Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 51(65), Fascicola 1-2, 2006

Logic Analyzer for Development Systems Equipped with AT89C51 Family Microcontroller

Petruţ Duma[1]

Abstract – The paper describes the hardware structure of the interface used by a logic analyzer for testing development systems equipped with AT89C51 family microcontroller. The interface is connected to a command and control unit equipped with I80C51 microcontroller and to a personal computer. The command program perform execution and testing functions for instructions run on a tested system, instruction display functions at instruction cycle, semi-cycle, state, phase and clock levels, initialisation function for the tested microcontroller, as well as other general purpose functions.
Keywords: logic analyzer, development system, ATMEL microcontroller, command software.

I. INTRODUCTION

The basic structure of a logic analyzer for a development system equipped with a microcontroller from AT89C51 family is shown in fig.1.

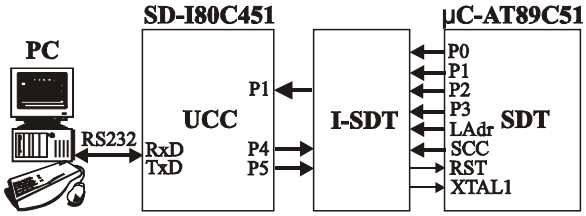

Fig. 1.

The logic analyzer consists of a command and control unit (UCC) - basically a development system (SD) equipped with microcontroller INTEL 80C51, the interface for the tested development system (I-SDT) and the tested development system itself (SDT), equipped with a microcontroller from AT89C51 family. The command and control unit communicates through the serial interface with a personal computer (PC).

II. THE STRUCTURE OF THE TESTED DEVELOPMENT SYSTEMS

The development system based on the AT89C51 microcontroller has the structure shown in fig.2
Microcontroller AT89C51 is connected to a circuitry including a 11,0592 MHz quartz crystal and two 30pF capacitors for the clock oscillator, an RC group (10 KΩ and 10 µF), a 1N4148 diode and a switch K for initialisation, and two capacitors (100 nF and 10 µF) for decoupling.
The system has external program memory and data memory, the pin /EA being therefore connected to logical 0. In order to address the external memory, the P2 port contains the high part of the address bus, while port P0 holds the low part of the address bus, multiplexed with the data bus. The low part of the address bus is latched using an 8-bit external register (74373), on the descending edge of the ALE signal. In this way the low part of the address bus is de-multiplexed from the data bus, port P0 standing subsequently for data bus. The EPROM memory (27256) has a 32 KB capacity and is addressed in the memory space ranging from 0000H to 7FFFH by connecting the address bus BA_{15} to /CE and the signal /PSEN./RD to /OE. This is designed to make sure that programs are executed from the external memory and that data is also read from that memory.
The SRAM (55257) has a capacity of 32 KB and is addressed in the memory space from 8000H to FFFFH by connecting the inverted address line BA_{15} to /CE and the signal /PR./WR=/PSEN./RD./WR to /OE. This connection scheme is designed to allow the execution of the programs from the external SRAM memory, but also the storage/retrieval of data from that same memory.
The logical levels from input (RxD)/output (TxD) of the serial interface are converted using circuit MAX232 in order to perform communication on a line with a personal computer.

III. THE INTERFACE FOR THE TESTED DEVELOPMENT SYSTEM

The tested development system has a large number of digital signals that are to be processed by the command and control unit through interface I-SDT; these digital signals are:
- port P0 – provides the low part of the address bus multiplexed with the data bus;
- port P1 – used as input/output port;
- port P2 – provides the high part of the address bus;

[1] Technical University „Gh.Asachi" Iaşi, Faculty of Electronic and Telecomunications,
Bd. Copou Nr. 11, Iaşi, e-mail: pduma@etc.tuiasi.ro

