CprE 381: Computer Organization and Assembly-Level Programming

Project Part 2 Report

Team Members: Andrew Wilken

Haris Khan

Wonjun Choi

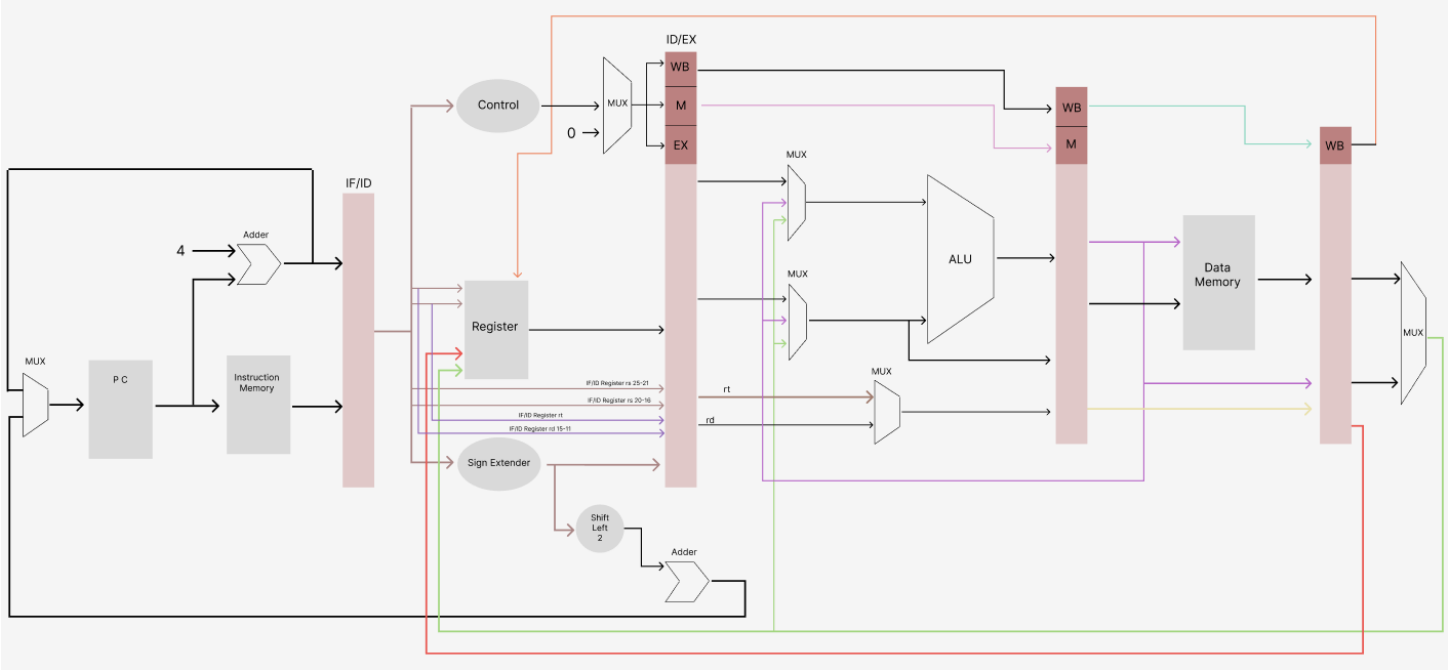
Project Teams Group # : 7-1

1. Software-Scheduled Pipeline.

[1.a] Come up with a global list of the datapath values and control signals that are required during each pipeline stage.

IF	ID	EX	MEM	WB
s_Inst	s_IFIDinst	s_IDEXfunc	s_EXmemMux	s_WBdmem
iCLK	s_IFIDnewPC	s_IDEXwb	s_EXmemRegoMux	s_WBmemALUo
s_PCFour	s_aluControl	s_IDEXmem	s_EXmemNewPC	s_WBmemNewPC
s_IMemAddr	s_Branch	s_IDEXex	s_EXmemWB	s_RegWrAddr
s_NextInstAddr	s_beqS	s_IDEXreg1	s_EXmemMEM	s_WBmemOut
iRST	s_j	s_IDEXreg2	s_EXmemZero	s_Halt
iInstLd	s_jr	s_IDEXrs	s_EXmemALUo	s_RegWr
iInstAddr	s_jal	s_IDEXrt	s_EXmemHalt	s_tempram
iInstExt	s_memRead	s_IDEXrd	s_EXmemMux	s_RegWrData
	s_memToReg	s_IDEXnewPC	s_EXmemALUo	
	s_DmemWR_temp	s_IDEXextend	iCLK	
	s_aluSrc	s_IDEXinst	iRST	
	s_RegWr_temp	s_IDEXhalt		
	s_signExtend	s_RegWrA_temp		
	s_regDst	s_aluout		
	s_Halt_temp	s_ZERO		
	s_sub	s_AS		
	s_ignrOvfl	s_mux2out		
	s_movn			
	s_regout1			
	s_regout2			
	s_extended			
	s_WRmuxO			
	iCLK			
	iRST			
	s_RegValue			
	s_BaddO			
	s_BmuxO			
	s_JshiftO			
	s_JmuxO			
	s_JRmuxO			
	s_Carry			
	s_extended			
	s_WRmuxO			
	s_WRtemp			
	s_RegWrData			

[1.b.ii] high-level schematic drawing of the interconnection between components.

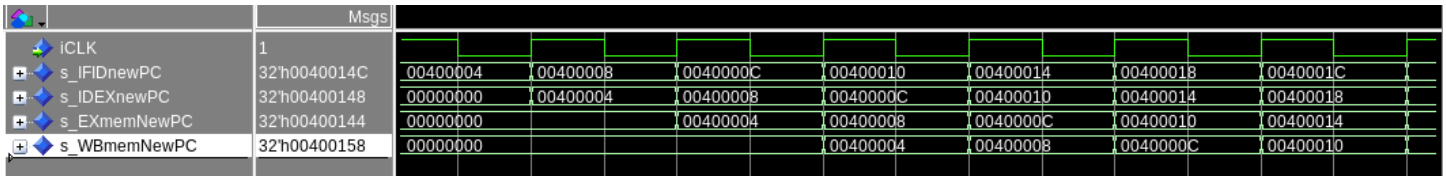


[1.c.i] include an annotated waveform in your writeup and provide a short discussion of result correctness.

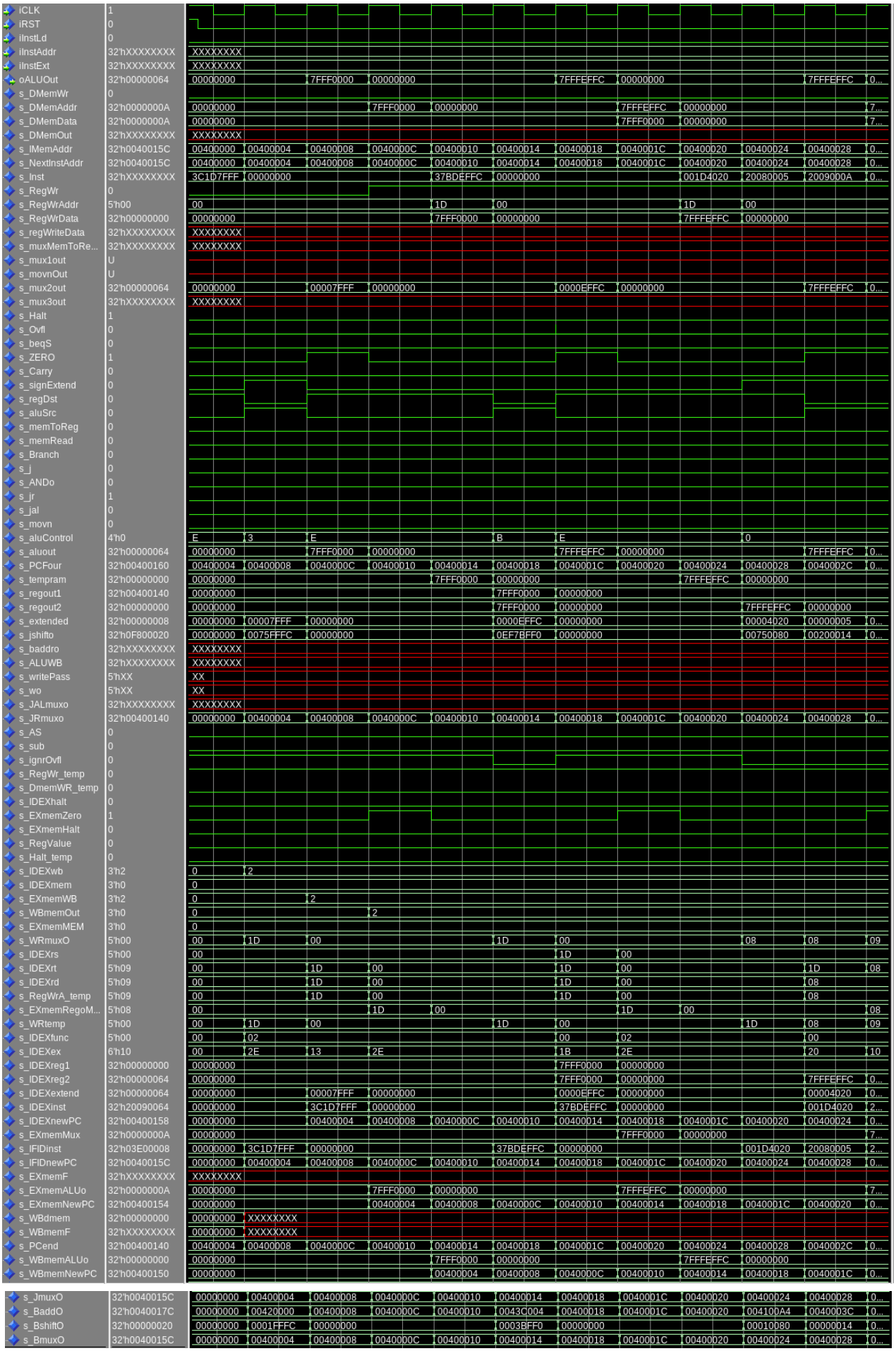
```
bash-4.2$ ./381_tf.sh test proj/mips/Inst_With_Nop.s
Using LAB Python Environment
Testing
All VHDL src files compiled successfully
Testing file: proj/mips/Inst_With_Nop.s
Mars simulation: pass
Modelsim simulation: pass
Test Result: pass
Mars Instructions: 73
Processor Cycles: 80
CPI:   1.1
Results in: output/Inst_With_Nop.s
-----------------------------------------------
bash-4.2$
```

We created a test that works with the pipeline processor because the base test of project part 1 lacked NOP instructions. For illustrational purposes, we have attached a waveform that just shows the pipeline stages rather than the whole waveform. If you look at the list of signals, we organized the pipeline stage in chronological order. The PC+4 value is passed from one pipeline step to the next, while the PC+8 value is passed from one pipeline step to the preceding. The whole waveform shown below includes more specific data and shows that the test was error-free. Signals that are undefined all the way through were unused for this portion, and since removed (no error).

