LAB3

Submitted by: Bar Harel 313611113

System Review

I have expanded the basic MIPS data path given in the moodle to support the required additional assembly instruction. The system is implemented by 5 components that represent the 5 data path parts: IF,ID,CONTROL,EXE,MEM. In addition, I added 3 more components for the input and output interface which are:

frequency divider – alternating the operating clock to adjust the speed in which the deta is presented in the output leds and digital 7-segment.

Hex_decoder – to translate the data that needs to present in the 7-segment .

Perifers – output interface .

RTL

MIPS- top level

logic usage: 2841

description: the top level entity. Receives the signals : clk, reset and the switch value. The output is the needed value to represent on the leds and 7 segment display. Other outputs are cor compilation use and convenience only.

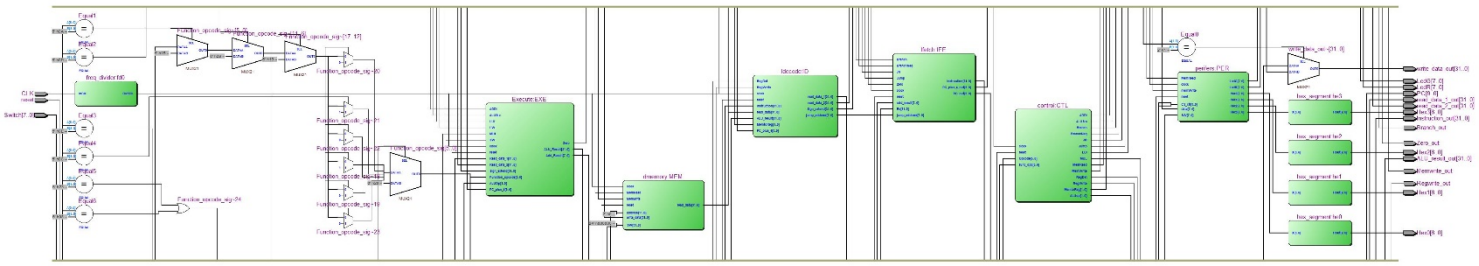

Fig 1

Ifetch

logic usage: 43

description: the instruction fetch stage of the mips data path. Receives a control signals and eject the instruction to be executed in the current operating cycle, and the next pc.

