MICROCONTROLLER'S PROGRAMMER

Todor Mladenov Mladenov, Vladimir Mladenov Mladenov

Faculty of Communications and Communications Technologies, Faculty of Computer Systems and Control, Technical University of Sofia, 8, Kliment Ochridski St., Sofia - 1000, Bulgaria,

phone: +359 2 965 21 34, e-mail: dsptexas@mail.bg, v_mladenov@mail.bg

Keywords: CPLD, microcontroller, programming, SPI

In the past few years every manufacturer equips its products with the ability of In-Circuit Serial Programming (ISP) for convenience and flexibility. Nowadays, with the presence of programmable logic, digital circuits are more and more easy to build and reconfigurations can be done even during operation. The programmer, that is to be described, is based on a CPLD (complex programmable logic device) matrix from XILINX and utilizes thirty six macrocells. It is capable of programming Atmel's AVR and 8051 architectures based microcontrollers and Microchip's PIC architecture based microcontrollers. A parallel connection with a PC is used. The CPLD matrix is programmed as an application specific serial peripheral interface (SPI). Microcontrollers' different instruction formats and specialities are easily managed because programming depends mainly on PC software. Proposed realization of a highly functional programmer is inexpensive and flexible.

1. INTRODUCTION

Serial method for programming is very popular and widely used nowadays. It simplifies the process of developing, testing and production. It is supported by most of the manufacturers and preferred by customers. This is a reason for future development and improvement that will lead to faster and more reliable serial programming protocols.

2. PROGRAMMING METHOD

Microcontrollers, mentioned above, rely on SPI for serial programming without need of high voltage (more than 5V). For Microchip this is true when a LVP (low voltage programming) method is used. Atmel has only low voltage serial and high voltage parallel programming methods. Both manufacturers use different SPI realizations and instruction formats that are to be shortly described.

2.1 Microchip

PIC microcontrollers use Harvard architecture with fourteen bits instruction word. When programming no external clock signal or quartz crystal is needed. The build in SPI uses two lines - one for data and one for clock. The data line is bidirectional. For programming simple command sequences are used, entered in serial fashion with the data being latched on the falling edge of the clock pulse. Data is send and received with least significant bit (LSB) first. Commands are six bits long and data is fourteen bits long. No Address has to be send. There is no special command for entering programming mode also. Varieties of commands depend on model but some main ones are always available. An example of read from program memory command for PIC16F628 is presented (Fig.1). The data following the command is rearranged in two bytes beginning with start bit and ending with stop bit. Only Tset1 and Thld1 (both are 100 ns) limit the SPI speed.

