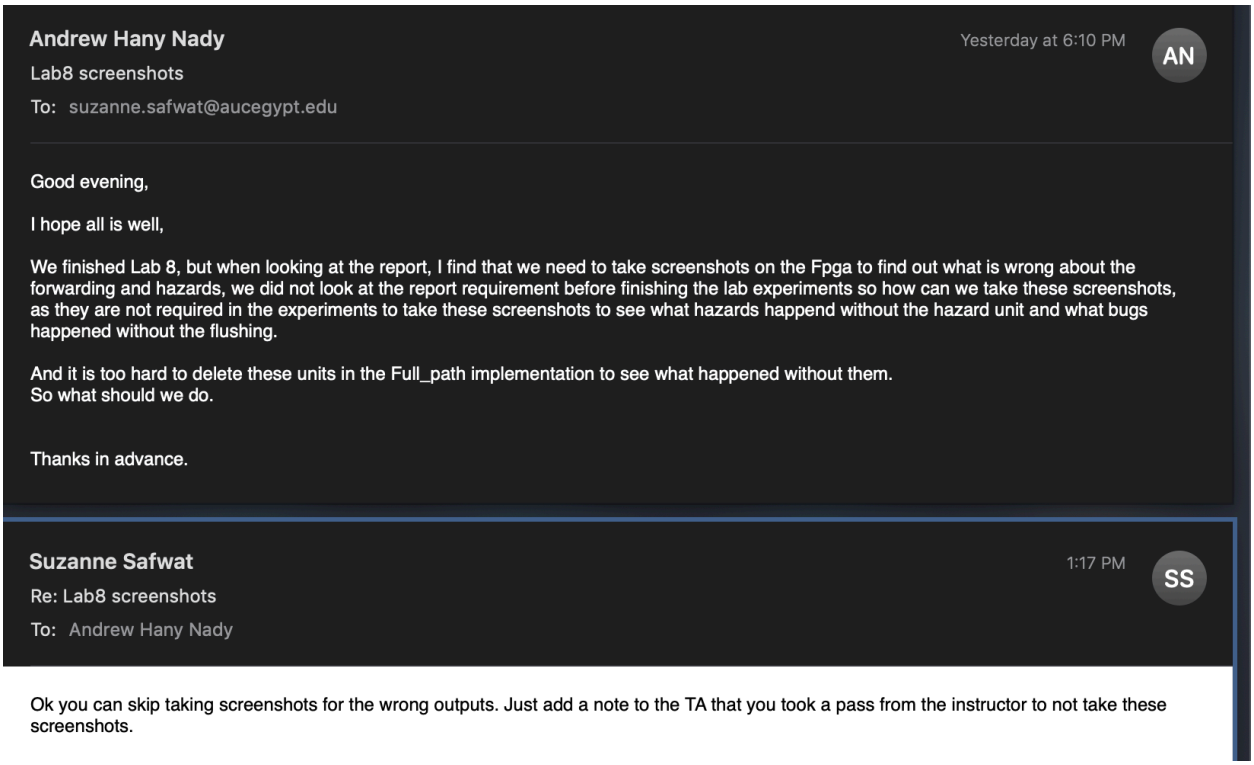
Lab 8

Andrew Nady

Question 2:

Code is provided in a separate file

Question 3&4:



Andrew Hany Nady Yesterday at 6:10 PM AN

Lab8 screenshots

To: suzanne.safwat@aucegypt.edu

Good evening,

I hope all is well,

We finished Lab 8, but when looking at the report, I find that we need to take screenshots on the Fpga to find out what is wrong about the forwarding and hazards, we did not look at the report requirement before finishing the lab experiments so how can we take these screenshots, as they are not required in the experiments to take these screenshots to see what hazards happend without the hazard unit and what bugs happened without the flushing.

And it is too hard to delete these units in the Full_path implementation to see what happened without them.
So what should we do.

Thanks in advance.

