Tennessee Technological University

ECE 4120 – Fundamentals of Computer Design

Department of Electrical and Computer Engineering

Written By: Josh Adair, Cameron Dempsey, Brantlee Garland, Hunter Goodson, and Randy Wilhoite

Phase One: Instruction Fetch Unit

Table of Contents

Modified Block Diagram:

Figure 1 below shows the instruction fetch unit block diagram with bit values and any additional connections. You can see from the diagram that PC has a 32 bit input and output, the adder has a 32 bit and constant 1 input with a 32 bit output, and the 8 LSB of PC are input into the instruction memory, with a 32 bit output. The constant required changing to 1 instead of 4 due to the fact that adding four caused addresses to be skipped. Instead of the address going from 4 to 8, 8 to 12, 12 to 16, etc, we wanted it to go from 1 to 2, 2 to 3, etc. This is why we changed the constant to a 1.

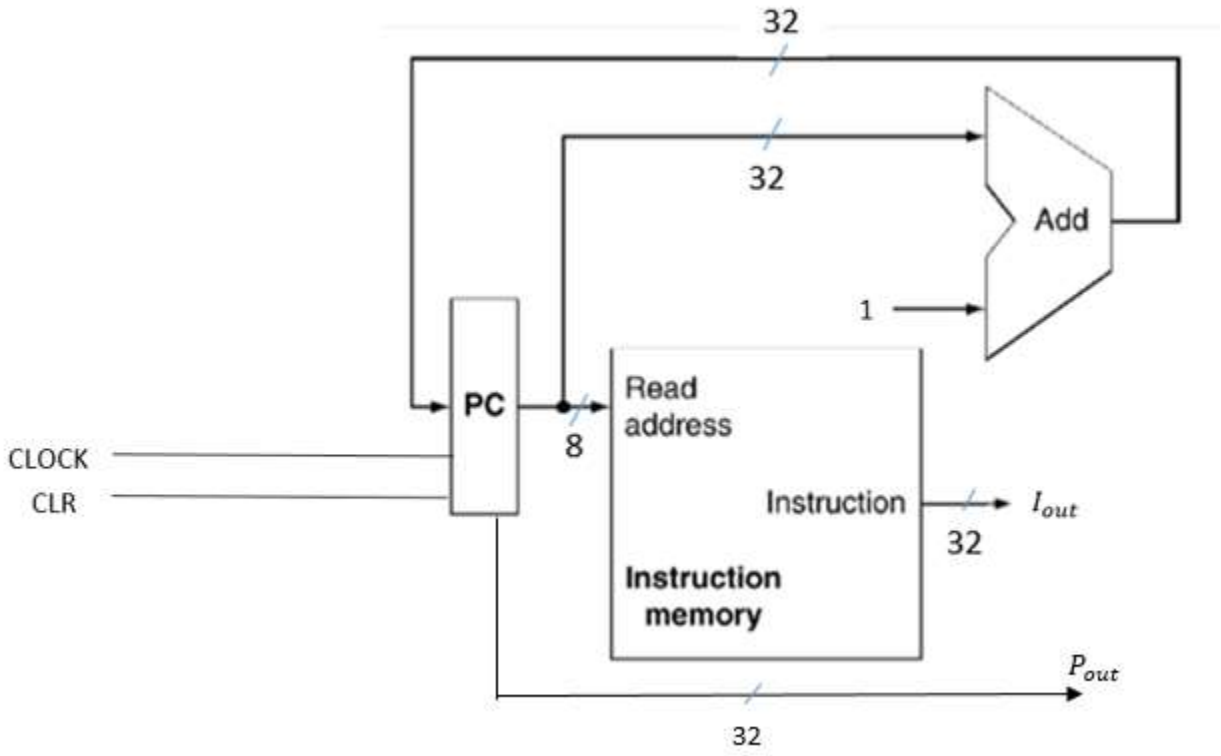

Figure 1. Modified block diagram

Phase One Objectives:

Phase one of this project had many objectives, and the big picture once all of these objectives are completed is to fully implement the instruction fetch unit of the MIPS processor. The first objective was to create a program counter (PC) in VHDL. This needed to be a synchronous block with a 32 bit input and a 32 bit output. The second objective was to create an adder in VHDL as a combinational block. The adder is 32 bits with two inputs, a constant 4 and the 32 bit output from the PC. The third objective and final block that needed to be created was the synchronous for instruction memory. The input to this block is the 8 least significant bits being output from the PC. The output should be 32 bits. The fourth objective was to ensure that the MIPS instructions were stored through the test bench in the first five

locations of instruction memory. The final objective was to simulate everything we created using Quartus and the TimeQuest timing analyzer in order to obtain relevant data, such as Fmax and timing reports.

VHDL Implementation Elaboration:

For the adder, we chose create an entity called Add4 with a 32 bit standard logic vector input A. We also have Q, a 32 bit standard logic vector output. We then created an architecture from Add4 that included a constant value of 4 and a process. Inside the process, we assign the output Q to the sum of 4 plus whatever the input was. The process and architecture are then ended.

