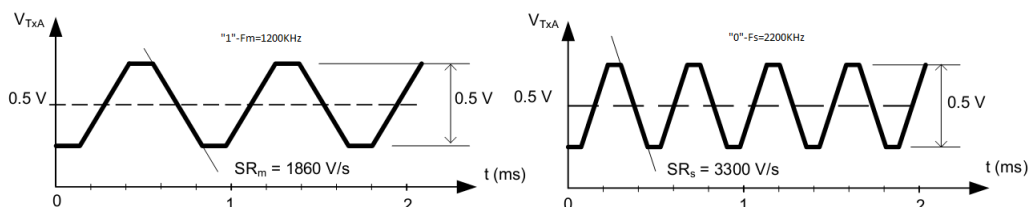


Fig.5.55. Semnale furnizate de către emulatorul HART

Utilizând principiile expuse în [82], am generat la ieșirea analogică dedicată, semnale trapezoidale, conform specificațiilor HART, de tipul celor prezentate în figura 5.55. Acestea au fost aduse, prin circuitele de formare adecvate, la intrarea RxHART a SSIHE și s-a verificat recepționarea corectă a valorilor „0” (space) respectiv „1” (mark).

În mod similar, la intrarea analogică dedicată a emulatorului HART, s-au aplicat semnalele trapezoidale generate de către ieșirea TxHART a SSIHE și apoi au fost contorizate valorile „0” respectiv „1” logic.

Pentru nivelul logic „1” am generat un semnal trapezoidal având factorul „slew rate” $SR_m = 1860 \text{ V/s}$ iar pentru nivelul logic „0” un semnal cu $SR_s = 3300 \text{ V/s}$.

Emulatorul HART conceput, bazat pe sistemul de dezvoltare cu microcontrolerul PIC16F84, este prezentat în figura 5.56.

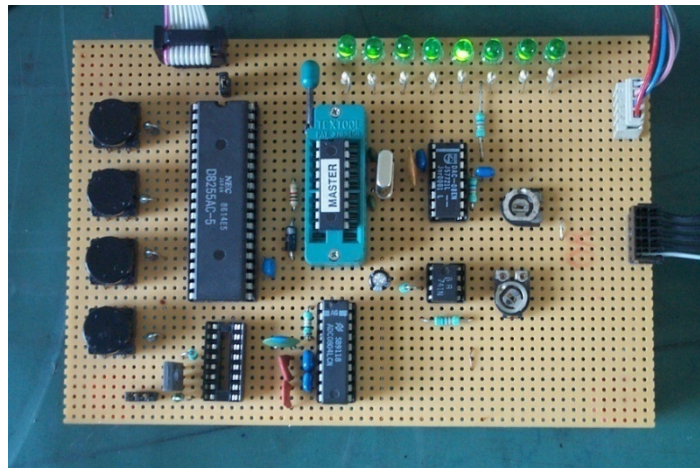


Fig.5.56. Emulator HART bazat pe sistem de dezvoltare cu PIC16F84

Următoarea etapă constă în elaborarea unor rutine în limbajul Visual C pentru testarea comunicării SSIHE pe rețeaua Ethernet. Aceste rutine vor rula pe un calculator gazdă și vor utiliza protocolul de comunicare UDP. Am elaborat două ferestre prin care utilizatorul va avea posibilitatea de a transmite către SSIHE comenzi HART respectiv să recepționeze valori ale variabilelor de proces provenite de la SSIHE prin citirea dispozitivelor de teren HART conectate.

Fereastra generată de către rutina de transmisie este prezentată în figura 5.57.

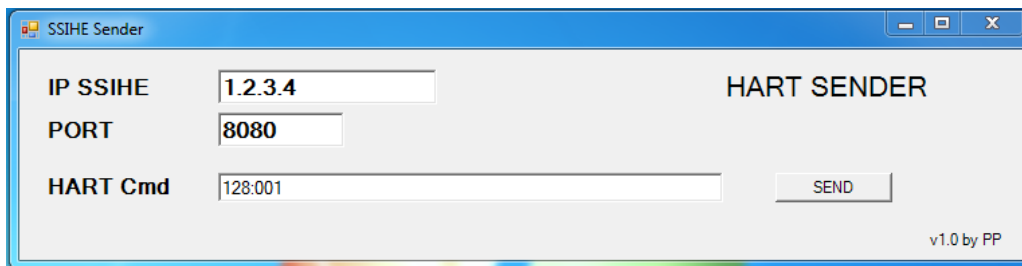


Fig.5.57. Fereastra de transmitere a unei comenzi HART

Fereastra „HART Sender” ne permite stabilirea adresei IP a SSIHE (câmpul IP SSIHE), portul de acces (câmpul PORT) precum și emiterea comenzii HART (câmpul HART Cmd).

Rutina de recepție interacționează cu utilizatorul prin intermediul ferestrei expuse în figura 5.58.

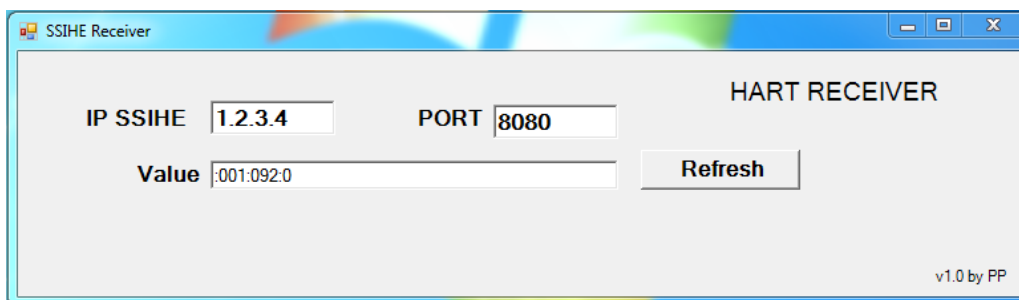


Fig.5.58. Fereastra de recepție HART

Fereastra „HART Receiver” ne oferă posibilitatea preluării prin rețeaua Ethernet a unei valori citite dintr-un dispozitiv de teren HART conectat la SSIHE, cu o anumită adresă. Valorile IP SSIHE și PORT sunt similare celor prezentate anterior.

În cadrul tezei am atins doar câteva etape necesare testării funcționale complete a SSIHE datorită necesității obținerii unor licențe de utilizare a diverselor blocuri predefinite și oferite de către firma Xilinx. Se propune totuși un algoritm de testare constând în transmiterea către dispozitivul de teren HART bazat pe un sistem cu microcontrolerul PIC18F4550 (figura 5.59) a unei variabile de proces care stabilește în procente factorul de umplere al unui semnal PWM de comandă a turației unui motor de curent continuu (FAN1). Valoarea preluată este afișată local (Fan1: xx %) și apoi se comandă motorul Fan1. În continuare, pentru testarea recepției, se preia de la dispozitivul de teren HART o variabilă de proces care va conține o valoare stabilită anterior, în procente, pentru factorul de umplere al semnalului PWM. Această valoare se va afișa în fereastra de control „SSIHE Receiver”.

Pentru a putea accesa dispozitivul de teren HART, se va stabili pentru acesta adresa „001”.

Urmează emiterea comenzii HART 129 care permite transmiterea către dispozitivul conectat a unei valori numerice de tip întreg, pe 8 biți. Structura comenzii va fi de forma: „129:001:036”, unde 129 reprezintă codul comenzii HART, 001 adresa perifericului HART iar 036 valoarea în procente a factorului de umplere.

Ca rezultat returnat de către dispozitivul HART adresat prin intermediul SSIHE, se va preconiza secvența „:001:036:0” care indică adresa dispozitivului (001), valoarea recepționată (036) și codul de eroare (0 – fără eroare).

În urma acestei comenzi, SSIHE va extrage comanda efectivă HART din cadrul Ethernet, va interpreta semnificația acestei comenzi și o va transmite modulată FSK dispozitivului de teren HART atașat, având adresa 001.

Ca răspuns, dispozitivul HART adresat va transmite către SSIHE codul de eroare și în continuare va genera semnalul PWM cu factorul de umplere 36% necesar comenzii motorului de curent continuu Fan1.

Adresa IP a SSIHE trebuie prestabilită la o valoare validă.

Ultima etapă propusă, constă în testarea recepționării de către calculatorul gazdă, prin rețeaua Ethernet, a informației provenite de la SSIHE. În acest sens,

prin comenzi locale la nivelul dispozitivului de teren HART, se prestabilește de exemplu o valoare de 92% pentru factorul de umplere al semnalului PWM.

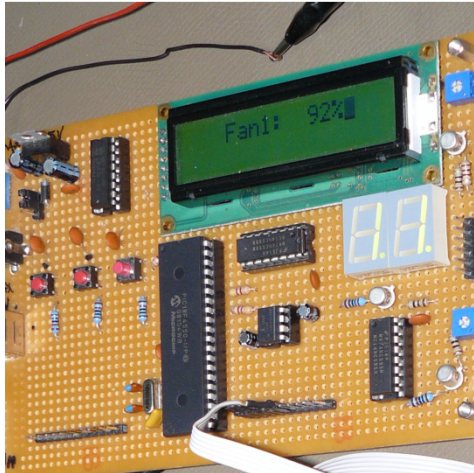


Fig.5.59. Sistem bazat pe microcontrolerul PIC18F4550

Emitem comanda HART 128 prin intermediul ferestrei „SSIHE Sender”, de pe calculatorul gazdă și vom urmări, în cazul funcționării corecte, obținerea răspunsului „:001:092:0” în fereastra SSIHE Receiver.

Semnificația răspunsului indică valoarea 092 provenită de la dispozitivul cu adresa 001, având codul de eroare 0.

Testarea funcțională a SSIHE în cadrul tezei, s-a rezumat strict la determinarea capacităților de comunicare între sistemul elaborat și un dispozitiv de teren HART, în cazul nostru emulat cu o structură bazată pe microcontrolerul PIC16F84. În plus, am elaborat rutinele în limbaj C care asigură baza software pentru comunicarea SSIHE cu un calculator gazdă, prin intermediul rețelei Ethernet.

5.4. Concluzii și contribuții personale

Considerând soluțiile actuale de interfațare a dispozitivelor HART cu rețeaua Ethernet, prezentate în Capitolul IV, paragraful 4.3.4, prin adoptarea sistemului propus se oferă posibilitatea simplificării acestei operații prin eliminarea anumitor dispozitive auxiliare, cu precădere a lanțului de modemuri RS232-Ethernet-RS485-HART precum și reducerea semnificativă a costurilor de implementare. Astfel, prin particularitățile SSIHE expuse în acest capitol, se poate indica modul de interfațare a dispozitivelor HART cu rețeaua Ethernet prin utilizarea SSIHE cu ajutorul topologiei prezentate în figura 5.60.

Structura expusă în figura 5.60 prezintă posibilitatea de conectarea a mai multor SSIHE pe o rețea industrială Ethernet, prin intermediul unui distribuitor de tip switch, agreat în majoritatea aplicațiilor industriale (vezi Capitolul III).

Fiecare SSIHE permite conectarea atât a unui număr variabil de dispozitive de teren HART cât și a unor transductoare non HART prin intermediul interfeței dedicate GPMI.

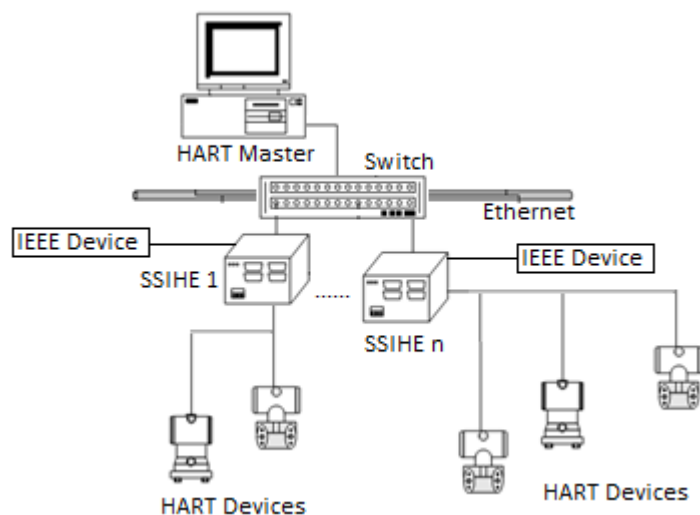


Fig.5.60. Topologia HART – Ethernet utilizând SSIHE

Procesul de proiectare a subsistemului inteligent de interfațare HART – Ethernet (SSIHE), prezentat pe parcursul Capitolului V, a impus atât conceperea într-un limbaj de nivel înalt de sinteză hardware (în cazul nostru VHDL) a blocurilor funcționale cât și elaborarea unor rutine software de comandă a acestor blocuri. Bazându-se pe o structură NCAP compatibilă cu standardul IEEE 1451, sistemul propus conține un nucleu alcătuit dintr-un procesor soft de tip Microblaze care permite rularea rutinelor de comandă scrise în limbaj C/C++. SSIHE conține atât blocuri standard care implementează anumite funcții cum ar fi comunicarea pe rețeaua Ethernet, gestionarea memoriilor externe, arbitrarea întreruperilor cât și blocuri originale, concepute de către autor în vederea implementării exclusiv a noilor funcții, caracteristice SSIHE.

Etapele aferente implementării subsistemului propus au fost următoarele:

- i) selectarea suportului fizic, de tip FPGA, pentru implementarea SSIHE;
- ii) delimitarea blocurilor hardware – software din componența SSIHE și stabilirea modului de implementare a acestora;
- iii) elaborarea structurii blocurilor componente ale SSIHE;
- iv) stabilirea configurației interne a SSIHE ca „Solution on a Chip”;
- v) integrarea elementelor hardware – software în structura SSIHE;
- vi) sinteza elementelor SSIHE originale și testarea funcțională.

5.4.1. Contribuții personale

Capitolul V reunește atât aspecte clasice cu privire la implementarea unor blocuri funcționale larg utilizate în domeniul de specialitate cât și numeroase idei novatoare în conceperea și testarea unor module cu funcții specifice aplicației propuse în lucrare. Astfel, putem sintetiza următoarele contribuții personale:

1. **Selectarea platformei sistem ML-401 cu FPGA Virtex 4** pentru implementarea SSIHE;
2. **Identificarea corespondenței dintre perifericele** oferite de către platforma ML-401 și **blocurile funcționale ale SSIHE;**

3. **Conceperea diagramei bloc a SSIHE;**
4. **Stabilirea soluțiilor de integrare** pe un singur circuit FPGA a structurii SSIHE;
5. **Alegerea tipului de magistrală internă și exploatarea facilităților** oferite de către modulele IPIF predefinite;
6. **Particularizarea fiecărui modul IPIF funcție de cerințele impuse** de către blocurile funcționale ale SSIHE;
7. **Conceperea structurii modului TIM intern (IPTIM);**
8. **Oferirea unor soluții originale de integrare** a blocurilor UART și HART Modem în structura IPTIM;
9. **Propunerea introducerii interfeței GPMI** în structura SSIHE;
10. **Elaborarea configurației** canalelor transductor dedicate dispozitivelor HART și a celor aferente transductoarelor generice IEEE 1451;
11. **Alcătuirea logicii de comandă** aferente fiecărui canal transductor în parte;
12. **Conceperea secvențelor în VHDL** pentru descrierea blocurilor UART, HART Modem, HARTFetcher și H-Bridge;
13. **Implementarea** cu ajutorul Xilinx EDK/XPS a **structurii interne** a SSIHE, cu **includerea** nucleului Microblaze;
14. **Adaptarea/elaborarea rutinelor software** în limbajul C/C++, executate de către nucleul Microblaze;
15. **Conceperea structurii vectorilor de testare** în vederea simulării funcționale a blocurilor concepute ale SSIHE;
16. **Interpretarea** diagramelor și a rezultatelor obținute;
17. **Proiectarea și realizarea** unor module hardware auxiliare, necesare testării funcționale a SSIHE (bazate pe microcontrolere din familia PIC);
18. **Alcătuirea unui software** de comunicare între un calculator gazdă (HART Master) și SSIHE, prin intermediul rețelei Ethernet.