Suzanne Safwat 1:17 PM SS

Re: Lab8 screenshots

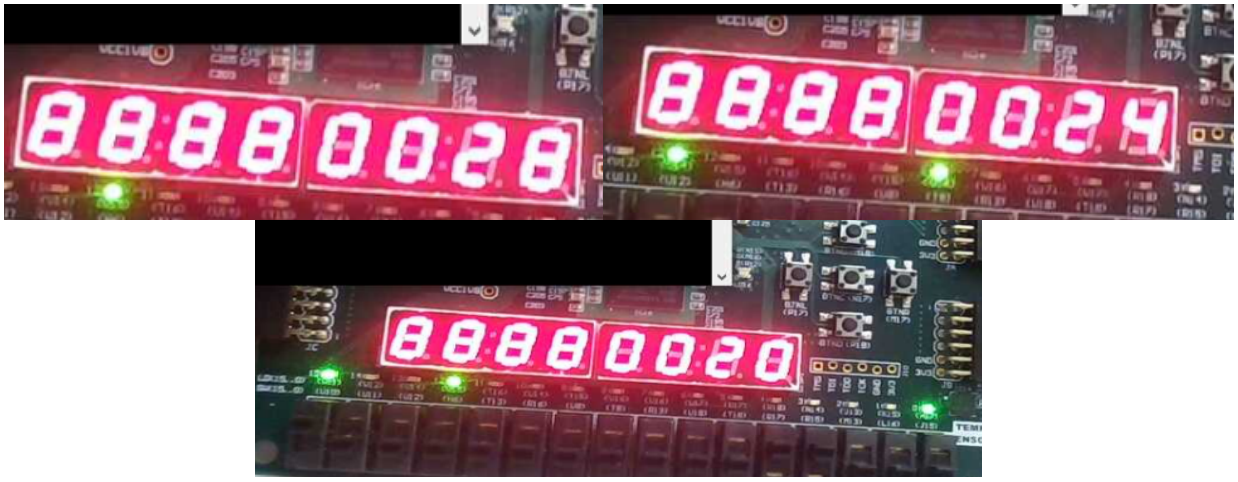
To: Andrew Hany Nady

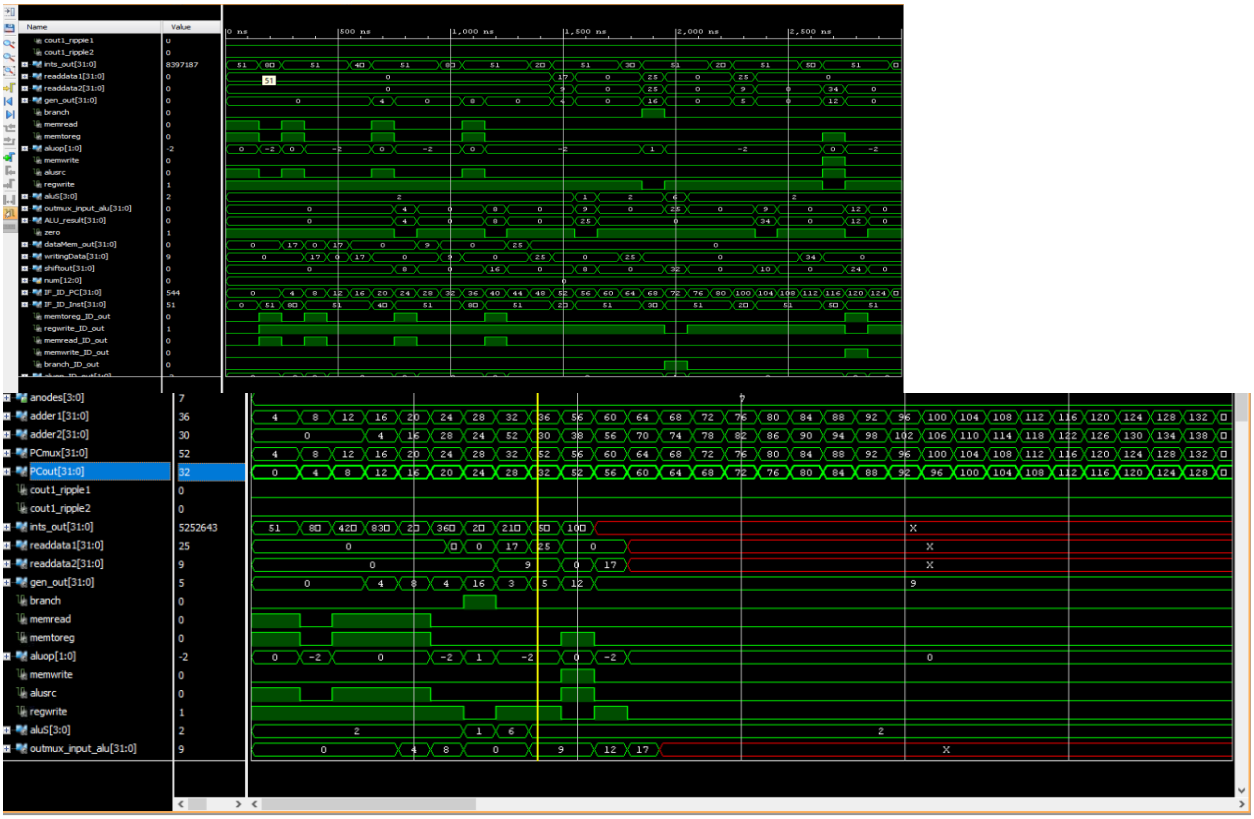
Ok you can skip taking screenshots for the wrong outputs. Just add a note to the TA that you took a pass from the instructor to not take these screenshots.

