Indian Institute of Technology, Goa

iitgoa.ac.in

Implementation of Three and Four Stage Pipelining MIPS

-by Abhinav Kumar Ray

Bachelor of
technology in
Computer Science

Indian Institute of Technology, Goa
Near ITI College, Farmagudi, Ponda
Goa – 403401

Project started on 9th November 2019.

Abstract

Vivado (**I**ntegrated **S**ynthesis **E**nvironment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. Here, we try to perform three and four stage MIPS pipelined data path and analyse it accordingly.

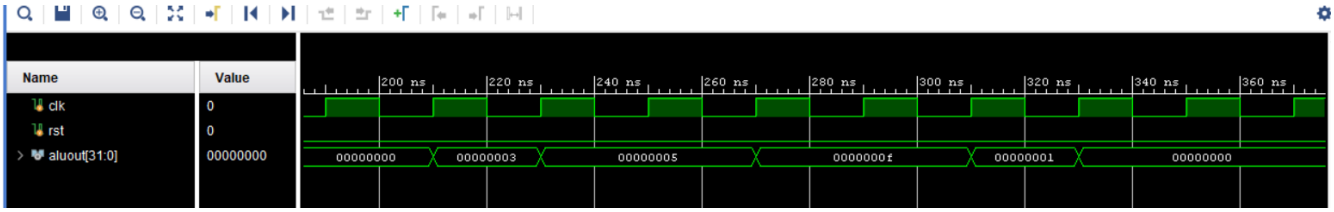
Part A

The steps to implement the three stage pipeline are mentioned below.

- Increment the program counter in PC.v file
- IF_ID.v file implements the IF/ID stage. It has one register which takes instruction as input and stores.
- Update ID_EXEstage file. It has 6 output registers
- Instantiate every module in pipelined_3stage.v file and make appropriate connections.
- Hard code the memory.v file.
- Create a testbench.v file for simulation of our design.

Assumption: max_dsp is set to 0.

Simulating in Vivaldo, we get the following timing diagram.(Note: See the output sequence)



The following report utilization was obtained after synthesis of the design.

Hierarchy

Name	Slice LUTs (48000)	Slice Registers (96000)	F7 Muxes (32000)	F8 Muxes (16000)	Bonded IOB (338)	BUFGCTRL (32)
pipelined_3stage	1398	1128	193	32	34	1

- Area is measured as quantity directly proportional to number of Slice LUTs. Here there are 1398 slices.

From the Report Timing Summary after synthesis, we conclude following: -

- Max Setup Total Delay – 17.896 ns
- Min Setup Total Delay – 17.068 ns (Assumption- 10th path was considered to report minimum time)

After Implementing the design, we get the following utilization and timing: -

Hierarchy

Name	Slice LUTs (48000)	Slice Registers (96000)	F7 Muxes (32000)	F8 Muxes (16000)	Slice (16000)	LUT as Logic (48000)	LUT Flip Flop Pairs (48000)	Bonded IOB (338)	BUFGCTRL (32)
pipelined_3stage	1359	1128	193	32	704	1359	79	34	1