Pe parcursul procesului de proiectare a subsistemului inteligent, am apelat la anumite rezultate preliminare, valorificate prin intermediul lucrărilor **[80]** și **[82]**, citate în cadrul acestui capitol.

6. CONCLUZII FINALE

Conducerea proceselor tehnologice industriale implică operații complexe și acoperă domenii de știință multidisciplinare. Se pot contura însă câteva direcții principale de cercetare prin care se pot introduce anumite elemente novatoare privind îmbunătățirea performanțelor și reducerea costurilor de implementare a sistemelor implicate în acest domeniu.

6.1. Sinteza globală a contribuțiilor personale

Teza de față vine astfel în sprijinul proiectanților sistemelor de conducere a proceselor tehnologice industriale prin abordarea a trei direcții de cercetare: **clasificarea sistemelor de comunicație industriale pe baza celor mai noi standarde cu determinarea parametrilor care influențează semnificativ performanțele rețelelor de teren, analiza și predicția prin metode statistice a acestor performanțe și conceperea unui subsistem inteligent** a cărui utilizare asigură optimizarea conducerii unor procese tehnologice industriale.

Pornind de la anumite considerente actuale referitoare la conducerea proceselor tehnologice industriale, în Capitolul II se urmărește conturarea unei viziuni de ansamblu asupra structurilor fundamentale de vehiculare a datelor în sistemele de comunicație industriale. În acest sens sun de menționat următoarele contribuții personale:

1. **Identificarea problematicii actuale** în alegerea și utilizarea sistemelor de comunicație industriale;
2. **Ierarhizarea după cele mai noi criterii** a tipurilor de rețele de teren;
3. **Sintetizarea principalelor cerințe** referitoare la caracteristicile fundamentale ale rețelelor de teren;
4. **Clasificarea generală** a principalelor tipuri de rețele de teren după criteriile de importanță majoră din punctul de vedere al autorului;
5. **Expunerea soluțiilor adoptate în prezent** pentru implementarea standardului IEC61784-2.

Prin introducerea acestor noțiuni se prefațează posibilitatea elaborării unei metode de analiză și evaluare a performanțelor rețelelor de teren existente la ora actuală.

În continuare se efectuează o analiză statistică prin aplicarea metodei „D-Optimal” în cadrul unei proiectări experimentale de tip DOE (Design of Experiment). Capitolul III tratează pe larg modalitatea obținerii în urma efectuării a unui număr restrâns de experimente a unui model statistic de predicție a eficienței de transmisie și a întârzierii introduse în rețeaua de comunicație. Au fost definiți doi parametri care descriu acești indicatori și s-a evaluat efectul a cinci factori de influență asupra acestora. În urma acestor analize rezultă informații utile care stau la baza obținerii unor soluții optime pentru îmbunătățirea performanțelor globale ale rețelelor de teren, sintetizându-se următoarele contribuții personale:

1. **Identificarea**, pe baza unei analize anterioare (Capitolul II), a **principalilor factori de influență și indicatori de performanță** corespunzători rețelelor de teren conforme standardului IEC 61784;
2. **Aplicarea proiectării experimentale** (DOE) bazate pe metoda D-Optimal la analiza rețelelor de teren, fapt nementionat la ora actuală în literatura de specialitate;
3. **Selectarea și utilizarea** unui software de analiză și modelare statistică pe calculator, care s-a dovedit a fi suficient de performant pentru scopul urmărit: SAS JMP;
4. **Transpunerea datelor** inițiale în tabele aferente JMP;
5. **Efectuarea experimentelor** impuse în urma aplicării metodei D-Optimal cu ajutorul mediului de simulare OpNet;
6. **Urmărirea simultană a doi indicatori de performanță** definiți prin eficiența transmisiei și întârzierea din rețea;
7. **Stabilirea legii de distribuție și determinarea expresiei matematice** a funcției densitate de probabilitate asociată celor doi factori de răspuns concretizate prin relațiile (3.16) și (3.21);
8. **Generarea și interpretarea** diagramelor care indică atât efectele principale cât și cele de interacțiune ale factorilor de influență, asupra variabilelor de răspuns, prezentate în figurile 3.74, 3.75, 3.76, 3.77, 3.79 și 3.80;
9. **Conceperea unui „tabel de influență”**, cu notații intuitive, care descrie caracterul efectelor introduse de către factorii de influență (tabelul 3.12);
10. **Elaborarea expresiilor modelelor** pentru factorii de răspuns eficiența transmisiei și întârziere, descrise de relațiile (3.26) și (3.30), prin introducerea unor descriptori matriciali ai variabilelor independente (factorii de influență). Prin utilizarea acestor forme matriciale, proiectantul are posibilitatea de a stabili extrem de simplu structura rețelei pentru care dorește predicția indicatorilor de performanță;
11. **Verificarea gradului de potrivire** al acestor modele, atât cu ajutorul a două metode grafice cât și prin interpretarea parametrilor statistici ce exprimă erorile de predicție.

Conceptul actual al transductorului inteligent (smart transducer) a condus la perceperea acestuia ca mai mult decât elementul simplu, care transformă o formă de energie în alta, conferindu-i proprietatea de a asigura comunicarea bidirecțională cu o unitate de control ierarhic superioară.

Standardul IEEE 1451 introduce elemente cum ar fi: transductorul inteligent, împreună cu noțiuni ca TIM (Transducer Interface Module), TII (Transducer Independent Interface), TEDS (Transducer Electronic Data Sheet) și NCAP (Network Capable Application Processor), ele constituind o abordare inovatoare a întregii problematice referitoare la conducerea proceselor tehnologice.

Migrarea către noul standard este preocuparea curentă în domeniu, iar în Capitolul IV **s-a expus o modalitate originală**, prin conceperea unui „subsistem inteligent” care oferă o cale de tranziție către noul standard cu costuri minime datorită posibilității utilizării în paralel atât a vechilor dispozitive cât și a celor noi.

Sistemul clasic HART de comunicare cu dispozitivele de control al proceselor tehnologice, larg răspândit la ora actuală în diverse arii de activitate suportă îmbunătățiri structurale și funcționale care au format obiectul cercetării în ultimele două capitole de dezvoltare ale tezei.

Modalitatea propusă urmărește compatibilizarea dispozitivelor de teren HART existente cu cerințele noului standard IEEE 1451, fără o modificare

constructivă a acestora. Astfel **s-a conceput o „punte de legătură”**, materializată prin intermediul unui subsistem inteligent, bazat pe principiile introduse de către standardul IEEE 1451. Nucleul subsistemului îl constituie un bloc NCAP în combinație cu un modul TIM înglobat în structura generală, împreună cu circuitele necesare comunicării cu dispozitivele de teren HART. În plus, prin intermediul NCAP, noul sistem va avea capacitatea de a comunica pe rețeaua Ethernet cu alte dispozitive de control compatibile IEEE 1451.

Elaborarea subsistemului inteligent de interfațare HART – Ethernet (SSIHE) se bazează pe următoarele contribuții personale:

1. **Sintetizarea principalelor concepte actuale** referitoare la structura și cerințele impuse sistemelor de conducere a proceselor tehnologice;
2. **Identificarea elementelor** care necesită o anumită restructurare în scopul îmbunătățirii performanțelor funcționale;
3. **Expunerea printr-o viziune proprie**, ilustrativă și concentrată, a specificațiilor standardului IEEE 1451;
4. **Propunerea unui „subsistem inteligent”**, destinat compatibilizării dispozitivelor de teren HART cu cerințele standardului IEEE 1451;
5. **Elaborarea unor metode software originale** de modelare și implementare a informațiilor conținute în blocul TEDS;
6. **Utilizarea microcontrolerelor din familia PIC18F44x** ca suport hardware și metodele software de la pct.5, pentru conceperea modulelor TIM;
7. **Studiul soluțiilor actuale de implementare HART** pe o rețea Ethernet;
8. **Conceperea structurii funcționale a unui subsistem inteligent** bazat pe un bloc NCAP și un modul TIM;
9. **Introducerea interfeței GPMI** (General Purpose Multichannel Interface), cu denumirea propusă de către autor;
10. **Elaborarea modelului informațional NCAP** cu ajutorul limbajului UML și pe baza specificațiilor existente în literatura de specialitate;
11. **Identificarea operațiilor necesare implementării comunicației Client – Server** în cadrul subsistemului inteligent;
12. **Definirea modelului funcțional al blocului transductor**, necesar implementării comunicației cu dispozitivele de teren HART la nivelul NCAP;
13. **Propunerea unei metode de înglobare a comenzilor HART** în cadrele Ethernet, care nu periclitează compatibilitatea față de protocoalele Ethernet existente;
14. **Conceperea organigramei** care descrie funcționalitatea subsistemului inteligent la nivel global.

Capitolul V **expune modul de implementare efectivă și testarea funcțională a structurilor bloc ale SSIHE**, cu ajutorul unui circuit FPGA de tip Virtex 4, produs de către Xilinx, inclus pe o placă de dezvoltare ML-401 precum și testarea funcțională a blocurilor componente.

Subsistemul inteligent a fost implementat bazându-ne pe conceptul de „Solution on a chip” – SoC (soluție pe un singur cip) care utilizează pe deplin avantajele oferite de circuitele FPGA moderne.

Procesul de proiectare al subsistemului inteligent de interfațare HART – Ethernet (SSIHE), prezentat pe parcursul Capitolului V, a impus atât conceperea într-un limbaj de nivel înalt de sinteză hardware (în cazul nostru VHDL) a blocurilor funcționale cât și elaborarea unor rutine software de comandă a acestor blocuri. Bazându-se pe o structură NCAP compatibilă cu standardul IEEE 1451, sistemul propus conține un nucleu alcătuit dintr-un procesor soft de tip Microblaze care

permite rularea rutinelor de comandă scrise în limbaj C/C++. Au fost luate în considerare atât aspecte clasice cu privire la implementarea unor blocuri funcționale larg utilizate în domeniul de specialitate cât și numeroase idei novatoare în conceperea și testarea unor module cu funcții specifice aplicației propuse în lucrare, rezultând de aici următoarele contribuții personale:

1. **Selectarea platformei sistem ML-401 cu FPGA Virtex 4** pentru implementarea SSIHE;
2. **Identificarea corespondenței dintre perifericele** oferite de către platforma ML-401 și **blocurile funcționale ale SSIHE**;
3. **Conceperea diagramei bloc a SSIHE**;
4. **Stabilirea soluțiilor de integrare** pe un singur circuit FPGA a structurii SSIHE;
5. **Alegerea tipului de magistrală internă și exploatarea facilităților** oferite de către modulele IPIF predefinite;
6. **Particularizarea fiecărui** modul IPIF funcție de **cerințele impuse** de către blocurile funcționale ale SSIHE;
7. **Conceperea structurii modului TIM** intern (IPTIM);
8. **Oferirea unor soluții originale de integrare** a blocurilor UART și HART Modem în structura IPTIM;
9. **Propunerea introducerii interfeței GPMI** în structura SSIHE;
10. **Elaborarea configurației** canalelor transductor, dedicate dispozitivelor HART și a celor aferente transductoarelor generice IEEE 1451;
11. **Alcătuirea logicii de comandă** aferente fiecărui canal transductor în parte;
12. **Conceperea secvențelor în VHDL** pentru descrierea blocurilor UART, HART Modem, HARTFetcher și H-Bridge;
13. **Implementarea** cu ajutorul Xilinx EDK/XPS a **structurii interne** a SSIHE, cu **includerea** nucleului Microblaze;
14. **Adaptarea/elaborarea unor rutine software** în limbajul C/C++, executate de către nucleul Microblaze;
15. **Conceperea structurii vectorilor de testare** în vederea simulării funcționale a blocurilor concepute ale SSIHE;
16. **Interpretarea** diagramelor și a rezultatelor obținute;
17. **Proiectarea și realizarea** unor module hardware auxiliare, necesare testării funcționale a SSIHE (bazate pe microcontrolere din familia PIC);
18. **Alcătuirea unui software** de comunicare între un calculator gazdă (HART Master) și SSIHE, prin intermediul rețelei Ethernet.

Prin adoptarea sistemului propus se oferă posibilitatea simplificării interfațării HART - Ethernet prin eliminarea anumitor dispozitive auxiliare, cu precădere a lanțului de modemuri RS232-Ethernet-RS485-HART precum și reducerea semnificativă a costurilor de implementare. Fiecare SSIHE permite conectarea atât a unui număr variabil de dispozitive de teren HART cât și a unor transductoare non HART prin intermediul interfeței dedicate GPMI (General Purpose Multichannel Interface).

Rezultatele expuse în cadrul tezei sunt consolidate și prin anumite eforturi depuse de către autor în prealabil, vizând domeniile de cercetare menționate. În acest sens a fost elaborat un număr de **opt articole** publicate în **reviste și volumele unor conferințe naționale și internaționale**, citate în conținutul tezei și marcate prin caractere accentuate (bold), după cum urmează: [1], [2], [51], [54], [55]-ISI, [62]-ISI, [80]-ISI și [82].

6.2. Direcții de cercetare viitoare

Domeniul vast al automatizărilor industriale împreună cu diversele problematice pe care le generează, conduce în mod implicit la existența numeroaselor soluții suplimentare de optimizare a conducerii proceselor tehnologice industriale. De aici derivă și câteva preocupări viitoare ale autorului, axate pe aceleași direcții de cercetare.

O primă dezvoltare rezultă din structura primară a SSIHE, care pe viitor se poate restructura, eliminând în totalitate circuitele necesare realizării modulației FSK, păstrând în schimb formatul comenzilor HART. Se va obține astfel o interfață complet digitală, capabilă să execute și să retransmită comenzi HART, bazată pe o comunicare strict digitală. Autorul propune astfel un nou standard având denumirea novatoare „**Smart HART**”.

O direcție de cercetare suplimentară va consta și în reproiectarea circuitelor de interfațare a dispozitivelor de teren HART clasice pentru a corespunde noului standard „Smart HART”.

O ultimă abordare a problematicei expuse va consta în studiul posibilităților de introducere a unor proprietăți de reconfigurare automată a unor structuri interne ale SSIHE pentru atingerea flexibilității maxime conform cerințelor impuse de modificările operațiilor de conducere a proceselor tehnologice industriale.

