

**CONTRIBUȚII LA MODELAREA,
SIMULAREA ȘI EMULAREA
CIRCUITELOR DEDICATE
PENTRU CONTROLUL INTELIGENT
AL POMPELOR DE COMBUSTIBIL
DE ÎNALTĂ PRESIUNE
DIN DOMENIUL AUTOMOTIVE**

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

ing. Laszlo Molnar

Conducător științific:
Referenți științifici:

prof.univ.dr.ing Gontean Aurel
prof.univ.dr.ing Dan Lascu
prof.univ.dr.ing Dan Pitică
prof.univ.dr.ing Paul Mugur Svasta

Ziua susținerii tezei: 27.09.2019

Seriile Teze de doctorat ale UPT sunt:

- | | |
|---|--|
| 1. Automatică | 9. Inginerie Mecanică |
| 2. Chimie | 10. Știința Calculatoarelor |
| 3. Energetică | 11. Știința și Ingineria Materialelor |
| 4. Ingineria Chimică | 12. Ingineria sistemelor |
| 5. Inginerie Civilă | 13. Inginerie energetică |
| 6. Inginerie Electrică | 14. Calculatoare și tehnologia informației |
| 7. Inginerie Electronică și Telecomunicații | 15. Ingineria materialelor |
| 8. Inginerie Industrială | 16. Inginerie și Management |

Universitatea Politehnica 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, 2019

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 Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300159 Timișoara, Bd. Republicii 9,
Tel./fax 0256 403823
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 Universității Politehnica Timișoara.

Mulțumiri deosebite se cuvin conducătorului de doctorat, prof.dr.ing. Aurel Gontean pentru îndrumare, discuții prelungite, colaborare excelentă și corectură atentă. Îi mulțumesc domnului șl. dr. ing. Alexandru Amaricăi pentru sfaturile competente și încurajările pe parcursul simulărilor efectuate. Mulțumiri lui Bogdan Komlosi care a lucrat sub directa mea îndrumare și a proiectat circuitul imprimat pe care l-a și populat, a dezvoltat interfața de utilizator în C și o parte de funcționalități VHDL.

Mulțumiri colegilor de la Continental Powertrain Engineering, Luca Octavian și Alexandru Pătru pentru suportul acordat în dezvoltarea modelelor de simulare Saber.

Nu în ultimul rând îi mulțumesc soției mele Raluca pentru înțelegere, răbdare și sprijin în momente cheie.

Timișoara, Septembrie 2019

Laszlo Molnar

Molnar, Laszlo

**CONTRIBUȚII LA MODELAREA, SIMULAREA ȘI EMULAREA
CIRCUITELOR DEDICATE PENTRU CONTROLUL INTELIGENT AL
POMPELOR DE COMBUSTIBIL DE ÎNALTĂ PRESIUNE DIN DOMENIUL
AUTOMOTIVE**

Teze de doctorat ale UPT, Seria 7, Nr. 84, Editura Politehnica, 2019, 144 pagini,
87 figuri, 2 tabele.

ISSN: 1842-7014

ISBN: 978-606-35-0311-5

Cuvinte cheie: ASIC, VHDL, LSD, HSD, HPFP, simulare, validare,

Rezumat,

Obiectivul tezei este acela de a perfecționa metodele de testare și emulare a ASIC-urilor (Application Specific Integrated Circuit) dedicate controlerelor de motor termic și de a găsi posibile îmbunătățiri ale procesului de testare prin simularea codului VHDL (Very High Speed Hardware Description Language) împreună cu cea a blocului analogic necesare îndeplinirii funcției ASIC-ului. Am introdus suplimentar în simulare modele electrice ale modulului (componente discrete care nu pot fi integrate în ASIC) din care face parte ASIC-ul și un model electric al valvei electro-mecanice a pompei de combustibil de înalta presiune (HPFP – High Pressure Fuel Pump) comandată de către ASIC. Cu ajutorul simulării și a modelelor dezvoltate am putut propune și implementa în ASIC-ul emulat cu FPGA (Field Programmable Gate Array) și circuite analogice discrete o strategie de protecție și diagnostic alături de un algoritm de detecție a mișcării electro-valvei.

Cuprins

Lista abrevierilor	9
Lista de tabele	10
Lista de figuri	10
1. Introducere.....	13
1.1 Conținutul tezei	13
1.2 Detalii generale despre procesul de dezvoltare a circuitelor integrate dedicate	14
2. Testarea codului VHDL.....	16
2.1 Metode de simulare a erorilor “Fault simulation”	17
2.1.1 Simularea erorilor comportamentale “Behavioral Fault”	17
2.1.2 Clase de defecte “Fault Clases”	18
2.1.3 Comparația dintre simularea la nivel comportamental și nivel de poartă logică.....	19
2.1.4 Simularea la nivel comportamental cu modele avansate.....	22
2.2 Metode de simulare funcțională.....	23
2.2.1 Simularea circuitelor integrate mixte	24
2.2.2 Validarea limbajului de descriere Hardware folosind metrice Software ...	26
2.3 Metodologii avansate de verificare.....	28
2.3.1 VMM - Verification Methodology Manual.....	28
2.3.2 UVM - Metodologia universală de verificare	28
2.3.3 OVM - Metodologia liberă de verificare	29
2.4 Concluzii și contribuții	32
2.4.1 Rezumat.....	32
2.4.2 Concluzii și contribuții subcapitolele 2.2 și 2.3	33
3. Arhitecturi de validare și simulare a circuitelor integrate dedicate	36
3.1 Arhitecturi de validare hardware a circuitelor integrate dedicate ..	36
3.1.1 Platforma modulară de test propusă.....	37
3.1.2 Placa de validare modulară.....	40
3.1.3 Platforma de validare în stil ECU	41
3.2 Arhitecturi de simulare: existente și cea propusă	43
3.2.1 Arhitectura de co-simulare	43
3.2.2 Arhitectura cu interfață directă Virtualizer® - SaberRD®	45

3.2.3	Arhitectura propusă; SaberRD®	46
3.3	Concluzii și contribuții	47
3.3.1	Concluzii	47
3.3.2	Contribuții	47
4.	Controlul pompelor de injecție	48
4.1	Detalii generale despre sistemul cu injecție directă a benzinei.....	48
4.2	Bazele sistemului GDI.....	49
4.2.1	Componentele mecanice și hidraulice	49
4.2.2	Circuitul electric de control al valvei DIV	51
4.2.3	Principiul de control a pompei de injecție HPFP	53
4.3	Concluzii și contribuții	54
4.3.1	Concluzii	54
4.3.2	Contribuții	54
5.	Rezultate experimentale	55
5.1	Simularea sistemului de control al valvei DIV	55
5.1.1	Blocul de filtraj și alimentare	56
5.1.2	Emularea blocului uC	56
5.1.3	Circuitul de putere	57
5.1.4	Modelul ierarhic, comportamental al ASIC-ului	59
5.1.5	Propunere de model de sarcină DIV	67
5.1.5.1	Simularea sarcinii reale DIV	67
5.1.5.2	Modelul simplu de simulare	68
5.1.5.3	Măsurări realizate pentru a determina variația parametrilor electrici ai supapei DIV.....	68
5.1.5.4	Modele intermediare propuse pentru simulare	73
5.1.5.5	Extragerea inductanței pentru modelul de simulare	74
5.1.5.6	Performanța modelelor intermediare.....	74
5.1.5.7	Modelul final, ierarhic de bobină DIV, propus pentru simulare.....	75
5.2	Implementarea circuitului fizic pentru validare	78
5.2.1	Blocul de filtraj și alimentare	78
5.2.2	Blocul cu microcontroler.....	79
5.2.3	Circuitul de putere	79
5.2.4	Emularea ASIC-ului	80
5.2.5	Implementarea circuitului pe PCB	84
5.3	Compararea rezultatelor simulării cu performanțele sistemului fizic..	86
5.3.1	Rezultatele circuitului experimental	86
5.3.2	Analiză de scenarii complexe prin simulare.....	87
5.3.3	Rezultatele simulării	88

5.4	Concluzii și contribuții	91
5.4.1	Concluzii.....	92
5.4.2	Contribuții.....	93
6.	<i>Detecția mișcării, reducerea vibrației și diagnoza electrică a valvei DIV.....</i>	94
6.1	Reducerea zgomotului valvei.....	94
6.1.1	Statusul existent referitor la reducerea zgomotului DIV	94
6.1.2	Modalitatea originală propusă de reducere a zgomotului DIV.....	96
6.1.3	Rezultatele obținute	98
6.1.4	Probleme întâmpinate și rezolvate în timpul dezvoltării procedurii de reducere a zgomotului.....	102
6.1.4.1	Rezolvarea problemei mișcării precoce	102
6.1.4.2	Rezolvarea problemei variației cu temperatura și tensiunea de alimentare a timpului natural de închidere al valvei	103
6.2	Algoritmul de detecție a mișcării supapei	104
6.3	Diagnoza electrică; protecția circuitului de comandă a solenoidului DIV	110
6.3.1	Metoda actuala de diagnoză	110
6.3.2	Metoda de diagnoză propusă.....	114
6.3.3	Rezultate de simulare; metoda de diagnoză propusă	118
6.4	Concluzii și contribuții	121
6.4.1	Concluzii.....	121
6.4.2	Contribuții.....	122
7.	<i>Concluzii și contribuții personale</i>	124
7.1	Concluzii finale.....	124
7.2	Contribuții personale	126
7.3	Lista lucrărilor	129
7.4	Lista citărilor	130
7.5	Listă patente.....	130
Anexe	131
A1.	– Schema electrică a sistemului emulat; pagina 1.....	131
A2.	– Schema electrică a sistemului emulat; pagina 2.....	132
A3.	– Placa cu circuitul fizic.....	133
A4.	– Placa cu circuitul fizic și placa cu FPGA	134

A5. – Mediul de testare și simulare	135
A6. – Pompa de înaltă presiune; Continental 250bar	136
A7. – Rapoarte de simulare	137
A8. – Comparație rezultate tranzistor ideal – tranzistor real	138
<i>Bibliografie</i>	139

Lista abrevierilor

ADC	Analog to Digital Converter
ASIC	Application Specific Integrated Circuit
ATE	Automatic Test Equipmet
ATPG	Automatic Test Pattern Generator
BDC	Bottom Dead Center
BFM	Behavioral Fault Mapper
BIST	Built-in Self-Test
CAN	Control Area Network
CDV	Coverage Driven Verification
CFG	Control Flow Graph
CLCC	Closed Loop Current Control
CSV	Comma Separated Values
DAC	Digital to Analog Converter
DIV	Digital Inlet Valve
DUT	Device Under Test
ECU	Engine Control Unit
EMI	Electro-Magnetic Interference
ESD	Electro-Static Discharge
FPGA	Field Programmable Gate Array
FWD	Free Wheel Diode
GDI	Gasoline Direct Injection
GUI	Graphical User Interface
HDL	Hardware Description Language
HPFP	High Pressure Fuel Pump
HSD	High Side Driver
HW	Hardware
IC	Integrated Circuit
IP	Intellectual Property
LPFP	Low Pressure Fuel Pump
LSD	Low Side Driver
LUT	Look Up Table
MDB	Microcontroller Development Board
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MPI	Multi Point Injection
MTP	Modular Test Platform
MUX	Multiplexor
MVB	Modular Validation Board
N-ch	N channel
NOK	Not OK
OC	Over Current
OL	Open Load
OPA	Operational Amplifier
OVM	Open Verification Methodology
PCB	Printed Circuit Board
P-ch	P channel
PLA	Programmable Logic Array

PWM	Pulse Width Modulation
RTL	Register Transfer Level
SoC	System-On-Chip
SPTI	Single Point Injection
SPI	Serial Peripheral Interface
SSF	Simulation Support Files
SSL	Single Stuck Line
TBG	Test Bench Generator
TC	Test Chip
TDC	Top Dead Center
TGA	Test Generation Algorithm
TVS	Transient Voltage Suppressor
uC	Microcontroller
UUT	Unit Under Test
UVM	Universal Verification Methodology
VHDL	Very High Speed Hardware Description Language
VHDL-AMS	VHDL Analog Mixed Signal
VMM	Verification Methodology Manual (for Verilog)

Lista de tabele

Tabel 2a. Contribuția autorilor din domeniul "Simulare funcțională și de erori"	34
Tabel 2b. Contribuția autorilor din domeniul HDL "Simulare comportamentală"	35

Lista de figuri

Fig. 1.1. Schema bloc a procesului de dezvoltare a unui CI dedicat (original)	14
Fig. 2.1 Schema bloc a algoritmului de generare automată a secvențelor de testare [22] (original)	17
Fig. 2.2 Punct de ramificare [22] (original)	21
Fig. 2.3 Adiacențe logice ale clauzelor [22] (original)	23
Fig. 2.4 Prezentare generală a simulării funcționale și simulării de erorilor [23] (original)	24
Fig. 2.5 Procesul de verificare [23] (original)	25
Fig. 2.6 Structura de bază a procesului de validare [23] (original)	27
Fig. 2.7 Îmbunătățiri legate de simularea defectelor comportamentale[22] (original).....	32
Fig. 2.8 Cronologie; Evoluția simulării funcționale și a defecțiunilor [23] (original).....	33
Fig. 3.1 Platforma modulară de test propusă [34] (original)	37
Fig. 3.2 Placa de validare modulara [34] (original)	40
Fig. 3.3 Platforma de validare în stil ECU [34] (original)	41
Fig. 3.4 Schema bloc a procesului de validare [34] (original)	42
Fig. 3.5 Arhitectura de co-simulare [34] (original).....	43
Fig. 3.6 Simularea la nivel de sistem prin interfață directă [34] (original).....	45
Fig. 3.7 Arhitectura de simulare propusă [34] (original).....	46
Fig. 4.1 Arhitectura sistemului GDI din lucrarea mea [38], bazată pe [36] (original)	49
Fig. 4.2 Detaliu HPFP bazat pe [36] (original)	50
Fig. 4.3 Pompa de presiune înaltă Continental-350Bar [37].....	51

Fig. 4.4 Schema bloc a circuitului electric de comandă [38] (original).	52
Fig. 4.5 Exemplu de control al HPFP [38] (original).	53
Fig. 5.1 Schema ierarhică de simulare (original).	55
Fig. 5.2 Schema de simulare: blocul de filtraj și alimentare (original).	56
Fig. 5.3 Schema de simulare: blocul de emulare a uC (original).	57
Fig. 5.4 Schema de simulare: circuitul de putere (original).	58
Fig. 5.5 Simbolul de simulare: Modelul comportamental, ierarhic al ASIC-ului (original).	60
Fig. 5.6 Schema de simulare: ierarhia ASIC-ului (original).	61
Fig. 5.7 Ierarhia blocului digital al ASIC-ului (original).	62
Fig. 5.8 Ierarhia blocului analogic al ASIC-ului (original).	63
Fig. 5.9 Blocul: măsurare de curent LS (original).	64
Fig. 5.10 Blocul: control de curent în buclă închisă (original).	64
Fig. 5.11 Blocul: măsurare de curent HS (original).	65
Fig. 5.12 Blocul: comparator de supracurent (original).	66
Fig. 5.13 Variația inductivității bobinei DIV în funcție de curent [39] (original).	68
Fig. 5.14 Model simplu de sarcină pentru simulare [39] (original).	68
Fig. 5.15 Curentul prin sarcina dummy comparație cu cea reală [39] (original).	69
Fig. 5.16 Curentul DIV funcție de tensiunea de alimentare; 700RPM; 200bar [39] (original).	71
Fig. 5.17 Curentul DIV funcție de presiune; 700RPM; 14V [39] (original).	72
Fig. 5.18 Curentul DIV funcție de cuplu motor; 700RPM; 200bar; 14V [39] (original).	73
Fig. 5.19 Modele electrice a bobinei DIV dezvoltate pentru simulare [39] (original).	74
Fig. 5.20 Inductivitatea valvei funcție de curent; la închiderea valvei [39] (original).	74
Fig. 5.21 Performanța modelelor intermediare [39] (original).	75
Fig. 5.22 Blocul: model de bobina a supapei DIV (original).	76
Fig. 5.23 Curent prin sarcina DIV (original).	77
Fig. 5.24 Implementarea blocului de filtraj și alimentare (original).	78
Fig. 5.25 Implementarea blocului cu microcontroler (original).	79
Fig. 5.26 Implementarea circuitului de putere (original).	80
Fig. 5.27 Implementarea circuitului de măsurare de curent LS (original).	81
Fig. 5.28 Implementarea circuitului: Control de Curent în Buclă Închisă (original).	82
Fig. 5.29 Implementarea circuitului de măsurare de curent HS; identic cu cel de LS (original). ..	83
Fig. 5.30 Implementarea circuitului imprimat (original).	85
Fig. 5.31 Oscilogramă circuit experimental (original).	86
Fig. 5.32 Rezultate simulare (original).	87
Fig. 5.33 Rezultate de simulare în comparație cu cele reale (original).	88
Fig. 5.34 Profilul de curent real comparație cu cel simulat – detaliu figura 5.33 (original).	89
Fig. 5.35 Semnalizarea detecției mișcării: real comparație cu cel simulat – detaliu figura 5.33 (original).	90
Fig. 5.36 Ieșirea DAC-ului: real comparație cu simulare – detaliu figura 5.33 (original).	90
Fig. 5.37 Comanda tranzistorului HS: real comparație cu simulare – detaliu figura 5.33 (original).	91
Fig. 6.1 Optimizarea pulsului de închidere pentru reducere a zgomotului [38] (original).	95
Fig. 6.2 Optimizarea pulsului de deschidere pentru reducere a zgomotului [38] (original).	96
Fig. 6.3 Particularitățile controlului valvei DIV [38] (original).	97
Fig. 6.4 Variația inductivității bobinei DIV [38] (original).	98

12 Lista de figuri

Fig. 6.5 Închiderea valvei DIV, alimentare 14V, 25C, fără presiune de combustibil [38] (original).	99
Fig. 6.6 Închiderea și deschiderea valvei DIV, alimentare 14V, 25C, fără presiune de combustibil [38] (original).	100
Fig. 6.7 Zgomotul creat de valva DIV; cu reducere de zgomot; fără reducere de zgomot [38] (original).	101
Fig. 6.8 Închiderea valvei DIV, alimentare 14V, 25C, fără presiune în rampa comună [38] (original).	102
Fig. 6.9 Timpul de închidere/deschidere natural, fără presiune de rampă [38] (original).	103
Fig. 6.10 Controlul timpului natural de închidere, fără presiune de combustibil [38] (original).	104
Fig. 6.11 Algoritm de detecție a mișcării supapei: diagrama de stări (original).	105
Fig. 6.12 Scenariu de simulare: algoritm de detecție a mișcării (original).	107
Fig. 6.13 Detaliu al zonei "1" din Fig. 6.12 (original).....	107
Fig. 6.14 Detaliu al zonei "2" din Fig. 6.12 (original).....	108
Fig. 6.15 Detaliu al zonei "3" din Fig. 6.12 (original).....	109
Fig. 6.16 Arhitectură tipică de ASIC (original).	110
Fig. 6.17 Arhitectura cu două DAC-uri: funcționare normală (original).....	111
Fig. 6.18 Arhitectura cu două DAC-uri: Supracurent la ieșirea LS (original).	112
Fig. 6.19 Arhitectura cu două DAC-uri: solenoidul DIV neconectat (original).	113
Fig. 6.20 Arhitectura "diagnoză bazată pe constanta de timp a circuitului" (original).	114
Fig. 6.21 Arhitectura "diagnoză bazată pe constanta de timp a circuitului" ; funcționare normală (original).	115
Fig. 6.22 Arhitectura "diagnoză bazată pe constanta de timp a circuitului"; supra-curent (OC) (original).	116
Fig. 6.23 Arhitectura "diagnoză bazată pe constanta de timp a circuitului"; sarcină neconectată (OL) (original).	116
Fig. 6.24 Arhitectura "diagnoză bazată pe constanta de timp a circuitului" ; protecție la supra- curent (OC) pentru sarcini rezistive folosind CCC_DAC pentru generare de prag de OC (original).	117
Fig. 6.25 Rezultate de simulare; metoda de diagnoză propusă; funcționare normală (original).	118
Fig. 6.26 Rezultate de simulare; detaliu Fig. 6.25 (original).....	118
Fig. 6.27 Rezultate de simulare; sarcină neconectată(OL) (original).	119
Fig. 6.28 Rezultate de simulare; detaliu Fig. 6.27 (original).....	120
Fig. 6.29 Rezultate de simulare; supra-curent(OC) la tranzistorul LS (original).....	120

1. Introducere

1.1 Conținutul tezei

Obiectivul tezei este acela de a perfecționa metodele de testare și emulare a ASIC-urilor (Application Specific Integrated Circuit) dedicate controlerelor de motor termic și de a găsi posibile îmbunătățiri ale procesului de testare prin simularea codului VHDL (Very High Speed Hardware Description Language) împreună cu cea a blocului analogic necesare îndeplinirii funcției ASIC-ului. Am introdus suplimentar în simulare modelele electrice ale modulului (componente discrete care nu pot fi integrate în ASIC) din care face parte ASIC-ul și un model electric al valvei electro-mecanice a pompei de combustibil de înalta presiune comandată de către ASIC. Cu ajutorul simulării și a modelelor dezvoltate am putut propune și implementa în ASIC-ul emulat cu FPGA (Field Programmable Gate Array) și circuite analogice discrete o strategie de protecție și diagnostic alături de un algoritm de detecție a mișcării electro-valvei.

Teza este organizată în opt capitole, fiecare dintre ele descriind etapele relevante ale activității de cercetare și dezvoltare a doctoratului desfășurat pe o durată de 3 ani.

Capitolul 1 este o introducere în procesul de proiectare de circuite integrate ASIC, adaptat acestei lucrări.

Capitolul 2 conține detalii referitoare la metodele de testare a codului VHDL care subliniază diferența dintre simularea defecțiunilor/erorilor și simularea funcțională.

Capitolul 3 constă într-o revizuire a arhitecturilor de validare hardware pentru ASIC-uri și descrie dezvoltarea unei arhitecturi nou propuse, iar pe de altă parte descrie o platformă de validare virtuală propusă de mine pentru a simula codul VHDL împreună cu modelele blocului analog de ASIC și un model electric original îmbunătățit al valvei HPFP care este controlat de ASIC.

Capitolul 4 abordează subiectul simulărilor și al validărilor, adică funcționarea pompei de benzină de înaltă presiune și strategia de acționare a supapei de scurgere care controlează cantitatea de combustibil transferat în rampa comună către injectoare.

Capitolul 5 conține rezultatele experimentale și implementarea simulării/validării circuitului fizic. Descrie un caz folosit pentru simularea platformei virtuale, atunci când sunt utilizate modele complexe de simulare pentru MOSFET-urile (Metal Oxide Semiconductor Field Effect Transistor) externe comandate de ASIC pentru a controla valve HPFP. Domeniul de aplicare al simulării este de a determina dacă tranzistoarele MOSFET alese sunt potrivite pentru a controla sarcina specifică inductivă în orice caz utilizat, mai ales când tranzistoarele MOSFET funcționează în saturație. Putem astfel determina temperatura maximă a joncțiunii tranzistoarelor MOSFET în timpul activării și să evaluăm puterea medie disipată a dispozitivului semiconductor.

Capitolul 6 cuprinde descrierea unui algoritm propus pentru detecția mișcării supapei în cazul în care se implementează funcționalitatea de reducere a zgomotului produs de elementele mobile ale supapei de scurgere.

Capitolul 7 cuprinde concluziile finale, revendicările și lista lucrărilor, respectiv a patentelor publicate.

Capitolul 8 conține bibliografia consultată pe durata de desfășurare a activității de cercetare.

1.2 Detalii generale despre procesul de dezvoltare a circuitelor integrate dedicate

Dezvoltarea unui ASIC este un proces complex și consumator de timp. În timpul fazei de dezvoltare dar și după fabricare, echipa de ingineri trebuie să verifice implementările digitale și analogice ale tuturor funcțiilor specifice ale ASIC-ului.

Procesul de dezvoltare este prezentat în figura 1.1 și începe cu colectarea specificațiilor pentru funcțiile dorite ale ASIC-ului. ASIC-ul va funcționa într-un sistem, astfel încât arhitectul sistemului și echipa de dezvoltare trebuie să colaboreze în această fază pentru a înțelege mai bine funcțiile necesare și interacțiunea cu restul sistemului.

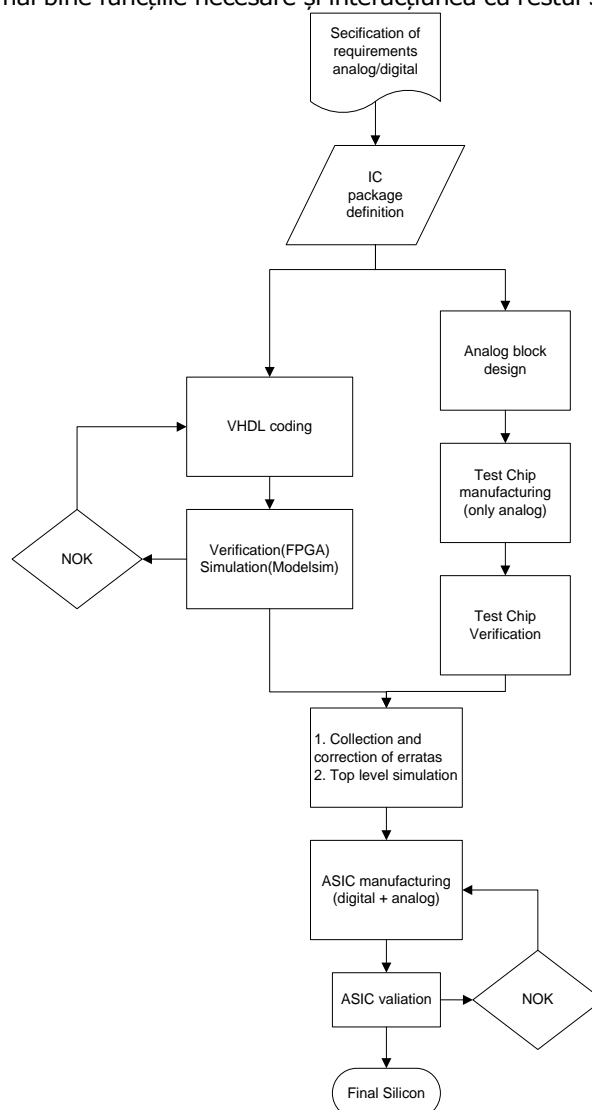


Fig. 1.1. Schema bloc a procesului de dezvoltare a unui CI dedicat (original).

După finalizarea acestui pas, începe implementarea fizică a funcțiilor. Există două procese în paralel, unul, elaborarea codului VHDL pentru a implementa funcțiile digitale definite, iar celălalt, pentru a implementa circuite analogice de mică și mare putere care cooperează cu funcțiile digitale. Circuitele de putere mică sunt comparatoare, surse de referință pentru curent și tensiune, senzori de temperatură, ADC (Analog to Digital Converter), DAC (Digital to Analog Converter) și așa mai departe. Circuitele de mare putere sunt etaje de comandă a porții tranzistoarelor MOSFET, tranzistoare de putere MOSFET N-ch (N channel) și P-ch (P channel) integrate pe pastila de siliciu.

Proiectanții VHDL dar și cei ai blocurilor analogice din ASCII realizează simulări pentru testarea implementării lor.

Suplimentar față de simulare, verificarea codului VHDL se realizează pe plăci FPGA pentru a verifica funcția digitală. În cazul în care sunt găsite erori, codul VHDL este corectat și verificat din nou până când se realizează o implementare fără probleme.

În următorul pas se fabrică un "cip de test" care conține numai blocul analogic, și electronica de putere, care se folosește pentru verificarea implementării analogice. Blocul digital este emulat de echipamente externe pentru a implementa interacțiunea digitală necesară pentru blocul analogic.

Toate constatările sunt colectate ca o listă de erată și corectate în implementarea noii versiuni a CI.

La sfârșit, se realizează o simulare de nivel superior în care blocul digital și cel analog funcționează împreună, dar de data asta și cu adnotări din layout-ul siliciului pentru a evidenția eventualele efecte parazite ale implementării specifice.

Următorul pas este așa numitul „tape-out” atunci când proiectul este înghețat și procesul de fabricație a ASIC-ului (Application Specific Integrated Circuit) începe, dar în această fază doar în număr limitat, până la 100 de bucăți pentru a fi trimise înapoi echipei de proiectare pentru a le verifica.

Pasul următor este verificarea ASIC-ului împreună cu modulul său; toate funcțiile sunt testate acum în interacțiunea cu componentele discrete externe (componente electronice care nu se pot integra în capsula unui circuit integrat) necesare pentru a implementa corect funcțiile ASIC-ului.

Dacă se găsesc comportamente nedorite ale oricărei funcții, acestea sunt investigate și corectate de echipa de proiectare. În acest caz se produce un nou lot de siliciu și verificarea este efectuată din nou cel puțin pentru funcțiile care au fost găsite defecte, dar și pentru cele care ar putea fi afectate de modificările implementate în procesul de corectare a erorilor găsite.

Atunci când nu mai există nici o defecțiune, versiunea finală intră în producția de serie și va fi livrată clienților.

În domeniul circuitelor integrate dedicate pentru automobile, procesul menționat mai sus este necesar datorită standardelor de siguranță, în cazul ASIC-urilor care implementează, de exemplu, controlul injectoarelor, bujiilor și a pompelor de alimentare cu combustibil de înaltă presiune. Toate funcțiile implementate trebuie să funcționeze corespunzător. În cazul în care acest lucru nu se întâmplă, trebuie ca ASIC-ul să poată intra într-o stare de siguranță automat, oprindu-și toate ieșirile de comandă a sarcinilor mai sus enumerate

2. Testarea codului VHDL

În capitolul 1 am prezentat un proces simplificat de proiectare a ASIC-urilor, un pas important fiind testarea codului VHDL. În acest capitol voi extinde acest proces, insistând asupra abordării proprii și contribuțiilor personale

Codul VHDL este testat în faza de proiectare prin simulare și ulterior pe plăci cu FPGA. După finalizarea fazei de proiectare, codul VHDL va fi transferat în pastila de siliciu a ASIC-ului și va deveni o funcție digitală implementată de blocuri logice predefinite cum ar fi: porți logice, MUX (Multiplexor), LUT (Look Up Table), PLA (Programmable Logic Array).

În timpul procesului de fabricare se poate întâmpla ca implementarea VHDL fără bug-uri să devină o funcție defectuoasă datorită unei erori în procesul de fabricație – de exemplu, atunci când se creează un strat de metalizare pentru interconectarea pe pastila de siliciu, un scurtcircuit la o linie de alimentare va afecta funcționarea unei celule (*stack-at failure*).

Există două tipuri principale de metode de detectare a defecțiunilor: injectare de vectori de test prin echipamente de testare automată, ATE (Automatic Test Equipment) și cel de auto-testare, BIST (Built-In Self Test) implementat în ASIC.

Injectarea vectorilor de test necesită o așa-numită simulare a defecțiunilor pentru a defini un set de vectori de testare care sunt necesari pentru programul de testare care controlează ATE-ul. Simularea defecțiunilor se bazează pe introducerea artificială în codul VHDL (corect) a unor defecțiuni conform modelelor de defecțiuni predefinite. Fiecare descriere voit eronată este apoi simulată și comparată cu rezultatele simulării codului fără defecțiuni. Pe baza acestei comparații se poate analiza ce efect are „injectarea” unei defecțiuni în codul fără erori.

BIST este implementat în structura ASIC-ului și este pornit de fiecare dată când acesta este alimentat. Acest test este consumator de timp și de obicei, testează doar o parte limitată a logicii ASIC-ului (blocuri de siguranță). BIST este utilizat pentru a detecta defecțiunile apărute în timpul duratei de viață a componentelor care au fost considerate componente "conforme" după ce toate verificările au fost efectuate în timpul procesului de fabricare. În timpul duratei de viață a componentei s-ar putea întâmpla evenimentele externe, cum ar fi descărcările ESD (Electro-Static Discharge) sau un scurtcircuit între pinii acestuia astfel încât să afecteze o funcție internă a ASIC-ului; executarea procedurii de BIST la fiecare inițializare a ASIC-ului testează aceste funcții, dezactivând ASIC-ul în cazul în care o funcție testată este afectată.

2.1 Metode de simulare a erorilor "Fault simulation"

Prezentul capitol este un studiu al mai multor lucrări în metodele de simulare a erorilor HDL (Hardware Description Language) dezvoltate începând cu sfârșitul anilor 80 până în prezent. Datorită complexității crescânde a ASIC-urilor, este necesar să se dezvolte întotdeauna o metodă de testare mai bună și mai rapidă pentru a garanta un design sigur și un timp scurt de introducere pe piață.

În ultimii 25 de ani, ideea de bază nu sa schimbat, doar câteva modele noi de defecte comportamentale, instrumente comerciale și putere de calcul au fost introduse ca îmbunătățiri. Avantajele și utilitatea diferitelor modele comportamentale au fost prezentate prin compararea a șase articole de cercetări [1], [4], [5] - [8].

2.1.1 Simularea erorilor comportamentale "Behavioral Fault"

Ward și Armstrong au propus pentru prima dată un algoritm de generare a testelor (TBG – Test Bench Generator) și algoritmului de mapare a defecțiunilor comportamentale (BFM – Behavioural Fault Mapper) care susțin funcționarea procedurii automate de generare a secvențelor (vectorilor) de test (ATPG - Automatic Test Pattern Generator) și a altor medii standard de simulare a defecțiunilor folosind modele de defecțiuni comportamentale [1].

Potrivit acestora [1], mediul de simulare a erorilor este cel prezentat în Fig. 2.1. după cum urmează:

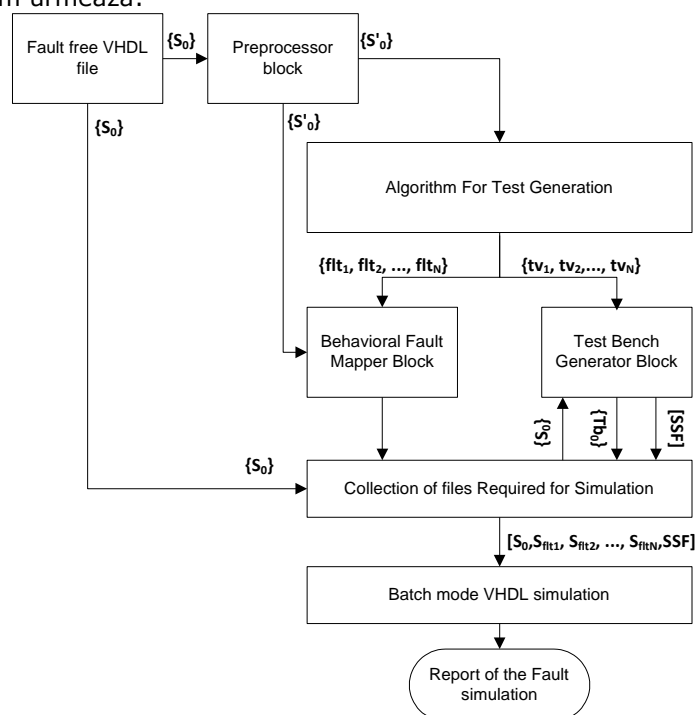


Fig. 2.1 Schema bloc a algoritmului de generare automată a secvențelor de testare [22] (original).

- Fișierul sursă VHDL $\{S_0\}$ este orice descriere VHDL fără defecte.
- Preprocesorul selectează și numerotează liniile fișierului sursă care va fi ținta defectelor comportamentale. Fișierul de ieșire este $\{S'_0\}$ [1].
- Algoritmul de generare a testelor (TGA – Test Generation Algorithm) este considerat a fi unitatea sub test (UUT – Unit Under Test) în acest sistem, deoarece eficacitatea ATPG-ului este cea care urmează a fi determinată. Rezultatul așteptat este un fișier care conține lista de erori și un fișier care conține vectorii de testare [1].
 - BFM folosește fișierul sursă $\{F'_0\}$ preprocesat și lista de erori $\{f_{it1}, f_{it2}, \dots, f_{itN}\}$ de la TGA și produce N fișiere sursă cu defecte artificial introduse $\{S_{fit1}, S_{fit2}, \dots, S_{fitN}\}$ [1].
 - Maparea prin BFM trebuie efectuată astfel încât codul sursă original este modificat cât mai puțin posibil [1].
 - TBG acceptă ca intrare o copie a fișierului sursă $\{S_0\}$ și vectorii de testare $\{tv_1, tv_2, \dots, tv_N\}$, din care produce fișierul pentru testare $\{Tb_0\}$ și o colecție de fișiere de suport pentru simulare [SSF] (Simulation Support Files) [1].
 - Blocul de simulare VHDL conține toate procedurile necesare pentru executarea codului [1].

2.1.2 Clase de defecte "Fault Classes"

Lista defectelor utilizate de BFM a fost definită de Ward și Armstrong [1] pe baza cercetărilor anterioare [2] și [3]. Aceste modele de defecțiuni a porților logice la nivel comportament sunt împărțite în opt clase:

1. Stuck-Then
2. Stuck-Else
3. Assignment Control
4. Dead Process
5. Dead Clause
6. Micro-operation
7. Local Stuck-data
8. Global Stuck-data

Defecțiunea **Stuck-Then** este disfuncția instrucțiunii **IF**. Eroarea acționează executând numai porțiunea "then" indiferent de condiția logică. Maparea acestei erori se efectuează prin înlocuirea condiției logice cu valoarea TRUE (tip de dată boolean), după cum urmează:

```
IF (TRUE) then X <= '1';
else X <= '0';
```

Modificarea de mai sus va forța să se execute doar ramura "then" a instrucțiunii **IF**.

Defecțiunea **Stuck-Else** este similară cu **Stuck-Then**, dar maparea se face înlocuind condiția logică cu valoarea booleană FALSE, astfel că întotdeauna se execută ramura "else".

Assignment Control reprezintă o eroare a operatorului de atribuire în VHDL de a atribui o nouă valoare unui semnal. Maparea defecțiunii este:

```
X <= X; // în cazul în care X nu este un semnal de ieșire
- X <= expresie; // în cazul în care X este un semnal de ieșire
```

Dead Process este eșecul executării declarațiilor dintr-o sintaxă **PROCESS**. Maparea defecțiunii se face înlocuind lista de sensibilitate a procesului cu un semnal care nu își schimbă niciodată valoarea, astfel încât procesul nu va fi declanșat. Maparea defecțiunii este:

```
process(constant_value_signal)
begin
    statement_x;
    statement_y;
end process;
```

Dead Clause este eșecul sintaxei **CASE** de execuție a unei dintre clauze. Exemplu:

```
case INPUT is
    when "10" => x <= "1010";
    when others => null;
end case;
```

BFM va implementa **Dead Process** prin introducerea unei erori de control a atribuirii (*Assignment Control fault*) în sintaxa dorită:

```
when "10" => x <= x;
```

Micro-operation este eșecul unui operator de a-și îndeplini funcția dorită. Operatorul poate să se transforme în orice alt operator din clasa sa. Exemplu:

```
X <= A and not B; // bun
X <= A or not B; // defecțiune: AND a rezultat în operatorul OR
```

Local Stuck-data este eșecul unui semnal sau al unei variabile de a lua valoarea corectă. **Local Stuck-data** este defect cu influență doar în expresia în care este mapată [1].

```
Defecțiune: stuck at 0;
IF (IN = 1) then... // bun
IF (IN = 0) then... // cu defecțiune
```

Global Stuck-data reprezintă eșecul unui semnal sau al unei variabile de a-și schimba valoarea în cadrul arhitecturii VHDL. Anomaliile legate de aceste date nu se limitează doar la anularea unei singure linii a codului VHDL (ca în cazul *Local Stuck-data*), ci va bloca schimbarea stării variabilei sau a semnalului într-o zonă definită a codului sau chiar în întreaga arhitectură VHDL.

2.1.3 Comparația dintre simularea la nivel comportamental și nivel de poartă logică

Chen și Perumal [4] au realizat o analiză a simulării defecțiunilor la nivel comportamental și la nivel de poartă logică. Ward și Armstrong [1] abordează doar modelarea de bază a defectelor comportamentale, dar nu și modelarea defectelor la nivel de poartă logică.

Noutatea abordării constă în folosirea unui set extins de modele de defecțiuni clasificate în șase categorii diferite bazate pe modelele de defecte de la [1]. Aceștia au introdus defecțiunea **Input Stuck At** ca nou model de defecțiune care reprezintă defectarea unei variabile de intrare de a-și schimba valoarea.

1. Input-Stuck-At-fault (**nou**)
2. IF-Stuck-Then-fault
3. IF-Stuck-Else-fault
4. Assignment-Statement-fault (**diferit**)
5. Dead-Clause-fault (**diferit**)
6. Local-Stuck-Data-fault (**diferit**)

Eroarea **Assignment Statement** este mapată diferit de către BFM, față de [1]. Abordarea din [4] este maparea defecțiunii prin înlocuirea expresiei din partea dreaptă a operatorului de atribuire cu "1" sau "0" și nu cu expresia din stânga ca în [1].

Defecțiunea **Dead Clause** invocă și ea Assignment Statement ca în [1], doar că în modul definit mai sus și nu definit ca în [1].

Eroarea **Local Stuck Data** reprezintă o defecțiune în ramurile „fan-out” ale unui circuit. Maparea defecțiunilor se face prin apelarea defecțiunii „stuck-at” și înlocuirea semnalului de ieșire a ramificației cu "0" sau "1" în cadrul instrucțiunilor care sunt afectate de ramificația defectă.

Pentru a lega modelele de defecțiuni comportamentale cu cele de nivel de poartă logică, Chen și Perumel [4] au făcut o analiză a unei descrieri comportamentale VHDL și a implementării la nivel de poartă, pentru care s-au explicat câteva terminologii și definiții utilizate în simularea la nivel de poartă logică:

Definiția 1: "Echivalența defecțiunilor: pentru două defecțiuni existente α și β , α este o eroare echivalentă cu β dacă $F_\alpha = F_\beta$, adică

$$T_\alpha = (F \oplus F_\alpha) = (F \oplus F_\beta) = T_\beta \quad (1)$$

Unde:

- F: funcție adevărată;
- F_α : funcția defectuoasă atunci când există o eroare α ;
- F_β : funcția defectuoasă atunci când există o eroare β ;
- T_α : secvența de testare detectând defectul α ;
- T_β : secvența de testare detectând defectul β "[4].

Definiția 2: Dominanța defectelor: pentru două defecte existente α și β , defectul β domină defectul α dacă T_β este un super-set al lui T_α . Un test pentru β este, de asemenea, un test pentru α , prin urmare, pentru o defecțiune „single stuck-at”, trebuie luată în considerare numai defecțiunea β "[4].

Lemă: "Echivalența și dominanța defecțiunii pentru punctul de ramificație: Considerând că o intrare A se ramifică în două linii B și C așa cum se arată în Fig. 2.2, o defecțiune asupra liniei "A" (stuck-at-1) poate fi detectată prin propagarea efectului prin;

1. B (numai)
2. C (numai)
3. B și C (împreună)

În cazul 1. testele detectează A stuck-at-1, indicat de A=1 și B=1 dar nu și C=1,

În cazul 2. testele detectează A=1 și C=1 dar nu și B=1.

În general, pentru un circuit în care converge fan-out-ul, nu se aplică relația de echivalență și dominanță între defecțiunile de pe ramurile și defecțiunile de pe ramificația fan-out-ului. În cazul în care ramurile nu se converg, atunci defectele de pe intrarea fan-out-ului domină defectele de pe ramuri [4].

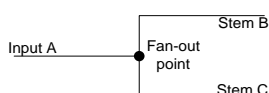


Fig. 2.2 Punct de ramificare [22] (original).

Fiecare conexiune de la implementare la nivel de poartă logică este numerotată și se efectuează o analiză a echivalenței / dominantei pentru a descoperi toate defecțiunile posibile. Pentru simularea la nivel de poartă logică se folosește doar modelul „stuck-at”.

Chen și Perumal au simulat la nivel de porți logice dar și la nivel comportamental circuitul propus în [4]. Simularea la nivel comportamental a detectat 92% din erorile la nivel de poartă logică, cu un avantaj de timp de simulare semnificativ mai redus față de cel necesar pentru simularea la nivel de poartă logică.

Noh, Chen și Chung [5] au îmbunătățit modelele și fluxul de testare utilizat în [4]. Aceștia prezintă un sistem de simulare a erorilor ATPG/comportamental (folosind scripturile C-shell Unix) bazat pe rezultatele anterioare ale lui Ward și Armstrong [1], simularea la nivel comportamental și a erorilor la nivel de poartă logică sunt comparate și analizate. Simulatorul de defecțiuni la nivel de poartă logică utilizează aceleași modele de testare ca și simulatorul de defecțiuni la nivelul comportamental. Produse software comerciale au fost utilizate pentru a susține sistemul de simulare.

Modelele de defecțiuni comportamentale sunt puțin diferite față de cele din [1] și [4]. În [5] se utilizează modele de defecte ale căror efecte au fost studiate și au fost îmbunătățite pentru a trata alte tipuri de date (bit / std_logic; bit_vector / std_logic_vector; boolean):

1. Input-Stuck-At
2. Output-Stuck-At (**nou**)
3. IF-Stuck-then
4. IF-Stuck-else
5. ELSIF-Stuck-then (**nou**)
6. ELSIF-stuck-else (**nou**)
7. Assignment-Statement
8. Micro-operation (**nou față de [4], la fel ca [1]**)
9. Local Stuck-data
10. Global Stuck-data

Sistemul de simulare a defecțiunilor fiind bazat pe cel prezentat în [1], nou este programul de control care controlează întregul sistem în timpul simulării defecțiunilor. Programul de control coordonează următorul flux:

- selectarea defectului țintă din lista de erori
- generarea bancului software de încercare (*Test Bench*) pe baza defectului țintă selectat
- crearea fișierelor de stimuli VHDL
- actualizarea / crearea mediului de testare bazat pe fișierul „fără defect” și fișierele „defecte” generate de BFM

- execuția simulării
- întregul proces se repetă până când se simulează toate defectele.

Chen și-a continuat cercetarea, în [6] și [7]. El prezintă mai detaliat același sistem ATPG / Simulator la nivel comportamental ca și în [5]. În ceea ce privește evoluția viitoare, Chen menționează dezvoltarea unor noi modele de defecțiuni comportamentale, cu o legătură mai bună spre nivelul porții logice și către defectele fizice, pentru a crește eficiența simulării de defecțiuni comportamentale. Totodată menționează și posibilitatea simulării paralele a mai multor defecte într-un banc de testare pentru a accelera simularea.

2.1.4 Simularea la nivel comportamental cu modele avansate

Hayne [8] a dezvoltat noi modele comportamentale în comparație cu [1], [4], [5], [6], [7] pe baza cercetării sale anterioare [9]. El a aplicat un set de vectori de testare extrași din modele de defecțiuni comportamentale la circuitele sintetizate la nivel de poartă logică care implementează funcții logice și aritmetice. Pentru a determina eficiența simulării la nivel comportamental cu modelele nou propuse, în prealabil a fost rulată o simulare la nivel de poartă logică cu modele clasice pentru a compara cele două rezultate.

Defecțiunea **Single Stuck Line (SSL)** de nivel poartă logică a fost introdusă în descrierile codului la nivel comportamental al circuitului. Informațiile necesare pentru analiza nivelului de poartă logica sunt determinate utilizând implementarea hardware specifică după sinteza constructelor VHDL utilizate pentru definiția circuitului. Cu utilizarea echivalenței de eroare și a dominanței, se extrage un set de defecte generalizate.

Pentru generarea vectorilor de testare s-au utilizat sintaxele **IF-THEN-ELSE** și **CASE**, folosind modelele lor de defecțiuni îmbunătățite. Au fost introduse recent defectele adiționale ale operatorilor VHDL (boolean, aritmetic, relațional) în lista modelelor de defecțiuni utilizate.

"Sinteza constructului **IF-THEN-ELSE** și **CASE** au ca rezultat arhitecturi hardware de tip multiplexor. Analiza funcțională produce un nou model de defecțiune în cazul în care canalele adiacente ale multiplexorului sunt alterate de două tipuri de defecțiuni comportamentale, **Clause-CORRUPT(OR)** și **Clause-CORRUPT(AND)**. Numărul defectelor **Clause-CORRUPT** depinde de complexitatea modelului și de numărul de adiacențe logice dintre clasele "[8].

Pentru a identifica numărul de defecte **WHEN-CORRUPT** pentru construcția **CASE**, diagrama lui Karnaugh este utilizată pentru a determina conexiunile logice dintre clauze, așa cum se arată în Fig. 2.3. Pentru fiecare clauză există două erori la nivel comportamental, două **WHEN-CORRUPT(OR)** și două **WHEN-CORRUPT(AND)**.

Exemplu:

```
case INP is
  when "00" =>      OUT <= X0;
  when "01" =>      OUT <= X1;
  when others =>    OUT <= X2;
end case;
```

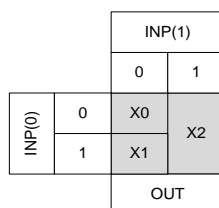


Fig. 2.3 Adiacențe logice ale clauzelor [22] (original).

2.2 Metode de simulare funcțională

În acest subcapitol se analizează tehnicile și metodele de simulare funcțională ale circuitelor integrate digitale și analogice mixte. Scopul studiului este de a colecta și a evidenția cele mai relevante metode de modelare și simulare, dar și de a spori cunoștințele din domeniul testării limbajelor HDL. Capitolul este organizat în 4 secțiuni. Prima secțiune este o introducere despre simulare în domeniul HDL.

Secțiunea a doua este o sinteză a mai multor lucrări și cercetări în acest domeniu specific. Studiul este organizat cronologic de la sfârșitul anilor 90 până în prezent, pentru a sublinia evoluția tehnicilor de simulare funcțională. A treia secțiune este o discuție despre rezultate. Ultima secțiune este dedicată concluziilor și a activităților viitoare.

Există trei tipuri majore de simulare:

- simulare defectelor
- simulare funcțională
- simulare de sincronizare a semnalelor intern

Simularea defectelor se bazează pe introducerea artificială a defecțiunilor conform modelelor de defecțiuni bine definite în descrierea VHDL fără defecțiuni. Fiecare descriere VHDL alterată este simulată și comparată cu rezultatele simulării codului VHDL fără defecte. Simularea se face folosind un vector de test care este generat de un algoritm ATPG, dacă rezultatele sunt diferite, înseamnă că eroarea injectată este detectată de vectorul de testare. Pentru fiecare defect injectat ar trebui să existe un vector de testare care să-l detecteze. Colectarea acestor vectori de testare este folosită de programul de testare pentru a controla echipamentul ATE și pentru a detecta eventualele erori de fabricare.

Simularea funcțională (prezice numai comportamentul logic al descrierii VHDL) datorită vitezei superioare a devenit soluția preferată în comparație cu simularea de propagare/sincronizare a semnalelor. Scopul simulării este de a genera ieșirile corecte pentru un anumit set de intrări. În timpul simulării funcționale, descrierea VHDL nu este încă sintetizată, astfel încât simulatorul nu ia în considerare detaliile de sincronizare/propagare/tehnologie/implementare.

Simularea de sincronizare/propagare a semnalelor este utilizată pentru a verifica corectitudinea designului cu parametrii de sincronizare, deci atât corectitudinea logică, cât și evoluția semnalelor în timp, în corelație cu celelalte semnale. Necesită un model fizic la nivel de poartă logică și este efectuat după ce descrierea VHDL a fost sintetizată. După simulare, este posibil să se găsească întârzieri (timpul de propagare de-a lungul diferitelor căi din circuitul sintetizat să

prezinte valori prea mari) și trebuie să se verifice dacă circuitul îndeplinește constrângerile de temporizare.

În Fig. 2.4 este prezentat un proces de dezvoltare ASIC simplificat. Pornind de la specificații, se dezvoltă descrierea comportamentală a circuitului. În timpul acestui proces se efectuează simulări de defecțiuni, de funcționare și de sincronizare/propagare a semnalelor interne.

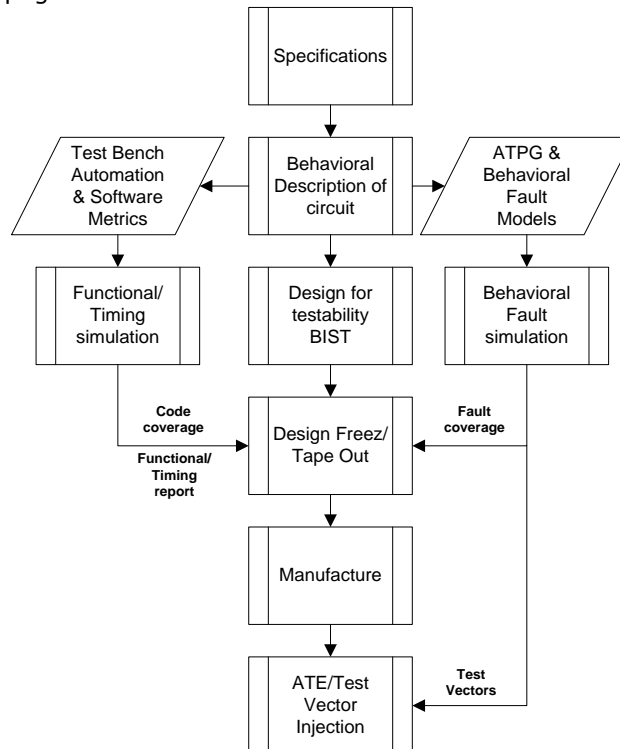


Fig. 2.4 Prezentare generală a simulării funcționale și simulării de erorilor [23] (original).

