Scholars Journal of Engineering and Technology (SJET) **ISSN 2321-435X (Online)**
Sch. J. Eng. Tech., 2015; 3(7):672-678 **ISSN 2347-9523 (Print)**

(An International Publisher for Academic and Scientific Resources)
www.saspublisher.com

Research Article

PIC12C5xx a Pipeline Processor Design and Implementation on XILINX FPGA Chip

Ahmed Abualsaud, Mohammad Awadh
King Abdulaziz University, Electrical and Computer Engineering Department, Jeddah 21372, Saudi Arabia

***Corresponding author**
Ahmed A. Abualsaud
Email: aajbaa2012@gmail.com

Abstract: This paper describes PIC12C5xx a pipelined processor design and implementation on Xilinx FPAG Chip, which named KAUPIC12. This project is conducted for the Computer Organization Course (EE361) at King Abdulaziz University. The project design has three main stages: one for the assembler design using the Java Eclipse software, the second stage is the single cycle design and the last stage is the pipelined design. The project course was a complete help for the students to understand the operations of pipelined processors and to excel in using the Basys 2 FPGA Board.
Keywords: PIC12C5xx; assembler; single cycle; pipeline.

INTRODUCTION

The title of this paper is "PIC12C5xx a Pipeline Processor Design and Implementation on Xilinx FPAG Chip". PIC is a microcontroller device that is made by Microchip Company; its internal architecture design is based on the registers and gates. These registers and gates are controlled by the CPU which used to arrange the control signals and the communication between the registers. Xilinx is a software tool and a hardware designing company, designing the processor in this project depends completely on this tool. This is the project proposal for the Computer Organization Course (EE361) focusing on design and implementation of any chosen processor by Xilinx FPGA. It contains the design specification that will illustrate the instruction set architecture and the general format that will be undertaken as an essential part of the first processor design. This paper first discusses the Materials and Methodology which contains brief introduction and explanation of Xilinx chip and Basys 2 Board, then the Results and Tables which have the PIC12C5xx assembler by using Java language, instructions set formats, instructions set architecture (ISA), single cycle design, test code, pipeline design, register transfer level (RTL), execution pipeline, pipeline hazard, and resources report of the implementation, and finally conclusion.

MATERIALS AND TOOLS

Software DesignTool

The software part of this paper covered one main software which is Xilinx ISE 14.7 (Integrated Synthesis Environment), it is a software design tool which manufactured by Xilinx for several different functions. These function sum up as follows:

- Analyzing HDL designs
- Enabling the developer to compile their designs
- Performing timing analysis
- Examining RTL diagrams
- Simulating a design's reaction to different stimuli
- Configuring the target device with the programmer

This software environment has two integrated hardware description languages, which are Verilog and VHDL. Verilog is mainly used to model electronic systems and to design and verify digital electric circuits. It is also employed to validate and analog circuits and mixed-signal circuits. VHDL stands for Very high speed integrated circuits Hardware Description Language. It is used in electronic designs to give a detailed description of digital and mixed signal system designs [1].

Hardware Design Tool

The Basys™2 Spartan-3E FPGA Board is a field programmable gate array circuit, which used to design and implement digital circuits. This board has a complete hardware integrated devices, which we can use it to create many electronic circuits without the need of any other components. One feature is that this board works with all versions of the Xilinx ISE software tools. It has the main following specifications [2]:

- Xilinx Spartan 3-E FPGA, 100K gates

- FPGA features 18-bit multipliers, 72Kbits of fast dual-port block RAM, and 500MHz+ operation
- USB 2 full-speed port for FPGA configuration and data transfers (using Adept 2.0 software available as a free download)
- XCF02 Platform Flash ROM that stores FPGA configurations indefinitely
- User-settable oscillator frequency (25, 50, and 100 MHz), plus socket for a second oscillator
- Three on-board voltage regulators (1.2V, 2.5V, and 3.3V) that allow use of 3.5V-5.5V external supplies
- 8 LEDs, 4-digit seven-segment display, four pushbuttons, 8 slide switches, PS/2 port, and a 8-bit VGA port

PIC12C5XXX ASSEMBLER

The Java Eclipse software (high level language) is used in order to program the assembler of PIC12C5xx. The programed assembler of the PIC12C5xx is depended on two factors which are arranging the written code in txt file then write to the new file which is called the mapper, and convert the mapper code to the respectable binary code. One of the advantages is that the assembler can deals with capital/small letters (coding). There is a standard to write a program through this assembler which is the main label, and end label with GOTO loop as shown in fig.1. Also, here is an example of how the assembler is working as shown in table-1.

```
      [ORG 0x00] ;Optional
Main: code
      code
End:  GOTO End   ;End/label
```

Fig-1: Standard writing assembler.

