CS211: LAB Report

Hardware Design Related Labs

By Rupesh Bhusare, 2203106

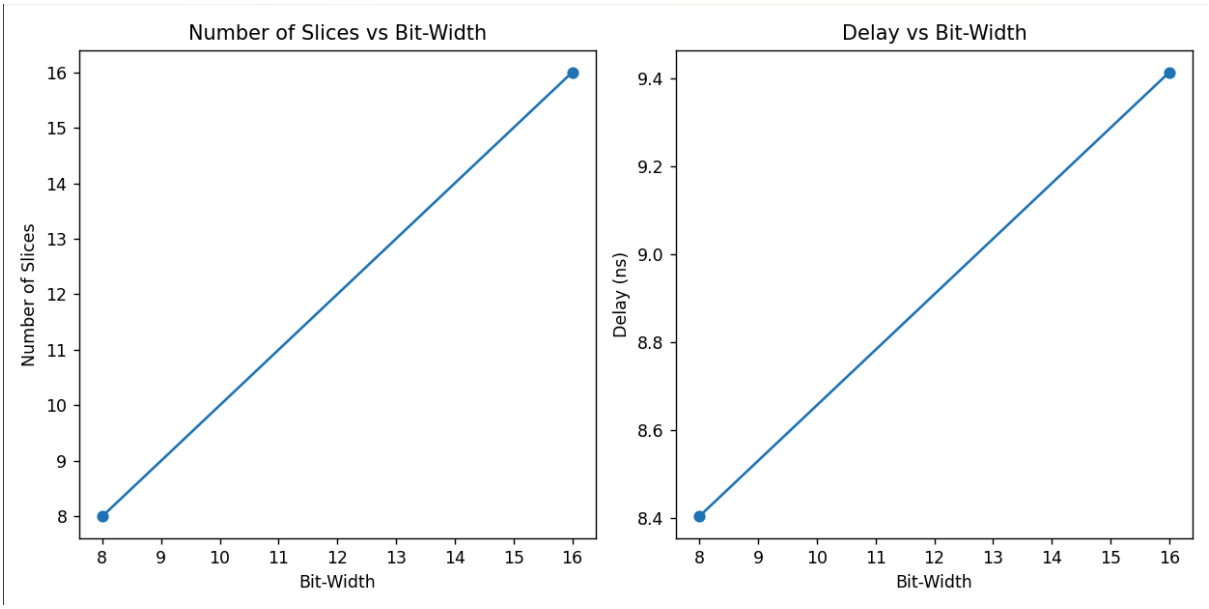
LAB 1:

Adder:

1) Set the parameter DSIZE as 8 and 16 one by one and note the number of slices used and delay (in ns) in different cases. You need to implement each time after you change the parameter value to find out the number of slices and delay in ns. Plot the graph, area (vs) bit-width as well as delay (vs) bit-width for the adder module.

Table 1: Slices and delay for adder

Parameter: DSIZE	BIT-WIDTH	No of LUT slices	Delay in ns
8	8	8	8.405
16	16	16	9.415

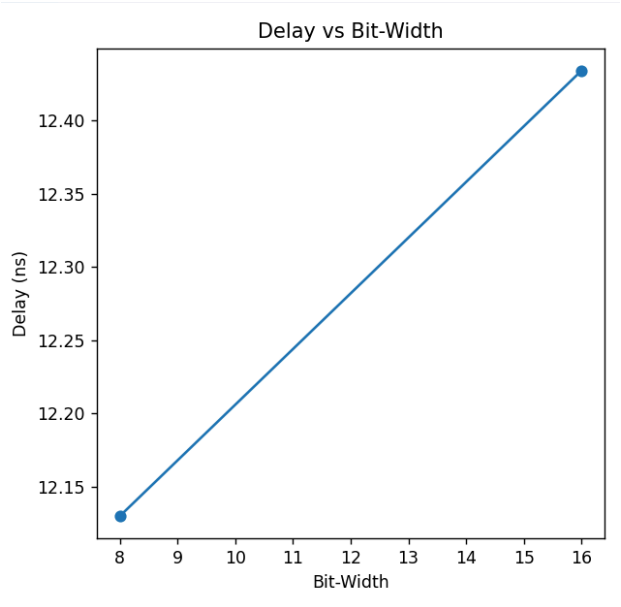


Multiplier:

1) Set the parameter DSIZE 8 and 16 one by one and note the number of slices occupied and delay in terms of (ns) for all cases. You need to synthesise each time (steps 6-13) after you change the parameter value to find out the number of slices and delay in ns. Plot the graph, area (vs) bit-width as well as delay (vs) bit-width for the multiplier module.