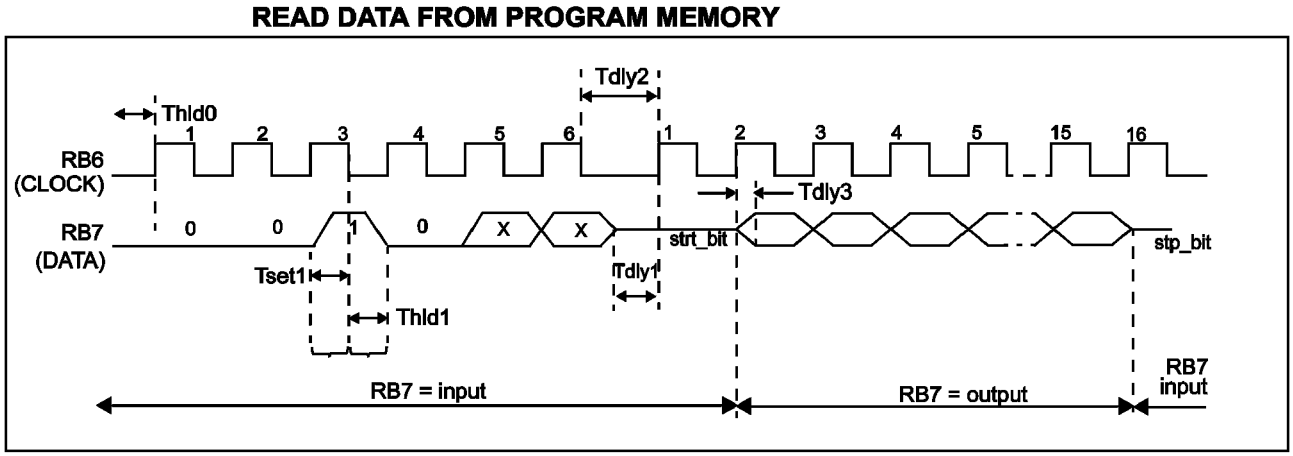

Fig.1

2.2 Atmel

Atmels' AVR and 8051 microcontrollers need either an external clock supplied or crystal to be connected when are to be programmed. The build in SPI has tree lines - two for reading data "MOSI" (master output, slave input) and for writing data "MISO" (master input, slave output), and one clock line. Here the microcontroller is 'slave', because the SPI clock is generated by the programmer. Maximum SPI speed should be at least 4 times slower than microcontrollers clock for AVR and at least 16 times slower for 8051. On Fig.2 are shown some serial programming waveform for AT90S8515. Data is send and received with most significant bit (MSB) first. When writing data to the microcontroller, data is clocked on the rising edge. When reading from the microcontroller, data is clocked on the falling edge. Commands are four bytes long including address and data. A specially dedicated command for entering

programming mode is used. Before programming, flash memory should be erased. User memory space is byte accessible. AVR's instruction word is two bytes long.

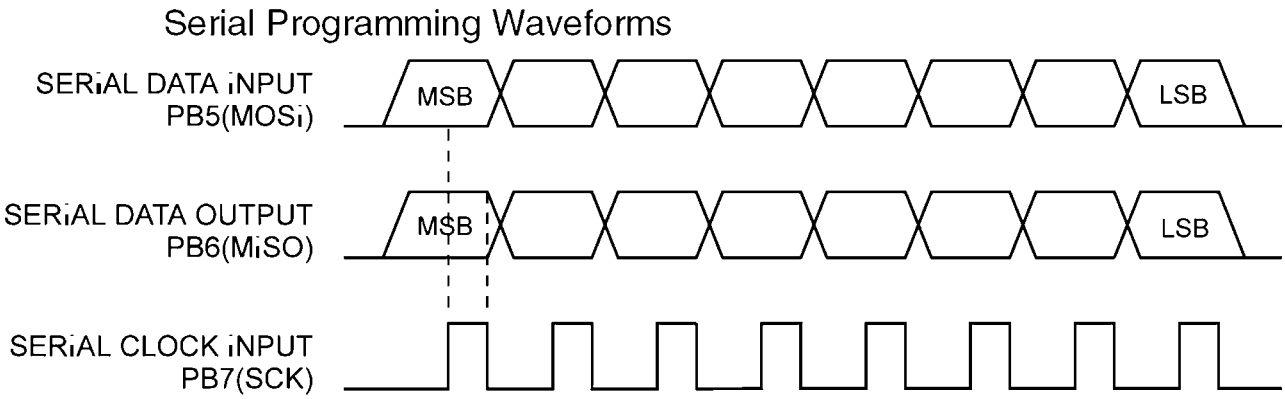

Fig.2

3. REALIZATION

For handling above mentioned special features, an application specific SPI is needed. In order to develop a common programmer supporting both manufacturers' microcontrollers, a XC9536 CPLD from XILINX is used. It has thirty six macrocells and 5V power supply. Internal logic could work with clock signal up to 80 MHz. Its price vary around one dollar and this makes the programmer inexpensive.

3.1 Application specific SPI

The whole available logic in XC9536 is used for creating the scheme presented on Fig.3. It contains two shift registers – a serial to parallel and a parallel to serial, two counters, decoder, multiplexer and some additional logic. The programmer communicates with PC through the parallel port. Data is received from and send to PC byte by byte. Because parallel interface gives only four additional lines for control a decoder is used. The eight bit counter is used for dividing the external clock and for defining the moments when shift registers need to be clocked. Here, for testing, a 20.000MHz external oscillator is used. With slower clock signal this counter could be smaller. The four bit counter and the multiplexer determines whether six or eight bits word will be used. This is needed for PIC microcontrollers. The two shift registers use a common bidirectional data bus for receiving or sending data. The serial to parallel one has tristate buffers to release the bus when PC sends data. When programming a PIC microcontroller, the two lines MOSI and MISO are combined in one. Otherwise a tristate buffer separates them. Both shift registers send or receive data with the MSB firs. In order to program PIC microcontrollers the application software, created especially for this programmer, rearrange data before sending it to the CPLD.