ANEXA 1 - Valorile obținute pentru experimentele 1-14

TRAFFIC E1-E5

t(s)	E1-PT (p/s)	E1-PR (p/s)	E2-PT (p/s)	E2-PR (p/s)	E3-PT (p/s)	E3-PR (p/s)	E4-PT (p/s)	E4-PR (p/s)	E5-PT (p/s)	E5-PR (p/s)
0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0
72	0,72222222	0,22222222	0,55555556	0,22222222	0,11111111	0,11111111	0,11111111	0,11111111	0,11111111	0,11111111
108	7,05555556	2,88888889	6,55555556	2,72222222	7,05555556	2,88888889	6,30555556	2,80555556	8,69444444	3,02777778
144	0,55555556	0,22222222	0,94444444	0,44444444	0,58333333	0,25	0,41666667	0,25	0,80555556	0,47222222
180	0,66666667	0,33333333	0,33333333	0,16666667	0,25	0,08333333	0,27777778	0,11111111	1,56666667	0,5
216	0,77777778	0,27777778	1,11111111	0,44444444	0,72222222	0,22222222	0,25	0,25	0,75	0,25
252	0,33333333	0,33333333	0,5	0,16666667	1,16666667	0,5	2,11111111	0,61111111	0,83333333	0,33333333
288	0,61111111	0,27777778	1,27777778	0,44444444	0,77777778	0,27777778	0,63888889	0,30555556	0,30555556	0,13888889
324	0,22222222	0,22222222	1,83333333	0,5	1,11111111	0,27777778	1,11111111	0,27777778	0,27777778	0,11111111
360	0,33333333	0	0,55555556	0,22222222	0,44444444	0,11111111	0,52777778	0,19444444	0,33333333	0,16666667
396	0,33333333	0,16666667	0,22222222	0,05555556	0,94444444	0,44444444	0,91666667	0,25	0,38888889	0,22222222
432	0,11111111	0,11111111	0,55555556	0,38888889	0,88888889	0,22222222	0,55555556	0,22222222	0,80555556	0,47222222
468	0,5	0,16666667	0,66666667	0,33333333	0,33333333	0,16666667	0,66666667	0,33333333	0,80555556	0,47222222
504	0,72222222	0,38888889	0,33333333	0,16666667	0,72222222	0,22222222	1,55555556	0,72222222	0,77777778	0,27777778
540	0,94444444	0,27777778	0,94444444	0,44444444	0,77777778	0,27777778	0,69444444	0,36111111	0,41666667	0,25
576	0,16666667	0,16666667	0,27777778	0,11111111	0,58333333	0,25	1,02777778	0,36111111	1,08333333	0,41666667
612	0,16666667	0,16666667	0,88888889	0,38888889	0,80555556	0,30555556	0,75	0,25	0,33333333	0,16666667
648	0,33333333	0,16666667	0,27777778	0,11111111	0,72222222	0,38888889	0,77777778	0,27777778	0,36111111	0,19444444
684	0,27777778	0,11111111	0,94444444	0,44444444	0,83333333	0,33333333	0,27777778	0,11111111	0,63888889	0,30555556
720	0,66666667	0,33333333	0,83333333	0,33333333	0,75	0,25	0,83333333	0,33333333	0,55555556	0,22222222
756	0,55555556	0,05555556	0,88888889	0,38888889	0,36111111	0,19444444	0,47222222	0,13888889	1,16666667	0,33333333
792	0,77777778	0,27777778	0,44444444	0,11111111	0,08333333	0,08333333	0,72222222	0,38888889	0,83333333	0,33333333
828	0,88888889	0,22222222	0,72222222	0,22222222	0,47222222	0,13888889	0,75	0,25	0,27777778	0,11111111
864	1,05555556	0,38888889	0,55555556	0,22222222	0,63888889	0,30555556	0,58333333	0,25	0,13888889	0,13888889
900	0,77777778	0,27777778	1,44444444	0,44444444	1,80555556	0,47222222	0,66666667	0,16666667	0,86111111	0,36111111
936	0,33333333	0,16666667	1,05555556	0,38888889	1,05555556	0,38888889	0,52777778	0,36111111	0,36111111	0,19444444
972	0,66666667	0,33333333	0,44444444	0,27777778	0,38888889	0,22222222	0,13888889	0,47222222	0,47222222	0,13888889
1008	0,27777778	0,11111111	0,55555556	0,22222222	1,13888889	0,47222222	0,47222222	0,30555556	0,55555556	0,22222222
1044	0,22222222	0,05555556	0,61111111	0,27777778	0,80555556	0,30555556	0,30555556	0,13888889	0,63888889	0,30555556
1080	0,94444444	0,27777778	0	0	0,05555556	0,05555556	0,30555556	0,13888889	0,13888889	0,25
1116	1,44444444	0,44444444	1,16666667	0,5	0,61111111	0,27777778	0,55555556	0,22222222	0,41666667	0,25

TRAFFIC E1-E5

t(s)	E1-PT (p/s)	E1-PR (p/s)	E2-PT (p/s)	E2-PR (p/s)	E3-PT (p/s)	E3-PR (p/s)	E4-PT (p/s)	E4-PR (p/s)	E5-PT (p/s)	E5-PR (p/s)
1152	0,555555556	0,222222222	1,111111111	0,444444444	0,833333333	0,333333333	0,944444444	0,444444444	1,222222222	0,555555556
1188	0,888888889	0,222222222	0,333333333	0,166666667	0,694444444	0,194444444	0,972222222	0,305555556	0,083333333	0,083333333
1224	0,166666667	0,166666667	0,5	0,333333333	0,5	0,166666667	1,111111111	0,444444444	1,444444444	0,444444444
1260	0,888888889	0,388888889	0,5	0,166666667	0,138888889	0,138888889	0,111111111	0,111111111	0,888888889	0,388888889
1296	0,722222222	0,222222222	0,333333333	0,166666667	0,222222222	0,055555556	0,777777778	0,277777778	0,333333333	0,166666667
1332	0,833333333	0,333333333	0,5	0,166666667	0,527777778	0,194444444	0,222222222	0,055555556	0,666666667	0,333333333
1368	0,333333333	0,166666667	0,222222222	0,055555556	0,333333333	0,166666667	0,5	0,166666667	0,583333333	0,25
1404	0,944444444	0,277777778	0,111111111	0,111111111	0,805555556	0,305555556	1,055555556	0,388888889	0,888888889	0,388888889
1440	0,833333333	0,333333333	0,888888889	0,388888889	0,833333333	0,333333333	0,861111111	0,361111111	0,222222222	0,055555556
1476	0,166666667	0,5	0,833333333	0,333333333	0,444444444	0,111111111	0,472222222	0,138888889	0,25	0,083333333
1512	0,722222222	0,388888889	0,111111111	0,111111111	1,138888889	0,472222222	0,555555556	0,222222222	0,861111111	0,361111111
1548	1,055555556	0,388888889	0,555555556	0,222222222	0,805555556	0,305555556	0,361111111	0,194444444	0,611111111	0,277777778
1584	0,444444444	0,277777778	0,333333333	0,166666667	0,944444444	0,444444444	0,388888889	0,222222222	0,805555556	0,305555556
1620	1,111111111	0,444444444	0,666666667	0,333333333	0,805555556	0,305555556	0,138888889	0,138888889	1,194444444	0,361111111
1656	1	0,333333333	0,555555556	0,222222222	1,361111111	0,527777778	0,222222222	0,055555556	1,138888889	0,472222222
1692	1,111111111	0,444444444	0,333333333	0,333333333	0,472222222	0,305555556	0,888888889	0,388888889	0,333333333	0,166666667
1728	0,555555556	0,222222222	0,111111111	0,111111111	0,472222222	0,138888889	0,611111111	0,277777778	0,472222222	0,305555556
1764	0,666666667	0,166666667	0,666666667	0,333333333	1,055555556	0,555555556	0,611111111	0,277777778	1,222222222	0,388888889
1800	0,722222222	0,361111111	0,611111111	0,277777778	0,888888889	0,222222222	1,388888889	0,555555556	0,305555556	0,138888889
1836	0,444444444	0,138888889	1,055555556	0,388888889	1,194444444	0,361111111	0,277777778	0,111111111	0,555555556	0,222222222
1872	0,5	0,166666667	0,944444444	0,444444444	0,388888889	0,222222222	1	0,333333333	0,333333333	0,166666667
1908	1,277777778	0,444444444	0,111111111	0,111111111	0,472222222	0,138888889	1,694444444	0,694444444	0,833333333	0,333333333
1944	1	0,333333333	0,5	0,166666667	0,388888889	0,222222222	1,25	0,416666667	2	0,5
1980	0,277777778	0,111111111	0,777777778	0,277777778	0,305555556	0,138888889	0,527777778	0,194444444	0,888888889	0,388888889
2016	0,388888889	0,222222222	0,833333333	0,333333333	1,416666667	0,416666667	0,583333333	0,25	0,888888889	0,222222222
2052	0,5	0,166666667	1,166666667	0,333333333	0,805555556	0,305555556	0,305555556	0,138888889	0,555555556	0,222222222
2088	0,611111111	0,277777778	0,555555556	0,194444444	0,833333333	0,333333333	0,861111111	0,361111111	0,694444444	0,194444444
2124	0,5	0,166666667	0,222222222	0,083333333	0,75	0,25	0,611111111	0,277777778	0,75	0,25
2160	0,333333333	0,166666667	0,611111111	0,277777778	0,888888889	0,388888889	0,055555556	0,055555556	0,833333333	0,333333333
2196	1,111111111	0,444444444	0,277777778	0,111111111	0,805555556	0,305555556	0,388888889	0,222222222	0,055555556	0,055555556
2232	0,055555556	0,277777778	0,444444444	0,277777778	0,583333333	0,25	0,416666667	0,25	0,277777778	0,111111111
2268	1,222222222	0,388888889	0,722222222	0,388888889	0,055555556	0,055555556	0,916666667	0,166666667	0,333333333	0,166666667

208 Anexa 1 Valorile obținute pentru experimentele 1-14

TRAFFIC E1-E5

t(s)	E1-PT (p/s)	E1-PR (p/s)	E2-PT (p/s)	E2-PR (p/s)	E3-PT (p/s)	E3-PR (p/s)	E4-PT (p/s)	E4-PR (p/s)	E5-PT (p/s)	E5-PR (p/s)
2304	0,61111111	0,27777778	0,55555556	0,22222222	0,36111111	0,19444444	0,52777778	0,19444444	0,5	0,33333333
2340	0,27777778	0,11111111	0,33333333	0,16666667	1,11111111	0,44444444	0,91666667	0,41666667	0,86111111	0,36111111
2376	0,16666667	0,16666667	0,55555556	0,22222222	0,08333333	0,08333333	0,55555556	0,22222222	0,63888889	0,30555556
2412	0,55555556	0,22222222	0,5	0,33333333	0,77777778	0,27777778	0,83333333	0,33333333	0,75	0,25
2448	0	0	1	0,33333333	0,77777778	0,11111111	1,27777778	0,44444444	1,13888889	0,47222222
2484	0,55555556	0,38888889	0,88888889	0,38888889	0,47222222	0,13888889	0,36111111	0,19444444	0,13888889	0,13888889
2520	0,44444444	0,11111111	1,27777778	0,44444444	1	0,33333333	0,66666667	0,33333333	1,19444444	0,36111111
2556	0,44444444	0,11111111	0,72222222	0,22222222	0,11111111	0,11111111	0,41666667	0,25	0,55555556	0,22222222
2592	0,05555556	0,05555556	0,55555556	0,38888889	0,27777778	0,11111111	0,77777778	0,44444444	0,33333333	0,16666667
2628	1	0,33333333	0,83333333	0,33333333	0,47222222	0,13888889	1,02777778	0,36111111	1,41666667	0,41666667
2664	0,94444444	0,44444444	0,5	0,16666667	1	0,33333333	0,86111111	0,36111111	0,77777778	0,44444444
2700	0,88888889	0,38888889	0,72222222	0,22222222	0,97222222	0,47222222	0,5	0,16666667	0,86111111	0,36111111
2736	0,61111111	0,44444444	0,88888889	0,38888889	0,86111111	0,19444444	0,88888889	0,22222222	0,61111111	0,27777778
2772	0,72222222	0,22222222	0,72222222	0,22222222	0,22222222	0,05555556	1	0,33333333	0,61111111	0,27777778
2808	0,94444444	0,27777778	1,11111111	0,27777778	0,13888889	0,13888889	1,25	0,41666667	0,44444444	0,27777778
2844	0,55555556	0,22222222	0,72222222	0,22222222	1,36111111	0,36111111	0,72222222	0,22222222	0,66666667	0,16666667
2880	0,72222222	0,22222222	1,61111111	0,44444444	1,25	0,41666667	0,97222222	0,30555556	0,05555556	0,05555556
2916	0,55555556	0,22222222	0,27777778	0,11111111	0,52777778	0,19444444	0,33333333	0,16666667	1,22222222	0,38888889
2952	0,22222222	0,22222222	0,55555556	0,22222222	0,58333333	0,33333333	0,83333333	0,33333333	0,75	0,25
2988	0,05555556	0,05555556	0,05555556	0,05555556	1,05555556	0,38888889	0,5	0,16666667	0,55555556	0,22222222
3024	0,83333333	0,33333333	0,66666667	0,33333333	0,55555556	0,22222222	0,83333333	0,5	0,52777778	0,19444444
3060	0,66666667	0,33333333	0,16666667	0,16666667	0,19444444	0,27777778	0,66666667	0,16666667	1,13888889	0,30555556
3096	0,27777778	0,11111111	0,61111111	0,27777778	0,72222222	0,22222222	0,30555556	0,30555556	0,25	0,08333333
3132	0,38888889	0,22222222	0,22222222	0,22222222	0,47222222	0,13888889	1,13888889	0,30555556	0,61111111	0,27777778
3168	0,66666667	0,16666667	0,61111111	0,27777778	1,41666667	0,41666667	0,88888889	0,38888889	0,97222222	0,47222222
3204	0,33333333	0,16666667	0,77777778	0,27777778	0,05555556	0,05555556	0,33333333	0,16666667	0,75	0,25
3240	0,77777778	0,27777778	0,61111111	0,27777778	0,55555556	0,22222222	0,44444444	0,27777778	0,41666667	0,25
3276	0,94444444	0,27777778	1,16666667	0,33333333	0,55555556	0,22222222	0,91666667	0,25	0,58333333	0,25
3312	0,52777778	0,33333333	0,72222222	0,22222222	0,88888889	0,22222222	1,33333333	0,5	0,33333333	0,16666667
3348	0,58333333	0,27777778	0,5	0,16666667	1	0,33333333	0,36111111	0,19444444	0,36111111	0,19444444
3384	0,55555556	0,22222222	0,72222222	0,22222222	0,44444444	0,27777778	1,5	0,5	1,13888889	0,47222222
3420	0,66666667	0,33333333	0,83333333	0,33333333	1,44444444	0,44444444	0,33333333	0,16666667	0,55555556	0,22222222

TRAFFIC E1-E5

t(s)	E1-PT (p/s)	E1-PR (p/s)	E2-PT (p/s)	E2-PR (p/s)	E3-PT (p/s)	E3-PR (p/s)	E4-PT (p/s)	E4-PR (p/s)	E5-PT (p/s)	E5-PR (p/s)
3456	0,5	0,166666667	1,388888889	0,388888889	0,75	0,25	0,722222222	0,388888889	1,055555556	0,388888889
3492	0,888888889	0,388888889	0,722222222	0,388888889	0,722222222	0,222222222	1	0,333333333	0,944444444	0,444444444
3528	1,388888889	0,388888889	1,027777778	0,333333333	0,472222222	0,138888889	0,638888889	0,305555556	0,666666667	0,166666667
3564	0,277777778	0,111111111	0,638888889	0,333333333	0,361111111	0,194444444	0,444444444	0,111111111	0,111111111	0,111111111
Total	66,5	27	70,94444444	28,94444444	72,69444444	27,52777778	73,47222222	30,13888889	72,88888889	29,22222222