Table-1: Code example

Code (.txt)	Binary (.bin)
ORG 0 ;Start Program at 0x00 address	000001000000
Main: CLRW ;Clear Working Register	000001100011
CLRF 0x03 ;Clear STATUS(0x03) Register	001000000110
MOVF 0x06, 0 ;Read SWs(0x06) to WREG	000000101010
MOVWF 0x0a ;Move WREG to LEDs(0x0a) Register	101000000100
End: GOTO End ;End loop	

INSTRUCTION SET FORMAT

According to the main architecture of PIC12C5xx, the designs of the PIC12C5xx have been designed through three different instruction formats. However, in this paper the formats are the same as the Microchip Company which is illustrated in table-2 [3].

Table-2: Instruction set format

1. Byte-oriented operations		
OPCODE 6-bit [11:6]	D (destination)1-bit [5]	F (file Address) 5-bit [4:0]
2. Bit-oriented operations		
OPCODE 4-bit [11:8]	B (Bit Number) 3-bit [7:5]	F (file Address) 5-bit [4:0]
3. Literal and control operations (except GOTO)		
OPCODE 4-bit [11:8]	K (literal) 8-bit [7:0]	
3. Literal and control operations (GOTO instruction)		
OPCODE 3-bit [11:9]	K (Jump Address) 9-bit [8:0]	

INSTRUCTION SET ARCHITECTURE

The PIC12C5xx contains 33 ISA, but the chosen instructions (24 ISA) are shown in table-3 and the opcode description is shown in table-4 [3].

Table-3: Instruction set architecture

Instruction Format PIC12C5xx	Operation	RTL	12-Bit Opcode	Status Affected
Byte-oriented File	ADDWF f,d	d ← W + F	0001 11df ffff	C,DC,Z
	ANDWF f,d	d ← W AND F	0001 01df ffff	Z
	COMF f,d	d ← F^{-1}	0010 01df ffff	Z
	IORWF f,d	d ← W OR F	0001 00df ffff	Z
	MOVF f,d	d ← F	0010 00df ffff	Z
	RLF f,d	d ← F << C	0011 01df ffff	C
	RRF f,d	d ← F >> C	0011 00df ffff	C
	SUBWF f,d	d ← W - F	0000 10df ffff	C,DC,Z
	SWAPF f,d	d<7:4> ← F<3:0> d<3:0> ← F<7:4>	0011 10df ffff	None
	XORWF f,d	d ← W XOR F	0001 10df ffff	Z
	CLRF f	F ← 0x00	0000 011f ffff	Z
	MOVWF f	F ← W	0000 001f ffff	None
	CLRW -	W ← 0x00	0000 0100 0000	Z
	DECF f, d	d ← F - 1	0000 11df ffff	Z
	INCF f, d	d ← F + 1	0010 10df ffff	Z
Bit-oriented File	BTFSC f,b	skip if (f<b>) = 0	0110 bbbf ffff	None
	BTFSS f,b	skip if (f<b>) = 1	0111 bbbf ffff	None
	BCF f, b	(f<b>) ← 0	0100 bbbf ffff	None
	BSF f, b	(f<b>) ← 1	0101 bbbf ffff	None
Literal and Control	ANDLW k	W ← W + k	1110 kkkk kkkk	Z
	IORLW k	W ← W OR k	1101 kkkk kkkk	Z
	MOVLW k	W ← k	1100 kkkk kkkk	None
	XORLW k	W ← W XOR k	1111 kkkk kkkk	Z
	GOTO k	PC<8:0> ← k	101k kkkk kkkk	None

Table-4: Opcode description

Field	Description	Field	Description
f	Register file address (0x00 to 0x7F)	d	Destination select; d = 0 (store result in W) d = 1 (store result in file register ’f’) Default is d = 1
W	Working register (accumulator)	( )	Contents
b	Bit address within an 8-bit file register	←	Assigned to
k	Literal field, constant data or label	<>	Register bit field

SINGLE CYCLE DESIGN

Referring to next fig.2, contains of the single cycle data path which have Program Counter (PC); 12 bits, Instruction Memory (IM); its size 4Kx12, Control Unit (CU); a combinational circuit because it is a single cycle, Register File (RF); five bits address so its size equals to 32x8, working register (WREG); its size 8 bits, Arithmetic Logic Unit (ALU); two inputs 8 bits and directly connected to the STATUS register, Next Program Counter (Next PC); inputs from CU and ALU to update the PC if it is required, Check Module that can read from STATUS or SWITCH registers (Check ST/SWs); the address number 3 and 6 in RF cannot be accessed and they are placed outside the RF to avoid the multiple writing on the RF, and four multiplexors. Here is an example of how does this system work? First, the program counter is increased, second, the instruction is fetched and the control signals are sent by CU to the system based on the opcode, third the RF has been decoded and the first input to the ALU is either RF value or literal depends on opcode and control signals and the second input to the ALU is the WREG, finally, the results is written to the chosen destination if it is equals to 1 that means the RF is the desired destination otherwise WREG [4].