2.2.1 Simularea circuitelor integrate mixte

Papanuskas [12] prezintă metodologia de simulare a circuitelor integrate specifice (analogice) pentru domeniul automobilelor și o analiză generală a VHDL-1076.1 (VHDL-AMS - VHDL Analog Mixed Signal), precum și efortul lor de extindere a standardului VHDL 1076 pentru a susține descrierea și simularea circuitelor și sisteme mixte analogice / digitale.

Ca o extensie a limbajului IEEE VHDL 1076, obiectivul standardului IEEE VHDL 1076.1 este de a oferi o descriere complexă pentru simulare a circuitelor de semnal mixt. Cu această abordare este posibilă metodologia de proiectare „top-down”, standardul VHDL 1076.1 face posibilă simularea sistemelor digitale sau mixte pe diferite nivele de abstractizare utilizând orice instrument de simulare capabil să ruleze VHDL-AMS. Industria a cerut ca VHDL 1076.1 a fi un super-set de VHDL 1076, sprijinind [12]:

- descrierea și simularea modelelor de semnale analogice, digitale și mixte la mai multe niveluri de abstractizare
- reutilizarea sintaxei / codului VHDL 1076 existent

- domeniul de timp și simularea frecvenței semnalului mic al modelelor tehnologice mixte (altele decât cele electrice)
- descriere controlată a fluxului de semnal
- interacțiuni flexibile și eficiente între părțile discrete ale unui model și părțile sale continue.

Marino, Forliti, Rocchi, Giambastini, Iozzi, De Marinis, Fanucci [13] prezintă o metodologie de verificare comportamentală pentru un system-on-chip folosind VHDL-AMS, care este reprezentat în Fig. 2.5.

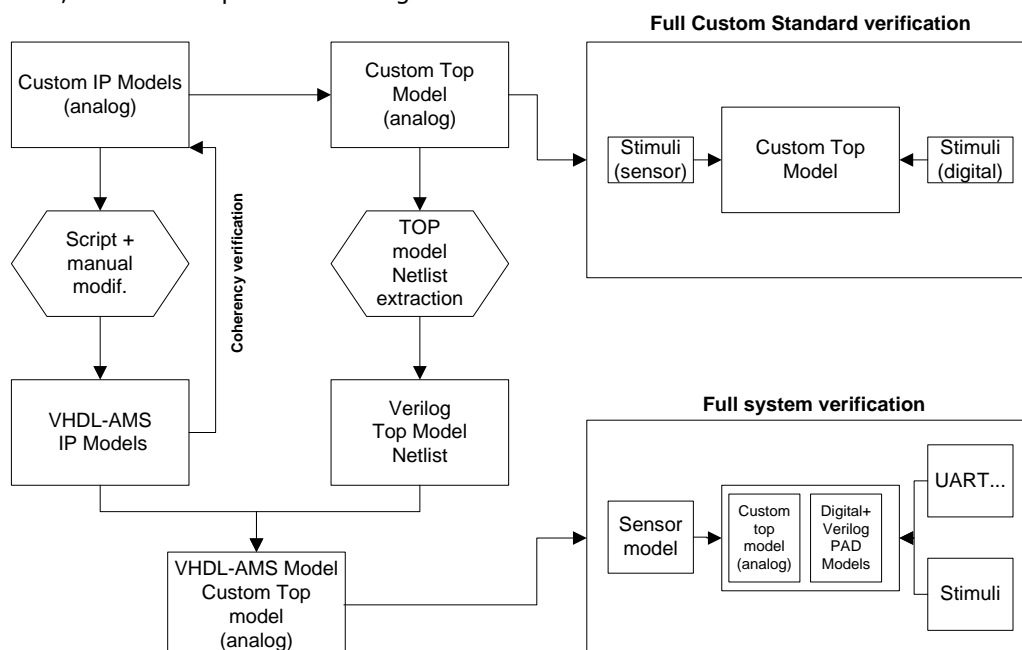


Fig. 2.5 Procesul de verificare [23] (original).

Pentru implementarea analogică, folosind un set de scripturi (C-Shell) se extrage automat un model VHDL-AMS. Modelul extras poate fi adaptat prin modificarea funcțiilor și parametrilor de bază. Utilizând șablonul VHDL-AMS personalizat, un model comportamental pentru blocul analog al SoC-ului (System on Chip) este obținut cu un proces automat care utilizează schema de analogică din Cadence™ și o transformă într-o listă de conexiuni ierarhică (netlist) Verilog a tuturor modelelor VHDL-AMS. Lista de conexiuni rezultată (pentru blocul analogic) este folosită pentru simularea și verificarea la nivel superior, împreună cu descrierea VHDL (blocul digital) și modelele Verilog a blocului de precondiționare a pinilor de intrare/ieșire a chip-ului. Obiectivul metodologiei de verificare menționat mai sus este de a reduce timpul de simulare/validare al SoC-ului și de a spori detectarea erorilor care nu au fost găsite la fel de eficient utilizând metodele anterioare.

Cecchini, Baldetti, Fenucci și Rocchi [14] au propus un mediu de testare a blocurilor analogice și digitale ale unui circuit integrat utilizând un flux semi-automat VHDL-AMS bazat pe [13]. Scripturile Python au fost folosite pentru a coordona simularea. Fluxul de testare este proiectat deoarece poate fi aplicat modelului „top”

Verilog sau chipului real; dispune de procesare de date în timp real, reutilizabilitate ridicată și flux complet de testare / validare.

Procesul de verificare a sistemului este cel propus în [13]. Modelul comportamental VHDL-AMS al macro blocurilor analogică sunt extrase din schema analogica la nivel de tranzistor (blocuri IP (Intellectual Property) / celule standard care sunt dependente de tehnologie/biblioteci). Aceste modele comportamentale pot fi personalizate în termeni de parametri de circuit, intern de către utilizator. Extragerea automată a listei de conexiuni - se executa pe baza următorului proces: "Odată ce toate modelele VHDL-AMS sunt disponibile și verificate, se poate efectua simularea de nivel superior (top level simulation): acest tip de verificare este eficientă numai cu condiția ca interconexiunile între blocurile din lista de conexiuni Verilog să fie la fel ca cele ale schemelor analogice originale. Pentru a păstra aceste conexiuni neschimbate, Cecchini et al. au dezvoltat un script CSH (C-Shell) numit `create_nets.csh` (a se vedea și Fig. 2.5). Scriptul scanează recursiv întreaga ierarhie analogică și prin intermediul Cadence™ "AMSdirect" extrage o listă de conexiuni Verilog coerentă cu structura bazei de date analogice" [14].

Rezultatul extragerii netlist-ului (analogice și digitale) este un fișier Verilog de nivel superior (top-level). Folosind acest fișier, o simulare la nivel de chip poate fi rulată reutilizând test bench-urile din simularea blocului digital, dar acum sunt disponibili feedback-ul și stimulii pentru blocul analogic, parcurgând aceeași procedură ca cea propusă în [13].

2.2.2 Validarea limbajului de descriere Hardware folosind metrici Software

Jusas și Neverdauskas [15] au dezvoltat un proces semiautomat pentru a genera stimuli de test pentru modele VHDL care conțin procese paralele. După sinteza descrierii, ei verifică vectorii de testare generați și pe plăci cu FPGA pentru a dovedi eficiența procesului de generare a vectorilor de test. În VHDL, sintaxa principală este declarația procesului (*process*). Partea de declarație a unei arhitecturi VHDL poate consta din mai multe procese concurente. La prima tranziție de tact al sistemului, toate instrucțiunile concurente sunt pornite și executate în paralel [15].

În programarea paralelă, în cele mai multe cazuri sunt folosite două abordări comune: „*lightweight process*” și „*heavyweight process*” [15].

În „*lightweight process*”, procesele partajează același spațiu de adrese, pe de altă parte în „*heavyweight process*” ele au propriul spațiu de adrese. Această abordare este mai generală, mai sigură, dar este mai complexă. Acest model de execuție este utilizat în VHDL. Comunicarea între procese se realizează utilizând semnale care sunt declarate în partea de declarație a arhitecturii [15].

Aceste semnale sunt vizibile pentru toate procesele care se află în arhitectură. Se impune restricția asupra drepturilor de scriere a acestor semnale; procesele singulare pot atribui valori numai anumitor semnale" [15].

Structura de testare de bază este prezentată în Fig. 2.6, se bazează pe activitatea lor anterioară [16] care se concentrează pe procese unice VHDL. Metodologia de testare a fost proiectată folosind limbajul Python.

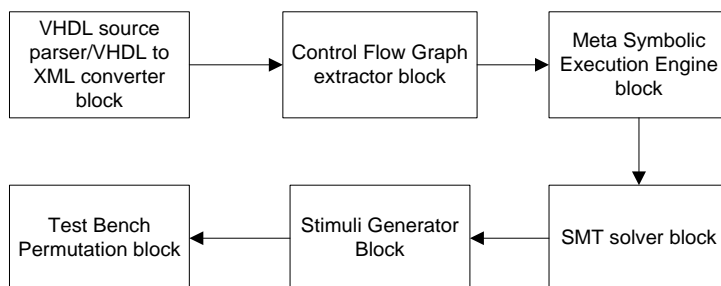


Fig. 2.6 Structura de bază a procesului de validare [23] (original).

Abordarea autorilor este de a extrage pentru fiecare proces un CFG (Control Flow Graph) pentru a crea o execuție simbolică a codului. În testarea software-ului, această metodă este bine cunoscută și folosită pe scară largă, dar validarea hardware-ului este încă în stadiu de dezvoltare. Una dintre primele cercetări care utilizează execuția simbolică a fost prezentată în [17].

Axiomă: "fiecare proiect descris în VHDL la nivel comportamental are cel puțin un proces principal. Pot exista mai multe procese principale dacă proiectul îndeplinește anumite funcții în mod independent. Având în vedere axioma putem distribui descrierile circuitelor în trei categorii:

Procesele independente: toate semnalele declarate în lista de sensibilitate a fiecărui proces sunt independente, iar procesele nu transferă informații. Execuția paralelă a fiecărui proces depinde numai de lista de sensibilitate și de valorile semnalului de la intrările primare.

Procesele explicite principal și secundar: procesul principal este activat de semnalul de tact. Procesele secundare sunt activate de semnalul generat de procesul principal.

Procesele implicite principal și secundar: mai multe sau toate procesele sunt dependente/lansate la semnalul de tact. Procesul de bază nu poate fi distins în funcție de sensibilitatea la semnalul de tact" [15].

Extractorul de CFG utilizând tipurile de proces definite mai sus stabilește relația dintre procesele dintr-o descriere VHDL la nivel comportamental.

Testarea fluxului de simulare se bazează pe realizările lui Jusas și Neverdauskas [16], rezultatele privind acoperirea prin testare a unui procent cât mai mare a codului sunt evaluate similar cu valorile de acoperire la testarea codului software (*software coverage metrics: statement, branch and toggle*) adică câte linii de cod au fost testate conținând declarații, câte linii de cod au fost testate conținând decizii care duc la schimbarea ramurilor de execuție (de exemplu într-un construct CASE) și în final câte linii de cod au fost testate (în urma executării lor duc la schimbarea unui semnal/variabile din cod).

2.3 Metodologii avansate de verificare

2.3.1 VMM - Verification Methodology Manual

Metodologia de verificare VMM (Verification Methodology Manual) a fost prima încercare de implementare, pe scară largă, pentru crearea de medii de verificare reutilizabile folosind System Verilog. Creat de Synopsys, unul dintre puternicii promotori ai System Verilog, VMM folosește caracteristici lingvistice, cum ar fi programarea orientată pe obiecte, randomizare, constrângeri, acoperire funcțională pentru a permite atât începătorilor, cât și experților să creeze medii de verificare puternice. VMM a contribuit cu un factor important în crearea UVM (Universal Verification Methodology), succesorul lui VMM [18], [19].

Această metodologie permite o productivitate mai mare, reutilizare și interoperabilitate între simulatoare. VMM furnizează o bibliotecă de clase și definește cele mai bune practici din industrie prin reguli de codare. Setul de reguli și recomandări deschide calea pentru crearea unor medii de verificare aleatorii la nivel de tranzacție, utilizând SystemVerilog.

VMM definește standardele pentru specificarea proprietăților reutilizabile care sunt eficiente pentru a simula și care pot fi verificate oficial. Definește standarde pentru crearea de descrieri de tranzacții și date pentru a facilita generarea lor aleatorie constrângătoare, menținând în același timp o capacitate flexibilă direcționată. Această metodologie standardizează modul în care modelele, monitoarele și tranzactorii cu funcționalitate de magistrală sunt proiectați să furnizeze funcții de stimulare și verificare relevante de la sistem la bloc. În plus, această metodologie stabilește standarde pentru integrarea diferitelor componente într-un mediu de verificare, astfel încât acestea să poată fi ușor combinate, controlate și ulterior rulate în medii diferite de simulare fără a mai modifica descrierea testelor. Metodologia definește de asemenea standardele pentru implementarea modelelor de acoperire și a mediilor de verificare a software-ului. Aceste standarde, atunci când sunt întrunite într-o metodologie coerentă, ajută la reducerea efortului necesar pentru verificarea unui design.

2.3.2 UVM - Metodologia universală de verificare

Metodologia universală de verificare UVM este o metodă standard de verificare de la Accellera, dezvoltată de către comunitatea de verificare pentru comunitatea de verificare. UVM reprezintă o noutate în cadrul tehnologiei de verificare și este concepută astfel încât să permită crearea unor IP-uri de verificare interoperabile și test-bench-uri robuste și reutilizabile.

Limbajul SystemVerilog a fost dezvoltat pentru a ajuta inginerii de test HDL. Limbajul prezintă câteva mecanisme, cum ar fi clasele, grupurile de acoperire și constrângerile care au facilitat anumite aspecte ale verificării unui design digital și apoi au început să apară noi metodologii de verificare bazate pe UVM.

Cu ajutorul UVM se pot crea test-bench-uri SystemVerilog personalizate pentru verificarea pe bază de acoperire CDV (Coverage Driven Verification).

Avantajele CDV menționate în Ghidul utilizatorului UVM [20]:

- CDV combină generarea automată a testelor, test-bench-urilor cu auto-verificare, folosind metrici de acoperire pentru a reduce semnificativ timpul petrecut pentru verificarea unui design
- este eliminat efortul și timpul petrecut pentru a crea sute de teste
- asigură o verificare aprofundată folosind obiective de verificare stabilite înainte de a porni testele
- se primesc notificări de eroare timpurie și se implementează verificarea timpului de execuție și analiza erorilor pentru a simplifica depanarea codului

Fluxul CDV este diferit de fluxul tradițional de testare direcționată. Cu CDV, se începe prin stabilirea obiectivelor de verificare utilizând un proces de planificare organizat. Apoi se creează un test-bench inteligent care generează stimuli și îi trimite la DUT. Monitoare de acoperire sunt adăugate în mediul înconjurător pentru a măsura progresul și pentru a identifica funcționalitățile netestate. Se adaugă Verificatori pentru a identifica comportamente neașteptate/defectuoase ale DUT-ului. Simulările sunt lansate după implementarea modelului de acoperire și a test-bench-urilor"[20].

Alte caracteristici:

- modificarea parametrilor test-bech-ului sau modificarea algoritmului de randomizare pentru crearea vectorilor de test
- pot fi adăugate constrângeri de testare în partea superioară a infrastructurii inteligente
- metrici de clasificare pentru identificarea testelor care contribuie la îndeplinirea obiectivelor de verificare și pentru eliminarea testelor redundante
- facilitează atât testele direcționate cât și cele aleatorii cu constrângeri.

2.3.3 OVM - Metodologia liberă de verificare

OVM este o metodologie de verificare funcțională folosind SystemVerilog, completată cu o bibliotecă de suport a codului SystemVerilog. OVM reprezintă metodologia liberă de verificare. OVM a fost creat de către Cadence și Mentor pe baza metodologiilor de verificare existente, care au fost inițiate în cadrul acestor două companii. Rădăcinile acestor metodologii constau în aplicarea limbajelor IEEE1800™ SystemVerilog, IEEE 1666™ SystemC și IEEE1647™ [21].

OVM este o metodologie pentru verificarea funcțională a hardware-ului digital, folosind în primul rând simularea. Hardware-ul sau sistemul de verificat este descris de obicei folosind Verilog, SystemVerilog, VHDL sau SystemC la orice nivel de abstractizare corespunzător. Descrierea poate fi comportamentală (*behavioural*), la nivel de registru de transfer (*register transfer level*) sau nivel de poară (*gate level*). OVM este în mod explicit orientat spre simulare, dar OVM poate fi folosit și împreună cu verificarea bazată pe aserțiuni, accelerarea hardware sau emulare.

Dacă se execută simulări RTL (*Register Transfer Level*) în Verilog sau VHDL, OVM se comportă ca înlocuitor pentru orice modalitate și stil de codare utilizat pentru bancurile de testare create. Bancurile de testare OVM sunt mai mult decât test bench-urile HDL tradiționale, care schimbă starea unor pini în design-ul testat și se bazează pe cel care rulează simularea pentru a inspecta o diagramele formelor de undă pentru

a verifica funcționarea corectă. Mecanismele de testare OVM sunt medii complete de verificare compuse din componente de verificare reutilizabile și utilizate ca parte a unei metodologii globale de verificare constrânsă.

Simularea ar putea fi descrisă ca fiind procesul de injectare a vectorilor de testare într-un model al design-ului de testat și observarea modului în care se comportă acest model. Un test bench tradițional Verilog sau VHDL ar putea conține procese pentru a citi vectori sau comenzi dintr-un fișier, folosite pentru a modifica în timp valorile semnalelor de intrări conectate la DUT, iar semnalele de ieșire din DUT sunt salvate într-un alt fișier.

Acest lucru este eficient pentru model simple de VHDL sau Verilog, însă acest proces nu se scalează ușor pentru a sprijini verificarea fiabilă a sistemelor complexe.

Verificarea sistemelor complexe nu ar trebui să se bazeze pe inspecția manuală a formelor de undă și a seturilor de vectori. Verificarea funcțională trebuie să fie automatizată, la fel ca și colecția metricilor de verificare (acoperirea testelor din planul de verificare și numărul de erori identificate de fiecare testare). Împreună cu planul de verificare, verificarea automată, colectarea și analiza de acoperire funcțională sunt bazele oricărei metodologii de verificare bune fiind abordate în mod explicit de SystemVerilog și OVM. Verificatoarele (*checkers*) și un model de acoperire funcțională legate de planul de verificare, necesită timp de dezvoltare pentru a fi create, dar au ca rezultat o calitate îmbunătățită a verificării.

Toate verificările bazate pe simulare suferă din cauza faptului că niciodată nu se pot rula suficienți vectori de testare pentru a verifica exhaustiv întregul design sau măcar o parte semnificativă a unui design complex.

O modalitate de a aborda această problemă este utilizarea unui stimul aleatoriu-constrâns pentru semnalele de intrare. Folosirea stimulului aleatoriu aduce două beneficii foarte importante. În primul rând, stimulul aleatoriu este excelent pentru descoperirea unor bug-uri neașteptate, dat fiind faptul că, având suficient timp și resurse, el poate permite design-ului să fie explorat fără condiții inițiale sau posibila prea direcționate legate strict de gândirea celui (inginerului de testare) care ar scrie un test-bench. În al doilea rând, stimulul aleatoriu permite utilizarea maximă a resurselor de calcul distribuite pe mai multe servere sau prin rularea testelor pe timp de noapte fără ca să fie nevoie de intervenție umană. Bineînțeles, stimulul pur aleator ar fi lipsit de sens. O parte importantă a procesului de verificare susținută în mod explicit de SystemVerilog și OVM este adăugarea de constrângeri pentru a face util un stimul arbitrar.

Cea mai bună modalitate de abordare a procesului de verificare este să înceapă cu teste simple (non-aleatoare) pentru a porni testarea proiectului. Ulterior se trece la teste complet aleatorii pentru a explora spațiul de stări într-un mod larg și pentru a elimina cât mai multe bug-uri posibil cu efort minim uman dedicat scrierii testelor manuale. Acest lucru va avea de obicei o acoperire funcțională mult mai mică de 100%, iar restul procesului de verificare este petrecut definind o serie de teste, fiecare dintre acestea constrânge și modelează stimulul aleatoriu, fiind un mod diferit de a forța designul în cazuri interesante de funcționare. Spațiul de stări al unui design tipic este atât de vast încât stimularea aleatorie nu este suficientă pentru a explora toate stările și cazurile de funcționare, dar testele direcționate sau foarte constrânse pot fi prea insuficiente pentru a oferi o acoperire generală bună. Stimularea arbitrară constrânsă este un compromis între cele două extreme, dar utilizarea eficientă se reduce la a face o serie de judecăți bune de testare. Soluția este de a utiliza prioritățile stabilite în planul de verificare pentru a direcționa resursele de verificare către zonele și stările de funcționare importante.

Verificarea aleatorie limitată se bazează pe Verificatoare (Checkers), Acoperire (Coverage) și Constrângeri (Constraints). În primul rând, verificatorii asigură corectitudinea funcțională. Injectarea a tot mai mulți stimuli aleatorii într-un design pentru a crește acoperirea funcțională la niveluri tot mai ridicate, nu este un beneficiu cu excepția cazului când DUT-ul este verificat automat pentru corectitudine funcțională. Verificatoarele pot fi implementate folosind aserțiuni (assertions) SystemVerilog sau utilizând codul procedural obișnuit. Aserțiunile pot fi încorporate în DUT, plasate pe interfețele externe sau pot face parte din mediul de verificare.

OVM furnizează mecanisme și ghidare pentru construirea Verificatorilor folosiți în mediul de simulare iar în final pentru crearea automată a rapoartelor de verificare/simulare.

În al doilea rând, testele de acoperire oferă o măsură a corectitudinii funcționale a testării și semnaleză când s-au îndeplinit obiectivele stabilite în planul de verificare și astfel, când s-a terminat simularea. SystemVerilog oferă două mecanisme separate pentru analiza acoperirii funcționale (Functional Code Coverage); acoperirea bazată pe proprietăți și acoperirea pe bază de eșantioane. Ambele pot fi utilizate într-un mediu de verificare OVM. Specificarea și executarea modelului de acoperire este strâns legată de planul de verificare și multe instrumente de simulare sunt capabile să adauge informații noi pe parcursul simulării de acoperire funcțională în planul de verificare, facilitând controlul simulării și generarea constrângerilor pentru testele ce urmează.

În al treilea rând, constrângerile oferă mijloacele de atingere a obiectivelor de acoperire prin modelarea stimulului aleatoriu pentru a conduce DUT-ul în stări funcționale interesante. Fără modelare, stimulul aleatoriu poate fi insuficient pentru a exercita toate stările DUT-ului. Stimularea arbitrară constrânsă este în continuare aleatoare, dar distribuția statistică a vectorilor este astfel concepută încât să se asigure că sunt atinse toate stările/cazurile de funcționare. SystemVerilog are caracteristici lingvistice dedicate pentru exprimarea constrângerilor, iar OVM merge mai departe prin furnizarea de mecanisme care permit ca constrângerile să fie scrise ca parte a unui test, mai degrabă decât încorporate în componentele de verificare dedicate. Aceasta și alte caracteristici ale OVM facilitează crearea unor componente de verificare reutilizabile.

Testele direcționate sunt scrise cu scopul de a împinge designul în anumite stări și de a exercita cazuri specifice. Cu testarea aleatorie constrânsă, rolul testelor se schimbă ușor. Deși un test de aleatoriu cu constrângere poate fi scris cu obiective de acoperire specifice, nu se presupune că un anumit test va verifica de fapt o altă caracteristică decât cea pentru care a fost conceput. Testul aleatoriu cu constrângeri este rulat, iar modelul de acoperire este folosit pentru a determina empiric ce caracteristici/cazuri ale DUT-ului au fost atinse de testul rulat. Testele pot fi clasificate ulterior după datele de acoperire, iar cele mai eficiente teste, adică cele care obțin cea mai mare acoperire în cel mai mic număr de cicluri, pot fi folosite pentru a forma baza unui set de teste de regresie.

Productivitatea verificării poate fi îmbunătățită prin reutilizarea componentelor de verificare și acesta este un obiectiv important al OVM. Verificarea reutilizării este activată prin existența unui mediu modular de verificare, în care fiecare componentă are responsabilități clar definite, permițând flexibilitatea modului în care componentele sunt configurate și utilizate, prin crearea unui mecanism care să permită personalizarea componentelor importate și prin ghidare/standardizare bine definită a modului de scriere a codului pentru a asigura coerența.

Arhitectura OVM a fost concepută pentru a încuraja mediile de verificare modulare pe diferite niveluri, în care componentele de verificare de la toate nivelurile pot fi refolosite în medii diferite. Întregul mediu de verificare poate fi reutilizat de către mai multe teste și configurat de automat de către aceste teste. În cele din urmă, scenariile de testare pot fi reutilizate de la aplicație la aplicație. Acest grad de reutilizare este posibil prin faptul că

componentele de verificare OVM pot fi configurate într-un mod flexibil fără modificarea codului lor sursă. Această flexibilitate este integrată în biblioteca de clase OVM.

2.4 Concluzii și contribuții

2.4.1 Rezumat

De-a lungul anilor a avut loc un proces de îmbunătățire a simulării defectelor comportamentale așa cum se arată în Fig. 2.7 [22] pornind de la ideile inițiale ale lui Ward și Armstrong cu privire la un sistem complet ATPG cu modele de defecte și submodule îmbunătățite.

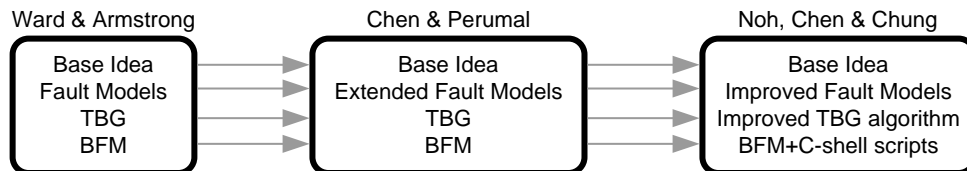


Fig. 2.7 Îmbunătățiri legate de simularea defectelor comportamentale[22] (original).

Ward și Armstrong [1] au dezvoltat două instrumente pentru simularea defecțiunilor VHDL: BFM și TBG. "BFM este un instrument eficient pentru crearea automată a modelelor VHDL cu defecte, dintr-un model sursă fără defecte și o listă de erori. Au fost descrise opt modele de defecțiuni comportamentale și modul în care sunt injectate în fișierul sursă pentru a crea fișierele de către BFM. Datorită dimensiunilor semnificative ale test-bench-ului tipic de testare a simulării defectelor, autorii au creat un algoritm îmbunătățit, TBG-ul care oferă o metodă eficientă de testare. Metoda este ușor de utilizat pentru a produce test-bench-ul pentru simularea defecțiunilor și fișierele de suport aferente simulării" [1].

Chen și Perumal [4] au analizat diferența dintre simularea la nivel de poartă logică și cel la nivel comportamental prin studierea defectelor comportamentale propuse în [1] și a efectului lor la nivelul porților logice. Ei au ajuns la concluzia: chiar dacă acoperirea de testare a simulării defectelor comportamentale este mai mică decât cea la nivel de poartă logică, fluxul de lucru pentru simularea defectelor comportamentale este foarte simplu și nu necesită interacțiune umană.

Noh, Chen și Chung [5] au implementat un sistem comportamental de ATPG și simulare a erorilor bazat pe cel propus în [1]. Modelele de defecțiuni comportamentale utilizate pentru generarea modelelor de testare s-au bazat pe cele propuse în [1], dar îmbunătățite pentru compatibilitatea cu alte tipuri de date. "Rezultatele simulării de eroare arată că un număr relativ mic de modele de testare generate de sistemul ATPG a detectat aproximativ 98% din toate defecțiunile testabile ale porților" [5].

Noile modele de defecțiuni dezvoltate de Heyne [8] "oferă o legătură bine definită cu defectele de bază care lipseau în modelele anterioare de defecțiuni comportamentale. Analiza funcțională SSL a ajutat la determinarea relațiilor VHDL /

hardware după procesul de sinteză. Această analiză a defectelor la nivelul porților logice adaptează tehnicile funcționale utilizate de Abraham [10] și Hayes [11] cu două extensii importante:

- defectele funcționale nu sunt legate de o implementare specifică, ele sunt generalizate pentru a fi aplicabile la mai multe rezultate ale sintezei
- defectele sunt abstracte în domeniul comportamental prin relația lor cu construcția VHDL originală" [8].

Prezentul capitol este un studiu al mai multor lucrări în metodele de simulare a erorilor HDL dezvoltate începând cu sfârșitul anilor 80 până în prezent. Datorită complexității crescânde a ASIC-urilor, este necesar să se dezvolte întotdeauna o metodă de testare mai bună și mai rapidă pentru a garanta un design sigur și un timp scurt de introducere pe piață.

În ultimii 25 de ani, ideea de bază nu sa schimbat, doar câteva modele noi de defecte comportamentale, instrumente comerciale și putere de calcul au fost introduse ca îmbunătățiri. Avantajele și utilitatea diferitelor modele comportamentale au fost prezentate prin compararea a mai multor articole de referință [1], [4], [5] - [8].

2.4.2 Concluzii și contribuții subcapitolele 2.2 și 2.3

În subcapitolele 2.2 și 2.3 este analizată evoluția și metodologia simulării funcționale pe baza lucrării al cărui autor sunt [23], din ea am selectat Fig. 2.8 o hartă a timpului în domeniul simulării limbajului HDL.

Datorită creșterii complexității designului a fost necesar să se dezvolte strategii de verificare mai inteligente și mai rapide, acesta fiind motivul unui proces continuu de dezvoltare în ceea ce privește metodologiile de simulare / validare.

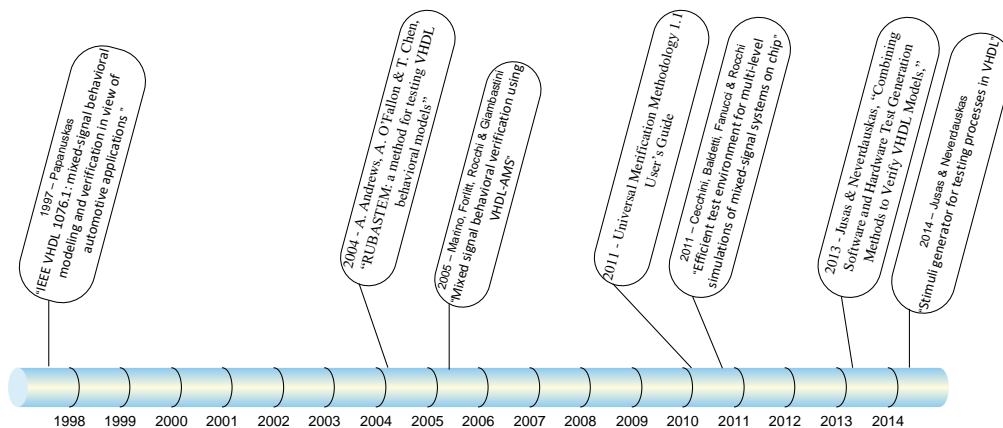


Fig. 2.8 Cronologie; Evoluția simulării funcționale și a defecțiunilor [23] (original).

Papanuskas în lucrarea sa [12] a prezentat necesitatea unui nou standard HDL care să fie adecvat și pentru modelarea comportamentală a circuitelor integrate mixte propunând o metodologie de simulare bazată pe limbajul standard VHDL1076.

Marino, Forliti, Rocchi, Giambastini, Iozzi, De Marinis, Fanucci [13] și Cecchini, Baldetti, Fenucci și Rocchi [14] au introdus metodologia de simulare și verificare a circuitelor integrate cu semnale mixte la un nivel mai eficient și mai complex. Pe baza metodologiei prezentate în [13], au fost îmbunătățite prin includerea scripturilor CSH și Python pentru a automatiza procesul și pentru a le face complet reutilizabile pentru alte medii de simulare, precum și pentru testele de laborator a chip-urilor reale.

Jusas și Neverdauskas [15] au propus o metodologie semi-automată pentru testarea proceselor VHDL paralele, eficacitatea metodologiei se determină prin calcularea procentului de acoperire a codului utilizând metrice asemănătoare cu cele utilizate pentru software.

Capitolul precedent este un studiu al mai multor lucrări și cercetări în metodele de simulare funcțională HDL dezvoltate începând cu sfârșitul anilor '90 până în prezent. Datorită complexității crescânde a IC-urilor (Integrated Circuit), este necesar să se dezvolte întotdeauna o metodă de testare mai bună și mai rapidă pentru a garanta un design sigur și un timp scurt de introducere pe piață.

În domeniul simulării funcționale, în special în cazul aplicațiilor de circuite integrate cu semnale mixte, stadiul tehnicii are un nivel mai ridicat de complexitate datorită introducerii VHDL-AMS și a softurilor puternice de simulare. Avantajele și utilitatea diferitelor metodologii de simulare au fost prezentate prin prisma a patru cercetări [12], [13], [14] și [15].

În ultimii ani s-au dezvoltat tot mai mult metodologiile de verificare (VMM, UVM, OVM) bazate pe System Verilog create de grupuri de lucru din domeniul HDL cu scopul de a armoniza metodologiile de verificare care oferă flexibilitate, portabilitate și reutilizare superioare față de vechile metode bazate pe diverse scripturi particularizate ale diverselor grupuri individuale de lucru din domeniul HDL.

În final în Tabelul 2a. și 1b. am prezentat succint contribuția autorilor lucrărilor consultate în acest capitol.

Tabel 2a. Contribuția autorilor din domeniul "Simulare funcțională și de erori"

Cine	Ce a/au făcut	Anul	Avantaje	Dezavantaje
Simulare funcțională și de erori				
Ward și Armstrong	Au propus modele de testare a erorilor la nivel de poartă logică	1990	Algoritm TBG Algoritm BFM	Definesc doar un set restrâns de erori
Chen și Perumal	Au continuat ideile lui Ward și Armstrong	1993	Set extins de erori față de cele precedente	Timp de simulare semnificativ pentru a testa întregul design
Noh, Chen și Chung	Au îmbunătățit modelele de erori din setul extins	1994	Algoritm BFB îmbunătățit Au introdus Script-uri pentru automatizarea procesului de testare	Portabilitate și reutilizare ineficiente la sisteme noi
Hayne	Noi modele de defecțiuni Generalizarea defecțiunilor	2011	Analiza funcțională SSL Portabilitate ridicată	Acoperire de detectare a erorilor doar de 98% Restul erorilor trebuie direcționate manual

Tabel 2b. Contribuția autorilor din domeniul HDL "Simulare comportamentală"

Cine	Ce a/au făcut	Anul	Avantaje	Dezavantaje
Simulare comportamentală				
Papanuskas	Contribuții în extinderea standardului VHDL pentru circuite integrate mixte	1997	Metodologie de verificare a IC-urilor mixte	Portabilitate scăzută datorită particularităților metodologie de verificare
Marino et al.	Metodologie eficientă și complexă de testare a IC-urilor cu semnale mixte	2005	Scripturi CSH și Python Teste reutilizabile	Specializat pentru SoC-uri
Cecchini et al.	Au propus un mediu de testare a IC analog-digital bazat pe cercetările lui Marino et al.	2011	Modele VHDL-AMS extrase din schema la nivel de tranzistori a implementării reale pe chip	Proces elaborat și costisitor pentru a genera întregul setup de simulare
Jusas și Neverdauskas	Au dezvoltat un proces semiautomat pt. generare de stimuli de test	2013	Tratează procese paralele VHDL	Portabilitate și reutilizare scăzute
Comunitate de testare/ grup de lucru din domeniul HDL	VMM – UVM - OVM	2011 - prezent	Portabilitate și reutilizare eficientă Independent de software-ul utilizat datorită faptului că sunt bazate pe System Verilog	

3. Arhitecturi de validare și simulare a circuitelor integrate dedicate

Capitolul 2 a tratat testarea codului VHDL (la nivel de poartă, dar și comportamental) care stă la baza blocului digital al oricărui ASIC și testarea blocului analogic prin extragerea unor modele comportamentale din schema analogică la nivel de tranzistor a implementării particulare a unui ASIC; toate aceste teste se realizează în faza premergătoare fabricării fizice a ASIC-ului.

În acest capitol voi prezenta metodele folosite în prezent la testarea ASIC-ului, atunci când el este deja fabricat, dar voi propune și o metodă alternativă de testare atunci când doar codul VHDL este disponibil și se cunoaște arhitectura analogică, dar implementarea în siliciu încă nu este disponibilă, tot ASIC-ul fiind emulat de diverse componente (de exemplu blocul digital de către un FPGA, iar blocul analogic de către componente discrete interconectate conform arhitecturii ASIC-ului).

Suplimentar față de testarea fizică prin emulare, prezint metodele actuale și propun o metodă alternativă de simulare a blocului digital și cel analogic pentru testarea arhitecturii, atunci când încă nici ASIC-ul nu este fabricat și nici setup-ul care emulează funcția ASIC-ului nu sunt disponibile.

Cel mai important proces în faza de dezvoltare a unui ASIC după faza de proiectare este verificarea. Pentru un ASIC complex, procesul de verificare poate dura mai mult decât faza de proiectare.

Pentru domeniul automobilelor, procesul de dezvoltare este de obicei 2-3 ani, din care 1 an este faza de proiectare - restul este verificarea în raport cu parametrii de proiectare, acordând o atenție deosebită standardelor de siguranță implementate în design [24].

Într-o arhitectură standard ECU (Engine Control Unit), fiecare ASIC este supravegheat de un microcontroler printr-un protocol de comunicație.

Pentru fiecare ASIC nou, driverele și software-ul trebuie să fie dezvoltate, de asemenea; aici cel mai bun caz este o activitate paralelă în faza de proiectare a ASIC-ului. În realitate, există întotdeauna o întârziere din cauza timpului necesar pentru fabricarea primelor mostre de ASIC, așa-numitul "siliciu A0", prima etapă de siliciu [25] deoarece driverele nu pot fi testate până la existența fizică a circuitului integrat.

Pentru un proces de dezvoltare eficient a software-ului, este important să existe o platformă virtuală într-un simulator care să conțină un microcontroler, ASIC-ul, electronica de putere și în final, modelele de sarcină/actuator.

3.1 Arhitecturi de validare hardware a circuitelor integrate dedicate

În domeniul automobilelor și în special în cazul ECU-urilor, datorită standardelor stricte de siguranță, validarea hardware-ului este o necesitate [26]. În funcție de faza de dezvoltare a ASIC-ului, funcția sa este testată cu sau fără modulul de care aparține.

Modulul este colecția componentelor discrete necesare pentru a facilita funcțiile ASIC-ului, de exemplu: condensatori de decuplare (1uF/35V/X7R), tranzistoare de putere (putere disipată >1W) controlate de ASIC, rezistențe de

măsură a curentului (putere disipată $>1W$), practic toate componentele care nu pot fi integrate în ASIC datorită dificultăților fizice și procesului de fabricare a ASIC-ului. De exemplu, nu este posibilă implementarea în siliciu a unui condensator de valoare și tensiune mare (100nF/100V/X7R) sau a unui tranzistor cu capacitate mare de curent, care va extinde excesiv suprafața de siliciu, prin urmare și dimensiunea capsulei ASIC-ului.

Dezvoltarea blocurilor analogice și digitale se desfășoară în paralel, dar este realizată de echipe diferite [27]; în faza timpurie a dezvoltării când blocurile analogice și digitale ale ASIC sunt gata - dar nu împreună, un TC (Test Chip) este fabricat fără blocul digital, cu toate circuitele analogice deja integrate (ADC, surse de curent, etaje push-pull, tranzistori de ieșire de putere și așa mai departe).

3.1.1 Platforma modulară de test propusă

Propunerea dezvoltată în cadrul tezei este potrivită pentru validarea întregului sistem într-o fază timpurie, când numai TC-ul (Test Chip) este fabricat fizic. Pentru această fază, propun arhitectura de validare MTP (Modular Test Platform) cu platforme individuale pentru fiecare componentă principală a ECU-ului, după cum se arată în Fig. 3.1.

Platforma de validare este construită pe o placă principală, având rolul de interconectare și suport mecanic pentru diferitele PCB-uri (Printed Circuit Board) implementând diversele blocuri necesare. Există o placă cu sursa de alimentare principală, care generează tensiuni de alimentare de 1V2, 3V3 și 5V dintr-o baterie sau o sursă externă de alimentare.

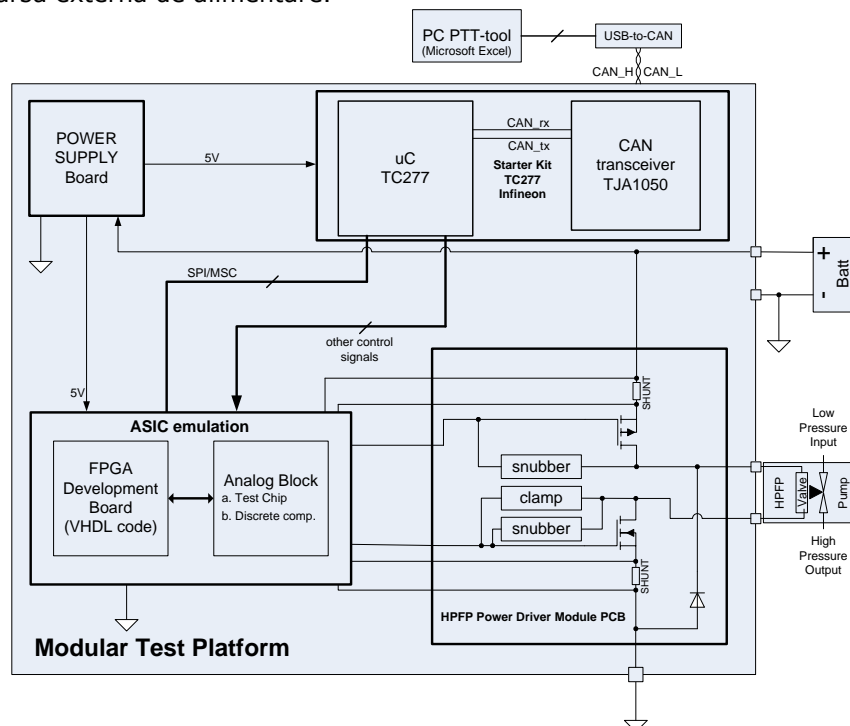


Fig. 3.1 Platforma modulară de test propusă [34] (original).

Emularea ASIC se realizează prin două componente. Blocul digital este emulat de o placă de dezvoltare FPGA care implementează codul VHDL. Pentru emularea blocului analog, sunt disponibile două opțiuni diferite.

Prima opțiune este de a construi un PCB cu componente discrete [28], care îndeplinesc funcțiile blocului analogic ASIC; avantajul acestei opțiuni este testarea blocului digital înainte de fabricarea TC-ului (design analogic). Propun o a doua opțiune care ar putea folosi însăși TC-ul care conține doar blocurile analogice ale ASIC-ului, dar aceasta înseamnă că designul digital și analog trebuie să fie terminat aproape în același timp. Principalul avantaj comparativ cu soluția anterioară este că blocul analog este foarte asemănător cu ASIC-ul, deci comportamentul circuitului este același.

În ambele cazuri, blocul analogic este conectat la placa de dezvoltare cu FPGA, deci la codul VHDL care definește funcțiile blocului digital.

Avantajul utilizării TC-ului ca bloc analogic este faptul că acesta conține circuitul analogic exact care va fi în interiorul ASIC-ului și astfel toate efectele parazitare vor fi reflectate în funcționalitatea analogică.

ASIC-ul ar putea avea diferite funcții, de exemplu așa cum se arată în Fig. 3.1 un circuit de comandă în buclă închisă cu control de curent pentru HPFP (High Pressure Fuel Pump). Circuitul de putere pentru comanda HPFP este proiectat pe un PCB care utilizează componente discrete. Acest modul este o topologie Half-Bridge care conține un MOSFET cu canal-P (HSD - High Side Driver) și un MOSFET cu canal-N (LSD - Low Side Driver) și circuite suplimentare, snubber, recircularea energiei (free-wheel pe HS și active clamping pe LS), rezistențe de măsură a curentului.

Pentru circuitul suplimentar se poate folosi o sarcină înlocuitoare sau o sarcină reală.

O sarcină înlocuitoare/echivalentă este un rezistor de putere în serie cu un inductor care simulează parametrii bobinei actuatorului (valvei) reale.

Sarcina reală în acest caz este o pompă HPFP, alcătuită din două elemente, pompa și valva electro-mecanică. Pompa este acționată de lobii arborelui cu came al motorului termic; o pompă de combustibil de joasă presiune alimentează cu combustibil camera compresie a pompei HPFP. Cantitatea de combustibil este controlată de supapa de scurgere (în mod normal deschisă); după ce supapa este închisă, când pistonul pompei este acționat de un lob al arborelui cu came al motorului, se va forma o presiune ridicată și va fi livrată în rampa principală, de unde sunt alimentate injectoarele care în final pulverizează combustibil individual în fiecare cilindru al motor.

Scopul strategiei de control al HPFP este acela de a menține o presiune constantă în rampa comună, astfel încât injectoarele să poată livra aceeași cantitate de combustibil cu scopul de a nu se livra combustibil excesiv în camera de ardere, astfel încât să aibă loc o combustie completă cu emisii reduse.

Injectoarele fiind controlate temporal, dacă presiunea variază, rezultă că și cantitatea de combustibil va varia pentru aceeași lățime de puls (cât timp sunt injectoarele deschise) astfel ducând la posibile situații când se injectează prea mult combustibil deși nu este nevoie.

Avantajul general al MTP este flexibilitatea din punctul de vedere al actualizărilor, orice modificare a blocurilor individuale poate fi rezolvată cu ușurință, astfel încât celelalte blocuri rămân la fel și nu trebuie construite din nou, fiind suficientă re-proiectarea PCB-ului cu blocul dorit a fi modificat.

Din punctul de vedere al validării, un factor important este accesul la semnalele din "interiorul ASIC-ului", conexiunile dintre blocul digital și cel analogic, care nu sunt accesibile în cazul în care produsul final (ASIC-ul fabricat) este utilizat pentru validare și nu pentru emularea sa.

Un dezavantaj al MTP este complexitatea interconexiunilor dintre blocurile individuale care ar putea introduce oscilații / modificări temporale ale semnalelor de viteză mare datorită rezistenței de contact a conectorilor și a lungimii excesive a traseelor de pe PCB.

După ce designul digital și analog a fost validat folosind MTP, ASIC-ul este fabricat, implementând modificările necesare eventualelor probleme descoperite în timpul validării.

3.1.2 Placa de validare modulară

Următorul pas de validare este utilizarea MVB-ului (Modular Validation Board), după cum se arată în Fig. 3.2. Scopul acestei platforme de validare este de a testa în ASIC-ul deja fabricat împreună cu modulul din care face parte. MVB conține acum un ASIC fabricat împreună cu toate componentele discrete ale modulului, aspectul plăcii PCB este optimizat și asemănător cu stilul produsului final, un ECU.

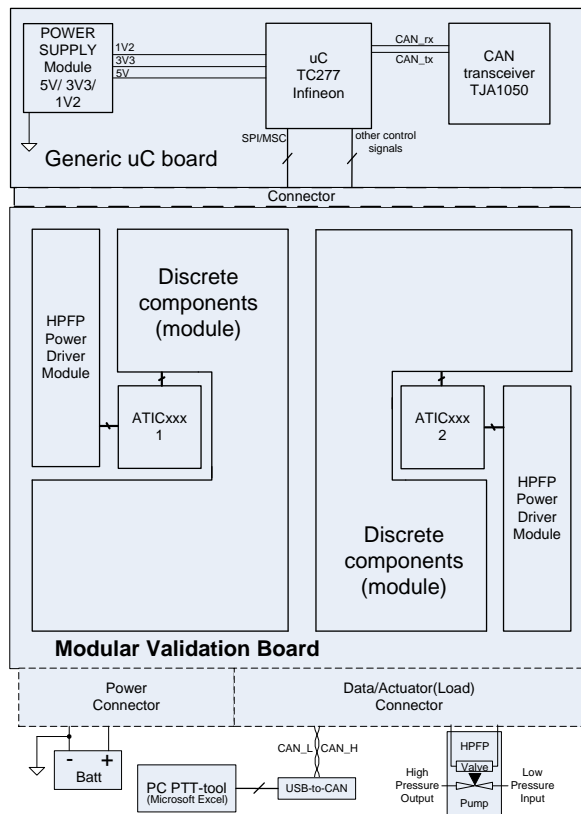


Fig. 3.2 Placa de validare modulara [34] (original).

Microcontrolerul, sursa de alimentare și transmițătorul CAN se află pe un PCB separat și sunt conectate la MVB printr-un conector multi-pin. Avantajul plăcii separate cu uC (Microcontroller) este flexibilitatea: atunci când se testează un nou ASIC împreună cu modulul său, numai MVB trebuie înlocuit pentru a testa diferite configurații ale aceluiași modul.

MVB este echipat cu un conector similar ca și produsul final de pe mașină, incluzând un mănunchi de cabluri de lungime egală cu cea din aplicație pentru a alimenta diversele sarcini a fi controlate. Interfața cu utilizatorul este aceeași cu cea utilizată pentru MVP.

În comparație cu MVP, accentul se pune în acest caz pe testarea ASIC-ului împreună cu modulul complet de care aparține (presupunând că funcția ASIC-ului este

corectă și că a fost testată în etapele anterioare de validare, unde s-a insistat pe funcțiile digitale și analogice ale ASIC-ului).

Neajunsul acestei platforme de validare este faptul că în cazul în care se găsește o problemă funcțională, e necesară o depanare extinsă deoarece nu mai există acces la interfața dintre blocurile analogice și cel digital al ASIC-ului. Pentru diagnosticarea problemei se pot folosi numai date parțiale și procesate care pot fi citite prin intermediul protocolului de comunicație implementat de către blocul digital.

3.1.3 Platforma de validare în stil ECU

A treia etapă de validare este efectuată utilizând produsul final, ECU-ul, așa cum se arată în Fig. 3.3. În această etapă, scopul este de a verifica modul în care diferite module sunt integrate în produsul final, presupunând că toate modulele funcționează individual în mod corect. Această placă de validare este optimizată din punct de vedere al aspectului și dimensiunilor PCB-ului pentru a fi cât mai aproape de ECU, dar și din punct de vedere al costurilor componentelor electronice. Produsul final este încapsulat folosind o carcasă din aluminiu, turnată sub presiune, având o performanță termică bună pentru a disipa căldura întregului circuit electronic datorită puterilor disipate.

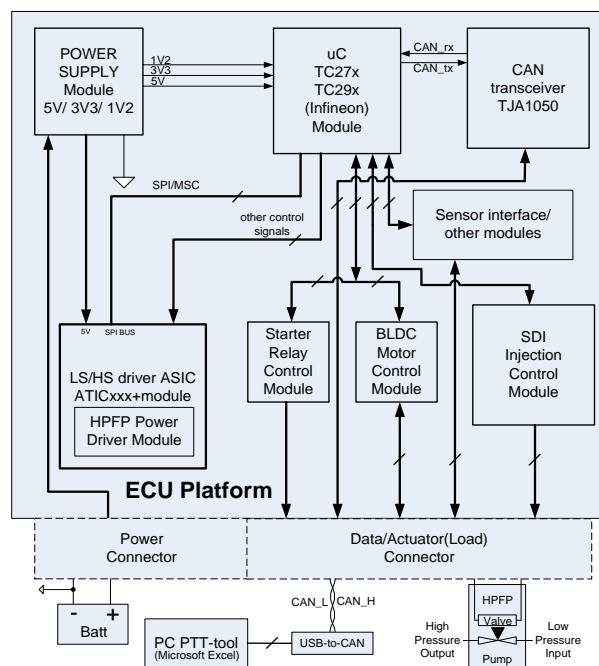


Fig. 3.3 Platforma de validare în stil ECU [34] (original).

Avantajul acestei platforme de validare față de celelalte platforme prezentate este testarea performanței produsului final, având cea mai bună amplasare a componentelor cu efecte parazite minime datorită eliminării totale a conectorilor intermodulari, păstrând doar conectorul necesar mănunchiului de cabluri conectate la sarcinile ce vor fi controlate. După ce validarea a trecut cu succes, rămâne doar o singură etapă, calificarea în automobil a produsului.

Arhitecturile de validare prezentate mai sus necesită un efort semnificativ de proiectare și, mai presus de toate, trebuie alocat un buget considerabil pentru aceste activități. Platformele de validare hardware prezentate mai sus sunt ineficiente din punctul de vedere al modificărilor în cazul în care parametrii de proiectare au fost ignorați, deci necesita cel puțin o reproiectare parțială, prin urmare timp suplimentar și bani investiți - care ar putea influența timpul maxim disponibil pentru procesul de dezvoltare.

În Fig. 3.4 este o prezentare generală a procesului de validare cu privire la scopul principal al fiecărei etape de validare [26, 29].