210 Anexa 1 Valorile obținute pentru experimentele 1-14

TRAFFIC E6-E10

t(s)	E6-PT (p/s)	E6-PR (p/s)	E7-PT (p/s)	E7-PR (p/s)	E8-PT (p/s)	E8-PR (p/s)	E9-PT (p/s)	E9-PR (p/s)	E10-PT (p/s)	E10-PR (p/s)
0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0
72	0,555555556	0,222222222	0,888888889	0,194444444	0,111111111	0,111111111	0,555555556	0,222222222	0,722222222	0,222222222
108	6,111111111	2,777777778	7,777777778	2,972222222	6,055555556	2,722222222	7,055555556	2,722222222	7,055555556	2,861111111
144	0,944444444	0,444444444	0,388888889	0,222222222	0,777777778	0,277777778	1,611111111	0,444444444	0,722222222	0,416666667
180	0,944444444	0,277777778	0,222222222	0,055555556	0,777777778	0,277777778	1	0,333333333	0,555555556	0,222222222
216	0,777777778	0,277777778	0,611111111	0,277777778	1,111111111	0,444444444	0,444444444	0,277777778	0,777777778	0,277777778
252	0,444444444	0,277777778	0,611111111	0,277777778	0,361111111	0,194444444	1,055555556	0,388888889	0,777777778	0,277777778
288	0,5	0,166666667	0,722222222	0,222222222	0,638888889	0,305555556	0,333333333	0,166666667	0,944444444	0,277777778
324	0,277777778	0,111111111	0,944444444	0,277777778	0,861111111	0,361111111	1	0,333333333	1,277777778	0,444444444
360	0,222222222	0,055555556	0,5	0,333333333	0,611111111	0,277777778	0,666666667	0,166666667	0,5	0,166666667
396	0,777777778	0,277777778	0,555555556	0,166666667	0,944444444	0,444444444	0,833333333	0,333333333	0,222222222	0,055555556
432	1,055555556	0,388888889	0,833333333	0,222222222	1,138888889	0,472222222	0,833333333	0,333333333	1	0,333333333
468	0,333333333	0,333333333	0,833333333	0,5	0,527777778	0,194444444	0,833333333	0,333333333	0,777777778	0,444444444
504	0,166666667	0,166666667	0,138888889	0,138888889	0,583333333	0,25	0,055555556	0,055555556	0,666666667	0,333333333
540	0,777777778	0,277777778	1,138888889	0,305555556	0,833333333	0,333333333	0,555555556	0,222222222	0,277777778	0,111111111
576	0,722222222	0,222222222	0,777777778	0,444444444	0	0	0,277777778	0,111111111	0,388888889	0,222222222
612	1,166666667	0,5	0,305555556	0,111111111	0,527777778	0,361111111	0,222222222	0,055555556	0,138888889	0,138888889
648	0,555555556	0,222222222	0,472222222	0,166666667	0,305555556	0,138888889	0,666666667	0,333333333	0,638888889	0,138888889
684	1	0,333333333	0,5	0,166666667	0,611111111	0,277777778	0,222222222	0,055555556	0,722222222	0,388888889
720	0,833333333	0,333333333	0,111111111	0,111111111	0,944444444	0,277777778	0,722222222	0,222222222	1	0,5
756	0,722222222	0,222222222	0,944444444	0,277777778	0,555555556	0,222222222	0,222222222	1	0,333333333	0,055555556
792	1,222222222	0,388888889	0,111111111	0,111111111	0,777777778	0,25	1,111111111	0,444444444	1,166666667	0,5
828	0,555555556	0,222222222	0,555555556	0,222222222	1,777777778	0,638888889	0,277777778	0,111111111	1,388888889	0,388888889
864	0,277777778	0,111111111	0,833333333	0,333333333	0,305555556	0,138888889	1	0,333333333	0,166666667	0,166666667
900	1,444444444	0,611111111	1,444444444	0,444444444	0,416666667	0,25	0,944444444	0,444444444	1,444444444	0,444444444
936	1,222222222	0,555555556	0,444444444	0,277777778	0,194444444	0,194444444	0,111111111	0,111111111	1,055555556	0,388888889
972	0,277777778	0,111111111	0,555555556	0,388888889	1,444444444	0,444444444	0,222222222	0,222222222	0,611111111	0,277777778
1008	0,722222222	0,222222222	0,388888889	0,222222222	0,694444444	0,194444444	0,666666667	0,166666667	0,666666667	0,333333333
1044	0,722222222	0,222222222	1,055555556	0,388888889	0,861111111	0,361111111	0,611111111	0,277777778	0,333333333	0,166666667
1080	0,666666667	0,333333333	0,277777778	0,111111111	1,388888889	0,555555556	0,666666667	0,333333333	1,166666667	0,5
1116	0,555555556	0,388888889	0,611111111	0,277777778	1,027777778	0,361111111	0,277777778	0,111111111	0,888888889	0,388888889

Anexa 1 Valorile obținute pentru experimentele 1-14 211

TRAFFIC E6-E10

t(s)	E6-PT (p/s)	E6-PR (p/s)	E7-PT (p/s)	E7-PR (p/s)	E8-PT (p/s)	E8-PR (p/s)	E9-PT (p/s)	E9-PR (p/s)	E10-PT (p/s)	E10-PR (p/s)
1152	0,777777778	0,277777778	0,111111111	0,111111111	0,694444444	0,361111111	0,333333333	0,166666667	0,555555556	0,222222222
1188	1,111111111	0,444444444	1,166666667	0,5	0,472222222	0,305555556	0,5	0,333333333	0,777777778	0,444444444
1224	0,888888889	0,388888889	0,333333333	0,166666667	0,5	0,166666667	0,75	0,222222222	0,555555556	0,222222222
1260	0,777777778	0,277777778	0,277777778	0,111111111	2,138888889	0,638888889	0,138888889	0,166666667	0,055555556	0,055555556
1296	0,944444444	0,277777778	0,777777778	0,777777778	0,388888889	0,222222222	0,5	0,333333333	0,555555556	0,222222222
1332	0,111111111	0,111111111	0,722222222	0,388888889	0,388888889	0,222222222	1,166666667	0,333333333	0,555555556	0,222222222
1368	0,611111111	0,277777778	0,972222222	0,305555556	0,333333333	0,333333333	0,777777778	0,277777778	0,277777778	0,111111111
1404	0,833333333	0,333333333	0,305555556	0,138888889	1,722222222	0,555555556	1,111111111	0,444444444	0,888888889	0,388888889
1440	0,222222222	0,055555556	0,555555556	0,222222222	1,305555556	0,472222222	0,833333333	0,333333333	0,444444444	0,277777778
1476	0,888888889	0,388888889	0,833333333	0,333333333	0,555555556	0,222222222	0,055555556	0,055555556	0,777777778	0,277777778
1512	0,777777778	0,277777778	1,444444444	0,444444444	0,972222222	0,472222222	0,388888889	0,222222222	0,333333333	0,166666667
1548	0,277777778	0,111111111	0,722222222	0,222222222	0,305555556	0,138888889	0,111111111	0,111111111	0,611111111	0,277777778
1584	1,111111111	0,277777778	0,388888889	0,222222222	0,666666667	0,333333333	1,111111111	0,444444444	0,555555556	0,222222222
1620	0,444444444	0,277777778	0,5	0,166666667	1,111111111	0,444444444	0,777777778	0,277777778	0,222222222	0,222222222
1656	0,833333333	0,5	0,583333333	0,305555556	0,472222222	0,138888889	0,611111111	0,277777778	0,277777778	0,277777778
1692	1,222222222	0,388888889	0,305555556	0,305555556	0,472222222	0,138888889	0,555555556	0,222222222	0,333333333	0,166666667
1728	1,444444444	0,444444444	0,416666667	0,416666667	0,722222222	0,222222222	1,277777778	0,444444444	0,722222222	0,222222222
1764	0,833333333	0,333333333	0,138888889	0,138888889	1,416666667	0,416666667	0,388888889	0,222222222	0,444444444	0,277777778
1800	1,833333333	0,666666667	0,888888889	0,388888889	0,361111111	0,194444444	0,722222222	0,222222222	1,722222222	0,722222222
1836	0,444444444	0,111111111	0,416666667	0,222222222	0,75	0,25	1,055555556	0,388888889	0,333333333	0,166666667
1872	1,111111111	0,444444444	0,361111111	0,222222222	0,722222222	0,222222222	0	0	0	0
1908	0,5	0,333333333	0,222222222	0,055555556	0,527777778	0,194444444	0,333333333	0,166666667	1	0,333333333
1944	0,777777778	0,277777778	0,555555556	0,222222222	0,361111111	0,194444444	0,388888889	0,222222222	1,055555556	0,388888889
1980	0,944444444	0,277777778	0,583333333	0,25	0,916666667	0,25	0,277777778	0,111111111	1,166666667	0,333333333
2016	0,388888889	0,222222222	0,75	0,25	0,638888889	0,305555556	0,5	0,166666667	0,833333333	0,333333333
2052	1,055555556	0,388888889	0,5	0,333333333	0,444444444	0,277777778	0,611111111	0,277777778	0,5	0,166666667
2088	0,5	0,166666667	0,944444444	0,444444444	0,694444444	0,194444444	1,111111111	0,444444444	0,111111111	0,111111111
2124	1,388888889	0,388888889	0,666666667	0,333333333	0,25	0,083333333	1,388888889	0,388888889	0,333333333	0,166666667
2160	0,555555556	0,222222222	0,555555556	0,222222222	1,222222222	0,388888889	1,277777778	0,444444444	0,055555556	0,055555556
2196	0,777777778	0,277777778	1,222222222	0,388888889	0,722222222	0,222222222	0,777777778	0,277777778	0,944444444	0,444444444
2232	0,111111111	0,111111111	0,222222222	0,222222222	0,472222222	0,138888889	0	0	0,944444444	0,444444444
2268	0,444444444	0,277777778	1,111111111	0,444444444	0,25	0,083333333	1,444444444	0,444444444	0,111111111	0,111111111

212 Anexa 1 Valorile obținute pentru experimentele 1-14

TRAFFIC E6-E10

t(s)	E6-PT (p/s)	E6-PR (p/s)	E7-PT (p/s)	E7-PR (p/s)	E8-PT (p/s)	E8-PR (p/s)	E9-PT (p/s)	E9-PR (p/s)	E10-PT (p/s)	E10-PR (p/s)
2304	1,166666667	0,333333333	0,166666667	0,166666667	0,333333333	0,166666667	0,5	0,333333333	0,666666667	0,166666667
2340	0,277777778	0,111111111	0,777777778	0,277777778	0,666666667	0,333333333	0,833333333	0,333333333	0,666666667	0,333333333
2376	0,277777778	0,111111111	0,611111111	0,277777778	0	0	0,277777778	0,277777778	0,833333333	0,333333333
2412	0,555555556	0,222222222	1,055555556	0,388888889	1,25	0,416666667	0,833333333	0,333333333	0,555555556	0,222222222
2448	0,5	0,333333333	0	0	0,611111111	0,277777778	0,444444444	0,277777778	0,361111111	0,194444444
2484	0,333333333	0,166666667	0,388888889	0,222222222	0,666666667	0,166666667	0,277777778	0,111111111	0,861111111	0,361111111
2520	0,444444444	0,277777778	0,833333333	0,333333333	0,888888889	0,222222222	1,055555556	0,388888889	0,555555556	0,222222222
2556	0,944444444	0,277777778	0,666666667	0,333333333	0,611111111	0,277777778	1,111111111	0,444444444	0,555555556	0,222222222
2592	0,722222222	0,222222222	0,777777778	0,277777778	0,555555556	0,222222222	0,222222222	0,055555556	1,166666667	0,277777778
2628	1,222222222	0,388888889	1,222222222	0,388888889	0,277777778	0,111111111	0,222222222	0,555555556	0,777777778	0,277777778
2664	0,333333333	0,166666667	0,888888889	0,388888889	0,583333333	0,25	0,5	0,166666667	0,333333333	0,166666667
2700	0,777777778	0,277777778	0,722222222	0,222222222	0,611111111	0,277777778	0,777777778	0,277777778	0,722222222	0,222222222
2736	0,833333333	0,5	0,555555556	0,222222222	0,611111111	0,277777778	0,722222222	0,222222222	1,138888889	0,305555556
2772	0,333333333	0,166666667	0,833333333	0,333333333	0,694444444	0,194444444	0,388888889	0,222222222	1,416666667	0,416666667
2808	0,444444444	0,277777778	0,888888889	0,388888889	0,277777778	0,027777778	0,666666667	0,166666667	1,222222222	0,388888889
2844	0,666666667	0,333333333	0,777777778	0,277777778	0,388888889	0,194444444	0,722222222	0,222222222	0,333333333	0,166666667
2880	0,444444444	0,277777778	0,777777778	0,277777778	1,194444444	0,388888889	0,305555556	0,111111111	0,333333333	0,166666667
2916	0,722222222	0,222222222	0,555555556	0,222222222	0,361111111	0,194444444	0,583333333	0,277777778	0,111111111	0,111111111
2952	0,777777778	0,444444444	0,5	0,166666667	0,388888889	0,222222222	1,166666667	0,5	0,666666667	0,333333333
2988	0,555555556	0,222222222	1,055555556	0,388888889	1	0,333333333	0,944444444	0,277777778	0,333333333	0,166666667
3024	0,944444444	0,277777778	0,777777778	0,277777778	0,444444444	0,111111111	0,444444444	0,277777778	0,444444444	0,277777778
3060	0,611111111	0,277777778	0,722222222	0,222222222	1	0,305555556	0,833333333	0,333333333	0,722222222	0,388888889
3096	1,5	0,5	0,555555556	0,388888889	0,916666667	0,444444444	0,388888889	0,222222222	0,777777778	0,277777778
3132	0,111111111	0,111111111	0,111111111	0,111111111	0,555555556	0,222222222	0,888888889	0,222222222	0,5	0,166666667
3168	0,444444444	0,111111111	1	0,333333333	0,694444444	0,194444444	0,888888889	0,222222222	1,222222222	0,388888889
3204	1,166666667	0,333333333	1,75	0,583333333	1,027777778	0,361111111	0,555555556	0,222222222	0,722222222	0,222222222
3240	0,277777778	0,111111111	0,416666667	0,25	1,166666667	0,333333333	0,444444444	0,277777778	0,5	0,166666667
3276	0,888888889	0,388888889	1,166666667	0,333333333	0,333333333	0,333333333	0,5	0,166666667	0,666666667	0,166666667
3312	1,611111111	0,444444444	1,111111111	0,444444444	0,75	0,25	0,277777778	0,111111111	1,111111111	0,444444444
3348	0,777777778	0,444444444	0,722222222	0,222222222	0,916666667	0,166666667	0,666666667	0,166666667	1	0,333333333
3384	0,111111111	0,111111111	0,222222222	0,222222222	0,583333333	0,25	0,777777778	0,277777778	0,972222222	0,472222222
3420	1,333333333	0,5	0,222222222	0,055555556	0,166666667	0,166666667	0,916666667	0,388888889	0,916666667	0,25

TRAFFIC E6-E10

t(s)	E6-PT (p/s)	E6-PR (p/s)	E7-PT (p/s)	E7-PR (p/s)	E8-PT (p/s)	E8-PR (p/s)	E9-PT (p/s)	E9-PR (p/s)	E10-PT (p/s)	E10-PR (p/s)
3456	0,611111111	0,277777778	1,666666667	0,5	0,111111111	0,111111111	1,25	0,611111111	0,777777778	0,277777778
3492	0,333333333	0,166666667	0,611111111	0,277777778	0,666666667	0,166666667	0,444444444	0,277777778	0,611111111	0,444444444
3528	0,833333333	0,333333333	0,388888889	0,222222222	0,333333333	0,166666667	0,555555556	0,222222222	0,722222222	0,222222222
3564	0,722222222	0,222222222	0,555555556	0,222222222	0,805555556	0,305555556	1,277777778	0,444444444	0,555555556	0,222222222
Total	76,11111111	30,61111111	71,33333333	29	72,91666667	28,75	70,77777778	28,11111111	73,11111111	29,77777778