Table 2: Slices and delay for Multiplier

Parameter: DSIZE	BIT-WIDTH	No of LUT slices	Delay in ns
8	8	34	12.130
16	16	1 DSP	12.434

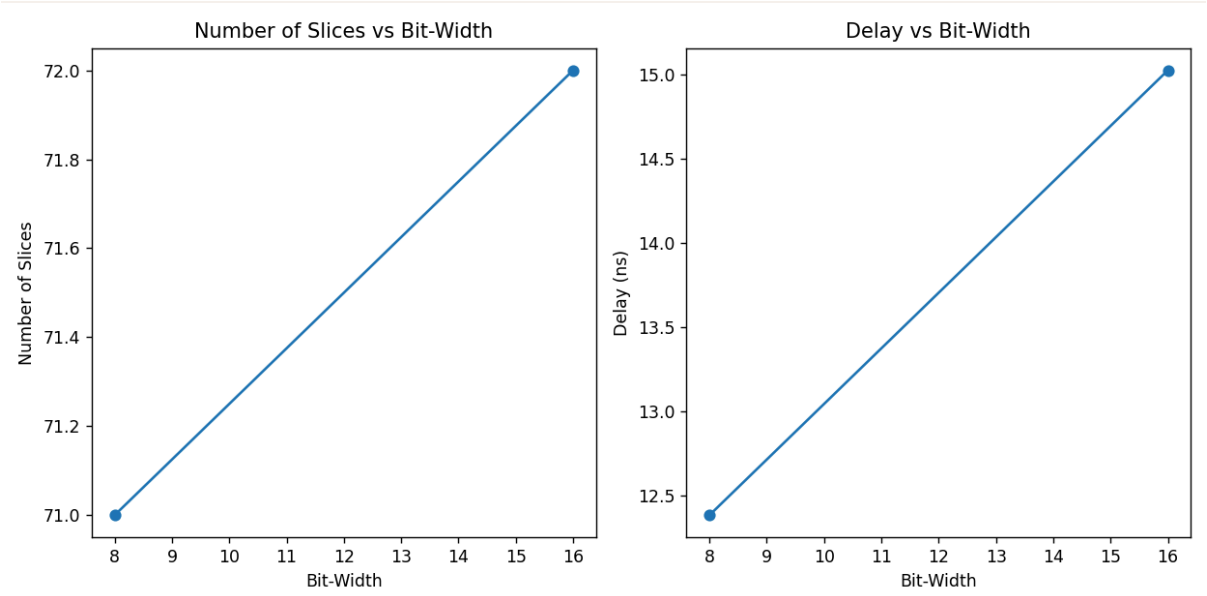


ALU module:

B. You need to set the input bit-width to 8 and 16 one by one and find out the number of slices used and the maximum combinational path delay in each case. You need to plot area (vs) bit-width as well as delay (vs) bit-width for the ALU module. In FPGA you cannot find the area directly so instead of the area you can take the number of slices, which would be considered proportional to the area.

Table 5: Slices and delay for ALU module

Parameter: DSIZE	BIT-WIDTH	No of slices	Delay in ns
8	8	71	12.387
16	16	72	15.024