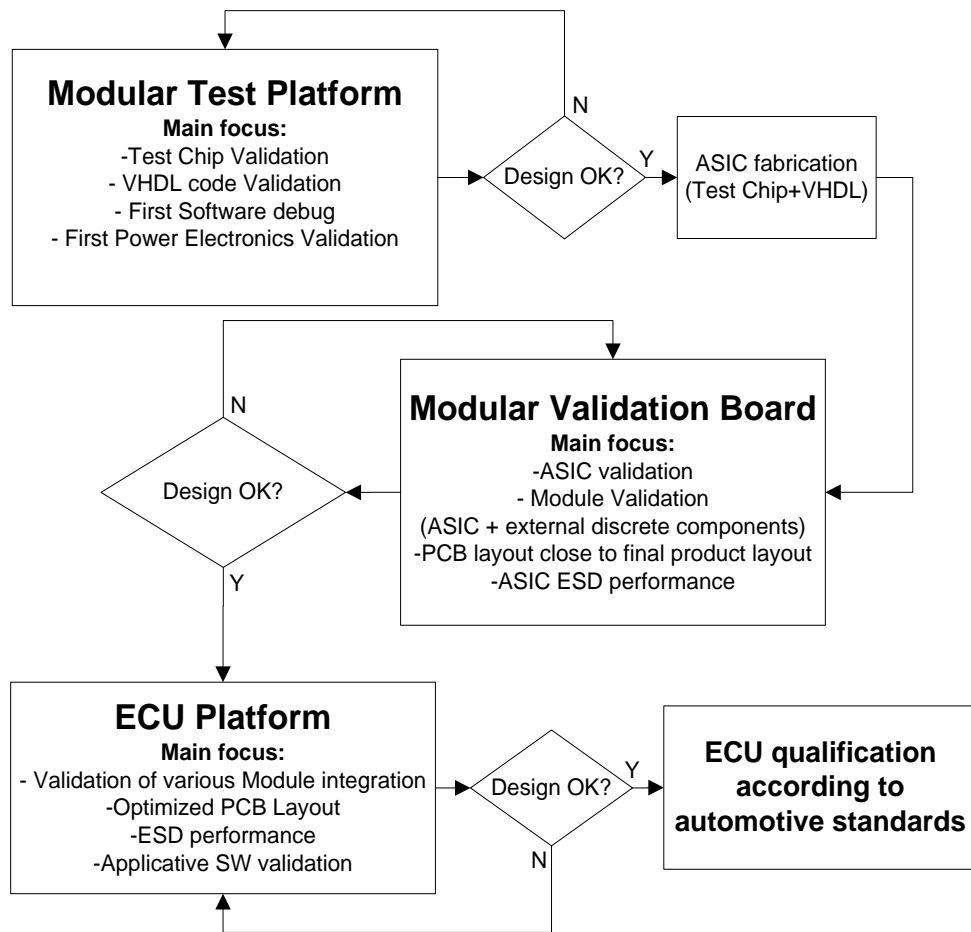


Fig. 3.4 Schema bloc a procesului de validare [34] (original).

3.2 Arhitecturi de simulare: existente și cea propusă

Simularea este o alternativă la validarea HW (Hardware). Cu toate acestea, validarea HW a produsului final (platforma ECU) nu poate fi înlocuită de simulare datorită standardelor stricte de siguranță.

Chiar dacă substituirea prin simulare a MTP și MVB ar putea fi riscantă, deoarece erorile vor fi descoperite doar în faza târzie a dezvoltării ECU-lui, în timpul validării, procesul de simulare aduce un beneficiu semnificativ datorită flexibilității sale, modificarea platformei de simulare fiind mult mai simplă decât modificarea platformelor de validare, necesitând doar intervenție în programul de simulare.

Pentru a reduce riscul de bug-uri de proiectare nedescoperite, toate modelele utilizate în simulare trebuie să fie fiabile, prin urmare, comportamentul digital al ASIC ar putea fi ușor reprodus prin importarea aceluiași cod VHDL utilizat în procesul de dezvoltare a ASIC-ului. Reutilizarea codului VHDL, incluzând software-ul care rulează pe uC, modelele analogice ale componentelor discrete aparținând modulului din care face parte ASIC-ul și în final, modelele electromecanice ale sarcinilor care trebuie controlate ar putea aduce un beneficiu semnificativ în procesul de validare deoarece comportamentul sistemului ar putea fi simulat înainte ca ASIC să fie fabricat.

Pentru simularea la nivel de sistem, în conformitate cu cele mai bune cunoștințe, există o opțiune principală în practică, arhitectura de co-simulare [30, 31], așa cum este prezentată în Fig. 3.5. În continuare, față de arhitectura de co-simulare, propun două arhitecturi alternative de simulare la nivel de sistem.

3.2.1 Arhitectura de co-simulare

Pentru simularea cu arhitectura de co-simulare a unui ECU și a sistemului de care aparține, sunt necesare multiple simulatoare, dar și un co-simulator care să coordoneze celelalte simulatoare, așa cum se arată în Fig. 3.5.

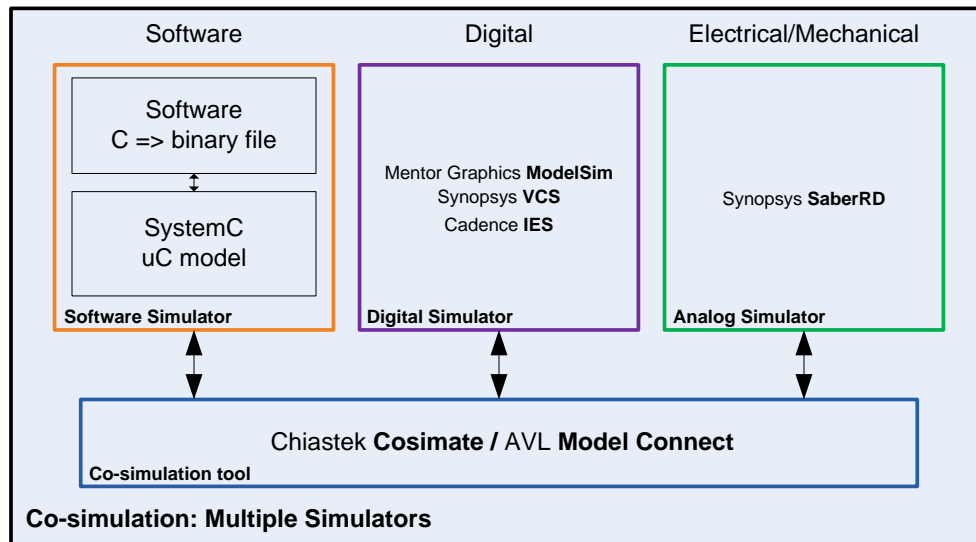


Fig. 3.5 Arhitectura de co-simulare [34] (original).

Software-ul (fișier binar) care rulează pe microcontroler și modelul SystemC al microcontrolerului este gestionat de un simulator, așa-numitul Simulator de Software. Codul VHDL al ASIC-ului rulează pe un simulator digital, blocul electric (blocul analogic al ASIC, modulul de care aparține ...) și modelele de sarcină (HPFP) rulează într-un simulator analogic.

Sincronizarea dintre simulatoare de diferite domenii este gestionată de către Co-Simulator, care este responsabil pentru toate schimburile de date în timpul simulărilor.

Avantajele acestei arhitecturi sunt:

- presupunând că simulatoarele din diferite domenii funcționează corect, pentru problemele de co-simulare este necesară contactarea echipei suport a dezvoltatorului Co-simulatorului
- cele mai bune instrumente de simulare din diferite domenii pot fi combinate dacă co-simulatorul le suportă [31]
- utilizatorul poate defini cu ușurință cât de des să se producă schimbul de date între simulatoare și Co-simulator:
 - simulare condusă de eveniment (schimb de date optimizat => simularea este accelerată) [32]
 - simulare cu pas de timp fix (mai multe detalii sunt disponibile, dar timpul de simulare total va crește)

Există trei dezavantaje majore ale Co-simulării:

- necesitatea licențelor individuale pentru fiecare simulator (cost ridicat)
- dificultăți în realizarea unui timp de simulare uniform și comun între diferitele simulatoare, adesea erorile apar în legătură cu discontinuitatea în rezolvarea ecuațiilor în special din domeniul analogic datorită timpului de simulare ne comun între simulatoare
- inginerii care rulează simularea sub-sistemului au nevoie de instruire specializată cu privire la modul de utilizare a fiecărui simulator de domeniu diferit

Pentru a depăși neajunsurile Co-simulării, propunem două arhitecturi pentru simularea sub-sistemelor, prima, așa cum se arată în Fig. 3.6, folosind două simulatoare diferite, iar a doua în Fig. 3.7, utilizând doar un singur simulator.

3.2.2 Arhitectura cu interfață directă Virtualizer® - SaberRD®

În Fig. 3.6 este introdusă o interfață directă între Synopsys Virtualizer® și Synopsys SaberRD®.

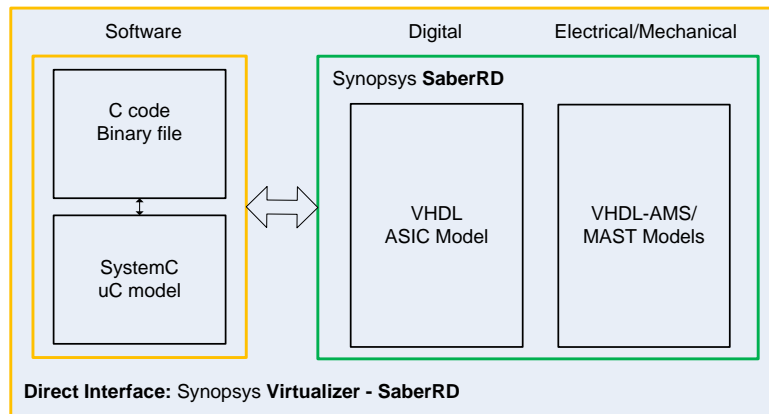


Fig. 3.6 Simularea la nivel de sistem prin interfață directă [34] (original).

Virtualizer® poate să compileze și să execute modelul SystemC al microcontrolerului împreună cu fișierul binar (software-ul care rulează pe uC).

SaberRD® este un simulator multi-domeniu capabil să simuleze împreună:

- blocul analogic al ASIC-ului
- blocul digital al ASIC-ului
- modulul de care aparține ASIC-ului (componentele electronice care nu pot fi integrate în ASIC)

- model electro-mecanic al pompei HPFP.

Avantajele arhitecturii cu interfață directă sunt:

- doar două simulatoare sunt implicate în simularea sistemului; prin urmare sunt necesare doar două licențe

- transferul de date între simulatoare este manipulat printr-o interfață directă datorită aceluiași dezvoltator al programelor de simulare; prin urmare se elimină discrepanțele de timp de simulare la sincronizarea schimbului de date dintre cele două instrumente de simulare.

3.2.3 Arhitectura propusă; SaberRD®

A doua propunere de simulare la nivel de sistem are arhitectura „All-In-One” folosind doar SaberRD®, după cum se arată în Fig. 3.7.

Următoarele capitole vor prezenta implementarea unei platforme virtuale a arhitecturii HW din Fig. 3.1, bazată pe metodologia de simulare a subsistemului „All-In-One”.

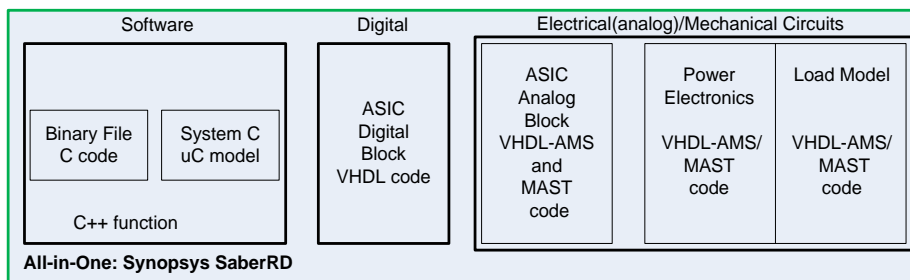


Fig. 3.7 Arhitectura de simulare propusă [34] (original).

Această metodologie prezintă o arhitectură de simulare promițătoare, bazată pe experiența mea anterioară de utilizare a SaberRD® ca simulator pentru codul VHDL. În plus, acesta este un simulator electric specializat pentru circuite electronice, având incluse biblioteci conținând o gamă largă de modele de componente electronice. Modelele electronice (analogice) sunt construite folosind MAST (limbajul de descriere a hardware-ului proprietar Synopsys) sau VHDL-AMS. Totodată permite și translatarea automată a modelelor Spice în MAST pentru a fi utilizate cu succes modele de componente electronice puse la dispoziție de către producătorul lor.

Rularea unui model SystemC al uC-ului împreună cu fișierul binar (software compilat) în SaberRD® este o metodă complet nouă. Simulatorul este capabil să apeleze o funcție C++; pentru aceasta este necesar să se creeze în prealabil funcția C++ și să fie stocată în folder-ul de simulare sub forma unui fișier ".dll". Scopul meu este de a implementa modelul uC împreună cu software-ul corespunzător care rulează pe acesta într-o funcție C++ care are o listă de parametri transferați din simulare, după care funcția este executată și la final rezultatele sunt returnate către simulare. Rezultatele simulării subsistemelor [33] sunt comparate cu circuitul real utilizând MTP propus în Fig. 3.1.

Scopul final al comparației este verificarea acurateții și eficienței simulării sistemului în comparație cu mediul software și hardware care conține și sarcina (HPFP) care va fi utilizată.

O altă provocare este aceea de a modela sarcina electro-mecanică în ceea ce privește parametrii esențiali și relevanți din sistemul hidraulic, deoarece ei pot dezvălui schimbări cantitative a inductanței supapei de scurgere a pompei; de exemplu modificările presiunii din rampa comună pot duce la variația inductanței valvei sau a timpului de închidere.

3.3 Concluzii și contribuții

Prima parte a capitolului 3 este o prezentare generală a procedurilor de testare/validare și a platformelor unei Unități de Control a Motoarelor, pornind de la dezvoltarea internă a unui ASIC până la calificarea ECU-ului conform standardelor automobilelor.

Cea de-a doua parte a capitolului este o prezentare a celor mai des utilizate arhitecturi de simulare a sistemelor cu ASIC.

Capitolul 5 va trata implementarea platformei de simulare și a platformei de emulare a ASIC-ului urmând să se compare rezultatele celor două platforme pentru aceleași cazuri de funcționare.

3.3.1 Concluzii

1. Este esențială testarea în faze timpurii de proiectare a ASIC-ului deoarece eventualele greșeli de concept sau proiectare depistate devreme se pot corecta înainte de producerea fizică a ASIC-ului.
2. Toate erorile/greșelile descoperite după ce ASIC-ul a fost produs, vin cu costuri mari de re-proiectare și re-fabricare.
3. Platformele de emulare a ASIC-ului sunt o modalitate ieftină, rapidă dar și eficientă pentru testarea conceptuală a ASIC-ului în faze timpurii ale proiectării.
4. Platformele de validare a sistemului cu ASIC sunt o modalitate de testare și mai timpurie a conceptului, chiar și atunci când platforma de emulare nu este încă fizic disponibilă, dar totuși se pot neglija unele elemente parazite (inductanța firelor, capacități parazite, etc.) în comparație cu platforma de emulare a ASIC-ului.

3.3.2 Contribuții

1. Propun pe baza lucrării proprii [34] platforma de validare MTP care poate îmbunătăți performanța de validare datorită utilizării directe a TC-ului pentru a emula blocul analogic al ASIC-ului. MTP poate fi reutilizat cu ușurință pentru noile proiecte datorită structurii sale modulare.
2. Propun tot pe baza lucrării proprii [34] două arhitecturi alternative de simulare a sistemelor fără Co-simulare cu număr redus de simulatoare utilizând un simulator multi-domeniu pentru a realiza simularea de arhitectură a ASIC-ului atunci când încă nu este disponibilă platforma MTP de emulare a ASIC-ului pentru a realiza teste și a valida funcționalitatea conceptuală corectă în faze timpurii ale dezvoltării ASIC-ului.

4. Controlul pompelor de injecție

Capitolul 4 prezintă informații din domeniul sistemelor de injecție directă, în special pentru cele pe benzină. Acest capitol este necesar pentru a face legătura între Capitolul 3 și Capitolele 5 și 6 pentru a explica detaliile specifice de funcționare ale sistemelor de injecție de benzină cu detalii relativ la funcționarea mecanică și controlul electric al valvei pompei de benzină de înaltă presiune.

Pompa de injecție sau pompa de înaltă presiune este componenta electro-mecanică care dă combustibilului presiunea ridicată de care are nevoie pentru a pătrunde cu viteză mare în camera de ardere a motorului termic și a facilita atomizarea amestecului aer-combustibil.

4.1 Detalii generale despre sistemul cu injecție directă a benzinei

În domeniul automobilelor există două tipuri de motoare cu combustie internă: benzină și motorină. Pentru vehiculele de pasageri cel mai frecvent utilizat este motorul pe benzină din cauza emisiilor mai mici de gaze de eșapament și control mai bun al emisiilor pentru a atinge normele de poluare.

Motoarele pe benzină moderne sunt clasificate în două categorii principale: motoare mai ieftine și cu performanțe reduse cu sistem SPTI (Single Point Injection) sau MPI (Multi Point Injection) și cel GDI (Gasoline Direct Injection), sistem de injecție directă mai scump, cu un control mai bun al consumului de combustibil, oferind motoare mai puternice la aceeași capacitate cilindrică.

Sistemele SPTI și MPI injectează combustibilul în galeria de admisie, pentru SPTI în zona comună a galeriei de admisie, iar pentru MPI în fiecare ramificație individuală a galeriei de admisie, chiar în amonte de supapa de admisie. În aceste cazuri, combustibilul se amestecă cu aerul în afara cilindrului. Presiunea combustibilului injectată este de maxim 4 bari, iar injectoarele folosite sunt actuatori cu solenoid cu timp de răspuns lent.

Sistemul GDI este descris de Sarwar, Sankavaram și Lu [35], cu un accent deosebit pe pompa de combustibil de înaltă presiune (HPFP).

Sistemul GDI utilizează injectoare de înaltă performanță pentru a injecta combustibil direct în camera de ardere, cu o presiune a combustibilului de până la 500Bar. Datorită injecției directe, capul pistonului este răcit de combustibil, permițând rapoarte mai mari de compresie și o aprindere prin scânteie mai devreme producând putere mai mare.

Injectoarele GDI pot fi acționate prin solenoid sau piezo. Injecțiile multiple sunt posibile cu ambele, în timpul unui ciclu motor, în fiecare cilindru individual. Pentru a permite mai multe injecții pe ciclu motor, injectoarele trebuie să aibă un răspuns rapid, în special la închidere, deoarece dozarea combustibilului depinde de o

procedură de comandă bazată exclusiv pe timp, astfel încât controlul electronic al injectoarelor este important și trebuie să fie eficient și precis.

Injectoarele piezoelectrice și cele cu solenoid sunt alimentate de tensiuni mai mari decât tensiunea de 12V a sistemelor auto. Surse de tensiune în comutație de tip ridicător (boost) sunt utilizate pentru a crește nivelul de tensiune la 65V pentru injectoarele solenoidale și la 400V pentru injectoarele piezo, permițând astfel o deschidere rapidă și o închidere rapidă prin limitarea valorii maxime (clamping) a tensiunii electromotoare induse a injectorului solenoid la 65V.

4.2 Bazele sistemului GDI

4.2.1 Componentele mecanice și hidraulice

În Fig. 4.1 se prezintă componentele de bază ale unui sistem GDI. Benzina din rezervorul de combustibil este pompată de o pompă de combustibil de joasă presiune, LPFP (Low Pressure Fuel Pump), prin conducte cu o presiune maximă de 4 bari către pompa de combustibil de înaltă presiune, HPFP, care este activată de arborele cu came al motorului generând o presiune de până la 500Bar. Combustibilul cu înaltă presiune este acumulat în rampa comună (tampon de combustibil), iar injectoarele sunt alimentate din rampa comună.

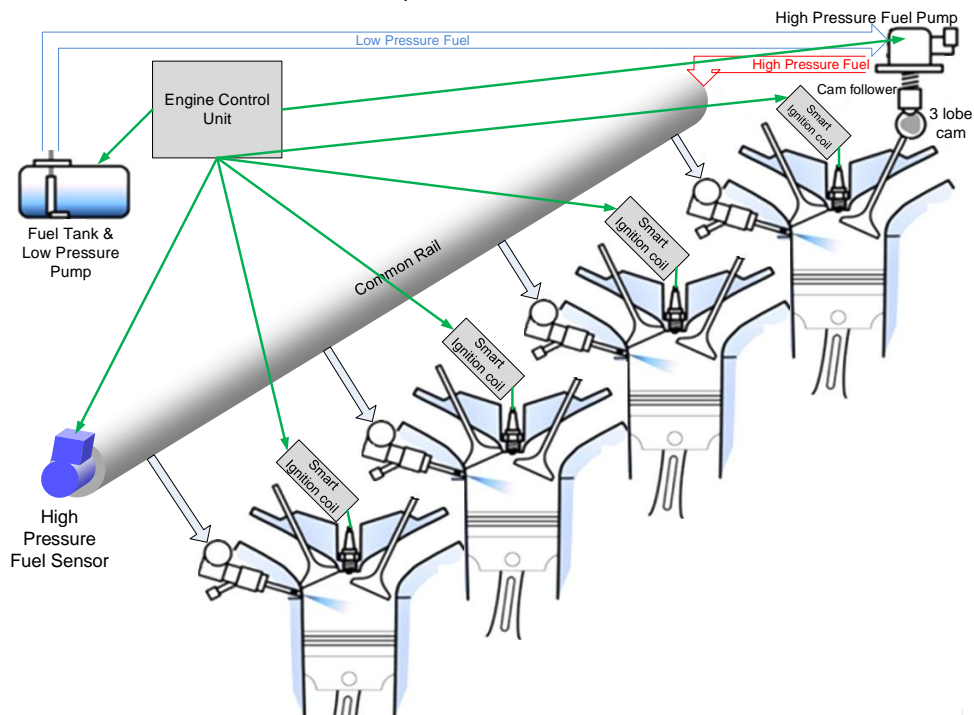


Fig. 4.1 Arhitectura sistemului GDI din lucrarea mea [38], bazată pe [36] (original).

Pentru a avea o injecție precisă a combustibilului, presiunea în sistemul cu rampă comună trebuie să fie în mod ideal constantă pentru a avea întotdeauna aceeași cantitate de combustibil pulverizat în cilindru de fiecare injector având timpul t_{ON} de activare (injecție).

În Fig. 4.2 este prezentată schema simplificată a unei pompe HPFP constând în pompa în sine și valva DIV (Digital Inlet Valve) de control a cantității de combustibil.

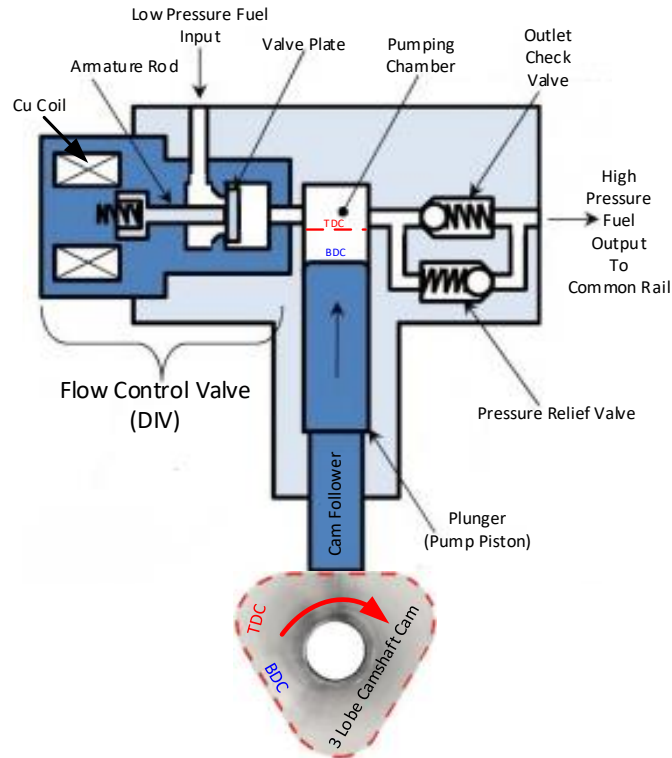


Fig. 4.2 Detaliu HPFP bazat pe [36] (original).

Pentru a menține o presiune constantă în sistemul cu rampă comună, reglarea presiunii pornește de la controlul supapei electro-mecanice atașate pompei HPFP. Pentru a controla cantitatea de combustibil de joasă presiune care intră în camera de presiune a HPFP, valva electro-mecanică (DIV), normal deschisă, este activată după o durată Δt_1 față de punctul mort superior (TDC – Top Dead Center) al poziției pistonului pompei. LPFP pompează combustibilul în cilindru de compresie a HPFP, iar apoi algoritmul de control al presiunii implementat în microcontrolerul unității de control al motorului (ECU) calculează cantitatea de combustibil necesară în funcție de sarcina și viteza motorului.

ECU-ul comandă închiderea supapei electro-mecanice DIV prin intermediul unui profil de curent „peak and hold”, permițând astfel ca rotirea lobilor arborelui cu came care acționează pistonul pompei HPFP să comprime combustibilul din camera de presiune. Dacă valva de scurgere DIV nu este închisă (profil de curent activ), combustibilul va fi pompat înapoi în sistemul de joasă presiune.

La ieșirea HPFP-ului există o supapă mecanică de reținere preîncărcată de un arc.

Atunci când presiunea din camera pompei HPFP este mai mare decât presiunea din rampa comună, combustibilul este transferat în rampa comună.

Algoritmul de control al presiunii combustibilului utilizează informațiile de la un senzor de presiune montat pe rampa comună.

Pe baza vitezei și încărcării motorului, ECU comandă injectoarele individual pentru fiecare cilindru; când se detectează poziția corectă a pistonului motorului termic, bobina de inducție generează o scânteie prin bujie în camera de combustie, rezultând o explozie a amestecului de combustibil.

În Fig. 4.3 este prezentată pompa de presiune înaltă Continental, capabilă să producă o presiune de până la 350Bar în rampa comună (benzină).

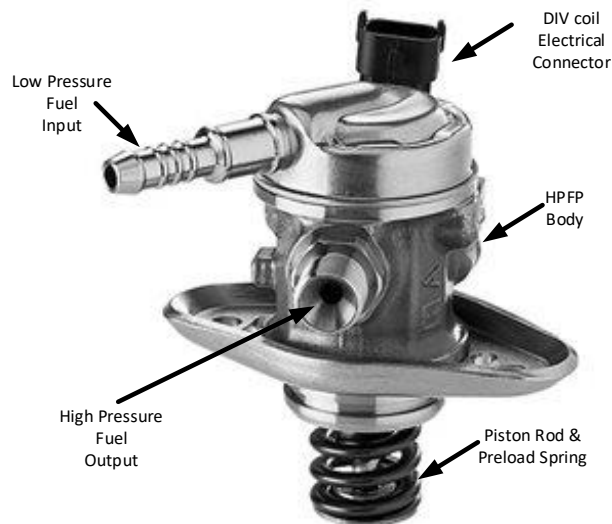


Fig. 4.3 Pompa de presiune înaltă Continental-350Bar [37].

4.2.2 Circuitul electric de control al valvei DIV

În Fig. 4.4 se prezintă o diagramă bloc de bază a circuitului de comandă electrică pentru valva electro-mecanică DIV a pompei HPFP, bazată pe propria lucrare [38].

Începând din partea dreaptă, DIV este conectat între două tranzistoare de putere, un LSD care acționează ca un comutator de siguranță și un HSD ca element activ de comutare pentru reglarea curentului.

Pompa HPFP este un element relevant pentru siguranța motorului. Dacă apare un scurtcircuit la potențialul de referință a mașinii (șasiu) sau la potențialul pozitiv al bateriei la una dintre conexiunile valvei DIV prin cablaj, sistemul trebuie să poată opri fluxul de curent prin DIV, prin urmare este necesară topologia de semi-punte.

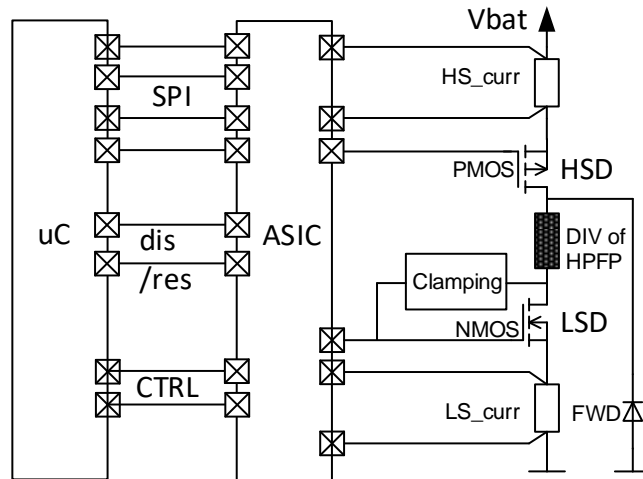


Fig. 4.4 Schema bloc a circuitului electric de comandă [38] (original).

Atât LSD-ul, cât și HSD-ul sunt protejate împotriva supracurentului prin intermediul monitorizării de curent: circuitul integrat specific aplicației (ASIC) monitorizează în mod constant căderea de tensiune a rezistențelor traductoare de curent a tranzistoarelor LSD și HSD.

Dacă se detectează o cădere de tensiune prea mare pe rezistențele traductoare de curent, ASIC-ul oprește LSD/HSD și apoi raportează diagnosticul prin intermediul protocolului de comunicare SPI (Serial Peripheral Interface). Limita de curent la care să se detecteze supracurentul poate fi selectată prin comenzi SPI.

Datorită caracteristicilor electrice a supapei DIV, sarcina echivalentă este un rezistor ($R \approx 1-2\Omega$) în serie cu un inductor ($L = 1,5-2\text{mH}$), fiind deci necesar un mijloc de recirculare a energiei.

După cum s-a menționat anterior, HSD este elementul activ de comutare, astfel încât o diodă de „freewheel” (FWD - Free Wheel Diode) este utilizată pentru a diminua tensiunea electromotoare indusă a DIV (LSD-ul este întotdeauna activ în timp ce profilul de curent este activ). FWD-ul asigură o scădere a curentului lent la fiecare ciclu de comutare, reducând astfel emisiile radiate (în termeni de compatibilitate electromagnetică) pe care altfel le-ar provoca tranzistoarele prin comutarea lor rapidă.

La oprirea LSD-ului (oprit numai la sfârșitul profilului de curent pentru deschiderea supapei) deoarece energia magnetică înmagazinată datorită profilului de curent aplicat trebuie disipată rapid, soluția cu dioda FWD nu este adecvată datorită timpului lent de scădere a curentului. Soluția utilizată este de a implementa o sursă de tensiune cu tranzistorul LSD astfel încât el să funcționeze în regim activ modificându-și rezistența internă pentru a păstra tensiunea de drenă prestabilită de o rețea externă drenă-sursă cu o diodă Zenner, așa numitul circuit de „active clamping”.

„Active clamping” este un circuit cu reacție care forțează o tensiune în poarta LSD-ului conducând astfel MOSFET-ul în regiunea de saturație, până când energia magnetică înmagazinată în bobina valvei DIV este complet disipată.

4.2.3 Principiul de control a pompei de injecție HPFP

Pompa de combustibil de înaltă presiune face parte din sistemul de alimentare cu combustibil, acționat mecanic (de către arborele cu came al motorului) pentru comprimarea combustibilului. Cantitatea de combustibil care trebuie comprimată este controlată de o supapă electromagnetă (deschisă = nu circulă curent prin ea), numită și valvă de intrare digitală (DIV), deoarece are două stări: Deschis și Închis. DIV-ul este controlat electric de către ECU prin intermediul unui profil de curent controlat în buclă închisă, CLCC (Closed Loop Current Control) sincron cu TDC a pistonului pompei, ca în Fig. 4.5.

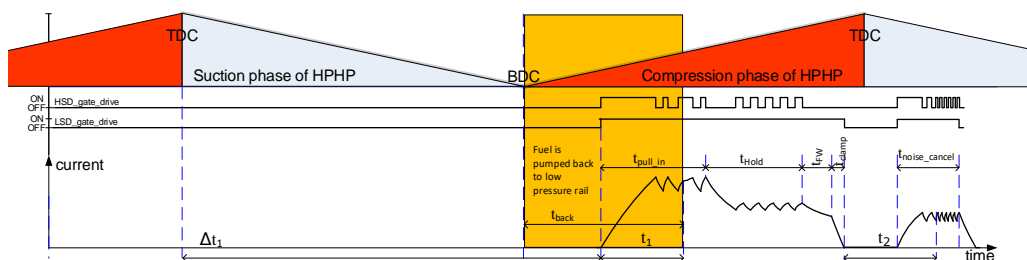


Fig. 4.5 Exemplu de control al HPFP [38] (original).

Durata Δt_1 pentru activarea DIV depinde de:

- Profilul lobilor arborelui cu came
- Turația motorului
- Cantitatea de combustibil necesară (sarcina motorului)

Profilul CLCC poate fi împărțit în trei secțiuni:

1. Faza de „Vârf”: se creează un câmp magnetic pentru a închide DIV prin mișcarea armăturii, așa-numitul timp de „pull-in”:

$$t_{pull_in} = t_1 + t_s \quad (1)$$

unde t_1 este timpul de închidere al DIV și t_s este un timp de siguranță adăugat pentru a fi siguri că supapa a fost închisă;

➤ t_1 depinde de:

- Viteza de rotație a motorului
 - Tensiunea bateriei (timpul de creștere a curentului de „pull-in”)
 - Temperatura bobinei DIV
 - Temperatura combustibilului.
2. Faza de „reținere”: este utilizată pentru a menține câmpul magnetic pentru a menține supapă închisă până când presiunea din camera pompei este suficient de mare pentru a ține supapă închisă: în această fază este posibilă acumularea de presiune în camera de compresie a HPFP.
 3. Faza de „evacuare”: evacuarea energiei magnetice stocate în miezul magnetic al valvei DIV pentru deschiderea supapei; eliminarea energiei trebuie să fie rapidă pentru a reduce disiparea energiei în bobină și miezul magnetic; pe de altă parte, reduce timpul de deschidere, astfel încât să se evite pornirea unui nou ciclu cu valva închisă, permițând astfel introducerea combustibilului în camera de compresie a HPFP la următorul ciclu de compresie.

4.3 Concluzii și contribuții

Cele prezentate mai sus oferă informații relevante din domeniului sistemelor de injecție directă, în special de benzină. Acest capitol este necesar pentru a face legătura între Capitolul 3 și Capitole 5 și 6 pentru a explica detaliile specifice de funcționare a sistemelor de injecție de benzină oferind detalii la funcționarea mecanică și controlul electric al valvei pompei de benzină de înaltă presiune.

4.3.1 Concluzii

1. Pentru a crește eficiența și puterea motoarelor termice pe benzină, dar în același timp să se țină pasul cu evoluția normelor de poluare, este nevoie de sisteme cu injecție directă și control electronic al motoarelor.
2. Performanțele sistemului de injecție constând din injectoare, pompa de benzină de înaltă presiune, bujii și modulul electronic de control reprezintă elementul cheie pentru a îndeplini cele expuse la punctul anterior.
3. Controlul precis al combustibilului necesită o presiune constantă în rampa comună.
4. Presiunea constantă din rampa comună este direct influențată de performanța pompei de benzină și a controlului său.
5. Controlul electro-valvei pompei de înaltă presiune trebuie să fie robust (ASIC-ul trebuie să ofere către uC diagnoza circuitului de putere și în același timp să ia decizii independente de uC pentru protecția locală a circuitului de putere) și cu un timp de răspuns rapid (uC trebuie să poată modifica parametrii de profil de curent înainte de fiecare puls, iar la final de puls este important să evacueze rapid energia înmagazinată în bobina DIV pentru a scurta cât mai mult timpul de deschidere a valvei scurtând astfel timpul necesar până la aplicarea următorului puls de curent) pentru a facilita dezvoltarea algoritmilor de reglare a presiunii în rampa comună.
6. Pentru a spori confortul auditiv pentru unele pompe de benzină de înaltă presiune este necesar un control electronic sofisticat cu dublu scop, de a menține presiunea constantă în rampa comună, dar și de a aplica un profil de curent optim astfel încât deschiderea și închiderea valvei electro-mecanice a pompei de benzină să producă vibrații (zgomote) cât mai mici.

4.3.2 Contribuții

1. Prezint și explic pe baza lucrării proprii [38] funcționarea sistemelor GDI cu detalii specifice referitoare la acționarea mecanică și comanda electronică a pompei de benzină de înaltă presiune, respectiv a valvei electro-mecanice constituate a acesteia.

5. Rezultate experimentale

În acest capitol analizez implementarea celor două platforme propuse în Capitolul 3, platforma de simulare cu un singur simulator multi-domeniu și platforma MTP de emulare a ASIC-ului. Ambele platforme facilitează testarea în fază timpurie a procesului de dezvoltare a ASIC-ului evitând propagarea eventualelor erori/defecte de concept/arhitectură în producția ASIC-ului. În cele ce urmează voi prezenta schema globală de simulare organizată ierarhic, voi intra în detalii specifice referitoare la fiecare bloc component al ierarhiei. Un subiect important va fi dezvoltarea modelului electric de simulare pentru valva electro-mecanică DIV. Următorul pas va fi explicarea implementării fizice a platformei de emulare a ASIC-ului organizată tot pe blocuri ierarhice. În final voi compara rezultatele simulării cu cele ale implementării fizice.

5.1 Simularea sistemului de control al valvei DIV

Conform practicii uzuale am început prin a simula în simulatorul mixt Saber® (Synopsys), așa cum este prezentat în Fig. 5.1. Scopul meu a fost să împart simularea pe blocuri funcționale, astfel încât să fie posibilă migrarea blocurilor în orice moment într-un proces de dezvoltare a circuitelor integrate mixte. Fiecare bloc funcțional poate fi ulterior extins la o implementare specifică într-un proces tehnologic de fabricare a circuitelor integrate.

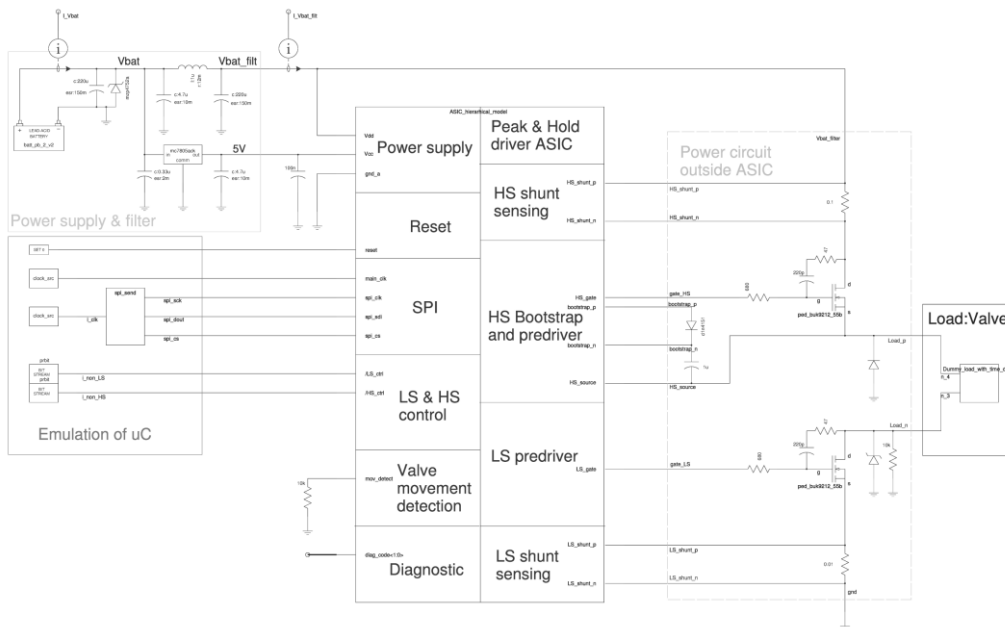


Fig. 5.1 Schema ierarhică de simulare (original).

Simularea este organizată într-o arhitectură ierarhică care constă dintr-o schemă de nivel superior și un simbol principal (simbolul ASIC-ului care efectuează funcția de comandă a supapei DIV).

Schema de nivel superior este împărțită în 5 blocuri:

- Alimentare și filtru
- Emularea uC
- Bloc comportamental ASIC (Peak & Hold driver)
- Componente electronice discrete de putere (care nu fac parte din ASIC)
- Modelul electric a supapei DIV.

De menționat este faptul că în schemele de simulare este același simbol pentru condensatori indiferent că sunt polarizați sau nepolarizați. Pentru a face diferența între tipuri de condensatori, modelul se poate parametriza corespunzător.

5.1.1 Blocul de filtraj și alimentare

Alimentarea cu energie și blocul de filtrare fac parte din schema de nivel superior. În Fig. 5.2 este prezentat blocul de alimentare, constă dintr-un model comportamental al bateriei cu plumb, care poate fi parametrizat, un condensator electrolitic de filtraj și tampon, o diodă supresoare pentru suprimarea pulsului tranzitorii pe bara de alimentare de la baterie, un filtru PI pentru a reduce curentul pulsator (*ripple current*) și un regulator liniar de tensiune de 5V pentru alimentarea circuitelor digitale.

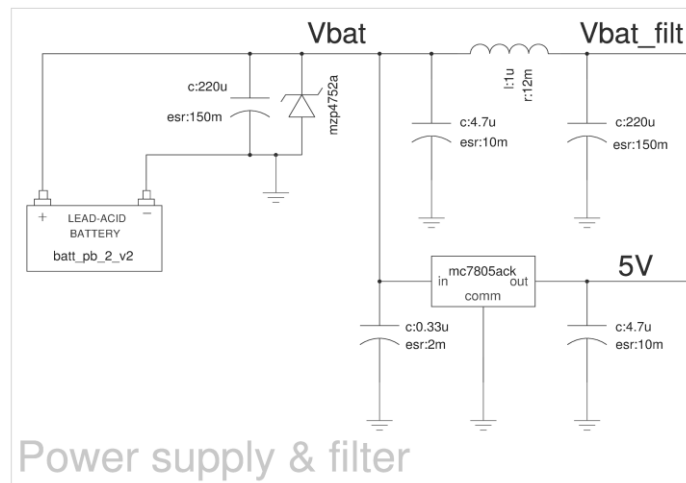


Fig. 5.2 Schema de simulare: blocul de filtraj și alimentare (original).

5.1.2 Emularea blocului uC

Pentru a avea o configurație și o funcție adecvată de simulare, am implementat în simulare un bloc care emulează funcția unui uC, așa cum este prezentat în Fig. 5.3.

Acest bloc conține toate semnalele necesare pentru ca ASIC-ul să fie configurat astfel încât să funcționeze corect. Lista acestor semnale este: reset, tact principal, bloc VHDL pentru comunicarea SPI (fără răspuns de la ASIC), semnale digitale ON/OFF active jos (NON_LS, NON_HS) pentru pornirea și oprirea profilului de curent programat. Toate semnalele anterioare sunt date de intrare pentru ASIC.

Ca răspuns al ASIC-ului către uC, avem un semnal digital "move_detect" activ atunci când valva este în stare închisă și semnalele de diagnoză pe 2 biți care prezintă starea de eroare a electronicii de putere după cum urmează:

- **"01" - supra-curent HS** (scurtcircuit la masă) => acțiune => dezactivarea tranzistoarelor LS și HS
- **"10" - supra-curent LS** (scurtcircuit la baterie) => acțiune => dezactivare a tranzistoarelor LS și HS
- **"11" - ieșire în gol** (sarcina nu este conectată între LS și HS) => nu este necesară nici o acțiune de protecție, trebuie doar semnalat către sistem
- **"00" - fără defecțiuni LS & HS** => funcționare normală

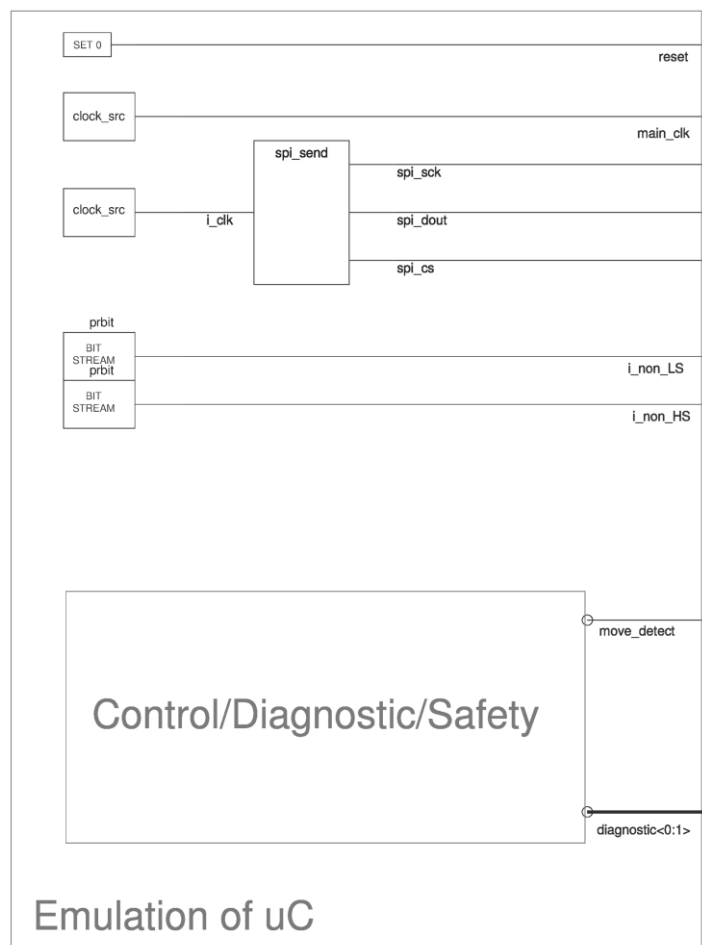


Fig. 5.3 Schema de simulare: blocul de emulare a uC (original).

5.1.3 Circuitul de putere

Închiderea supapei electro-magnetice DIV are nevoie de un curent ridicat și prin urmare, la o turație ridicată a motorului, puterea medie disipată este

semnificativă (1W pentru tranzistorul HS, 0.2W pentru tranzistorul LS, 0.8W pentru dioda TVS - Transient Voltage Suppressor, 0.8W pentru FWD). Dacă însumăm aceste valori, este evident că ar fi extrem de complicat să se integreze toate aceste componente în ASIC; mai mult, va exista o distribuție neuniformă a temperaturii pe suprafața plachetei de siliciu rezultând stres termic și astfel o îmbătrânire prematură a componentei.

De asemenea, o altă problemă tehnologică ar fi combinarea pe aceeași plachetă de siliciu a componentelor de putere cu circuitele de logică digitală necesară pentru a efectua funcția de control.

Acesta este motivul pentru care mi-am propus să folosesc componente discrete pentru aceste elemente de putere, așa cum se arată în Fig. 5.4.

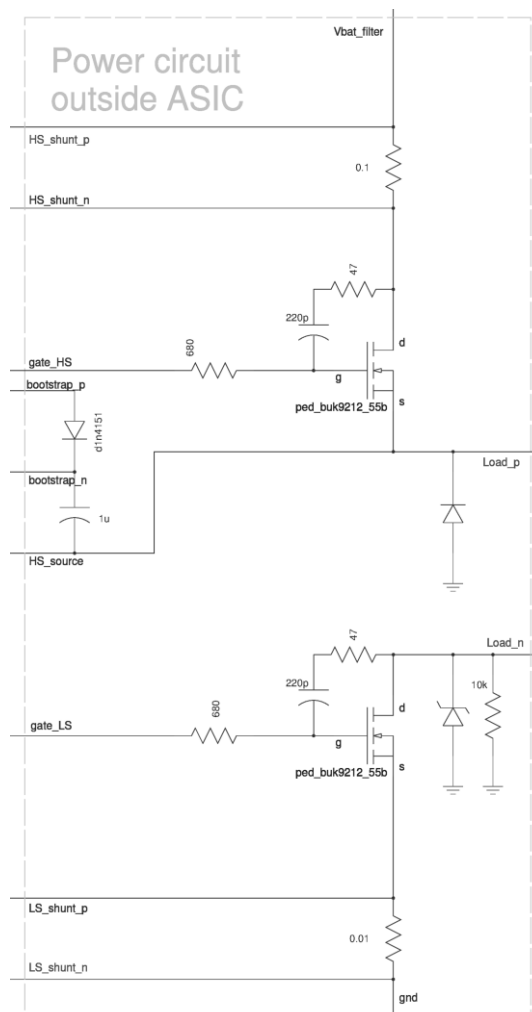


Fig. 5.4 Schema de simulare: circuitul de putere (original).

Fig. 5.4 reprezintă blocul de simulare a componentelor de putere care, în realitate, este construit folosind componente discrete montate pe PCB.

Deoarece pompa HPFP este considerată un element relevant pentru siguranță în sistemul GDI, în cazul în care un scurtcircuit este prezent la oricare dintre terminalele de ieșire a sarcinii, ar putea provoca o activare nedorită a supapei DIV. Pentru a proteja valva DIV de la activarea neintenționată, trebuie implementată o topologie de comandă în semi-punte. De exemplu, în cazul unui scurtcircuit la masă (0V) la terminalul "Load_n", s-ar crea o cale directă de curent (dacă nu ar exista nici un tranzistor HS care să acționeze ca un comutator de siguranță), existând întotdeauna curent prin sarcină independent de starea tranzistorului LS, distrugând astfel bobina supapei DIV. În cazul în care avem tranzistorul HS, este foarte ușor să monitorizăm curentul prin rezistorul de șunt de pe HS și să dezactivăm tranzistorul HS atunci când este depășită o anumită limită de curent.

Pentru a reduce costul circuitului am ales două tranzistoare cu canal N, dar pentru că unul este configurat ca HS, avem nevoie de un circuit bootstrap (montaj autoelevator) pentru a-l activa. Logica de bootstrap și circuitul de putere mică sunt integrate în ASIC, iar dioda de bootstrap și condensatorul sunt componente discrete în afara ASIC-ului.

Terminalul LS acționează ca un element de comutare pentru activarea căii de curent, tranzistorul HS acționează ca element de comutare în timpul profilului de curent cu scop de control de curent.

Circuitul de recirculare a energiei compus dintr-o diodă FWD pe HS care acționează în faza de reglare de curent și dioda TVS pentru limitarea pasivă a tensiunii de drenă a tranzistorului LS pentru evacuare rapidă a energiei din bobina supapei DIV completează acest modul. Evacuarea rapidă a energiei la sfârșitul profilului de curent este necesară pentru a minimiza timpul de deschidere a supapei (lipsa curentului prin bobina valvei DIV = valvă deschisă).

Rezistorul de șunt de pe LS are dublu scop:

- Senzor de curentul pentru reglarea de curentului
- protecție la supra-curent

În final în schemă mai apar un rezistor de grilă și un circuit de încetinire grilă-drenă pentru a menține viteza de creștere a tensiunii de ieșire sub limita de 10V/us pentru a reduce emisiile EMI (Interferențe electromagnetice).

5.1.4 Modelul ierarhic, comportamental al ASIC-ului

Pentru a implementa funcția de control a supapei DIV, este necesar un circuit integrat mixt, în care blocul digital intermediază intrarea și ieșirea blocului analogic, iar blocul analogic este interfața pentru semnalele de intrare și ieșire ale circuitelor de putere exterioare ASIC-ului.

Arhitectura ASIC-ului propus de mine este împărțit în următoarele blocuri, așa cum se arată în Fig. 5.5:

- Sursă de alimentare
- Reset
- SPI
- Control LS & HS
- Detectarea mișcării supapei
- Diagnoză
- Măsurare tensiune pe șuntul HS
- Pre-driver și Bootstrap HS
- Pre-driver LS
- Măsurare tensiune pe șuntul LS

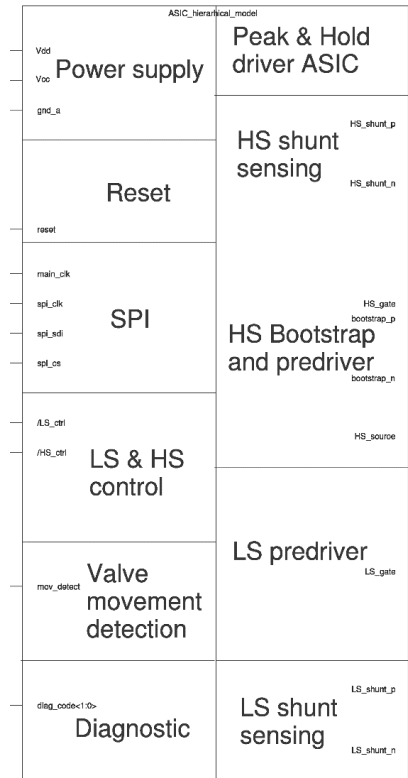


Fig. 5.5 Simbolul de simulare: Modelul comportamental, ierarhic al ASIC-ului (original).