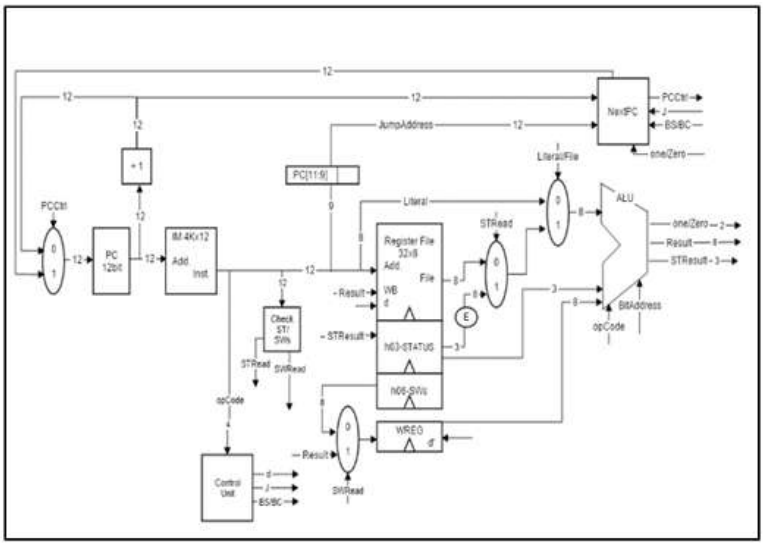

Fig-2: Single cycle data path.

TEST CODE

According to fig.3, it is illustrated the demo that has tested by the Xilinx Tools 14.7. The idea is read from switches and displayed the reading to the LDEs. After that, the shift right process will be started until it is finished. Then the left shift will be operated and finally the main program will be started again.

```
        ORG 0  ;Strat the program from address(0x00)
Main: CLRW  ;Clear WREG
        CLRF 0x03       ;Clear STATUS = RF(0x03)
        MOVF 0x06, 0  ;Read SWs = RF(0x06) to WREG
        MOVWF 0x0a   ;Move WREG value to LEDs = RF(0x0A)
        CLRF 0x0a       ;Clear LEDs
        BSF 0x0a, 7       ;Set bit no.7 of LEDs to 1
Loop1 RRF 0x0a, 1      ;Start Shift Right LEDs with Carry
        BTFSS 0x03, 0  ;Check if the STATUS.Carry == 1
        GOTO Loop1    ;Skip if Carry == 1 otherwise jump to Loop1
Loop2 RLF 0x0a, 1      ;Start Shift Left LEDs with Carry
        BTFSS 0x03, 0  ;Check if the STATUS.Carry == 1
        GOTO Loop2    ;Skip if Carry == 1 otherwise jump to Loop2
End:    GOTO Main      ;Start the program again
```

Fig- 3: Test code.

PIPELINE DESIGN

In fig.4, the pipeline datapath is demonstrated as two pipeline stages which are Fetch Stage (IF); it used to fetch the instruction, check the instruction that is related to status or switch registers, and to store the control signals. The second stage is Execution Stage (EX); it used to save the results and update the status and the destination register either WREG or RF. Also, there are two differences between the single cycle and pipeline designs which are the switch multiplexor that is placed after the STATUS multiplexor so defiantly the output after these multiplexors is a register file value and now we can execute more instructions in the switch register rather than just reading from it. Second, changing the comparing process, in the signal cycle was placed on the internal design of the ALU, but now it is placed outside the ALU so the PC will be updated much faster than it was. In the single cycle design, if the jump is placed after the comparison, the comparison will be faster than the next fetch [5].