214 Anexa 1 Valorile obținute pentru experimentele 1-14

TRAFFIC E11-E14

t(s)	E11-PT (p/s)	E11-PR (p/s)	E12-PT (p/s)	E12-PR (p/s)	E13-PT (p/s)	E13-PR (p/s)	E14-PT (p/s)	E14-PR (p/s)
0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0
72	0,111111111	0,111111111	0,722222222	0,222222222	0,555555556	0,194444444	0,555555556	0,222222222
108	7,611111111	2,944444444	5,944444444	2,777777778	7,277777778	2,638888889	5,833333333	2,833333333
144	1,944444444	0,611111111	0,833333333	0,333333333	1	0,333333333	0,888888889	0,388888889
180	0,25	0,083333333	0,166666667	0,166666667	0,555555556	0,222222222	0,611111111	0,277777778
216	0,75	0,25	0,777777778	0,277777778	1,111111111	0,444444444	0,777777778	0,277777778
252	1	0,333333333	0,277777778	0,111111111	0,611111111	0,277777778	0,555555556	0,222222222
288	1,083333333	0,416666667	0,833333333	0,333333333	0,527777778	0,166666667	1,666666667	0,5
324	0,694444444	0,194444444	1,277777778	0,444444444	0,305555556	0,166666667	0,722222222	0,222222222
360	0,361111111	0,194444444	0,944444444	0,277777778	1	0,333333333	0,944444444	0,277777778
396	0,638888889	0,305555556	0,722222222	0,222222222	0,611111111	0,277777778	0,777777778	0,277777778
432	1	0,333333333	0,277777778	0,111111111	0,277777778	0,083333333	0,555555556	0,388888889
468	0,527777778	0,194444444	0,888888889	0,388888889	0,5	0,194444444	0,666666667	0,333333333
504	0,472222222	0,138888889	0,805555556	0,472222222	1,277777778	0,444444444	0,333333333	0,166666667
540	0,277777778	0,111111111	0,583333333	0,25	0,277777778	0,277777778	0,555555556	0,222222222
576	0,055555556	0,055555556	0,555555556	0,222222222	2,111111111	0,777777778	0,166666667	0,166666667
612	1,111111111	0,277777778	0,083333333	0,083333333	0,916666667	0,25	0,833333333	0,333333333
648	0,277777778	0,111111111	0,583333333	0,25	0,583333333	0,25	0,555555556	0,222222222
684	0,694444444	0,194444444	0,333333333	0,166666667	0,75	0,222222222	0,611111111	0,277777778
720	0,944444444	0,277777778	0,666666667	0,333333333	0,694444444	0,388888889	0,611111111	0,277777778
756	0,861111111	0,194444444	0,888888889	0,388888889	0,555555556	0,222222222	0,777777778	0,277777778
792	0,833333333	0,333333333	1	0,333333333	0	0	0,666666667	0,333333333
828	0,611111111	0,111111111	0,888888889	0,388888889	0	0	0,222222222	0,055555556
864	1,111111111	0,444444444	0,888888889	0,222222222	0,166666667	0,166666667	0,444444444	0,111111111
900	0,666666667	0,166666667	0,111111111	0,111111111	0,111111111	0,111111111	1,944444444	0,611111111
936	1	0,5	1,666666667	0,5	0,722222222	0,222222222	0,722222222	0,222222222
972	0,5	0,166666667	0,5	0,166666667	0,5	0,166666667	0,555555556	0,222222222
1008	0,694444444	0,361111111	0,25	0,25	0,444444444	0,111111111	0,388888889	0,222222222
1044	0,416666667	0,083333333	0,472222222	0,305555556	0,333333333	0,166666667	0,944444444	0,444444444
1080	0,805555556	0,305555556	0,444444444	0,111111111	0,611111111	0,277777778	0,222222222	0,055555556
1116	1,25	0,416666667	0,611111111	0,277777778	0,194444444	0,194444444	0,611111111	0,277777778

TRAFFIC E11-E14

t(s)	E11-PT (p/s)	E11-PR (p/s)	E12-PT (p/s)	E12-PR (p/s)	E13-PT (p/s)	E13-PR (p/s)	E14-PT (p/s)	E14-PR (p/s)
1152	0,5	0,166666667	0,611111111	0,277777778	1,694444444	0,527777778	0,444444444	0,277777778
1188	0,222222222	0,222222222	0,666666667	0,333333333	0,5	0,166666667	1,111111111	0,444444444
1224	0,777777778	0,277777778	1,111111111	0,444444444	0,805555556	0,305555556	0,277777778	0,5
1260	0,638888889	0,305555556	0,444444444	0,277777778	0,638888889	0,472222222	0,611111111	0,111111111
1296	0,638888889	0,305555556	0,333333333	0,166666667	0,5	0,166666667	0,111111111	0,277777778
1332	1,194444444	0,527777778	0,166666667	0,166666667	1,111111111	0,444444444	0,277777778	0,111111111
1368	1,055555556	0,388888889	0,444444444	0,111111111	0,777777778	0,444444444	0,666666667	0,166666667
1404	0,944444444	0,277777778	0,111111111	0,111111111	0,777777778	0,277777778	0,666666667	0,166666667
1440	0,555555556	0,222222222	0,722222222	0,222222222	1,055555556	0,388888889	0,222222222	0,055555556
1476	0,416666667	0,25	0,444444444	0,111111111	1,666666667	0,5	0,944444444	0,444444444
1512	0,75	0,25	0,666666667	0,333333333	0	0	0,5	0,166666667
1548	1,388888889	0,555555556	0,833333333	0,333333333	1	0,333333333	0,111111111	0,111111111
1584	0,611111111	0,277777778	0,333333333	0,166666667	0,611111111	0,277777778	0,555555556	0,222222222
1620	0,75	0,25	0,5	0,166666667	0,333333333	0,166666667	0,555555556	0,222222222
1656	0,166666667	0,166666667	0,444444444	0,277777778	0,277777778	0,277777778	0,666666667	0,333333333
1692	0,416666667	0,25	0,555555556	0,222222222	0,666666667	0,333333333	0,388888889	0,222222222
1728	0,361111111	0,194444444	0,444444444	0,277777778	0,833333333	0,333333333	0,388888889	0,222222222
1764	1,472222222	0,472222222	0,333333333	0,166666667	1	0,333333333	0,277777778	0,277777778
1800	0,777777778	0,25	0,611111111	0,444444444	0,305555556	0,138888889	0,166666667	0,166666667
1836	0,527777778	0,222222222	0,555555556	0,222222222	1,25	0,416666667	1,111111111	0,444444444
1872	0,194444444	0,194444444	0,555555556	0,222222222	0,5	0,166666667	0,555555556	0,222222222
1908	1,25	0,416666667	0,555555556	0,222222222	0,722222222	0,222222222	0,888888889	0,388888889
1944	0,055555556	0,055555556	0,888888889	0,388888889	0,5	0,166666667	0,722222222	0,388888889
1980	0,777777778	0,277777778	0,666666667	0,333333333	0,888888889	0,388888889	0,5	0,166666667
2016	0,444444444	0,277777778	0,555555556	0,222222222	0,555555556	0,222222222	0,5	0,166666667
2052	0,75	0,25	0,5	0,166666667	1,555555556	0,527777778	0,888888889	0,388888889
2088	0,361111111	0,194444444	0,888888889	0,388888889	0,722222222	0,25	0,666666667	0,166666667
2124	0,25	0,083333333	0,722222222	0,222222222	0,388888889	0,222222222	0,777777778	0,277777778
2160	1,194444444	0,361111111	1	0,333333333	0,638888889	0,277777778	0,722222222	0,222222222
2196	0,166666667	0,166666667	0,722222222	0,222222222	0,361111111	0,222222222	0,5	0,166666667
2232	0,166666667	0,166666667	0,388888889	0,222222222	0,333333333	0,166666667	0,166666667	0,166666667
2268	0,833333333	0,333333333	0,5	0,166666667	0,277777778	0,111111111	0,333333333	0,166666667

216 Anexa 1 Valorile obținute pentru experimentele 1-14

TRAFFIC E11-E14

t (s)	E11-PT (p/s)	E11-PR (p/s)	E12-PT (p/s)	E12-PR (p/s)	E13-PT (p/s)	E13-PR (p/s)	E14-PT (p/s)	E14-PR (p/s)
2304	0,47222222	0,13888889	0,11111111	0,11111111	0,83333333	0,33333333	0,44444444	0,27777778
2340	1,16666667	0,33333333	1,27777778	0,61111111	0,05555556	0,05555556	0,72222222	0,38888889
2376	0,52777778	0,36111111	0,5	0,16666667	1,22222222	0,38888889	0,83333333	0,33333333
2412	0,16666667	0,16666667	0,61111111	0,27777778	0,33333333	0,16666667	0,27777778	0,11111111
2448	0,69444444	0,36111111	0,38888889	0,22222222	0,33333333	0,16666667	0,61111111	0,27777778
2484	1,02777778	0,36111111	1,05555556	0,38888889	0,55555556	0,22222222	1,05555556	0,38888889
2520	0,88888889	0,22222222	0,41666667	0,25	0,72222222	0,38888889	0,61111111	0,27777778
2556	0,77777778	0,27777778	0,97222222	0,30555556	0,66666667	0,33333333	1,5	0,5
2592	0,77777778	0,27777778	1,5	0,5	1,33333333	0,5	0,61111111	0,27777778
2628	0,66666667	0,33333333	0,77777778	0,44444444	1,38888889	0,38888889	0,61111111	0,27777778
2664	0,66666667	0,33333333	0,5	0,16666667	0,61111111	0,25	0,38888889	0,22222222
2700	0,58333333	0,25	0,27777778	0,11111111	1,25	0,44444444	0,61111111	0,27777778
2736	0,69444444	0,36111111	1,05555556	0,38888889	1,25	0,41666667	1,05555556	0,38888889
2772	1,08333333	0,41666667	0,94444444	0,44444444	0,58333333	0,38888889	0,66666667	0,33333333
2808	0,30555556	0,13888889	0,72222222	0,22222222	0,58333333	0,27777778	0,72222222	0,22222222
2844	0,69444444	0,19444444	0,88888889	0,88888889	0,94444444	0,44444444	0,66666667	0,16666667
2880	1,05555556	0,38888889	1,22222222	0,38888889	0,5	0,33333333	1,22222222	0,38888889
2916	1,91666667	0,58333333	1,22222222	0,38888889	0,88888889	0,38888889	1,44444444	0,44444444
2952	0,66666667	0,16666667	0,33333333	0,16666667	0,72222222	0,22222222	0,33333333	0,16666667
2988	0,55555556	0,22222222	0,22222222	0,05555556	1	0,33333333	0,05555556	0,05555556
3024	0,33333333	0,16666667	0,33333333	0,16666667	0,38888889	0,22222222	0,61111111	0,27777778
3060	0,33333333	0,16666667	0,36111111	0,19444444	0,61111111	0,27777778	0,55555556	0,22222222
3096	0,33333333	0,16666667	0,41666667	0,25	0,77777778	0,27777778	0,22222222	0,22222222
3132	0,5	0,16666667	0,38888889	0,22222222	0,83333333	0,33333333	0,61111111	0,27777778
3168	0,80555556	0,30555556	0,38888889	0,22222222	0,77777778	0,27777778	0,44444444	0,27777778
3204	0,36111111	0,19444444	0,02777778	0,02777778	0,55555556	0,22222222	0,05555556	0,05555556
3240	0,94444444	0,27777778	0,75	0,41666667	0,94444444	0,27777778	1,05555556	0,38888889
3276	1,13888889	0,30555556	0,22222222	0,05555556	1,05555556	0,38888889	0,5	0,16666667
3312	0,5	0,16666667	1	0,33333333	0,11111111	0,11111111	1,55555556	0,55555556
3348	0,38888889	0,22222222	0,94444444	0,27777778	0,72222222	0,38888889	0,66666667	0,16666667
3384	0,69444444	0,19444444	1	0,33333333	0,88888889	0,22222222	0,5	0,16666667
3420	0,80555556	0,30555556	0,94444444	0,27777778	0,55555556	0,22222222	0,61111111	0,27777778

TRAFFIC E11-E14

t(s)	E11-PT (p/s)	E11-PR (p/s)	E12-PT (p/s)	E12-PR (p/s)	E13-PT (p/s)	E13-PR (p/s)	E14-PT (p/s)	E14-PR (p/s)
3456	1,055555556	0,388888889	1,055555556	0,388888889	0,333333333	0,166666667	1,166666667	0,333333333
3492	0,75	0,25	1,444444444	0,444444444	0,361111111	0,194444444	1,222222222	0,388888889
3528	1	0,333333333	0,5	0,333333333	1,527777778	0,527777778	0,666666667	0,333333333
3564	0,138888889	0,138888889	1,388888889	0,388888889	0,611111111	0,277777778	1,055555556	0,388888889
Total	74,63888889	28,47222222	69,44444444	28,61111111	74,66666667	29,66666667	69,22222222	28,72222222