În Fig. 5.6 este prezentată schema din spatele simbolului ierarhic din Fig. 5.5. ASIC-ul este acum împărțit în două blocuri mari:

- Digital
- Analog

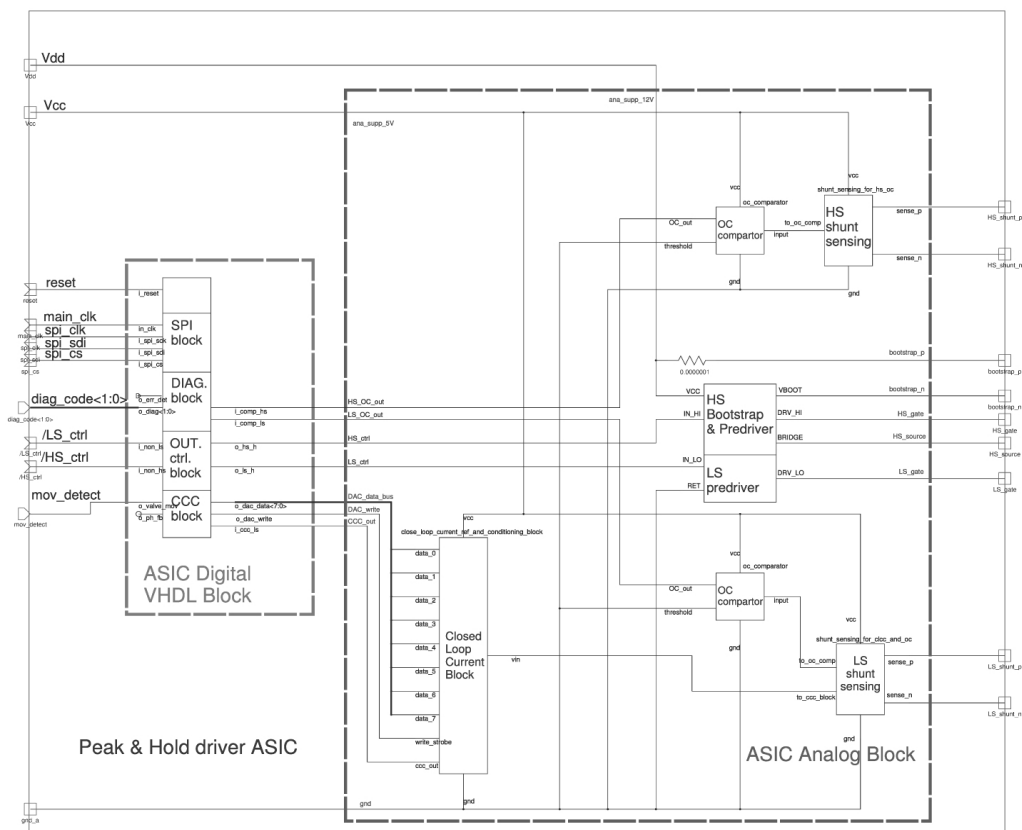


Fig. 5.6 Schema de simulare: ierarhia ASIC-ului (original).

Simulatorul utilizat este capabil să ruleze modele cărora li s-au asociat fișiere VHDL, care definesc funcția digitală de control și comunicare.

Fișierele VHDL sunt, de asemenea, organizate pe niveluri ierarhice și conectate împreună printr-o descriere de nivel superior. Conexiunea componentelor individuale din ierarhie se realizează prin constructul "port map" către porturile din fișierul de nivel superior.

Funcția digitală este împărțită pe două niveluri: nivel superior și componente. În Fig. 5.7 sunt prezentate componentele și instanțierea lor la nivel superior.

În prototipul realizat de mine funcția digitală este implementată de un FPGA folosind aceleași fișiere VHDL ca și modelul de simulare.

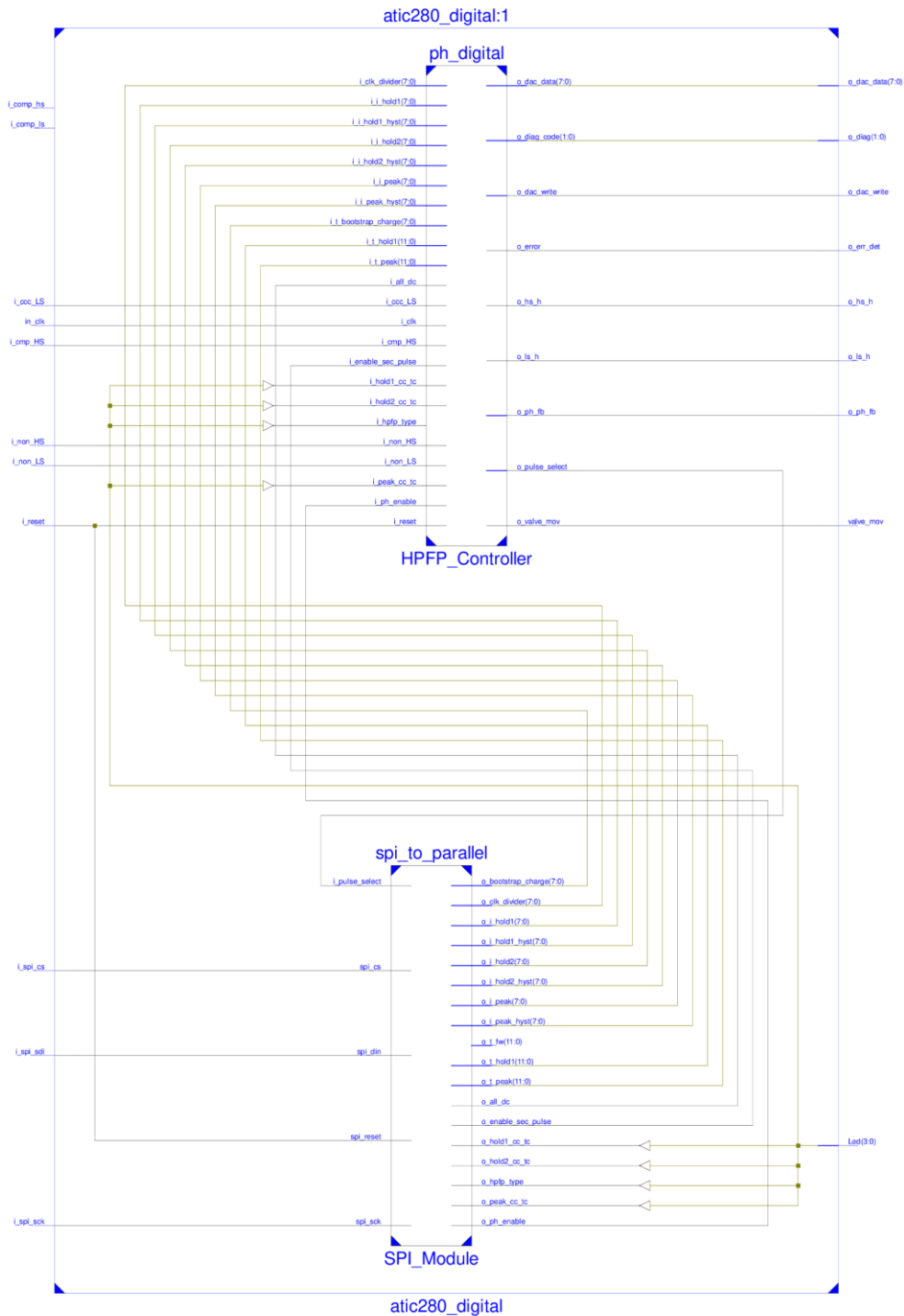


Fig. 5.7 Ierarhia blocului digital al ASIC-ului (original).

Blocul analogic al ASIC-ului este împărțit în mai multe blocuri mai mici având fiecare un simbol în ierarhie și o schemă proprie în spatele fiecărui simbol.

În Fig. 5.8 sunt prezentate simbolurile ierarhice și interconexiunile lor conținând următoarele blocuri:

- Control de Curent în Buclă Închisă (CLCC)
- Pre-driver MOSFET extern pentru LS și HS, incluzând circuit de bootstrap
- comparator supra-curent HS OC (Over Current)
- comparator supra-curent LS OC
- măsurare tensiune șunt HS
- măsurare tensiune șunt LS

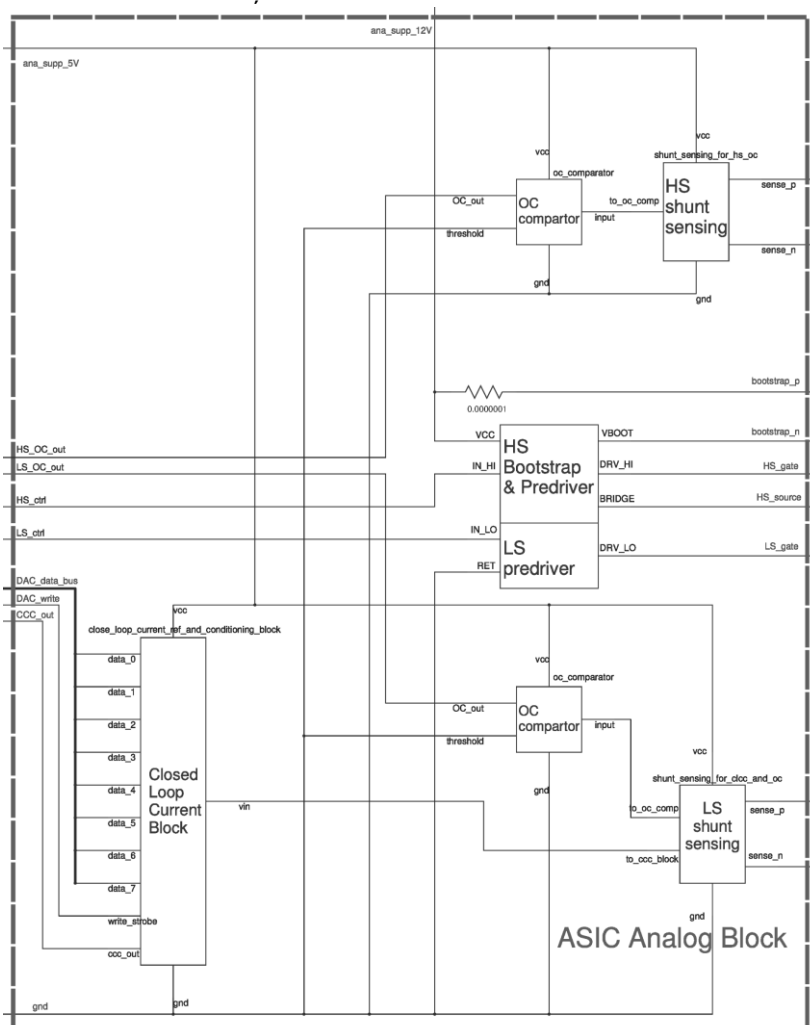


Fig. 5.8 Ierarhia blocului analogic al ASIC-ului (original).

Blocul "LS shunt sensing" este prezentat în Fig. 5.9. Acesta este un model simplificat al circuitului INA180 disponibil la furnizori ca circuit dedicat, amplificator de curent care prezintă un câștig prestabilit de 20; o tensiune de offset de intrare foarte mică care permite o detectare precisă și stabilă a curentului folosită de blocul

de control al curentului în buclă închisă pe întreaga gamă de temperaturi și tensiuni de alimentare.

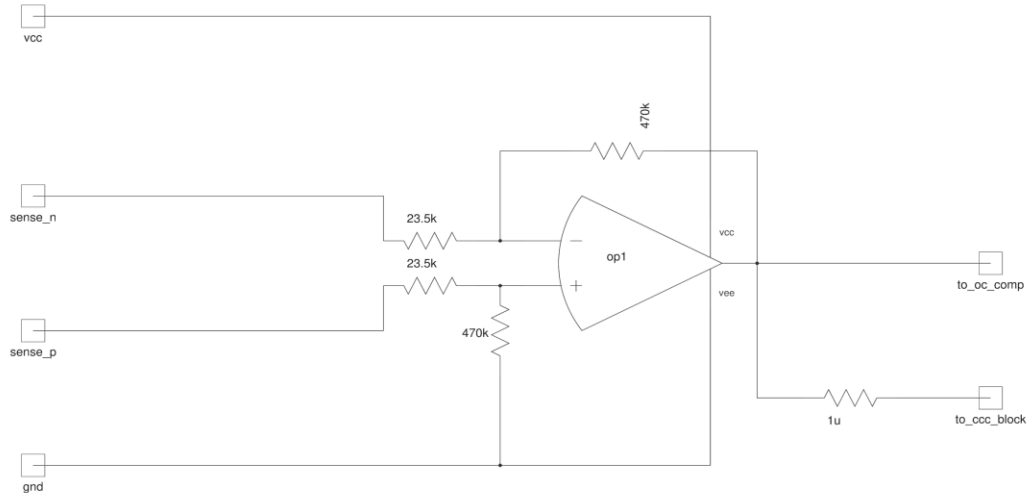


Fig. 5.9 Blocul: măsurare de curent LS (original).

Blocul "Controlul curentului în buclă închisă" conține un DAC, un OPA (Operational Amplifier) și un comparator cu inversare, așa cum se vede în Fig. 5.10.

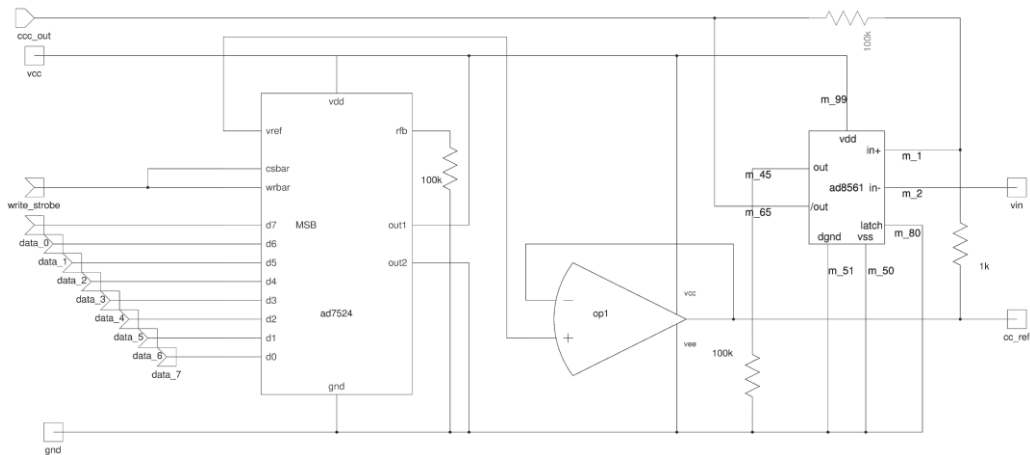


Fig. 5.10 Blocul: control de curent în buclă închisă (original).

DAC-ul primește un set dinamic de date de la blocul de control al curentului implementat în VHDL, își schimbă tensiunea de ieșire în funcție de valoarea de curent care trebuie atinsă în faza de vârf (Peak) sau de menținere (Hold) a profilului de control al curentului prin bobina valvei DIV descrisă în capitolele anterioare.

Amplificatorul neinversor cu câștig unitar acționează ca un tampon (buffer) pentru a avea un timp de răspuns rapid la intrarea neinversoare a comparatorului.

Comparatorul inversor monitorizează tensiunea de pe șuntul de LS amplificată de către blocul "LS shunt sensing" și o compară cu ieșirea DAC-ului controlat de magistrala de date de ieșire a blocului CLCC din VHDL. Ieșirea comparatorului își schimbă starea în consecință, astfel încât această ieșire este conectată la o intrare a

blocului CLCC din VHDL, iar acesta decide dacă trebuie să pornească sau să se oprească tranzistorul extern HS.

Dacă curentul de sarcină (= curentul prin tranzistorul LS) este sub valoarea dorită, tranzistorul HS trebuie pornit; de îndată ce curentul depășește valoarea setată, tranzistorul HS este oprit de către blocul CLCC implementat în VHDL.

Blocul "HS shunt sensing" este un model pentru același dispozitiv, INA180 utilizat pentru LS; aici pe lângă câștigul precis și tensiunea de offset mică, un alt parametru important trebuie implementat în modelul dispozitivului; tensiunea de intrare mare de mod comun și cea diferențială. Pentru aceasta se folosește o rețea de divizoare cu rezistoare pentru a transla tensiunea de mod comun a șuntului de HS în domeniul tensiunii de alimentare a OPA (5V) așa cum este prezentat în Fig. 5.11.

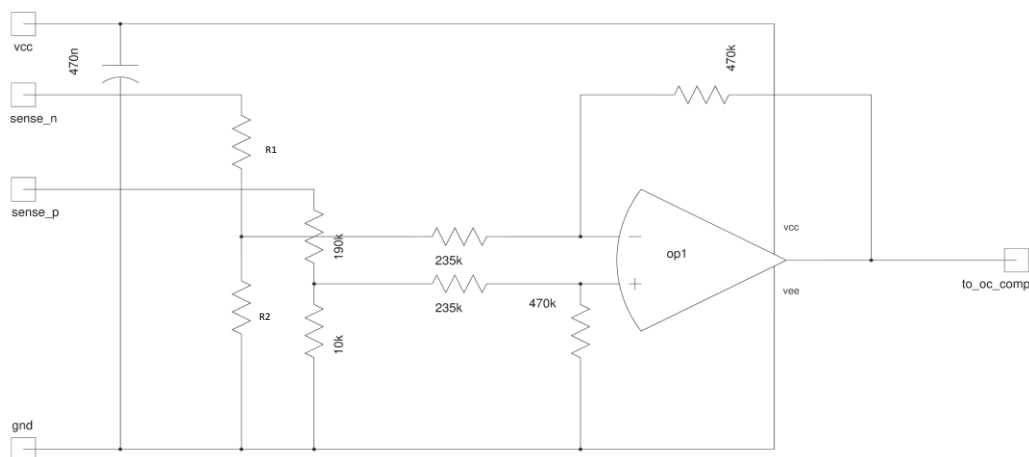


Fig. 5.11 Blocul: măsurare de curent HS (original).

Blocul "comparator OC" este prezentat în Fig. 5.12 și are un rol în funcția de diagnostic, comparând căderea de tensiune a șunturilor LS și HS cu tensiunea generată de DAC-ul utilizat pentru controlul de curent în buclă închisă.

În prototipul nostru pentru emularea acestui bloc se utilizează un comparator AD8561 cu tensiune de offset mică și timp de propagare redus, cu o histereză adițională introdusă extern.

Modelul utilizat în simulare este dezvoltat de către producătorul comparatorului și a fost translatat de mine din Pspice în MAST cu ajutorul utilitarului Model Architect al simulatorului Saber.

În mod normal, pentru controlul curentului în buclă închisă și protecția de supracurent (OC), ar fi necesare două DAC-uri individuale, unul pentru generarea pragului de curent necesar pentru comparatorul din blocul CLCC și altul pentru pragurile de la blocurile de comparator de curent ale tranzistoarelor HS și LS.

Propunerea mea pentru a reduce prețul circuitului este de a utiliza o metodă de detecție de OC bazată pe constanta de timp a circuitului, fiind necesar doar un DAC conectat la comparatoarele de OC, DAC-ul generând același profil de tensiune necesar pentru CLCC. În locul celui de al doilea DAC sunt necesare două numărătoare implementate în FPGA (ASIC digital în implementarea finală).

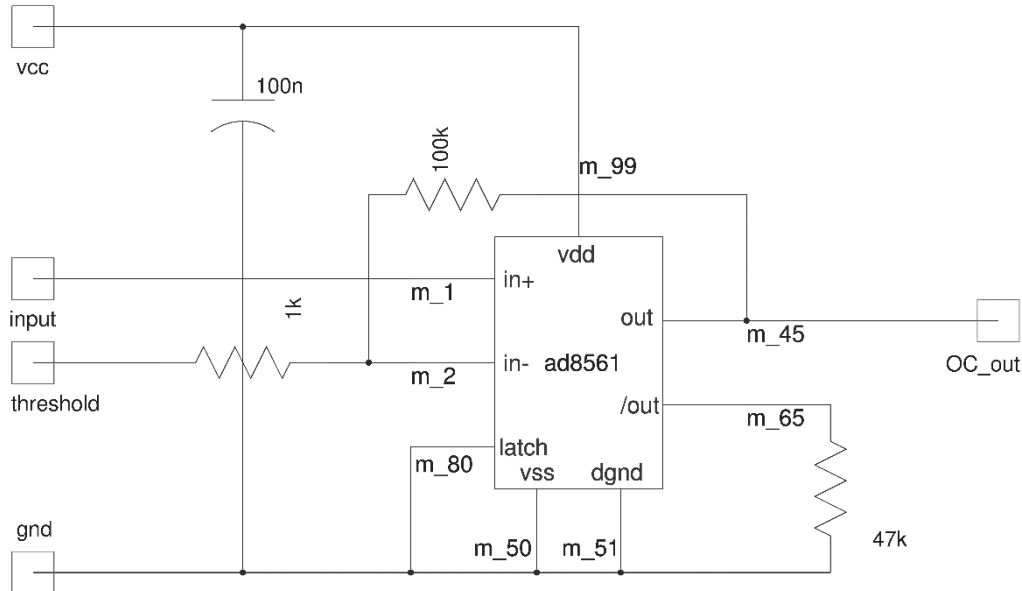


Fig. 5.12 Blocul: comparator de supracurent (original).

Pentru o implementare reușită a detecției de OC bazat pe constanta de timp a circuitului avem următoarele ipoteze:

1. Când se utilizează CLCC, se utilizează întotdeauna o sarcină inductivă
2. Trebuie să existe un parametru programabil pentru diferite valori de inductanță a sarcinii

Practic, numărătoarele implementate în VHDL verifică timpul (timpul de creștere) necesar pentru a atinge un nivel de curent specific cu inductivitatea sarcinii în funcționare normală.

Abordăm următorul raționament:

Dacă acest timp este:

1. mai rapid decât cel în funcționare normală => curentul crește prea rapid => condiție de OC (scurtcircuitul este mai puțin inductiv)
2. mult mai lung ca și cel în funcționare normală => curentul nu crește => circuit în gol (OL – Open Load) => sarcina nu este conectată => nu este posibilă atingerea pragul de curent dorit

În ambele cazuri DAC-ul din blocul CLCC este folosit ca generator de tensiune de prag pentru blocul de comparator de OC al tranzistorilor HS și LS.

La pornirea profilului CLCC, în VHDL se comandă pornirea a două numărătoare, unul pentru măsura timpul necesar ca prin tranzistorul HS curentul să ajungă la o limită stabilită de DAC-ul din blocul de CLCC, celălalt numărător face același lucru dar pentru tranzistorul de LS.

Când curentul a ajuns la limita dorită, comparatorul din blocul de OC, HS și LS își schimbă ieșirea rezultând oprirea numărătorului.

Acum funcția digitală implementată în VHDL poate compara valoare numărătoarelor cu cele preprogramate în registre și poate lua o decizie bazată pe logica sa (explicată mai sus la punctele 1. și 2). Mai multe detalii despre strategia de diagnostic se regăsesc în subcapitolul 6.3.

Controlul digital propus de mine aici este de asemenea posibil pentru un mod de „Control Direct”, atunci când tranzistorul HS nu este controlat automat ON/OFF pe baza unui profil de curent programat în blocul de CLCC. În acest caz tranzistoarele LS și HS urmăresc semnalele lor de intrare "NON_LS și NON_HS", conectate la porturile de ieșire ale uC. În acest mod, blocul analogic CLCC nu este folosit.

Modul "Control Direct" este folosit pentru sarcini rezistive, prin urmare nu se poate regla curentul prin ele, deci blocul CLCC este în mare parte inutil. În acest caz, strategia de detectare a OC și OL menționată anterior (bazată pe constanta de timp a circuitului) nu poate funcționa așa cum este explicată mai sus.

În acest caz însă, deoarece blocul CLCC nu este utilizat, este posibil să se utilizeze DAC-ul din blocul CLCC exclusiv pentru generarea pragurilor OC și OL pentru blocul de comparare OC pentru LS și HS.

DAC-ul este controlat secvențial pentru a stabili două praguri în timpul fazei ON a fiecărui tranzistor:

1. un prag "mare", mai mare decât curentul de funcționare normală = pentru detectarea OC
2. un prag "mic", mult mai mic decât curentul de funcționare normal = pentru detectarea OL

Acum, codul VHDL trebuie doar să detecteze când comparatorul își schimbă starea și apoi să declare situația de avarie pe baza logicii sale. Blocul de diagnostic din VHDL trebuie să verifice dacă curentul este mai mare decât pragul „mare” rezultând în eroare de OC sau dacă curentul este mai mic decât pragul „mic” rezultând o eroare de OL. Dacă curentul este între cele două praguri, rezultă o funcționare normală.

5.1.5 Propunere de model de sarcină DIV

5.1.5.1 Simularea sarcinii reale DIV

Sarcina „dummy” (rezistor de putere + inductor toroidal) este destinată a fi utilizată pentru testarea circuitului de comandă electrică. Avantajul unei sarcini „dummy” realizată numai de o pereche de inductor-rezistor este prețul său redus în comparație cu pompa HPFP reală.

Dezavantajul este comportamentul său electric diferit datorită influențelor mecanice ale HPFP, de exemplu modificarea întrefierului bobinei, care duce la variația inductivității, ilustrată în Fig. 5.13 când se închide valva DIV. Inductanța a fost măsurată la diferiți curenți, după cum se vede mai jos; inductivitatea sarcinii „dummy” devine saturată atunci când curentul crește. Pentru valva DIV a pompei HPFP, datorită mișcării interne a axului supapei, inductanța nu se modifică liniar.

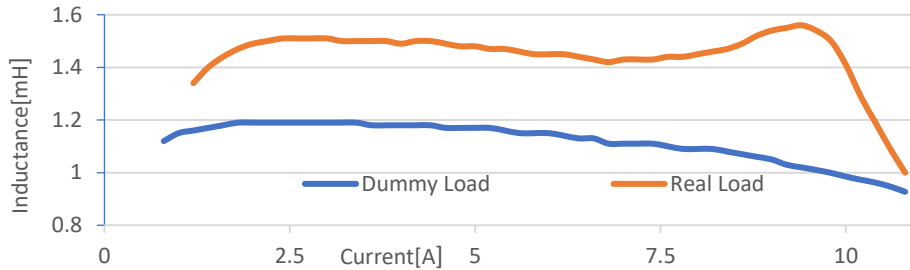


Fig. 5.13 Variația inductivității bobinei DIV în funcție de curent **Error! Reference source not found.** (original).

5.1.5.2 Modelul simplu de simulare

Simularea este utilizată pentru a ne asigura că tranzistoarele LS, HS și diodele FWD, TVS pot rezista puterii disipate în timpul fazei CLCC. După cum se observă în Fig. 5.14, modelul de simulare electrică a supapei DIV, folosit în trecut, a fost un circuit simplu serie R-L.

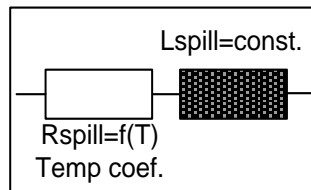


Fig. 5.14 Model simplu de sarcină pentru simulare [39] (original).

Rezistorul este echivalent cu rezistența sârmei din cupru a bobinei care are atașat un coeficient de temperatură pentru a simula variația rezistenței cu temperatura ambiantă. Inductanța bobinei DIV are o valoare constantă și nu replică comportamentul magnetic real al elementului electro-mecanic.

După cum se vede în Fig. 5.13 și Fig. 5.15, un model serie R-L nu va avea niciodată același comportament al inductivității ca și sarcina reală ($L = f(I)$ & $L = f(t)$).

Când efectuăm o simulare electrică pentru a evalua performanța circuitului de comandă (puterea maximă, puterea medie, energia limitată/clamping, curentul de freewheeling, temperatura pe joncțiunea componentelor active ...) este important să fie efectuată cu o sarcină cât mai apropiată posibil de cea reală, pentru a evita supra sau subdimensionarea componentelor active.

Din acest motiv este necesar un model de simulare cât mai exact pentru valva DIV. Partea rămasă din acest subcapitol se concentrează pe găsirea unui astfel de model pentru valva DIV a pompei HPFP.

5.1.5.3 Măsurări realizate pentru a determina variația parametrilor electrici ai supapei DIV

Pentru a găsi un model exact de simulare electrică pentru valva DIV, am realizat un set de experimente cu diverse scenarii pe diverse bancuri de încercări.

Scopul testelor este de a caracteriza comportamentul mecanic și electric al supapei DIV determinând parametrii non-electrici care influențează parametrii electrici al bobinei supapei DIV.

A. Măsurări pe bancul electric de încercare

Măsurările pe bancul electric de test sunt utile pentru caracterizarea pompelor atunci când nu există acces la un banc de testare dedicat, unde pompa HPFP funcționează într-un mediu real cu partea hidraulică și mecanică inclusă (în care inductanța bobinei și timpul de închidere/deschidere a valvei DIV variază suplimentar și datorită presiunii de combustibil).

În cazul bancului de test electric putem afirma următoarele:

- Pompa HPFP nu face parte dintr-un sistem hidraulic
- Tensiune variabilă de alimentare
- Temperatura pompei este $\sim 25^{\circ}\text{C}$
- Nu există presiune de combustibil în pompă/valvă
- Turația (RPM) este variabilă, variația turației motorului termic fiind emulată prin creșterea frecvenței de activare a supapei DIV
- Este asigurat accesul și modificarea flexibilă a profilului de curent CLCC prin bobina DIV

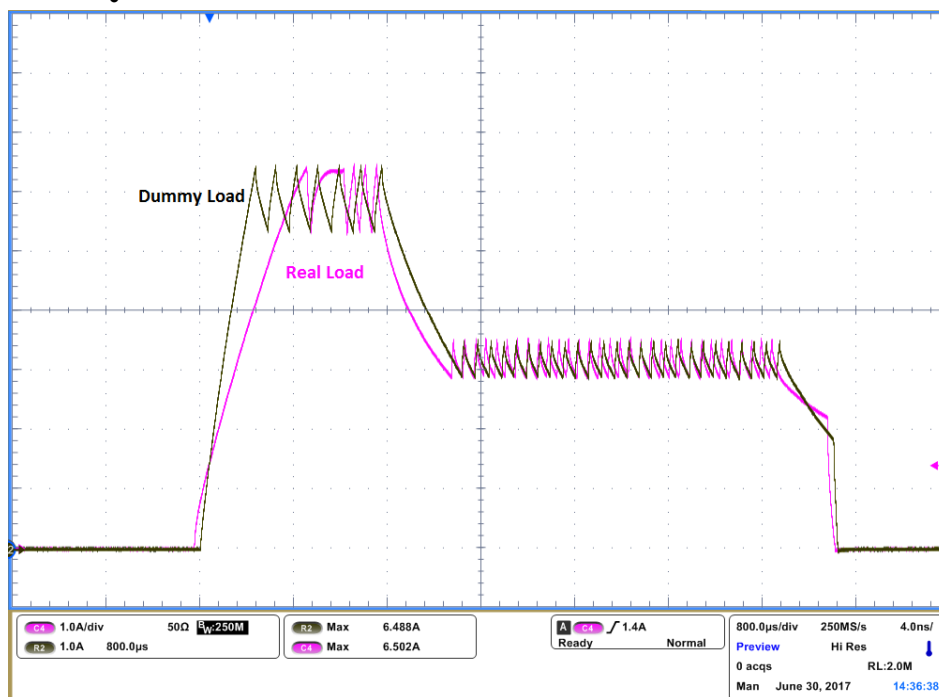


Fig. 5.15 Curentul prin sarcina dummy comparație cu cea reală [39] (original).

Măsurări comparative ale profilului de curent în funcționare normală sunt prezentate în Fig. 5.15. Curentul prin sarcina „dummy” crește mai repede decât curentul prin valva DIV. Este vizibil faptul că sarcina reală (DIV) în faza de vârf are o

discontinuitate a curentului datorită mișcării mecanice a supapei care duce la dispariția spațiului de aer (întrefier) în miezul bobinei și, prin urmare, inductanța se schimbă (crește brusc), rezultând în „glitch-ul” vizibil pe forma de curent.

Această metodă de determinare a inductivității a sarcinii reale a fost utilizată anterior de mine, valorile rezultate fiind folosite pentru construirea sarcinii „dummy” înlocuitoare a DIV. În cele din urmă însă a trebuit să analizez dacă există diferențe atunci când caracterizarea supapei de scurgere se face în scenarii mai apropiate de aplicația reală.

B. Măsurări pe bancul de test cu pompă și injectoare

Această metodă este cel mai bun compromis între măsurările pe bancul electric de încercare și măsurările pe bancul dinamometric cu motor termic real.

Acesta conține un sistem hidraulic minim și injectoare care pulverizează combustibil într-o cameră (fără explozie de combustibil).

Putem influența următorii parametri:

- Tensiunea variabilă de alimentare
- Putem emula turația variabilă a motorului
- Profilul de curent CLCC poate fi modificat
- Presiunea variabilă a carburantului

Pentru a determina parametrii care influențează variația inductivității bobinei DIV, am ales să fac următoarele măsurări:

- domeniul tensiunii de alimentare 9V-14V-16V
- turația motorului: 700 RPM (Ralanti/Idle Speed)
- presiunea combustibilului în intervalul 100bar-150bar-200bar-250bar

În Fig. 5.16 sunt prezentate un set de prelucrări în MATLAB® a unor date format CSV (Comma Separated Values) furnizate de osciloscop. Turația și presiunea motorului sunt constante, dar tensiunea de alimentare este variabilă. Este vizibil faptul că, în funcție de tensiunea de alimentare (bateria), panta de creștere a curentului este diferită, ceea ce influențează în mod direct timpul de închidere al supapei.

În momentul în care se închide supapa, apare o scădere bruscă a curentului la sfârșitul fazei de vârf a profilului de curent.

Când armătura supapei se deplasează, dispăre întrefierul din miezul magnetic, prin urmare, inductanța miezului va crește, după care miezul se va satura brusc, astfel că în forma de undă a curentului apare o scădere bruscă.

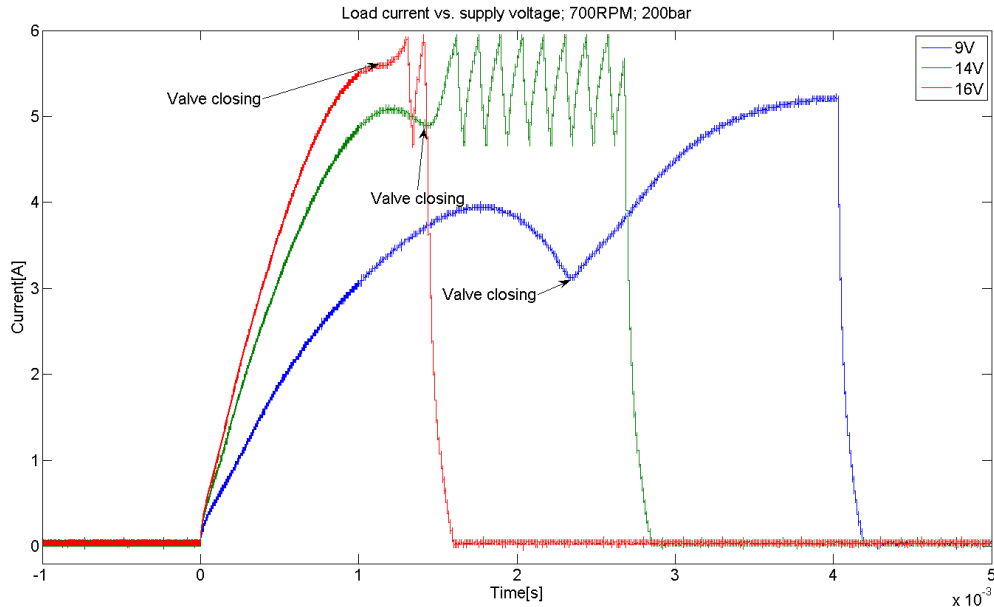


Fig. 5.16 Curentul DIV funcție de tensiunea de alimentare; 700RPM; 200bar [39] (original).

Timpul de închidere este influențat de magnetizarea miezului, supapa este ținută deschisă de un arc, forța arcului trebuie să fie învinsă de forța magnetică a bobinei pentru a mișca armătura.

Un alt efect poate fi de asemenea observat atunci când tensiunea de alimentare este mare, scăderea curentului datorată saturației bobinei DIV este mai puțin vizibilă; acest efect se datorează magnetizării mai rapide a miezului, astfel încât va imprima o forță mai mare armăturii mobile a supapei DIV rezultând într-o desfășurare mai rapidă a fenomenului de saturație.

În Fig. 5.17 este prezentată o altă măsurătoare - unde tensiunea de alimentare și turația motorului sunt constante, dar presiunea combustibilului este variabilă.

Se observă că, pe măsură ce crește presiunea, scăderea curentului este mai semnificativă în timpul închiderii supapei. Este evidentă o întârziere pentru timpul de închidere a supapei la valori diferite a presiunii de combustibil.

În concluzie saturația bobinei este dependentă de presiune, viteza de micșorare (semnificativă) a întrefierului (spațiul de aer dintre miez și armatură).

Analizând aceste rezultate pot afirma că variația presiunii de combustibil influențează forma profilului de curent în faza de vârf.

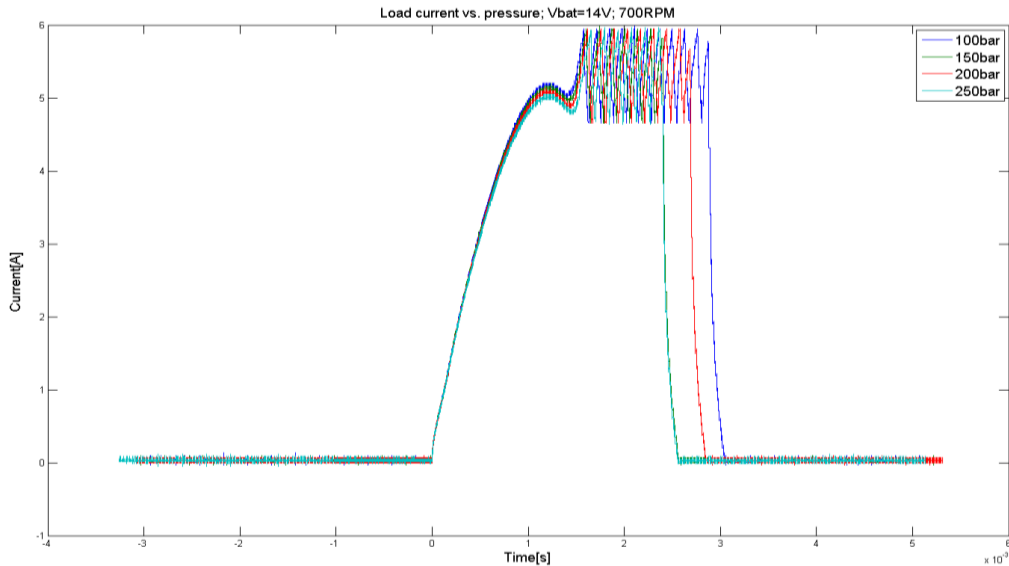


Fig. 5.17 Curentul DIV funcție de presiune; 700RPM; 14V [39] (original).

C. Măsurări pe bancul dinamometric de probă cu motorul termic

Aceste măsurări sunt cele mai apropiate de scenariul real de aplicare. Pompa HPFP este atașată la motor și conectat la sistemul hidraulic de combustibil. Scopul strategiei de control este acela de a asigura o presiune constantă în sistemul cu rampă comună, care alimentează injectoarele.

Injectoarele sunt controlate temporal, cantitatea de combustibil livrată depinde în mod direct de momentul și durata injecției. Dacă presiunea combustibilului nu este constantă, cantitatea de combustibil injectată va varia de la o injecție la alta.

În Fig. 5.18 se arată dependența curentului prin bobina DIV față de cuplul aplicat motorului.

În comparație cu măsurările anterioare, avem acum posibilitatea de a aplica motorului termic printr-un dinamometru cuplu constant, astfel încât sistemul de alimentare cu combustibil să fie testat în diferite condiții de funcționare, dar tensiunea de alimentare, temperatura și turația motorului să rămână constante.

Este vizibil faptul că amplitudinile profilului de curent nu sunt influențate de cuplul aplicat asupra motorului, singura diferență fiind momentul de comandă al închiderii supapei.

Cantitatea de combustibil în camera de presiune a HPFP este controlată printr-o întârziere de timp în comparație cu BDC (Bottom Dead Center) al pistonului pompei HPFP.

Supapa de scurgere DIV este în mod normal deschisă, combustibilul cu presiune scăzută umple camera de presiune a pompei de îndată ce pistonul pompei începe mișcarea sa în jos, de la TDC către BDC (Fig. 4.2).

Când pistonul pompei începe să se deplaseze din poziția cea mai de jos (BDC), începe faza de compresie (s-ar comprima combustibilul dacă s-ar comanda închiderea supapei DIV). Dacă supapa de scurgere nu este închisă (profil de curent aplicat),

combustibilul nu este comprimat și livrat către rampa de înaltă presiune, în schimb este pompat înapoi către sistemul de combustibil de joasă presiune.

Când supapa DIV se închide datorită profilului de curent aplicat, combustibilul va fi comprimat până când presiunea din camera de presiune a pompei va fi mai mare decât presiunea din rampa comună. Există o supapă de sens mecanică între pompa HPFP și rampa comună care oprește combustibilul de înaltă presiune să pătrundă înapoi în camera de presiune a pompei HPFP.

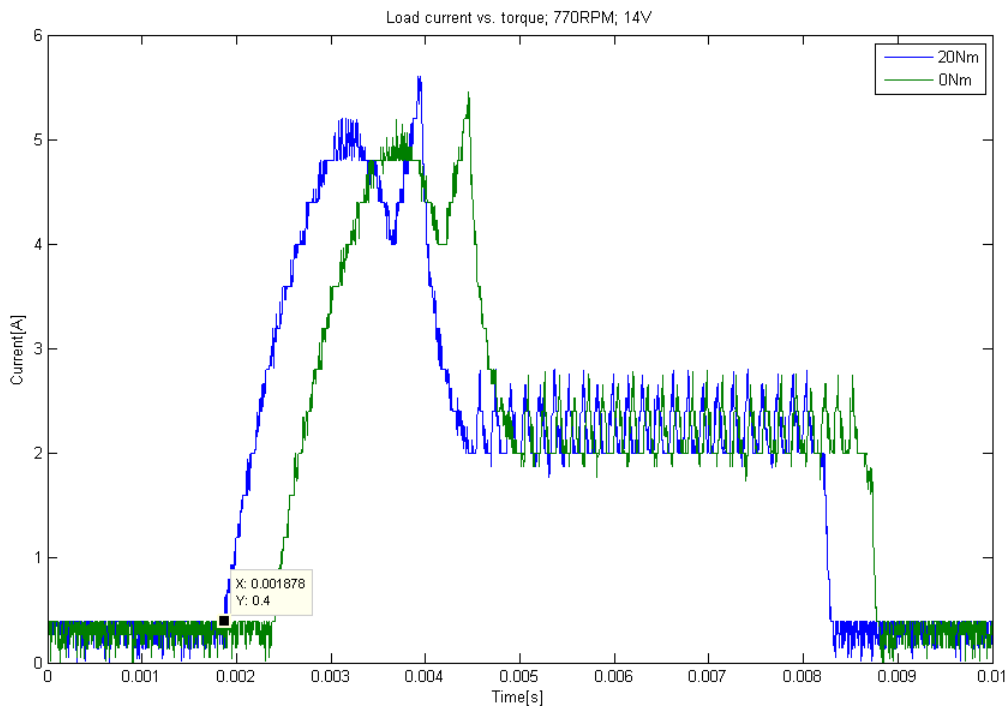


Fig. 5.18 Curentul DIV funcție de cuplu motor; 700RPM; 200bar; 14V [39] (original).

În cazul aplicării cuplului, este nevoie de mai mult combustibil, astfel încât valva DIV este închisă mai devreme comparativ cu cazul în care nu se aplică cuplu. Pentru cazul în care se aplică cuplu, mai puțin carburant este pompat înapoi în sistemul de joasă presiune și mai mult este transferat către rampa de înaltă presiune, așa cum se vede în Fig. 5.18.

5.1.5.4 Modele intermediare propuse pentru simulare

Pentru a obține rezultate fiabile de simulare, este important să avem un model de simulare electric cât mai aproape de realitate. Pe baza măsurărilor experimentale anterioare propun următorul model îmbunătățit de simulare electrică pentru valva DIV, partea dreaptă a Fig. 5.19.

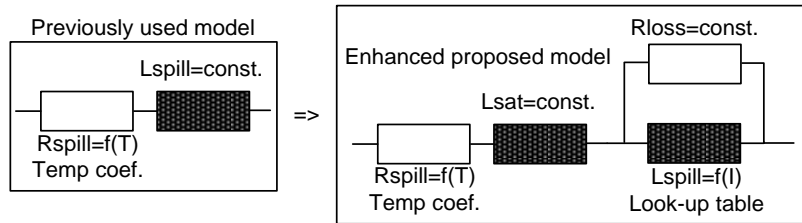


Fig. 5.19 Modele electrice a bobinei DIV dezvoltate pentru simulare [39] (original).

În comparație cu modelul anterior am introdus o inductanță de saturație a bobinei care este valabilă pentru curenții mari și o inductanță bazată pe un tabel de căutare (LUT) care este valabilă pentru curenții mici, sub curențul de saturație. Ultimul element introdus este un rezistor în paralel cu inductorul pentru a modela disiparea de putere datorată curenților turbionari din miezul magnetic.

Folosind LUT-ul, putem defini inductanța bobinei DIV în funcție de curent astfel încât la curențul de saturație să aibă valoarea zero, așadar numai inductanța de saturație fixă este luată în calcul la simulare.

5.1.5.5 Extragerea inductanței pentru modelul de simulare

Inductanța a fost extrasă pe baza măsurătorilor cu osciloscopul, am măsurat tensiunea diferențială pe bobină și curentul aplicând profilul de curent în funcționare normală. Folosind relația dintre tensiune și curent pentru un inductor (2) am calculat în mai multe puncte inductanța.

$$v(t) = L \frac{di}{dt} \quad (2)$$

După aceea, folosind o funcție polinomială de ordinul 6, valorile au fost interpolate, astfel încât toate datele necesare pentru tabelul de căutare LUT sunt disponibile cu orice rezoluție necesară a curentului. În Fig. 5.20 se arată punctele de inductanță măsurate (punctele albastre), iar linia punctată reprezintă graficul punctelor de inductivitate interpolate pe baza funcției polinomiale.

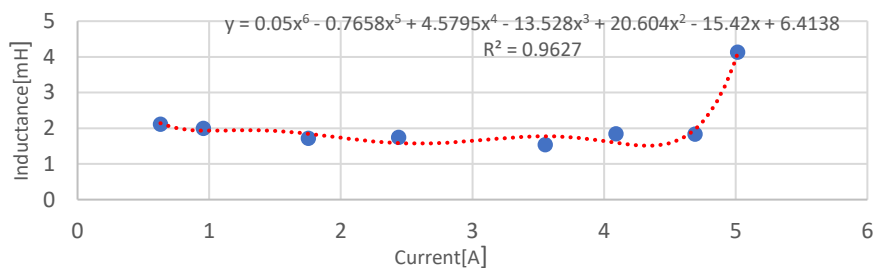


Fig. 5.20 Inductivitatea valvei funcție de curent; la închiderea valvei [39] (original).

5.1.5.6 Performanța modelelor intermediare

Simularea a fost efectuată cu ajutorul simulatorului de semnal mixt Saber® de la Synopsys.

În Fig. 5.21 este prezentată comparația dintre rezultatul de simulare electrică a supapei DIV pentru modelul simplu R-L și modelul îmbunătățit propus în comparație

cu curentul prin bobina supapei reale DIV, importat dintr-o oscilogramă care este acum forma de undă de referință.

Este vizibil faptul că modelul R-L de pe panta de creștere (faza de „pull-in”) nu urmărește forma de undă de referință, deoarece este doar o inductanță fixă, în timp ce modelul bazat pe tabelul de căutare se mulează pe forma de undă de referință foarte bine până când partea mecanică a valvei reale începe să se deplaseze, reducându-se întrefierul miezului magnetic, apărând o creștere bruscă a inductanței, prin urmare curentul prin sarcina reală începe să scadă brusc.

În faza de „hold” pentru modelul R-L există o diferență semnificativă a frecvenței de comutare, aproape dublă comparativ cu forma de undă de referință.

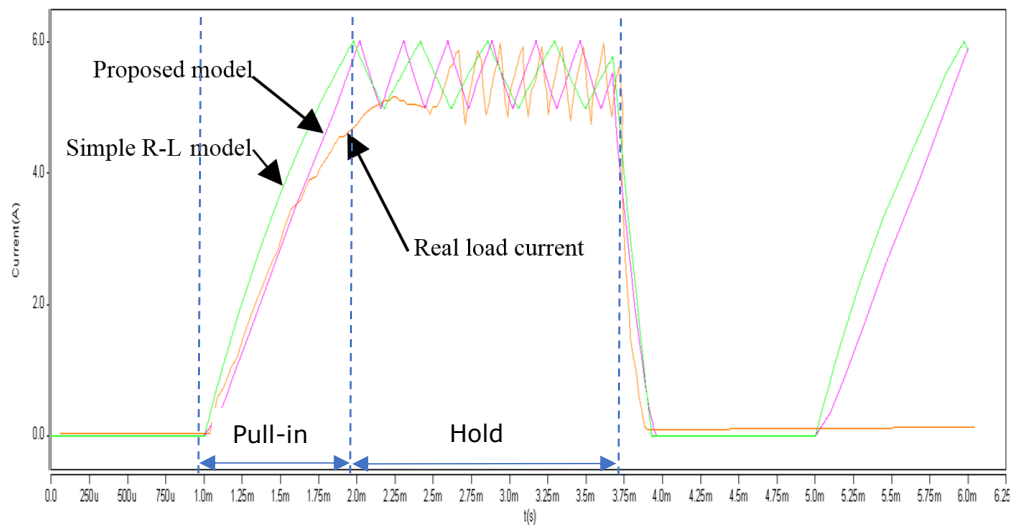


Fig. 5.21 Performanța modelelor intermediare [39] (original).

5.1.5.7 Modelul final, ierarhic de bobină DIV, propus pentru simulare

Pentru o simulare cât mai apropiată de realitate, pentru a evalua strategiile de diagnostic și de detecție a mișcării propuse și pentru a evalua corect puterea disipată a componentelor externe, este obligatoriu să dezvolt un model de sarcină precis, așa cum este prezentat în Fig. 5.22.

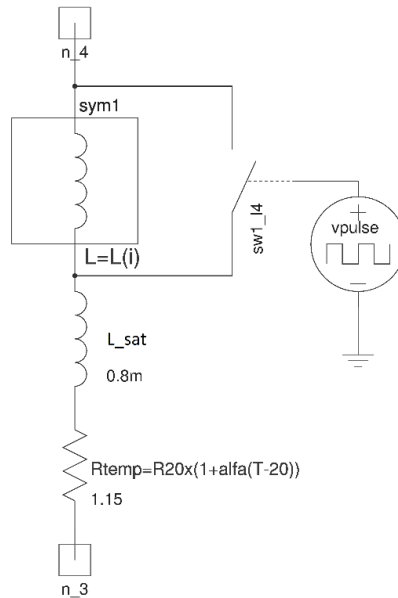


Fig. 5.22 Blocul: model de bobina a supapei DIV (original).

Modelul electric al bobinei DIV are implementate următoarele detalii:

1. Inductanța depinde de curentul prin sarcină
2. Efectul de saturație: după ce valva este alimentată și este închisă, inductanța bazată pe LUT este șuntată de un întrerupător temporal și numai inductanța de saturație rămâne în circuit
3. Rezistența bobinei este dependentă de temperatură

Principiul detectării mișcării supapei este tratat și explicat în capitolul 6, acum doresc doar să arăt cum este introdusă testarea algoritmului prin simulare, modificând pur și simplu modelul bobinei DIV. Detectarea mișcării se bazează pe analiza trecerii de la PWM (Pulse Width Modulation) relativ lent la PWM mai rapid (frecvență de circa două ori mai mare, Fig. 5.23) în faza de „hold” a profilului de curent cu control în buclă închisă. În realitate, acest lucru se întâmplă din cauza saturației inductanței bobinei DIV după ce supapa se închide.

Acest efect variază în timp, dar și în funcție de tensiunea de alimentare, practic fiind proporțional cu forța magnetică aplicată bobinei.

În Fig. 5.23 este prezentat un rezultat tipic de simulare, în care este vizibilă trecerea de la PWM lent la rapid în faza de „hold”, iar semnalul digital „o_valve_mov” își schimbă valoarea pe "1" atunci când algoritmul detectează evenimentul de mișcare.



Fig. 5.23 Curent prin sarcina DIV (original).

Pentru a realiza aceste rezultate, a fost necesar să se modifice modelul de simulare a bobinei DIV prin introducerea comutatorului temporal care îndepărtează inductanța bazată pe LUT și lasă în circuit doar inductanța de saturație.

În realitate, acest efect temporal este dependent de tensiunea de alimentare și de temperatură. Acest efect nu este încă implementat, însă în lista activităților viitoare de dezvoltare este un punct de realizat. În principiu, trebuie adăugat un al doilea LUT multi-dimensional care să returneze momentul de timp când trebuie șuntată inductanța bazată pe LUT-ul deja implementat. LUT-ul multi-dimensional trebuie să înglobeze variația timpului de închidere a supapei DIV și nu doar o sursă simplă de tensiune pulsatorie, cu o întârziere preprogramată așa cum este prezentă astăzi în model.

5.2 Implementarea circuitului fizic pentru validare

În acest subcapitolul este prezentată implementarea circuitului fizic pentru emularea funcțiilor necesare pentru controlul supapei(DIV) unei pompe HPFP. Practic, acum sunt combinate și implementate fizic același blocuri ca și cele din schema de simulare, dar acum pe un PCB. Schema și placa cu circuite imprimate au fost proiectate în programul Eagle Autodesk.