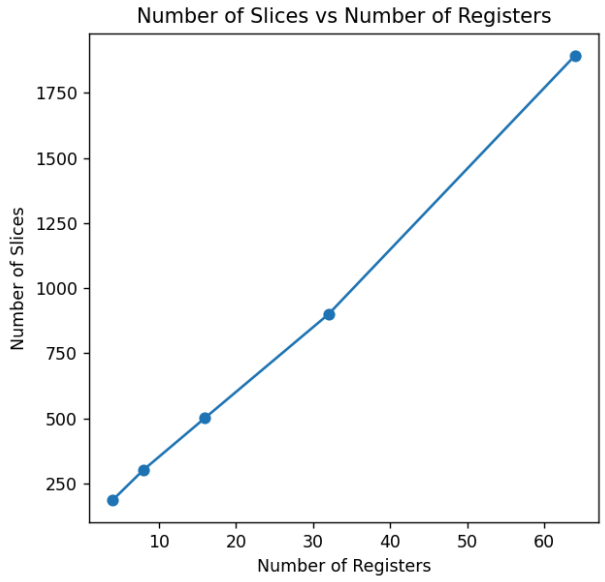


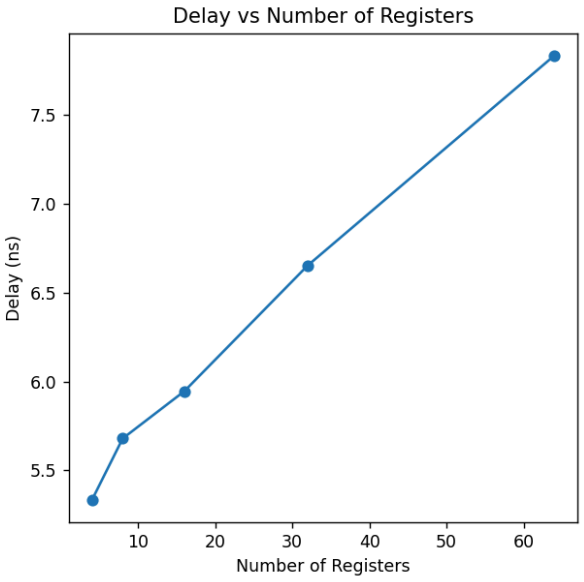
LAB 2:

Evaluation 1

1) Plot the graph, area in LUT slices (vs) No. of registers (NREG) as well as delay (vs) No. of registers (NREG) for the register file module for NREG =4, 8, 16, 32, 64. Set DSIZE (bit-width of each register) =32.

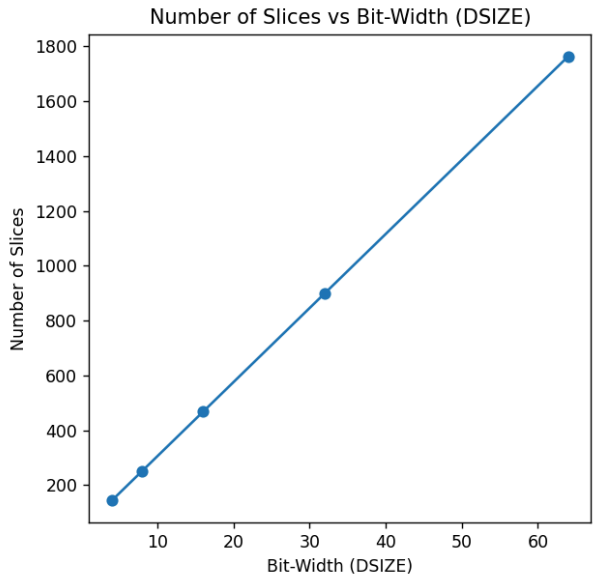
No. of registers(NREG)	Bit-width of the register(Dsize)	No of register slices used	No of LUT slices used	Minimum clock Period in ns
4	32	128	186	5.333
8	32	256	302	5.680
16	32	512	502	5.946
32	32	1024	900	6.653
64	32	2048	1894	7.834

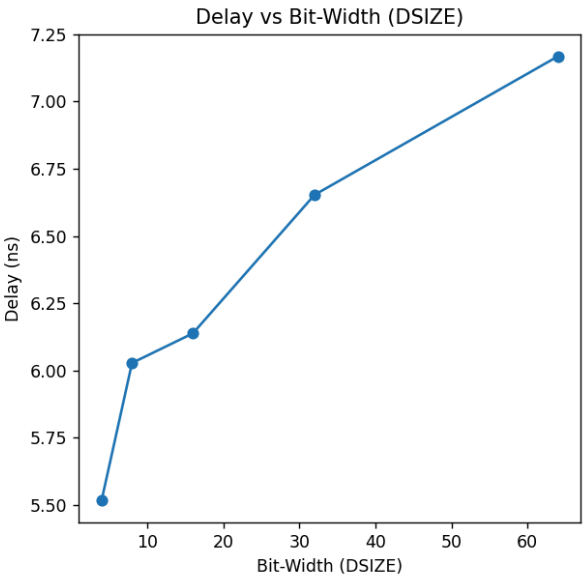




2) Plot the graph, area in LUT slices (vs) bit-width of register (DSIZE) as well as delay (vs) bit-width of register (DSIZE) for the register file module for DSIZE =4, 8, 16, 32, 64. Set NREG (number of registers) =32.