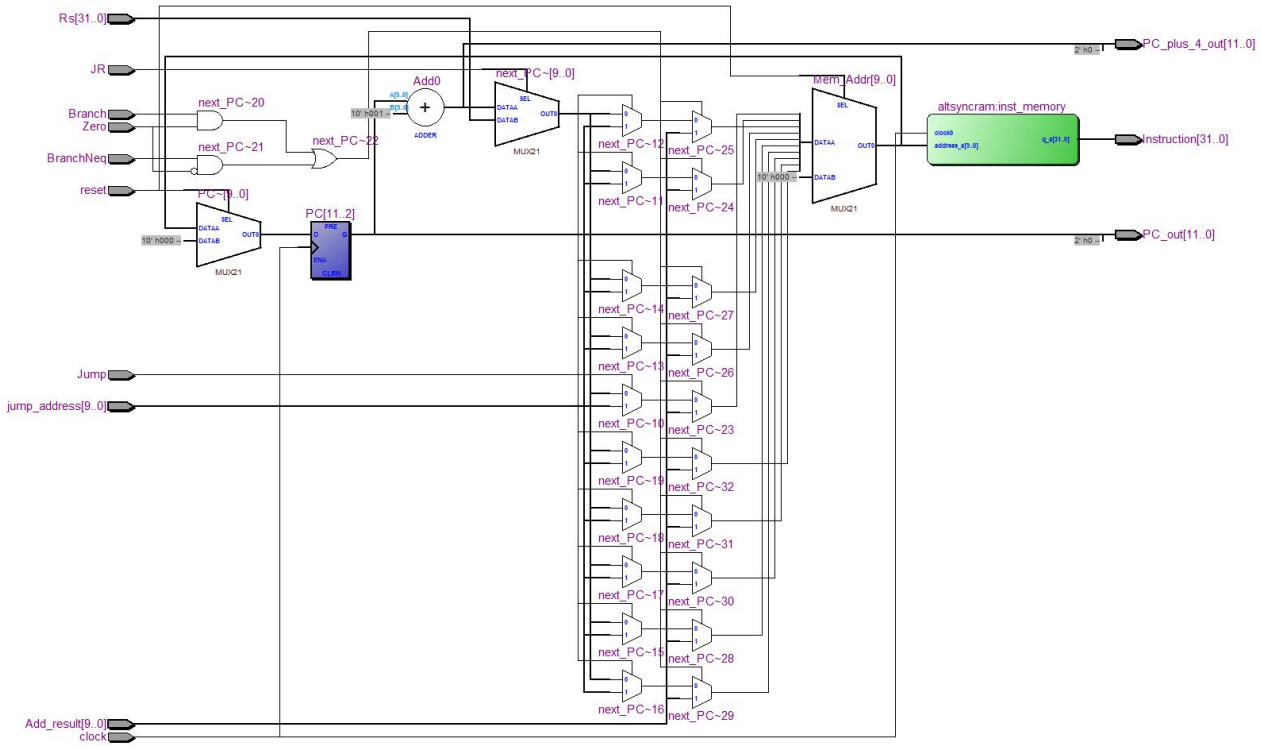

Fig 2

Idecode

logic usage:1473

description: decode the instruction fetched by the IF stage. Fetermindes the operators to be use in the EXE stage and handle the register file.