5.2.1 Blocul de filtraj și alimentare

În Fig. 5.24 este prezentată implementarea conceptului de alimentare pentru circuitul de control DIV.

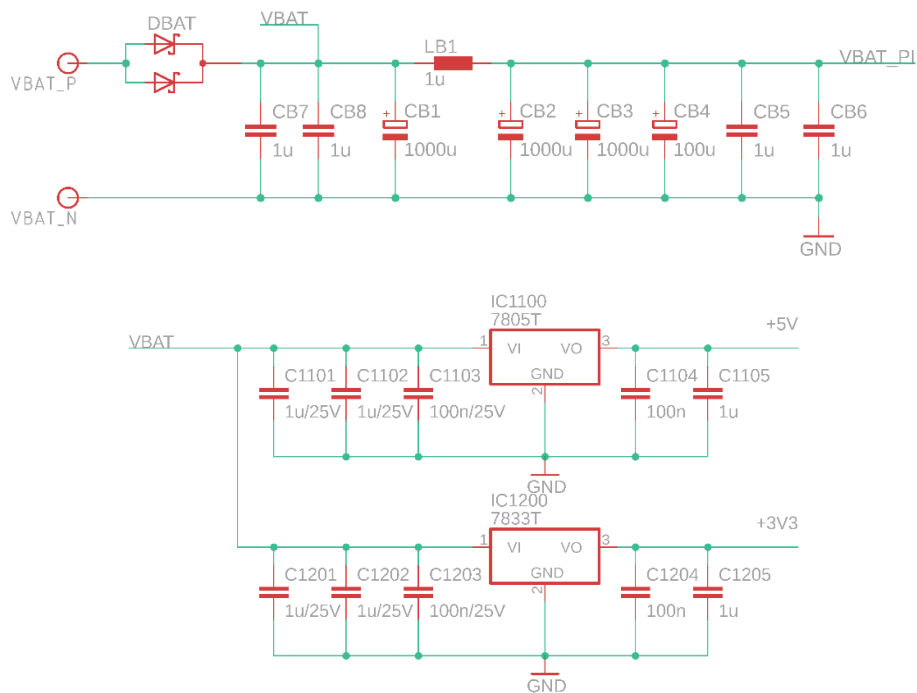


Fig. 5.24 Implementarea blocului de filtraj și alimentare (original).

Linia de alimentare este conectată la baterie printr-un conector de 4mm de tip banană, există o diodă Schottky de putere pentru protecția la alimentare cu polaritate inversă, după care este implementat un filtru PI pentru a reduce emisiile EMI (Electro-Magnetic Interference) prin introducerea unor perturbații pe linia de alimentare. La ieșirea filtrului PI a fost implementat un banc de condensatori ceramici separatori pentru a reduce curentul de pulsație (ripple) consumat din linia de alimentare (bateria de 12V a mașinii).

După filtrul PI am implementat două regulatoare de tensiune liniară:

- 5V necesar pentru circuitul analogic de condiționare a semnalului
- 3.3V necesari pentru alimentarea circuitelor digitale.

5.2.2 Blocul cu microcontroler

Funcția de microcontroler utilizată pentru comunicarea cu blocul digital ASIC-ului emulat este o placă Arduino Uno bazată pe un microcontroler ATmega328P. Placa este pur și simplu conectată la PCB prin conector de tip pini (pin header) tată/mamă, prin urmare în schema electrică este necesară doar simbolul plăcii Arduino UNO cu conexiunile de utilizator disponibile așa cum se prezintă în Fig. 5.25.

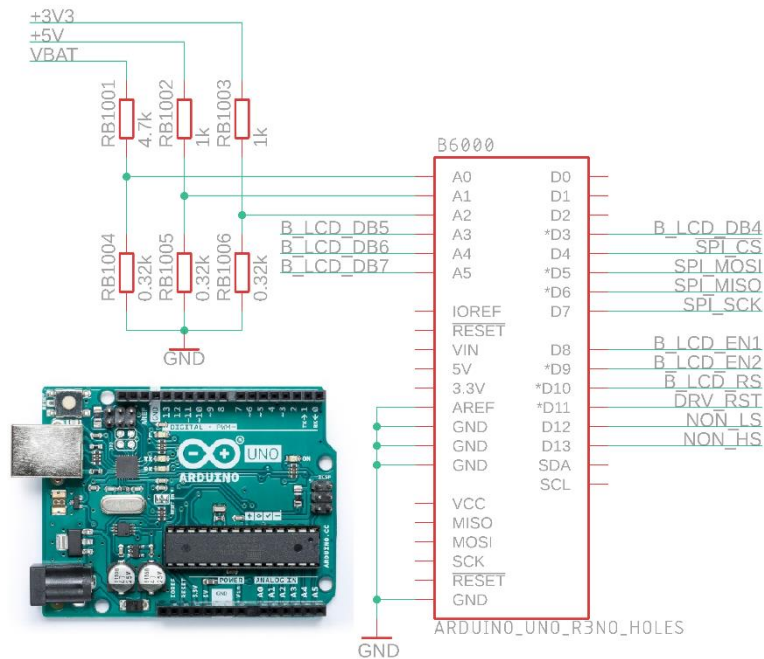


Fig. 5.25 Implementarea blocului cu microcontroler (original).

5.2.3 Circuitul de putere

Pentru a controla valva electro-magnetică DIV avem nevoie de câteva componente de putere care nu ar putea fi integrate într-un ASIC deoarece ar duce la o disipare de putere prea mare, prin urmare, ar crește dimensiunea capsulei pentru ASIC, acesta fiind motivul pentru care aceste componente sunt acum în afara ASIC-ului (în Fig. 5.26).

Componentele principale sunt cele două tranzistoare cu canal N în capsula LFPAK33, cu o dimensiune de numai 3 x 3 mm, utilizate pentru etajul final de comandă a curentului prin bobina DIV. Pentru recircularea energiei pe LS există o diodă TVS în capsulă SMPC și pe HS o diodă FWD în capsula SMB. Ambii tranzistori au un circuit simplu de încetinire între poartă-drenă pentru a reduce EMI prin controlul pantei tensiunii de ieșire.

Rezistoarele de poartă pentru a limita curentul de poartă, controlând astfel timpul de pornire/oprire al tranzistorilor, tot ca măsură EMI.

Cele două rezistoare de șunt utilizate pentru măsurarea curentului și pentru protecție și reglare de curent, având o rezistență de 10mR, capsulă 1206 și 1% toleranță.

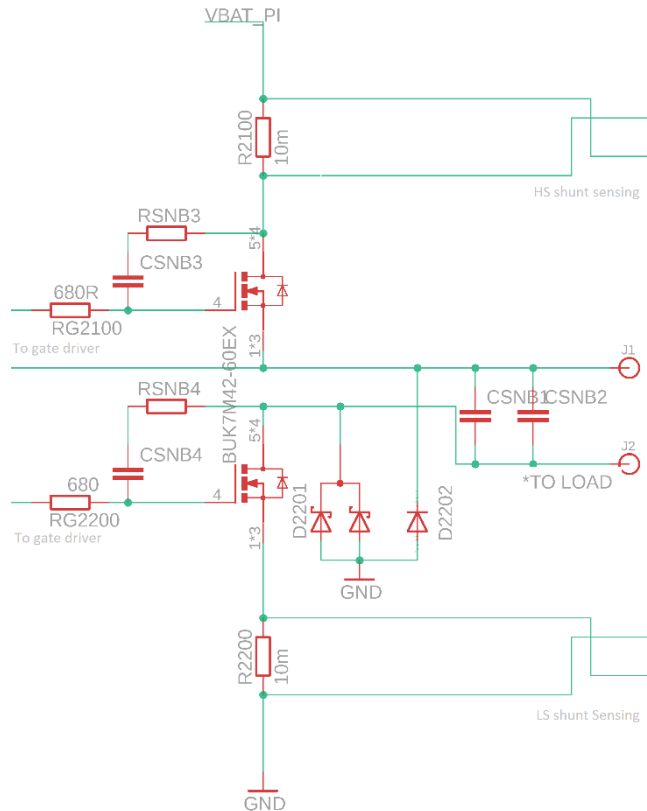


Fig. 5.26 Implementarea circuitului de putere (original).

5.2.4 Emularea ASIC-ului

În proiectare m-am concentrat pe menținerea funcțiilor ASIC-ului în blocuri separate pentru posibilitatea ulterioară de a transfera designul analogic și digital într-un ASIC (dacă este necesar). Pentru a testa funcțiile descrise în subcapitolul 5.1, concepute și descrise prin simulare, acum trebuie să implementez același circuit folosind exact aceleași componente.

„**Blocul digital**” al ASIC-ului este emulat de o placă de dezvoltare Digilent Nexys 3 cu FPPA Spartan 6, care este conectată la placa de bază prin șirul de pini de tip mamă/tată. Au fost utilizate 29 de intrări/ieșiri digitale ale plăcii de dezvoltare pentru a interfața toate semnalele digitale necesare din blocul analogic (semnale care au doar două stări „1” și „0”, transmițând informații între blocul analogic și digital) al ASIC-ului după cum urmează:

Semnalele de ieșire pentru FPGA:

- magistrala de intrare paralelă a DAC-ului

- semnal de scriere pentru DAC
- semnale pentru a controla intrarea IC-ului pre-driver HS / LS
- Diagnostic LS/HS pe 2 biți
- Detectare eroare
- Supapă Deschisă
- Supapă Închisă
- SPI MOSI

Semnalele de intrare pentru FPGA:

- Semnalul de reset de la microcontroler
- Tact principal 100MHz
- SPI_CLK
- SPI_CS
- SPI_MISO
- Comanda tranzistoarelor HS/LS de la microcontroler
- Ieșirea comparatorului de curent în buclă închisă de la tranzistorul LS
- Ieșirea comparatorului de supra-curent LS / HS

Codul VHDL folosit pentru modelul de simulare a fost compilat și sintetizat utilizând software-ul Xilinx ISE Design 14.7. Au fost executate operațiuni de plasare și rutare la pinii FPGA dorit a logicii rezultate după sintetizare. Fișierul bitstream rezultat a fost tradus în format de fișier HEX cu extensia ".mcs" și programat în memoria SPI FLASH de pe placa de dezvoltare FPGA cu ajutorul utilitarului ADEPT din lanțul de instrumente Xilinx/Digilent. În acest mod, FPGA la fiecare secvență de inițializare după alimentare este încărcat automat cu fișierul dorit.

„**Blocul analogic al ASIC-ului**” este emulat de următoarele sub-blocuri:

- Măsurare de curent prin șunt LS
- Controlul curentului în buclă închisă
- Pre-driver MOSFET extern pentru LS și HS inclusiv circuit de bootstrap
- Comparator supra-curent HS
- Comparator supra-curent LS
- Măsurare de curent prin șunt HS

Blocul "**LS shunt sensing**" este prezentat în Fig. 5.27, se bazează pe un amplificator de curent precis, INA180 cu un câștig intern stabilit la 20 și cu offset al tensiunii de intrare tipic de 150uV.

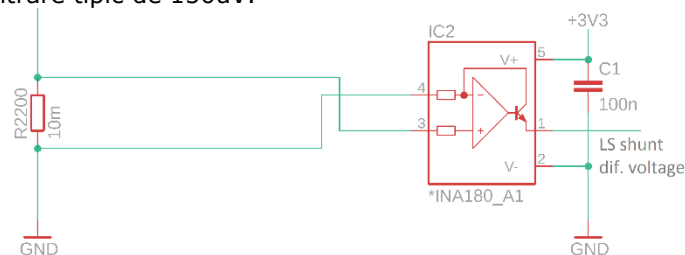


Fig. 5.27 Implementarea circuitului de măsurare de curent LS (original).

Amplificatorul operațional având un câștig fixat intern și alimentat de la linia de 3.3V, curentul tipic maxim de măsurare utilizând un șunt de 10mR este:

$$I_{sense_max} = \frac{V_{sense_max}}{R_{shunt}} = \frac{V_{supply}[V]-0.03}{Gain} * \frac{1}{R_{shunt}[\Omega]} = \frac{3.3-0.03}{20*0.01} = 16.35A \quad (3)$$

Pentru bobina DIV a unei pompe HPFP pentru benzină, curentul maxim de funcționare este de aproximativ 7A, iar pentru pompa HPFP la sistemele cu motorină

este de 13A, acoperind practic ambele valori. Ieșirea amplificatorului diferențial este conectată la intrarea comparatorului din blocul de control de curent în buclă închisă dar și la comparatorul de supracurent din blocul de detectoare de OC.

Blocul "**Closed Loop Current Control**" are rolul de control de curent în buclă închisă și este emulat de un DAC cu intrare paralelă pe 8 biți și de un comparator în contra timp (push-pull) rapid prezentat în Fig. 5.28.

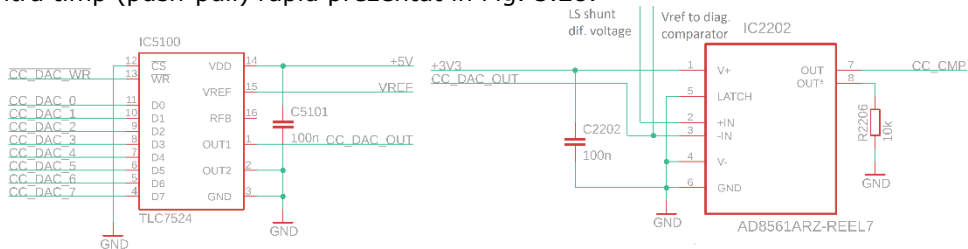


Fig. 5.28 Implementarea circuitului: Control de Curent în Buclă Închisă (original).

FPGA-ul emulează blocul digital al ASIC-ului controlând întreaga funcție de reglare de curent în buclă închisă, CLCC. Microcontrolerul este responsabil pentru programarea registrelor ASIC-ului cu profilul de curent dorit, care urmează să fie generat. Blocul digital trimite pe o magistrală paralelă datele necesare către DAC transformând datele digitale într-o tensiune care este conectată la intrarea inversoare a comparatorului din blocul de CLCC, după care tranzistorul HS este pornit (tranzistorul LS a fost activat în prealabil ca și condiție de start a profilului de curent) până când se detectează curentul de sarcină dorit măsurând căderea de tensiune pe șuntul de pe LS. Tensiunea diferențială de pe șuntul de LS prin blocul "LS shunt sensing" este feedback-ul de către intrarea neinversoare a comparatorului de CLCC.

Atunci când curentul prin tranzistorul LS (echivalent cu curentul prin șunt sau curentul prin bobina DIV) a atins pragul setat de DAC, comparatorul își schimbă ieșirea (conectată la un port de intrare al FPGA) astfel FPGA-ul controlează driverul tranzistorului HS să se oprească până la următorul prag de curent setat (mai mic decât pragul anterior). Prin acest mecanism este implementat controlul curentului între praguri bine definite.

Scopul principal al acestui bloc este de a avea un timp de răspuns al lanțului analog-digital mai rapid decât constanta de timp definită de inductanța tipică a bobinelor DIV. Funcția CLCC de control de curent este destinat exclusiv pentru sarcini inductive. Sarcinile pur rezistive sau cu inductivitate mică nu sunt adecvate controlului de curent datorită creșterii prea rapide a acestuia.

Blocurile **"HS shunt sensing"** și **"LS shunt sensing"** sunt prezentate în Fig. 5.29.

Componenta principală este amplificatorul diferențial INA180 cu tensiune de intrare de mod comun mare (26V), adecvat pentru măsurarea căderii de tensiune a șuntului de pe HS. Șuntul fiind conectat la linia de alimentare (bateria mașinii), tensiunea de mod comun este:

$$V_{CM} = \frac{V_{in+} + V_{in-}}{2} \quad (4)$$

Pentru majoritatea amplificatoarelor operaționale tensiunea maxima de mod comun $V_{CM}(4)$ trebuie să fie mai mică decât tensiunea de alimentare (5).

$$V_{CM} < V_{CC} \quad (5)$$

În acest caz pentru un amplificator operațional uzual în configurație de amplificator diferențial tensiunea de mod comun pentru tensiunea de alimentare de 3.3V este aproximativ 3.3V în funcție de componenta aleasă.

Este clar vizibil că în cazul șuntului conectat la „baterie” tensiunea de mod comun pentru cazul cel mai defavorabil când bateria mașini poate avea maxim 16V va fi (6), unde V_{shunt} este neglijabil pentru un șunt de 10mR la un curent de maxim 13A.

$$V_{CMwc} = \frac{V_{bat} + (V_{bat} - V_{shunt})}{2} \quad (6)$$

Putem spune că în acest caz $V_{CMwc}=8V$, ceea ce este mult peste tensiunea de mod comun care o poate accepta un amplificator operațional uzual alimentat la 3.3V, motiv pentru care am ales INA180 care are datorită structurii interne diferite o tensiune de mod comun de 26V independentă de tensiunea de alimentare a sa.

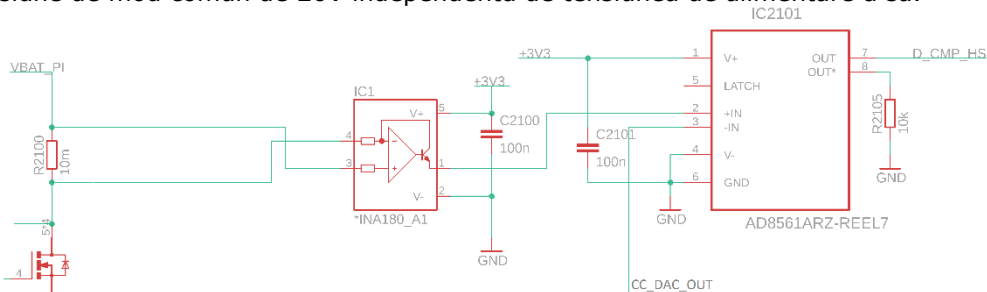


Fig. 5.29 Implementarea circuitului de măsurare de curent HS; identic cu cel de LS (original).

Ieșirea amplificatorului diferențial este conectată la intrarea neînversoare a unui comparator rapid; intrarea înversoare este conectată la o tensiune de referință generată de DAC-ul controlat de blocul digital.

Ieșirea comparatorului este intrare către blocul digital, astfel modulul de comandă știe întotdeauna dacă curentul este peste un anumit prag definit și poate lua măsuri de oprire a tranzistorului HS ca să reducă curentul. Pentru a evita detectarea falsă a OC din cauza pulsurilor tranzitorii de pe linia de baterie sau din cauza zgomotului de pe ieșirea comparatorului, în blocul digital au fost implementate filtre digitale cu numărătoare care ignoră orice tranziție a semnalului de intrare mai scurtă decât o limită predefinită digital pentru validarea doar a evenimentelor OC mai lungi decât timpul de filtrare stabilit (de numărătoare).

5.2.5 Implementarea circuitului pe PCB

Amplasarea componentelor/blocurilor funcționale pe placa de circuite imprimare este prezentată în Fig. 5.30 conform schemei electrice din anexele A1 și A2. Pornind de la colțul din stânga sus unde este amplasat conectorul de alimentare, placa este alimentată cu până la 19V/5A (curent mediu, deoarece bobina DIV este alimentată cu o tensiune cu frecvența și durata variabile în funcție de turația motorului termic) prin intermediul conectorilor de tip banană cu orificii de 4 mm.

Prima componentă este o diodă dublă, de putere, pentru a proteja întregul circuit împotriva alimentării inverse. Următorul bloc este un filtru PI pentru a reduce emisiile electro-magnetice atunci când se comută curenții mari prin bobina DIV. Filtrul PI este utilizat numai pentru alimentarea circuitului de putere care nu face parte din emularea ASIC-ului, adică pentru calea de curent prin tranzistorul HS spre potențialul de referință de alimentare prin intermediul tranzistorului LS.

Imediat după circuitul de protecție la alimentare inversă sunt două regulatoare de tensiune liniare care generează liniile de alimentare de 5V și 3,3V necesare pentru alimentarea blocurilor analogice, respectiv digitale.

În colțul din dreapta sus sunt amplasate circuitele de putere care nu fac parte din ASIC:

- tranzistorul HS
- tranzistorul LS
- dioda FWD de pe HS
- dioda TVS de pe LS și șunturile de LS/HS
- conectorii de tip banană de 4 mm pentru bobina DIV

În partea inferioară a plăcii, în colțul din stânga este atașată placa Arduino Uno utilizând șir de conectori cu pini tată/mamă.

Ușor spre dreapta sunt LED-urile de stare care oferă informații pentru utilizator despre starea circuitului, inclusiv date de diagnosticare.

În partea inferioară sunt amplasate cinci blocuri cu conectori șir de pini dubli la 90 de grade, care servesc drept interfață de conectare pentru placa de dezvoltare Xilinx Nexys 3 FPGA.

În cele din urmă, în partea din dreapta-mijloc a plăcii sunt circuitele integrate și componente discrete folosite pentru emularea blocului analogic al ASIC-ului după cum urmează:

- DAC-ul folosit în blocul de control al curentului în buclă închisă (CLCC)
- Comparatorul din blocul CLCC
- Comparatoarele de supra curent de pe HS/LS
- Amplificator diferențial șunt HS
- Amplificator diferențial șunt ȘS
- Pre-driver al tranzistorilor HS/LS incluzând circuitul de bootstrap.

Realizarea fizică a PCB-ului este prezentată în anexa A3, A4, A5, A6.

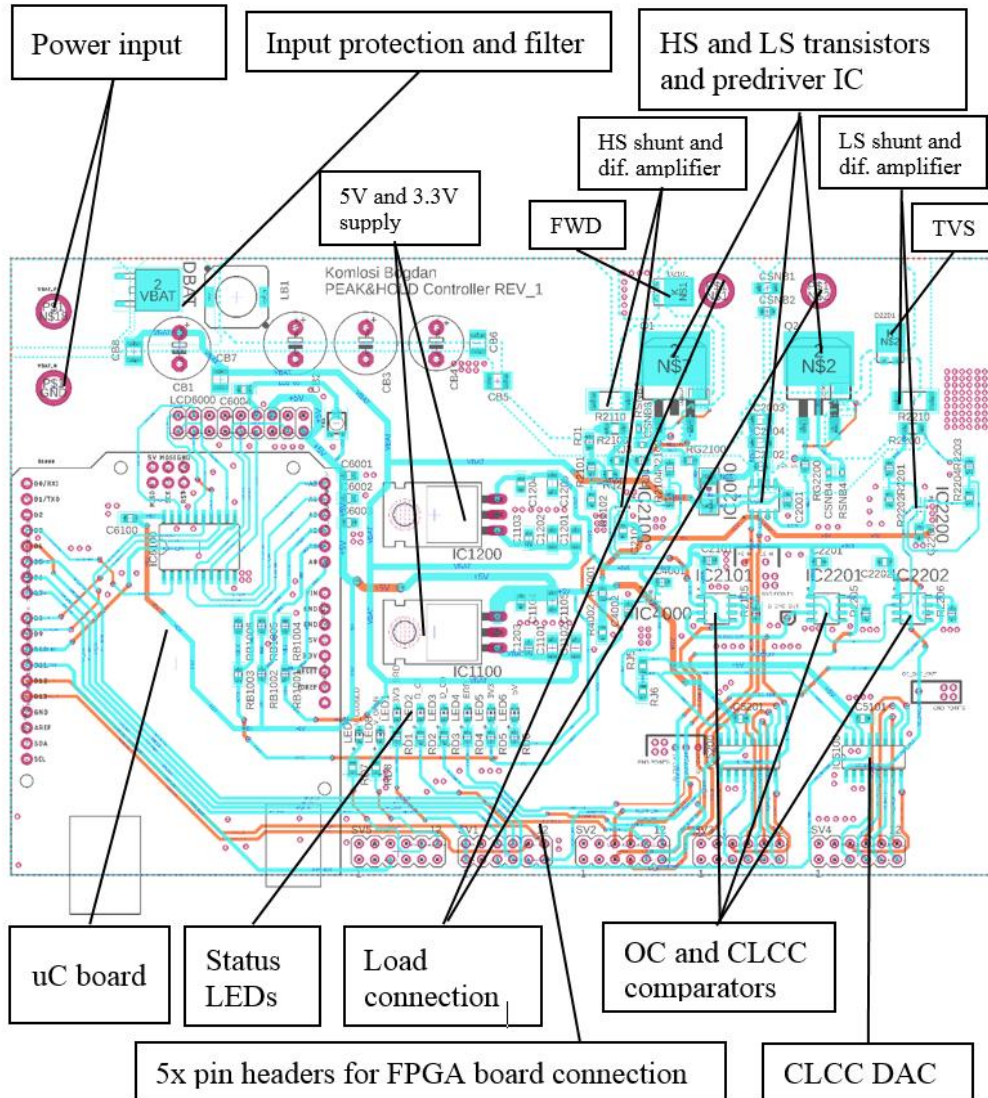


Fig. 5.30 Implementarea circuitului imprimat (original).

5.3 Compararea rezultatelor simulării cu performanțele sistemului fizic

Următoarea etapă este compararea rezultatelor simulării cu comportamentul circuitului fizic implementat. FPGA-ul este actualizat cu cea mai recentă versiune de cod VHDL și uC cu cel mai recent software. O interfață PC utilizator dezvoltată în colectiv, este utilizată pentru a actualiza și a realiza comunicarea dintre uC și FPGA. O sarcină reală, pompa HPFP cu bobina DIV este conectată la bornele circuitului de putere a platformei de emulare a ASIC-ului.

5.3.1 Rezultatele circuitului experimental

În Fig. 5.31 este prezentat un rezultat de validare care prezintă un profil tipic de curent pentru controlul supapei DIV constând din două pulsuri de curent consecutive, primul pentru închiderea supapei și al doilea pentru frânarea deschiderii supapei cu scopul reducerii zgomotului când supapa atinge poziția ei finală.

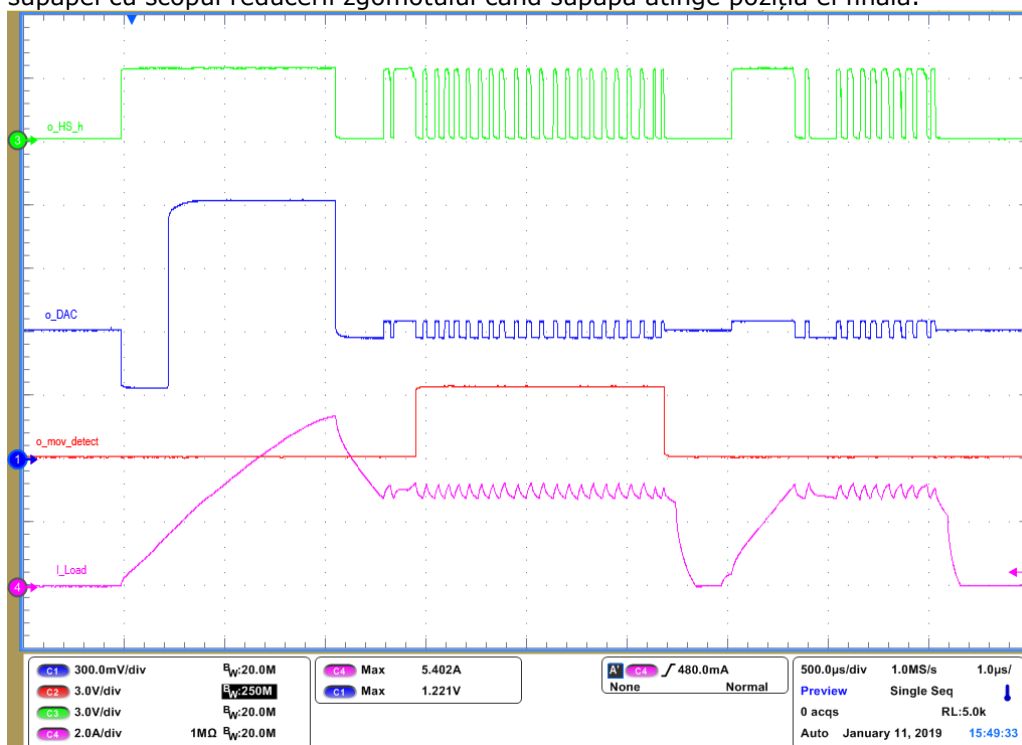


Fig. 5.31 Oscilogramă circuit experimental (original).

Canalul 4 "I_Load" reprezintă curentul prin bobina DIV, **canalul 1** este ieșirea DAC-ului din blocul CLCC generând o tensiune de referință dinamică pentru comparatorul din blocul CLCC (pe cealaltă intrare a comparatorului este tensiunea diferențială măsurată pe șuntul LS, care nu este prezentă în oscilogramă). Ieșirea

comparatorului este conectată la FPGA și se modifică dinamic în funcție de valoarea curentului de ieșire și a tensiunii de la ieșirea DAC-ului. În cazul în care căderea de tensiune pe șuntul de curent LS este mai mică decât tensiunea de referință definită de DAC, blocul digital controlează tranzistorul HS ON; când tensiunea pe șunt este mai mare decât referința programată, tranzistorul HS este oprit, așa cum se arată prin semnalul "o_hs_h" pe **canalul 3** al oscilogrammei.

Datorită construcției valvei DIV, când aceasta este energizată, după o constantă de timp definită de anumiți parametri, supapa se închide. Acest fenomen este evidențiat de „gâlma” de curent din faza de "hold", așa cum se observă în semnalul "I_Load" după aproximativ 1.4 ms de la începutul profilului de curent. Acest efect este folosit pentru a determina starea supapei prin algoritmul "detectarea mișcării supapei" prezentat în subcapitolul 6.2 implementat în FPGA, emulând blocul digital al ASIC-ului.

Starea supapei (deschisă sau închisă) devine semnal de intrare către uC prin "o_mov_detect" reprezentat pe **canalul 2** al oscilogrammei, astfel încât software-ul care rulează pe uC cu scopul de a controla presiunea combustibilului în rampa comună știe întotdeauna starea supapei. În prezent, detectarea mișcării supapei este implementată numai pentru primul impuls de curent.

5.3.2 Analiză de scenarii complexe prin simulare

În subcapitolul 5.3.1 am analizat comparatorul din blocul de CLCC, dar nu aveam toate semnalele de intrare/ieșire ale sale în oscilograma din Fig. 5.31, deoarece osciloscopul disponibil are doar patru canale de intrare analogice.

În Fig. 5.32 este prezentat modul în care funcționează blocul CLCC, pentru a înțelege mai bine relația dintre tensiunea diferențială_șunt_LS <-> DAC <-> comparator_CLCC <-> bloc digital.

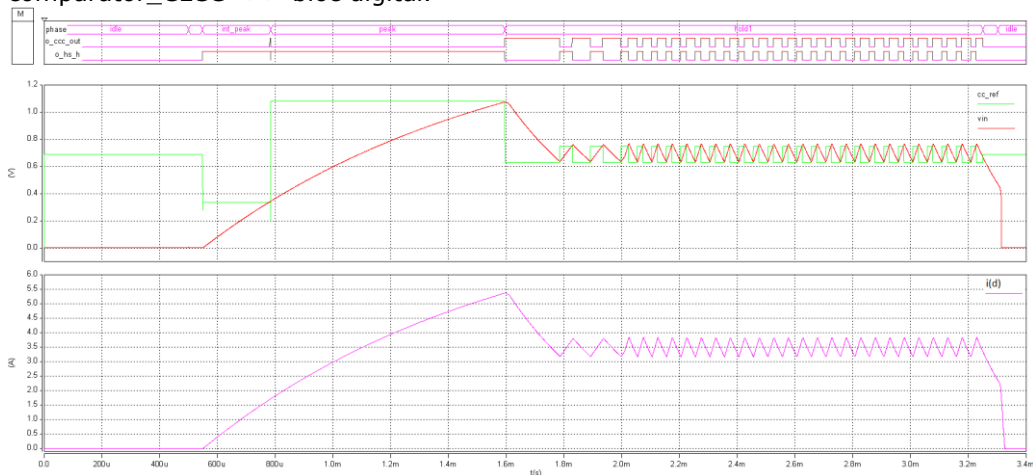


Fig. 5.32 Rezultate simulare (original).

Pornind de jos, "**i(d)**" este curentul de drenă al tranzistorului LS, șuntul este conectat între terminalul de sursă al tranzistorului și potențialul de referință (masă) al circuitului. Pentru a evita măsurarea falsă a curentului cauzată de deplasarea tensiunii de referință (ground shift) datorată căderii de tensiune la curenți mari pe conductorul de masă, se utilizează o măsurare diferențială pentru a detecta căderea

de tensiune a șuntului LS reprezentat de semnalul **"vin"** amplificat cu 20 (câștig intern al INA181).

Această tensiune este conectată la intrarea neinversoare a comparatorului. La intrarea inversoare a comparatorului este conectată ieșirea DAC-ului ca tensiune de referință (semnalul **"cc_ref"**). Semnalul **"o_ccc_out"** este ieșirea comparatorului de curent; atunci când curentul de ieșire este sub pragul definit de DAC, ieșirea comparatorului este **"low"**, imediat ce curentul de ieșire este peste pragul definit, ieșirea comparatorului este **"high"**. Ieșirea comparatorului este conectată la un pin de intrare al FPGA (bloc digital ASIC), astfel blocul actual CLCC de control al curentului "va ști" să controleze tranzistorul HS, pornindu-l (ON state) atunci când curentul este sub valoarea dorită și oprindu-l (OFF state) atunci când curentul este peste pragul definit, așa cum indică semnalul **"o_hs_h"**. Acest semnal este în continuare conectat la blocul de pre-driver de tranzistoare, servind ca semnal de comandă pentru tranzistorul HS.

5.3.3 Rezultatele simulării

În Fig. 5.33 sunt comparate rezultatele reale din oscilograma din Fig. 5.31 cu rezultatele simulării aceluiași semnale folosind date de configurare identice cu cele pentru FPGA .

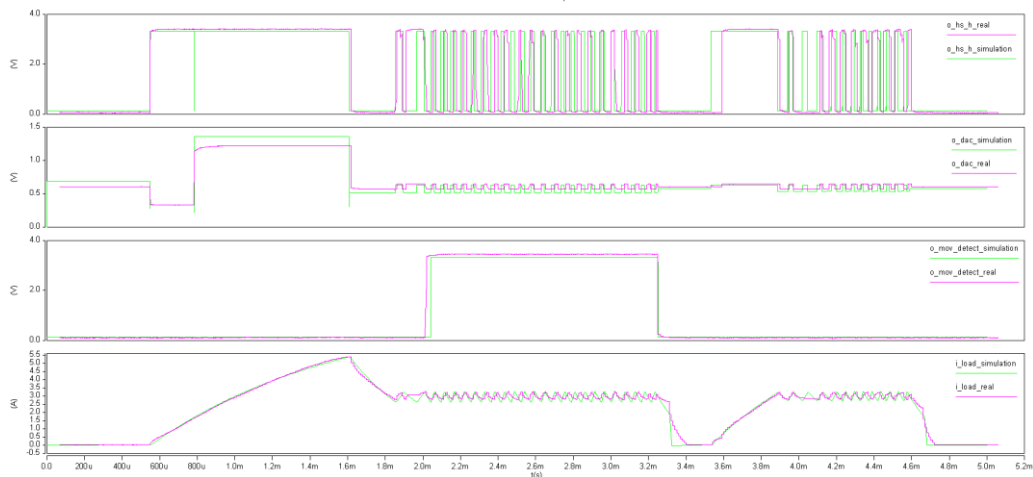


Fig. 5.33 Rezultate de simulare în comparație cu cele reale (original).

Începând din partea de jos, semnalele **"i_load"** (reprezentând curentul de sarcină) se potrivesc foarte bine între ele, existând o diferență în faza de „hold” atunci când supapa se închide, curentul real are o schimbare specifică a inductanței rezultând o fază de ON mai lungă pentru controlul HS. Modelul de sarcină nu este capabil să reproducă exact același efect, dar își schimbă valoarea de inductanță, rezultând după aceea o comutare mai rapidă exact ca în cazul validării reale, așa cum se observă în graficul cu detalii din Fig. 5.34.

Cealaltă diferență este la sfârșitul fiecărui puls de curent, panta curentului în faza de clamping este diferit reprodușă în comparație cu cea reală (în simulare disiparea puterii datorită tensiunii electromotoare induse este mai rapidă chiar dacă fenomenul a pornit de la același curent, indicând o inductanță puțin mai mică la

sfârșitul profilului de curent pentru modelul de simulare comparație cu bobina reală DIV).

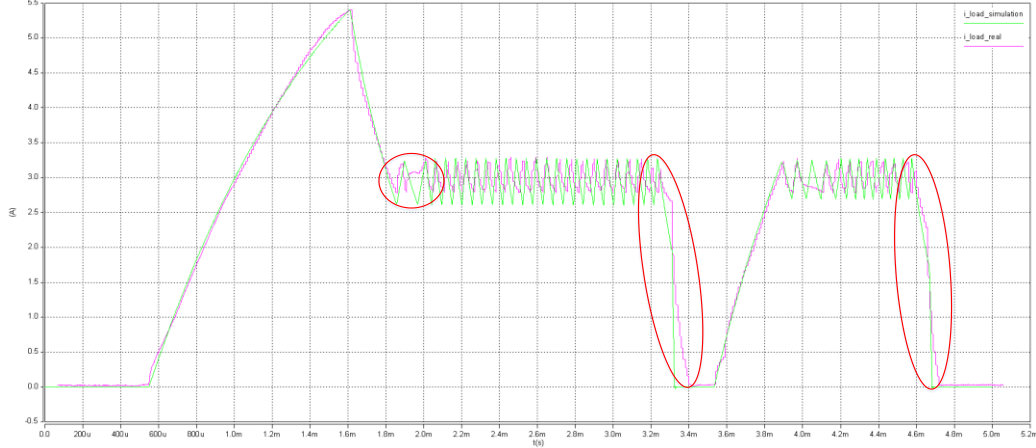


Fig. 5.34 Profilul de curent real comparație cu cel simulat – detaliu figura 5.33 (original).

Pornind de la stânga, unde curentul începe să crească (faza de pull-in), ambele profiluri de curent, real și simulat, se urmăresc unul pe celălalt. În faza de „hold”, curentul simulat iarăși îl urmărește pe cel real cu o mică diferență la valoarea curentului de histereză. Curentul simulat are o histereză de $3.2 - 2.6 = 0,6A$ iar curentul real are o histereză de $3.2 - 2.8 = 0.4A$, deci o diferență de $0,2A$.

La cel de-al doilea impuls, faza de creștere a curentului (pull-in) se potrivește chiar mai bine, pentru acest puls diferența dintre cele două histereze fiind mult mai mică. Rămâne de investigat motivul care duce la diferența de comportament.

Comparând semnalele **"o_mov_detect_real"** și **"o_mov_detect_simulation"**, rezultă doar o mică diferență temporală cauzată de diferența dintre modelul real și modelul de simulare DIV, în esență momentul comutării inductanței în modelul de simulare definește momentul de "închidere a supapei", așa cum se arată în Fig. 5.35.

Această constantă de timp trebuie să fie inclusă într-o tabelă de căutare funcție de temperatură, tensiunea bateriei și presiunea de combustibil în rampa comună pentru a avea un timp de închidere precis.

În cazul ideal, amplitudinea celor 2 semnale trebuia să fie $3.3V$, însă se observă o ușoară diferență de aproximativ $100mV$, tensiunea în circuitul real fiind de $3.4V$.

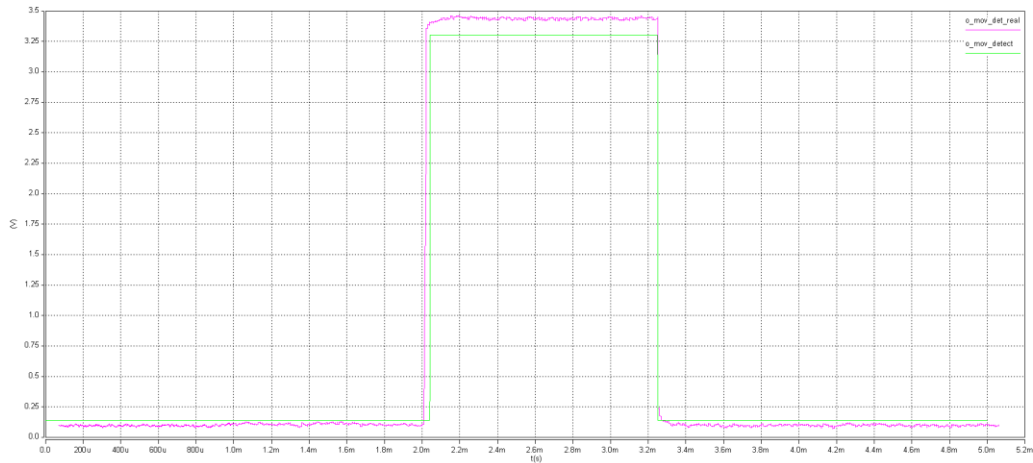


Fig. 5.35 Semnalizarea detecției mișcării: real comparație cu cel simulat – detaliu figura 5.33 (original).

Semnalele "**o_dac_real**" și "**o_dac_simulation**" sunt și ele în concordanță, existând doar o diferență în panta de creștere a semnalului real, cel mai probabil capacitatea traseelor de PCB și capacitatea de intrare a celorlalte circuite conectate la ieșirea DAC-ului limitează timpul de creștere a tensiunii. Diferența mică de amplitudine este cauzată de precizia tensiunii de referință conectată la DAC, așa cum este arătat în Fig. 5.36.

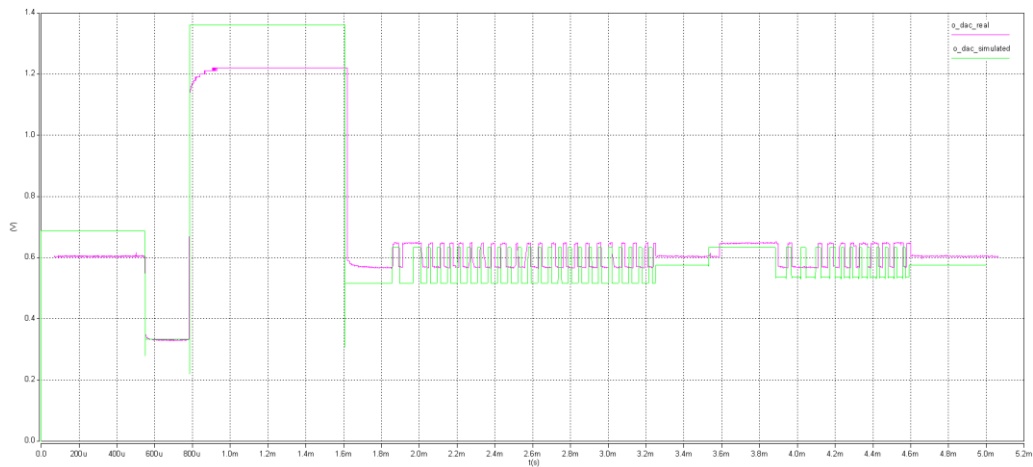


Fig. 5.36 Ieșirea DAC-ului: real comparație cu simulare – detaliu figura 5.33 (original).

În cele din urmă, în Fig. 5.37 sunt prezentate semnalele "**o_hs_h_simulated**" și "**o_hs_h_real**" reprezentând semnalele de comandă conectate la intrarea blocului de comandă al tranzistorului HS. Când semnalul este "SUS", tranzistorul este comandat ON. Tranzistorul este comandat OFF atunci când căderea de tensiune pe rezistorul de derivație LS este mai mare decât tensiunea de referință definită de DAC, astfel încât curentul de ieșire să atingă pragul dorit.

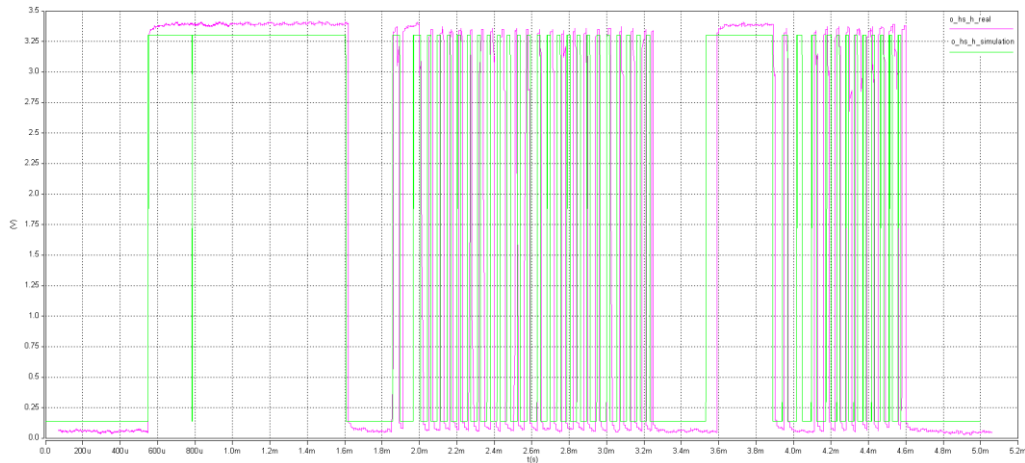


Fig. 5.37 Comanda tranzistorului HS: real comparație cu simulare – detaliu figura 5.33 (original).

Diferența între amplitudinile "low" și "high" ale semnalului real comparat cu cel simulat se explică prin faptul că simulatorul are praguri predefinite pentru vizualizarea (afișarea) semnalului digital. Aceste semnale sunt simulate ca cele digitale dar au opțiunea de a fi afișate ca cele analogice. O valoare digitală "low" este afișată ca 0.2V și un "high" digital este afișat ca 3.3V.

Punctele de pornire ale profilului de curent sunt bine definite de către ambele semnale, totuși semnalul simulat nu poate reproduce mai exact comportamentul temporal pe durata fazei de „hold” datorită inductanței diferite a modelului de sarcină DIV comparativ cu inductanța bobinei valvei DIV reale.

5.4 Concluzii și contribuții

În acest capitol am tratat implementarea celor două platforme propuse în Capitolul 3, platforma de simulare cu un singur simulator multi-domeniu și platforma MTP de emulare a ASIC-ului. Ambele platforme facilitează testarea în fază timpurie a procesului de dezvoltare a ASIC-ului evitând propagarea eventualelor erori/defecte de concept/arhitectură în producția ASIC-ului. Am prezentat schema globală de simulare organizată ierarhic, cu detalii specifice referitoare la fiecare bloc component a ierarhiei. Am ilustrat modelul electric de simulare pentru valva electro-mecanică DIV. Am explicat implementarea fizică a platformei de emulare a ASIC-ului organizată tot pe blocuri ierarhice. În final am comparat rezultatele simulării cu cele ale implementării fizice. Schema platformei de emulare a ASIC-ului a fost concepută bazată pe schema de simulare și folosind exact aceleași componente ca în simulare.

Blocul digital din simulare este emulat de un FPGA, configurarea și comunicarea fiind realizate de un uC ATMEGA328.

5.4.1 Concluzii

1. Organizarea ierarhică a schemei de simulare ușurează depanarea schemei și facilitează împărțirea/delimitarea circuitelor folosite ulterior pentru emulare ASIC-ului prin platforma MTP.
2. Organizarea schemei pe blocuri funcționale individuale ajută ulterior la refolosirea lor în procese noi de dezvoltare a altor ASIC-uri.
3. Detaliile de implementare a blocurilor funcționale au influență directă asupra rezultatelor de simulare și a timpului de execuție a simulării. De exemplu în cazul schemei de simulare care utilizează model de tranzistor HS și LS ideali, fără toleranțe, 5.4ms timp de simulare se execută în 103s, așa cum arată raportul din anexa A7.
4. Modelarea cu prea puține detalii poate fi o sursă de erori când comparăm rezultatele cu cele ale platformei de emulare a ASIC-ului. În anexa A8 este prezentat cazul unde se compară rezultatele simulării cu tranzistori HS și LS ideali cu cel cu tranzistori reali. Se observă o diferență semnificativă în caracteristicile de comutație ale tranzistorului de HS, care rezultă o putere disipată mai mare în cazul în care se utilizează model ideal de tranzistor. Puterea disipată medie per ciclu de comutație în cazul tranzistorului ideal este de aprox 0.81W iar în cazul cu model real este 0.47W, o diferență de 42% ceea ce poate conduce la o posibilă supradimensionare a circuitului de putere.
5. Modelarea cu prea multe detalii crește semnificativ timpul de simulare fiind nevoie de putere de calcul adițională pentru a rula simularea în timp util. De exemplu în cazul schemei de simulare care utilizează model de tranzistor HS și LS real, cu toleranțe, 5.4ms timp de simulare se execută în 128s, timp de simulare cu 25% mai lung față de simularea cu model de tranzistor ideal, așa cum arată raportul din anexa A7.
6. Sistemul hidraulic nu influențează considerabil performanțele electrice ale valvei DIV.
7. Cuplul aplicat motorului introduce doar o întârziere în pornirea activării supapei în comparație cu BDC.
8. Efectul relevant pentru variația inductanței valvei electro-mecanice DIV este datorat mișcării supapei de scurgere.
9. Rezistența bobinei DIV este influențată direct de temperatura mediului în care aceasta se găsește.
10. Pentru a caracteriza modelul de simulare a bobinei DIV cât mai precis, a fost nevoie de măsurări efectuate și pe un banc de încercare unde pompa funcționează în sistem hidraulic pentru a determina influențele non-electrice care conduc la variația inductivității.
11. Pentru rezultate de simulare cât mai precise este important să se introducă efecte temporale în variația inductivității bobinei DIV, modelând variația timpului de închidere/deschidere a sa.
12. Rezultatele comparate arată o potrivire bună a semnalelor atât în timp cât și în amplitudine și formă, totuși potrivirea rezultatelor nu este perfectă necesitând investigații la nivelul modelului de simulare a DAC-ului folosit în blocul de CLCC.
13. Metoda de simulare propusă ca și etapă de început în procesul de dezvoltare a unui ASIC este potrivită pentru a valida și verifica conceptul/blocurile de arhitectură gândite pentru ASIC, cu avantajul că putem utiliza în simulare și modulul din care face parte ASIC-ul, respectiv sarcina comandată, valva DIV.

5.4.2 Contribuții

1. În subcapitolul 5.1.5 am efectuat un număr semnificativ de scenarii de funcționare cu diferite medii de testare și am analizat datele așa cum se arată în lucrarea mea [39].
2. Am propus un model nou de simulare comparativ cu modelul simplu R-L așa cum se arată în [39]. Noul model este capabil să urmărească mai bine forma curentului prin valvă fizică utilizată datorită structurii tabelului de căutare.
3. Am modelat de asemenea comportamentul de saturație al inductanței, sporind performanța generală a simulării.
4. Modelul de simulare electrică a bobinei DIV propus în [39] a fost îmbunătățit prin introducerea efectului temporal pentru a simula închiderea supapei. Închiderea supapei s-a introdus prin modificarea inductivității bobinei în funcție de un semnal sincronizat cu începutul profilului de curent și se manifestă prin modificarea frecvenței de comutație în faza de „hold” atunci când se folosește strategia de CLCC pentru reglarea de curent prin bobină așa cum se arată în subcapitolul 5.1.5.7.
5. Am realizat o **platformă de simulare** conform cu cea propusă în lucrarea mea [34] și explicată în subcapitolul 3.2.3
6. Am realizat o **platformă de validare** conform cu cea propusă în [34] și explicată în subcapitolul 3.1.1 cu excepția implementării etapei în care am propus să se înlocuiască blocul analogic emulat cu componente discrete cu o placă care să conțină Test Chip-ul (Test Chip-ul este un CI care conține doar blocuri analogice a fi implementate în ASIC).
7. Am comparat rezultatele **platformei de simulare** cu rezultatele **platformei de validare**. Comparația arată o potrivire bună a rezultatelor, demonstrând eficiența platformei de validare.

6. Detecția mișcării, reducerea vibrației și diagnoza electrică a valvei DIV

Capitolul 6 tratează îmbunătățirea metodelor de control a solenoidului DIV, reducerea zgomotului produs de vibrațiile supapei valvei DIV la închidere și deschidere și găsirea unei metode de diagnoză electrică mai ieftină pentru circuitul de comandă a solenoidului DIV. Cu ajutorul platformei de simulare și a modelelor dezvoltate în Capitolul 5 am propus și simulat un algoritm de detecție a mișcării supapei valvei DIV; totodată am propus și simulat o metodă de diagnostic și protecție a circuitului de comandă. În final utilizând platforma MTP realizată am testat implementarea celor 3 contribuții originale: algoritmul de reducere a zgomotului, algoritmul de detecție a mișcării și diagnoza/protecția circuitului de comandă a solenoidului DIV.

Dat fiind faptul că sarcina controlată (valva DIV) este un element electro-mecanic cu elemente cinetice, care se mișcă în momentul în care bobina este energizată, apar fenomene de vibrație când părțile care se mișcă își ating poziția/limitele de capăt. Aceste vibrații generează efecte fonice nedorite, de aceea s-a propus o modalitate de frânare a mișcării fără a afecta funcționalitatea normală a valvei și totodată a reduce zgomotul produs de vibrații atunci când valva se închide sau deschide.

Pentru a ne asigura că procedura de reducere a zgomotului nu afectează funcționarea normală a valvei este nevoie să ne asigurăm că valva s-a deschis/închis pentru a asigura cantitatea de combustibil necesar în rampa comună, de aceea am implementat un algoritm care detectează acest lucru. Următoarele două subcapitole vor trata subiectul reducerii zgomotului produs de supapa valvei DIV și a algoritmului de detecție a mișcării supapei.