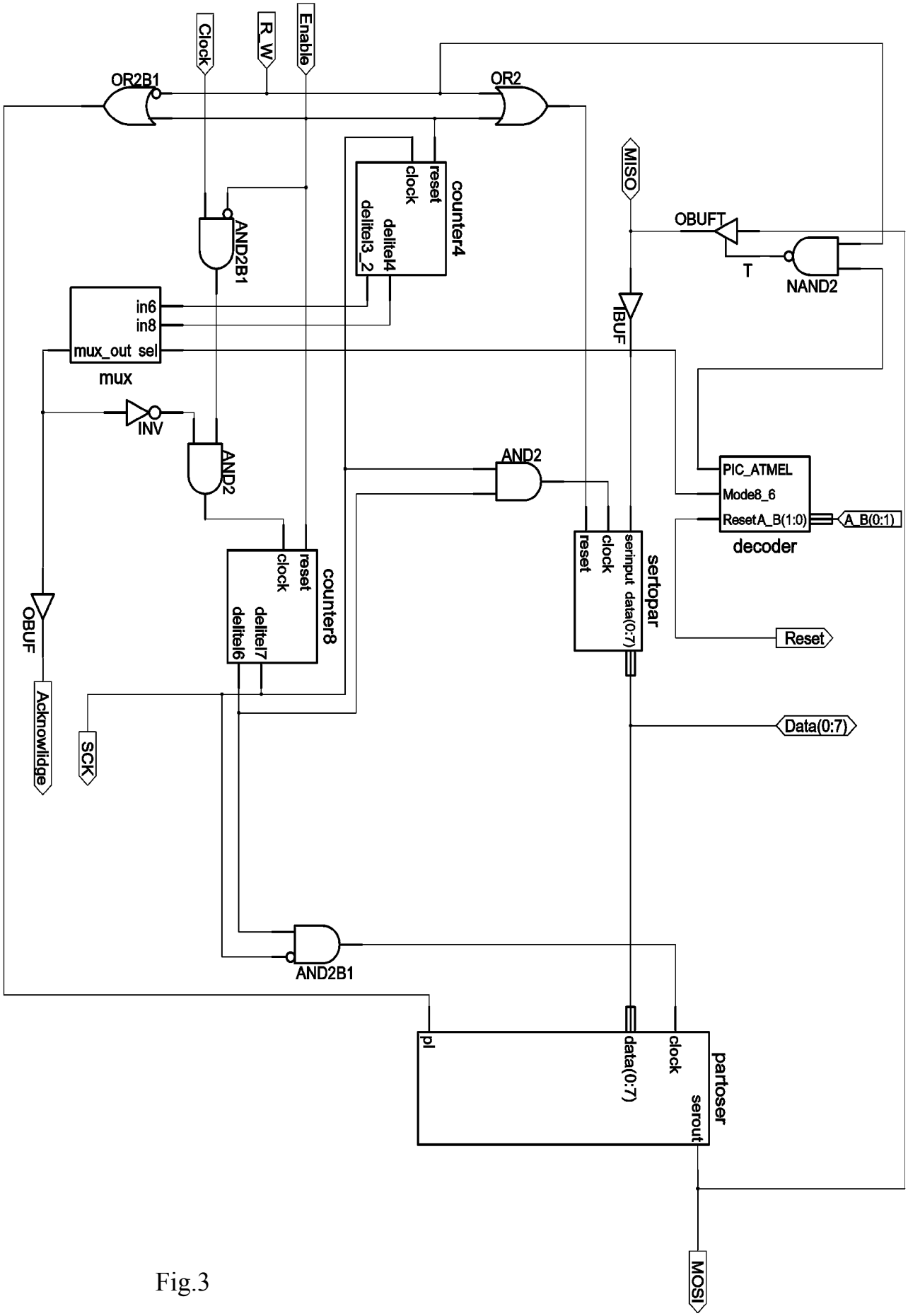
Clock
R_W
Enable
OR2B1
OR2
MISO
OBUFT
T
NAND2
IBUF
AND2B1
reset
clock
delitel4
delitel3_2
counter4
in6
in8
mux_out sel
mux
INV
AND2
PIC_ATMEL
Mode8_6
ResetA_B(1:0)
A_B(0:1)
decoder
reset
clock
serinput data(0:7)
sertopar
Reset
OBUF
Acknowlidge
SCK
reset
clock
delitel7
delitel6
counter8
Data(0:7)
AND2B1
pl
data(0:7)
clock
serout
partoser
MOSI