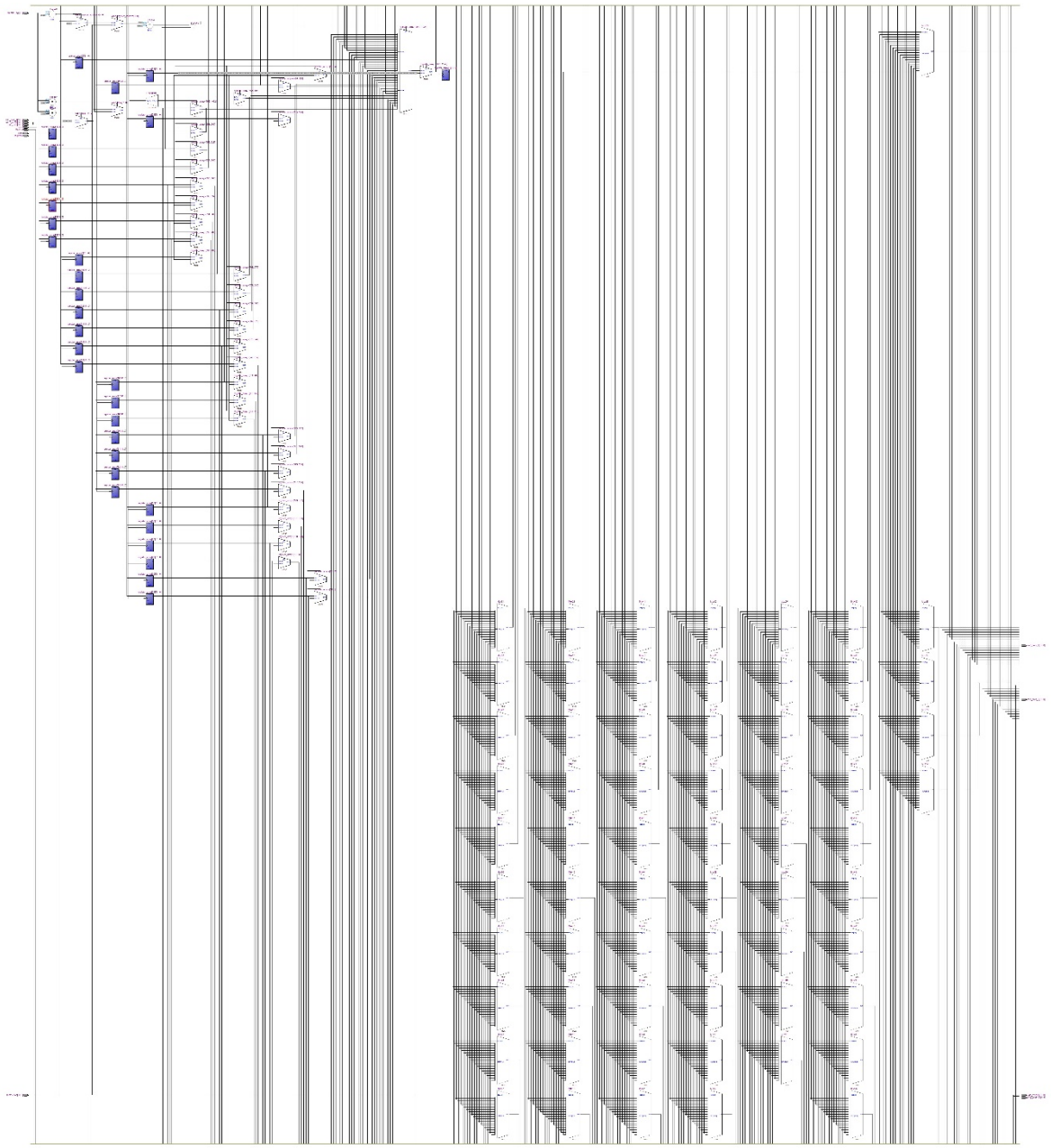

Fig 3

Control

logic usage: 21

description: receives the instruction opcode and function opcode and sends out control signals to all of the others components to be used to distinguished between the different instruction supported by the datapath.

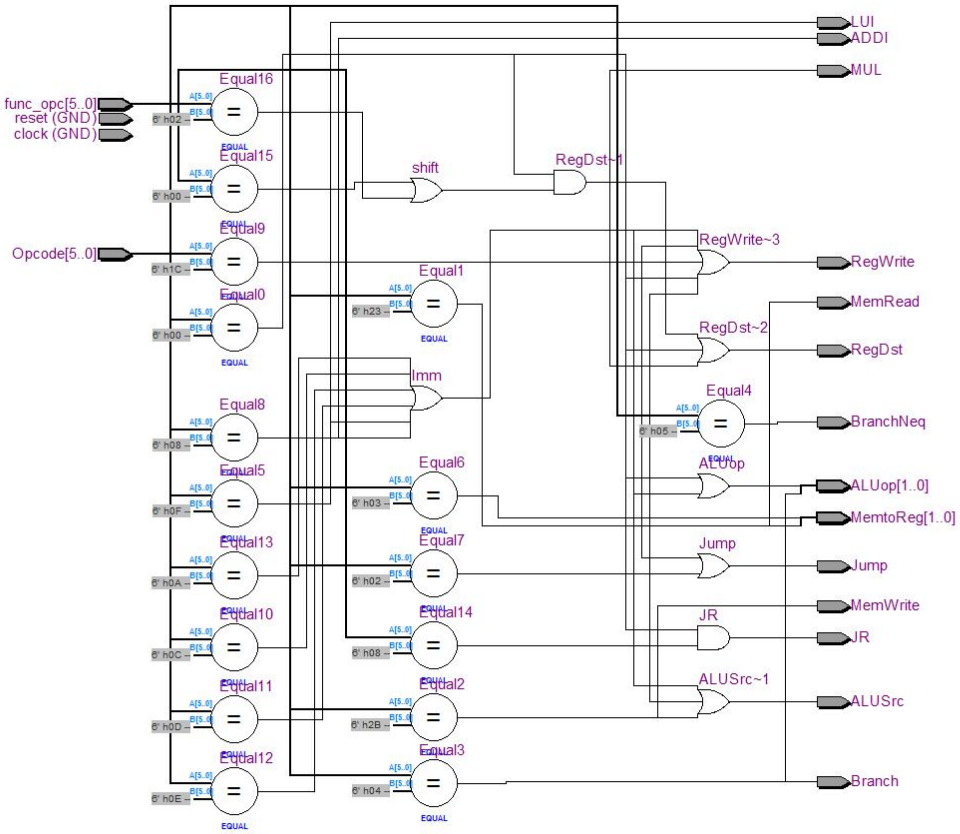

Fig 4

Execute

logic usage: 678

description: the datapath "calculator". Operates the needed arithmetic and logical functions on the operands fetched by the ID stage.