For the PC, we chose to create an entity called Program_Counter with 3 different inputs. The first input is D, a 32 bit standard logic vector. The second and third inputs are CLR and CLK, which adjusts the clear and clock variables. Next we created an architecture of Program_Counter called pc. Inside is a process that uses both CLK and CLR. If CLR is chosen to be 1, then it is cleared and starts at the beginning. If CLR is chosen to be 0, on the rising edge of the clock, Q is set to D. The process and architecture are then ended.

For the instruction memory, we chose to use the Megafunction using the steps from the notes. After selecting IP Catalog, you then choose basic functions. Next you choose RAM: 1-PORT, VHDL, and change the bits to 32 with 256 words. Ensure the 'q' output port is selected. Next we linked the Megafunction to our .mif file by selecting the option and then specifying our filename. After choosing the correct simulation library, the final step is to ensure the VHDL component declaration file option is selected.

Synthesis Results:

Analysis & Synthesis Summary

<<Filter>>

Analysis & Synthesis Status	Successful - Sun Mar 11 21:11:47 2018
Quartus Prime Version	17.1.0 Build 590 10/25/2017 SJ Lite Edition
Revision Name	Phase_1
Top-level Entity Name	Phase_1
Family	Cyclone V
Logic utilization (in ALMs)	N/A
Total registers	6
Total pins	34
Total virtual pins	0
Total block memory bits	8,192
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

Figure 2. Synthesis results from Quartus.

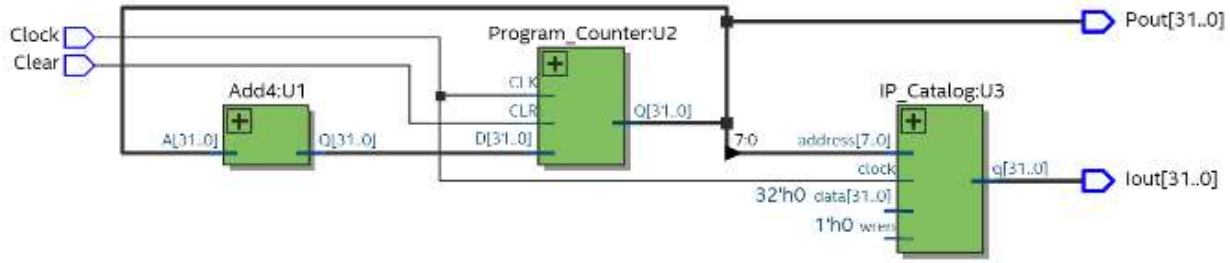

Figure 3. RTL Snapshot View

Test Bench Elaboration:

Our testbench is shown below. There are 16 instructions, but out test bench only lasts long enough to load three instructions. There is also only three instructions read as well.

```
library ieee;
use ieee.std_logic_1164.all;

entity tb_Phase_1 is
end entity tb_Phase_1;

architecture tb_Phase_1_arch of tb_Phase_1 is
        signal Clear: STD_LOGIC := '0';
        signal Clock : STD_LOGIC := '0';
        signal Iout : STD_LOGIC_VECTOR(31 DOWNTO 0);
        signal Pout : STD_LOGIC_VECTOR(31 DOWNTO 0);

component Phase_1 IS
  port (  Clear, Clock :in STD_LOGIC;
                        Iout, Pout : out STD_LOGIC_VECTOR(31 DOWNTO 0));
        END component ;

        Begin
                --Each time the clock transitions, an read occurs.
                DUT: Phase_1 port map (Clear,Clock, Iout(31 DOWNTO 0), Pout(31 DOWNTO 0));
                        Clock <= not Clock after 20ns;
                        Clear <= '1' after 100ns,
                                                '0' after 300ns;
end tb_Phase_1_arch;
```

Waveform Elaboration:

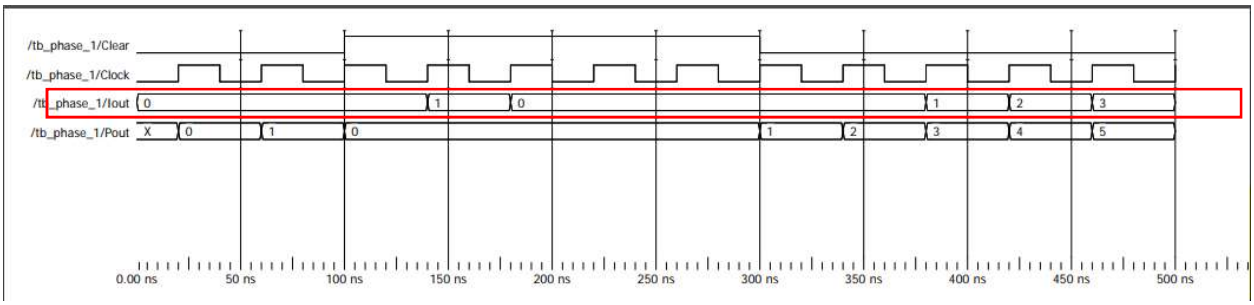

Figure 4. Waveform obtained from implementation

a) A .mif file was used, and the reading from memory is clearly shown on the waveform in red.
b) The waveform shows the successful implementation.