218 Anexa 1 Valorile obținute pentru experimentele 1-14

Ethernet Delay E1 – E10

t(s)	E1-Delay(s)	E2-Delay(s)	E3-Delay(s)	E4-Delay(s)	E5-Delay(s)	E6-Delay(s)	E7-Delay(s)	E8-Delay(s)	E9-Delay(s)	E10-Delay(s)
0	N/A	5,78933E-05	5,79019E-05	6,19014E-06	6,19014E-06	N/A	5,78433E-05	N/A	N/A	5,80301E-05
72	5,78933E-05	0,02839357	0,029221647	0,000607193	0,0002613	0,002186012	0,003213003	5,78864E-05	N/A	0,023815243
108	0,004737998	0,027349129	0,024025171	0,000699387	0,000276708	0,002217623	0,003213003	0,029906722	0,004653335	0,024916057
144	0,004880334	0,027534647	0,021300688	0,000617487	0,000267765	0,002189268	0,003813323	0,029171714	0,004062262	0,025222593
180	0,004917753	0,025988164	0,021757971	0,000623895	0,000254365	0,002256227	0,003997107	0,020557362	0,00315079	0,024148206
216	0,004497992	0,026015384	0,024088682	0,000744536	0,000284184	0,002209302	0,003712911	0,021028378	0,003099452	0,024848236
252	0,004166607	0,025555265	0,021902992	0,000727288	0,000251128	0,002194932	0,002881719	0,021762895	0,003029931	0,024374228
288	0,004204497	0,025564073	0,023423825	0,000642984	0,000275109	0,00227158	0,004329548	0,020066083	0,003187666	0,024107856
324	0,005664685	0,027145872	0,02369086	0,000739433	0,000214879	0,002252217	0,002914303	0,022680089	0,002961337	0,023680696
360	0,004120187	0,027969342	0,024338523	0,000738891	0,000286788	0,002306315	0,002989101	0,021417946	0,003119565	0,024117235
396	0,003853468	0,026727473	0,023063784	0,000698213	0,000252932	0,002210265	0,001330381	0,022047705	0,003346299	0,025278787
432	0,004010453	0,024922613	0,023897343	0,00070762	0,000262027	0,002242079	0,001830081	0,024444959	0,00358503	0,024948897
468	0,004390376	0,027771779	0,024587958	0,000640223	0,000282871	0,002178491	0,002476223	0,021829688	0,003175762	0,024175288
504	0,004972092	0,025994741	0,023190125	0,000705412	0,000287992	0,002200848	0,001691966	0,020827641	0,003102869	0,023764934
540	0,004487264	0,026180178	0,023142601	0,000701502	0,000232451	0,002237093	0,00304024	0,023272993	0,003328044	0,023536622
576	0,004719906	0,027959755	0,022245656	0,000701499	0,00026474	0,002244858	0,002585159	0,02196511	0,003150942	0,023869504
612	0,004845617	0,026426904	0,024251334	0,000753647	0,000236321	0,002216333	0,003900625	0,022901591	0,003295406	0,021585703
648	0,005639544	0,026765382	0,022904658	0,000614229	0,000281003	0,002219354	0,002722579	0,021285718	0,002759151	0,025475885
684	0,004166845	0,026918224	0,02344695	0,000678075	0,000289868	0,002226265	0,002948999	0,02251364	0,003366504	0,024510017
720	0,00489431	0,027571267	0,023411994	0,000640211	0,000278092	0,00223106	0,003456628	0,021685381	0,003335628	0,024293611
756	0,004120112	0,026601601	0,019885839	0,000748128	0,000271861	0,002176402	0,003807806	0,020868549	0,003197806	0,025278646
792	0,005550416	0,026125692	0,022530775	0,000640945	0,000214994	0,00223744	0,00172438	0,021344992	0,003175899	0,024217281
828	0,004344726	0,027644618	0,021494796	0,00074991	0,000287577	0,00218509	0,002059974	0,021462241	0,003165084	0,024349824
864	0,004737653	0,023794147	0,023591298	0,000676614	0,000294372	0,002185419	0,003300952	0,02135614	0,003231144	0,022622717
900	0,005102122	0,026741696	0,022913616	0,000743535	0,000283252	0,002223254	0,003773136	0,021573183	0,003181492	0,024412439
936	0,005090781	0,026942983	0,023863602	0,000679619	0,000293179	0,002215507	0,003119415	0,021896415	0,003119222	0,024478941
972	0,004472093	0,027506694	0,024812077	0,000735318	0,000277019	0,002185447	0,003170277	0,021483223	0,0030848	0,024632956
1008	0,005676172	0,028265578	0,023555713	0,000689837	0,000251257	0,002280649	0,002274448	0,023472242	0,002928875	0,023369948
1044	0,00578455	0,02605451	0,022643183	0,000748161	0,000266744	0,002243525	0,002832259	0,02193946	0,003213371	0,024561226
1080	0,004496714	0,026238804	0,019518863	0,000735443	0,000244129	0,002213732	0,003967599	0,021614155	0,003222876	0,024413222
1116	0,004730184	0,026765455	0,024648465	0,000624423	0,00023314	0,002197086	0,003275368	0,022304888	0,003237938	0,024319687
1152	0,004734865	0,025574146	0,022033159	0,00074573	0,000257861	0,002262994	0,003456635	0,021271853	0,0031294	0,024740566

Ethernet Delay E1 – E10

t(s)	E1-Delay(s)	E2-Delay(s)	E3-Delay(s)	E4-Delay(s)	E5-Delay(s)	E6-Delay(s)	E7-Delay(s)	E8-Delay(s)	E9-Delay(s)	E10-Delay(s)
1188	0,004557826	0,026790658	0,033260957	0,000702958	0,000132563	0,002202474	0,002460314	0,020940585	0,003221643	0,024471536
1224	0,005329373	0,026377202	0,02090101	0,000713621	0,000270395	0,002247144	0,00283043	0,022921473	0,003097023	0,023546152
1260	0,004648199	0,026383114	0,023094182	0,000743524	0,000287223	0,002236823	0,001898996	0,02247573	0,003052814	0,024202519
1296	0,005150188	0,027549029	0,026437677	0,000695619	0,000238636	0,002283831	0,002712574	0,02153625	0,003154936	0,024148167
1332	0,005059295	0,023115925	0,023125593	0,000743603	0,000251684	0,002201632	0,003096376	0,022707864	0,003116797	0,024148181
1368	0,004834873	0,028561765	0,022199282	0,000663709	0,000222693	0,002243266	0,003914139	0,021604628	0,003148113	0,024740654
1404	0,004310422	0,024892762	0,024121666	0,00070933	0,000286908	0,002230868	0,003709664	0,0237551	0,003077569	0,02466368
1440	0,005241786	0,026577571	0,02355496	0,000713632	0,000233291	0,002169271	0,003576835	0,022498064	0,003107064	0,023469855
1476	0,004297922	0,024433917	0,024029734	0,00072806	0,000134304	0,002227323	0,003882565	0,020613614	0,003221124	0,024848265
1512	0,004971059	0,025872065	0,023239633	0,000743466	0,000270009	0,002262903	0,003379499	0,02131534	0,003073276	0,02456116
1548	0,004279957	0,025179226	0,02342165	0,000570487	0,000276942	0,002254191	0,003630654	0,020658715	0,003115558	0,024633799
1584	0,005013043	0,026787593	0,023431155	0,000742794	0,000266607	0,002279321	0,002688162	0,021404005	0,00316979	0,022953726
1620	0,004148965	0,024677819	0,023103926	0,000611329	0,000279432	0,002222806	0,003572617	0,021256641	0,003557089	0,023017654
1656	0,004311954	0,027198352	0,024090707	0,000743553	0,000237478	0,002208412	0,003326926	0,022833999	0,003235891	0,023254598
1692	0,004619775	0,023841521	0,022454277	0,000711772	0,000238564	0,002236216	0,001472265	0,021388174	0,002955753	0,024561259
1728	0,003726371	0,026176125	0,025098574	0,000751226	0,000255298	0,002282877	0,002356858	0,022529611	0,00326986	0,025009621
1764	0,004425281	0,027254553	0,02364806	0,000728099	0,000274457	0,002252873	0,00163494	0,020781861	0,003278731	0,023943773
1800	0,004440902	0,025908249	0,025100127	0,000695656	0,000206462	0,002238697	0,002493398	0,020524362	0,003186589	0,024989524
1836	0,004262194	0,025895947	0,024770204	0,000743579	0,000214983	0,002304108	0,002669138	0,021461893	0,003263242	0,024561175
1908	0,004151027	0,02384154	0,023578403	0,000738017	0,000286773	0,002236639	0,004354305	0,021563636	0,003087347	0,02412364
1944	0,004356027	0,027561785	0,023514868	0,000744607	0,000239116	0,002196913	0,000154616	0,021155519	0,003175085	0,024478926
1980	0,004364685	0,026050851	0,023001174	0,00071527	0,000257152	0,002264666	0,003887171	0,020083279	0,002898706	0,02469799
2016	0,005305724	0,026765375	0,024180012	0,000751523	0,000266541	0,002285034	0,003985395	0,022016525	0,003024954	0,024345614
2052	0,00452445	0,025549119	0,023900015	0,000617449	0,000216549	0,002228022	0,003604804	0,0218957	0,002891519	0,023314594
2088	0,005166471	0,024467222	0,025927229	0,000673152	0,000251268	0,002202614	0,002160832	0,021375405	0,003192619	0,024202462
2124	0,004917811	0,031912706	0,024350527	0,000659126	0,000221201	0,002224318	0,003284931	0,021860537	0,003165807	0,0245612
2160	0,004940634	0,02581368	0,024703583	0,000647804	0,000248339	0,002291032	0,001947502	0,021146926	0,003260966	0,021832781
2196	0,003853418	0,02855218	0,022628475	0,000504143	0,00026159	0,002220342	0,004227866	0,020813323	0,003170036	0,023411615
2232	0,004209784	0,027025076	0,022113452	0,000743546	0,000141364	0,002181802	0,002445613	0,021694083	0,003117122	0,024008819
2268	0,004845792	0,026945496	0,02284852	0,000653217	0,000214933	0,002199997	0,003711194	0,022470799	0,003028482	0,024202468
2304	0,005284813	0,027071189	0,025438816	0,00063715	0,000238533	0,002222773	0,003218391	0,021811099	0,003094462	0,025278673
2340	0,004756683	0,025969077	0,022455412	0,000738974	0,000237976	0,002287855	0,003403625	0,017786166	0,003127053	0,024159862

220 Anexa 1 Valorile obținute pentru experimentele 1-14

Ethernet Delay E1 – E10

t(s)	E1-Delay(s)	E2-Delay(s)	E3-Delay(s)	E4-Delay(s)	E5-Delay(s)	E6-Delay(s)	E7-Delay(s)	E8-Delay(s)	E9-Delay(s)	E10-Delay(s)
2376	0,004566589	0,027663616	0,024337311	0,000745302	0,000271123	0,002254527	0,002896775	0,022005989	0,002983432	0,023950685
2412	0,005087551	0,02461867	0,022100152	0,000743506	0,00025427	0,002254934	0,003955465	0,021639787	0,003117367	0,024384838
2448	0,005151821	0,02761384	0,023410851	0,000623844	0,000245948	0,002222081	0,002700979	0,022736097	0,003081745	0,02414818
2484	0,004606214	0,026260342	0,022951257	0,000713621	0,000273055	0,002196686	0,002226597	0,021635948	0,003186343	0,02437321
2520	0,004159137	0,026911234	0,025556076	0,000751515	0,000276958	0,002235479	0,003282719	0,021510676	0,003928443	0,023720866
2556	0,0061612	0,025330335	0,024366496	0,000703571	0,000291068	0,002196055	0,003589854	0,022019172	0,003205221	0,024138492
2592	0,003175251	0,026453961	0,022500593	0,000676605	0,000251293	0,00223084	0,002773636	0,021623533	0,003200318	0,024680529
2628	0,00427521	0,026567989	0,019141095	0,000715741	0,000289639	0,002247055	0,002511572	0,021765203	0,00325708	0,024366615
2664	0,00459338	0,02677851	0,025896494	0,000738305	0,000272507	0,002276375	0,001735462	0,022370246	0,003230127	0,023771334
2700	0,00400958	0,025547474	0,024705047	0,000676868	0,000280198	0,002235488	0,003755765	0,021082187	0,003125989	0,024407641
2736	0,004153516	0,024900947	0,022760692	0,000663669	0,000227423	0,002237625	0,003657	0,022282258	0,003165474	0,024890398
2772	0,004611367	0,027654255	0,025978898	0,000743504	0,000287128	0,002194025	0,004331994	0,022573345	0,003066935	0,024617988
2808	0,00497887	0,027931469	0,025233674	0,000743487	0,000287124	0,002190001	0,00317982	0,000896998	0,003013982	0,024632692
2844	0,004672996	0,026163763	0,02281453	0,000707507	0,000258893	0,002167945	0,002846878	0,02032031	0,003260321	0,024561207
2880	0,00488456	0,027348659	0,024316596	0,000742874	0,000286885	0,00221168	0,002841036	0,023560736	0,002872817	0,024561199
2916	0,004182316	0,029154166	0,023329899	0,000708552	0,000285537	0,00221964	0,003094252	0,020305784	0,003109863	0,024202503
2952	0,004609428	0,026488488	0,02258424	0,000659341	0,00022606	0,002280502	0,00367092	0,020951604	0,003090127	0,024561204
2988	0,004120218	0,022064584	0,022706108	0,000663744	0,000293327	0,002194457	0,003509902	0,02251911	0,003351329	0,024561224
3024	0,004254842	0,025988249	0,02343627	0,000743506	0,000274244	0,002253145	0,002712573	0,022232095	0,003171101	0,023943811
3060	0,004498602	0,025618688	0,023591092	0,000718834	0,000255066	0,002257365	0,004045556	0,021129557	0,002911762	0,024171476
3096	0,004198043	0,026062131	0,023371922	0,000737644	0,000228467	0,002216528	0,003489377	0,022429801	0,002898242	0,024848238
3132	0,004480539	0,026803458	0,026488977	0,000693143	0,000296928	0,002370369	0,003297585	0,021896507	0,003080772	0,023327353
3168	0,005533574	0,027980849	0,022494625	0,000754803	0,000287156	0,002199608	0,003519034	0,019602118	0,003577949	0,023600661
3204	0,004906238	0,028431935	0,023554526	0,000743531	0,000213117	0,002237122	0,003434333	0,020650908	0,002984484	0,024417238
3240	0,005103651	0,02439212	0,025438893	0,000713302	0,000262106	0,00224334	0,00385662	0,02176418	0,003050042	0,024919963
3276	0,004516911	0,027345295	0,022368222	0,000599898	0,000212162	0,002254516	0,003046491	0,021273948	0,003072018	0,024476032
3312	0,003801935	0,027649422	0,02265091	0,000680643	0,000290762	0,002205943	0,003903552	0,019946848	0,003078997	0,024439561
3348	0,004725419	0,02835806	0,02304899	0,000699893	0,000238598	0,002275837	0,00339866	0,020304814	0,003360649	0,024919894
3384	0,004511797	0,027042612	0,022854804	0,000733134	0,000283681	0,002177689	0,001956638	0,021175735	0,003094148	0,0236444
3420	0,004509047	0,029440898	0,022184824	0,000690317	0,000239763	0,002202463	0,005065878	0,021499936	0,003083937	0,025180271
3456	0,004647702	0,027936865	0,024901603	0,000583913	0,000287551	0,0022278	0,003372917	0,021300086	0,00314031	0,024848202
3492	0,004446005	0,026276657	0,02306488	0,000606726	0,000267083	0,00224375	0,003337013	0,0232333153	0,003017117	0,024102719

Ethernet Delay E1 – E10

t(s)	E1-Delay(s)	E2-Delay(s)	E3-Delay(s)	E4-Delay(s)	E5-Delay(s)	E6-Delay(s)	E7-Delay(s)	E8-Delay(s)	E9-Delay(s)	E10-Delay(s)
3528	0,004672349	0,026446586	0,024093786	0,000703619	0,000232689	0,002237562	0,003943997	0,021030411	0,003164482	0,024407615
3564	0,004458877	0,026045504	0,025487864	0,000693084	0,000289652	0,002230613	0,003656993	0,028322491	0,003142558	0,024740568
Max.	0,0061612	0,031912706	0,0333260957	0,000754803	0,000296928	0,002370369	0,005065878	0,029906722	0,004653335	0,025475885