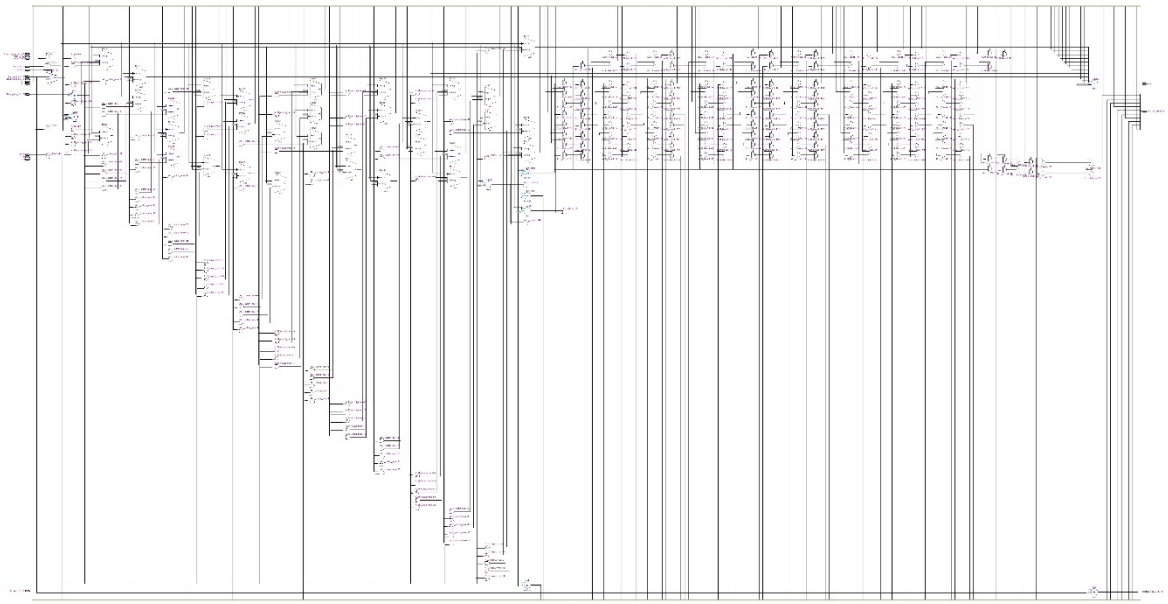

Fig 5

Dmemory

logic usage: 37

description: handle the system RAM.

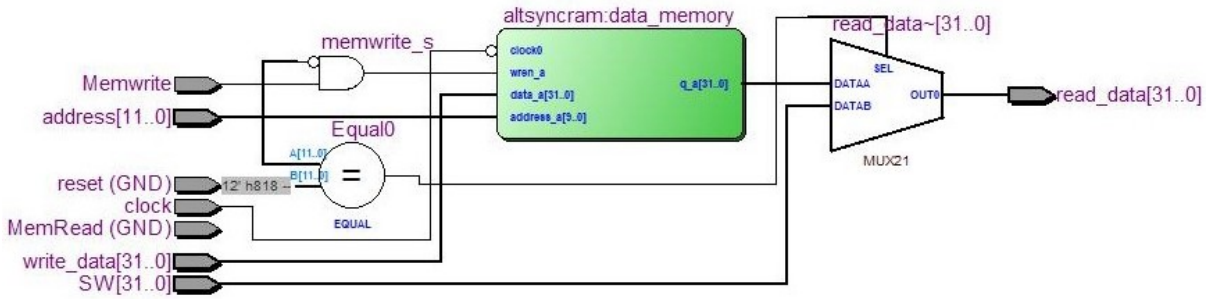

Fig 6

Frequency divider

logic usage: 2

description: use to slow down the display on the leds and 7 segment.

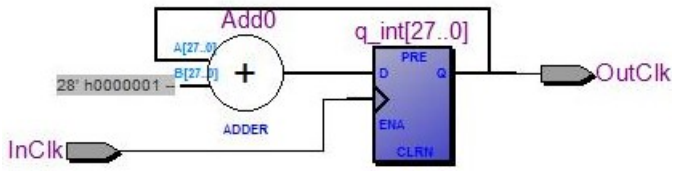

Fig7

f max

Slow Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name
1	INF MHz	36.89 MHz	Idecode:ID|register_array[10]
2	25.31 MHz	25.31 MHz	freq_divider:fd0|q_int[0]

Fig 8

the maximum frequency that my design can operates at. Determined by the critical path.

Hex_decoder

logic usage:7

description: translate the numbers from binary to the sequence needed to be enterd to the 7-segment so the numbers will be display properly.