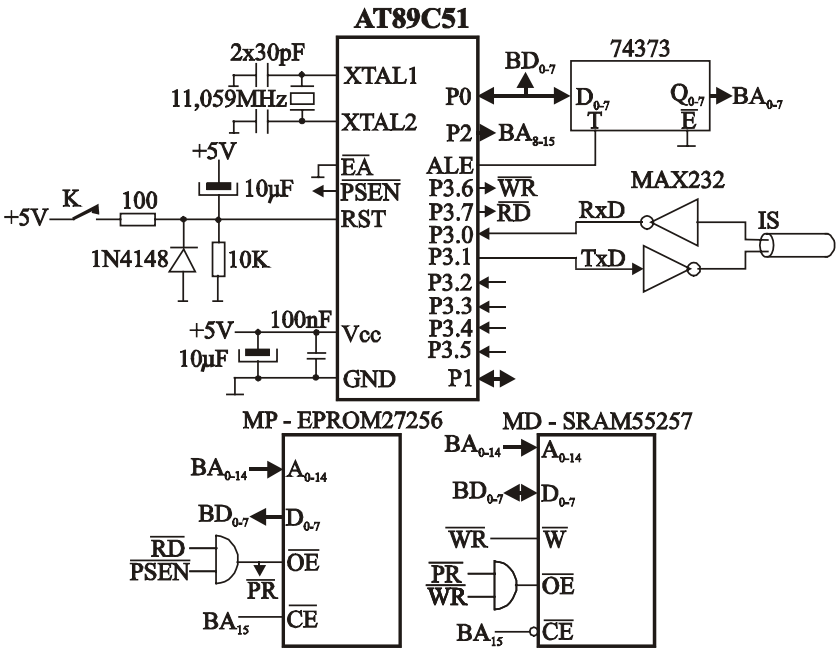

Fig. 2.

- port P3 – provides input RxD and output TxD of the serial interface, inputs for external interrupts /INT0, /INT1, inputs of counters T0, T1, outputs for the command signals /RD (read from the external data memory) and /WR (write in the external data memory);
- the low part of the address bus LAdr;
- the command and control signals SCC of the tested microcontroller: /EA (external program memory selection), ALE (de-multiplex of the low part of address bus from data bus), /PSEN (read the instruction code from the external program memory), XTAL1, XTAL2 (the test oscillator signals), Vcc and GND (microcontroller power supply).

Other tested development systems equipped with other types of microcontrollers from the same family might have in addition two ports, P4 and P5, used as input/output ports.

The command and control unit processes these signals using a 64-input analog multiplexer (fig. 3.).

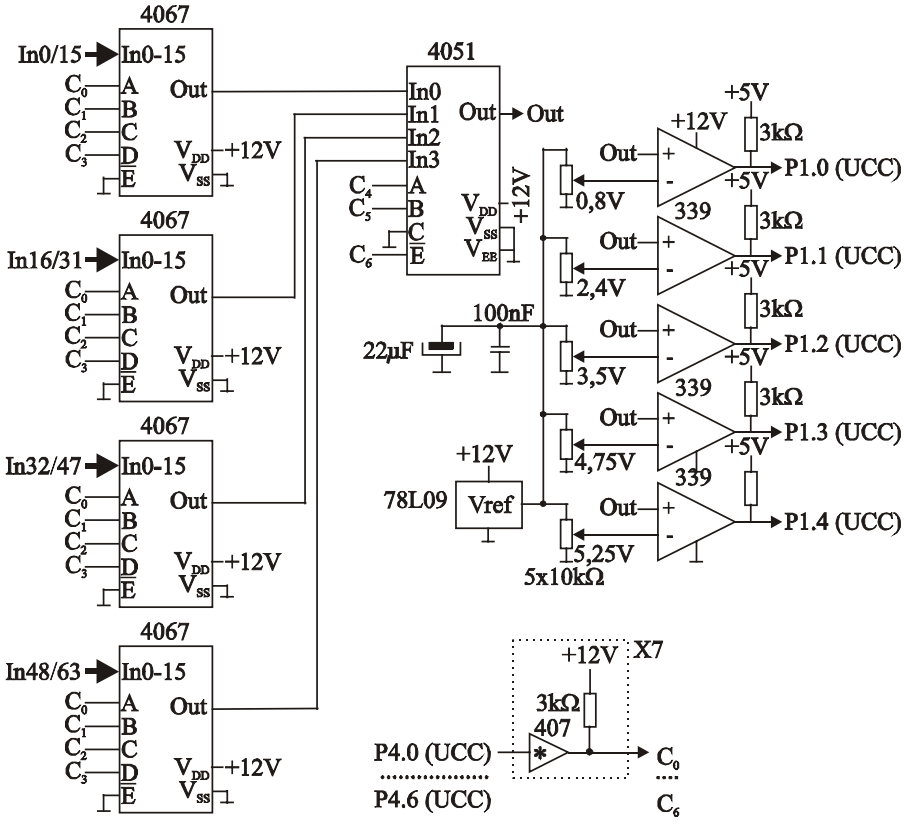

Fig. 3.