Fig.3

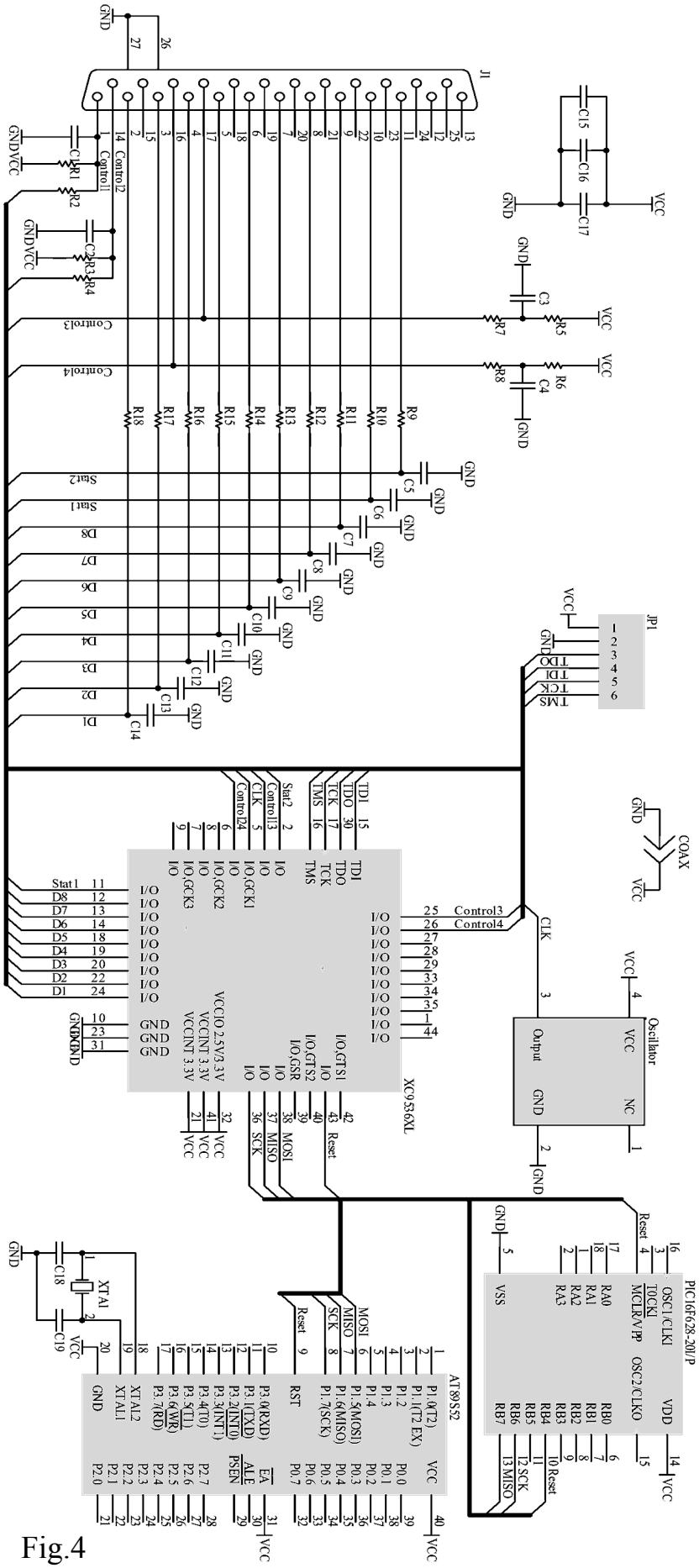
J1
JP1
XC9536XL
Oscillator
COAX
PIC16F628-20/P
AT89S52
Control1
Control2
Control3
Control4
Stat1
Stat2
Reset
MOSI
MISO
SCK
XTAL1
XTAL2
VCC
GND

Fig.4

3.1 PCB Scheme

On Fig.4 is shown the PCB Schema. Two microcontrollers from both manufacturers are presented, for showing how they should be connected. Other models very only in footprints and the connection lines are the same. Special measures for reducing parallel cable noise are taken. A JTAG interface for programming the CPLD is also available. There is only one 5V power supply.

4. CONCLUSION

The main aim for developing compact and multifunctional programmer was achieved. Furthermore, the device functions could be improved and extended for covering other devices by using a bigger CPLD device.

5. REFERENCES

[1] MICROCHIP In-Circuit Serial Programming™ (ICSP™) Guide, http://ww1.microchip.com/downloads/en/DeviceDoc/30277d.pdf

[2] ATMEL 8-bit AVR Microcontroller with 8K Bytes In-System Programmable Flash, http://www.atmel.com/atmel/acrobat/doc0841.pdf

[3] Knud Smed Christensen, Ole Winther, Zdravko Karakehayov, Embedded Systems Design With 8051 Microcontrollers : Hardware and Software, Marcel Dekker Inc., August 1999.