222 Anexa 1 Valorile obținute pentru experimentele 1-14

Ethernet Delay E11 – E14

t(s)	E11-Delay(s)	E12-Delay(s)	E13-Delay(s)	E14-Delay(s)	t(s)	E11-Delay(s)	E12-Delay(s)	E13-Delay(s)	E14-Delay(s)
0	6,05331E-06	6,06188E-06	N/A	6,00334E-06	1188	0,000430813	0,00248865	0,000330137	0,002632016
72	0,000369022	0,002297284	N/A	0,002541182	1224	0,000344826	0,002288162	0,000319078	0,002676653
108	0,000383077	0,002419514	6,04636E-06	0,002621071	1260	0,000376671	0,002336497	0,00031523	0,002481866
144	0,000390939	0,002386372	0,000274791	0,002575813	1296	0,000324508	0,002188684	0,00027423	0,002638616
180	0,000476252	0,002317243	0,000386906	0,00249328	1332	0,000388766	0,002275499	0,000311704	0,002546628
216	0,000482904	0,002616527	0,000341759	0,002661299	1368	0,000266821	0,001983489	0,000280956	0,002601346
252	0,000324878	0,002544846	0,00032941	0,002904612	1404	0,000230265	0,002378809	0,000345289	0,002819404
288	0,000336297	0,002242909	0,000289315	0,002535989	1440	0,000370619	0,002356987	0,00026448	0,002901018
324	0,000348302	0,002406383	0,000254873	0,00259817	1476	0,0003329	0,002472245	0,000325471	0,002678547
360	0,000329692	0,002222004	0,000229309	0,002610959	1512	0,000213835	0,002348529	0,000303891	0,002537396
396	0,000482425	0,002326828	0,000252364	0,002660918	1548	0,000410267	0,002467085	0,000295625	0,002667109
432	0,000299253	0,002302677	0,000318224	0,002510661	1584	0,000474794	0,002351114	0,00027567	0,002601413
468	0,000307542	0,002305413	0,000309505	0,002543274	1620	0,000379306	0,002338778	0,000296876	0,002783113
504	0,000316987	0,00230365	0,000315172	0,002744455	1656	0,000213222	0,002314901	0,000294621	0,002543625
540	0,000287433	0,002438234	0,000376283	0,002633475	1692	0,000394176	0,002157297	0,00031493	0,002664867
576	0,000357041	0,002332681	0,00031114	0,002546644	1728	0,000421899	0,002266106	0,000241117	0,002301479
612	0,00033316	0,00236617	0,000356373	0,00258069	1764	0,000371385	0,002185016	0,000417815	0,00271584
648	0,000395957	0,002189672	0,000275352	0,002631989	1800	0,000340485	0,002222833	0,000260835	0,002546605
684	0,000332132	0,002309538	0,000224649	0,002637499	1836	0,000456151	0,002335784	0,000202108	0,002752981
720	0,000401836	0,002447222	0,000320416	0,002736966	1908	0,000373179	0,002331635	0,000346376	0,002813734
756	0,000460576	0,002460822	0,000237084	0,002537778	1944	0,000390617	0,002304012	0,000309493	0,00268075
792	0,000422385	0,002382031	0,000243037	0,002643397	1980	0,000501128	0,002256712	0,000233855	0,002663461
828	0,00042889	0,002289609	0,000241245	0,002288966	2016	0,000360224	0,002371476	0,000360539	0,002497235
864	0,000350573	0,002289024	0,000322219	0,00265615	2052	0,00046124	0,00224614	0,000311714	0,002660427
900	0,00033802	0,002315888	0,000324982	0,002648571	2088	0,000301294	0,002259158	0,000396097	0,002473347
936	0,000374868	0,002496935	0,000254914	0,002477215	2124	0,000381799	0,002357415	0,00023152	0,00241133
972	0,00031204	0,002261075	0,000225416	0,00273452	2160	0,000341033	0,002480601	0,000316046	0,00254123
1008	0,000283841	0,002166747	0,000401829	0,002512304	2196	0,000322011	0,002339702	0,000250497	0,002689537
1044	0,000374108	0,00229159	0,000329055	0,002632653	2232	0,000381091	0,002449868	0,000178015	0,002621612
1080	0,000377344	0,002533527	0,000300444	0,00277856	2268	0,000405053	0,002220065	0,000179285	0,002667184
1116	0,00038708	0,002329229	0,000277748	0,002591599	2304	0,000311375	0,002342774	0,000317227	0,002784614
1152	0,000530137	0,00254827	0,000279643	0,00254328	2340	0,000457206	0,002439136	0,000246417	0,002689425

Ethernet Delay E11 – E14

t(s)	E11-Delay(s)	E12-Delay(s)	E13-Delay(s)	E14-Delay(s)	t(s)	E11-Delay(s)	E12-Delay(s)	E13-Delay(s)	E14-Delay(s)
2376	0,0004326	0,002384254	0,00027893	0,002698688	3528	0,000395213	0,002236677	0,000343735	0,002607098
2412	0,000364696	0,002462823	0,000300077	0,002621772	3564	0,000384785	0,002291356	0,000314566	0,002462025
2448	0,000405088	0,002258698	0,000306232	0,002783098	Max.	0,000538193	0,00277863	0,000401829	0,002904612
2484	0,000391006	0,002314807	0,000252464	0,002832626					
2520	0,000425981	0,002353813	0,000166212	0,002592535					
2556	0,000344066	0,002446892	0,000255689	0,002469187					
2592	0,000447296	0,002246724	0,000347841	0,002728276					
2628	0,000407704	0,002380046	0,000342087	0,002494817					
2664	0,000361636	0,002445293	0,000331501	0,002637894					
2700	0,000250441	0,002225905	0,000314534	0,002514222					
2736	0,00052787	0,002487288	0,000279432	0,002615319					
2772	0,000460683	0,002405515	0,000324111	0,002714155					
2808	0,000449635	0,002337553	0,000213046	0,002663786					
2844	0,000290855	0,002542399	0,000336864	0,00262784					
2880	0,000387133	0,002595532	0,000310006	0,002819405					
2916	0,000375297	0,002352598	0,000295901	0,002607363					
2952	0,000364301	0,002489486	0,000382301	0,002838254					
2988	0,000382858	0,001983758	0,000350981	0,002783973					
3024	0,000432145	0,00277863	0,000233503	0,002546605					
3060	0,000479349	0,002549286	0,000343481	0,002760301					
3096	0,000334665	0,002370261	0,000329984	0,002511026					
3132	0,000431116	0,002443752	0,000255448	0,002576759					
3168	0,000287906	0,002399578	0,000328814	0,002589714					
3204	0,000377061	0,002494702	0,000290951	0,002544433					
3240	0,000538193	0,002426832	0,000362237	0,002185182					
3276	0,000360614	0,002278426	0,000246582	0,002609518					
3312	0,000293636	0,002656073	0,000324062	0,002780952					
3348	0,000310593	0,002339026	0,000231197	0,002844119					
3384	0,000430966	0,002346359	0,000255925	0,002696951					
3420	0,000275114	0,002421005	0,000325838	0,002782251					
3456	0,000371564	0,002489398	0,000372781	0,002662729					
3492	0,000358481	0,002283646	0,000302497	0,002759997					

□

ANEXA 2 – Reprezentarea unităților fizice în TEDS

Field #	Description	# bytes
1	<p>ENUMERATION</p> <p>0: Unit is described by the product of SI base units raised to the powers recorded in fields 2 through 10.</p> <p>1: Unit is U/U, where U is described by the product SI base units raised to the powers recorded in fields 2 through 10.</p> <p>2: Unit is $\log_e(U)$, where U is described by the product of SI base units raised to the powers recorded in fields 2 through 10.</p> <p>3: Unit is $\log_e(U/U)$, where U is described by the product of SI base units raised to the powers recorded in fields 2 through 10.</p> <p>4: The associated quantity is digital data (e.g. a bit vector) and has no unit. Fields 2-10 shall be set to 128.</p> <p>5-255: Reserved</p>	1
2	$(2 * \text{<exponent of radians>} + 128)$	1
3	$(2 * \text{<exponent of steradians>} + 128)$	1
4	$(2 * \text{<exponent of meters>} + 128)$	1
5	$(2 * \text{<exponent of kilograms>} + 128)$	1
6	$(2 * \text{<exponent of seconds>} + 128)$	1
7	$(2 * \text{<exponent of amperes>} + 128)$	1
8	$(2 * \text{<exponent of kelvins>} + 128)$	1
9	$(2 * \text{<exponent of moles>} + 128)$	1
10	$(2 * \text{<exponent of candelas>} + 128)$	1

ANEXA 3 – Exemplu TEDS compatibil IEEE1451 (Implementare Texas Instruments)

Meta TEDS				
Field #	Description	Field Length (Bytes)	Field type	Field Contents
Data structure related information				
1	Meta-TEDS Length	4	U32	48
2	IEEE 1451 Standards Family Working Group Number	1	U8	2
3	TEDS Major Version Number	2	U16	2
4	Future Extensions Key	1	U8	0 (NONE)
5	CHANNEL_ZERO Industry Extensions Key	1	U8	0 (NONE)
6	End Users' Application Specific TEDS Key	1	U8	0
7	Number of Implemented Channels	1	U8	1
8	String Language Code	1	U8	0
9	Bytes per Character	1	U8	1
Timing related information				
10	Worst Case Channel Data Model Length	1	U8	2
11	Worst Case Channel Data Repetitions	2	U16	1
12	Worst Case Channel Update Time	4	F32	2.00E-05
13	Worst Case Channel Write Setup Time	4	F32	0
14	Worst Case Channel Read Setup Time	4	F32	8.00E-05
15	Input/Output Response Time	4	F32	5.00E-04
16	Calibration TEDS Write Time	4	F32	0
17	Worst Case Data Clock Frequency	4	U32	2.00E+05
18	Worst Case Channel Sampling Period	4	F32	2.00E-04
19	Worst Case Unit Warm Up Time	4	F32	1
Channel grouping related information				
20	Channel Groupings Data Sub-Block Length	2	U16	0
21	Number of Channel Groupings = G	0	U8	-
22	Group Name Length	0	U8	-
23	Group Name (<= 255)	0	STRING	-
24	Group Type	0	U8	-
25	Number of Group Members = N	0	U8	-
26	Member Channel Numbers List = M(N) (<= 255)	0	array of U8	-
Data integrity information				
27	Checksum for Meta-TEDS	2	U16	62856
Data structure related information				
28	Meta-Identification TEDS Length	4	U32	310
Identification related information				
29	Manufacturer's Identification Length	1	U8	55
30	Manufacturer's Identification (<= 255)	55	STRING	Texas Instruments Incorporated Control Product Division
31	Model Number Length	1	U8	9
32	Model Number (<= 255)	9	STRING	EX3514.XX
33	Revision Code Length	1	U8	2
34	Revision Code	2	STRING	01
35	Serial Number Length	1	U8	5
36	Serial Number (<= 255)	5	STRING	SN-01
37	Date Code Length	1	U8	25
38	Date Code (<= 255)	25	STRING	November 1, 1995, Shift 1
39	Product Description Length	2	U16	205
40	Product Description (<= 65535)	205	STRING	Description: Ratiometric Pressure Transducer Part Number: EX3514.XX Serial Number: SN-01 Pressure Range: 0 To 3000 PSIA Input Voltage: 5 Vdc Output Voltage: 0 To 5 Vdc Temperature Range: -40 To 85° C
Data integrity information data sub-block				
41	Checksum for Meta-Identification TEDS	2	U16	38702

226 Anexa 3 – Exemplu TEDS compatibil IEEE 1451

Channel TEDS				
Field #	Description	Field Length (Bytes)	Field type	Field Contents
Data structure related information				
42	Channel TEDS Length	4	U32	80
43	Calibration Key	1	U8	1 (FIXED)
44	Industry Extension Key	1	U8	0 (NONE)
Transducer related information				
45	Lower Range Limit	4	F32	0
46	Upper Range Limit	4	F32	20684190
47	Physical Units	10	UNITS	Pa (0,128,128,126,130,124,128,128,128,128)
48	Unit Type Key	1	U8	0 (SENSOR)
49	Unit Warm Up Time	4	F32	1
50	Self Test Key	1	U8	0 (NONE)
51	Uncertainty	4	F32	206842
Data converter related information				
52	Channel Data Model	1	U8	0 (N BYTE)
53	Channel Data Model Length	1	U8	2
54	Channel Model Significant Bits	2	U16	12
55	Channel Data Repetitions	2	U16	1
56	Series Increment	4	F32	0
57	Series Units	10	UNITS	0
58	Channel Update Time	4	F32	2.00E-05
59	Channel Write Setup Time	4	F32	0
60	Channel Read Setup Time	4	F32	8.00E-05
61	Data Clock Frequency	4	U32	2.00E+05
62	Channel Sampling Period	4	F32	2.00E-04
63	Timing Correction	4	F32	0
64	Trigger Accuracy	4	F32	5.00E-06
Data integrity information				
65	Checksum for Channel TEDS	2	U16	59968
Data structure related information				
66	Channel Identification TEDS Length	4	U32	8
Identification related information				
67	Manufacturer's Identification Length	1	U8	0
68	Manufacturer's Identification (<= 255)	0	STRING	-
69	Model Number Length	1	U8	0
70	Model Number (<= 255)	0	STRING	-
71	Revision Code Length	1	U8	0
72	Revision Code (<= 255)	0	STRING	-
73	Serial Number Length	1	U8	0
74	Serial Number (<= 255)	0	STRING	-
75	Channel Description Length	2	U16	0
76	Channel Description (<= 65535)	0	STRING	-
Data Integrity information				
77	Checksum for Channel Identification TEDS	2	U16	65527

Anexa 3 – Exemplu TEDS compatibil IEEE 1451 227

Calibration TEDS				
Field #	Description	Field Length (Bytes)	Field type	Field Contents
Data structure related information				
78	Calibration TEDS Length	4	U32	99
Calibration related information				
79	Last Calibration Date-Time	4	U32	0
80	Calibration Interval	4	U32	0
81	Number of Correction Input Channels = n	1	U8	1
82	Correction Input Channel List	1	U8	1
83	Correction Input Channel-Key List	1	U8	0
84	Channel Degree List = D(k)	1	U8	1
85	Number of Segments List = N _k	1	U8	5
86	Segment Boundary Values Table (Pa)	24	F32	0
	(segment 1 high boundary)		F32	4136838
	(segment 2 high boundary)		F32	8273676
	(segment 3 high boundary)		F32	12410514
	(segment 4 high boundary)		F32	16547352
	(segment 5 high boundary)		F32	20684190
87	Segment Offset Values Table (Pa)	20		
	(segment 1 offset)		F32	5051
	(segment 2 offset)		F32	5051
	(segment 3 offset)		F32	5051
	(segment 4 offset)		F32	5051
	(segment 5 offset)		F32	5051
88	Multinomial Coefficients	40		
	A ₀₀ (Pa)		F32	-126372
	A ₀₁ (Pa/count)		F32	5244
	A ₁₀		F32	-44141
	A ₁₁		F32	5144
	A ₂₀		F32	111220
	A ₂₁		F32	5049
	A ₃₀		F32	331826
	A ₃₁		F32	4959
	A ₄₀		F32	610811
	A ₄₁		F32	4874
Data integrity information				
89	Checksum for Calibration TEDS	2	U16	57092