TimeQuest Timing Analyzation:

After completing the TimeQuest analysis, we found the following data shown below.

a) Fmax value

	Fmax	Restricted Fmax	Clock Name	Note
1	490.2 MHz	315.06 MHz	Clock	limit due to minimum period restriction (tmin)

b) Shown is the timing report.

Path Summary

	Property	Value
1	From Node	IP_Catalog:U3|altsyncram:altsyncram_component|altsyncram_7j14:auto_generated|ram_block1a0~porta_we_reg
2	To Node	IP_Catalog:U3|altsyncram:altsyncram_component|altsyncram_7j14:auto_generated|q_a[9]
3	Launch Clock	Clock
4	Latch Clock	Clock
5	Data Arrival Time	5.243
6	Data Required Time	4.203
7	Slack	-1.040 (VIOLATED)

Statistics

	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.885					
3	Data Delay	1.055					
4	Number of Logic Levels		0				
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC		3	1.929	46	0.000	1.696
2	Cell		3	2.259	54	0.273	1.376
2	Data						
1	Cell		1	1.055	100	1.055	1.055
2	uTco		1	0.000	0	0.000	0.000
2	Required Path						
1	Clock						
1	IC		3	1.687	55	0.000	1.620
2	Cell		3	1.353	45	0.252	0.610

Data Path

Data Arrival

	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	4.188	4.188					clock path
1	0.000	0.000					source latency
2	0.000	0.000			1	PIN_AA15	Clock
3	0.000	0.000	RR	IC	1	IOIBUF_X40_Y0_N18	Clock~input|i
4	0.610	0.610	RR	CELL	1	IOIBUF_X40_Y0_N18	Clock~input|o

	Total	Incr	RF	Type	Fanout	Location	Element
5	0.843	0.233	RR	IC	1	CLKCTRL_G6	Clock~inputCLKENA0|inclk
6	1.116	0.273	RR	CELL	10	CLKCTRL_G6	Clock~inputCLKENA0|outclk
7	2.812	1.696	RR	IC	34	M10K_X58_Y2_N0	U3|altsyncram_component|auto_generated|ram_block1a0|clk0
8	4.188	1.376	RR	CELL	12	M10K_X58_Y2_N0	IP_Catalog:U3|altsyncram:altsyncram_component|altsyncram_7j14:auto_generated|ram_block1a0~porta_we_reg
3	5.243	1.055					data path
1	4.188	0.000		uTco	12	M10K_X58_Y2_N0	IP_Catalog:U3|altsyncram:altsyncram_component|altsyncram_7j14:auto_generated|ram_block1a0~porta_we_reg
2	5.243	1.055	RR	CELL	1	M10K_X58_Y2_N0	IP_Catalog:U3|altsyncram:altsyncram_component|altsyncram_7j14:auto_generated|q_a[9]

Data Required

	Total	Incr	RF	Type	Fanout	Location	Element
1	1.000	1.000					latch edge time
2	4.303	3.303					clock path
1	1.000	0.000					source latency
2	1.000	0.000			1	PIN_AA15	Clock
3	1.000	0.000	RR	IC	1	IOIBUF_X40_Y0_N18	Clock~input|i
4	1.610	0.610	RR	CELL	1	IOIBUF_X40_Y0_N18	Clock~input|o
5	1.677	0.067	RR	IC	1	CLKCTRL_G6	Clock~inputCLKENA0|inclk
6	1.929	0.252	RR	CELL	10	CLKCTRL_G6	Clock~inputCLKENA0|outclk
7	3.549	1.620	RR	IC	34	M10K_X58_Y2_N0	U3|altsyncram_component|auto_generated|ram_block1a0|clk0
8	4.040	0.491	RR	CELL	1	M10K_X58_Y2_N0	IP_Catalog:U3|altsyncram:altsyncram_component|altsyncram_7j14:auto_generated|q_a[9]
9	4.303	0.263					clock pessimism removed
3	4.203	-0.100					clock uncertainty
4	4.203	0.000		uTsu	1	M10K_X58_Y2_N0	IP_Catalog:U3|altsyncram:altsyncram_component|altsyncram_7j14:auto_generated|q_a[9]

Phase One Conclusion:

We were able to complete phase one of the project successfully and met all project constraints. We created our synchronous PC block, combinational adder block, and synchronous instruction memory block. We only used the Megafunction to implement the instruction memory. For the adder, it was required to change the constant to be 1 instead of 4. This was due to receiving complications when adding 4 because the addresses were being increased too much. Changing the constant to 1 fixed our problem. Both the PC and instruction memory have the same clock signal. We were able to use the top module to port map all of the components. Our project compiled and worked successfully, providing us the required data and results via Quartus/ModelSim and TimeQuest.