[Pipeline stages in our waveform]



[Waveform: Base Tests with NOPs]

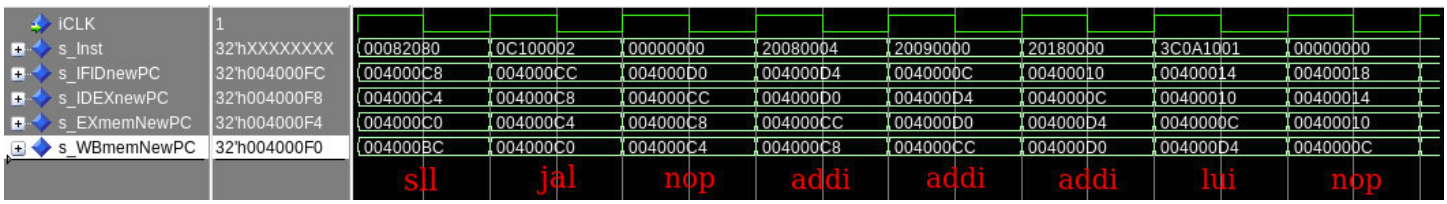


[1.c.ii] Include an annotated waveform in your writeup of two iterations or recursions of these programs executing correctly and provide a short discussion of result correctness. In your waveform and annotation, provide 3 different examples (at least one data-flow and one control-flow) of where you did not have to use the maximum number of NOPs.

```
bash-4.2$ ./381_tf.sh test proj/mips/Bubblesort_With_Nop.s
Using LAB Python Environment
Testing
All VHDL src files compiled successfully
Testing file: proj/mips/Bubblesort_With_Nop.s
Mars simulation: pass
Modelsim simulation: pass
Test Result: pass
Mars Instructions: 2078
Processor Cycles: 2256
CPI:  1.09
Results in: output/Bubblesort_With_Nop.s
```

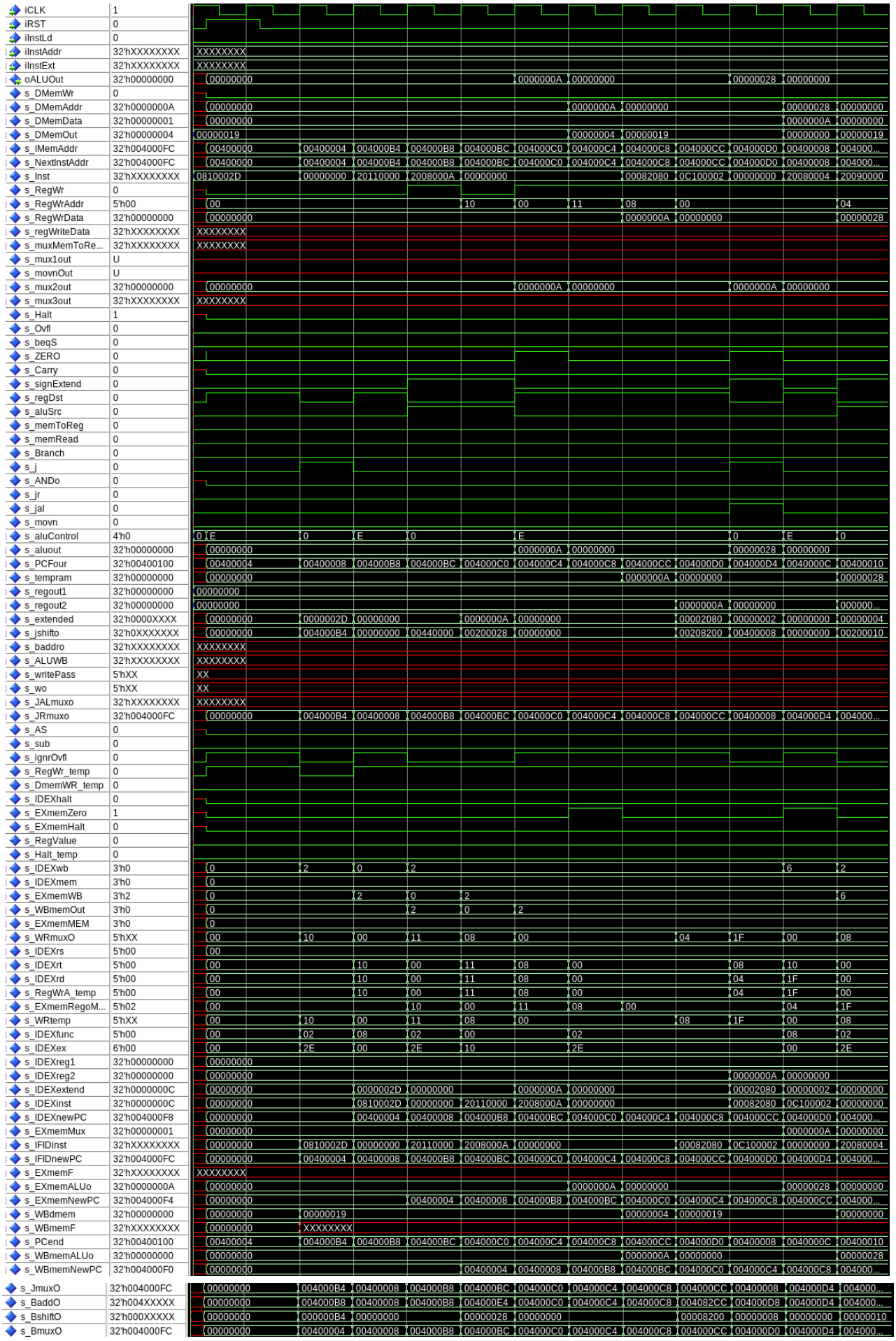
In contrast to project part 1, we used bubble sort mips code that included nop to test the pipeline.
In order to prevent errors at each step of the pipeline, we decided how many nops to use and placed them where they needed to be. As can be seen in the complete waveform screenshot below, all values are being located and transferred in the proper manner as a result.

[3 different examples in our waveform]



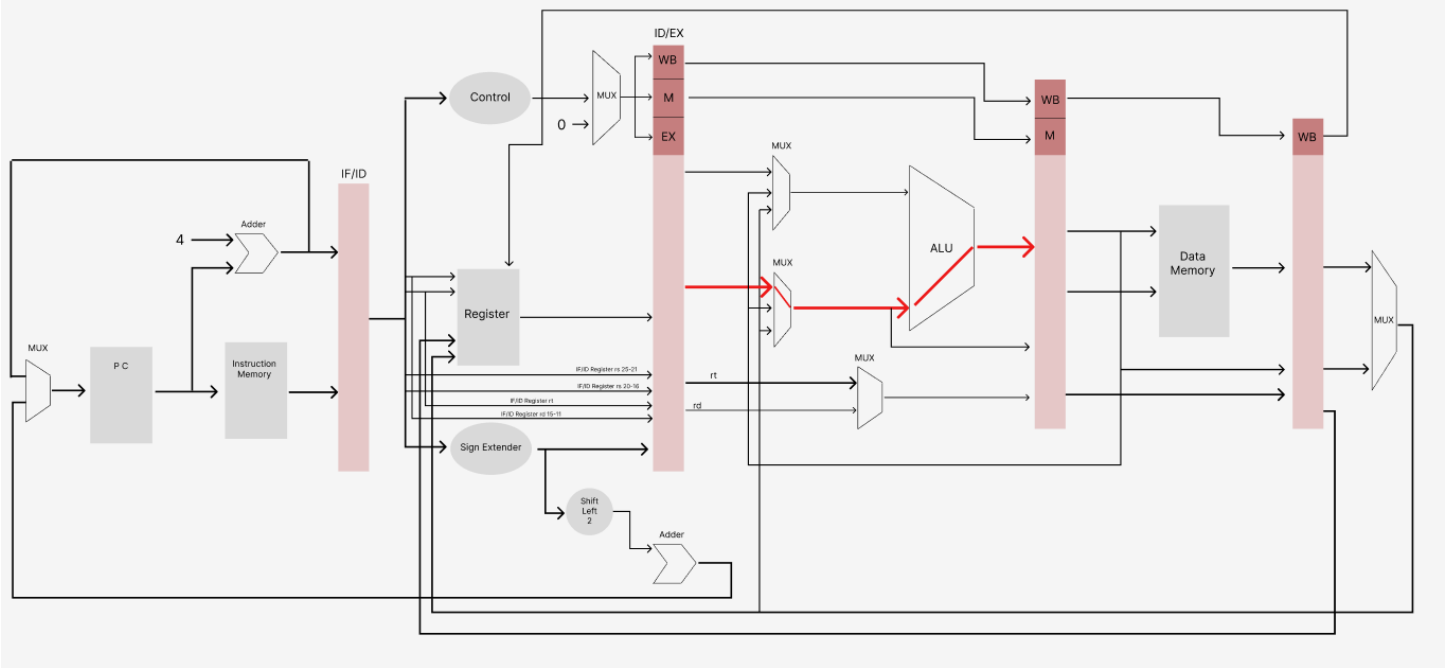
In this waveform you can see that it uses addi in succession without nops which is data flow and then it does jal which is control flow and these instructions are all used with either no nops or minimal nops.

[Waveform: Bubble sort with NOPs]



[1.d] report the maximum frequency your software-scheduled pipelined processor can run at and determine what your critical path is (specify each module/entity/component that this path goes through).

Our software-scheduled pipelined processor can run at a maximum frequency of 53.58 MHz in 18.625 nanoseconds.
Red lines in the diagram below denote critical routes. Specifically, starting with the ID/EX register, moving on to the ALU, EX/MEM register, and concluding.



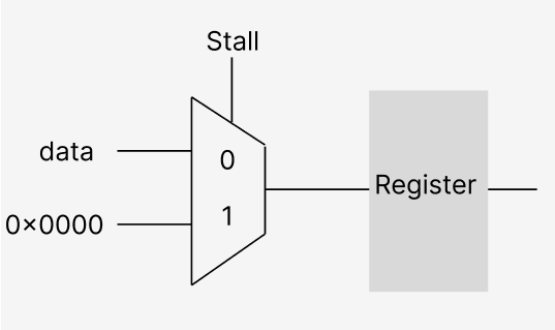
[Synthesis_sw: timing.txt]