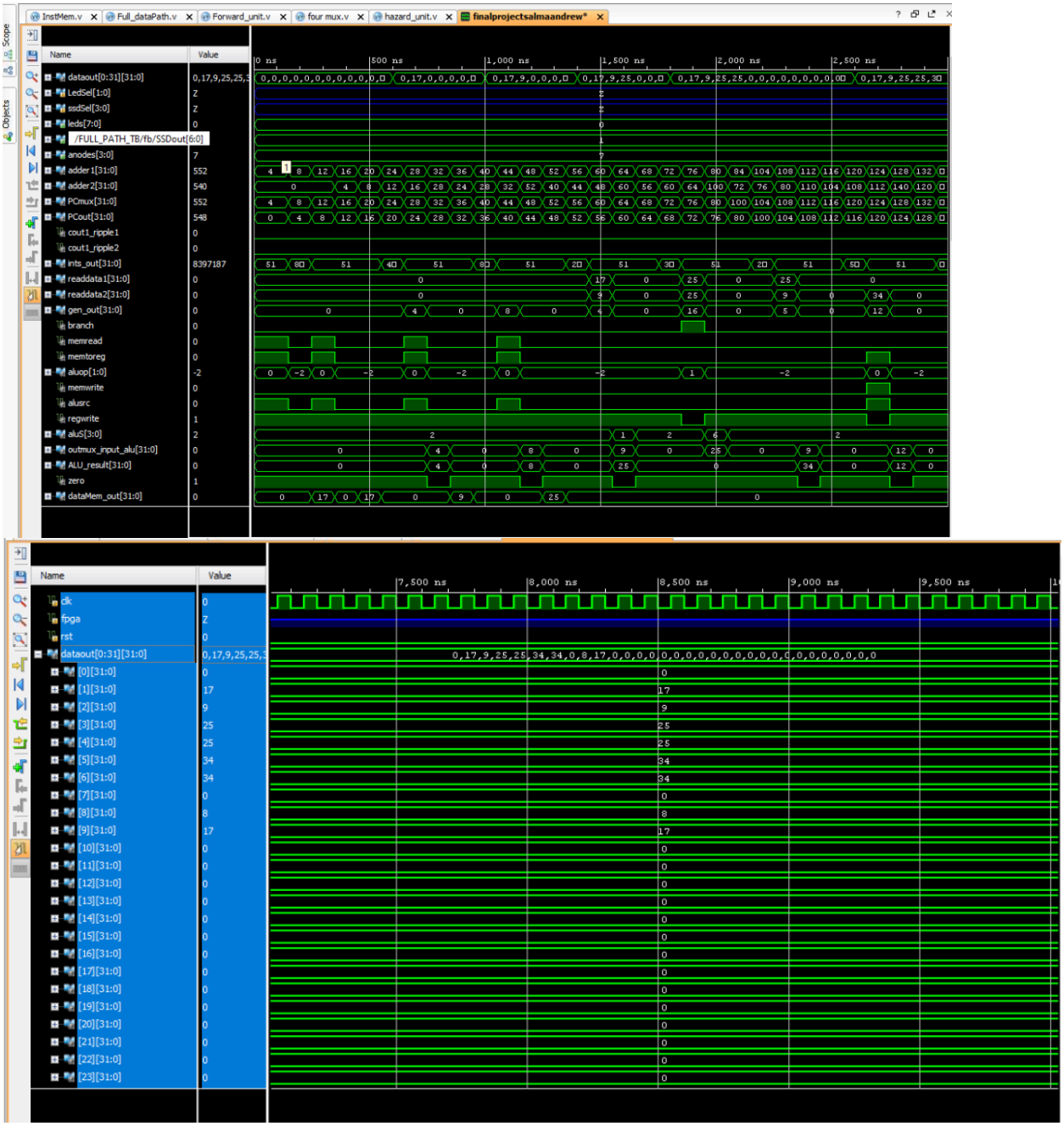
As we finished the lab before taking all the screenshots required, so we took a pass from the Doctor. However; we took some screenshot,