The multiplexer consists of four 4067 analog multiplexers with 16 outputs and one 4051 analog multiplexer with 8 inputs. The address of the 64-input analog multiplexer is provided by port P4 of the command and control unit through buffer 7407 that have the role to translate voltage levels. The multiplexer inputs are connected to the digital signals of the tested development system according to table 1.

Table 1

	Input MUX	Digital signal
1	In 0	P0.0 = BA_0 / BD_0
2	In 1	P0.1 = BA_1 / BD_1
…	…..	…….
8	In 7	P0.7 = BA_7 / BD_7
9	In 8	P1.0
10	In 9	P1.1
…	……	……
16	In 15	P1.7
17	In 16	P2.0 = BA_8
18	In 17	P2.1 = BA_9
…	…….	……..
24	In 23	P2.7 = BA_{15}
25	In 24	P3.0 = RxD
26	In 25	P3.1 = TxD
27	In 26	P3.2 = /INT0
28	In 27	P3.3 = /INT1
29	In 28	P3.4 = T0
30	In 29	P3.5 = T1
31	In 30	P3.6 = /WR
32	In 31	P3.7 =/RD
33	In 32	BA_0
34	In 33	BA_1
…	……	……
40	In 39	BA_7
41	In 40	SCC.0 = /EA
42	In 41	SCC.1 = ALE
43	In 42	SCC.2 = /PSEN
44	In 43	SCC.3 = XTAL2
45	In 44	SCC.4 = XTAL1
46	In 45	SCC.5 = RST
47	In 46	SCC.6 = Gnd
48	In 47	SCC.7 = Vcc
49	In 48	P4.0
50	In 49	P4.1
…	……	……
56	In 55	P4.7
57	In 56	P5.0
58	In 57	P5.1
…	……	……
64	In 63	P5.7

The digital signals can have correct or incorrect voltage levels; specifically, the logical 0 level must be lower that 0.8 V, logical 1 level must generally be higher that 2.4 V, except for logical 1 of signals RST and XTAL1 which should be no less than 3.5 V, while the power supply voltage must range between 4.75 V and 5.25 V.

In order to validate these conditions, the hardware structure required must compare the output Out of the analog multiplexer to voltage levels 0.8 V, 2.4 V, 3.5 V, 4.75 V and 5.25 V (fig.3).

This hardware structure consists of 339 comparators that have their inputs In+ connected to output Out of the analog multiplexer, and the In- inputs the standard comparison voltages delivered by a voltage reference adequately divided using multi-tour potentiometers.

The comparators' outputs are connected to port P1 of the command and control unit, and the logical levels obtained can be interpreted as in table 2.

Table 2

P1.4	P1.3	P1.2	P1.1	P1.0	Describe
0	0	0	0	0	0 logical (< 0.8V)
0	0	0	1	1	1 logical (>2.4V)
0	0	1	1	1	1 logical (> 3.5V)*
0	1	1	1	1	4.75V < Vcc < 5,25V
any other combination					wrong level

* for RST and XTAL1 signals

The interface must provide the clock signal and the initialisation signal for the tested development system (fig.4).

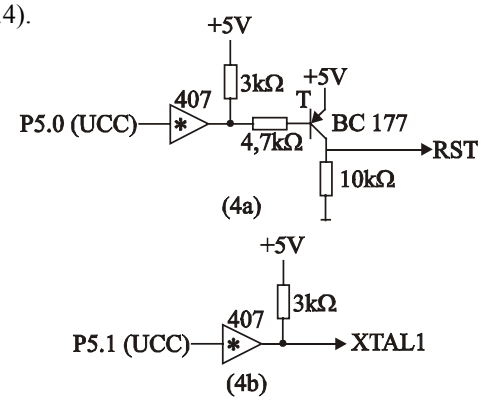

Fig. 4.