Open | *timing.txt ~/Downloads | Save

```
# CprE 381 toolflow Timing dump

FMax: 53.69mhz Clk Constraint: 20.00ns Slack: 1.37ns

The path is given below

 ================================================================
 From Node    : IDEX:id_ex|dffg_N:RegOut2|s_Q[1]
 To Node      : EXmem:ex_mem|dffg_N:RegALU|s_Q[31]
 Launch Clock : iCLK
 Latch Clock  : iCLK
 Data Arrival Path:
 Total (ns)  Incr (ns)     Type  Element
 ==========  ========= ==  ====  ===================================
      0.000     0.000           launch edge time
      3.090     3.090  R        clock network delay
      3.322     0.232     uTco  IDEX:id_ex|dffg_N:RegOut2|s_Q[1]
      3.322     0.000 FF  CELL  id_ex|RegOut2|s_Q[1]|q
      5.078     1.756 FF    IC  immMux|\G_NBit_MUX:1:MUXI|g_org1|o_F~0|dataa
      5.502     0.424 FF  CELL  immMux|\G_NBit_MUX:1:MUXI|g_org1|o_F~0|combout
      5.752     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:1:MUXI|g_or|o_F~1|datad
      5.877     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:1:MUXI|g_or|o_F~1|combout
      6.135     0.258 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:2:MUXI|g_or|o_F~0|datac
      6.416     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:2:MUXI|g_or|o_F~0|combout
      6.671     0.255 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:3:MUXI|g_or|o_F~0|datac
      6.952     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:3:MUXI|g_or|o_F~0|combout
      7.202     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:4:MUXI|g_or|o_F~0|datad
      7.327     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:4:MUXI|g_or|o_F~0|combout
      7.585     0.258 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:5:MUXI|g_or|o_F~0|datac
      7.866     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:5:MUXI|g_or|o_F~0|combout
      8.114     0.248 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:6:MUXI|g_or|o_F~0|datad
      8.239     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:6:MUXI|g_or|o_F~0|combout
      8.488     0.249 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:7:MUXI|g_or|o_F~0|datad
      8.613     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:7:MUXI|g_or|o_F~0|combout
      8.863     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:8:MUXI|g_or|o_F~0|datad
      8.988     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:8:MUXI|g_or|o_F~0|combout
      9.383     0.395 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:9:MUXI|g_or|o_F~0|datad
      9.508     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:9:MUXI|g_or|o_F~0|combout
      9.759     0.251 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:10:MUXI|g_or|o_F~0|datad
      9.884     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:10:MUXI|g_or|o_F~0|combout
     10.135     0.251 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:11:MUXI|g_or|o_F~0|datad
     10.260     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:11:MUXI|g_or|o_F~0|combout
     10.511     0.251 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:12:MUXI|g_or|o_F~0|datad
     10.636     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:12:MUXI|g_or|o_F~0|combout
     10.885     0.249 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:13:MUXI|g_or|o_F~0|datad
     11.010     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:13:MUXI|g_or|o_F~0|combout
     11.265     0.255 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:14:MUXI|g_or|o_F~0|datac
     11.546     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:14:MUXI|g_or|o_F~0|combout
     11.794     0.248 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:15:MUXI|g_or|o_F~0|datad
     11.919     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:15:MUXI|g_or|o_F~0|combout
     12.176     0.257 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:16:MUXI|g_or|o_F~0|datac
     12.457     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:16:MUXI|g_or|o_F~0|combout
     12.712     0.255 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:17:MUXI|g_or|o_F~0|datac
     12.993     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:17:MUXI|g_or|o_F~0|combout
     13.243     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:18:MUXI|g_or|o_F~0|datad
     13.368     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:18:MUXI|g_or|o_F~0|combout
     13.618     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:19:MUXI|g_or|o_F~0|datad
     13.743     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:19:MUXI|g_or|o_F~0|combout
     13.995     0.252 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:20:MUXI|g_or|o_F~0|datad
     14.120     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:20:MUXI|g_or|o_F~0|combout
     14.380     0.260 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:21:MUXI|g_or|o_F~0|datac
     14.661     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:21:MUXI|g_or|o_F~0|combout
     14.914     0.253 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:22:MUXI|g_or|o_F~0|datad
     15.039     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:22:MUXI|g_or|o_F~0|combout
     15.337     0.298 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:23:MUXI|g_or|o_F~0|dataa
     15.761     0.424 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:23:MUXI|g_or|o_F~0|combout
     16.011     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:24:MUXI|g_or|o_F~0|datad
     16.136     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:24:MUXI|g_or|o_F~0|combout
     16.559     0.423 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:25:MUXI|g_or|o_F~0|datac
     16.840     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:25:MUXI|g_or|o_F~0|combout
     17.090     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:26:MUXI|g_or|o_F~0|datad
     17.215     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:26:MUXI|g_or|o_F~0|combout
     17.465     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:27:MUXI|g_or|o_F~0|datad
     17.590     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:27:MUXI|g_or|o_F~0|combout
     17.828     0.238 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:28:MUXI|g_or|o_F~0|datad
     17.953     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:28:MUXI|g_or|o_F~0|combout
     18.211     0.258 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:29:MUXI|g_or|o_F~0|datac
     18.492     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:29:MUXI|g_or|o_F~0|combout
     18.727     0.235 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:30:MUXI|g_or|o_F~0|datad
     18.852     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:30:MUXI|g_or|o_F~0|combout
     19.082     0.230 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:31:MUXI|g_xor2|o_F|datad
     19.232     0.150 FR  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:31:MUXI|g_xor2|o_F|combout
     19.458     0.226 RR    IC  mainALU|g_mux9|Mux0~5|datad
     19.613     0.155 RR  CELL  mainALU|g_mux9|Mux0~5|combout
     20.878     1.265 RR    IC  mainALU|g_mux9|Mux0~6|datac
     21.165     0.287 RR  CELL  mainALU|g_mux9|Mux0~6|combout
     21.367     0.202 RR    IC  mainALU|g_mux9|Mux0~7|datad
     21.506     0.139 RF  CELL  mainALU|g_mux9|Mux0~7|combout
     21.506     0.000 FF    IC  ex_mem|RegALU|s_Q[31]|d
     21.610     0.104 FF  CELL  EXmem:ex_mem|dffg_N:RegALU|s_Q[31]
 Data Required Path:
 Total (ns)  Incr (ns)     Type  Element
 ==========  ========= ==  ====  ===================================
     20.000    20.000           latch edge time
     22.977     2.977  R        clock network delay
     22.985     0.008           clock pessimism removed
     22.965    -0.020           clock uncertainty
     22.983     0.018     uTsu  EXmem:ex_mem|dffg_N:RegALU|s_Q[31]
 Data Arrival Time  :    21.610
 Data Required Time :    22.983
 Slack              :     1.373
 ================================================================
```

2. Hardware-Scheduled Pipeline.

[2.a.ii] Draw a simple schematic showing how you could implement stalling and flushing operations given an ideal N-bit register.

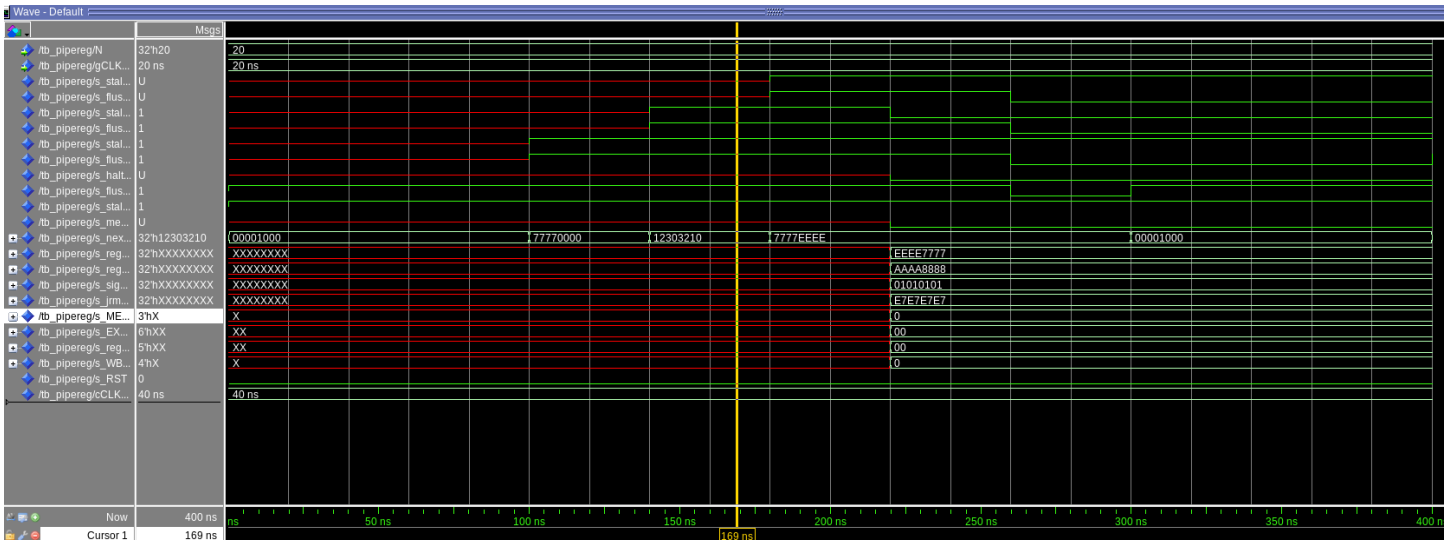


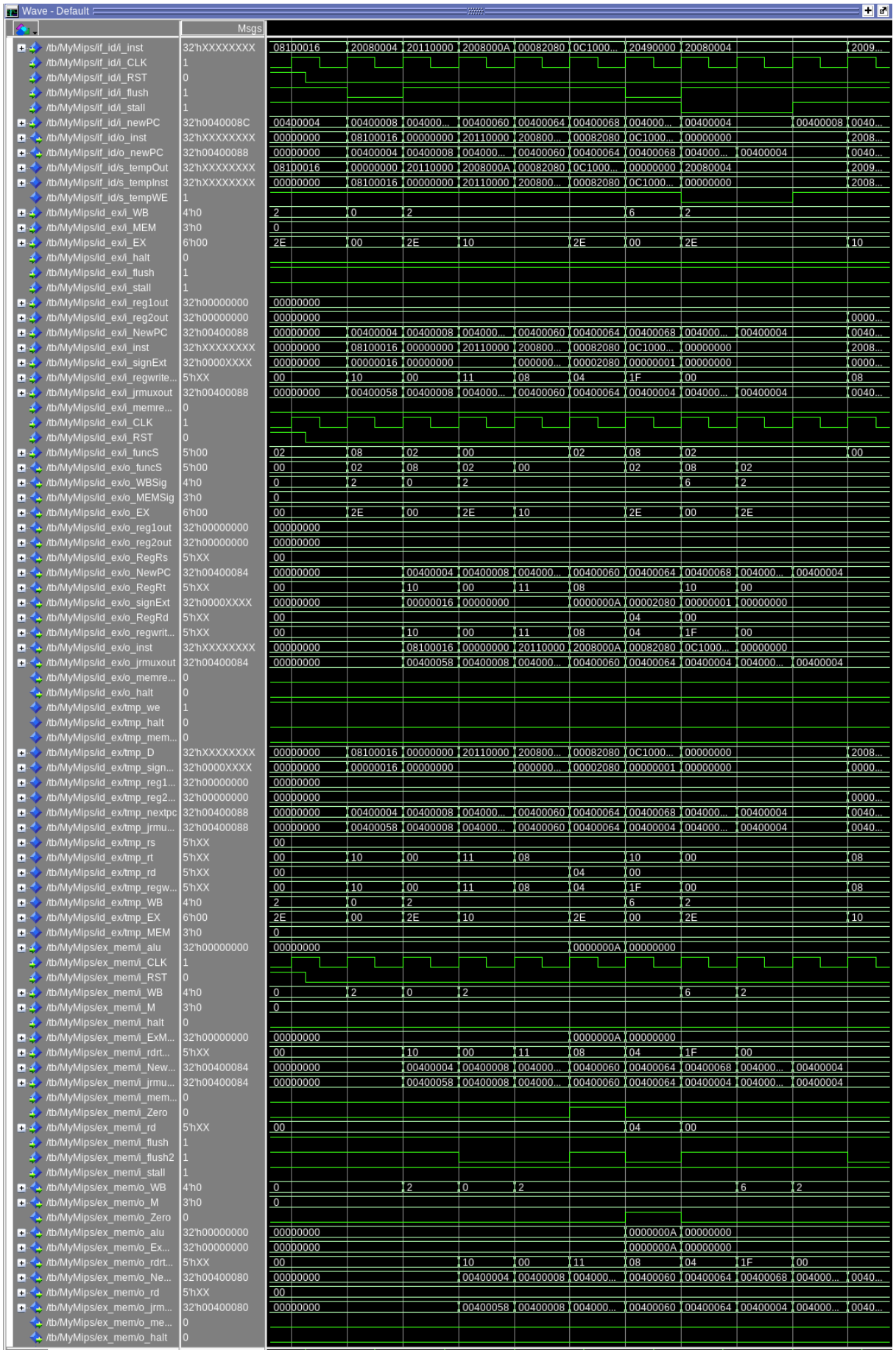
[2.a.iii] Create a testbench that instantiates all four of the registers in a single design. Show that values that are stored in the initial IF/ID register are available as expected four cycles later, and that new values can be inserted into the pipeline every single cycle. Most importantly, this testbench should also test that each pipeline register can be individually stalled or flushed.

