CMPS301 Computer Architecture

Project Report – Phase 2

Team #12

Students Name:	Amr Magdy	4220131
	Farouq Diaa	4220133
	Mahmoud Aly	4220141
	Mohamed Maher	4220140
	Omar Mohamed	1180336

Submitted to: Dr. Mostafa Ghouneem
Eng. Noor Eldin
Eng. Mostafa ElNashar

Submitted Date: 26/12/2024

Table of Contents

List of tables

List of Figures

Changes after phase 1

General Changes:

The ISA, we flip it the Opcode was in bits 0-4 in phase 1 but now since it flipped, the Opcode is on 11-15 and now the empty bits is 0,1 bits instead of 14,15 bits was the empty bits in phase 1

Example for this

Table 1 General Changes Example between phase 1 and phase 2

Phases	ISA
Phase 1	[Opcode(1-4)][Rsc1(5-7)] [Rsc2(8-10][Rd(11-13][additional (14)][additional (15)]
Phase 2	[additional (0)][additional(1)][Rd(2-4)][Rsc2(5-7)][Rsc1(8-10)][Opcode (11-15)]

Design Changes:

Fetch Stage:

Table 2 Changes happened in Fetch stage

The change	What was the purpose	Explaining
Removing MUX between PC and IM	It was choosing between it takes the Interrupt output and the PC output to the IM	We will take the Interrupt output from the store to the IM directly so no need for MUX

Decode Stage:

Table 3 Changes happened in Decode Stage

The change	What was the purpose	Explaining
Adding out Enable for instruction OUT in Control unit	NA	Enable for Out Instruction
Adding DM_adress_Selector signal in Control unit	NA	It goes to a MUX to choose between weather it will take alu or SP it will choose SP if the signal is ‘1’
Adding DM_data_Selector signal in Control unit	NA	It goes to a MUX to choose between weather it will take PC+1 or Rscrs it will choose PC+1 if the signal is ‘1’
Adding PC_Update_Enable signal in Control	NA	Signal Enable For PC
Adding PC_Update_Selector signal in Control unit	NA	It chooses weather the PC will update from the memory or from the register it will choose Memory if the signal is ‘1’
Adding WB_Scrs_Sell signal in Control unit	NA	Select the source of the Write back weather form Memory or ALU it will select ALU if the signal is ‘1’
Adding Interrupt_index signal in control unit	NA	Aither it will be 1 or 0 according to the instruction
Adding Store signal in Control unit	NA	to pass the Rscr2 in STD instruction

Execute Stage:

Table 4 Changes happened in Execute stage

The change	What was the purpose	Explaining
Removing Flag register	It was restoring the flags from the ALU	Push the flags is the stack Pointer with PC

Memory stage:

Table 5 Changes happened in Memory stages

The change	What was the purpose	Explaining
Adding MUX in memory stage	NA	Select between the SP output or the ALU output which one will go to the memory the selector is DM_Adress_Selector
Removing push, pop, Call, and Return signal from the Memory Input Signal	NA	We think we are not needed it
Adding SP update unit to the Memory Stage	NA	It gives the output of the SP to the mux that we added and it gives the updated SP back to the SP (SP+1orSP-1)

Write Back Stage:

There are no changes happened in the design of the Write back stage.

Pipeline Hazards and solutions

Design after edit:

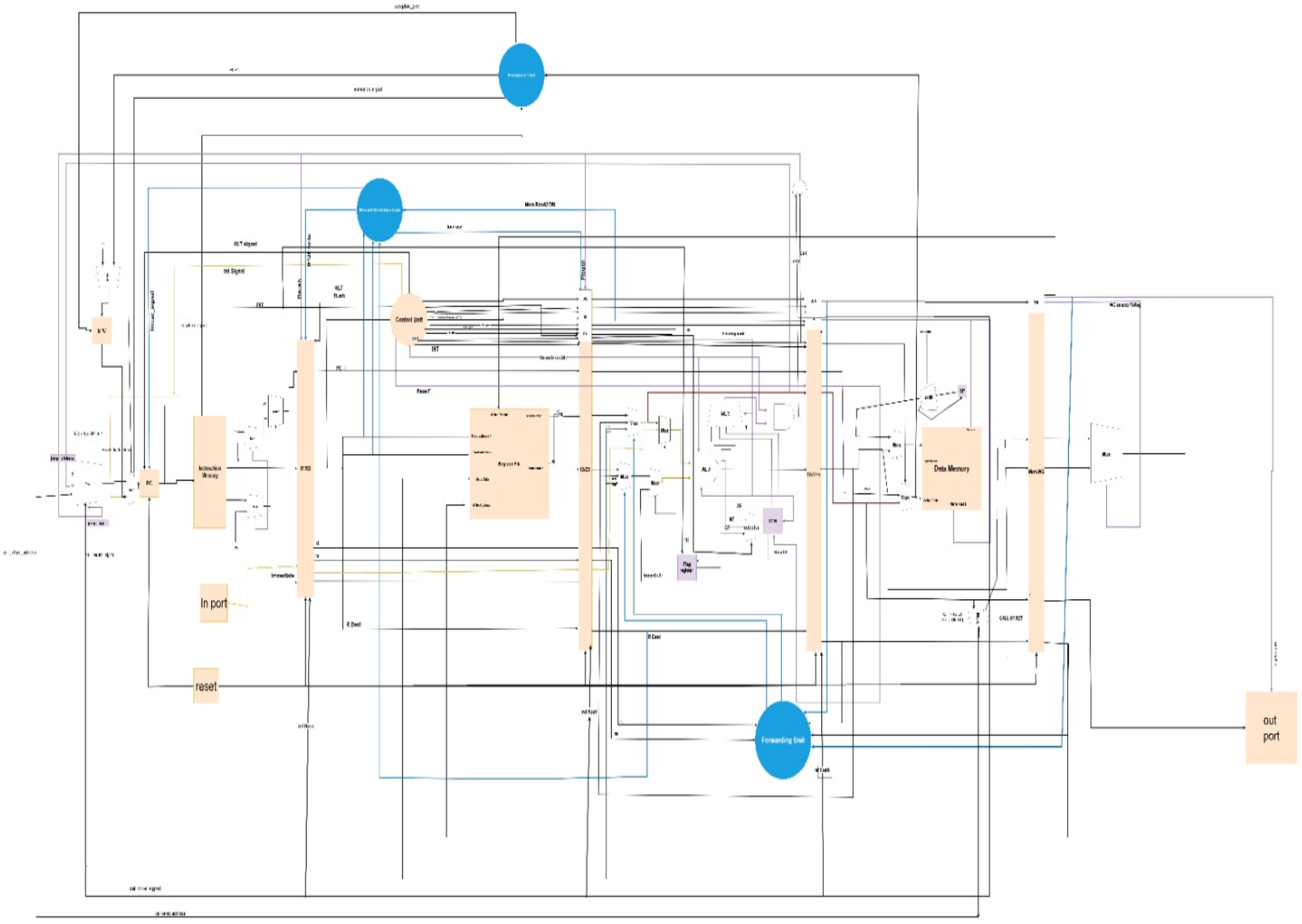


Figure 1 Design After Edit

Hazards and Solutions

Table 6 Hazards and solutions

Type of Hazard	Source of Hazard	Handling	
		Detection	Solution
Data Hazard	**Load-use**	**Hazard Detection Unit:** Checks if 1. emRead = 1 in ID/EX 2. dst in ID/EX = Rsrc1 in IF/ID 3. dst in ID/EX = Rsrc2 in IF/ID (1&&2) or (1&&3) -> Load-Use Hazard detected.	**Forwarding Unit:** Stall once (NOP) until data is read from memory. Forward ReadData from MEM/WB to EXECUTE stage. **LDD & POP instructions.**
	Read after write	**Hazard Detection Unit:** Checks if 1. egWrite_1 = 1 & MemRead = 0 in EX/MEM 2. dst in EX/MEM = Rsrc1 in ID/EX 3. dst in EX/MEM = Rsrc2 in ID/EX (1&&2) or (1&&3) -> Read after write Hazard detected.	**Forwarding Unit:** If (IN || LDM), Forward data accordingly: • N -> IN-Port value • DM -> Immediate value in ID/EX Else, forward WriteData from ALU result to EXECUTE stage.