Experiment 2:

Here we are not expecting that the branches will work properly, as we only did the forwarding. But did not do the flushing yet. Here some screenshots on the FPGA and the simulation which has some issues in the branches.

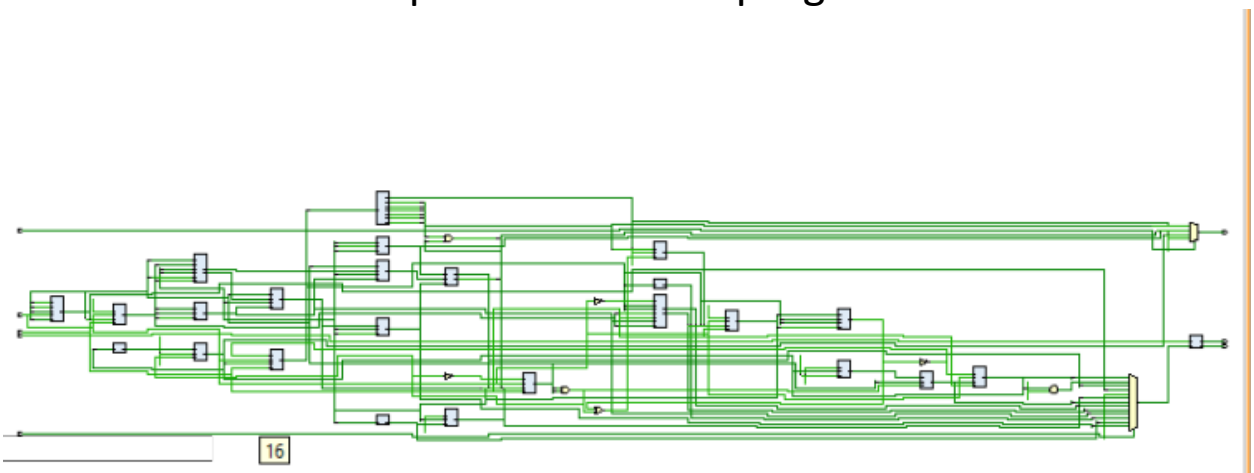


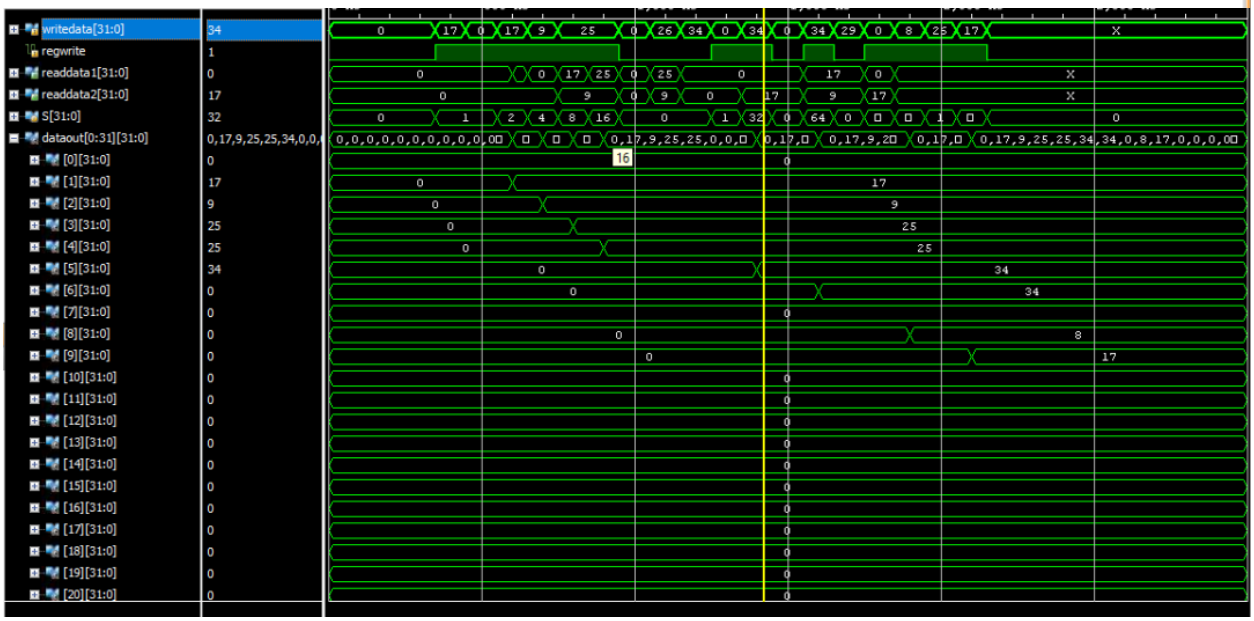



InstMem.v
Full_dataPath.v
Forward_unit.v
four mux.v
hazard_unit.v
finalprojectsalmaandrew*
Scope
Objects
Name
Value
0 ns
500 ns
1,000 ns
1,500 ns
2,000 ns
2,500 ns
dataout[0:31][31:0]
LedSel[1:0]
ssdSel[3:0]
leds[7:0]
/FULL_PATH_TB/fb/SSDout[6:0]
anodes[3:0]
adder1[31:0]
adder2[31:0]
PCmux[31:0]
PCout[31:0]
cout1_ripple1
cout1_ripple2
ints_out[31:0]
readdata1[31:0]
readdata2[31:0]
gen_out[31:0]
branch
memread
memtoreg
aluop[1:0]
memwrite
alusrc
regwrite
aluS[3:0]
outmux_input_alu[31:0]
ALU_result[31:0]
zero
dataMem_out[31:0]
7,500 ns
8,000 ns
8,500 ns
9,000 ns
9,500 ns
clk
fpga
rst
dataout[0:31][31:0]
0,17,9,25,25,34,34,0,8,17,0,0,0,0
[0][31:0]
[1][31:0]
[2][31:0]
[3][31:0]
[4][31:0]
[5][31:0]
[6][31:0]
[7][31:0]
[8][31:0]
[9][31:0]
[10][31:0]
[11][31:0]
[12][31:0]
[13][31:0]
[14][31:0]
[15][31:0]
[16][31:0]
[17][31:0]
[18][31:0]
[19][31:0]
[20][31:0]
[21][31:0]
[22][31:0]
[23][31:0]

Experiment 3:

In this point, we have a processor which is totally functional in all cases, and can deal with all hazards (data, control)
The below screenshot of the simulation indicates that all the results are true as expected from the program





Question 5:

If we have hazards in the mem and ex stages, then we need to forward it form the ex-stage, as it is the most updated value where if the value in mem stage of X3 =3, and then we modified it the ex-stage to be equal to 6, then it is more logical to forward the EX stage values

Question 6:

(EX/MEM.RegisterRd =/ 0): then means that rd =0 means that if the register not x0

That is because the register zero won’t be updates so we don’t need to forward it