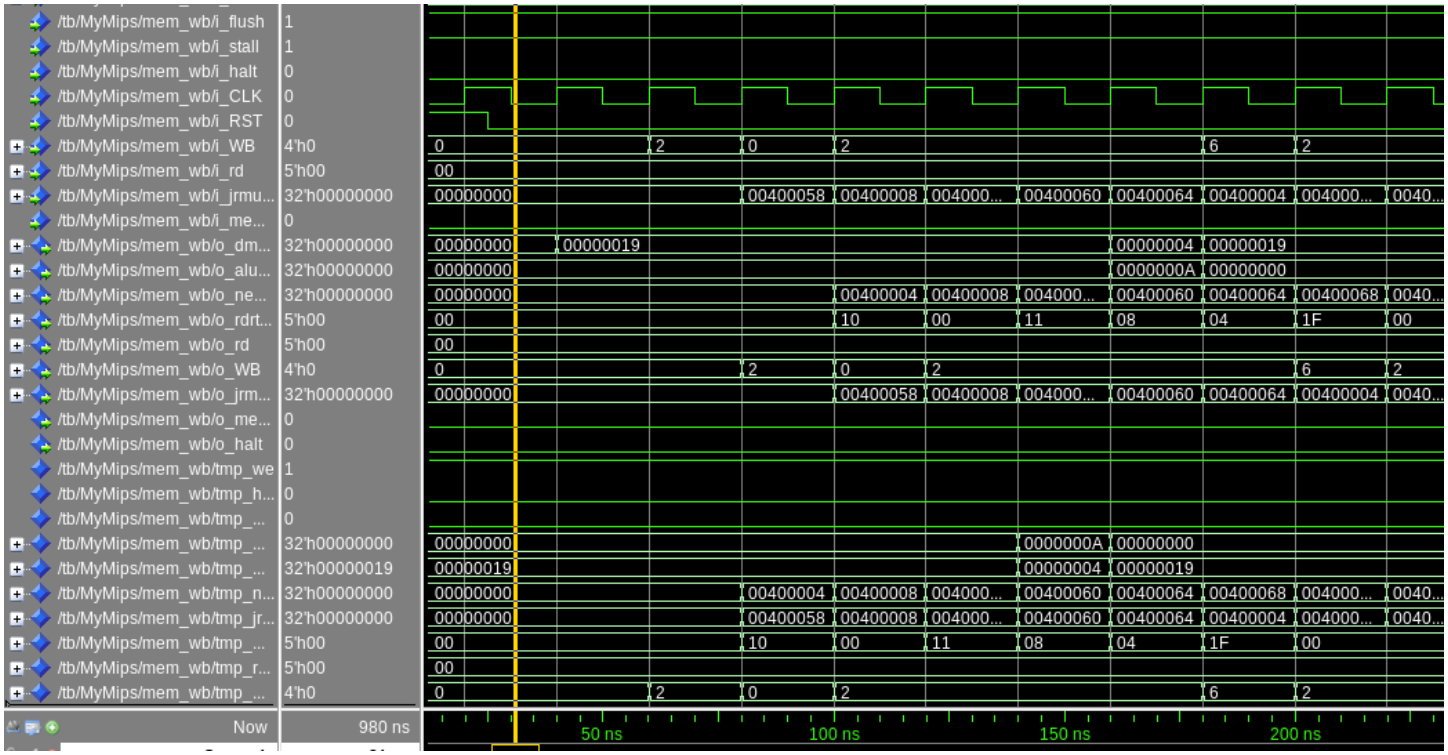
```
stallIF => s_stallIF,
stallID => s_IDEXstall,
stallEX => s_EXmemstall,
stallWB => s_memWBstall,
flushIF => s_IFIDflush,
flushID => s_IDEXflush,
flushEX => s_EXmemflush,
flushWB => s_memWBflush,
flushtemp => s_flush2
```

A testbench file exists but we included additional waveforms to show correctness. Below are the four register values when running grendel.s as well as the testbench (tb is first). As you can see, the values get passed down the pipeline as expected through all stages. Each stage can be stalled or flushed with the following code in the hazard unit. Since each stage has its own stall and flush, this is possible.

[Pipeline Register Waveform]




Wave - Default
Msgs
/tb/MyMips/if_id/i_inst 32'hXXXXXXXX 08100016 20080004 20110000 2008000A 00082080 0C1000... 20490000 20080004 2009...
/tb/MyMips/if_id/i_CLK 1
/tb/MyMips/if_id/i_RST 0
/tb/MyMips/if_id/i_flush 1
/tb/MyMips/if_id/i_stall 1
/tb/MyMips/if_id/i_newPC 32'h0040008C 00400004 00400008 004000... 00400060 00400064 00400068 004000... 00400004 00400008 0040...
/tb/MyMips/if_id/o_inst 32'hXXXXXXXX 00000000 08100016 00000000 20110000 200800... 00082080 0C1000... 00000000 2008...
/tb/MyMips/if_id/o_newPC 32'h00400088 00000000 00400004 00400008 004000... 00400060 00400064 00400068 004000... 00400004 0040...
/tb/MyMips/if_id/s_tempOut 32'hXXXXXXXX 08100016 00000000 20110000 2008000A 00082080 0C1000... 00000000 20080004 2009...
/tb/MyMips/if_id/s_tempInst 32'hXXXXXXXX 00000000 08100016 00000000 20110000 200800... 00082080 0C1000... 00000000 2008...
/tb/MyMips/if_id/s_tempWE 1
/tb/MyMips/id_ex/i_WB 4'h0 2 0 2 6 2
/tb/MyMips/id_ex/i_MEM 3'h0 0
/tb/MyMips/id_ex/i_EX 6'h00 2E 00 2E 10 2E 00 2E 10
/tb/MyMips/id_ex/i_halt 0
/tb/MyMips/id_ex/i_flush 1
/tb/MyMips/id_ex/i_stall 1
/tb/MyMips/id_ex/i_reg1out 32'h00000000 00000000
/tb/MyMips/id_ex/i_reg2out 32'h00000000 00000000 0000...
/tb/MyMips/id_ex/i_NewPC 32'h00400088 00000000 00400004 00400008 004000... 00400060 00400064 00400068 004000... 00400004 0040...
/tb/MyMips/id_ex/i_inst 32'hXXXXXXXX 00000000 08100016 00000000 20110000 200800... 00082080 0C1000... 00000000 2008...
/tb/MyMips/id_ex/i_signExt 32'h0000XXXX 00000000 00000016 00000000 000000... 00002080 00000001 00000000 0000...
/tb/MyMips/id_ex/i_regwrite... 5'hXX 00 10 00 11 08 04 1F 00 08
/tb/MyMips/id_ex/i_jrmuxout 32'h00400088 00000000 00400058 00400008 004000... 00400060 00400064 00400004 004000... 00400004 0040...
/tb/MyMips/id_ex/i_memre... 0
/tb/MyMips/id_ex/i_CLK 1
/tb/MyMips/id_ex/i_RST 0
/tb/MyMips/id_ex/i_funcS 5'h00 02 08 02 00 02 08 02 00
/tb/MyMips/id_ex/o_funcS 5'h00 00 02 08 02 00 02 08 02
/tb/MyMips/id_ex/o_WBSig 4'h0 0 2 0 2 6 2
/tb/MyMips/id_ex/o_MEMSig 3'h0 0
/tb/MyMips/id_ex/o_EX 6'h00 00 2E 00 2E 10 2E 00 2E
/tb/MyMips/id_ex/o_reg1out 32'h00000000 00000000
/tb/MyMips/id_ex/o_reg2out 32'h00000000 00000000
/tb/MyMips/id_ex/o_RegRs 5'hXX 00
/tb/MyMips/id_ex/o_NewPC 32'h00400084 00000000 00400004 00400008 004000... 00400060 00400064 00400068 004000... 00400004
/tb/MyMips/id_ex/o_RegRt 5'hXX 00 10 00 11 08 10 00
/tb/MyMips/id_ex/o_signExt 32'h0000XXXX 00000000 00000016 00000000 0000000A 00002080 00000001 00000000
/tb/MyMips/id_ex/o_RegRd 5'hXX 00 04 00
/tb/MyMips/id_ex/o_regwrit... 5'hXX 00 10 00 11 08 04 1F 00
/tb/MyMips/id_ex/o_inst 32'hXXXXXXXX 00000000 08100016 00000000 20110000 2008000A 00082080 0C1000... 00000000
/tb/MyMips/id_ex/o_jrmuxout 32'h00400084 00000000 00400058 00400008 004000... 00400060 00400064 00400004 004000... 00400004
/tb/MyMips/id_ex/o_memre... 0
/tb/MyMips/id_ex/o_halt 0
/tb/MyMips/id_ex/tmp_we 1
/tb/MyMips/id_ex/tmp_halt 0
/tb/MyMips/id_ex/tmp_mem... 0
/tb/MyMips/id_ex/tmp_D 32'hXXXXXXXX 00000000 08100016 00000000 20110000 200800... 00082080 0C1000... 00000000 2008...
/tb/MyMips/id_ex/tmp_sign... 32'h0000XXXX 00000000 00000016 00000000 000000... 00002080 00000001 00000000 0000...
/tb/MyMips/id_ex/tmp_reg1... 32'h00000000 00000000
/tb/MyMips/id_ex/tmp_reg2... 32'h00000000 00000000 0000...
/tb/MyMips/id_ex/tmp_nextpc 32'h00400088 00000000 00400004 00400008 004000... 00400060 00400064 00400068 004000... 00400004 0040...
/tb/MyMips/id_ex/tmp_jrmu... 32'h00400088 00000000 00400058 00400008 004000... 00400060 00400064 00400004 004000... 00400004 0040...
/tb/MyMips/id_ex/tmp_rs 5'hXX 00
/tb/MyMips/id_ex/tmp_rt 5'hXX 00 10 00 11 08 10 00 08
/tb/MyMips/id_ex/tmp_rd 5'hXX 00 04 00
/tb/MyMips/id_ex/tmp_regw... 5'hXX 00 10 00 11 08 04 1F 00 08
/tb/MyMips/id_ex/tmp_WB 4'h0 2 0 2 6 2
/tb/MyMips/id_ex/tmp_EX 6'h00 2E 00 2E 10 2E 00 2E 10
/tb/MyMips/id_ex/tmp_MEM 3'h0 0
/tb/MyMips/ex_mem/i_alu 32'h00000000 00000000 0000000A 00000000
/tb/MyMips/ex_mem/i_CLK 1
/tb/MyMips/ex_mem/i_RST 0
/tb/MyMips/ex_mem/i_WB 4'h0 0 2 0 2 6 2
/tb/MyMips/ex_mem/i_M 3'h0 0
/tb/MyMips/ex_mem/i_halt 0
/tb/MyMips/ex_mem/i_ExM... 32'h00000000 00000000 0000000A 00000000
/tb/MyMips/ex_mem/i_rdrt... 5'hXX 00 10 00 11 08 04 1F 00
/tb/MyMips/ex_mem/i_New... 32'h00400084 00000000 00400004 00400008 004000... 00400060 00400064 00400068 004000... 00400004
/tb/MyMips/ex_mem/i_jrmu... 32'h00400084 00000000 00400058 00400008 004000... 00400060 00400064 00400004 004000... 00400004
/tb/MyMips/ex_mem/i_mem... 0
/tb/MyMips/ex_mem/i_Zero 0
/tb/MyMips/ex_mem/i_rd 5'hXX 00 04 00
/tb/MyMips/ex_mem/i_flush 1
/tb/MyMips/ex_mem/i_flush2 1
/tb/MyMips/ex_mem/i_stall 1
/tb/MyMips/ex_mem/o_WB 4'h0 0 2 0 2 6 2
/tb/MyMips/ex_mem/o_M 3'h0 0
/tb/MyMips/ex_mem/o_Zero 0
/tb/MyMips/ex_mem/o_alu 32'h00000000 00000000 0000000A 00000000
/tb/MyMips/ex_mem/o_Ex... 32'h00000000 00000000 0000000A 00000000
/tb/MyMips/ex_mem/o_rdrt... 5'hXX 00 10 00 11 08 04 1F 00
/tb/MyMips/ex_mem/o_Ne... 32'h00400080 00000000 00400004 00400008 004000... 00400060 00400064 00400068 004000... 0040...
/tb/MyMips/ex_mem/o_rd 5'hXX 00
/tb/MyMips/ex_mem/o_jrm... 32'h00400080 00000000 00400058 00400008 004000... 00400060 00400064 00400004 004000... 0040...
/tb/MyMips/ex_mem/o_me... 0
/tb/MyMips/ex_mem/o_halt 0