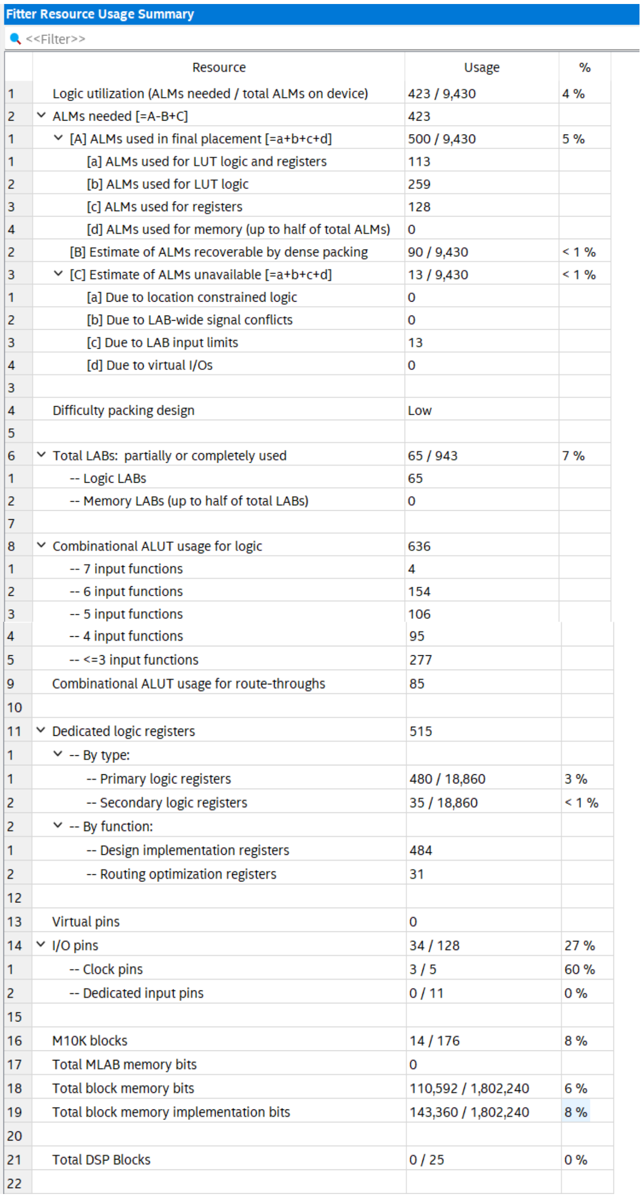
Structural Hazard	**Return Before Jump**	Checks if instruction in Execute stage is jump and instruction in memory stage is Return and update PC based on Return not jump
Control Hazard	**Branching**	**Static branch prediction:** Assume not taken 1. If prediction was correct, continue working normally. 2. If prediction was wrong, flush & change PC accordingly: Flush IF/ID & ID/EX registers (2 stall) Flush IF/ID & ID/EX & EX/MEM if RET (3 stalls)

Comparison between the processor and the pipeline processor:

Table 7 Comparison between single cycle processor and pipeline processor

Feature	Single cycle Processor	**5-Stage Pipelined Processor**
Clock Speed	115.998ns = 8.696MHz	11.246ns = 88.921MHZ