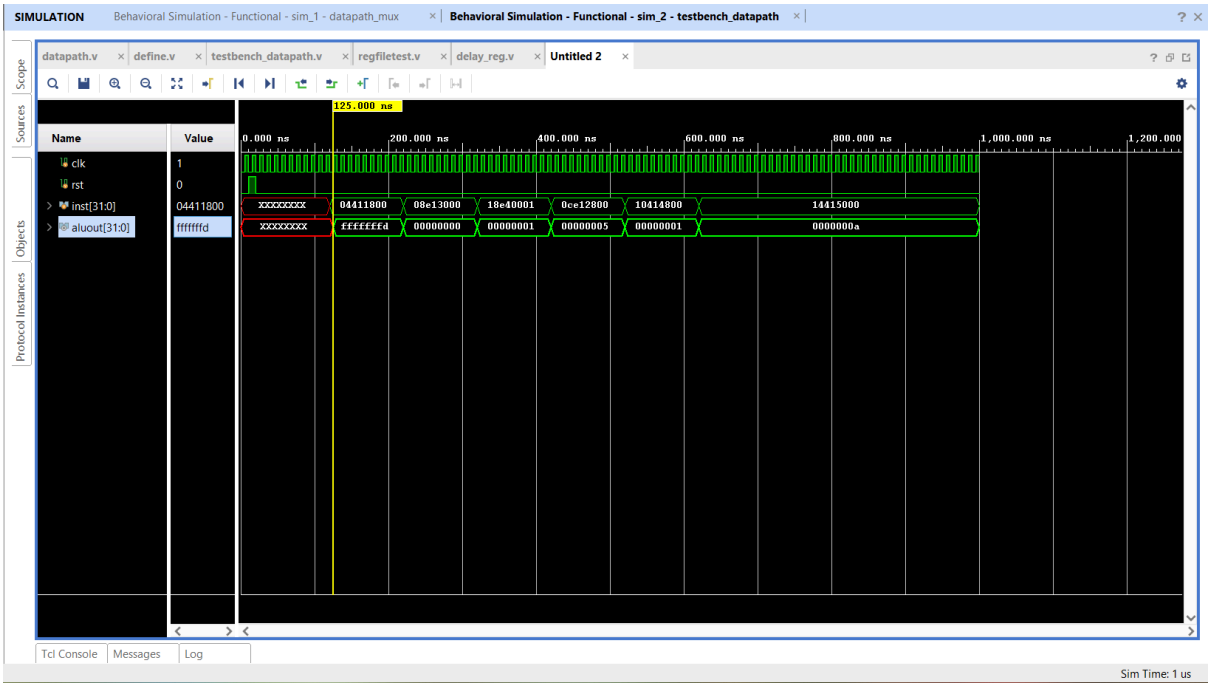
Bit-width of the register (DSIZE)	No. of registers (NREG)	No of register slices used	No of LUT slices used	minimum clock period in ns
4	32	128	144	5.517
8	32	256	252	6.028
16	32	512	468	6.138
32	32	1024	900	6.653
64	32	2048	1764	7.168





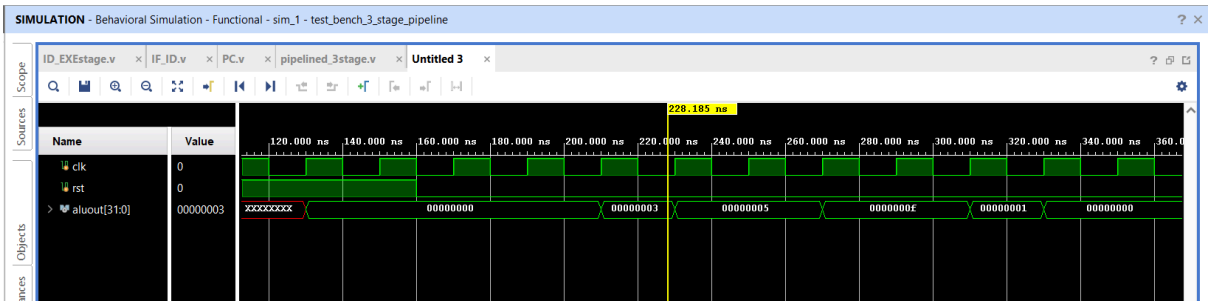
Evaluation 2

1. Test the 32 bit datapath implementation by generating a test bench. Test bench can be generated in the 'simulation mode' by right clicking the top module and adding 'New source' to be 'Verilog test fixture'. Name the Verilog test bench and choose the corresponding top module to generate the test bench. Once the test bench is generated, you need to check the bit-width of the inputs and outputs, add the clock and the test inputs as shown below.