[2.b.i] list which instructions produce values, and what signals (i.e., bus names) in the pipeline these correspond to.

Instr	Signals
add, addu, sub, subu, sra, srl, sll, and, or, nor, xor, slt, addi, addiu, jr, andi, ori, xori, lui, slti	s_RegWrA_temp, s_writeAddrEX, s_RegWrAddr, s_EXmemReadWR, s_memWBReadWR
j, jr, jal	s_JRmuxO, s_IDEXJRmuxO, s_EXmemJRmuxO, s_JALmuxo, s_JRmuxo
bne, beq	s_exmembranchaddr

[2.b.ii] List which of these same instructions consume values, and what signals in the pipeline these correspond to.

Instr	Signals
add, addu, sub, subu, sra, srl, sll, and, or, nor, xor, slt, addi, addiu, jr, andi, ori, xori, lui, slti	i_alu_A, i_alu_B

[2.b.iii] generalized list of potential data dependencies. From this generalized list, select those dependencies that can be forwarded (write down the corresponding pipeline stages that will be forwarding and receiving the data), and those dependencies that will require hazard stalls.

forwarded	require hazard stalls
s_RegWrA_temp, s_writeAddrEX, s_RegWrAddr,	s_EXmemReadWR, s_memWBReadWR

[2.b.iv] global list of the datapath values and control signals that are required during each pipeline stage

IF	ID	EX	MEM	WB
s_Inst	s_IFIDinst	s_IDEXfunc	s_EXmemMux	s_WBdmem
iCLK	s_IFIDnewPC	s_IDEXwb	s_EXmemRegoMux	s_WBmemALUo
s_PCFour	s_aluControl	s_IDEXmem	s_EXmemNewPC	s_WBmemNewPC
s_IMemAddr	s_Branch	s_IDEXex	s_EXmemWB	s_RegWrAddr
s_NextInstAddr	s_beqS	s_IDEXreg1	s_EXmemMEM	s_WBmemOut
iRST	s_j	s_IDEXreg2	s_EXmemZero	s_Halt
iInstLd	s_jr	s_IDEXrs	s_EXmemALUo	s_RegWr
iInstAddr	s_jal	s_IDEXrt	s_EXmemHalt	s_tempram
iInstExt	s_memRead	s_IDEXrd	s_EXmemMux	s_RegWrData
s_IFIDflush	s_memToReg	s_IDEXnewPC	s_EXmemALUo	s_memWBjrmuxO
s_stallIF	s_DmemWR_temp	s_IDEXextend	iCLK	s_memWBReadWR
s_PCstallAddr	s_aluSrc	s_IDEXinst	iRST	s_memwbrd
	s_RegWr_temp	s_IDEXhalt	s_EXmemReadWR	
	s_signExtend	s_RegWrA_temp	s_EXmemrd	
	s_regDst	s_aluout	s_tempForwardRegO	
	s_Halt_temp	s_ZERO	s_memwbflush	
	s_sub	s_AS	s_memwbstall	
	s_ignrOvfl	s_mux2out	s_DMemWr	
	s_movn	s_IDEXJRmuxO	s_DMemData	
	s_regout1	s_IDEXReadWRmem	s_DMemAddr	
	s_regout2	s_IDEXhalt		
	s_extended	s_EXmemFlush		
	s_WRmuxO	s_flush2		
	iCLK	s_EXmemStall		
	iRST	s_IDEXReadWRmem		
	s_RegValue	s_forwardBMuxOut		
	s_BaddO	s_forwardB		
	s_BmuxO	s_forwardA		
	s_JshiftO	s_forwardAMuxOut		
	s_JmuxO	s_forwardBranch2		
	s_JRmuxO	s_forwardBranch		
	s_Carry			
	s_extended			
	s_WRmuxO			
	s_WRtemp			
	s_RegWrData			
	s_IDEXflush			
	s_IDEXstall			
	s_ReadWRmem			
	s_regequal			
	s_temp_regout2			
	s_temp_regout1			

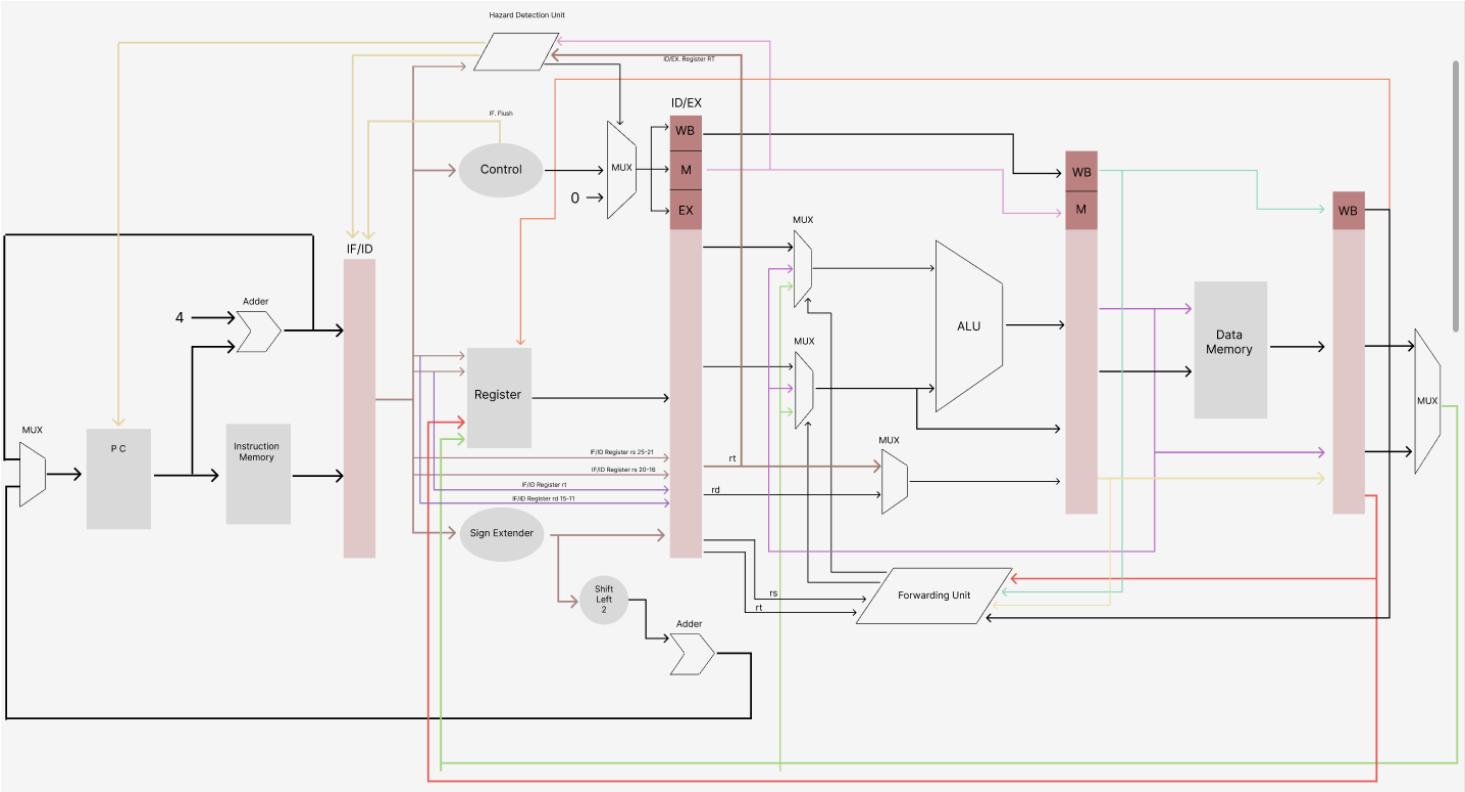
[2.c.i] list all instructions that may result in a non-sequential PC update and in which pipeline stage that update occurs.

instr	pipeline stage
bne, beq, j, jr, jal	wb

[2.c.ii] For these instructions, list which stages need to be stalled and which stages need to be squashed/flushed relative to the stage each of these instructions is in.