FPGA recourses Usage for Pipeline

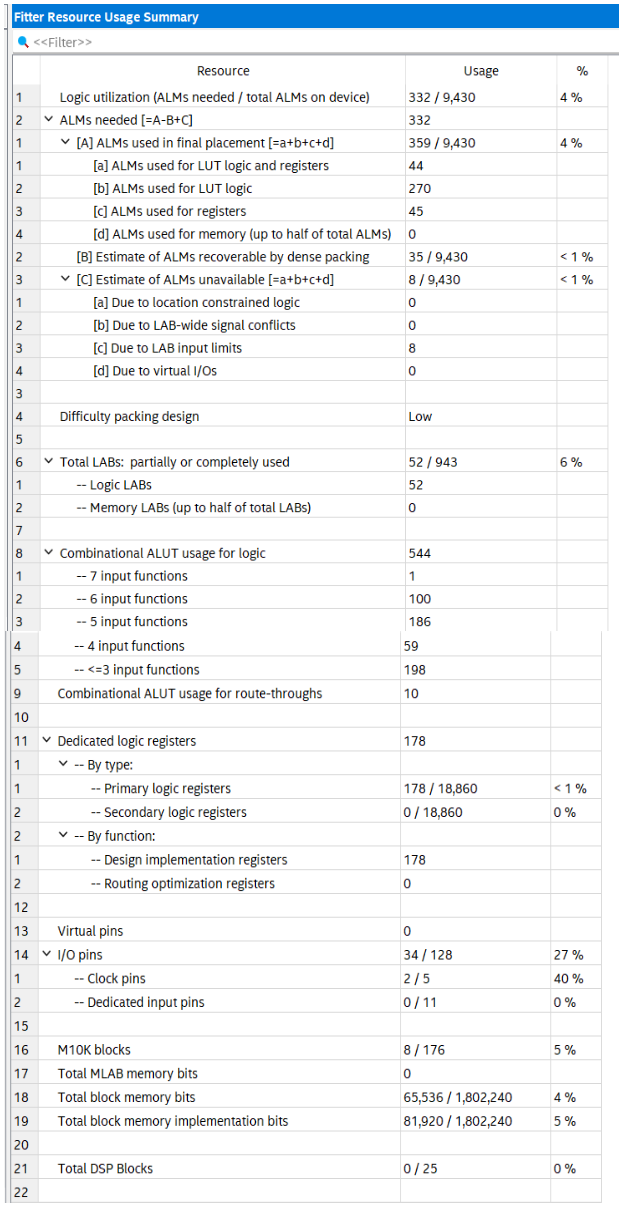


Fitter Resource Usage Summary

<<Filter>>

	Resource	Usage	%
1	Logic utilization (ALMs needed / total ALMs on device)	423 / 9,430	4 %
2	ALMs needed [=A-B+C]	423	
1	[A] ALMs used in final placement [=a+b+c+d]	500 / 9,430	5 %
1	[a] ALMs used for LUT logic and registers	113	
2	[b] ALMs used for LUT logic	259	
3	[c] ALMs used for registers	128	
4	[d] ALMs used for memory (up to half of total ALMs)	0	
2	[B] Estimate of ALMs recoverable by dense packing	90 / 9,430	< 1 %
3	[C] Estimate of ALMs unavailable [=a+b+c+d]	13 / 9,430	< 1 %
1	[a] Due to location constrained logic	0	
2	[b] Due to LAB-wide signal conflicts	0	
3	[c] Due to LAB input limits	13	
4	[d] Due to virtual I/Os	0	
3			
4	Difficulty packing design	Low	
5			
6	Total LABs: partially or completely used	65 / 943	7 %
1	-- Logic LABs	65	
2	-- Memory LABs (up to half of total LABs)	0	
7			
8	Combinational ALUT usage for logic	636	
1	-- 7 input functions	4	
2	-- 6 input functions	154	
3	-- 5 input functions	106	
4	-- 4 input functions	95	
5	-- <=3 input functions	277	
9	Combinational ALUT usage for route-throughs	85	
10			
11	Dedicated logic registers	515	
1	-- By type:		
1	-- Primary logic registers	480 / 18,860	3 %
2	-- Secondary logic registers	35 / 18,860	< 1 %
2	-- By function:		
1	-- Design implementation registers	484	
2	-- Routing optimization registers	31	
12			
13	Virtual pins	0	
14	I/O pins	34 / 128	27 %
1	-- Clock pins	3 / 5	60 %
2	-- Dedicated input pins	0 / 11	0 %
15			
16	M10K blocks	14 / 176	8 %
17	Total MLAB memory bits	0	
18	Total block memory bits	110,592 / 1,802,240	6 %
19	Total block memory implementation bits	143,360 / 1,802,240	8 %
20			
21	Total DSP Blocks	0 / 25	0 %
22			