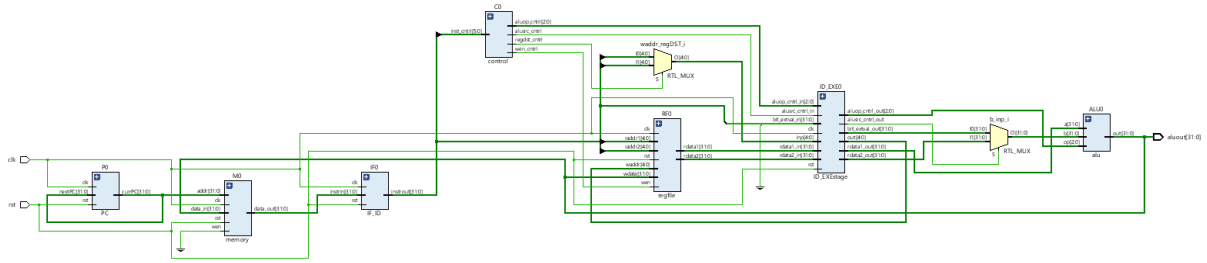


LAB 3:

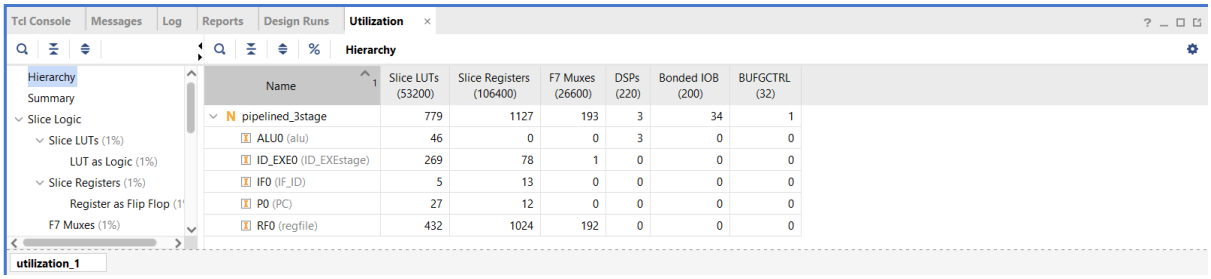
This is the simulation i got for **3 stage pipeline code**



And this is the **schematic diagram:**

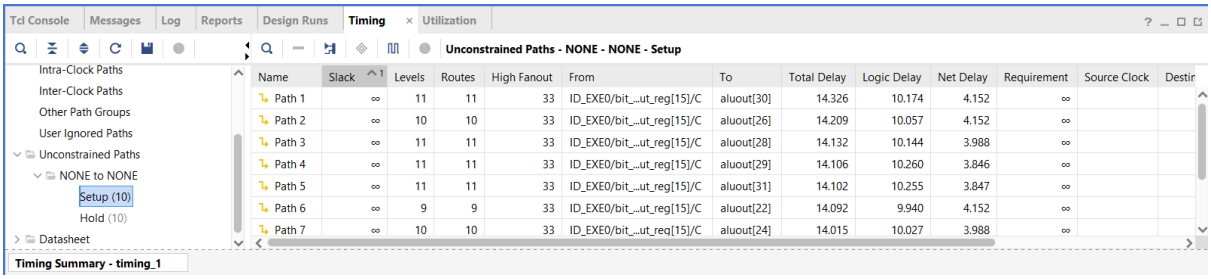


Report Utilization:



Name	Slice LUTs (53200)	Slice Registers (106400)	F7 Muxes (26600)	DSPs (220)	Bonded IOB (200)	BUFGCTRL (32)
pipelined_3stage	779	1127	193	3	34	1
ALU0 (alu)	46	0	0	3	0	0
ID_EXE0 (ID_EXEstage)	269	78	1	0	0	0
IF0 (IF_ID)	5	13	0	0	0	0
P0 (PC)	27	12	0	0	0	0
RF0 (regfile)	432	1024	192	0	0	0