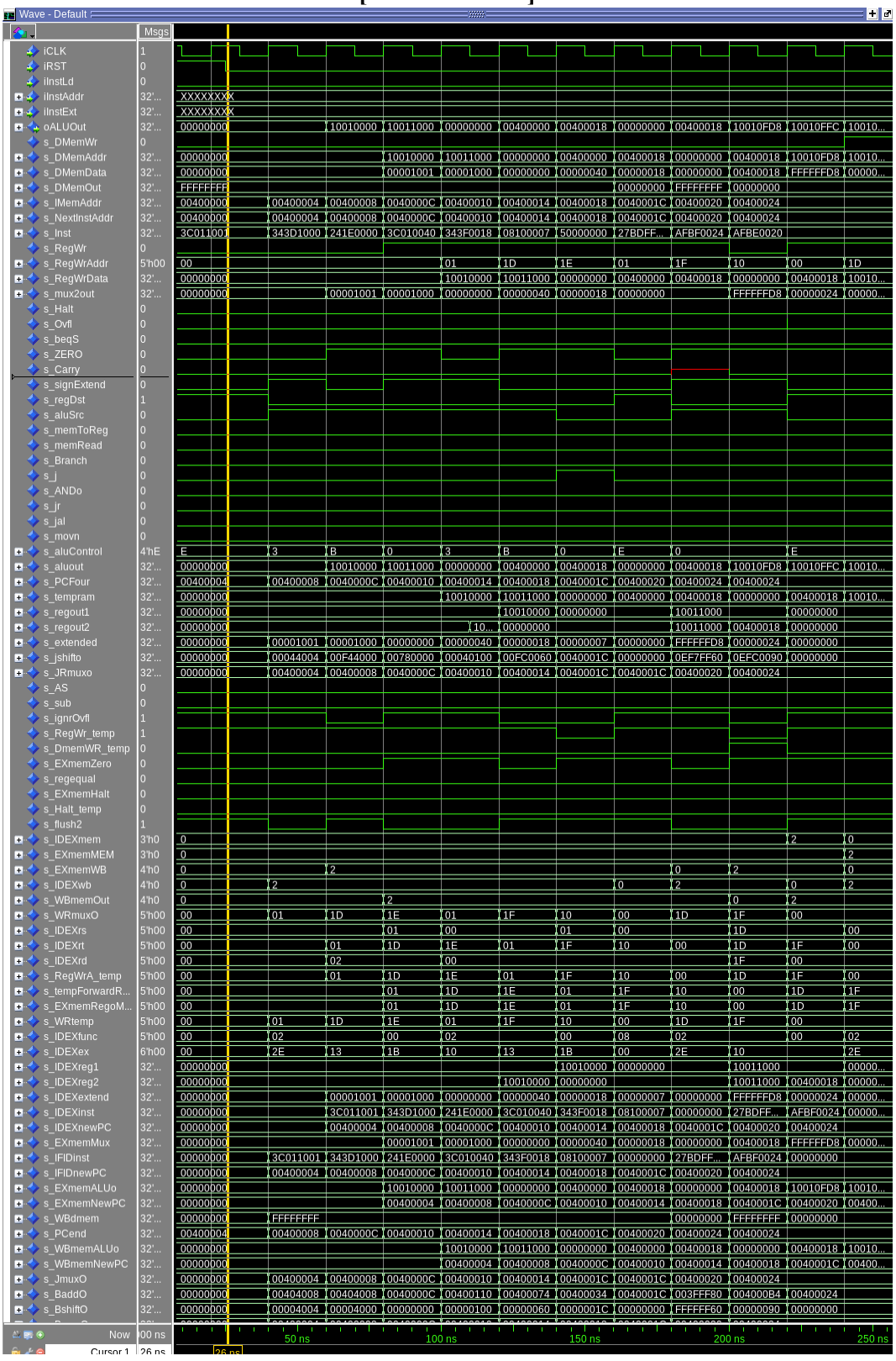
All the pipeline stages must stall while these instructions are in WB stage. The processor won't load an instruction if there are existing branching or jumping instructions in the pipeline because we positioned our Hazard Detection module inside the Fetch stage. Therefore, there is no need to flush stages, and the processor will stall itself while it waits.

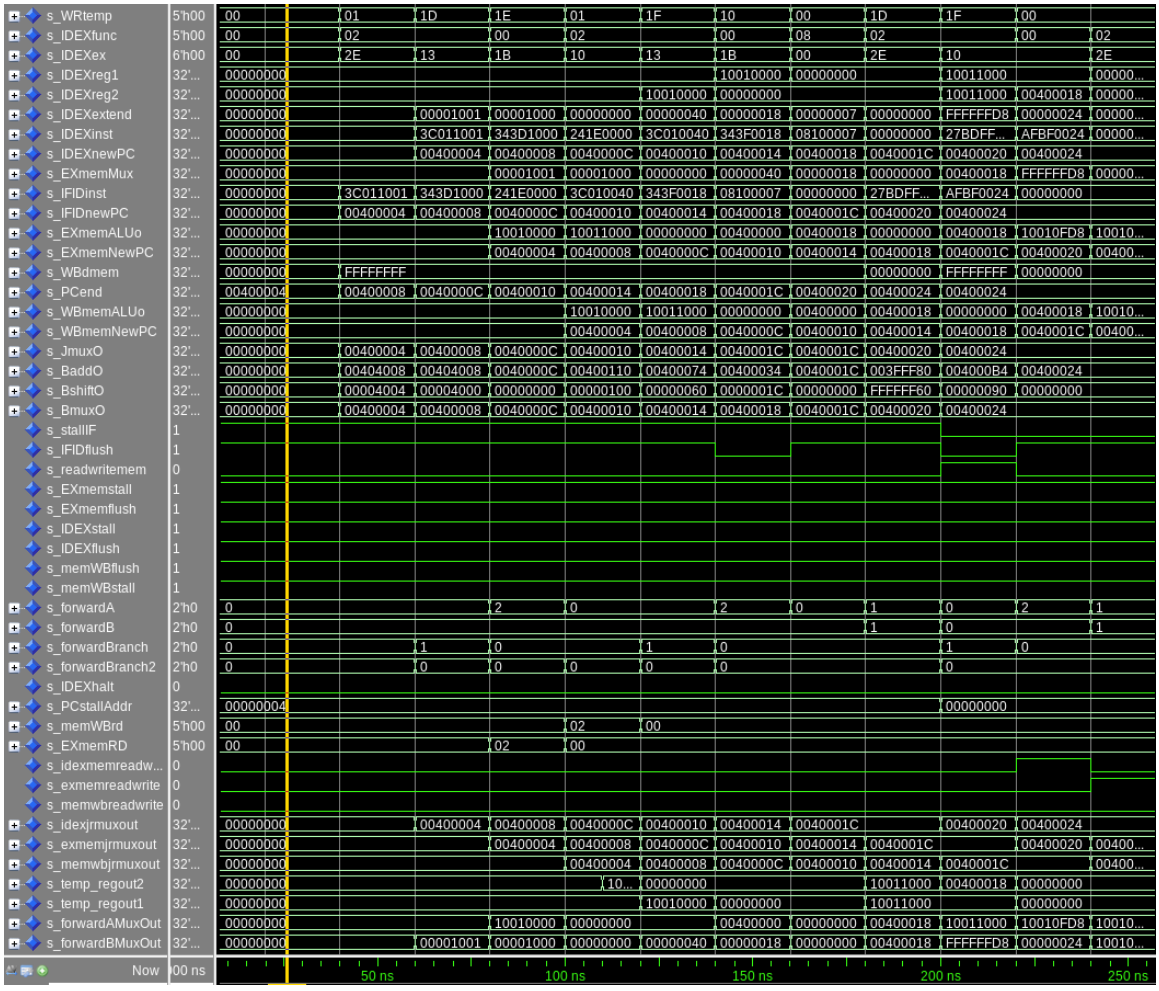
[2.d] implement the hardware-scheduled pipeline using only structural VHDL. As with the previous processors that you have implemented, start with a high-level schematic drawing of the interconnection between components.



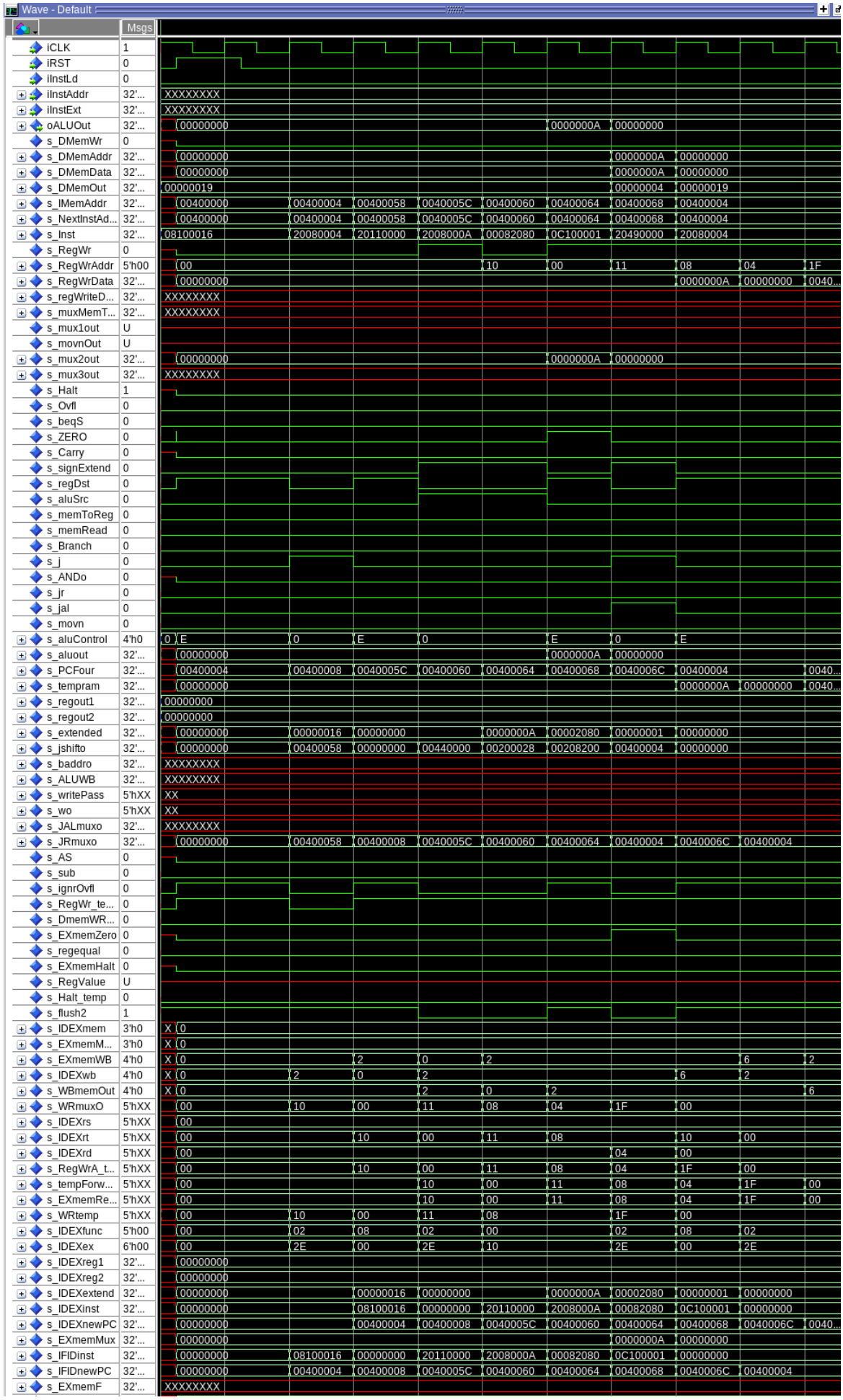
[2.e – i, ii, and iii] In your writeup, show the Modelsim output for each of the following tests, and provide a discussion of result correctness. It may be helpful to also annotate the waveforms directly.