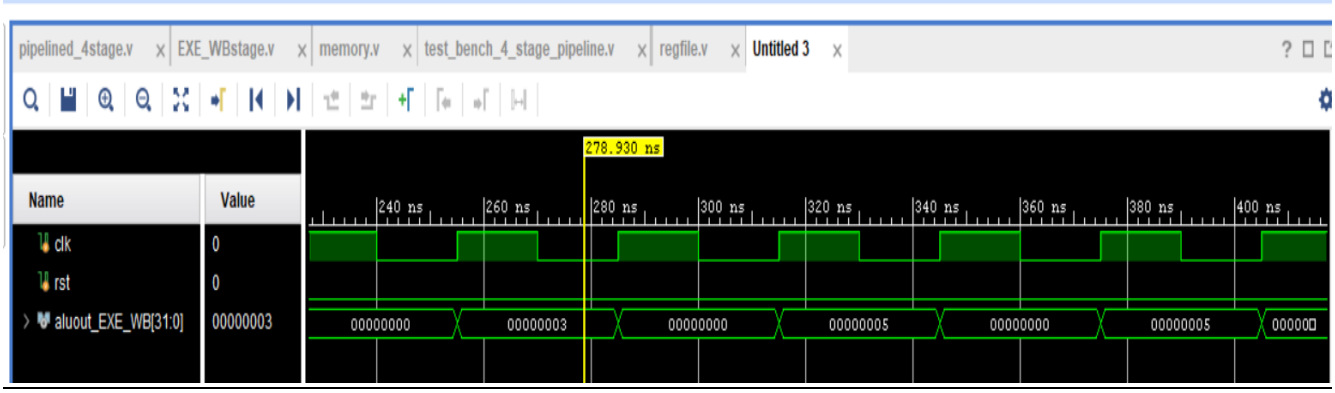
- Max Setup total delay – 28.025 ns

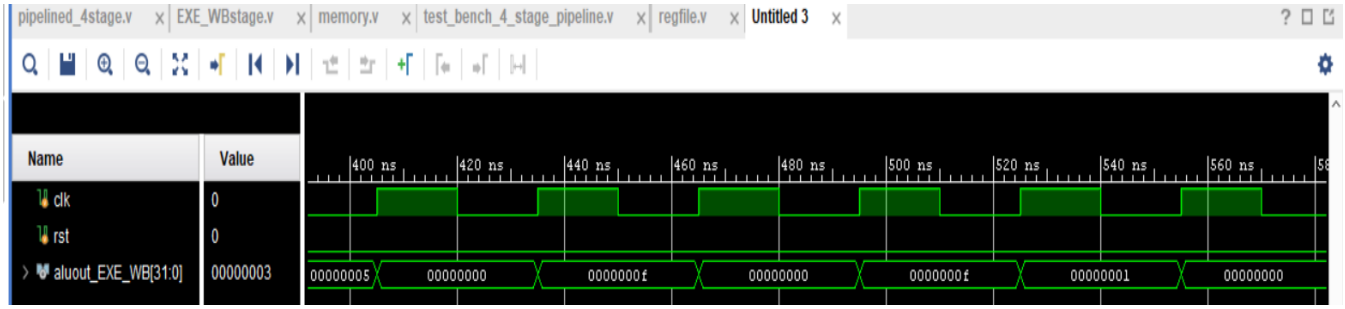
Part B

The steps to implement the four-stage pipeline are mentioned below.

- Increment the program counter in PC.v file
- IF_ID.v file implements the IF/ID stage. It has one register which takes instruction as input and stores.
- Update ID_EXEstage file. It has 6 output registers
- Update the EXE_WBstage.v file. It has2 output registers.
- Instantiate every module in pipelined_4stage.v file and make appropriate connections.
- Hard code the memory.v file. Also add NOPs wherever there is dependency between consecutive instructions.
- Create a testbench.v file for simulation of our design.

The simulation results are shown below-





(Note: Note the sequence of results obtained.)

Also, we observe 00000000 in between the sequence. These represent execution of NOP instructions.

The following report utilization was obtained after synthesis of the design.

Name	Slice LUTs (48000)	Slice Registers (96000)	F7 Muxes (32000)	F8 Muxes (16000)	Bonded IOB (338)	BUFGCTRL (32)
pipelined_regfile_4stage	1332	1165	193	64	34	1

The timing summary reported after synthesis is: -

- Max total delay – 14.465 ns

After implementation, the report utilization is: -

Hierarchy

Name	Slice LUTs (48000)	Slice Registers (96000)	F7 Muxes (32000)	F8 Muxes (16000)	Slice (16000)	LUT as Logic (48000)	LUT Flip Flop Pairs (48000)	Bonded IOB (338)	BUFGCTRL (32)
pipelined_regfile_4stage	1293	1197	193	64	692	1293	82	34	1

The timing summary is: -

- Max total setup delay – 23.242 ns

Observations

- There is difference between synthesis and implementation. Synthesis converts the RTL code to the netlist. Implementation tool takes this netlist as input and does optimization, placement and routing.
- The implementation reports less area in terms of Slice LUTs than synthesis. Reasoning becomes trivial from the first point.
- Also, implementation takes more time than synthesizing the same code.
- 3 Stage pipelined datapath takes more Slice LUTs than 4 stage pipelined datapath.
- 4 Stage pipelined datapath has a lower max total delay time than 3 staged pipelined datapath.
- Codes are uploaded on - https://github.com/abhinav9936/CS341lab/

Question still unanswered?

- Why 3 stage pipelined datapath takes more slice LUTs than 4 stage pipelined datapath?
- Why 4 stage pipelined datapath takes less total delay time than 3 stage pipelined datapath?