6.1 Reducerea zgomotului valvei

6.1.1 Statusul existent referitor la reducerea zgomotului DIV

Datorită construcției sale mecanice, supapa valvei DIV generează un "zgomot" atunci când placa de etanșare a supapei la închidere sau pinul supapei la deschidere își poziția lor finală. O strategie de reducere a zgomotului poate fi aplicată pentru turații reduse ale motorului, mai mici de 2000RPM atunci când zgomotul valvei este mai predominant decât zgomotul motorului termic în funcționare.

Există diferite abordări care se ocupă de acest subiect, multe fiind protejate prin brevete ca [43], [44] și [45].

În [43] se poate găsi o abordare interesantă; autorii explică o metodă de reducere a emisiei de zgomot la deschiderea și închiderea supapei prin aplicarea unui algoritm pentru a detecta parametrii optimi pentru profilul de curent (amplitudinea și durata fazei „pull-in”) la deschiderea și închiderea supapei.

Ei încearcă să optimizeze amplitudinea și durata impulsului de închidere, în plus, să aplice un al doilea impuls pentru a încetini mișcarea de deschidere a supapei

DIV pentru a reduce zgomotul atunci când placa ventilului sau pinul ating poziția lor finală.

Pentru a optimiza amplitudinea impulsului de închidere, au implementat un algoritm definit în software, prezentat în Fig. 6.1.

Sunt aplicate consecutiv grupuri de pulsuri (profil de curent cu amplitudine, histereză și durată bine definite) de închidere-deschidere, în fiecare grup amplitudinea pulsului de închidere este redusă până când se observă o diferență în forma pulsului de deschidere ca în grupul "3" din Fig. 6.1. Curentul prin bobina DIV este feedback-ul pentru uC, algoritmul detectează o creștere a curentului mai lentă în pulsul de deschidere (parametrii de control ai pulsului de deschidere nu se modifică de la un grup la altul), deoarece forța magnetică imprimată supapei prin pulsul de închidere nu a fost suficient de mare pentru a închide supapa; atunci când se aplică pulsul de deschidere (cu scopul de a încetini deschiderea), supapa este încă deschisă (are o inductanță mai mare) rezultând o creștere mai lentă a curentului.

În acest moment, autorii algoritmului observă că amplitudinea pulsului de închidere este prea mică pentru a închide cu siguranță supapa, prin urmare se crește amplitudinea cu o marjă de siguranță și se oprește procesul de adaptare rezultând un puls de închidere optim din punct de vedere forța magnetică aplicată – închidere sigură a supapei.

Scopul optimizării este: închiderea sigură a supapei electro-valvei DIV aplicând forța magnetică minimă posibil. Forța magnetică imprimată solenoidului este controlată prin durata și amplitudinea curentului în faza de „pull-in” a profilului de curent aplicat pentru închiderea valvei. Aplicând forța magnetică minimă, viteza imprimată elementelor cinetice va fi minimă, rezultând într-o vibrație mai mică în momentul în care își ating poziția de capăt, creând un nivel mai redus de zgomot.

Procesul de adaptare este reluat la fiecare pornire a motorului sau când condițiile de funcționare (temperatură, tensiunea bateriei, etc.) se modifică.

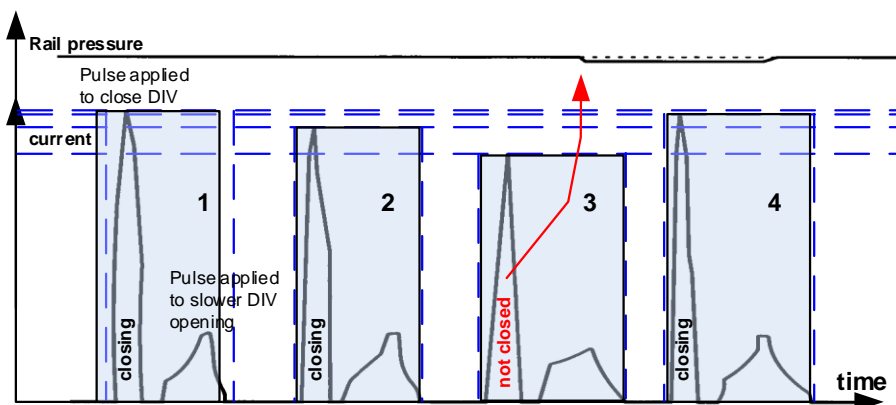


Fig. 6.1 Optimizarea pulsului de închidere pentru reducere a zgomotului [38] (original).

Adaptarea pulsului de deschidere optim, pentru a încetini viteza de deschidere a supapei, este o abordare similară celei pentru adaptarea pulsului de deschiderii a valvei. Amplitudinea pulsului de deschidere va fi crescută gradual până când va avea loc o a doua închidere a supapei în loc de deschidere, așa cum este reprezentat în Fig. 6.2.

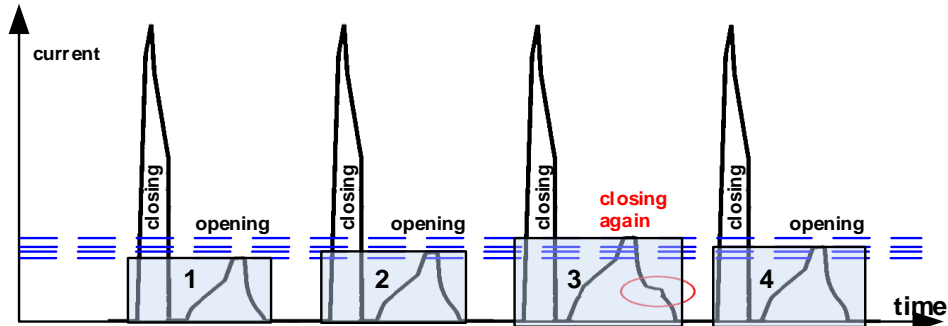


Fig. 6.2 Optimizarea pulsului de deschidere pentru reducere a zgomotului [38] (original).

În cea de-a treia secvență datorită creșterii secvențiale a amplitudinii pulsului de deschidere, în faza de cădere a curentului, este detectată o inflexiune convexă care demonstrează că supapa după o ușoară deschidere se închide din nou.

Metoda prezentată în [43] se bazează pe funcționarea în buclă deschisă (definită în software) a profilului curent: un semnal PWM generat de uC este utilizat pentru a controla curentul prin bobina DIV, dar nu printr-un control hardware în buclă închisă care generează un semnalul PWM în conformitate cu parametrii programați digital (CLCC), ca în lucrarea lui Chia et al. [44].

În [43] ADC-ul microcontrolerului este folosit pentru a achiziționa un semnal analogic proporțional cu curentul prin bobina DIV, astfel că uC-ul cunoaște întotdeauna valoarea curentului și poate adapta frecvența de comutație a tranzistorului HS, însă această informație este utilizată numai pentru a controla amplitudinile pulsului de deschidere și închidere a supapei DIV. Frecvența de comutație nu poate fi schimbată pe durata în care pulsul este activ, fiind un dezavantaj față de [44], unde curentul DIV este controlat foarte precis.

6.1.2 Modalitatea originală propusă de reducere a zgomotului DIV

Strategia noastră de reducere a zgomotului este prezentată în Fig. 6.3 și are un rol de amortizare a vibrațiilor produse prin încetinirea mișcării supapei DIV fără a interfera cu funcționarea normală a pompei/valvei.

Controlul curentului se bazează pe strategia de CLCC care asigură o valoare precisă a curentului cu parametrii programabili individual pentru faza de „pull-in” și „hold” pentru amplitudine, histereză și durată așa cum este reprezentat în Fig. 6.3, controlat exclusiv prin circuitul hardware: ASIC-ul sau, în prototipul nostru de test, printr-un FPGA și circuitele electronice discrete corespunzătoare necesare pentru emularea blocului CLCC.

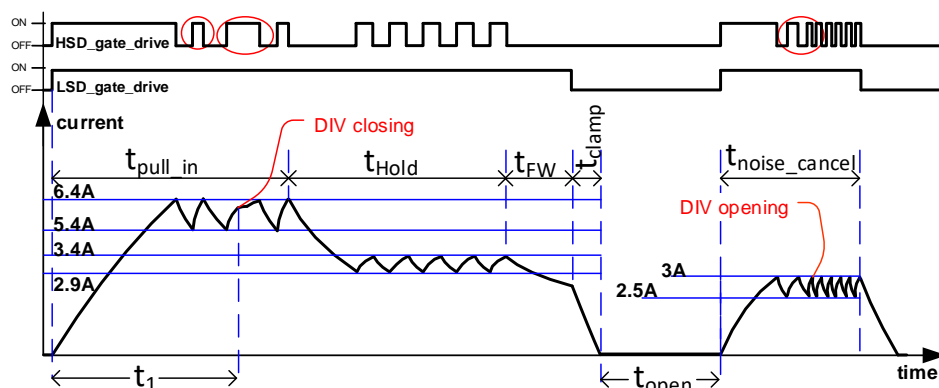


Fig. 6.3 Particularitățile controlului valvei DIV [38] (original).

Pentru a reduce emisia de zgomot la închiderea supapei DIV, curentul din faza de „pull-in” și durata $t_{\text{pull_in}}$ trebuie adaptate corespunzător:

- Curentul de pull-in trebuie să fie suficient de mare pentru a imprima destulă forță solenoidului valvei astfel încât să inițieze mișcarea supapei, dar să nu rezulte un impact puternic între discul de etanșare a supapei și flanșa de etanșare (fixă, care face parte din corpul valvei)
- Durata fazei „pull-in” trebuie să fie suficient de lungă pentru a asigura o închidere corespunzătoare a supapei, dar totodată cât mai scurtă posibil și oprită imediat după ce s-a detectat deplasarea supapei. Mișcarea plăcii valvei este detectată prin algoritmul implementat în blocul digital al ASIC-ului (în cazul nostru în FPGA)
- Mișcarea supapei DIV se detectează atunci când apare o „gâlmă” de curent în faza de „hold” datorită creșterii inductanței supapei rezultând într-o durată mai lungă de ON a tranzistorului HS în faza „hold”, așa cum se vede pe semnalul „HS_gate_drive” din Fig. 6.3 când apare închiderea valvei
- Pentru a reduce emisia de zgomot la deschiderea supapei (după ce se oprește curentul=sfârșitul primului puls), un al doilea puls de curent tot cu CLCC poate fi aplicat după un timp de t_{open} față de faza de „clamping”
- Pentru efect maxim, t_{open} trebuie să fie controlat cu precizie în funcție de parametrii sistemului: turația motorului, tensiunea bateriei, temperatura bobinei DIV și presiunea din rampa comună
- Acest al doilea puls este similar cu primul puls CLCC, dar acum obiectivul este de a deschide controlat-frânat supapa valvei DIV
- Faza de „pull-in” pentru al doilea puls trebuie să fie suficient de lungă pentru a încetini mișcarea discului de etanșare a supapei, dar totuși destul de scurtă pentru a nu risca închiderea supapei încă o dată
- Faza de „hold” trebuie să fie suficient de scurtă pentru a nu risca o a doua închidere a supapei
- Pentru a asigura o funcție adecvată de reducere a zgomotului, este necesar un algoritm adecvat de adaptare bazat pe turația motorului, tensiunea bateriei și punctul de presiune stabilit.

6.1.3 Rezultatele obținute

Pentru a aplica metoda de reducere a zgomotului propusă anterior, este important să se detecteze evenimentul de închidere și de deschidere supapei DIV.

Ne bazăm pe faptul că bobina DIV are trei regiuni diferite de inductanță, așa cum se arată în Fig. 6.4:

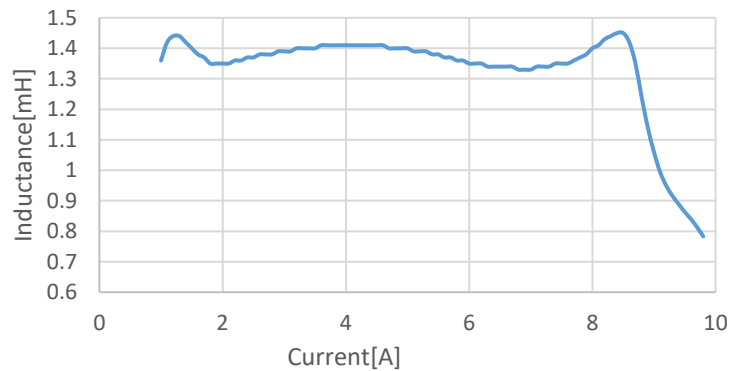


Fig. 6.4 Variația inductivității bobinei DIV [38] (original).

- În stare deschisă, inductanța este stabilă (aproximativ 1,5mH)
- În timpul procesului de închidere și imediat după închidere este în creștere (modificarea inductanței poate fi ignorată)
- În stare închisă, bobina se saturează și are valoare scăzută a inductanței (aproximativ 0,5mH).

Fig. 6.5 arată cum influențează schimbarea inductanței frecvența de comutare în cazul unui profil CLCC. Hardware-ul controlează tranzistorul HS pentru a menține curentul între două praguri: 6A și 5A (cu histerezis de 1A) în faza de vârf.

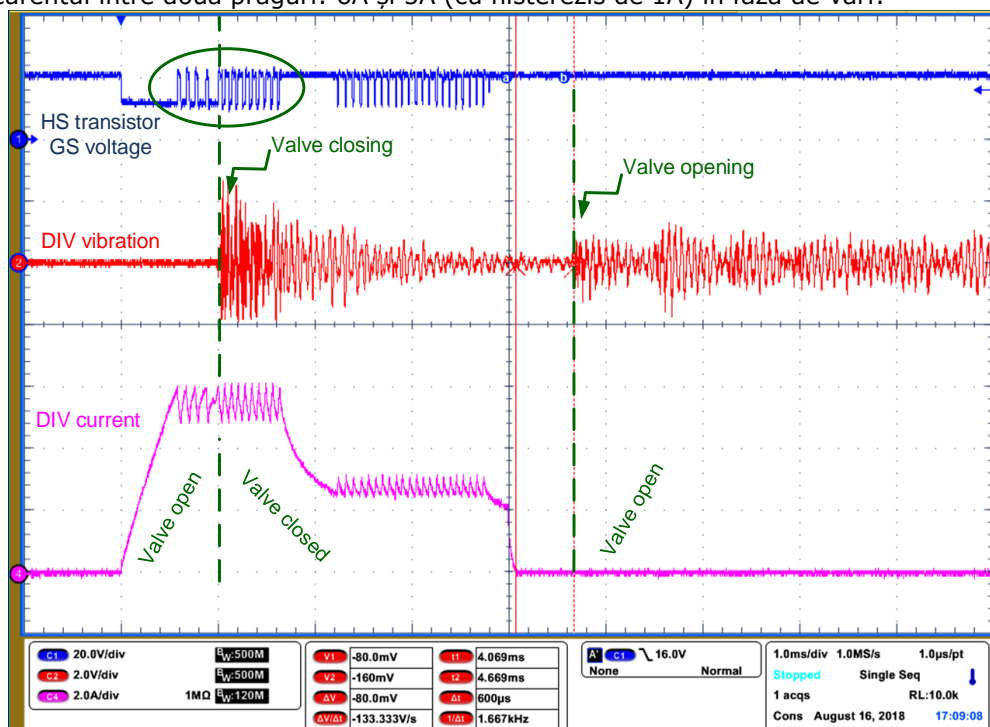


Fig. 6.5 Închiderea valvei DIV, alimentare 14V, 25C, fără presiune de combustibil [38] (original).

La începutul profilului de curent, când supapa este deschisă (inductanță mare), rezultă o frecvență inițială de comutare datorată celor două praguri de curent definite. Când forța magnetică este suficient de mare, supapa se închide, schimbându-se astfel inductivitatea solenoidului (inductanță redusă), rezultând o frecvență de comutare mai mare pentru a menține curentul între pragurile definite. Același fenomen este folosit și la pulsul de deschidere pentru a detecta starea deschisă a supapei DIV, ca în Fig. 6.6.

La sfârșitul pulsului de închidere, curentul prin solenoid este redus la zero, astfel încât inductanța bobinei începe să revină la starea inițială, chiar dacă supapa este încă închisă. Când impulsul de deschidere este aplicat pentru a încetini mișcarea de deschidere a supapei, mai întâi rezultă o comutație lentă, apoi una rapidă pentru menținerea curentului între pragurile definite în faza de „hold”: limita superioară 3A; limita inferioară 2,5A (cu o histereză de 0,5A). Prin eșantionarea cu un pin de intrare digital al uC-ului a semnalului de comandă al porții tranzistorului HS, este posibil să se detecteze atât evenimentele de închidere, cât și cele de deschidere ale supapei DIV. Un avantaj suplimentar este că nu este necesară o intrare de ADC a uC-ului, ca în cazul metodei propuse de mine în [43].

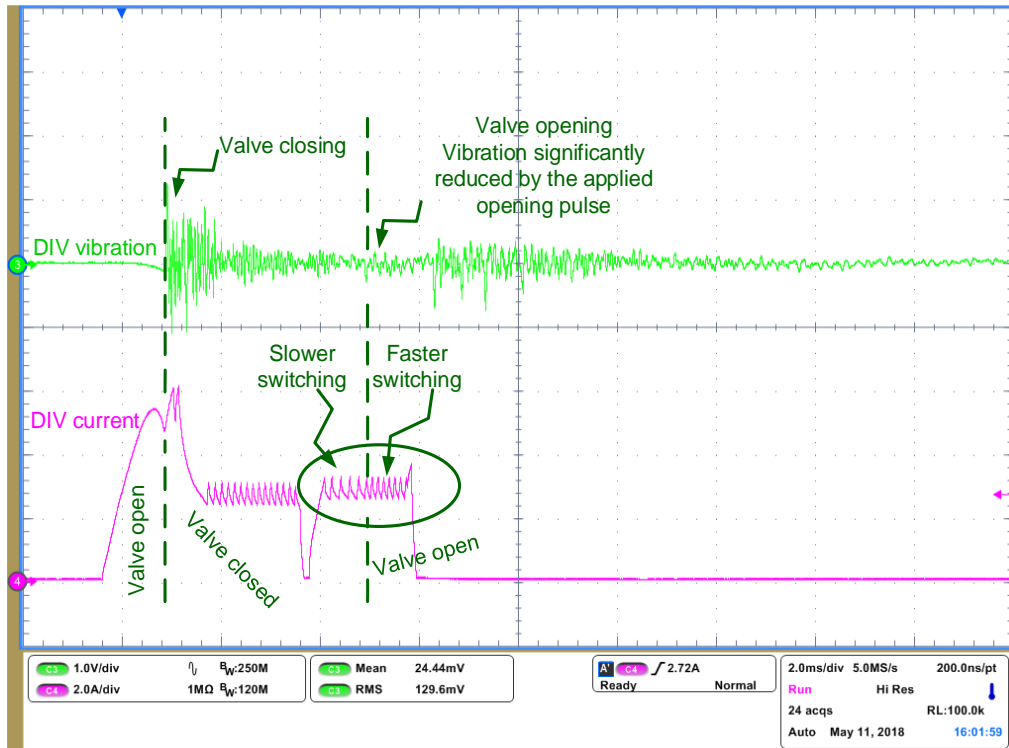


Fig. 6.6 Închiderea și deschiderea valvei DIV, alimentare 14V, 25C, fără presiune de combustibil [38] (original).

Rezultatele finale referitor la reducerea zgomotului generat de valva DIV sunt prezentate în Fig. 6.7.

Începând cu jumătatea de jos a oscilogramii, avem 2 semnale salvate ca referință ale cazului de control când nu aplicăm metoda propusă de reducere a zgomotului.

1. „Fără reducere de zgomot”

Putem observa forma profilului de curent „DIV_current_1” aplicat solenoidului DIV, curentul urcă până la o valoare de vârf de 6A, înainte să ajungă la valoarea de vârf, valva se închide (în faza de pull-in), moment marcat de apariția vibrației „DIV_vibration_1” când supapa își atinge poziția finală. Vibrația are o amplitudine maximă de aproximativ 2V. Apoi curentul scade în zona de menținere (hold) la aproximativ 3A unde este menținut cu o histereză de 0.4A, după care profilul de curent este oprit. La un timp după ce curentul ajunge în zero, apare cea de-a doua vibrație cu o amplitudine de aproximativ 1V, indicând deschiderea supapei.

2. „Cu reducere de zgomot”

În jumătatea de sus a oscilogramii avem profilul de curent „DIV_current_2”, unde acum avem 2 pulsuri de curent, primul pentru închiderea valvei, cel de al doilea pentru frânarea deschiderii valvei. Primul puls este diferit față de cazul când nu se aplică reducerea de zgomot. În acest caz s-a aplicat un semnal PWM, iar acum curentul nu crește doar cu limitare de inductivitate a bobinei, ci se aplică o încetinire adițională pentru a frâna închiderea supapei. Se observă că închiderea

valvei se produce în faza de reținere (hold) cu o întârziere, moment în care apare pulsul de vibrație „DIV_vibration_2” cu o amplitudine redusă față de cazul anterior de numai 1V, ceea ce indică efectul de reducere a zgomotului la închiderea supapei.

Curentul este menținut în faza de „hold” tot la 3A cu histereză de 0.4A până la sfârșitul profilului de curent.

Cea de a doua diferență față de cazul când nu se aplică reducere de zgomot este că acum la un interval de timp după ce se termină pulsul de închidere a valvei, în loc să lăsăm valva să se deschidă natural, o frânăm aplicând un al doilea puls de curent, motiv pentru care pe semnalul „DIV_vibration_2” al doilea puls de vibrații apare cu amplitudine neglijabilă subliniind eficiența pulsului de curent pentru frânarea deschiderii supapei DIV.

Măsurarea de vibrații cu senzorul piezo atașat pe corpul pompei este doar o măsurătoare relativă, care se pot folosi doar ca și comparație între două măsurători făcute în aceleași condiții de test, practic contează doar diferența dintre cele 2 măsurători ca valoare absolută.

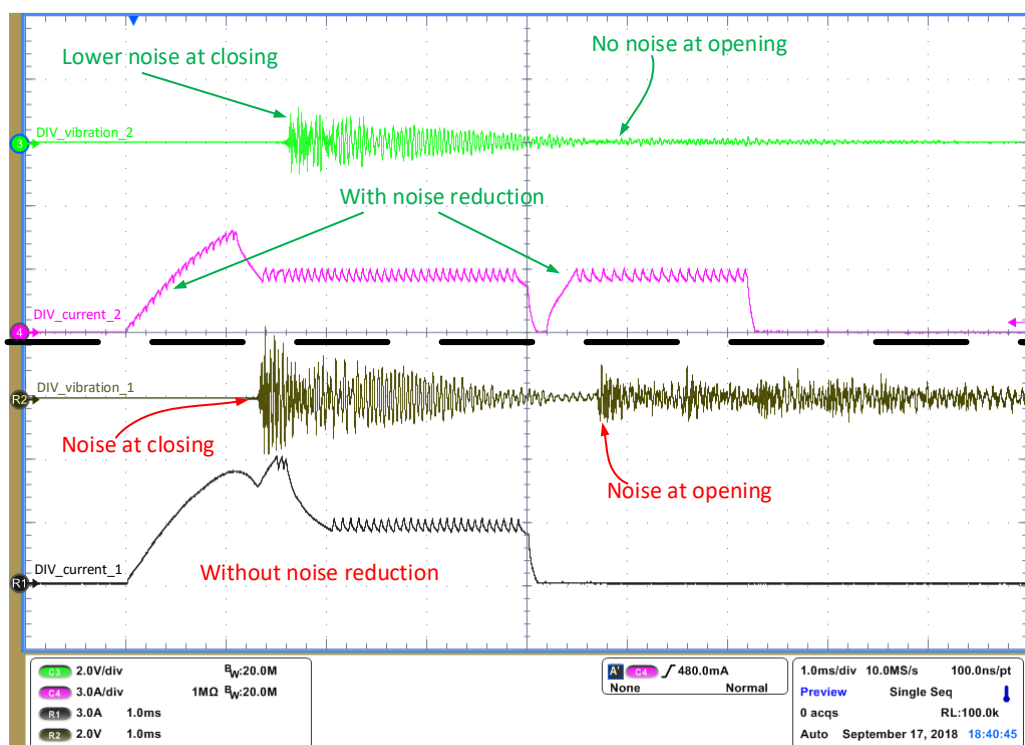


Fig. 6.7 Zgomotul creat de valva DIV; cu reducere de zgomot; fără reducere de zgomot [38] (original).

6.1.4 Probleme întâmpinate și rezolvate în timpul dezvoltării procedurii de reducere a zgomotului

6.1.4.1 Rezolvarea problemei mișcării precoce

Pentru a putea detecta închiderea/deschiderea valvei DIV, mișcarea supapei DIV trebuie să se producă în faza de control de curent (de comparat pulsul de curent de închidere din Fig. 6.5 și Fig. 6.6). În Fig. 6.5 închiderea valvei DIV se întâmplă în zona de reglare de curent din faza „pull-in”.

Dacă mișcarea propriu zisă a supapei are loc în timpul fazei „pull-in”, dar în zona de început de creștere a curentului, mișcarea nu poate fi detectată, ca în Fig. 6.6: evenimentul de închidere a supapei se întâmplă chiar la sfârșitul fazei de „pull-in”, astfel încât pentru pulsul de închidere nu se poate detecta o tranziție de la comutarea cu frecvență mică la comutare cu frecvență mai mare a tranzistorului HS.

Pentru a rezolva problema din Fig. 6.6, trebuie să încetinim mișcarea discului de etanșare a supapei prin scurtarea fazei de „pull-in” și sărind direct la a doua fază de control de curent astfel încât mișcarea supapei să se producă în această fază ca în Fig. 6.8.

Faza de „pull-in” este utilizată pentru a energiza bobina și trebuie terminată înainte ca discul de etanșare să-și atingă poziția finală. A doua fază de reglare de curent este utilizată pentru a menține supapa închisă până când presiunea din pompa HPFP crește suficient de mult și ține valva în mod natural închisă. Din păcate, soluția propusă nu este stabilă dacă se modifică condițiile ambientale (temperatura bobinei, presiunea din rampa comună sau tensiunea de alimentare).

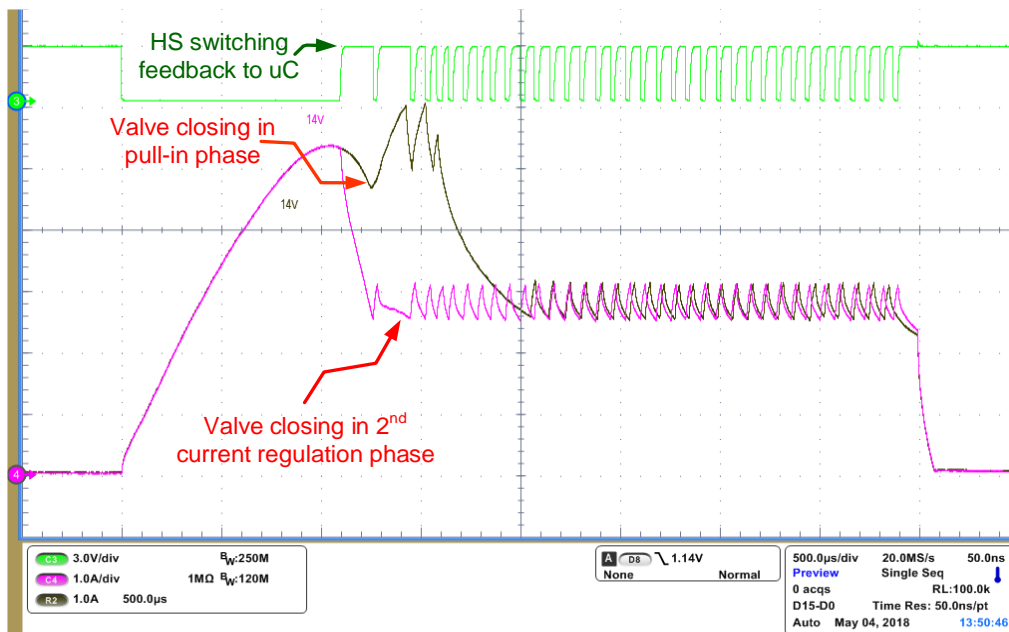


Fig. 6.8 Închiderea valvei DIV, alimentare 14V, 25C, fără presiune în rampa comună [38] (original).

6.1.4.2 Rezolvarea problemei variației cu temperatura și tensiunea de alimentare a timpului natural de închidere al valvei

Pentru o funcție adecvată de reducere a zgomotului, trebuie să rezolvăm problema de la 6.1.4.1 care depinde de condițiile ambientale. Am realizat un set de măsurări pentru a determina timpul de închidere / deschidere naturală față de variația temperaturii și a tensiunii de alimentare pentru o pompă HPFP cu valvă DIV utilizată în sistemele GDI actuale de 250 bari.

Rezultatele din Fig. 6.9 arată că timpul de închidere naturală este puternic dependent de temperatură, totuși cu creșterea tensiunii de alimentare până la 19V, acest efect este anulat probabil de forța magnetică puternică creată datorită tensiunii de alimentare ridicate.

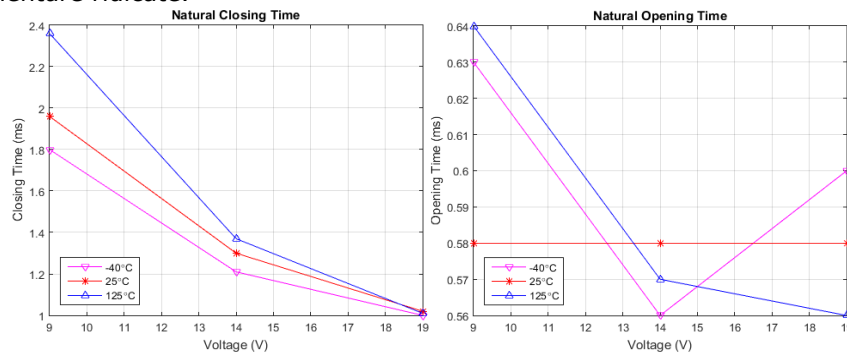


Fig. 6.9 Timpul de închidere/deschidere natural, fără presiune de rampă [38] (original).

Pe de altă parte, rezultatele din Fig. 6.9 arată că timpul de deschidere naturală nu este predictibil, prin urmare este necesară o calibrare înainte de aplicarea pulsului de încetinire a deschiderii supapei pentru a cunoaște întârzierea exactă dintre pulsul de închidere și deschiderea naturală a valvei, parametrul t_{open} marcat în Fig. 6.3.

Pentru a controla timpul natural de închidere, se aplică o tehnică de "aterizare controlată [45]" prin oprirea tranzistorului HS în faza de creștere a curentului în „pull-in” cu o perioadă și durată de OFF definită pentru a controla panta de creștere a curentului până când atinge valoarea maximă (peak current).

Prin această modalitate putem avea aceeași pantă de curent independent de tensiunea bateriei, controlând astfel forța magnetică aplicată solenoidului DIV, controlând astfel timpul natural de închidere, așa cum se poate vedea în Fig. 6.10.

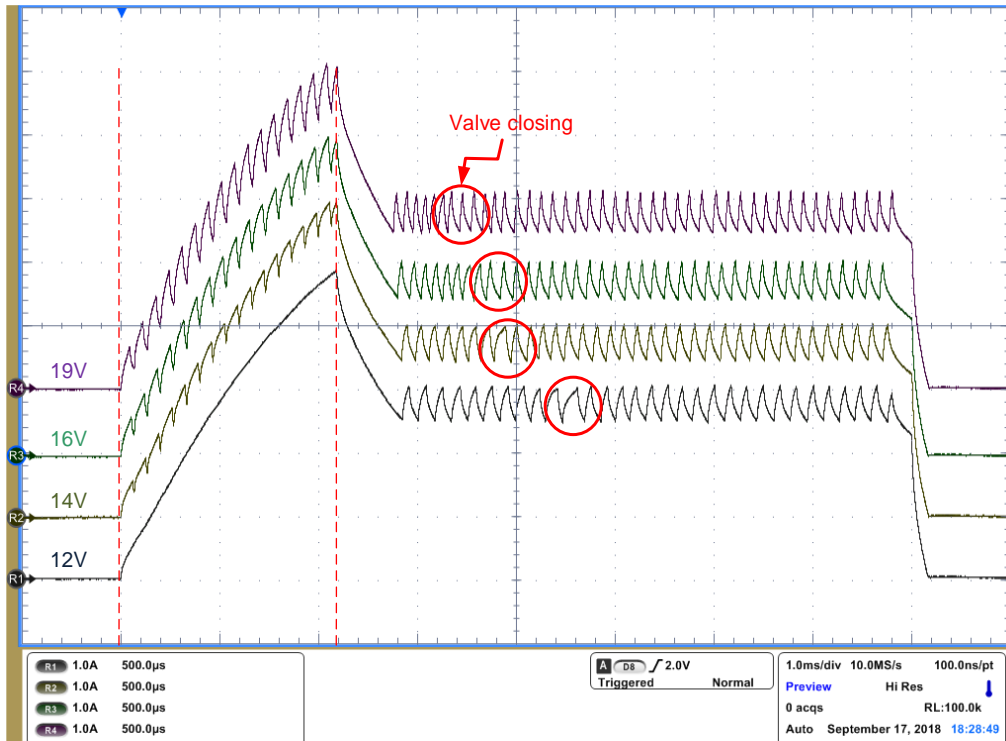


Fig. 6.10 Controlul timpului natural de închidere, fără presiune de combustibil [38] (original).

6.2 Algoritm de detecție a mișcării supapei

Simularea a avut un rol foarte important în dezvoltarea algoritmului de detectare a mișcării. În Fig. 6.11 este prezentată diagrama de stare a algoritmului de detectare a mișcării supapei din valva DIV.

Algoritm este implementat numai pentru pulsul de închidere a valvei DIV și pornește de îndată ce mașina de stare (implementată în VHDL) care controlează blocul CLCC sare de la faza "idle" (inactiv) la "peak" (vârf). Toate contoarele și variabilele utilizate pentru implementarea algoritmului sunt inițializate în acest moment. Algoritm așteaptă până când mașina de stare va sări de la faza "peak" la "hold" (menținere). De îndată ce faza "hold" este activă, există două contoare concurente, unul măsurând durata de ON a tranzistorului HS iar celălalt durata de OFF.

Scopul algoritmului este de a detecta în faza "hold" schimbarea de la frecvență de comutare lentă la mai rapidă a tranzistorului HS (pentru a menține aceleași limite de curent programate), cauzată de scăderea inductanței după închiderea valvei.

Datorită profilului de control specific, când curentul scade de la valoarea de „peak” până la curentul de „hold” prima durată de OFF a tranzistorului HS este ignorată, deoarece ar rezulta o valoare maximă prea mare comparativ cu restul duratelor de OFF din faza de „hold”.

Pornind de la cel de-al doilea eveniment de OFF și primul eveniment de ON al tranzistorului HS, algoritmul măsoară constant durata de OFF și ON a tranzistorului HS.

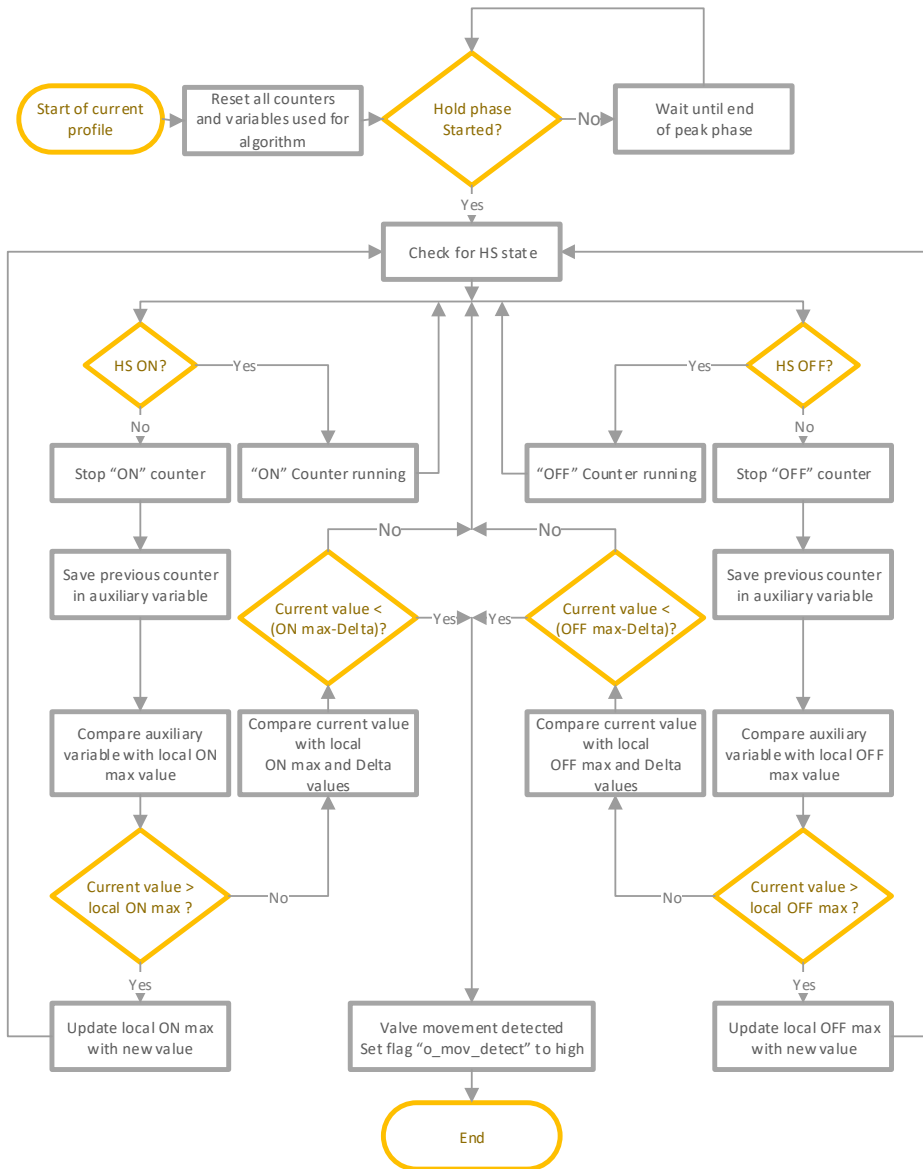


Fig. 6.11 Algoritmul de detecție a mișcării supapei: diagrama de stări (original).

Pentru a evita detectarea falsă a mișcării supapei datorită unei diferențe prea mici între două măsurări consecutive, este necesară o "valoare delta", care este predefinită într-un registru al ASIC-ului.

De îndată ce tranzistorul HS este controlat ON, starea anterioară de OFF este transferată într-un registru temporar și apoi este comparată cu valoarea "maximum time OFF" (valoarea maximă OFF este zero la începutul algoritmului), valoare salvată într-un registru. Dacă valoarea curentă este mai mare decât valoarea maximă deja salvată, va fi transferată în registrul "maximum time OFF". Dacă valoarea curentă este mai mică decât maximul deja stocat, algoritmul trece la următoarea fază de comparație când se efectuează o verificare logică în comparație cu o valoare "delta" prestabilită și stocată într-un alt registru.

Dacă diferența dintre valoarea curentă și cea maximă este mai mare decât valoarea „delta” memorată, înseamnă că ultimul timp de OFF măsurat a fost mai scurt decât toate cele măsurate anterior.

Timpul de OFF mai scurt indică modificarea inductanței datorată închiderii supapei, astfel blocul CLCC crește frecvența de comutare ca să mențină curentul între aceleași limite definite de uC la programarea ASIC-ului (FPGA în cazul nostru). În acest caz următoarea acțiune este ca algoritmul de detecție a mișcării să seteze pinul de ieșire al ASIC-ului dedicat pentru această funcționalitate pe „1” logic, adică "mișcare detectată".

În caz contrar, dacă diferența nu este mai mare decât „delta” memorată, înseamnă că ultimul timp de măsurare OFF este mai mic decât maximul cunoscut, dar nu suficient de scurt pentru a declara detectarea mișcării supapei, iar algoritmul așteaptă următoarea măsurare de timp de OFF.

Aceeași procedură este valabilă și pentru măsurarea timpului de ON al tranzistorului HS, implică numărătoare și registre separate, fiind independente de măsurarea de timp de OFF.

Ambele procese funcționează în paralel, iar cel care detectează mai întâi diferența mai mare decât „delta” predefinită dintre ultima măsurare și maximul stocat (unul pentru timpul de OFF și altul pentru timpul de ON) are prioritate de a semnaliza "mișcare detectată".

Este necesară măsurarea timpului de ON și OFF deoarece momentul închiderii supapei este dependent de tensiunea de alimentare, presiunea combustibilului și temperatura solenoidului, motiv pentru care nu poate fi anticipat dacă se întâmplă în timpul fazei ON sau OFF a tranzistorului HS, dar efectul este același, se scurtează semnificativ durata următoare de ON sau OFF.

În Fig. 6.12 este prezentat un scenariu tipic de simulare folosit pentru a evalua funcționarea algoritmului de detecție a mișcării. În partea superioară a graficului sunt semnalele și registrele VHDL. Registrele multi-bit sunt grupate în format de magistrală (bus). Semnalul "valve_det_high_check_hs" este setat pe "high", deoarece mașina de stare de control trece din faza "peak" la "hold" arătată de schimbarea registrului "phase"(vezi Fig. 6.12).

6.2 – Algoritm de detecție a mișcării supapei 107

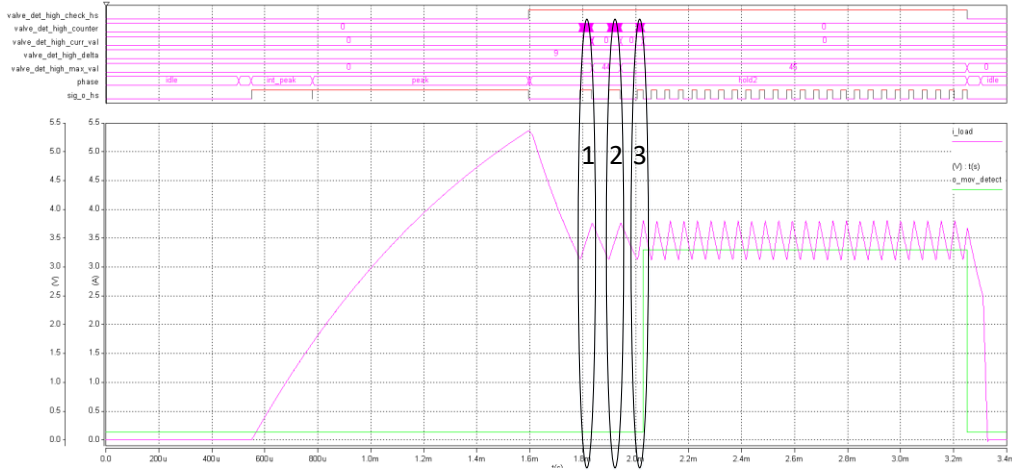


Fig. 6.12 Scenariu de simulare: algoritm de detecție a mișcării (original).

De îndată ce primul eveniment de ON începe (în faza „hold”) indicat prin semnalul "sig_o_hs", numărătorul începe să numere așa cum este arătat de tranzițiile registrului "valve_det_high_counter". După terminarea timpului de ON, valoarea contorului este transferată în registrul temporar "valve_det_cur_val" imediat după aceea, acest registru este comparat cu "valve_det_high_max_val" care este zero, motiv pentru care se stochează astfel "valve_det_cur_val" în registrul care trebuie să stocheze durata maximă de ON "valve_det_high_max_val", valoarea zecimală "44" corespunzătoare duratei de 44 μs, deoarece numărătorul are un tact de 1 MHz derivat din tactul principal de 100 MHz. Acest lucru este mai bine vizibil în Fig. 6.13, care este un detaliu al zonei marcate cu "1" din Fig. 6.12.

Săgețile din Fig. 6.13 arată transferul de la numărător către registrul auxiliar și apoi la cel care reține durata maximă a primei măsurări a duratei de ON. Pe măsură ce durata de ON s-a terminat, algoritmul măsoară deja durata de OFF folosind celălalt numărător și procedând la fel ca în cazul timpului de ON.

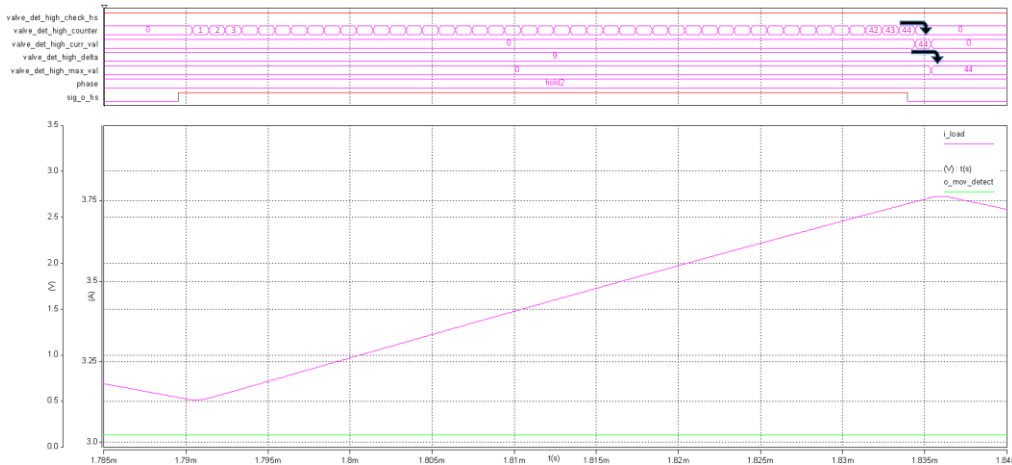


Fig. 6.13 Detaliu al zonei "1" din Fig. 6.12 (original).

6.2 – Algoritmul de detecție a mișcării supapei 108

În Fig. 6.14 apare o mărire a zonei marcate cu "2" din Fig. 6.12. Aceasta prezintă un detaliu al celei de-a doua măsurări de durată ON. Este vizibil faptul că numărătorul se oprește când atinge valoarea zecimală 45, indicând un timp de ON de 45us care este transferat în registrul auxiliar. Deoarece timpul de ON precedent a fost 44us, acum registrul "valve_det_high_max_val" este actualizat cu valoarea zecimală 45. Algoritmul începe acum aceeași măsurare pentru timpul de OFF și pentru următorul timp de ON al tranzistorului HS.

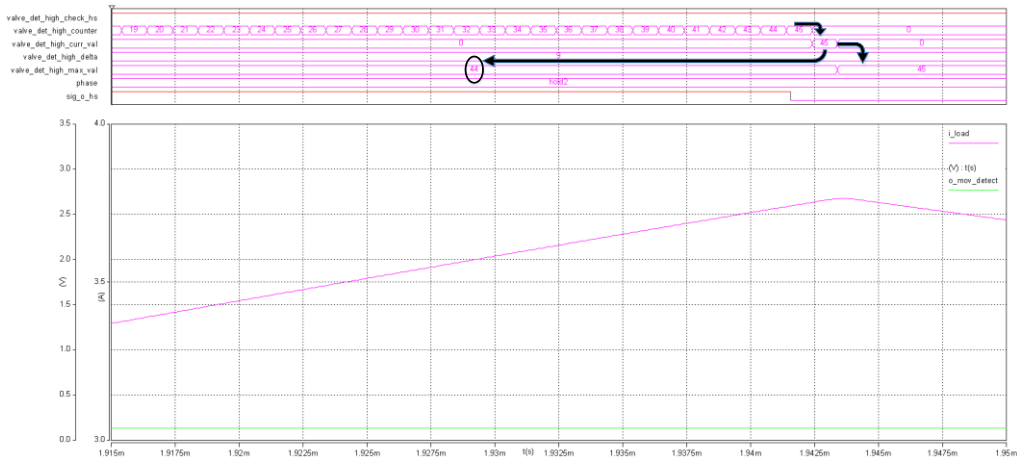


Fig. 6.14 Detaliu al zonei "2" din Fig. 6.12 (original).

În Fig. 6.15 este prezentată aria marcată cu "3" din Fig. 6.12. Aici apare procesul de măsurare pentru al treilea timp de ON; acum numărătorul se oprește la valoarea zecimală 22 indicând un timp de ON de 22us.

Algoritmul compară această valoare cu valoarea maximă stocată, care în acest caz este mai mică, sărind astfel la cealaltă ramură unde se compară diferența dintre valoarea maximă (max = 45) și măsurarea curentă (curr = 22) cu delta predefinit (delta = 9) .

De aici rezultă $45-22=23$, care este mai mare decât „delta”, stabilind astfel semnalul "o_mov_detect" pe "high" indicând că a fost detectată închiderea valvei DIV.

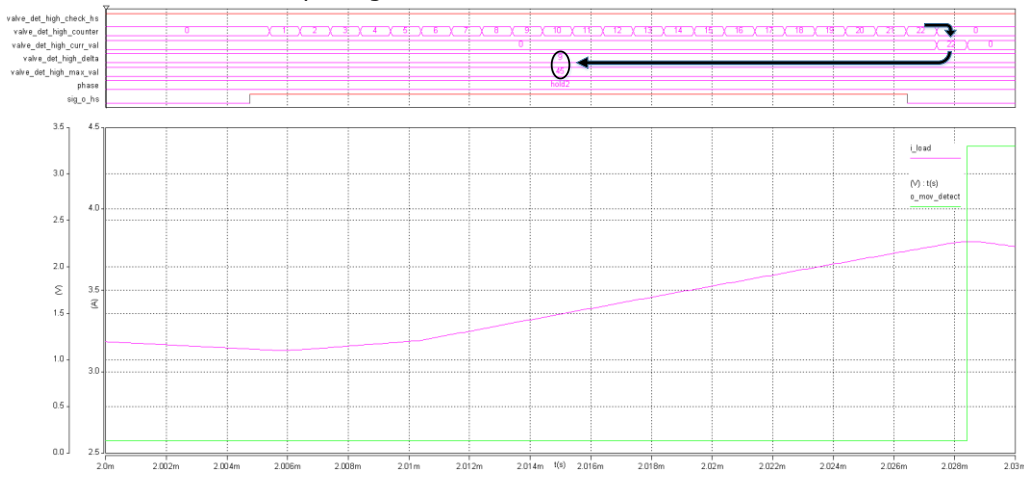


Fig. 6.15 Detaliu al zonei "3" din Fig. 6.12 (original).

Analiza de mai sus ar fi aproape imposibilă fără o simulare adecvată. Avantajul unei astfel de simulare de semnal mixt este flexibilitatea și ușurința de vizualizarea a registrelor și a semnalelor comparativ cu un circuit real cu un ASIC sau FPGA în care nu este posibilă vizualizarea în timp real a registrelor interne.

6.3 Diagnoza electrică; protecția circuitului de comandă a solenoidului DIV

Diagnoza electrică a circuitului de comandă este un subiect foarte important în unitățile electronice de control pentru automobile. Are două roluri, cel mai important fiind rolul de siguranță, acționează ca protecție și nu permite distrugerea componentelor electronice atunci când evenimentul de supra-curent se întâmplă pe tranzistorul LS sau HS. Al doilea rol este acela de a informa sistemul despre posibila funcționare defectuoasă a sarcinii comandate, de exemplu în cazul în care există un scurtcircuit la baterie pe ieșirea tranzistorului HS, bobina DIV-ului va fi în mod constant alimentată (nu funcționează corect). În acest caz, sistemul ar trebui să detecteze defecțiunea și să dezactiveze ieșirea LS pentru a opri calea de curent prin bobina DIV-ului. Aceeași acțiune este de așteptat în cazul unui scurtcircuit la potențialul de referință la ieșirea tranzistorului HS.

În cazul în care sarcina nu este conectată, nu este necesară nicio acțiune de dezactivare a tranzistoarelor, dar sistemul ar trebui să fie informat despre faptul că solenoidul DIV-ului nu este conectat sau este întrerupt (rezistență foarte mare), astfel că pompa HPFP nu va putea genera presiunea necesară a combustibilului în rampa comună.

6.3.1 Metoda actuala de diagnoză

În Fig. 6.16 este prezentată o arhitectură tipică de ASIC care are implementate funcțiile de diagnoză explicate mai sus.

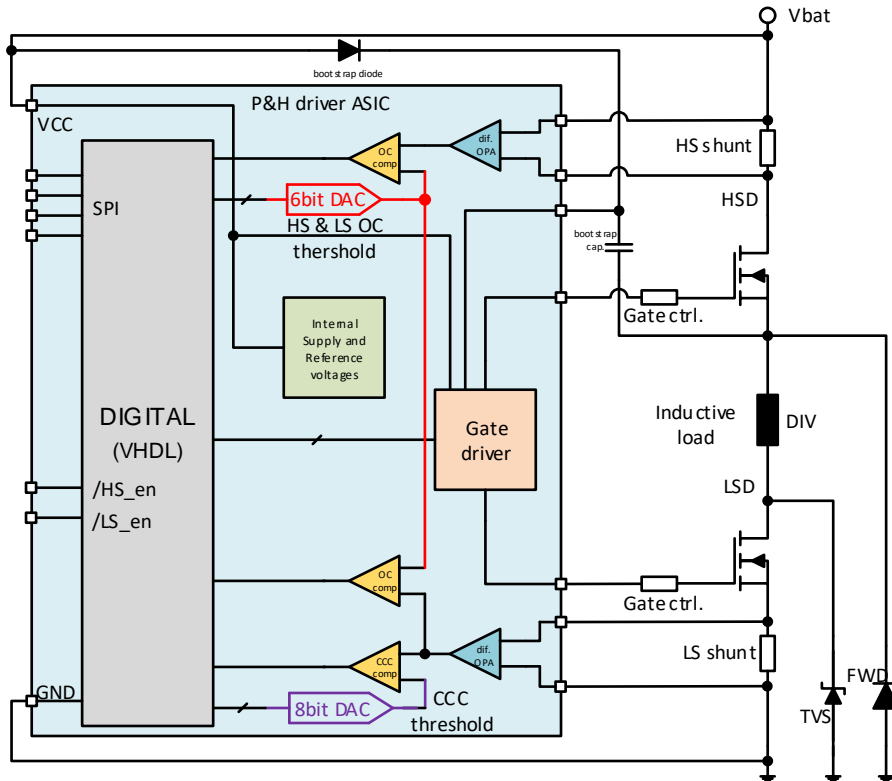


Fig. 6.16 Arhitectură tipică de ASIC (original).