An open-collector buffer (7407) is used in order to provide the clock signal, delivering a logical 1 voltage higher that 3.5 V. The command of the clock oscillator from the tested system is applied to XTAL1 using bit P5.1 of the command and control unit.

The quartz crystal must be removed from the tested development system in order to apply a software-generated clock.

The initialisation of the tested development system is made using an open-collector buffer (7407) and a transistor BC177. A logical 0 to the buffer's input, applied by bit P5.0 of the command and control unit determines transistor T to saturate and a voltage level of +5V-U_{CEsat} (higher than 3.5V) is applied to input RST of the microcontroller. It is required that the initialisation signal is active for at least 24 clock periods. A logical 1 applied to the buffer's input determines transistor T to block and a logical 0 to be delivered to input RST after 10 µF capacitor C of the

initialisation circuit is loaded. The microcontroller of the tested development system terminates the initialisation cycle, and then starts the execution of the program from address 0000H of the external program memory.

IV. THE EXECUTION OF THE INSTRUCTIONS STORED IN THE EXTERNAL PROGRAM MEMORY

After the microcontroller of the tested development system is initialised based on the signals software-provided by the command and control unit, the instructions stored in the external program memory are executed starting from address 0000H. The AT89C51 microcontroller family has one, two or three-byte long instructions. The first byte always consists of the instruction code, while the following one or two represent data, an address, a relative displacement, etc. An instruction is executing during an instruction cycle (CI) that consists of one, two or four machine cycles (CM) (table 3).

Table 3

No. bytes	No. machine cycles
1	1
1	2
1	4
2	1
2	2
3	2

Any machine cycle consists of six states (S1, S2,... S6), each of them including two phases (P1 and P2). A phase corresponds to one clock period. The command and control unit provides the clock of the tested system by software, and for each level of the clock signal, it loads in the system's memory all the digital signals through the analog multiplexer and the comparators.

The general time diagram including all the command signals for the execution from the program memory of a one, two or three-byte instruction in one, two or four machine cycles is shown in fig. 5. This diagram does not cover the MOVX instruction that addresses the external data memory.

The time diagrams with all command signals for the execution of an instruction stored in the external program memory, that reads or writes data in the external memory (MOVX) are shown in fig. 6 and fig.7.

V. THE COMMAND PROGRAM COMMANDS

The command program is loaded into the command and control unit from the personal computer, through the serial interface and is executed automatically. First it is executed the initialization sequence, which loads the system variables, defines the data memory zone, clears the console screen and displays a program launch message.

The program then enters a loop designed to interrogate the keyboard, during which various user commands may be input. The program collects the user command, tests the command syntax and then executes it. After the user command execution, the keyboard interrogation loop is resumed, waiting for a new command.