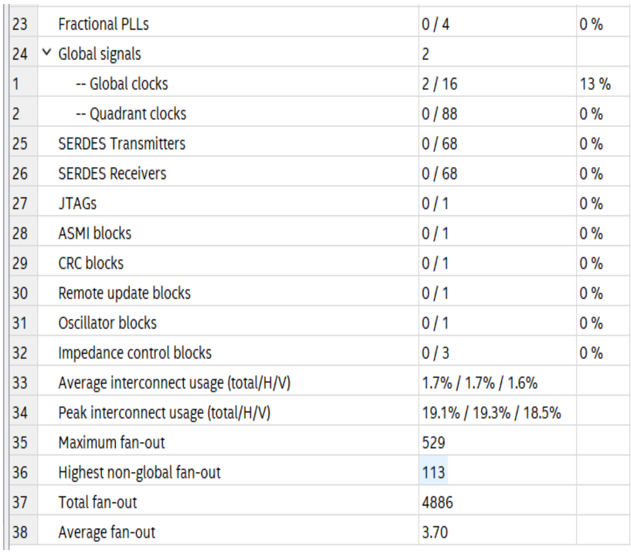
FPGA recourses Usage for Single cycle



Fitter Resource Usage Summary

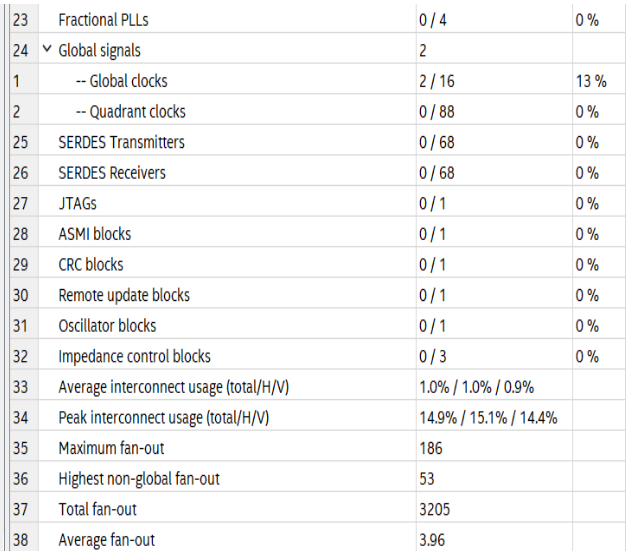
<<Filter>>

	Resource	Usage	%
1	Logic utilization (ALMs needed / total ALMs on device)	332 / 9,430	4 %
2	ALMs needed [=A-B+C]	332	
1	[A] ALMs used in final placement [=a+b+c+d]	359 / 9,430	4 %
1	[a] ALMs used for LUT logic and registers	44	
2	[b] ALMs used for LUT logic	270	
3	[c] ALMs used for registers	45	
4	[d] ALMs used for memory (up to half of total ALMs)	0	
2	[B] Estimate of ALMs recoverable by dense packing	35 / 9,430	< 1 %
3	[C] Estimate of ALMs unavailable [=a+b+c+d]	8 / 9,430	< 1 %
1	[a] Due to location constrained logic	0	
2	[b] Due to LAB-wide signal conflicts	0	
3	[c] Due to LAB input limits	8	
4	[d] Due to virtual I/Os	0	
3			
4	Difficulty packing design	Low	
5			
6	Total LABs: partially or completely used	52 / 943	6 %
1	-- Logic LABs	52	
2	-- Memory LABs (up to half of total LABs)	0	
7			
8	Combinational ALUT usage for logic	544	
1	-- 7 input functions	1	
2	-- 6 input functions	100	
3	-- 5 input functions	186	
4	-- 4 input functions	59	
5	-- <=3 input functions	198	
9	Combinational ALUT usage for route-throughs	10	
10			
11	Dedicated logic registers	178	
1	-- By type:		
1	-- Primary logic registers	178 / 18,860	< 1 %
2	-- Secondary logic registers	0 / 18,860	0 %
2	-- By function:		
1	-- Design implementation registers	178	
2	-- Routing optimization registers	0	
12			
13	Virtual pins	0	
14	I/O pins	34 / 128	27 %
1	-- Clock pins	2 / 5	40 %
2	-- Dedicated input pins	0 / 11	0 %
15			
16	M10K blocks	8 / 176	5 %
17	Total MLAB memory bits	0	
18	Total block memory bits	65,536 / 1,802,240	4 %
19	Total block memory implementation bits	81,920 / 1,802,240	5 %
20			
21	Total DSP Blocks	0 / 25	0 %
22			



23	Fractional PLLs	0 / 4	0 %
24	Global signals	2	
1	-- Global clocks	2 / 16	13 %
2	-- Quadrant clocks	0 / 88	0 %
25	SERDES Transmitters	0 / 68	0 %
26	SERDES Receivers	0 / 68	0 %
27	JTAGs	0 / 1	0 %
28	ASMI blocks	0 / 1	0 %
29	CRC blocks	0 / 1	0 %
30	Remote update blocks	0 / 1	0 %
31	Oscillator blocks	0 / 1	0 %
32	Impedance control blocks	0 / 3	0 %
33	Average interconnect usage (total/H/V)	1.7% / 1.7% / 1.6%	
34	Peak interconnect usage (total/H/V)	19.1% / 19.3% / 18.5%	
35	Maximum fan-out	529	
36	Highest non-global fan-out	113	
37	Total fan-out	4886	
38	Average fan-out	3.70	

Figure 3 Resources usage in Pipeline processor



23	Fractional PLLs	0 / 4	0 %
24	Global signals	2	
1	-- Global clocks	2 / 16	13 %
2	-- Quadrant clocks	0 / 88	0 %
25	SERDES Transmitters	0 / 68	0 %
26	SERDES Receivers	0 / 68	0 %
27	JTAGs	0 / 1	0 %
28	ASMI blocks	0 / 1	0 %
29	CRC blocks	0 / 1	0 %
30	Remote update blocks	0 / 1	0 %
31	Oscillator blocks	0 / 1	0 %
32	Impedance control blocks	0 / 3	0 %
33	Average interconnect usage (total/H/V)	1.0% / 1.0% / 0.9%	
34	Peak interconnect usage (total/H/V)	14.9% / 15.1% / 14.4%	
35	Maximum fan-out	186	
36	Highest non-global fan-out	53	
37	Total fan-out	3205	
38	Average fan-out	3.96	

Figure 2 Resources usage in Single cycle processor

In conclusion we can observe that FPGA resources used in Single cycle is Lower than in pipeline and this was expected, and Clock speed in pipeline is faster than Single cycle and both observation is Expected.