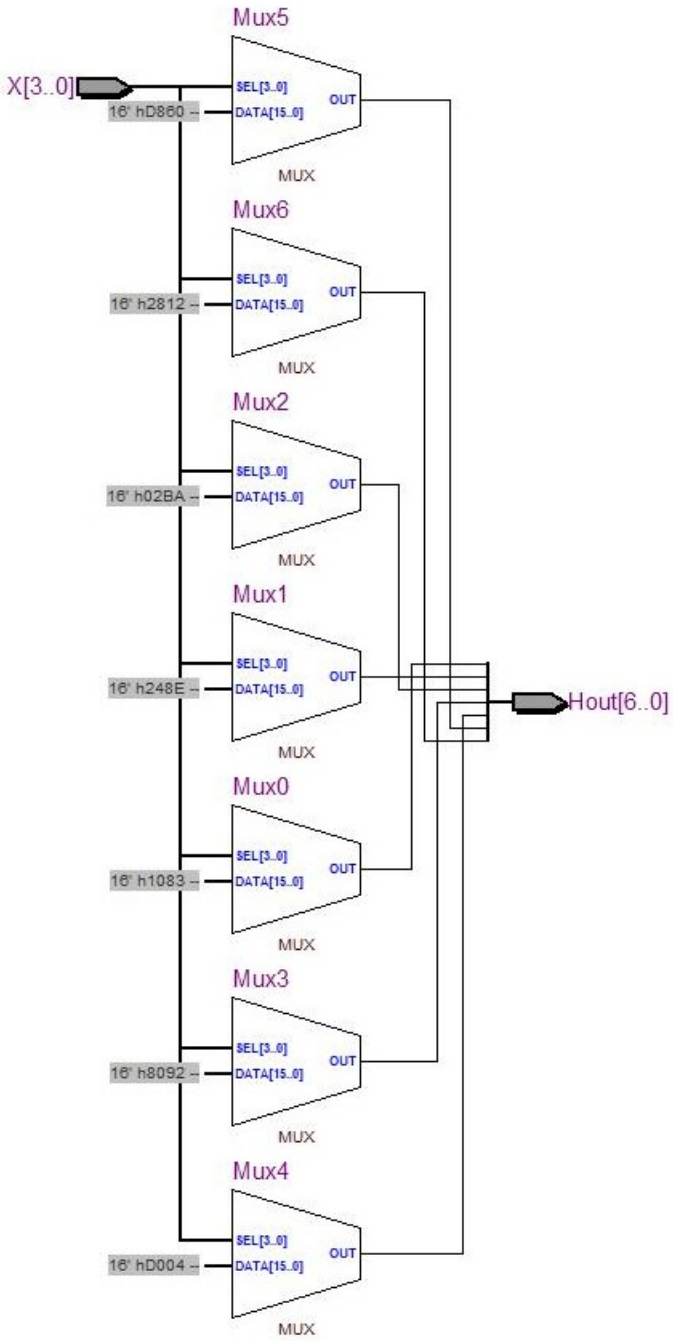

Fig 9

Perifers

logic usage:57

description: export the data to the output components at the appropriate time.