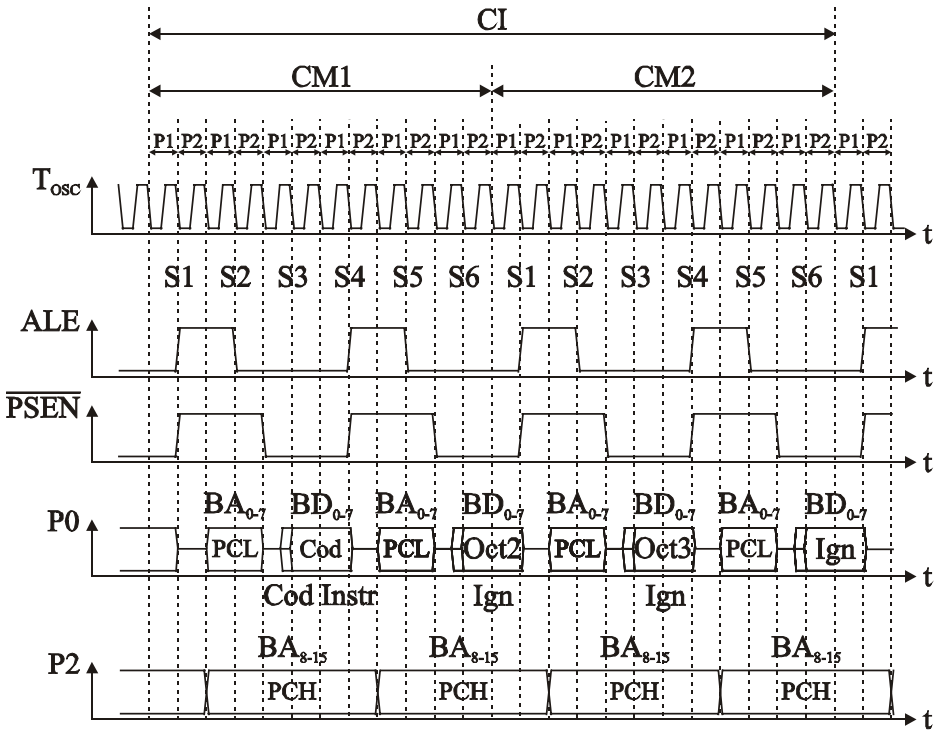

Fig. 5.

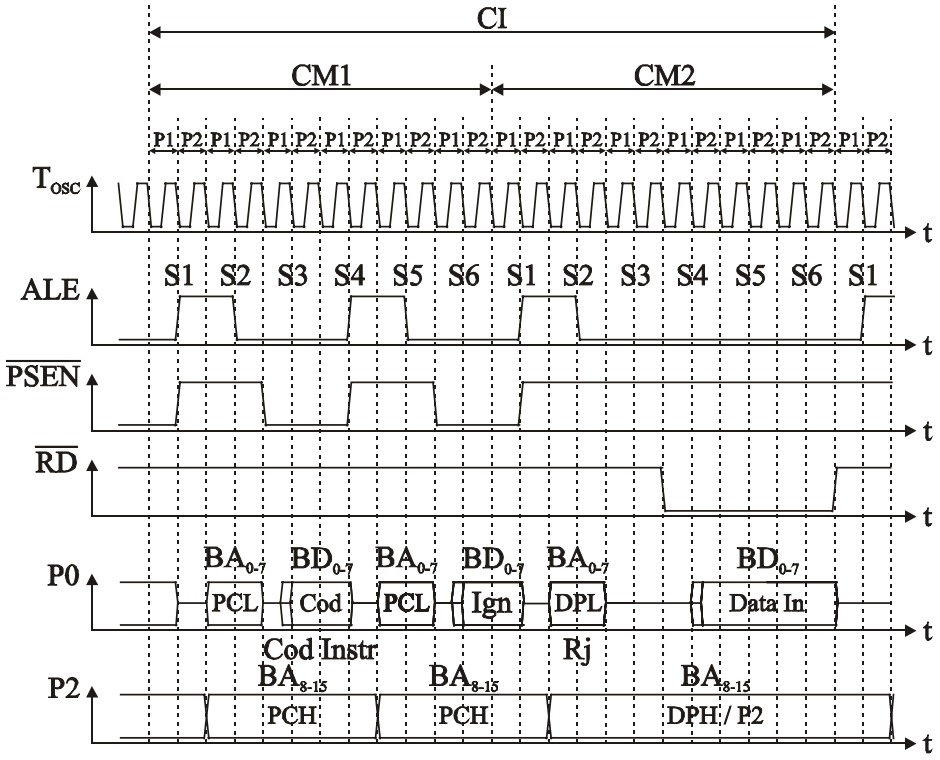

Fig. 6.