Pentru detectarea de supra-curent, căderea de tensiune pe șuntul de LS și HS este monitorizată cu amplificatoare diferențiale, a căror ieșiri sunt conectate la o intrare a "OC_comp". La cealaltă intrare a comparatoarelor de OC este adus pragul de tensiune generat de un convertor digital-analog pe 6 biți. DAC-ul este utilizat pentru a avea flexibilitatea unui prag programabil. În cazul unui eveniment de supra-curent, tensiunea de pe șuntul LS sau HS va fi mai mare decât pragul de siguranță configurat de blocul digital al ASIC-ului, programat în prealabil prin intermediul comunicării SPI de către uC. Comparatoarele de OC vor semnala acest lucru blocului digital, astfel încât acesta să poată lua măsurile corespunzătoare, pentru a bloca tranzistoarele LS și HS și a trimite codul de diagnostic corespunzător la uC prin intermediul SPI în funcție de eroarea detectată.

Pe de altă parte, atunci când solenoidul valvei DIV nu este conectat la ieșirea LS ori HS sau la niciuna dintre ele, nu va exista curent prin nici unul dintre șunturile LS și HS (nici prin tranzistoare) atunci când este programat un profil de curent, astfel nu va exista cădere de tensiune pe șunturi. Acest lucru este semnalat către blocul digital; astfel se poate trimite către uC prin SPI informația de diagnostic "sarcină neconectată".

În cele ce urmează se prezintă trei scenarii de funcționare:

1. Funcționare normală
2. Scurt-circuit între ieșirea LS și linia de alimentare (borna + a bateriei)
3. Solenoidul DIV neconectat

1. În Fig. 6.17 se prezintă un scenariu normal de funcționare pentru arhitectura din Fig. 6.16.

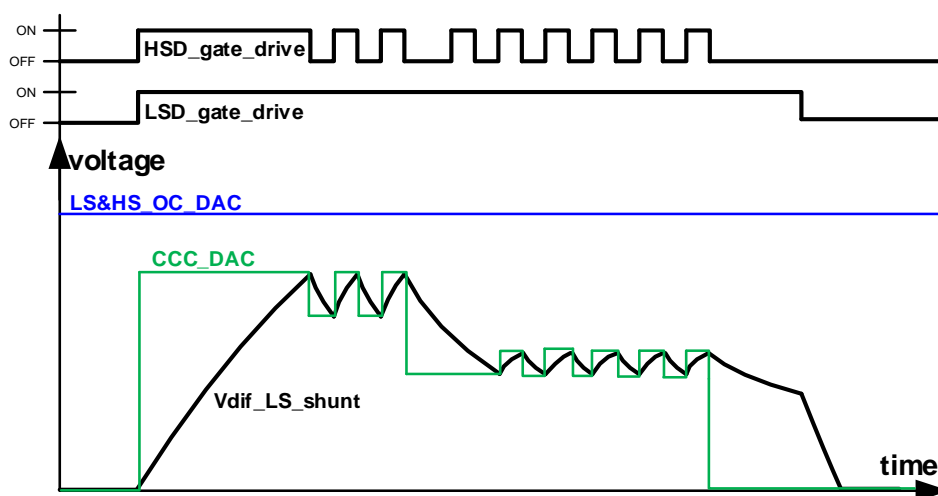


Fig. 6.17 Arhitectura cu două DAC-uri: funcționare normală (original).

Solenoidul DIV este controlat de un profil de curent „peak and hold” în buclă închisă. Există un al doilea convertor digital-analog cu rezoluție pe 8 biți, care prescrie pragurile necesare pentru a realiza profilul de curent în buclă închisă, așa cum arată semnalul "CCC_DAC". Profilul de curent este programat prin SPI de către uC și stocat în registrele ASIC-ului. Un comparator "CCC comp", care la o intrare are semnalul "CCC_DAC", iar pe cealaltă intrare are căderea de tensiune a șuntului LS măsurată

diferențial, așa cum arată semnalul "Vdif_LS_shunt". Tranzistorul LS servește întotdeauna ca o cale de activare a curentului prin bobina DIV, este activat în timpul întregului profil de curent, așa cum arată semnalul "LSD_gate_drive". Când tensiunea de pe șunt este sub pragul programat, tranzistorul HS este controlat de blocul digital pentru a porni (ON). Când tensiunea este mai mare decât cea dorită, tranzistorul HS este comandat OFF până când curentul scade sub valoarea histerezei minime dorită și definită prin schimbarea dinamică a semnalul "CCC_DAC".

DAC-ul pe 6 biți pentru protecția de supra-curent (OC) este întotdeauna setat la o valoare mai mare decât curentul de funcționare normală, astfel că nu se detectează eroare falsă atunci când curentul este sub acest prag, așa cum arată semnalul "LS & HS_OC_DAC" (căderea de tensiune pe șuntul de LS sau HS împărțită cu valoarea rezistenței șuntului dă informația măririi curentului prin bobina DIV). Atunci când, fie căderea de tensiune pe șuntul de LS, fie pe cel de HS, depășește pragul de OC setat, blocul digital este în măsură să acționeze corespunzător oprind cele două tranzistoare.

2. În Fig. 6.18 se prezintă un scenariu de eroare; supra-curent pe tranzistorul LS.

Această defecțiune poate fi cauzată datorită unui scurtcircuit la linia de baterie(+) sau înfășurarea bobinei însăși are partea rezistivă de valoare redusă datorită scurt-circuitului între spirele bobinei din cauza supraîncălzirii sau a altor tipuri de deteriorări mecanice/fizice.

Când există un scurtcircuit, panta ascendentă a curentului este mult mai rapidă, curentul atinge pragul "CCC_DAC", prin urmare tranzistorul HS este oprit și "CCC_DAC" este setat la valoarea curentului de histereză (până aici este ca în funcționarea normală). Însă curentul crește (prin tranzistorul LS) încă datorită căii de scurt-circuit la linia (borna + al bateriei) și pragul de siguranță "LS & HS_OC_DAC" este atins. În acest moment, tranzistorul LS este oprit și curentul începe să scadă. După oprirea LS, semnalul "CCC_DAC" este setat la zero așa cum se întâmplă și la sfârșitul profilului de curent în funcționarea normală.

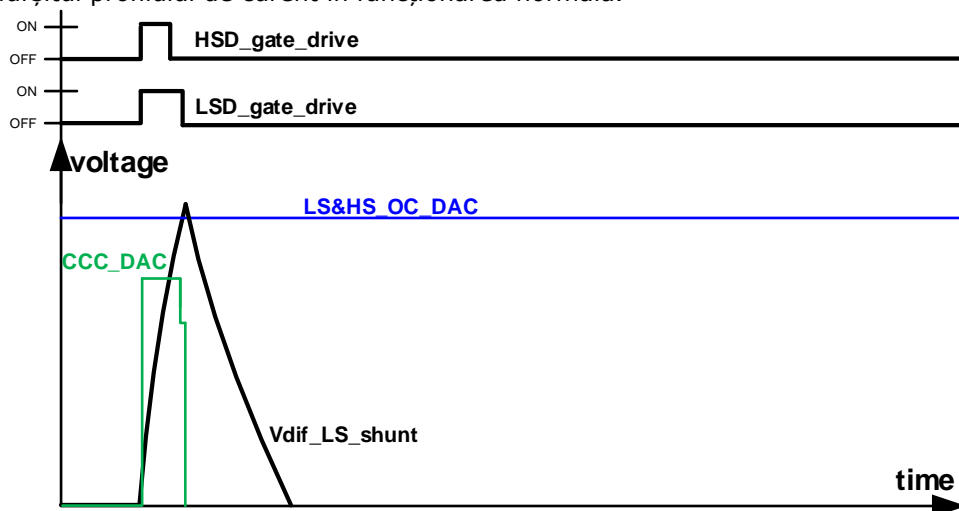


Fig. 6.18 Arhitectura cu două DAC-uri: Supracurent la ieșirea LS (original).

Acum, hardware-ul este în stare de siguranță și diagnosticul poate fi trimis la uC prin SPI cu informațiile respective privind erorile detectate.

3. În cazul în care nu există nicio sarcină conectată la ieșiri, scenariul arată ca în Fig. 6.19 .

Această defecțiune poate fi cauzată de deteriorarea unui fir sau a ambelor fire cu care este conectat solenoidul DIV la ECU sau prin simpla întrerupere a bobinei DIV datorită supraîncălzirii sau datorită altor tipuri de deteriorări mecanice/fizice.

Acum, deoarece sarcina nu este conectată, nu există o cădere de tensiune pe șuntul LS sau HS, motiv pentru care tranzistoarele HS și LS vor rămâne ON până la sfârșitul profilului de curent, iar pragurile de curent stabilite de "CCC_DAC" nu sunt atinse deloc. Acest lucru este detectat de către blocul digital și trimis către uC prin intermediul SPI, indicând că există o eroare de „sarcină neconectată”, adică OL în sistem.

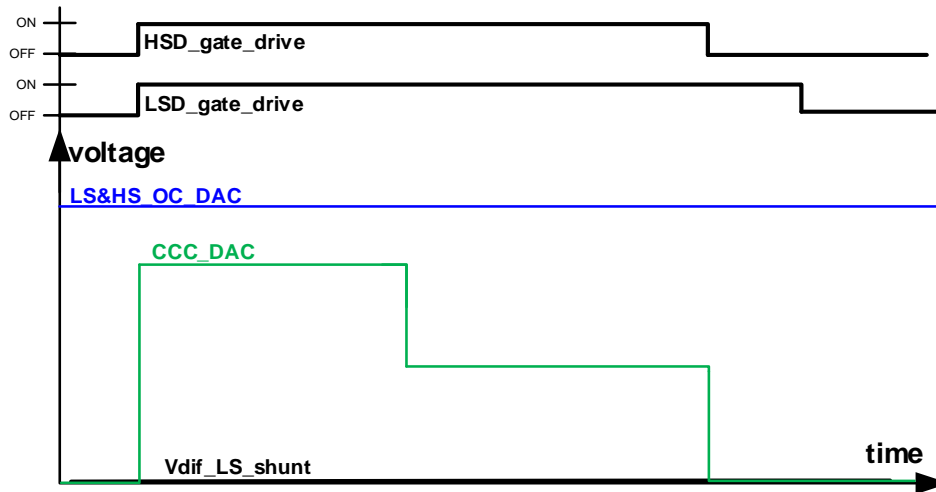


Fig. 6.19 Arhitectura cu două DAC-uri: solenoidul DIV neconectat (original).

Deși topologia și metoda descrisă mai sus funcționează bine și în cazul în care sarcina acționată nu are caracteristică inductivă, totuși funcția de control „peak and hold” prin blocul de CLCC este dezactivată pentru sarcinile rezistive (timpul de creștere a curentului este prea rapid), iar diagnosticul de OC și OL funcționează la fel cum a fost explicat anterior. Prin urmare, DAC-ul pe 8 biți rămâne neutilizat în cazul în care se dorește controlul unei sarcini rezistive sau cu caracter inductiv nesemnificativ, sub 200uH.

6.3.2 Metoda de diagnoză propusă

În acest subcapitol propun o metodă de diagnoză; detectarea supra-curentului (OC) și a sarcinii-neconectate(OL), adecvată pentru ambele tipuri de sarcini comandate, cele cu caracter predominant inductiv, dar și pentru cele cu caracter rezistiv. În cazul sarcinilor rezistive se reutilizează DAC-ul pe 8 biți din blocul de CLCC (oricum neutilizat pentru că în cazul în care este controlată o sarcină rezistivă funcția de CLCC nu se poate folosi).

În Fig. 6.20 se prezintă metoda propusă de diagnostic folosind un numărător implementat în blocul digital în locul unui convertor digital-analog implementat în tehnologie analogică.

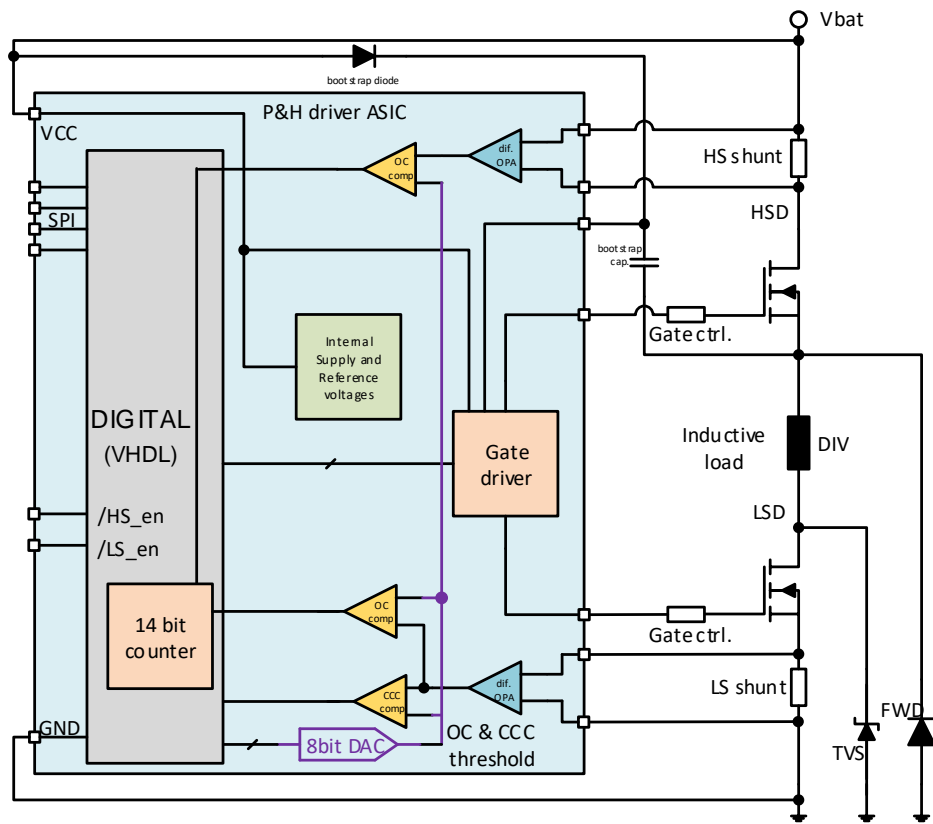


Fig. 6.20 Arhitectura "diagnoză bazată pe constanta de timp a circuitului" (original).

Principala diferență față de topologia anterioară este folosirea unui singur DAC în locul a două. DAC-ul pe 8 biți (cu rol principal în reglarea de curent din blocul CLCC), este folosit suplimentar pentru a ajuta și în strategia de diagnostic propusă. Al doilea DAC pe 6 biți din topologia anterior prezentată este înlocuit de un numărător de 14 biți implementat în blocul digital (mult mai ieftin și mai mic ca suprafață de siliciu pe chip).

În Fig. 6.21 este prezentat un scenariu de funcționare normală a metodologiei de diagnostic propuse.

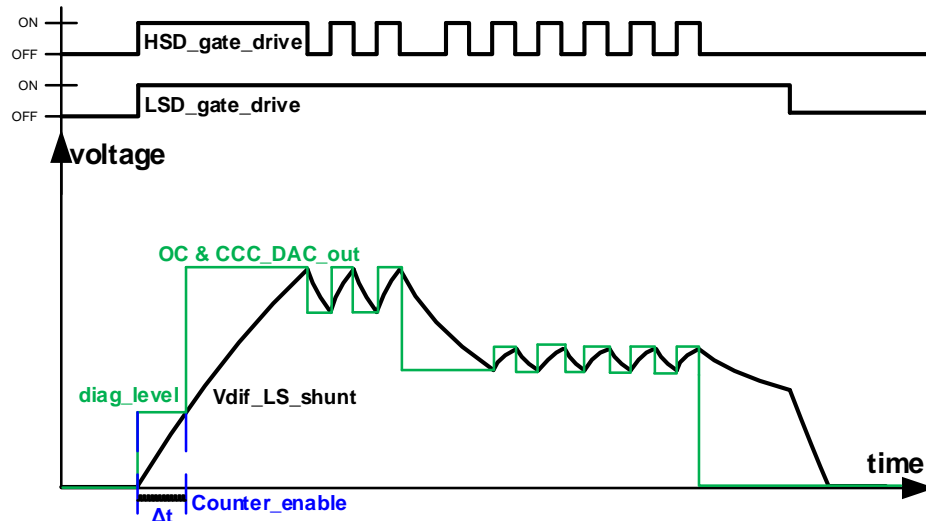


Fig. 6.21 Arhitectura "diagnoză bazată pe constanta de timp a circuitului" ; funcționare normală (original).

DAC-ul implementează acum un nivel intermediar de tensiune numit "diag_level", numărătorul se activează de îndată ce curentul începe să crească și este dezactivat imediat ce curentul de sarcină (tensiunea de pe șuntul de LS) atinge nivelul de diagnosticare prestabilit "diag_level".

În funcționare normală, în funcție de inductanța de sarcină utilizată, valoarea numărătorului trebuie să fie între anumite limite definite de constanta de timp a sistemului.

Pentru funcționarea corectă a metodei de diagnostic propusă mă bazez pe 2 aspecte:

- inductanța unui scurtcircuit (care duce la supra-curent) este mult mai mică decât inductanța normală a sarcinii utilizate (bobina DIV)
- inductanța și rezistența unei sarcini neconectate este mult mai mare decât inductanța și rezistența normală a sarcinii utilizate (bobina DIV)

Pe baza celor de mai sus pot afirma că:

- în cazul unui supra-curent, numărătorul va fi dezactivat mai repede decât în funcționarea normală => **ajunge la o valoare mai mică**
- în cazul unei sarcini neconectate, contorul va fi dezactivat mai târziu decât în funcționarea normală => **ajunge la o valoare mai mare**

În Fig. 6.22 se prezintă scenariul când un eveniment de supra-curent apare pe tranzistorul de LS, timpul necesar atingerii pragului de diagnosticare este mult mai mic decât în funcționarea normală, astfel încât tranzistoarele LS și HS sunt dezactivate (OFF) și se informează uC-ul prin intermediul SPI despre eroare. Datorită timpului de propagare și a întâzierii tranzistorilor, curentul continuă să crească, dar este oprit întotdeauna la o limită de siguranță înainte ca tranzistoarele să fie distruse datorită limitelor maxime din foaia de catalog (curentul maxim în puls / putere de vârf / temperatura de joncțiune de vârf).

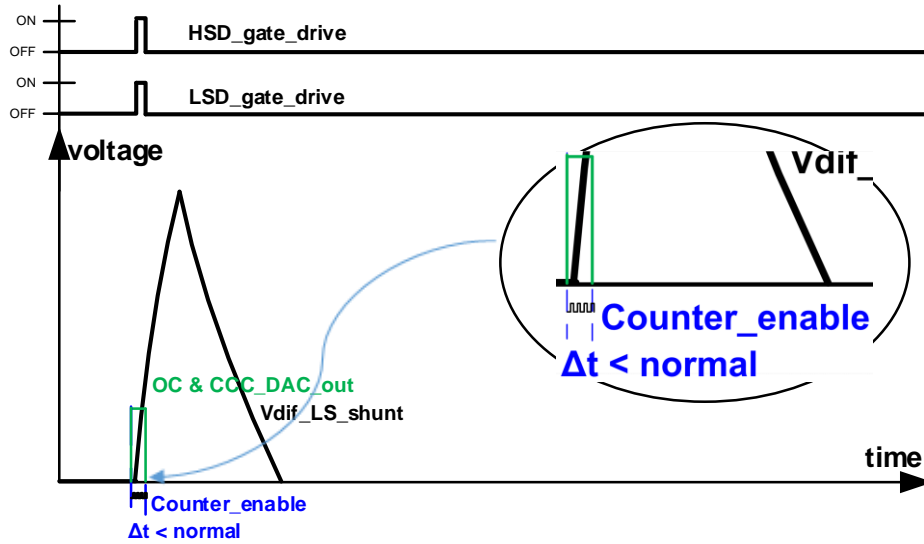


Fig. 6.22 Arhitectura "diagnoză bazată pe constanta de timp a circuitului"; supra-curent (OC) (original).

În Fig. 6.23 este prezentat scenariul de OL, când sarcina nu este conectată, nu va exista curent prin șuntul de LS sau HS, astfel nivelul de diagnosticare nu va fi atins, iar numărătorul va returna o valoare mult mai mare decât în funcționare normală, iar blocul digital va ști că a apărut o eroare de tip OL în sistem și poate informa uC-ul prin SPI.

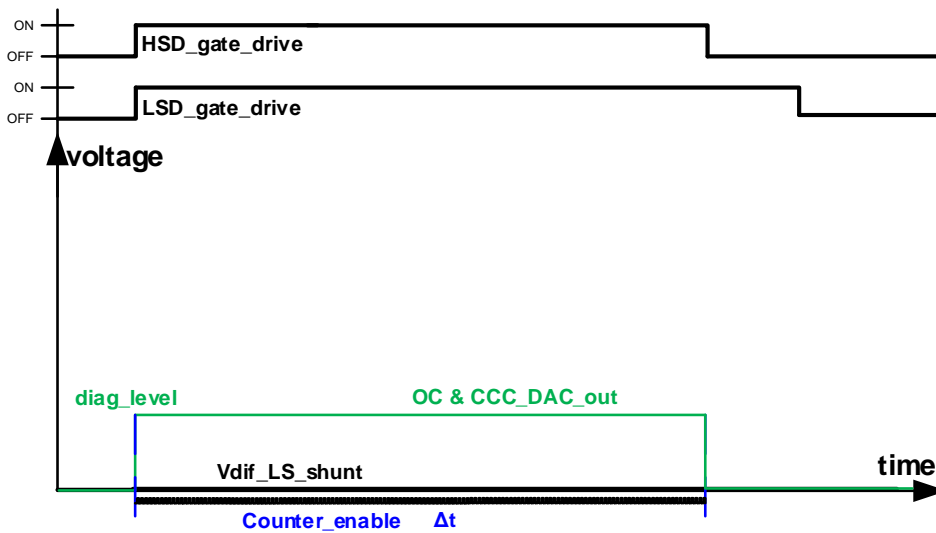


Fig. 6.23 Arhitectura "diagnoză bazată pe constanta de timp a circuitului"; sarcină neconectată (OL) (original).

În Fig. 6.24 este prezentată protecția de OC în cazul în care este controlată o sarcină rezistivă (nu solenoidul DIV). Deoarece o sarcină rezistivă nu poate fi controlată în modul „peak and hold”, panta curentului fiind prea rapidă, practic CCC_DAC rămâne nefolosit, acesta putând fi reutilizat pentru generarea pragului de OC, așa cum se arată în Fig. 6.20 atunci când nu contează ce este la intrările „CCC_comp”(deoarece nu se folosește pentru CLCC).

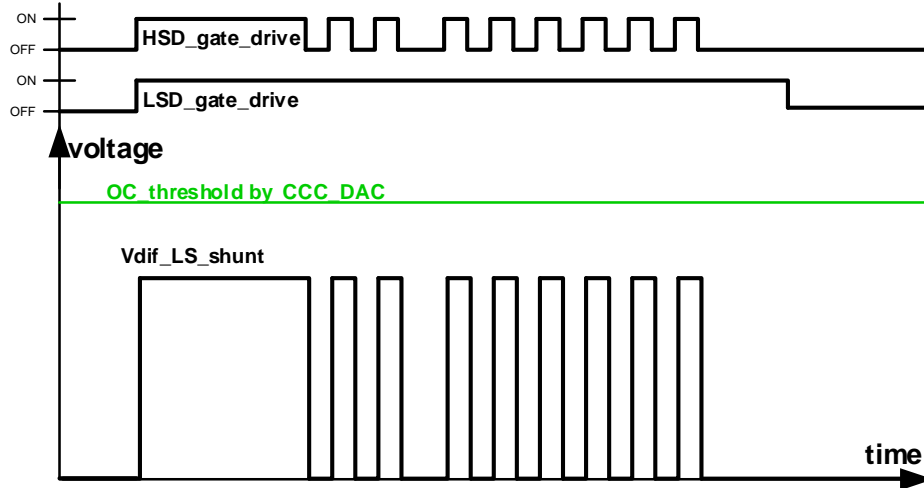


Fig. 6.24 Arhitectura „diagnoză bazată pe constanta de timp a circuitului” ; protecție la supra-curent (OC) pentru sarcini rezistive folosind CCC_DAC pentru generare de prag de OC (original).

În acest caz semnalul "Vdif_LS_shunt" are fronturi foarte rapide, deci detecția de OC bazată pe constanta de timp nu poate fi utilizată, pentru că nu se poate măsura timpul de creștere, dar după cum am arătat anterior, CCC_DAC poate fi folosit pentru a seta pragul de OC.

Când apare un eveniment de supra-curent(OC), semnalul "Vdif_LS_shunt" va trece pragul „OC_treshold_by_CCC_DAC”. Când pragul de OC este traversat, atunci comparatorul de OC va detecta acest eveniment și îl va semnala blocului digital, astfel încât tranzistorii HS și LS sunt dezactivați în timp util, iar apoi starea este semnalată către uC prin SPI.

6.3.3 Rezultate de simulare; metoda de diagnoză propusă

Pentru a dovedi validitatea metodei propuse, am realizat simulări cu funcționarea normală, OL și OC (scurt circuit la baterie) la ieșirea LS.

Fig. 6.25 prezintă rezultatele scenariului de funcționare normală:

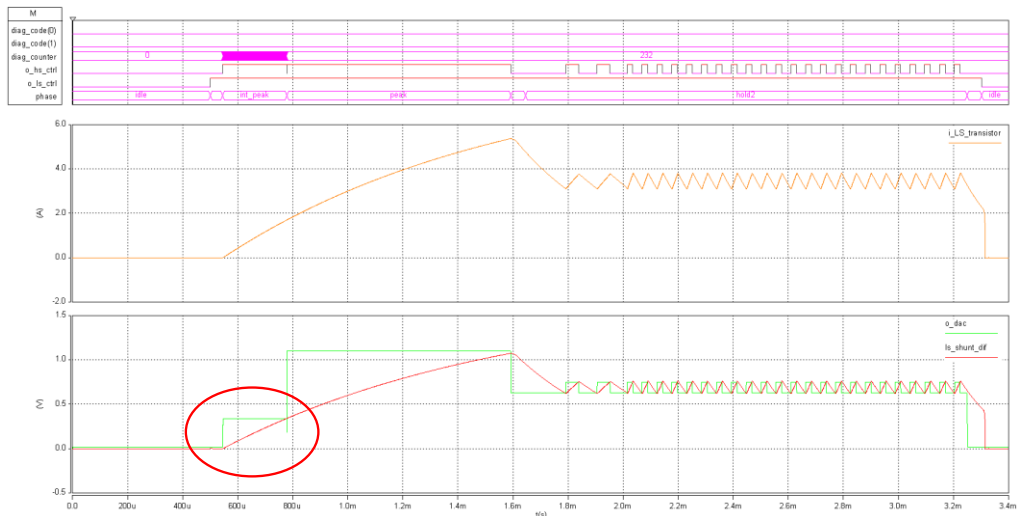


Fig. 6.25 Rezultate de simulare; metoda de diagnoză propusă; funcționare normală (original).

În Fig. 6.26 este un detaliu al zonei marcate din Fig. 6.25.

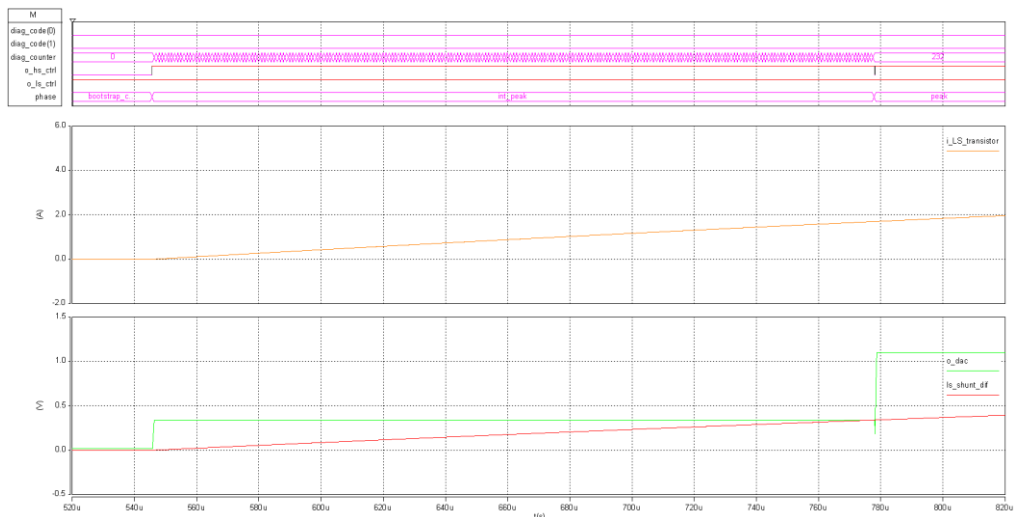


Fig. 6.26 Rezultate de simulare; detaliu Fig. 6.25 (original).

Semnalul „*o_dac*” reprezintă ieșirea convertorului digital-analog utilizat pentru reglarea de curent, ca intrare pentru comparatorul de curent în buclă închisă, CCC.

Semnalul „*ls_shunt_dif*” este cealaltă intrare a comparatorului CCC. Profilul de curent începe pe măsură ce, mașina de stare implementată în digital reprezentată de registrul „*phase*” sare din starea „*idle*” la „*bootstrap_charge*”, pentru a încărca condensatorul de bootstrap pentru comanda tranzistorului de HS. A doua tranziție este de la starea „*bootstrap_charge*” la „*int_peak*” utilizată pentru metoda de diagnostic propusă.

Semnalul „*o_dac*” ia acum valoarea conform nivelului de diagnosticare definit, numărătorul reprezentat prin registrul „*diag_counter*” va începe să crească și să se oprească imediat ce „*ls_shunt_dif*” ajunge la nivelul de diagnostic. În cazul din Fig. 6.26, valoarea finală este de 232 de incrementări; la un semnal de tact de 1 MHz, înseamnă că nivelul de diagnostic definit în funcționare normală este atins în 232us.

Codul de diagnostic generat este reprezentat de către cele două semnale „*diag_code(0)*” și „*diag_code(1)*” care vor lua valoarea „00”, ceea ce înseamnă „*no_failure*”, adică fără defecțiune (funcționare normală).

După ce starea „*int_peak*” se termină, faza de vârf „*peak*” începe, astfel, ieșirea DAC-ului este setată în funcție de căderea de tensiune pe șunt-ul de LS și în funcție de curentul de „*peak*” setat. Atunci când curentul de vârf este atins următoarea setare pentru DAC este pragul de jos al curentului de „*hold2*”, iar apoi când curentul scade sub această valoare, urmează pragul de sus al curentului de „*hold2*” controlând astfel curentul cu o histereză predefinită. Faza de „*hold2*” se termină după un timp definit, programat de către uC. În cele din urmă, faza „*frewheeling*” este executată pentru recircularea energiei.

Rezultatul simulării pentru cazul „*open_load*” este prezentat în Fig. 6.27.

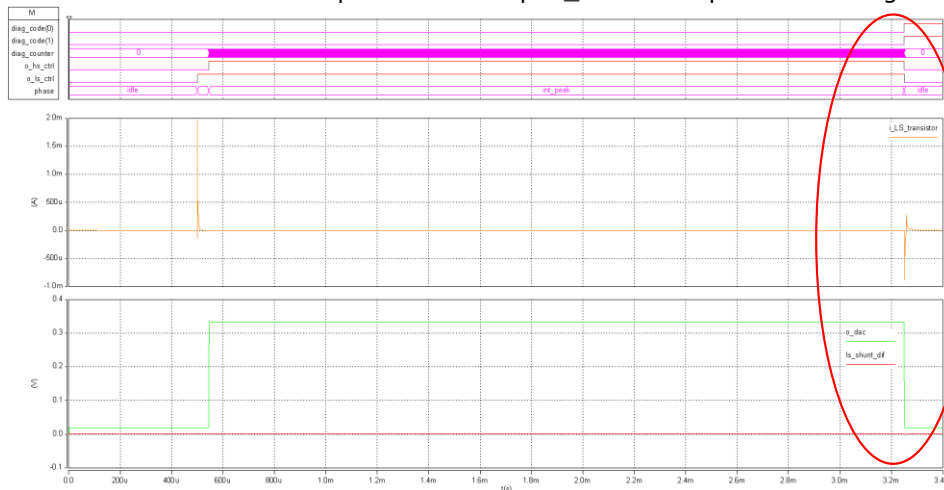


Fig. 6.27 Rezultate de simulare; sarcină neconectată(OL) (original).

Așa cum este prezentat în Fig. 6.27, când sarcina nu este conectată, când cel puțin un fir care conectează sarcina este întrerupt, mașina de stare începe ca în cazul funcționării normale, dar acum fazele „*peak*” și „*hold2*” nu pornesc deoarece „*int_peak*”, faza de diagnostic nu se termină. Faza „*int_peak*” nu se termină deoarece

sarcina nu este conectată, deci nu există curent prin șuntul de LS, astfel că nu este cădere de tensiune pe șunt, deci nu se atinge tensiunea setată de DAC. Faza „int_peak” se termină la sfârșitul profilului de curent, astfel încât „diag_counter” crește până în acel moment. La sfârșitul profilului de curent, valoarea numărătorului a ajuns la 2704, așa cum se vede în Fig. 6.28, detaliu al Fig. 6.27.

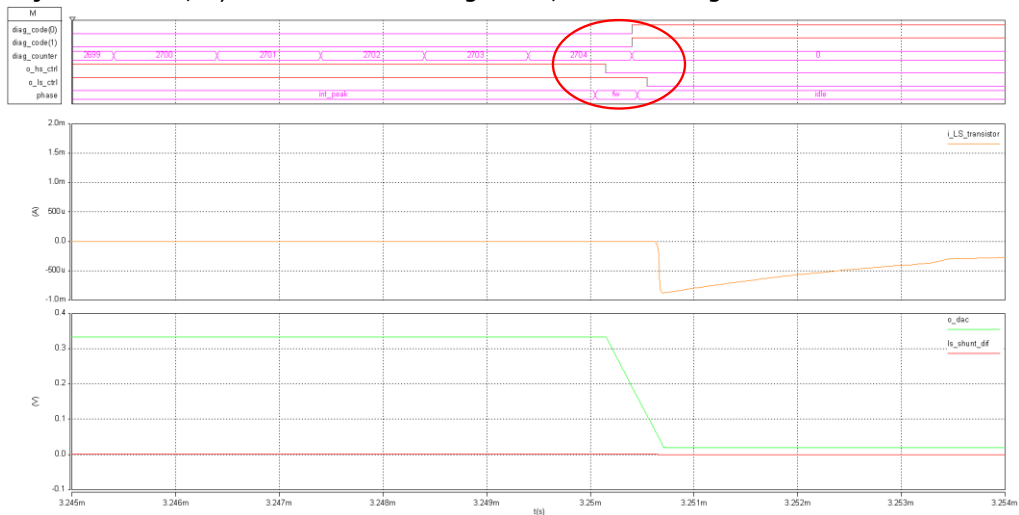


Fig. 6.28 Rezultate de simulare; detaliu Fig. 6.27 (original).

Valoarea numărătorului este mult mai mare decât în funcționarea normală, astfel încât semnalele de diagnostic „diag_code(0)” și „diag_code(1)” vor semnala sistemului „11”, ceea ce înseamnă „open_load”, după cum se vede în Fig. 6.28.

În Fig. 6.29 este prezentat scenariul unui scurtcircuit la baterie aplicat la pinul de ieșire al tranzistorului LS. Aceasta ar conduce la depășirea curentului de funcționare normală prin tranzistorul LS și eventual la deteriorarea acestuia, dacă nu este detectat în timp și acționat prin dezactivarea tranzistorului LS (Fig. 6.29).

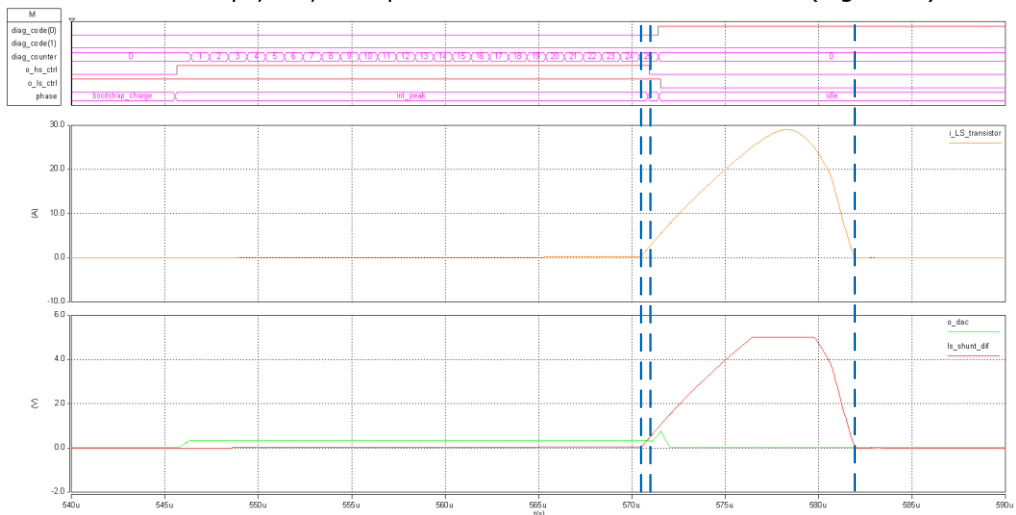


Fig. 6.29 Rezultate de simulare; supra-curent(OC) la tranzistorul LS (original).

Să presupunem că evenimentul de scurtcircuit se produce în timpul fazei "*int_peak*" chiar la prima linie punctată (până atunci curentul crește lent, așa cum se întâmplă în funcționare normală).

După ce a fost aplicat scurtcircuitul, curentul începe să crească cu panta mai rapidă prin tranzistorul LS așa cum arată semnalul "*i_LS_transistor*", dar reflectat și de semnalul "*ls_shunt_dif*".

Căderea de tensiune diferențială a șuntului de LS ajunge la pragul de diagnosticare definit de DAC, la a doua linie punctată, la mai puțin de o incrementare a numărătorului, deci mult mai rapid decât în funcționare normală, rezultând în blocarea imediată tranzistorilor LS și HS și schimbarea valorii semnalele de diagnostic "*diag_code(0)*" și "*diag_code(1)*" la "10" ceea ce înseamnă "**OC_LS**".

Ca efect semnalele "*o_hs_ctrl*" și "*o_ls_ctrl*" reprezentând semnalele de comandă a driverului tranzistoarelor își schimbă starea lor în „low”, dezactivând astfel tranzistoarele HS și LS. Există o întârziere de oprire a tranzistorului de LS (switch OFF delay + switch OFF time) până când se evacuează sarcina din capacitățile interne ale tranzistorului, timp în care curentul tot mai crește până la o valoare de vârf de aproximativ 30A, după ce tranzistorul LS devine complet OFF, astfel încât curentul începe să scadă punând etajul de ieșire în stare de siguranță, așa cum se arată în Fig. 6.29 la a treia linie punctată.

Un alt efect poate fi observat pe semnalul "*ls_shunt_dif*", care manifestă un comportament de saturație a amplificatorului (dar curentul la care se întâmplă saturația amplificatorului este 22A, mult peste curentul normal de funcționare, deci nerelevant pentru aplicația noastră).

6.4 Concluzii și contribuții

Capitolul 6 s-a axat pe îmbunătățirea metodelor de control a solenoidului DIV, reducerea zgomotului produs de vibrațiile valvei la închidere și deschidere și pe găsirea unei metode de diagnoză/protecție electrică mai ieftină pentru circuitul de comandă a solenoidului DIV.

Am studiat un număr de patente internaționale [43]-[65] din domeniul de control de pompe de înaltă presiune pentru combustibil înainte de a depune propunerea de patente proprii rezultate în urma acestei cercetări.

6.4.1 Concluzii

1. Algoritmul de reducere a zgomotului creat de vibrațiile valvei DIV și propus de Kraft et. al. [34] este eficient, însă **are dezavantajul că o intrare ADC a uC trebuie să monitorizeze constant curentul prin bobina DIV** (șuntul de LS).
2. Necesitatea unui port cu ADC al uC **este o resursă costisitoare** mai ales că **trebuie să se facă un număr mare de achiziții A/D** cu frecvență rapidă și să se detecteze închiderea/deschiderea valvei DIV, astfel încât să se optimizeze profilul de curent cu scop de reducere a zgomotului.
3. Kraft et. al. folosesc un al doilea puls aplicat pentru frânarea deschiderii valvei DIV, acest puls de curent este **controlat de un semnal PWM în buclă deschisă**, cu frecvență și factor de umplere constante.
4. În cazul în care avem control de curent în buclă închisă, este suficient doar să se urmărească cu un numărător implementat în VHDL duratele de ON și OFF ale

semnalului de comandă ca să se detecteze închiderea/deschiderea valvei DIV astfel să se optimizeze profilul de curent cu scop de reducere a zgomotului.

5. Îmbunătățirile obținute utilizând algoritmul de reducere a zgomotului aplicând două pulsuri de curent controlate în buclă închisă, sunt prezentate în Fig. 6.7.
6. Pulsul de închidere a valvei DIV în algoritmul propus de mine aplică în faza de „pull-in” o tehnică de „soft-landing” pentru a încetini închiderea valvei, astfel reducând zgomotul, așa cum se observă în Fig. 6.7.
7. Arhitectura cea mai sigură și precisă de diagnostic pentru supra curent este cea cu șunt, comparator și DAC, așa cum se arată în Fig. 6.16.
8. Metoda de diagnoză și protecție propusă de mine înlocuiește DAC-ul cu un numărător implementat în VHDL și se bazează pe timpul de creștere a curentului prin bobina DIV. Dacă timpul de creștere este mai rapid decât în funcționare normală rezultă că avem eveniment de supracurent datorită scurt-circuitului cu caracter inductiv mai mic față de bobina DIV. Dacă timpul de creștere este mult mai mare față de funcționare normală, rezultă că sarcina nu este conectată.
9. Metoda de diagnoză și protecție propusă de mine este destinată exclusiv sarcinilor cu caracter inductiv semnificativ, mai mare ca 200uH.
10. Metoda de diagnoză și protecție propusă de mine este eficientă, și ieftină însă susceptibilă la variația constantei de timp a circuitului (L,R), având toleranțe mai mari a pragului de detecție comparativ cu metoda standard de protecție cu șunt/comparator și DAC.

6.4.2 Contribuții

1. Am analizat stadiul actual al metodelor existente de reducere a zgomotului, așa cum se arată în lucrarea mea [38].
2. Am a propus o metodă alternativă de reducere a zgomotului, așa cum se arată în lucrarea mea [38].
3. Metoda alternativă propusă de mine și publicată în [38] se bazează pe [43], însă **nu necesită o intrare de ADC** pentru monitorizarea constantă a curentului prin bobina DIV.
4. **În metoda propusă de mine în [38], ADC-ul necesar în [43] este înlocuit cu un numărător implementat în VHDL, mult mai ieftin și simplu față de un ADC.** Acest lucru este posibil datorită controlului de curent în buclă închisă care aproximativ își dublează frecvență de comutare în momentul în care se închide valva DIV. Această frecvență de comutație este monitorizată de numărătorul implementat în VHDL care detectează pe baza algoritmului prezentat în Fig. 6.11 mișcarea valvei DIV.
5. Metoda propusă de mine aplică ambele pulsuri, cel de închidere și cel de deschidere cu **control de curent în buclă închisă**.
6. Am testat pe o valvă DIV folosită în sistemele de 250bari eficiența metodei propuse de reducere a zgomotului, așa cum se arată în lucrarea mea [38].
7. Am pus accent pe rezolvarea problemelor (mișcarea precoce a valvei și variația timpului natural de închidere/deschidere a valvei) găsite la implementarea modalității propuse de reducere a zgomotului, așa cum se arată în [38].
8. Am propus o metodă și un algoritm pentru detecția mișcării supapei DIV cu scopul de a facilita implementarea metodei propuse de reducere a zgomotului, așa cum se arată în [38],

9. Atât metoda de reducere a zgomotului cât și algoritmul de detecție a mișcării valvei DIV au fost simulate, dar și implementate în circuitul practic validând funcționalitatea.
10. Am analizat stadiul actual existent referitor la diagnoză și protecție a circuitului de comandă a valvei DIV.
11. Am propus o modalitate alternativă de diagnoză și protecție pentru sarcini inductive bazată pe constanta de timp a circuitului de putere (incluzând bobina DIV). Rezultatele de simulare atestă funcționarea noii modalități de diagnoză și protecție propusă.
12. Am deținut trei patente legate de îmbunătățirile aduse în acest capitol
 1. **L.Molnar**, B.Komlosi, „High Pressure Fuel Pump Solenoid Valve Movement Detection”, **2019E03508 RO**
 2. **L.Molnar**, B.Komlosi, „Time Constant Based Over current and Open Load diagnostic for inductive loads”, **2019E03495 RO**
 3. **L.Molnar**, B.Komlosi, „ High Pressure Fuel Pump Solenoid Valve Noise cancelling with two current controlled_current_pulses and soft landing featuring Movement Detection”, **2019E04397 RO**

7. Concluzii și contribuții personale

7.1 Concluzii finale

Capitolul 3

1. Este esențială testarea în faze timpurii de proiectare a ASIC-ului deoarece eventualele greșeli de concept sau proiectare depistate devreme se pot corecta înainte de producerea fizică a ASIC-ului.
2. Toate erorile/greșelile descoperite după ce ASIC-ul a fost produs, vin cu costuri mari de re-proiectare și re-fabricare.
3. Platformele de emulare a ASIC-ului sunt o modalitate ieftină, rapidă dar și eficientă pentru testarea conceptuală a ASIC-ului în faze timpurii ale proiectării.
4. Platformele de validare a sistemului cu ASIC sunt o modalitate de testare și mai timpurie a conceptului, chiar și atunci când platforma de emulare nu este încă fizic disponibilă, dar totuși se pot neglija unele elemente parazite (inductanța firelor, capacități parazite, etc.) în comparație cu platforma de emulare a ASIC-ului.

Capitolul 4

5. Pentru a crește eficiența și puterea motoarelor termice pe benzină, dar în același timp să se țină pasul cu evoluția normelor de poluare, este nevoie de sisteme cu injecție directă și control electronic al motoarelor.
6. Performanțele sistemului de injecție constând din injectoare, pompa de benzină de înaltă presiune, bujii și modulul electronic de control reprezintă elementul cheie pentru a îndeplini cele expuse la punctul anterior.
7. Controlul precis al combustibilului necesită o presiune constantă în rampa comună.
8. Presiunea constantă din rampa comună este direct influențată de performanța pompei de benzină și a controlului său.
9. Controlul electro-valvei pompei de înaltă presiune trebuie să fie robust (ASIC-ul trebuie să ofere către uC diagnoza circuitului de putere și în același timp să ia decizii independente de uC pentru protecția locală a circuitului de putere) și cu un timp de răspuns rapid (uC trebuie să poată modifica parametrii de profil de curent înainte de fiecare puls, iar la final de puls este important să evacuem rapid energia înmagazinată în bobina DIV pentru a scurta cât mai mult timpul de deschidere a valvei scurtând astfel timpul necesar până la aplicarea următorului puls de curent) pentru a facilita dezvoltarea algoritmilor de reglare a presiunii în rampa comună.
10. Pentru a spori confortul auditiv pentru unele pompe de benzină de înaltă presiune este necesar un control electronic sofisticat cu dublu scop, de a menține presiunea constantă în rampa comună, dar și de a aplica un profil de curent optim astfel încât deschiderea și închiderea valvei electro-mecanice a pompei de benzină să producă vibrații (zgomote) cât mai mici.

Capitolul 5

11. Organizarea ierarhică a schemei de simulare ușurează depanarea schemei și facilitează împărțirea/delimitarea circuitelor folosite ulterior pentru emulare ASIC-ului prin platforma MTP.
12. Organizarea schemei pe blocuri funcționale individuale ajută ulterior la re folosirea lor în procese noi de dezvoltare a altor ASIC-uri.
13. Detaliile de implementare a blocurilor funcționale au influență directă asupra rezultatelor de simulare și a timpului de execuție a simulării. De exemplu în cazul schemei de simulare care utilizează model de tranzistor HS și LS ideali, fără toleranțe, 5.4ms timp de simulare se execută în 103s, așa cum arată raportul din anexa A7.
14. Modelarea cu prea puține detalii poate fi o sursă de erori când comparăm rezultatele cu cele ale platformei de emulare a ASIC-ului. În anexa A8 este prezentat cazul unde se compară rezultatele simulării cu tranzistori HS și LS ideali cu cel cu tranzistori reali. Se observă o diferență semnificativă în caracteristicile de comutație ale tranzistorului de HS, care rezultă o putere disipată mai mare în cazul în care se utilizează model ideal de tranzistor. Puterea disipată medie per ciclu de comutație în cazul tranzistorului ideal este de aproximativ 0.81W iar în cazul cu model real este 0.47W, o diferență de 41% ceea ce poate conduce la o posibilă supradimensionare a circuitului de putere.
15. Modelarea cu prea multe detalii crește semnificativ timpul de simulare fiind nevoie de putere de calcul adițională pentru a rula simularea în timp util. De exemplu în cazul schemei de simulare care utilizează model de tranzistor HS și LS real, cu toleranțe, 5.4ms timp de simulare se execută în 128s, timp de simulare cu 25% mai lung față de simularea cu model de tranzistor ideal, așa cum arată raportul din anexa A7.
16. Sistemul hidraulic nu influențează considerabil performanțele electrice ale valvei DIV.
17. Cuplul aplicat motorului introduce doar o întârziere în pornirea activării supapei în comparație cu BDC.
18. Efectul relevant pentru variația inductanței valvei electro-mecanice DIV este datorat mișcării supapei de scurgere.
19. Rezistența bobinei DIV este influențată direct de temperatura mediului în care aceasta se găsește.
20. Pentru a caracteriza modelul de simulare a bobinei DIV cât mai precis, a fost nevoie de măsurări efectuate și pe un banc de încercare unde pompa funcționează în sistem hidraulic pentru a determina influențele non-electrice care conduc la variația inductivității.
21. Pentru rezultate de simulare cât mai precise este important să se introducă efecte temporale în variația inductivității bobinei DIV, modelând variația timpului de închidere/deschidere a sa.
22. Rezultatele comparate arată o potrivire bună a semnalelor atât în timp cât și în amplitudine și formă, totuși potrivirea rezultatelor nu este perfectă necesitând investigații la nivelul modelului de simulare a DAC-ului folosit în blocul de CLCC.
23. Metoda de simulare propusă ca și etapă de început în procesul de dezvoltare a unui ASIC este potrivită pentru a valida și verifica conceptul/blocurile de arhitectură gândite pentru ASIC, cu avantajul că putem utiliza în simulare și modulul din care face parte ASIC-ul, respectiv sarcina comandată, valva DIV.

Capitolul 6

24. Algoritm de reducere a zgomotului creat de vibrațiile valvei DIV și propus de Kraft et. al. [34] este eficient, însă **are dezavantajul că o intrare ADC a uC trebuie să monitorizeze constant curentul prin bobina DIV** (șuntul de LS).
25. Necesitatea unui port cu ADC al uC **este o resursă costisitoare** mai ales că **trebuie să se facă un număr mare de achiziții A/D** cu frecvență rapidă și să se detecteze închiderea/deschiderea valvei DIV, astfel încât să se optimizeze profilul de curent cu scop de reducere a zgomotului.
26. Kraft et. al. folosesc un al doilea puls aplicat pentru frânarea deschiderii valvei DIV, acest puls de curent este **controlat de un semnal PWM în buclă deschisă**, cu frecvență și factor de umplere constante.
27. În cazul în care avem control de curent în buclă închisă, este suficient doar să se urmărească cu un numărător implementat în VHDL duratele de ON și OFF ale semnalului de comandă ca să se detecteze închiderea/deschiderea valvei DIV astfel să se optimizeze profilul de curent cu scop de reducere a zgomotului.
28. Îmbunătățirile obținute utilizând algoritmul de reducere a zgomotului aplicând două pulsuri de curent controlate în buclă închisă, sunt prezentate în Fig. 6.7.
29. Pulsul de închidere a valvei DIV în algoritmul propus de mine aplică în faza de „pull-in” o tehnică de „soft-landing” pentru a încetini închiderea valvei, astfel reducând zgomotul, așa cum se observă în Fig. 6.7.
30. Arhitectura cea mai sigură și precisă de diagnostic pentru supra curent este cea cu șunt, comparator și DAC, așa cum se arată în Fig. 6.16.
31. Metoda de diagnoză și protecție propusă de mine înlocuiește DAC-ul cu un numărător implementat în VHDL și se bazează pe timpul de creștere a curentului prin bobina DIV. Dacă timpul de creștere este mai rapid decât în funcționare normală rezultă că avem evenimente de supracurent datorită scurt-circuitului cu caracter inductiv mai mic față de bobina DIV. Dacă timpul de creștere este mult mai mare față de funcționare normală, rezultă că sarcina nu este conectată.
32. Metoda de diagnoză și protecție propusă de mine este destinată exclusiv sarcinilor cu caracter inductiv semnificativ, mai mare ca 200uH.
33. Metoda de diagnoză și protecție propusă de mine este eficientă, și ieftină însă susceptibilă la variația constantei de timp a circuitului (L,R), având toleranțe mai mari a pragului de detecție comparativ cu metoda standard de protecție cu șunt/comparator și DAC.