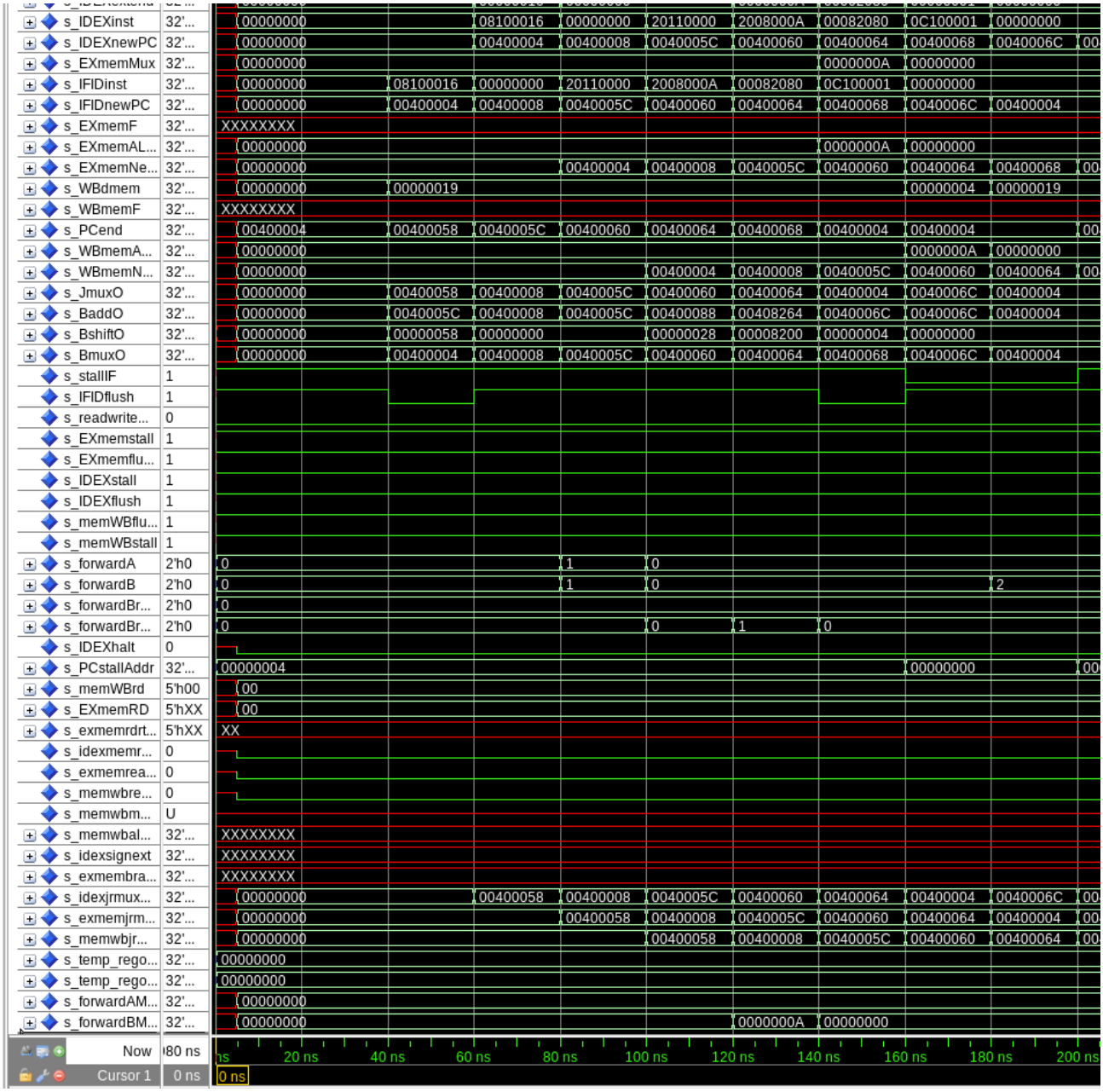
[Grendel Test]



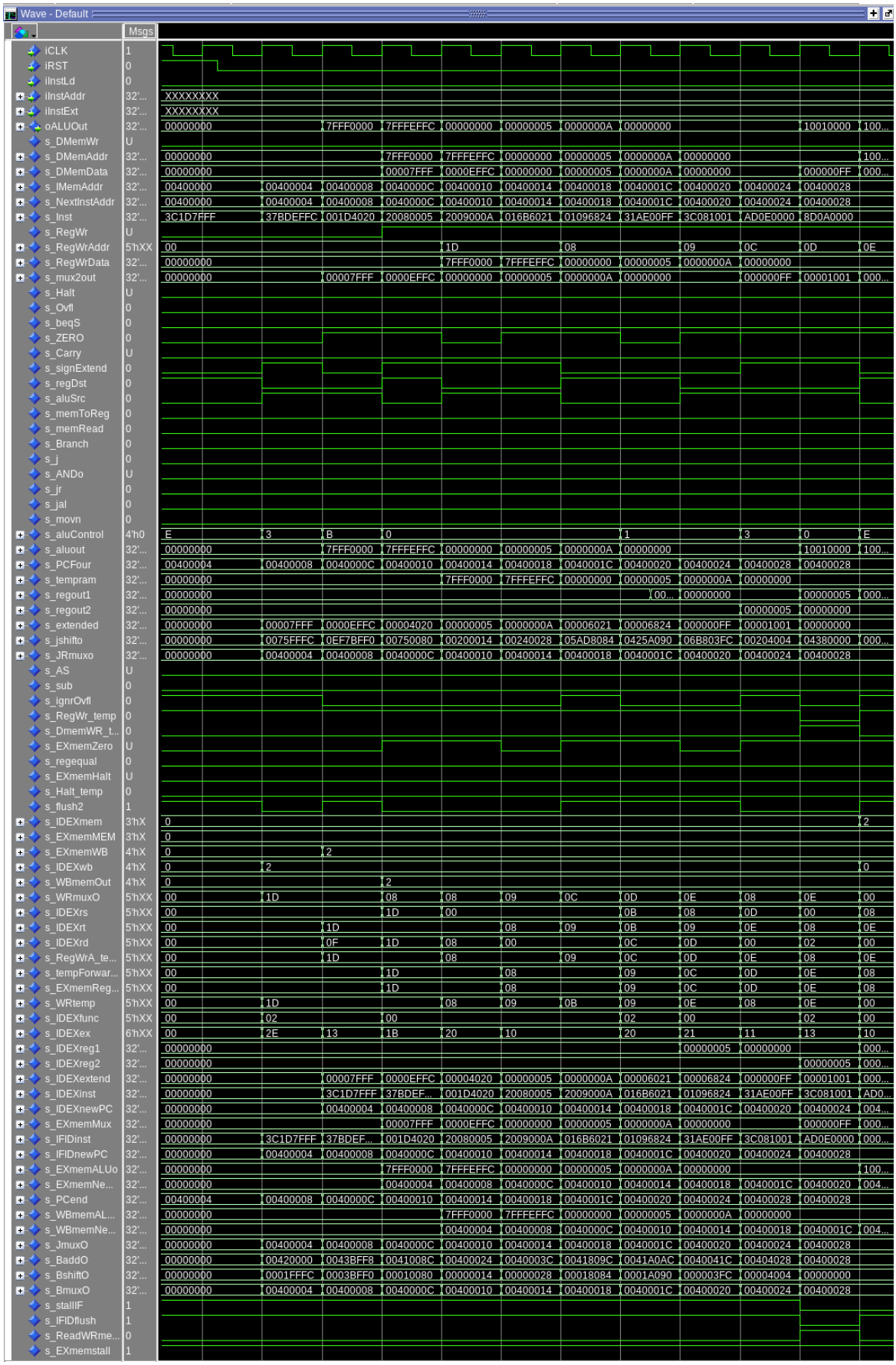

s_WRtemp 5'h00 00 01 1D 1E 01 1F 10 00 1D 1F 00
s_IDEXfunc 5'h00 00 02 00 02 00 08 02 00 02
s_IDEXex 6'h00 00 2E 13 1B 10 13 1B 00 2E 10 2E
s_IDEXreg1 32'... 00000000 10010000 00000000 10011000 00000...
s_IDEXreg2 32'... 00000000 10010000 00000000 10011000 00400018 00000...
s_IDEXextend 32'... 00000000 00001001 00001000 00000000 00000040 00000018 00000007 00000000 FFFFFFD8 00000024 00000...
s_IDEXinst 32'... 00000000 3C011001 343D1000 241E0000 3C010040 343F0018 08100007 00000000 27BDFF... AFBF0024 00000...
s_IDEXnewPC 32'... 00000000 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020 00400024
s_EXmemMux 32'... 00000000 00001001 00001000 00000000 00000040 00000018 00000000 00400018 FFFFFFD8 00000...
s_IFIDinst 32'... 00000000 3C011001 343D1000 241E0000 3C010040 343F0018 08100007 00000000 27BDFF... AFBF0024 00000000
s_IFIDnewPC 32'... 00000000 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020 00400024
s_EXmemALUo 32'... 00000000 10010000 10011000 00000000 00400000 00400018 00000000 00400018 10010FD8 10010...
s_EXmemNewPC 32'... 00000000 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020 00400...
s_WBdmem 32'... 00000000 FFFFFFFF 00000000 FFFFFFFF 00000000
s_PCend 32'... 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020 00400024 00400024
s_WBmemALUo 32'... 00000000 10010000 10011000 00000000 00400000 00400018 00000000 00400018 10010...
s_WBmemNewPC 32'... 00000000 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400...
s_JmuxO 32'... 00000000 00400004 00400008 0040000C 00400010 00400014 0040001C 0040001C 00400020 00400024
s_BaddO 32'... 00000000 00404008 00404008 0040000C 00400110 00400074 00400034 0040001C 003FFF80 004000B4 00400024
s_BshiftO 32'... 00000000 00004004 00004000 00000000 00000100 00000060 0000001C 00000000 FFFFFF60 00000090 00000000
s_BmuxO 32'... 00000000 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020 00400024
s_stallIF 1
s_IFIDflush 1
s_readwritemem 0
s_EXmemstall 1
s_EXmemflush 1
s_IDEXstall 1
s_IDEXflush 1
s_memWBflush 1
s_memWBstall 1
s_forwardA 2'h0 0 2 0 2 0 1 0 2 1
s_forwardB 2'h0 0 1 0
s_forwardBranch 2'h0 0 1 0 1 0 1 0
s_forwardBranch2 2'h0 0 0 0 0 0 0
s_IDEXhalt 0
s_PCstallAddr 32'... 00000004 00000000
s_memWBrd 5'h00 00 02 00
s_EXmemRD 5'h00 00 02 00
s_idexmemreadw... 0
s_exmemreadwrite 0
s_memwbreadwrite 0
s_idexjrmuxout 32'... 00000000 00400004 00400008 0040000C 00400010 00400014 0040001C 00400020 00400024
s_exmemjrmuxout 32'... 00000000 00400004 00400008 0040000C 00400010 00400014 0040001C 00400020 00400...
s_memwbjrmuxout 32'... 00000000 00400004 00400008 0040000C 00400010 00400014 0040001C 00400...
s_temp_regout2 32'... 00000000 10... 00000000 10011000 00400018 00000000
s_temp_regout1 32'... 00000000 10010000 00000000 10011000 00000000
s_forwardAMuxOut 32'... 00000000 10010000 00000000 00400000 00000000 00400018 10011000 10010FD8 10010...
s_forwardBMuxOut 32'... 00000000 00001001 00001000 00000000 00000040 00000018 00000000 00400018 FFFFFFD8 00000024 10010...
Now 00 ns
50 ns
100 ns
150 ns
200 ns
250 ns