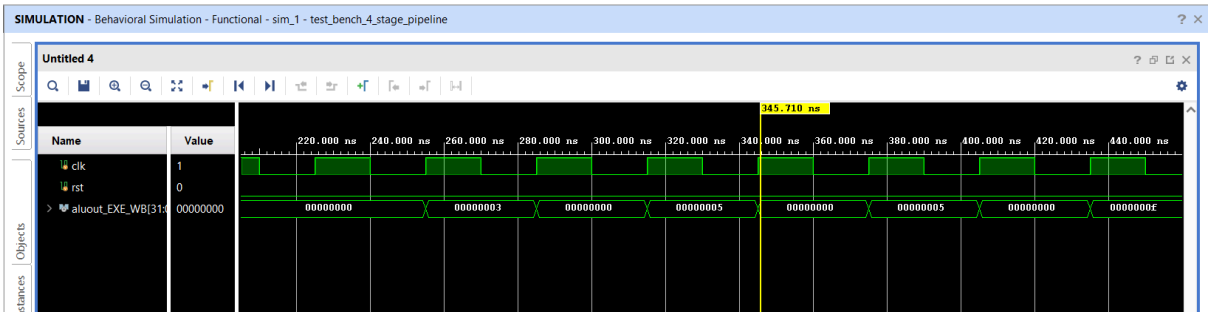
Time summary:



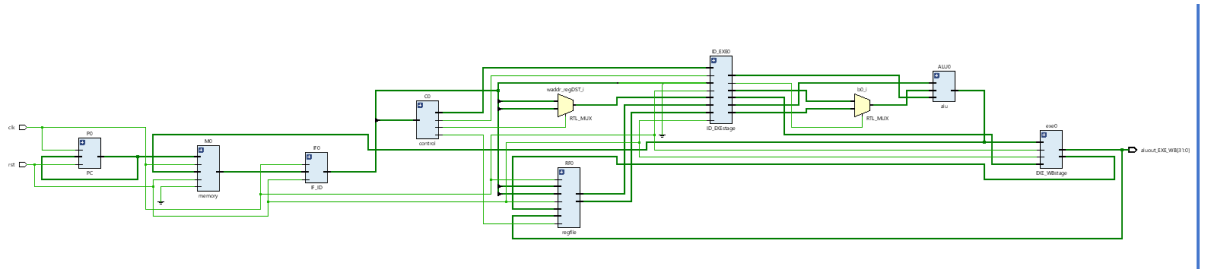
Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destin
Path 1	∞	11	11	33	ID_EXE0/bit_...ut_reg[15]/C	aluout[30]	14.326	10.174	4.152	∞		
Path 2	∞	10	10	33	ID_EXE0/bit_...ut_reg[15]/C	aluout[26]	14.209	10.057	4.152	∞		
Path 3	∞	11	11	33	ID_EXE0/bit_...ut_reg[15]/C	aluout[28]	14.132	10.144	3.988	∞		
Path 4	∞	11	11	33	ID_EXE0/bit_...ut_reg[15]/C	aluout[29]	14.106	10.260	3.846	∞		
Path 5	∞	11	11	33	ID_EXE0/bit_...ut_reg[15]/C	aluout[31]	14.102	10.255	3.847	∞		
Path 6	∞	9	9	33	ID_EXE0/bit_...ut_reg[15]/C	aluout[22]	14.092	9.940	4.152	∞		
Path 7	∞	10	10	33	ID_EXE0/bit_...ut_reg[15]/C	aluout[24]	14.015	10.027	3.988	∞		

LAB 4:

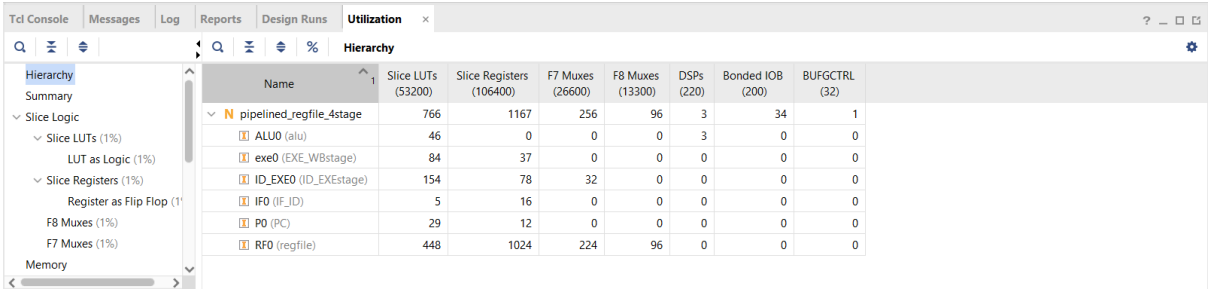
This is the simulation i got for **4 stage pipeline code:**