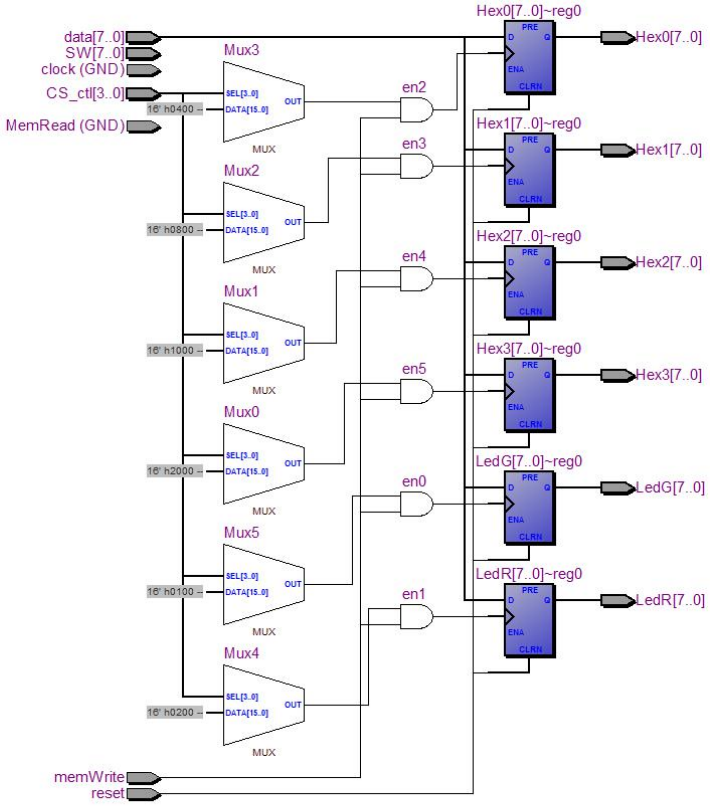

Fig 10

Critical path

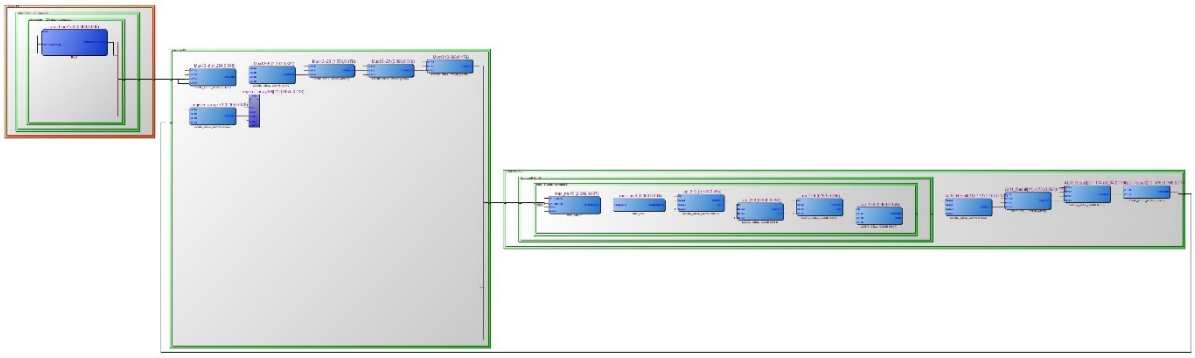

Fig11

The critical path starts at the RAM at the IF stage, goes through the write register mux into the EXE stage . in the EXE stage the longest instruction is the mult, from there it goes into the ID stage into the register array and waits to be written back to the GPR file at the clock second half of the cycle.

The critical path can be shortened be implementing the pipeline principle, in which we bound each stage of the datapath be flip flops.

The minimal path starts at the PC output, to the PC+4 adder, to the branch mux and back to the PC input.

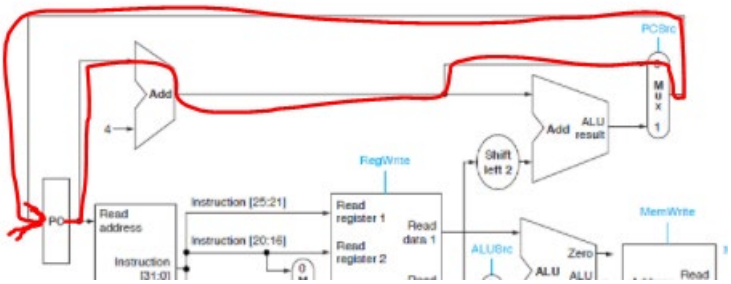

Fig 12

Testbenches

Test t0

overview: the files dmemory and program from t0 file are loaded to the modelsim simulator and the design operate as it should, counting upward by 1.