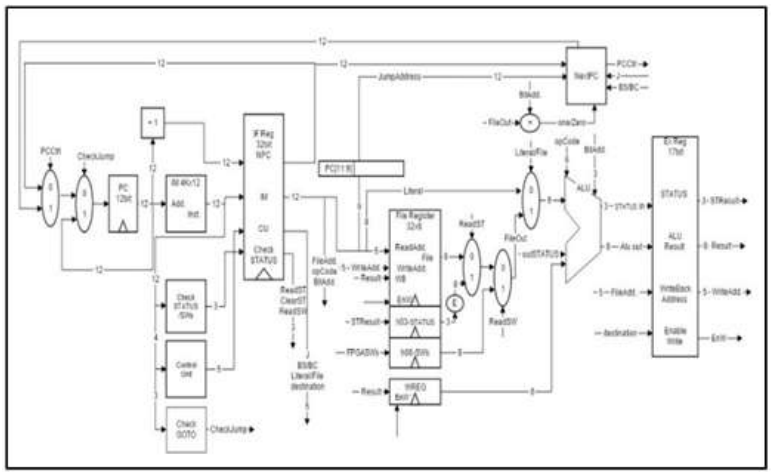

Fig-4: Pipeline design

PIPELINE REGISTER TRANSFER LEVEL

According to fig.5, the RTL design of the pipeline PIC12C5xx in Xilinx Design Suite 14.7 that have four main modules which are CLK Divider to slow down the FPGA CLK, Control Unit to control the data path design, Data path contains the architecture design of PIC12C5xx as shown in fig.5, and display7Seg to display the seven segment value in the FPGA.

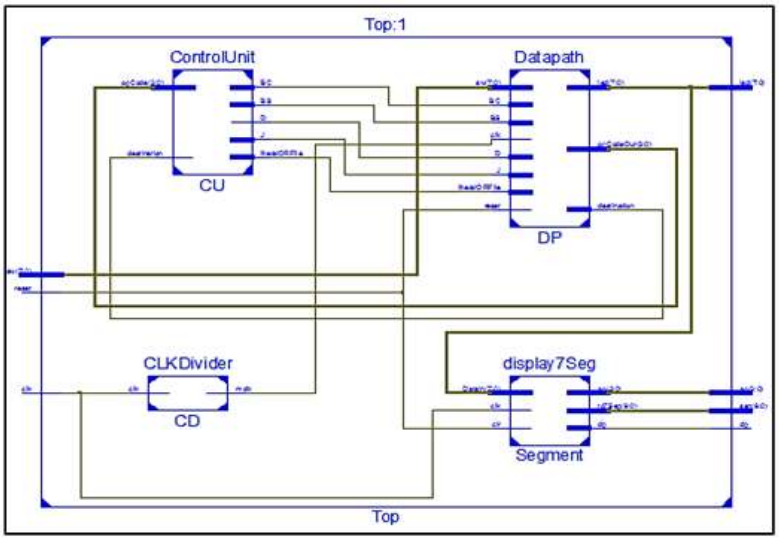

Fig-5: Pipeline register transfer level

PIPELINE HAZARD

Based on fig.4, CheckJump signal that is generated when GOTO instruction is fetched. For instant, if there is an instruction written to the next GOTO instruction, so the PC is very fast and it will fetch the next instruction. However, in this design the PC is delayed by placed a multiplexor that is controlled by CheckJump signal to slow down the PC and fetch the instruction that is placed on the jump address. So the penalty is losing one cycle to fetch the correct instruction [4].

RESOURCES REPORT

After the implantation of the pipeline PIC12C5xx, the design summery of the resources is illustrated as bellow:

- 11% of the Slice Flip Flops (CLBs) 230 out of 1920
- 22% 4 inputs LUTs 438 out of 1920
- 34% of the occupied Slices 333 out of 960
- 36% number of bounded IOBs
- 26% total number of 4 input LUTs 503 (438 as logic, and 65 as a route-thru) out of 1920

CONCLUSION

By the end of doing this project, we presented PIC12C5xx pipelined processor design and implementation on Xilinx FPAG Chip as a final project for the Computer Organization Course (EE361) at King Abdulaziz University. The main objectives of doing this project were verified and the results were satisfied. For future work, we would like to design and implement a cash memory for our processor.

REFERENCES

1. X Company; ISE Design Suite 14: Release Notes, Installation, and Licensing, 2013.
2. D Company; Digilent Basys2 Board Reference Manual, 2010.
3. M Comapany; 8-Pin, 8-Bit CMOS Microcontrollers, 2003
4. Patterson DA, Hennessy JL; Computer Organization and Design, Waltham, USA: Elsevier Inc, 2012.
5. Li Y, Chu W; Aizup- A Pipelined Processor Design and Implementation on XILINX FPGA Chip, IEEE Symposium on FPGAs for Custom Computing Machines, 1996; 98-106.