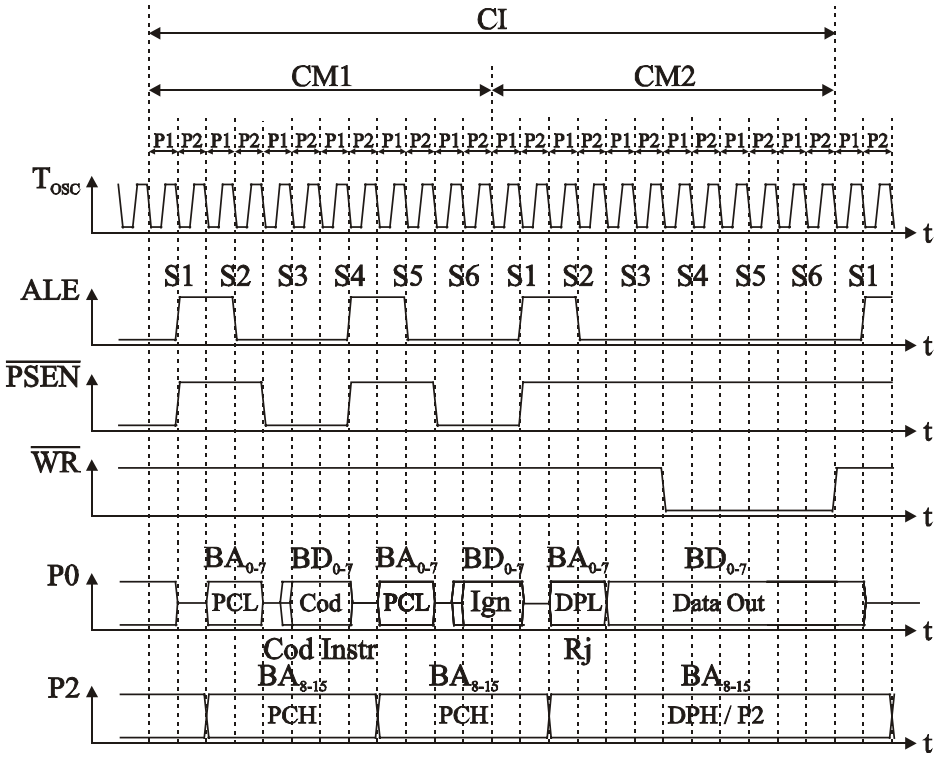

Fig. 7.

The logic analyzer commands syntax and their summary description are briefly presented below.

A. – displays on the console the list of commands accepted by the logic analyzer;

B. – selects the resources of the tested microcontroller that are to be displayed also at bit level;

DI. – displays the last instruction executed (address, instruction code, the second and the third instruction byte, mnemonic);

DC – displays the last instruction executed at semi-cycle level;

DS. - displays the last instruction executed at state level;

DP. - displays the last instruction executed at phase level;
DX. - displays the last instruction executed at clock level;
GX. – executes the current instruction at clock level;
GP. – executes the current instruction at phase level;
GS. – executes the current instruction at state level;
GC. – executes the current instruction at semi-cycle level;
GI. - executes integrally the current instruction;
GNp1 – executes p1 instructions on the tested system;
GAp1 – executes instructions on the tested system up to address p1;
Q. – exits the command program to the monitor program;
R. – initialises the tested microcontroller and executes the instruction stored at address 0000H.
The command program signalises incorrect voltage levels of any signal from the tested system and incorrect activation/de-activation logic of these signals.

VI. CONCLUSIONS

The interface of the logic analyzer described has a minimal hardware structure, it was build in practice and has proven itself a useful testing tool for development systems equipped with microcontroller AT89C51. Development systems can be tested if they are based on microcontrollers from this family and have no more than 6 parallel ports.
The command program uses a memory area of some 10 KB and features particularly useful functions.
The logic analyzer described can also be used to test application systems equipped with any AT89C51 family microcontroller, requiring adequate adjustment of the command program.

REFERENCES

[1]. Căpăţînă O., Proiectarea cu microcalculatoare integrate, Ed. Dacia, Cluj Napoca, 1992.
[2]. Hintz J.K., Tabak D., Microcontrollers. Arhitecture, Implementation and Programming, McGaw Hill, 1993.
[3]. Duma P., Microcontrolerul INTEL 8051. Aplicații. Ed. „TEHNOPRESS", Iași, 2004.
[4]. Peatmann B.J., Design with Microcontrollers, McGraw Hill, 1998.
[5]. Somnea D., Vlăduț T., Programarea în Assembler, Ed. Tehnică, Bucureşti, 1992.
[6]. Sztojanov I., Borcoci E., Tomescu N., Bulik D., Petrec M., Petrec C., De la poarta TTL la microprocesor, Ed. Tehnică, Bucureşti, 1987.
[7]. XXX ATMEL AT89C51 Family Microcontroller, Data Book, 1998.
[8]. XXX Texas Instruments, Data Book, 1992.