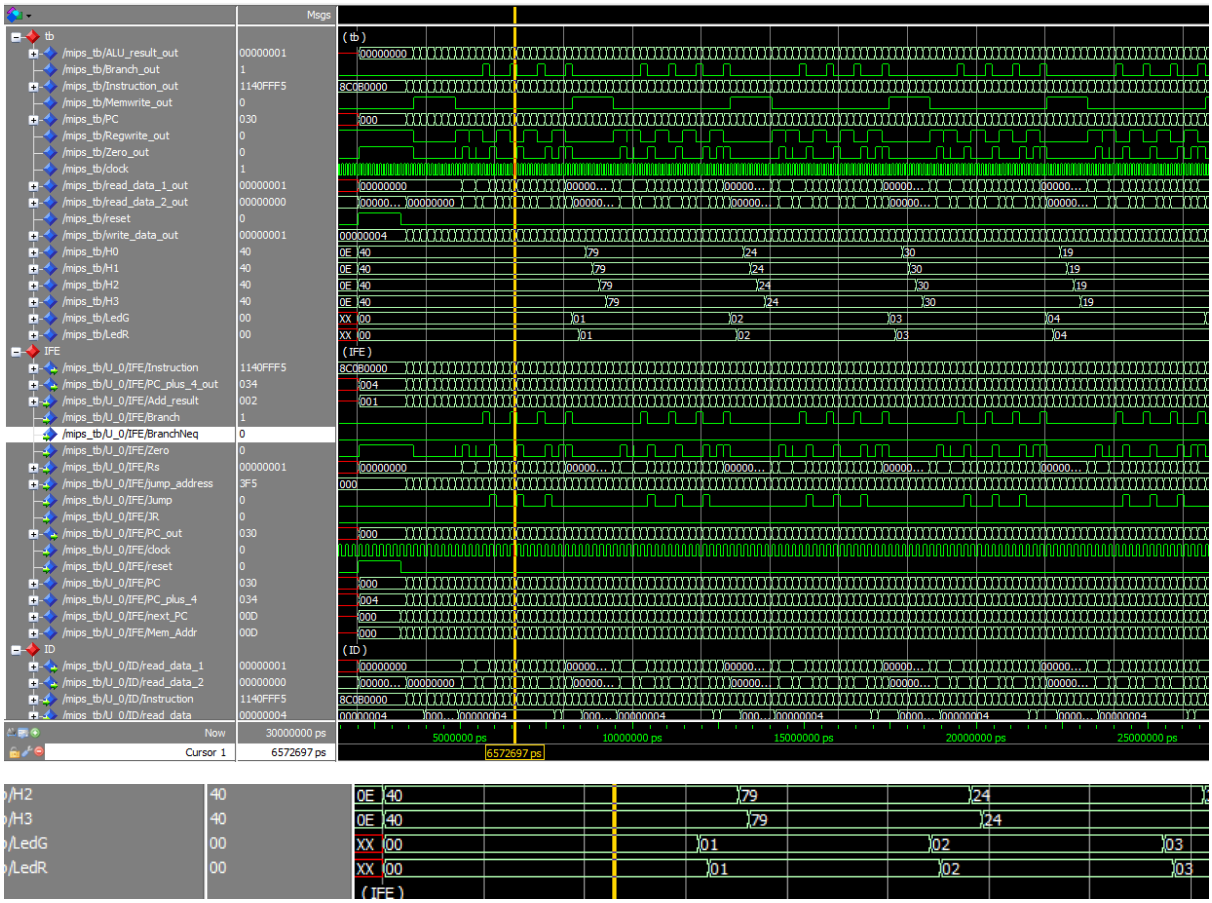

Fig 13

Signal tap of t0

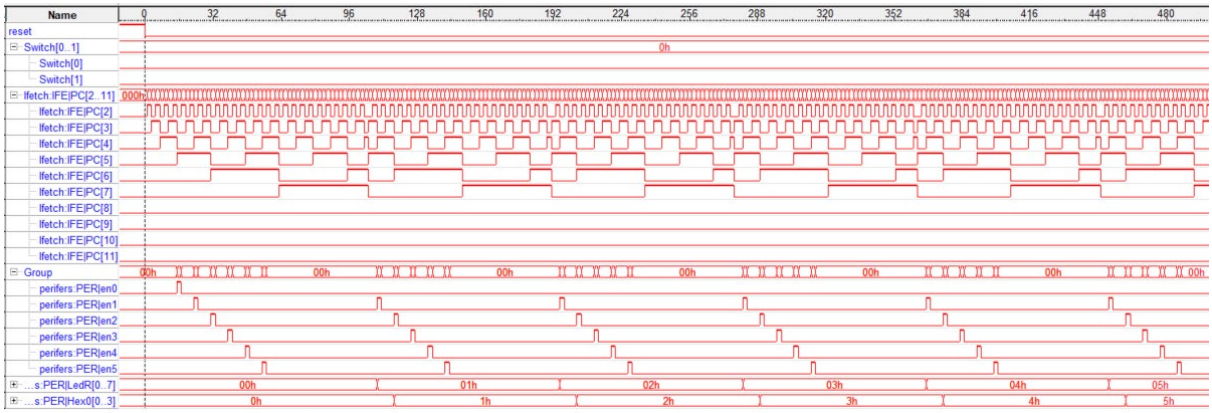

Fig 14