ANEXA 4 – Semnificația semnalelor OPB-IPIC

Signal Name	Range	I/O	Description
Bus2IP_Addr	0:C_<bus>_AWIDTH-1	I	Address to User Logic
Bus2IP_BE	0:C_<bus>_DWIDTH/8-1	I	Byte enables to User Logic
Bus2IP_Burst	none	I	Burst-mode qualifier to User Logic
Bus2IP_Clk	none	I	IPIC clock. Identical to the <bus> clock
Bus2IP_CE	0:C_NUM_CE-1	I	“chip” enable to User Logic
Bus2IP_CS	0:C_NUM_CS-1	I	“chip” select to User Logic
Bus2IP_Data	0:C_<bus>_DWIDTH-1	I	Data to User Logic
Bus2IP_Freeze	none	I	Tells the User Logic to freeze
Bus2IP_RdCE	0:C_NUM_CE-1	I	Read enables to User Logic
Bus2IP_Reset	none	I	Signal to reset the User Logic
Bus2IP_RNW	none	I	Read/Not Write Signal to User Logic
Bus2IP_WrCE	0:C_NUM_CE-1	I	Write enables to User Logic
IP2Bus_Ack	none	O	Acknowledgement from User Logic
IP2Bus_Data	0:C_<bus>_DWIDTH-1	O	Data from IP
IP2Bus_Error	none	O	Error response
IP2Bus_Intr	0:C_IP_INTR_NUM-1	O	Interrupt event signals from User Logic
IP2Bus_PostedWrInh	none	O	Posted write inhibit from User Logic
IP2Bus_Retry	none	O	Retry response from User Logic
IP2Bus_ToutSup	none	O	Timeout suppress from User Logic
Bus2IP_MstError	none	I	Master Error from IPIF
Bus2IP_MstLastAck	none	I	Master Last Acknowledge from IPIF
Bus2IP_MstAck	none	I	Master Acknowledge from IPIF
Bus2IP_MstRetry	none	I	Master Retry from IPIF
Bus2IP_MstTimeOut	none	I	Master Timeout from IPIF
IP2Bus_Addr	0:C_<bus>_AWIDTH-1	O	<bus> address for the master transaction
IP2Bus_Clk	none	O	Possible future signal to allow for dual-clock-domain (asynchronous) FIFOs
IP2Bus_MstBE	0:C_<bus>_DWIDTH/8-1	O	Byte-enables qualifiers from User Logic
IP2Bus_MstBurst	none	O	Burst qualifier from User Logic
IP2Bus_MstBusLock	none	O	Bus-lock qualifier from User Logic
IP2Bus_MstNum	0:3	O	Burst size indicator from User Logic
IP2Bus_MstReq	none	O	Master request from User Logic
IP2Bus_MstRNW	none	O	Read/Not Write from User Logic
IP2IP_Addr	0:C_<bus>_AWIDTH-1	O	Local device address for the master transaction

BIBLIOGRAFIE

- [1] **Modeling, Analysis and Simulation Results Regarding a Power Factor Correction Rectifier - M. Băbăiță, V. Popescu, A. Popovici, P. Papazian, Scientific Bulletin of Politehnica University of Timisoara, Transactions on Electronics and Communications, Vol. 51(65), No. 1, p136-141, ISSN 1583-3380, Timișoara, 2006**
- [2] **A New Technique For PFC Rectifiers Design – M. Băbăiță, V. Popescu, A. Popovici, P. Papazian, RSEE 2006, ISSN 1454-9239, pg.21-24, Oradea**
- [3] Standard field bus networks for industrial applications – Gianluca Cena, Luca Durante, Adriano Valenzano, Computer Standards & Interfaces 17, 1995, Elsevier
- [4] International Electrotechnical Commission – Electrical equipment of industrial machines-Serial data link for real-time communication between controls and drives, Draft International Standard, Revision 7, 1993
- [5] ISO Manufacturing message specification – Service definition, DIS 9506-1 TC 184/SC 5/WG 2, 1988
- [6] ISO Manufacturing message specification – Protocol Specification, DIS 9506-2 TC 184/SC 5/WG 2, 1988
- [7] IEEE Token bus access method, Std. No. 802.4, 1985
- [8] International Electrotechnical Commission, IEC 61158, Digital data communications for measurement and control - Fieldbus for use in industrial control systems, 2003
- [9] International Electrotechnical Commission, IEC 61784-1, Digital data communications for measurement and control - Part 1: Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems, 2003
- [10] International Electrotechnical Commission, IEC 61784-2, Digital data communications for measurement and control - Part 2: Additional profiles for ISO/IEC 8802-3 based communication networks in real-time applications, 65C/350/CD circulated for comments 2004-10-22 until 2005-01-28
- [11] Real-Time Ethernet – Industry Prospective – M. Felser, Proceedings of the IEEE, vol. 93, no.6, 2005
- [12] The CIP family of fieldbus protocols and its newest member Ethernet/IP, Emerging Technologies and Factory Automation – V. Schiffer, Proceedings 8th IEEE International Conference, 15-18 Oct. 2001, vol.1

- [13] ISO/IEC 8802-3:2001, Information technology – Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and Physical Layer specifications
- [14] IEEE 1588, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, 2002
- [15] IEC: Real time Ethernet: P-NET on IP, Proposal for a Publicly Available Specification for Real-Time Ethernet, document IEC, 65C/360/NP, Date of circulation: 2004-12-17
- [16] IEC: Real time Ethernet Vnet/IP, Proposal for a Publicly Available Specification for Real-Time Ethernet, document IEC, 65C/352/NP, Date of circulation: 2004-11-19
- [17] Schneider Automation, Modbus messaging on TCP/IP implementation guide, May 2002, <http://www.modbus.org/>
- [18] IEC: Real time ethernet modbus-RTPS, Proposal for a Publicly Available Specification for Real-Time Ethernet, document IEC 65C/341/NP, Date of circulation: 2004-06-04
- [19] Rețele de calculatoare (ediția a IV-a) – Andrew S. Tanenbaum, Editura Byblos, 2004
- [20] Rețele de calculatoare – Peter Norton, Dave Kearns, Editura Teora, 1999
- [21] On the Ethernet use for real-time publish subscribe based applications – O.Dolejs, P.Smolik, Z. Hanzalek, 2004 IEEE International Workshop on Factory Communication Systems, September 22 – 24, 2004, Vienna, Austria
- [22] PROFINET – Scalable Factory Communication for all Applications – J.Feld, 2004 IEEE International Workshop on Factory Communication Systems, September 22 – 24, 2004, Vienna, Austria
- [23] IEC: Real Time Ethernet: EtherNet/IP with time synchronization, Proposal for a Publicly Available Specification for Real-Time Ethernet, document IEC, 65C/361/NP, Date of circulation: 2004-12-17
- [24] IEC: Real time ethernet TCnet (Time-critical Control Network), Proposal for a Publicly Available Specification for Real-Time Ethernet, document IEC, 65C/353/NP, Date of circulation: 2004-11-19
- [25] IEC: Real Real Time Ethernet EPL (ETHERNET Powerlink), Proposal for a Publicly Available Specification for Real-Time Ethernet, document IEC, 65C/356a/NP, Date of circulation: 2004-12-03
- [26] CiA DS 301, CANopen Application Layer and Communication Profile, Version 4.02, February 2002

- [27] IEC: Real time Ethernet EPA (Ethernet for Plant Automation), Proposal for a Publicly Available Specification for Real-Time Ethernet, document IEC 65C/357/NP, Date of circulation: 2004-11-26
- [28] Enhancements to the Time Synchronization Standard IEEE-1588 for a System of Cascaded Bridges – J.Jasperneite, K.Shehab, K.Weber, 2004 IEEE International Workshop on Factory Communication Systems, September 22 – 24, 2004, Vienna, Austria
- [29] Real-time ethernet the EtherCAT solution – D. Jansen,H. Buttner, Computing & Control Engineering Journal, Volume: 15 , Issue: 1, Feb.-March 2004
- [30] IEC 61491, Electrical equipment of industrial machines – Serial data link for real time communication between controls and drives SERCOS, 2002-10
- [31] 22G/132/Q; 65C/346/Q; Questionnaire, IEC 61491 Maintenance and reorganisation, 2004-07-23
- [32] IEC: Real time ethernet SERCOS III, Proposal for a Publicly Available Specification for Real-Time Ethernet, document IEC, 65C/358/NP, Date of circulation: 2004-12-03
- [33] Proiectarea statistică a experimentelor. Fundamente și studii de caz – Viorel Gh. Vodă, Alexandru Isaic-Maniu, Ed. Economică, 2006
- [34] Study of Network Design Factors That Influence Industrial Fieldbus Network-Based System Integration – Eun Oh,The Ohio State University, 2009
- [35] Fieldbus Network Simulation Using a Time Extended Estelle Formalism – Luc Hohwiller, Serge Wendling, MASCOT '00 Proceedings, IEEE Computer Society, Washington, 2000
- [36] Large-Scale Network Simulations With Gtnets – George F. Riley, Proceeding of the 2003 Winter Simulation Conference, 2003
- [37] OpNet Modeler and NS2: Comparing the Accuracy of Network Simulators for Packet-Level Analysis using a Network Testbed – Gilberto Flores Lucio, Marcos Paredes-Farrera, Emmanuel Jammeh, Martin Fleury, Martin J. Reed, Electronic Systems Engineering Department University of Essex, 2003
- [38] Factor Analysis – Garson, G. David, *Statnotes: Topics in Multivariate Analysis*, <http://faculty.chass.ncsu.edu/garson/PA765/statnote.htm>, 2009
- [39] Design of Experiments: Principles and Applications – Eriksson L., Johansson E., Kettaneh-Wold N., Wikström C. & Wold S., Learnways AB, Umea, 2000
- [40] Design of Experiments: The D-Optimal Approach and Its Implementation As a Computer Algorithm – Fabian Triefenbach, Umetrics 2006
- [41] Experiments: Planning, Analysis, and Parameter Design Optimization – Wu & Hamada, Wiley-Interscience Publication, New York, 2000

- [42] An introduction to statistical analysis – D. G. Rossiter, ITC, 2006,
<http://www.itc.nl/personal/rossiter>
- [43] Multivariate statistical analysis: A conceptual introduction (2nd ed.) – S.K. Kachigan, New York, NY, US: Radius Press, 1991
- [44] Soluții actuale de prelucrare a informației în sistemele de conducere a proceselor tehnologice – P. Papazian, Referat din cadrul pregătirii tezei de doctorat, UPT, 2006**
- [45] Goodness-Of-Fit Techniques – R. B. D'Agostino, M. A. Stephens, Marcel-Dekker, New York, 1986
- [46] Multiple response variables and multispecies interactions. Design and Analysis of Ecological Experiments 2nd edition – S.M. Scheiner, (Scheiner & Gurevitch) Oxford University Press, Oxford, 2001
- [47] Statistică și Econometrie – Eugenia Harja, Alma Mater, Bacău, ISBN 978 - 606 - 527 - 031 - 2, 2009
- [48] Multivariate Statistics Course – Gregory Carey, University of Colorado, Boulder, 1998
- [49] Introducere în statistica multivariată – V. Clocotici, Universitatea „Alexandru Ioan Cuza”, Iași, 2007
- [50] A Unifying Standard for Interfacing Transducers to Networks – IEEE-1451.0 – K. Lee, J. Wiczer, ISA Expo, 2005
- [51] Using the PIC16F84 Microcontroller in Intelligent Stepper Motor Control – P. Papazian, M. Băbăiță, A. Popovici, Journal of Electrical and Electronics Engineering, 01/01/2008, Volume 1, Issue 1, pg.223-226, ISSN 18446035**
- [52] HART Communication Application Guide – HCF, HCF_LIT-039 Rev.1.0 Preliminary, 2010
- [53] IEEE P1451.2 transducer to microprocessor interface – S. Woods, Proceedings Sensors Expo Anaheim, Helmers Publishing, 1996
- [54] Using PIC18F448 In Wireless DC Motor Control – P. Papazian, Lucrările sesiunii de comunicări științifice Doctor ETC 2009, Timișoara 24-25 Septembrie, Ed. Politehnica, pg.77-79, ISSN:2066-883X**
- [55] Wireless Power Supply Using PIC18F448 – P. Papazian, M. Băbăiță, A. Gontean, International Symposium on Electronics and Telecommunications ISETC 2010, Ninth Edition, Timișoara, 11-12 Noembrie, IEEE, pg.111-114, ISBN: 978-1-4244-8458-4, ISBN: 978-1-4244-8460-7**

- [56] IEEE P1451.2 Standard for Smart Transducer Interface, Object Model Specification, Draft 1.61 – J.C. Eidson, IEEE Standards Association, 2005
- [57] IEEE P1451.2 D2.01 IEEE Draft Standard for A Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats, Institute of Electrical and Electronics Engineers, 1996.
- [58] IEEE-P1451 Network Capable Application Processor Information Model – J. Warrior, Proceedings Sensors Expo Anaheim, pp. 15-21, April, 1996, Helmers Publishing.
- [59] A Standardized Approach for Transducer Interfacing: Implementing IEEE-P1451 Smart Transducer Interface Standards, K. Lee, R. Schneeman, Proceedings of Sensor Conference, 1996
- [60] Status of P1451.5 802.11 Sub-Specification – R. Coleman, rcoleman@3eti.com, 2004
- [61] IEEE 1451.0, IEEE Standard for a Smart Transducer Interface for Sensors and Actuators– Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats – Instrumentation and Measurement Society, TC-9, Institute of Electrical and Electronic Engineers, New York, N.Y. 10016-5997, 2005
- [62] Hardware implementation of a PIC18F448 based TIM for IEEE1451.2 compliant actuator control – P. Papazian, M. Băbăiță, International Symposium on Electronics and Telecommunications ISETC 2010, Ninth Edition, Timișoara, 11-12 Noembrie, IEEE, pg.119-122, ISBN: 978-1-4244-8458-4, ISBN: 978-1-4244-8460-7**
- [63] HART Communication Application Guide – HCF, HCF_LIT-034, HART Communication Foundation, 1999, USA
- [64] HART Communication – Technical Information, SAMSON AG., Frankfurt, 2000
- [65] HART Over Ethernet Networks, Pervasive Solutions – J. Tandy, Arcom Control Systems, 2002
- [66] ISO/IEC/IEEE 21451-1:2010, Network Capable Application Processor (NCAP) information model – IEEE 2000, Switzerland, 2010
- [67] System-On-A-Chip Solution for Plug and Play Networked Smart Transducers – G. E. Lopez, University of Pittsburgh, 2004
- [68] The Unified Modeling Language Guide – G. Booch, J. Rumbaugh, I. Jacobson, Addison Wesley, First Edition, 1998
- [69] [Automata theory with modern applications](#) – J. A. Anderson, T. J. Head, Cambridge University Press, 2006

[70] Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML – R. B'Far, Cambridge University Press, USA, 2005

[71] An overview of FPGAs and FPGA programming; Initial Experiences at Daresbury – R. Wain, I. Bush ș.a., CCLRC Daresbury Laboratory, Daresbury, UK, 2006

[72] Features, design tools and application domains of FPGAs – J. Rodriguez-Andina, M.J. Moure, M.D. Valdes, IEEE Transactions on Industrial Electronics, vol. 54, no. 4, 2007

[73] Comparing CPU and FPGA Application Performance – V. Kindratenko, D. Pointer, D. Raila, C. Steffen, 2006

[74] Hardware/Software Co-Design – G. De Micheli and M. Sami, Amsterdam: Kluwer, 1996

[75] ML40x EDK Processor Reference Guide – Xilinx, UG082, 2006

[76] On-Chip Peripheral Bus Architecture Specifications v.2.1 – IBM Corporation, USA, 2001

[77] User Core Templates Reference Guide – Xilinx Inc., 2003

[78] DS473 OPB Interrupt Controller – Xilinx Inc., 2003

[79] DS469 OPB Arbiter – Xilinx Inc., 2005

[80] Industrial Smart Power Supply – M. Băbăiță, P. Papazian, 35th International Conference on Telecommunications and Signal Processing, TSP 2012 Proceedings, ISBN 978-1-4673-1116-8, pg.307-310, Praga, 2012

[81] AC326 GPIO Expansion Using UART – Actel Corp., USA, 2009

[82] Signal Generating Techniques using PIC16F84 Microcontrollers – P. Papazian, M. Băbăiță, A. Popovici, A. Avram, I. Lie, ICMCS-05, Proceedings of the 4'th International Conference on Microelectronics and Computer Science 15-17 September 2005, Technical University of Moldova & Academy of Sciences of Moldova, Vol.II, ISBN 9975-66-040-1, pg.401-404, Chișinău