And this is the **schematic diagram:**

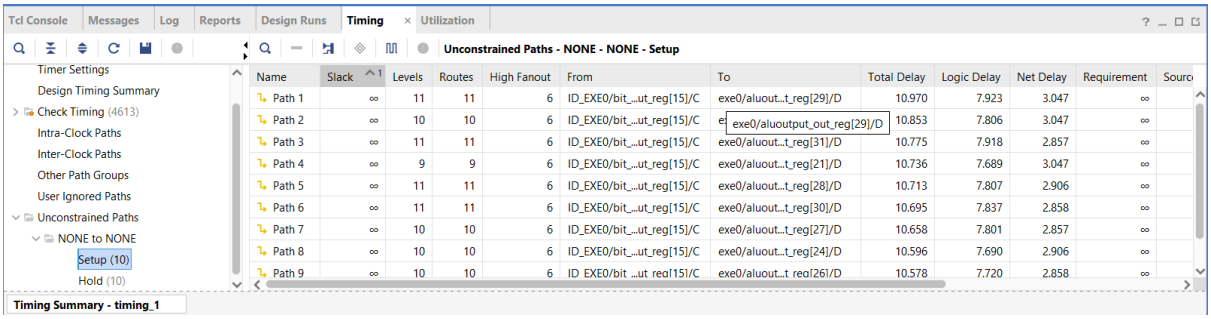


Report Utilization:



Name	Slice LUTs (53200)	Slice Registers (106400)	F7 Muxes (26600)	F8 Muxes (13300)	DSPs (220)	Bonded IOB (200)	BUFGCTRL (32)
pipelined_regfile_4stage	766	1167	256	96	3	34	1
ALU0 (alu)	46	0	0	0	3	0	0
exe0 (EXE_WBstage)	84	37	0	0	0	0	0
ID_EXE0 (ID_EXEstage)	154	78	32	0	0	0	0
IF0 (IF_ID)	5	16	0	0	0	0	0
P0 (PC)	29	12	0	0	0	0	0
RF0 (regfile)	448	1024	224	96	0	0	0

Time summary:



Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source
Path 1	∞	11	11	6	ID_EXE0/bit_...ut_reg[15]/C	exe0/aluout...t_reg[29]/D	10.970	7.923	3.047	∞	
Path 2	∞	10	10	6	ID_EXE0/bit_...ut_reg[15]/C	exe0/aluoutput_out_reg[29]/D	10.853	7.806	3.047	∞	
Path 3	∞	11	11	6	ID_EXE0/bit_...ut_reg[15]/C	exe0/aluout...t_reg[31]/D	10.775	7.918	2.857	∞	
Path 4	∞	9	9	6	ID_EXE0/bit_...ut_reg[15]/C	exe0/aluout...t_reg[21]/D	10.736	7.689	3.047	∞	
Path 5	∞	11	11	6	ID_EXE0/bit_...ut_reg[15]/C	exe0/aluout...t_reg[28]/D	10.713	7.807	2.906	∞	
Path 6	∞	11	11	6	ID_EXE0/bit_...ut_reg[15]/C	exe0/aluout...t_reg[30]/D	10.695	7.837	2.858	∞	
Path 7	∞	10	10	6	ID_EXE0/bit_...ut_reg[15]/C	exe0/aluout...t_reg[27]/D	10.658	7.801	2.857	∞	
Path 8	∞	10	10	6	ID_EXE0/bit_...ut_reg[15]/C	exe0/aluout...t_reg[24]/D	10.596	7.690	2.906	∞	
Path 9	∞	10	10	6	ID_EXE0/bit_...ut_reg[15]/C	exe0/aluout...t_reg[26]/D	10.578	7.720	2.858	∞	

A 3-stage pipelined processor consumes more Slice Look-Up Tables (LUTs) compared to a 4-stage pipelined processor. Additionally, the maximum total delay is higher in a 3-stage pipelined processor than in a 4-stage pipelined processor.