7.2 Contribuții personale

Capitolul 3

1. Propun pe baza lucrării proprii [34] platforma de validare MTP care poate îmbunătăți performanța de validare datorită utilizării directe a TC-ului pentru a emula blocul analogic al ASIC-ului. MTP poate fi reutilizat cu ușurință pentru noile proiecte datorită structurii sale modulare.
2. Propun tot pe baza lucrării proprii [34] două arhitecturi alternative de simulare a sistemelor fără Co-simulare cu număr redus de simulatoare utilizând un simulator multi-domeniu pentru a realiza simularea de arhitectură a ASIC-ului atunci când încă nu este disponibilă platforma MTP de emulare a ASIC-ului pentru a realiza teste și a valida funcționalitatea conceptuală corectă în faze timpurii ale dezvoltării ASIC-ului

Capitolul 4

3. Prezint și explic pe baza lucrării proprii [38] funcționarea sistemelor GDI cu detalii specifice referitoare la acționarea mecanică și comanda electronică a pompei de benzină de înaltă presiune, respectiv a valvei electro-mecanice constituate a acesteia.

Capitolul 5

4. În subcapitolul 5.1.5 am efectuat un număr semnificativ de scenarii de funcționare cu diferite medii de testare și am analizat datele așa cum se arată în lucrarea mea [39].
5. Am propus un model nou de simulare comparativ cu modelul simplu R-L așa cum se arată în [39]. Noul model este capabil să urmărească mai bine forma curentului prin valvă fizică utilizată datorită structurii tabelului de căutare.
6. Am modelat de asemenea comportamentul de saturație al inductanței, sporind performanța generală a simulării.
7. Modelul de simulare electrică a bobinei DIV propus în [39] a fost îmbunătățit prin introducerea efectului temporal pentru a simula închiderea supapei. Închiderea supapei s-a introdus prin modificarea inductivității bobinei în funcție de un semnal sincronizat cu începutul profilului de curent și se manifestă prin modificarea frecvenței de comutație în faza de „hold” atunci când se folosește strategia de CLCC pentru reglarea de curent prin bobină așa cum se arată în subcapitolul 5.1.5.7.
8. Am realizat o **platformă de simulare** conform cu cea propusă în lucrarea mea [34] și explicată în subcapitolul 3.2.3
9. Am realizat o **platformă de validare** conform cu cea propusă în [34] și explicată în subcapitolul 3.1.1 cu excepția implementării etapei în care am propus să se înlocuiască blocul analogic emulat cu componente discrete cu o placă care să conțină Test Chip-ul (Test Chip-ul este un CI care conține doar blocuri analogice a fi implementate în ASIC).
10. Am comparat rezultatele **platformei de simulare** cu rezultatele **platformei de validare**. Comparația arată o potrivire bună a rezultatelor, demonstrând eficiența platformei de validare.

Capitolul 6

11. Am analizat stadiul actual al metodelor existente de reducere a zgomotului, așa cum se arată în lucrarea mea [38].
12. Am a propus o metodă alternativă de reducere a zgomotului, așa cum se arată în lucrarea mea [38].
13. Metoda alternativă propusă de mine și publicată în [38] se bazează pe [43], însă **nu necesită o intrare de ADC** pentru monitorizarea constantă a curentului prin bobina DIV.
14. **În metoda propusă de mine în [38], ADC-ul necesar în [43] este înlocuit cu un numărător implementat în VHDL, mult mai ieftin și simplu față de un ADC.** Acest lucru este posibil datorită controlului de curent în buclă închisă care aproximativ își dublează frecvența de comutare în momentul în care se închide valva DIV. Această frecvență de comutație este monitorizată de numărătorul implementat în VHDL care detectează pe baza algoritmului prezentat în Fig. 6.11 mișcarea valvei DIV.
15. Metoda propusă de mine aplică ambele pulsuri, cel de închidere și cel de deschidere cu **control de curent în buclă închisă**.

16. Am testat pe o valvă DIV folosită în sistemele de 250bari eficiența metodei propuse de reducere a zgomotului, așa cum se arată în lucrarea mea [38].
17. Am pus accent pe rezolvarea problemelor (mișcarea precoce a valvei și variația timpului natural de închidere/deschidere a valvei) găsite la implementarea modalității propuse de reducere a zgomotului, așa cum se arată în [38].
18. Am propus o metodă și un algoritm pentru detecția mișcării supapei DIV cu scopul de a facilita implementarea metodei propuse de reducere a zgomotului, așa cum se arată în [38].
19. Atât metoda de reducere a zgomotului cât și algoritmul de detecție a mișcării valvei DIV au fost simulate, dar și implementate în circuitul practic validând funcționalitatea.
20. Am analizat stadiul actual existent referitor la diagnoză și protecție a circuitului de comandă a valvei DIV.
21. Am propus o modalitate alternativă de diagnoză și protecție pentru sarcini inductive bazată pe constanta de timp a circuitului de putere (incluzând bobina DIV). Rezultatele de simulare atestă funcționarea noii modalități de diagnoză și protecție propusă.
22. Am depus trei patente legate de îmbunătățirile aduse în acest capitol
 1. **L.Molnar**, B.Komlosi, „High Pressure Fuel Pump Solenoid Valve Movement Detection”, **2019E03508 RO**
 2. **L.Molnar**, B.Komlosi, „Time Constant Based Over current and Open Load diagnostic for inductive loads”, **2019E03495 RO**
 3. **L.Molnar**, B.Komlosi, “ High Pressure Fuel Pump Solenoid Valve Noise cancelling with two current controlled_current_pulses and soft landing featuring Movement Detection”, **2019E04397 RO**

7.3 Lista lucrărilor

LISTA PUBLICAȚIILOR REZULTATE ÎN URMA CERCETĂRII DOCTORALE, PUBLICATE SAU ACCEPTATE SPRE PUBLICARE, SUB AFILIERE UPT

Ing. *Laszlo MOLNAR* student doctorand

1. Lucrări științifice publicate în reviste indexate Web of Science-WoS (ISI)

1.

2. Lucrări științifice publicate în volumele unor manifestări științifice (Proceedings) indexate Web of Science-WoS (ISI) Proceedings

1. **L. Molnar** and A. Gontean, "Functional Simulation methodes," 2016 12th IEEE International Symposium on Electronics and Telecommunications (ISETC), Timisoara, 2016, pp. 198-202. (WOS:000390717800044)
2. **L. Molnar** and A. Gontean, "Fault simulation methodes," 2016 12th IEEE International Symposium on Electronics and Telecommunications (ISETC), Timisoara, 2016, pp. 194-197. (WOS:000390717800043)
3. **L. Molnar** and A. Gontean, "ASIC validation architectures: Hardware versus simulation," 2017 21st International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, 2017, pp. 671-676. (WOS:000427419900111)
4. **L. Molnar**, A. Gontean, A. Kotlar and P. Svasta, "Simulation model for automotive high pressure fuel pump," 2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME), Constanta, 2017, pp. 231-234. (WOS:000428032300048)
5. **L. Molnar**, O. Luca and A. Gontean, "Simulation Method to Evaluate Thermal Runaway of MOSFETs," 2018 41st International Spring Seminar on Electronics Technology (ISSE), Zlatibor, 2018, pp. 1-6. (WOS:000449866600011)
6. **L. Molnar** and A. Gontean, "Hardware Difficulties and Improvements for High Pressure Fuel Pump Solenoid Valve Noise Cancellation," 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME), Iasi, 2018, pp. 84-89. (WOS:000466960400015)
7. **L. Molnar**, A. Kotlar and S. Lica, "High Power Switch using IGBT and GaN MOSFET A solution proposal," 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME), Iasi, 2018, pp. 43-48. (WOS:000466960400008)

3. Lucrări științifice publicate în reviste de specialitate indexate BDI (cu specificarea BDI)

1.
2.

4. Lucrări științifice publicate în volumele unor manifestări științifice (Proceedings) indexate BDI (cu specificarea BDI)

1.
2.

5. Lucrări științifice publicate în volumele unor manifestări științifice internaționale (Proceedings) din străinătate

1.
2.

6. Lucrări științifice publicate în volumele unor manifestări științifice

1.
2.

Data:

AVIZAT, Conducător științific	INTOCMIT, Student doctorand
<i>Prof.univ.dr.ing.</i>	<i>ing.</i>

7.4 Lista citărilor

Citări pentru lucrarea:

[22] L. Molnar and A. Gontean, "Fault simulation methodes," *2016 12th IEEE International Symposium on Electronics and Telecommunications (ISETC)*, Timisoara, 2016, pp.194-197. doi: 10.1109/ISETC.2016.7781089

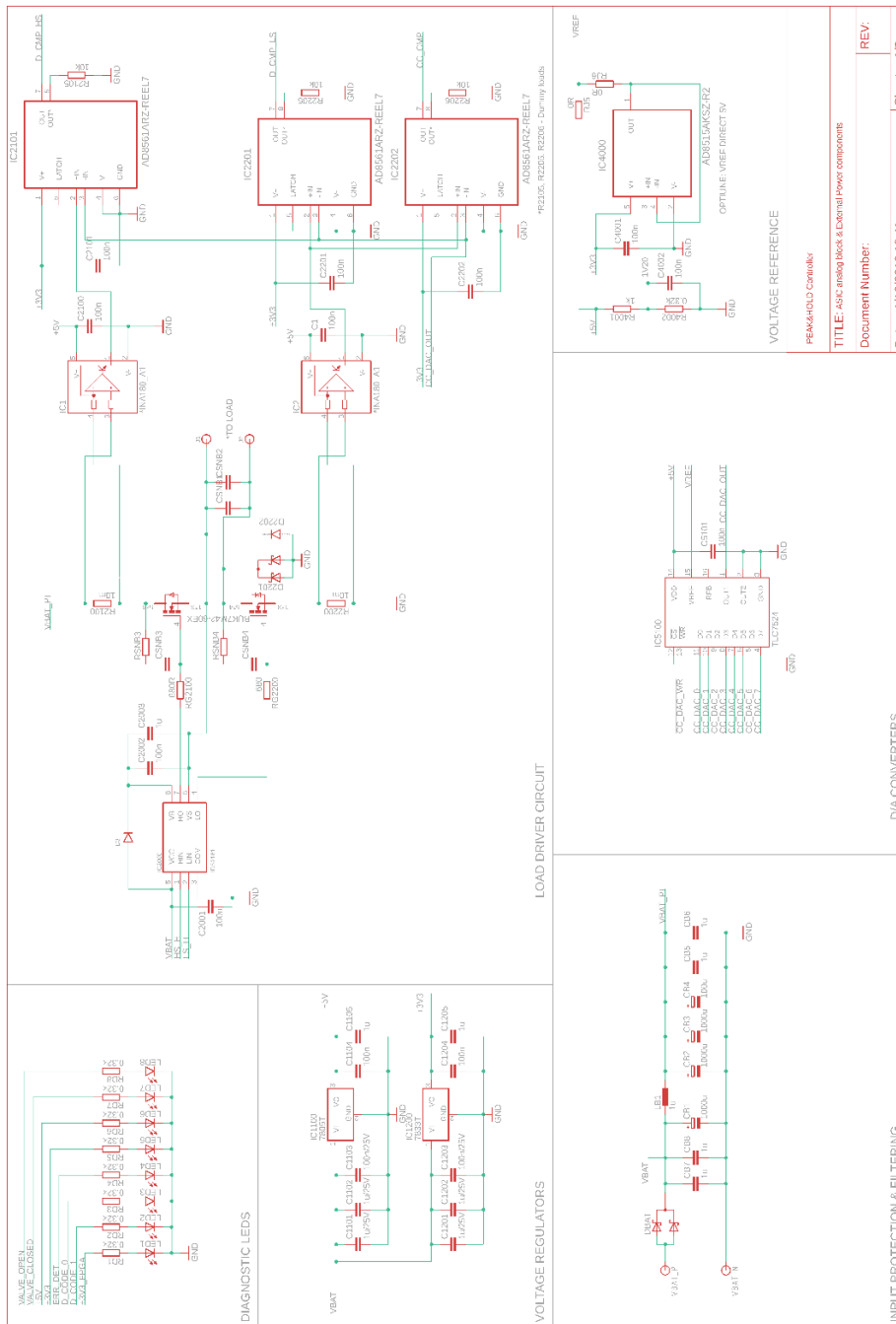
1. Bhattacharya, Rahul, S. H. M. Ragamai, and Subindu Kumar. "SFG Based Fault Simulation of Linear Analog Circuits Using Fault Classification and Sensitivity Analysis." *International Symposium on VLSI Design and Test*. Springer, Singapore, 2017.
Indexată ISI: WOS:000436402900019
2. Hahanov, Ivan, et al. "Deductive qubit fault simulation." *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*. IEEE, 2017.
3. Khakhanov, V.I., et al. "The qubit method of deductive fault analysis for logic circuits." *Electronic Modeling* (2017).
4. Hahanov, Vladimir, et al. "Quantum Computing for Test Synthesis." *Cyber Physical Computing for IoT-driven Services*. Springer, Cham, 2018. 95-134.
5. Hahanov, V. I., et al. "QUBIT METHOD FOR DEDUCTIVE FAULT ANALYSIS OF LOGIC CIRCUITS." *Elektron. model* 39.6 (2017): 59-92.
6. Litvinova, Evgeniya Ivanovna, Igor Valeriyovych Yemelyanov, and Ivan Vladimirovich Khakhanov. "Cloud service for testing and verifying crystalline systems." *Radio Electronics and Informatics* 3 (2017).

7.5 Listă patente

1. L.Molnar, B.Komlosi, „High Pressure Fuel Pump Solenoid Valve Movement Detection”, **2019E03508 RO**
2. L.Molnar, B.Komlosi, „Time Constant Based Over current and Open Load diagnostic for inductive loads”, **2019E03495 RO**
3. L.Molnar, B.Komlosi, “ High Pressure Fuel Pump Solenoid Valve Noise cancelling with two current controlled_current_pulses and soft landing featuring Movement Detection”, **2019E04397 RO**

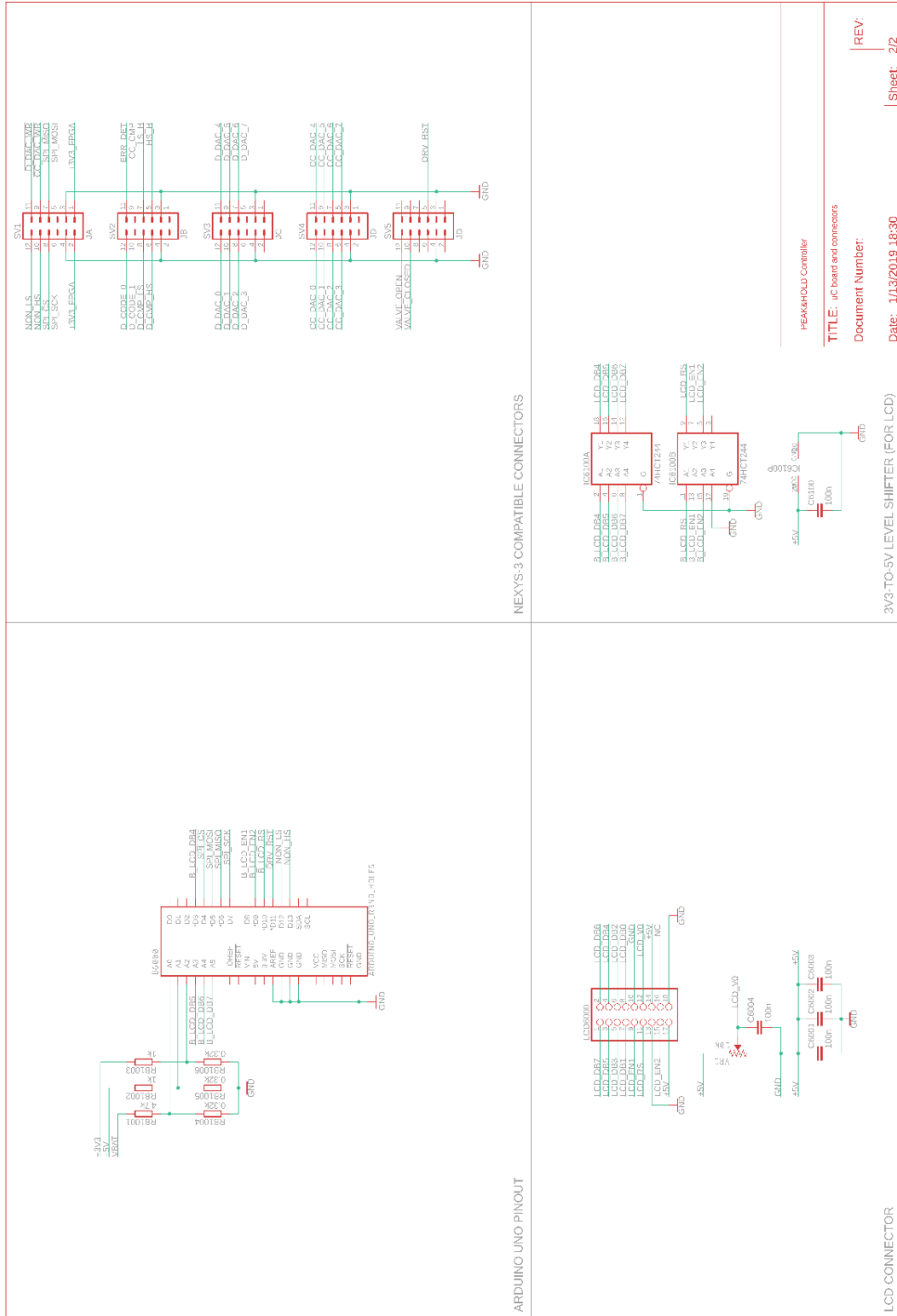
Anexe

A1. – Schema electrică a sistemului emulat; pagina 1.

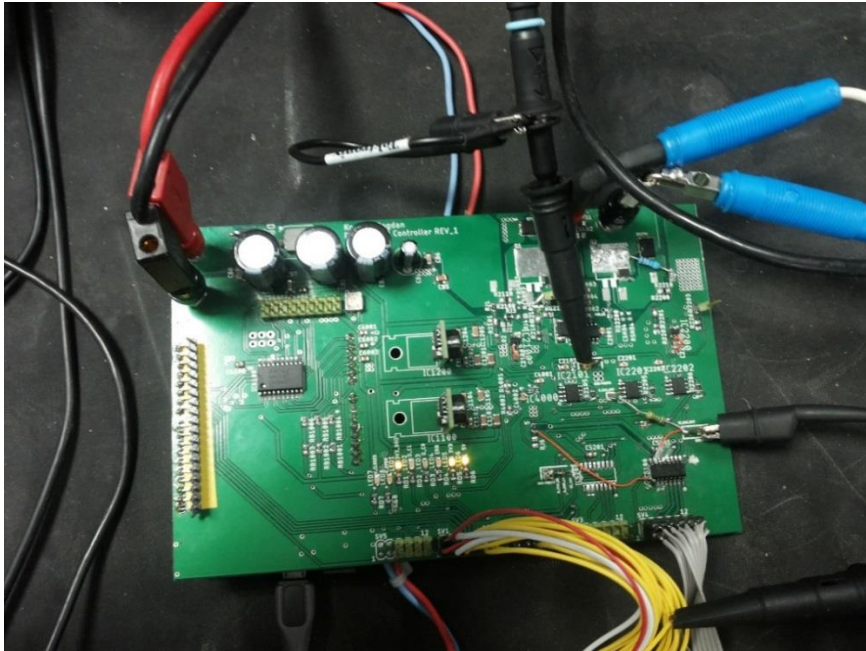


TITLE: ASIC analog back & External Power components	REV:
Document Number:	Sheet 1/2
Date: 1/13/2019 18:41	

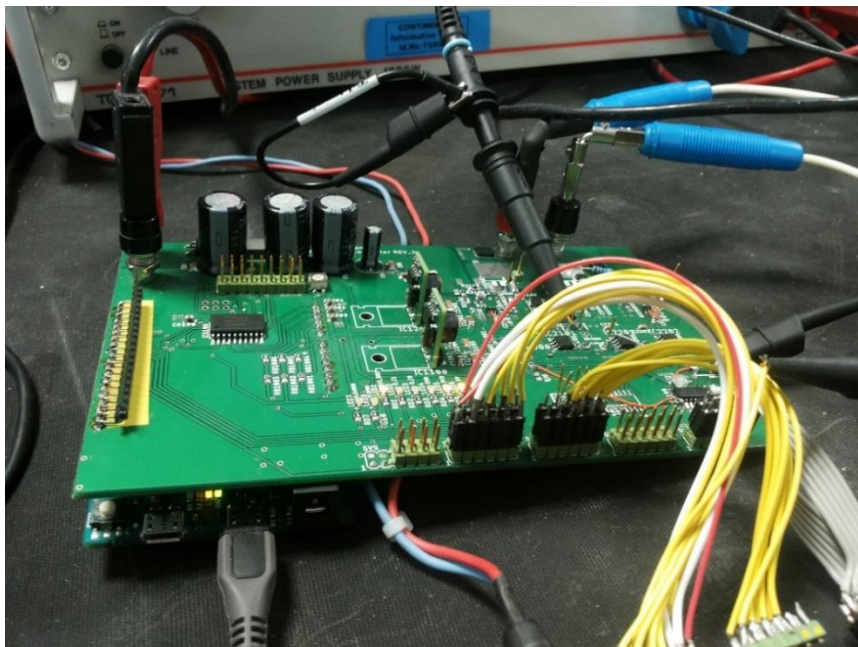
A2. – Schema electrică a sistemului emulat; pagina 2.



A3. – Placa cu circuitul fizic

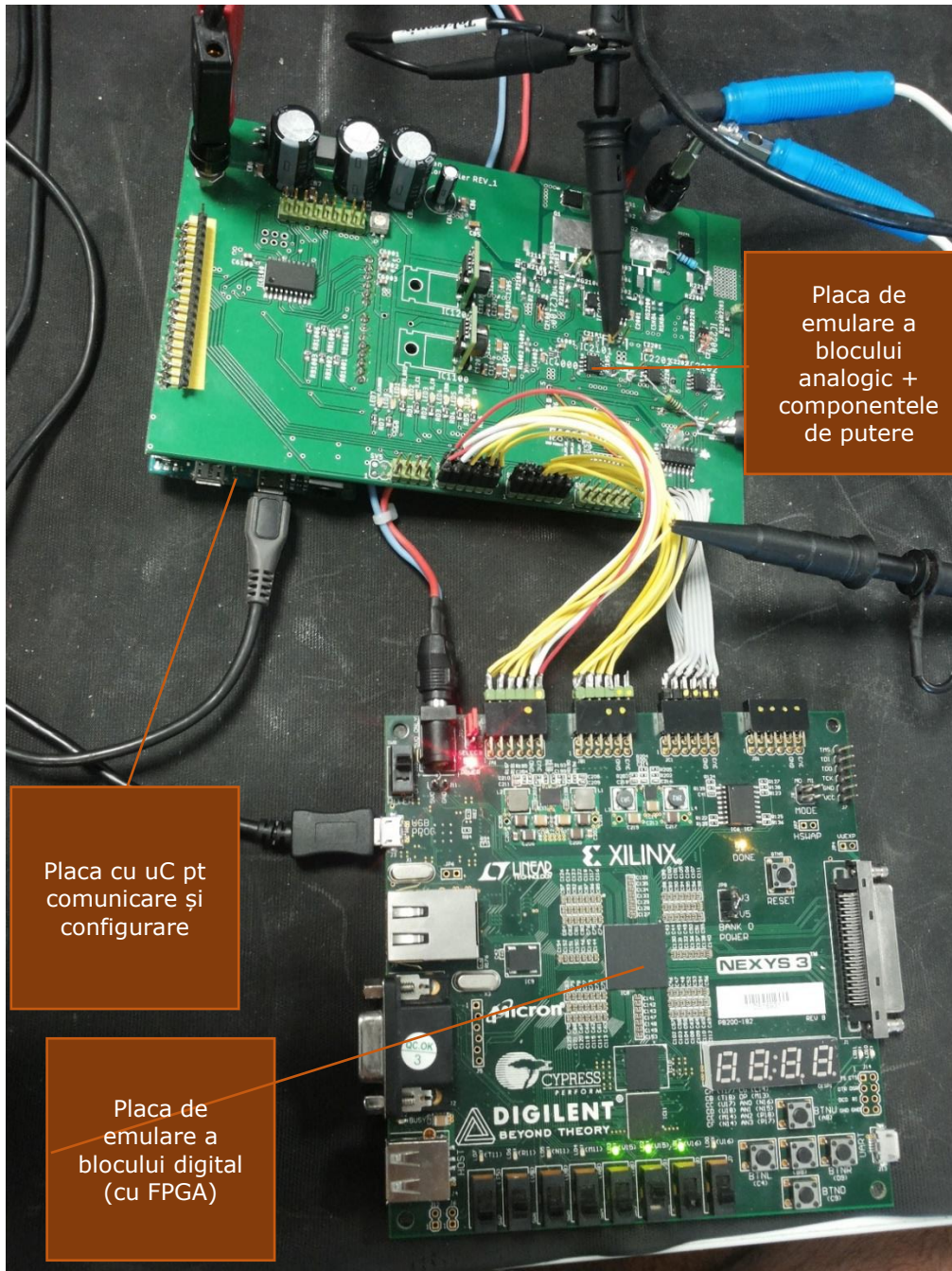


A3a. Placa de emulare a circuitului fizic, construita de B.Komlosi



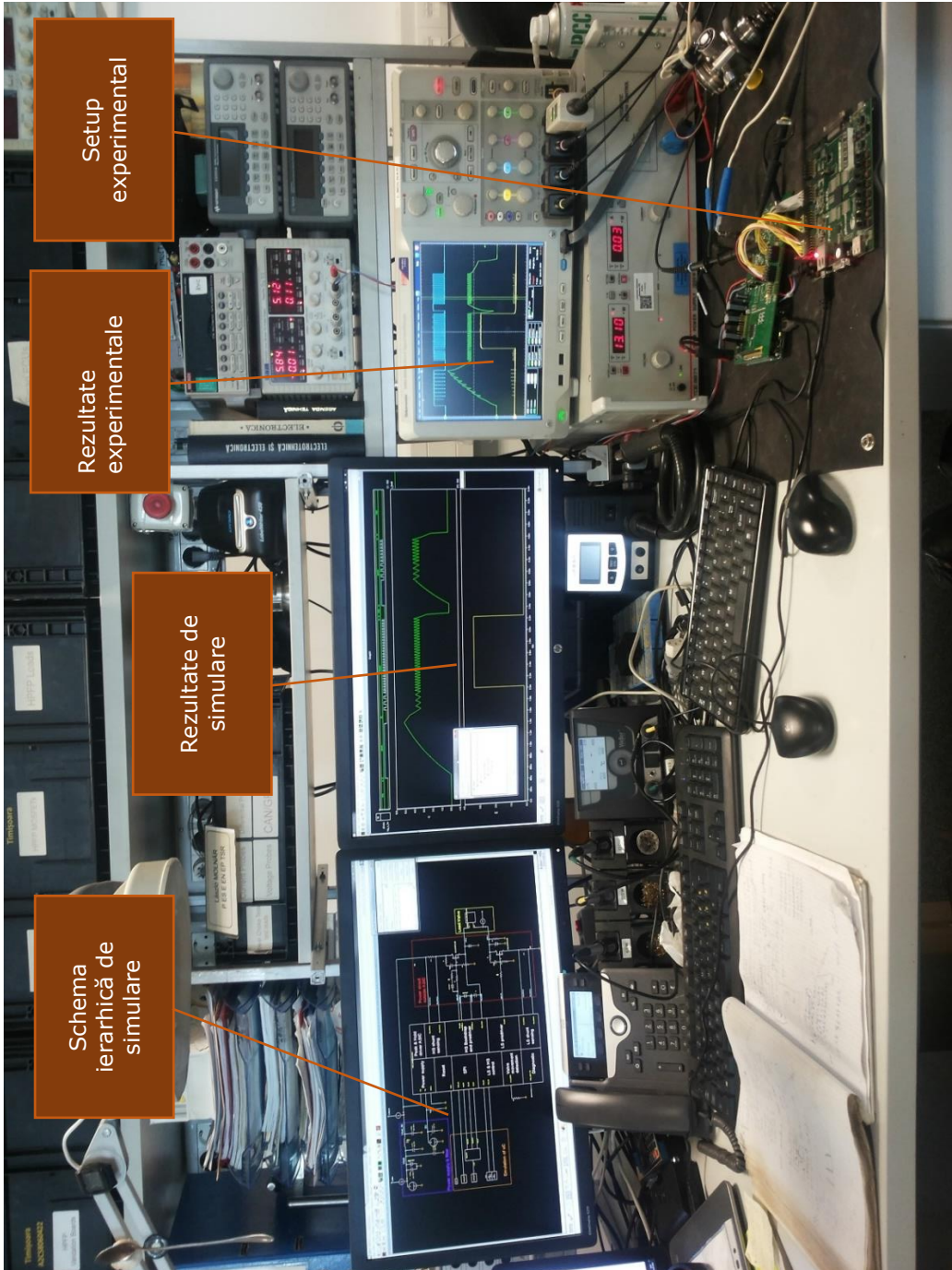
A3b. Placa Arduino Uno(sub placa de emulare); produs achiziționat de la dezvoltator

A4. – Placa cu circuitul fizic și placa cu FPGA

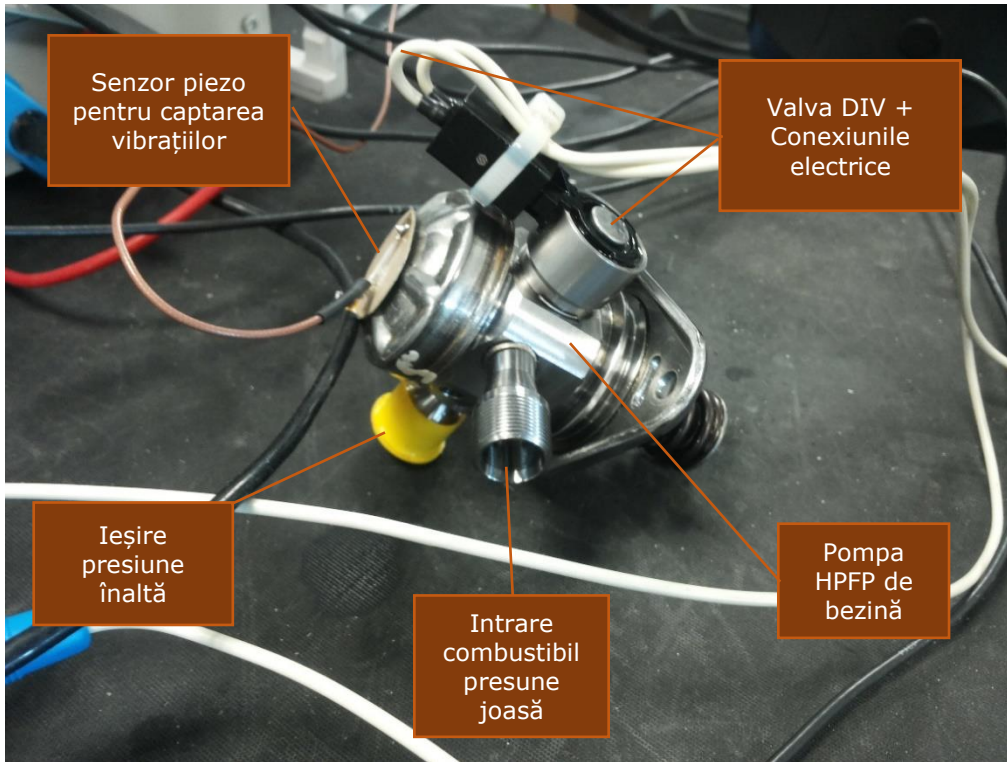


Placa Nexys3; produs achiziționat de la dezvoltator

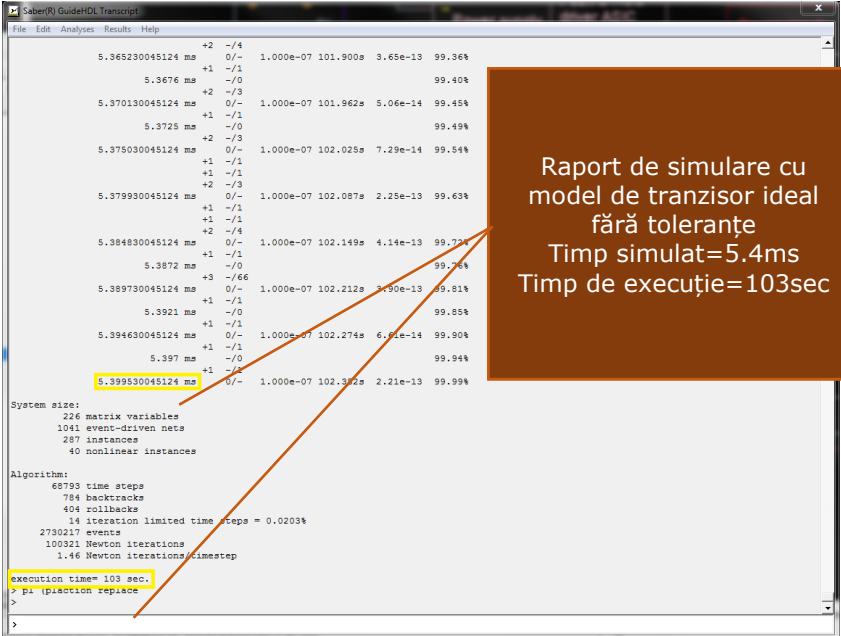
A5. – Mediul de testare și simulare



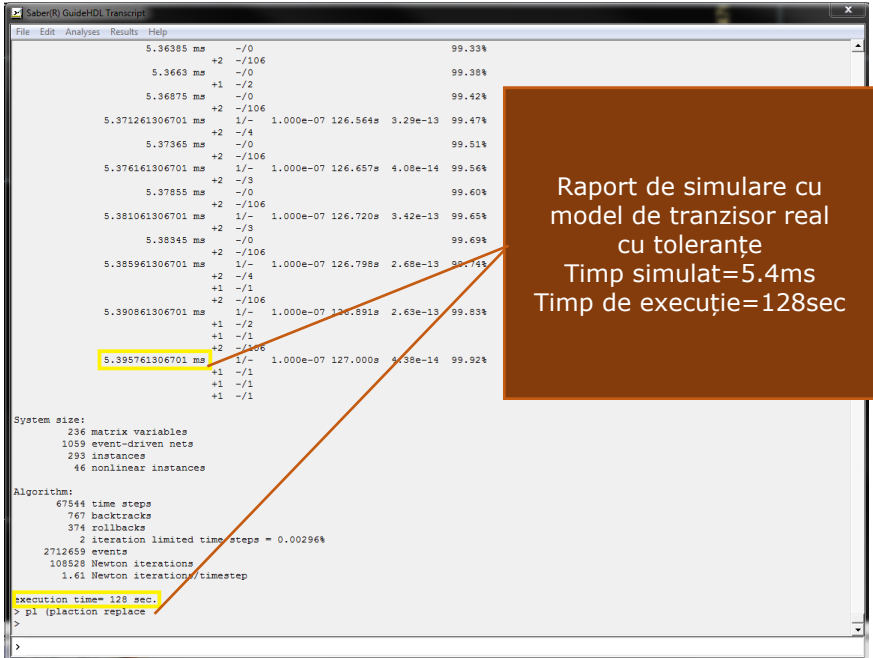
A6. – Pompa de înaltă presiune; Continental 250bar



A7. – Rapoarte de simulare

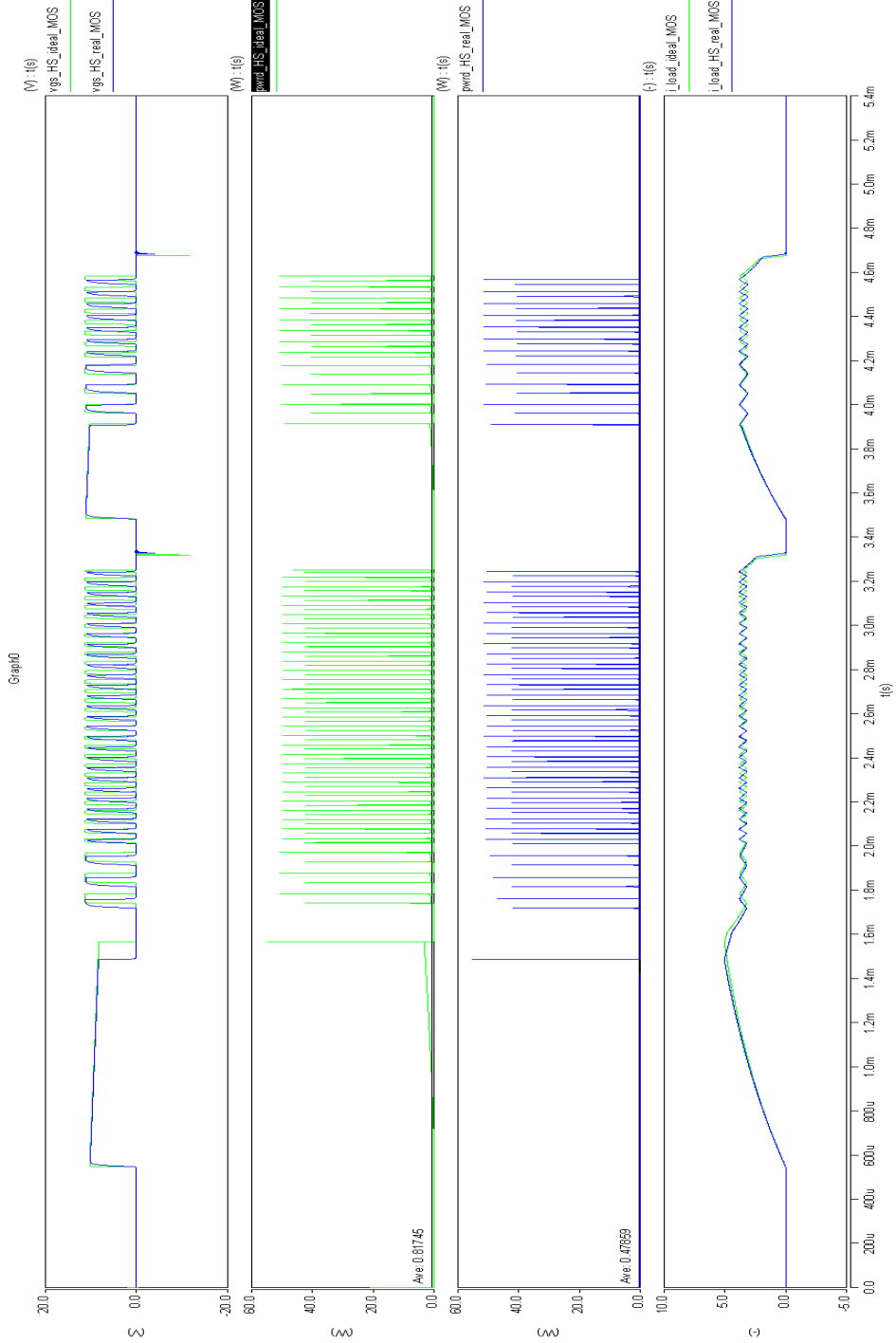


A7a. Raport de simulare cu model de tranzistor ideal



A7b. Raport de simulare cu model de tranzistor real(parametrizat)

A8. – Comparație rezultate tranzistor ideal – tranzistor real



Bibliografie

- [1] P.C Ward and J.R. Armstrong, "Behavioral Fault Simulation in VHDL", 27th ACM/IEEE Design Automation Conference, pp. 587-593, 1990
- [2] O'Neill, M.D., D.D. Jani, C.H. Cho and J.R. Armstrong, "BTG: A Behavioral Test Generator", CHDL 1989, Elsevier Science Publishing Co., New York, New York, June 1989, pp. 347-361
- [3] C. H. Cho, "A Chip Level Fault Coverage Experiment," Research Report, Chip Level Modeling Group, E.E.Dept., Virginia Polytechnic Institute and State University, August 1986
- [4] Chien-In Henry Chen and Sathi Perumal, "Analysis of the Gap between Behavioral and Gate Level Fault Simulation", ASIC Conference and Exhibit, 1993. Proceedings, Sixth Annual IEEE International, pp. 144-147, 1993
- [5] T. Noh , C. I. Chen and S. Chung, "Behavioral fault simulation and ATPG, system for VHDL", ASIC Conference and Exhibit, 1994 Proceedings, Seventh Annual IEEE International, pp. 412-416, 1994
- [6] C. I. H. Chen and T. H. Noh, "VHDL behavioral ATPG and fault simulation of digital systems", IEEE Transactions on Aerospace and Electronic Systems (Volume:34 , Issue: 2), pp.428-447, 1998
- [7] Chien-In Henry Chen "Behavioral test generation/fault simulation", IEEE Potentials (Volume:22 , Issue: 1), pp.27-32, 2003
- [8] R. J. Hayne , "Presynthesis test generation using VHDL behavioral fault models", Southeastcon, 2011 Proceedings of IEEE, pp. 264-267, 2011
- [9] R.J. Hayne and B.W. Johnson, "Behavioral Fault Modeling in a VHDL Synthesis Environment," Proceedings VLSI Test Symposium, April 1999, pp. 333-340.
- [10] J.A. Abraham and K. Fuchs, "Fault and Error Models for VLSI," Proceedings of the IEEE, Vol. 74, No. 5, May 1986, pp. 639-654.
- [11] M.C. Hansen and J.P. Hayes, "High Level Test Generation using Physically-Induced Faults," Proceedings VLSI Test Symposium, May 1995, pp.20-28
- [12] J. Papanuskas, "IEEE VHDL 1076.1: mixed-signal behavioral modeling and verification in view of automotive applications", IEEE VHDL 1076.1: mixed-signal behavioral modeling and verification in view of automotive applications, pp.252-257, 1997
- [13] C. Marino, M. Forliti, A. Rocchi, A. Giambastiani, F. Iozzi, M. De Marinis, L. Fanucci, "Mixed signal behavioral verification using VHDL-AMS", Research in Microelectronics and Electronics, 2005 PhD (Volume:2), pp. 115-118, 2005
- [14] T. Cecchini, T. Baldetti, L. Fanucci, A. Rocchi, "Efficient test environment for multi-level simulations of mixed-signal systems on chip", Industrial Electronics and Applications (ISIEA), 2011 IEEE Symposium on, pp. 69-74, 2011
- [15] V. Jusas; T. Neverdauskas, "Stimuli generator for testing processes in VHDL", NORCHIP, 2014, pp.1-4, 2014
- [16] V. Jusas and T. Neverdauskas, "Combining Software and Hardware Test Generation Methods to Verify VHDL Models" ITC, vol. 42, no.4, pp. 362-368, 2013.
- [17] A. Andrews, A. O'Fallon, and T. Chen, "RUBASTEM: a method for testing VHDL behavioral models" in High Assurance Systems Engineering, 2004. Proceedings. Eighth IEEE International Symposium on, 2004, pp. 187-196.
- [18] Verification Methodology Manual 1.2 Standard Library User Guide, Version E-2011.03, March 2011
- [19] Verification Methodology Manual for SystemVerilog, Hunter, Alan, Nightingale, Andy, Cerny, Eduard, Bergeron, Janick, Springer (2005), ISBN 10: 0028600789
- [20] Universal Verification Methodology 1.1 User's Guide, May 18, 2011 http://accelera.org/images/downloads/standards/uvm/uvm_users_guide_1.1
- [21] Open Verification Methodology 2.0 <https://www.doulos.com/knowhow/sysverilog/ovm/> - last visited June, 2019
- [22] **L. Molnar** and A. Gontean, "Fault simulation methodes," 2016 12th IEEE International Symposium on Electronics and Telecommunications (ISETC), Timisoara, 2016, pp. 194-197
- [23] **L. Molnar** and A. Gontean, "Functional Simulation methodes," 2016 12th IEEE International Symposium on Electronics and Telecommunications (ISETC), Timisoara, 2016, pp. 198-202.
- [24] Jaehyun Park, Kyeongchan Ra, Younggwon Lee and Sungjoon Park, "Development of mixed signal ESC system on chip", 2015 IEEE 11th International Conference on Power Electronics and Drive Systems, Pages: 143 - 147, DOI: 10.1109/PEDS.2015.7203571, 2015

- [25] Eric Chesters, "Role of the verification team throughout the ASIC development life cycle", 2009 46th ACM/IEEE Design Automation Conference, Pages: 216 - 219, DOI: 10.1145/1629911.1629971, 2009
- [26] Fei Gong, Meenal Vaidya, Rishvanth Kora, Daniel Harshbarger, Brad Ulery and William Meyer, "FPGA based prototype verification in automotive mixed signal integrated circuit development", 2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS), 2013
- [27] Andreas Mauderer, Marvin Freier, Jan-Hendrik Oetjens and Wolfgang Rosenstiel, "Efficient digital design for automotive mixed-signal ASICs using simulink", 2012 IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), Pages: 372 - 377, DOI: 10.1109/DDECS.2012.6219090, 2012
- [28] S. Alpe, G. Botto and M. De Giuseppe, "FPGA based ASIC fast prototyping system for new generation piezo injectors control units", 6th Conference on Ph.D. Research in Microelectronics & Electronics, Pages: 1 - 4, 2010
- [29] Alejandro Cook, Dominik Ull, Melanie Elm, Hans-Joachim Wunderlich, Helmut Randoll and Stefan Döhren, "Reuse of Structural Volume Test Methods for In-System Testing of Automotive ASICs", 2012 IEEE 21st Asian Test Symposium, Pages: 1200 - 1203, DOI: 10.1109/MWSCAS.2013.6674869, 2013
- [30] Makoto Ishikawa, D. J. McCune, George Saikalas and Shigeru Oho, "CPU Model-Based Hardware/Software Co-design, Co-simulation and Analysis Technology for Real-Time Embedded Control Systems", 13th IEEE Real Time and Embedded Technology and Applications Symposium (RTAS'07), Pages: 3 - 11, DOI: 10.1109/RTAS.2007.9, 2007
- [31] R. R. Colenbrander, A. S. Damstra, C. W. Korevaar, C. A. Verhaar and A. Molderink, "Co-design and Implementation of the H.264/AVC Motion Estimation Algorithm Using Co-simulation", 2008 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools ,Pages: 210 - 215, DOI: 10.1109/DSD.2008.63, 2008
- [32] Dai Araki and Hidehisa Takamizawa, "Co-simulation of multi-ECU hardware/software system using timing back-annotation method", 2009 International SoC Design Conference (ISOCC), Pages: 380 - 383, DOI: 10.1109/SOCCDC.2009.5423837, 2009
- [33] Anton Pirker-Fruhauf, Karsten Schonherr, Arnaud Laroche and Georg Pelz, "Worst-case modeling and simulation of an automotive throttle in VHDL-AMS", 2007 IEEE International Behavioral Modeling and Simulation Workshop, 148 - 152, DOI: 10.1109/BMAS.2007.4437542, 2007
- [34] **L. Molnar** and A. Gontean, "ASIC validation architectures: Hardware versus simulation," 2017 21st International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, 2017, pp. 671-676.
- [35] A. Sarwar, C. Sankavaram and X. Lu, "Control adaptation approach for fault detection and isolation in SIDI high pressure fuel pump," 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), Dallas, TX, 2017, pp. 117-123.
- [36] D. Spegar, Timothy & Chang, Shi-Ing & Das, Sudhakar & Norkin, Eugene & Lucas, Robert. (2009). An Analytical and Experimental Study of a High Pressure Single Piston Pump for Gasoline Direct Injection (GDi) Engine Applications. 10.4271/2009-01-1504.
- [37] <https://www.continental-automotive.com/en-gl/Passenger-Cars/Powertrain/Gasoline-Technologies/Injection-System/Gasoline-High-Pressure-Pump> - last visited June,2019
- [38] **L. Molnar** and A. Gontean, "Hardware Difficulties and Improvements for High Pressure Fuel Pump Solenoid Valve Noise Cancellation," 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME), Iasi, 2018, pp. 84-89.
- [39] **L. Molnar**, A. Gontean, A. Kotlar and P. Svasta, "Simulation model for automotive high pressure fuel pump," 2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME), Constanta, 2017, pp. 231-234
- [40] Tarek A. Tutunji a, Ashraf Saleem "A methodology for identification and control of electro-mechanical actuators", MethodsX, Volume 2, 2015, pp. 219-231
- [41] Alessandro di Gaeta, Giovanni Fiengo, Angelo Palladino, Veniero Giglio, "A control oriented model of a Common-Rail System for Gasoline Direct Injection Engine", CDC, Shanghai, China, 2009, pp. 6614 - 6619
- [42] Yoichi Iihoshi, Shin Yamauchi, Ryusei Miura, Yoshikuni Kurashima, Toshio Hori," Model-Based Control for High-Pressure Fuel Pumps", ICROS-SICE, Fukuoka, Japan, 2009, pp. 4120-4123.
- [43] Kraft T., Chia T. K. B., Sassler W. (2013), European Patent No. DE102013207162 (A1), Retrieved from Google Patents.
- [44] Chia T. K. B., Kraft T., Bodensteiner A., Sassler W (2016), Europea Patent No. DE102016219890 (B3), Retrieved from Google Patents.
- [45] Sakamoto Y. (2012), U.S. Patent No. US20130243608 (A1), Retrieved from Google Patents.
- [46] Terry J. Ahrendt, Stephen G. Abel (1999), US Patent No. US6211665B1, Retrieved from Google Patents.

-
- [47] Yuuki Sakamoto (2013), US Patent No. US20160186707A1, Retrieved from Google Patents.
 - [48] Gregory J. Smith, Glenn Chance Dunlap, III, Jeremy A. Carlson, Igor Furlan (2006), US Patent No. US7541795B1, Retrieved from Google Patents.
 - [49] Vadim V. Ivanov, David R. Baum (2001), US Patent No. US6807040B2, Retrieved from Google Patents.
 - [50] Gustavo James Mehas, Atul Wokhlu, Naveen Jain, Xiaole Chen (2008), US Patent No. US20120257312A1, Retrieved from Google Patents.
 - [51] Shengyuan Li, Akihiko Doi (2009), US Patent No. US8009546B2, Retrieved from Google Patents.
 - [52] Jorg Remele, Andreas Schneider (1994), US Patent No. US5650909A Retrieved from Google Patents.
 - [53] Thomas Wolfgang, NehlSuresh Gopalakrishnan, Avoki M. Omekanda, Chandra S. Namuduri (2014), US Patent No. US9777686B2, Retrieved from Google Patents.
 - [54] Shengyuan Li, Akihiko Doi (2009), US Patent No. US8009546B2, Retrieved from Google Patents.
 - [55] Suresh Gopalakrishnan, Thomas Wolfgang Nehl, Chandra S. Namuduri, Avoki M. Omekanda (2015), US Patent No. US9863355B2, Retrieved from Google Patents.
 - [56] Scott E. Parrish, Ronald J. Zink (2014), US Patent No. US9435281B2, Retrieved from Google Patents.
 - [57] Yuki Sakamoto (2012), Japan Patent No. JP5692131B2, Retrieved from Google Patents.
 - [58] Takahisa Natori, Tomoyuki Takagawa, Hiroshi Yamaguchi, Yasuo Hirata (2015), US Patent No. US10161342B2, Retrieved from Google Patents.
 - [59] Gopichandra Surnilla, Daniel Dusa, Joseph Lyle Thomas, Joseph Norman Ulrey (2008), US Patent No. US8091530B2, Retrieved from Google Patents.
 - [60] Ross Dykstra Pursifull, Paul Zeng (2013), US Patent No. US20150090227A1, Retrieved from Google Patents.
 - [61] Kenichiro Nakata, Koji Ishizuka (2008), US Patent No. US7747377B2, Retrieved from Google Patents.
 - [62] Yoshikuni Kurashima, Hori Toshio, Masahiro Toyohara Japan Patent No. JP5085483B2, Retrieved from Google Patents.
 - [63] Takashi Okamoto, Hiroyuki Yamada, Kousaku Shimada, Koji Matsufuji (2002), US Patent No. US7299790B2, Retrieved from Google Patents.
 - [64] Takashi Okamoto (2008), US Patent No. US8079345B2, Retrieved from Google Patents.
 - [65] Yasuo Hirata (2009), US Patent No. US20100043754A1, Retrieved from Google Patents.