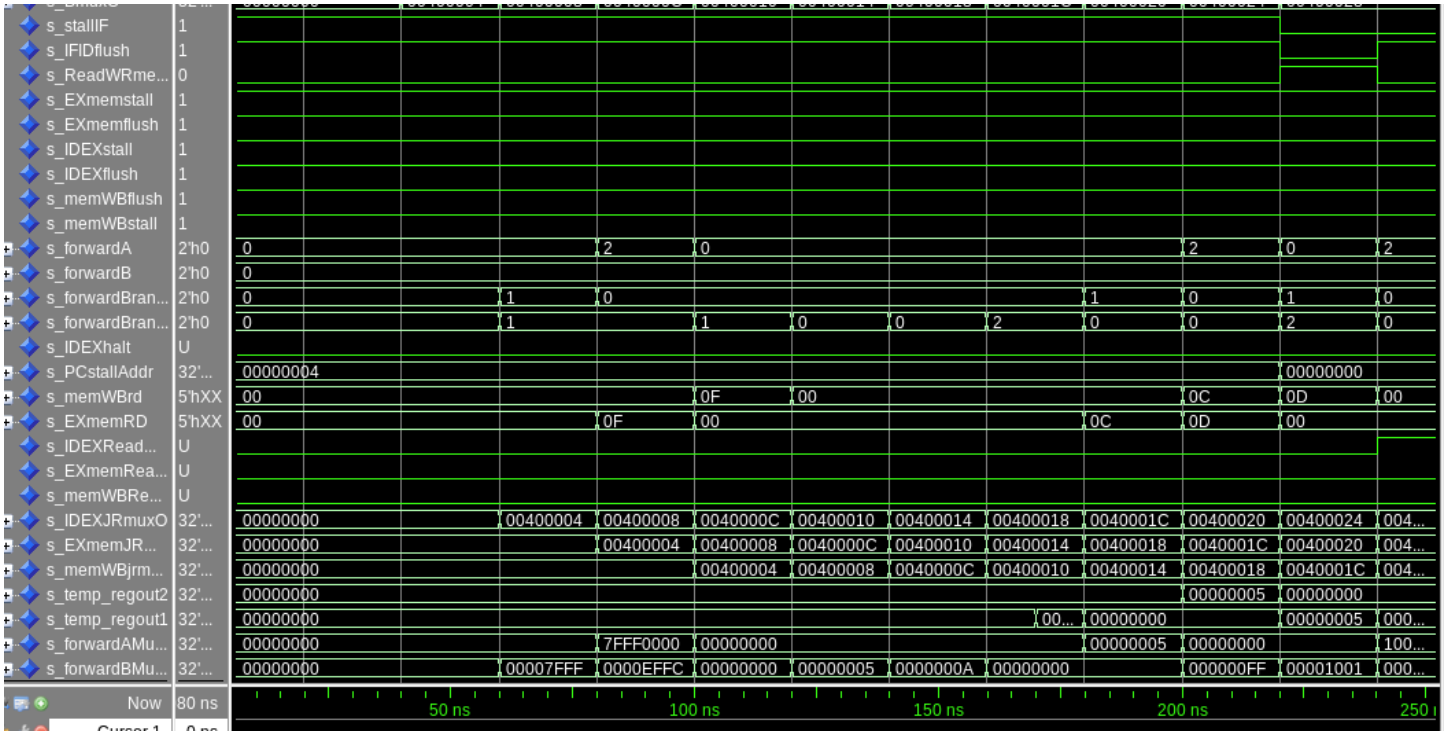
[Bubble Sort without Nops Test]




s_IDEXinst 32'... 00000000 08100016 00000000 20110000 2008000A 00082080 0C100001 00000000
s_IDEXnewPC 32'... 00000000 00400004 00400008 0040005C 00400060 00400064 00400068 0040006C
s_EXmemMux 32'... 00000000 0000000A 00000000
s_IFIDinst 32'... 00000000 08100016 00000000 20110000 2008000A 00082080 0C100001 00000000
s_IFIDnewPC 32'... 00000000 00400004 00400008 0040005C 00400060 00400064 00400068 0040006C 00400004
s_EXmemF 32'... XXXXXXXX
s_EXmemAL... 32'... 00000000 0000000A 00000000
s_EXmemNe... 32'... 00000000 00400004 00400008 0040005C 00400060 00400064 00400068
s_WBdmem 32'... 00000000 00000019 00000004 00000019
s_WBmemF 32'... XXXXXXXX
s_PCend 32'... 00400004 00400058 0040005C 00400060 00400064 00400068 00400004 00400004
s_WBmemA... 32'... 00000000 0000000A 00000000
s_WBmemN... 32'... 00000000 00400004 00400008 0040005C 00400060 00400064
s_JmuxO 32'... 00000000 00400058 00400008 0040005C 00400060 00400064 00400004 0040006C 00400004
s_BaddO 32'... 00000000 0040005C 00400008 0040005C 00400088 00408264 0040006C 0040006C 00400004
s_BshiftO 32'... 00000000 00000058 00000000 00000028 00008200 00000004 00000000
s_BmuxO 32'... 00000000 00400004 00400008 0040005C 00400060 00400064 00400068 0040006C 00400004
s_stallIF 1
s_IFIDflush 1
s_readwrite... 0
s_EXmemstall 1
s_EXmemflu... 1
s_IDEXstall 1
s_IDEXflush 1
s_memWBflu... 1
s_memWBstall 1
s_forwardA 2'h0 0 1 0
s_forwardB 2'h0 0 1 0 2
s_forwardBr... 2'h0 0
s_forwardBr... 2'h0 0 0 1 0
s_IDEXhalt 0
s_PCstallAddr 32'... 00000004 00000000
s_memWBrd 5'h00 00
s_EXmemRD 5'hXX 00
s_exmemrdrt... 5'hXX XX
s_idexmemr... 0
s_exmemrea... 0
s_memwbre... 0
s_memwbm... U
s_memwbal... 32'... XXXXXXXX
s_idexsignext 32'... XXXXXXXX
s_exmembra... 32'... XXXXXXXX
s_idexjrmux... 32'... 00000000 00400058 00400008 0040005C 00400060 00400064 00400004 0040006C
s_exmemjrm... 32'... 00000000 00400058 00400008 0040005C 00400060 00400064 00400004
s_memwbjr... 32'... 00000000 00400058 00400008 0040005C 00400060 00400064
s_temp_rego... 32'... 00000000
s_temp_rego... 32'... 00000000
s_forwardAM... 32'... 00000000
s_forwardBM... 32'... 00000000 0000000A 00000000
Now 180 ns
Cursor 1 0 ns
20 ns 40 ns 60 ns 80 ns 100 ns 120 ns 140 ns 160 ns 180 ns 200 ns
0 ns

[Base Test No Nops]



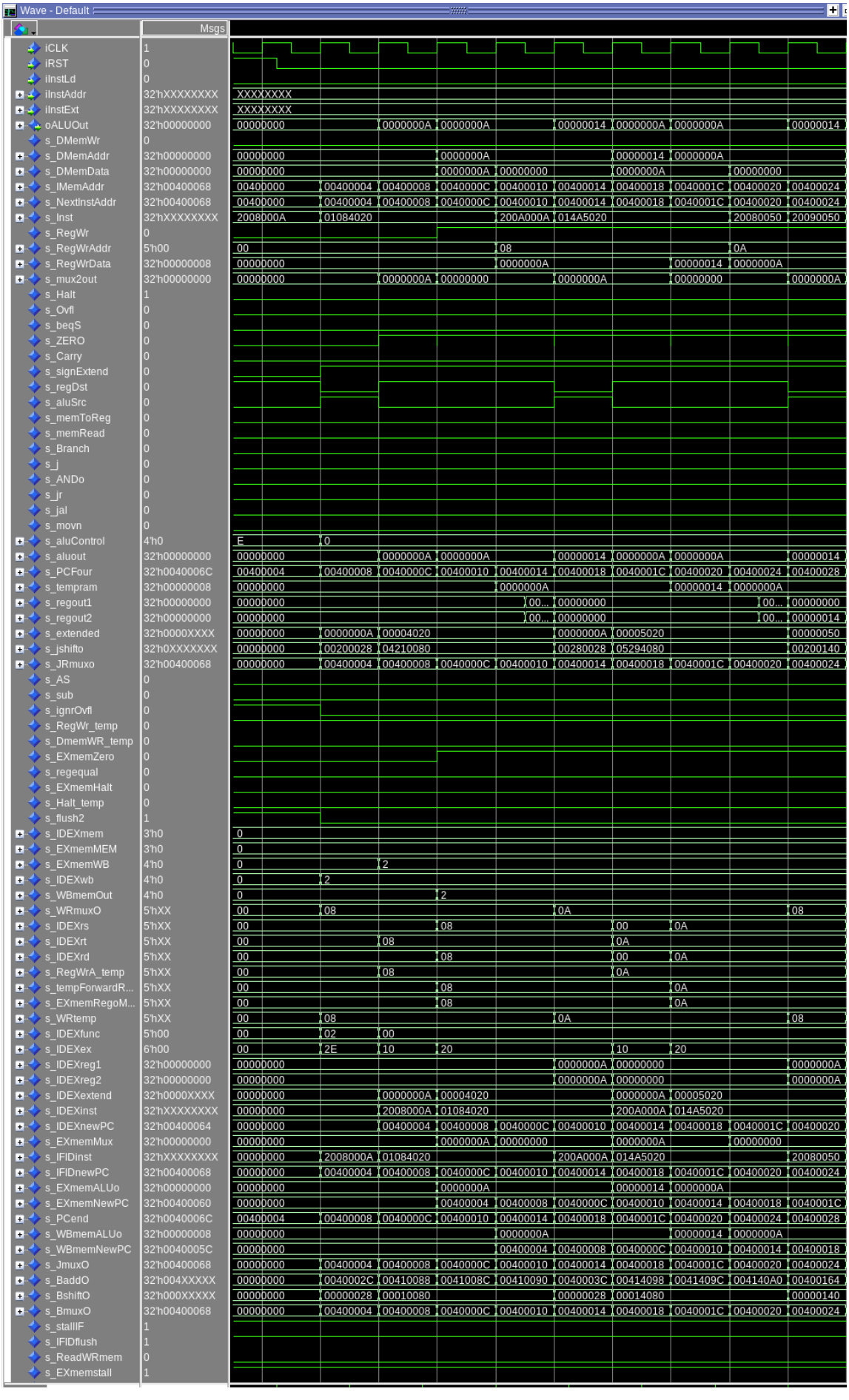


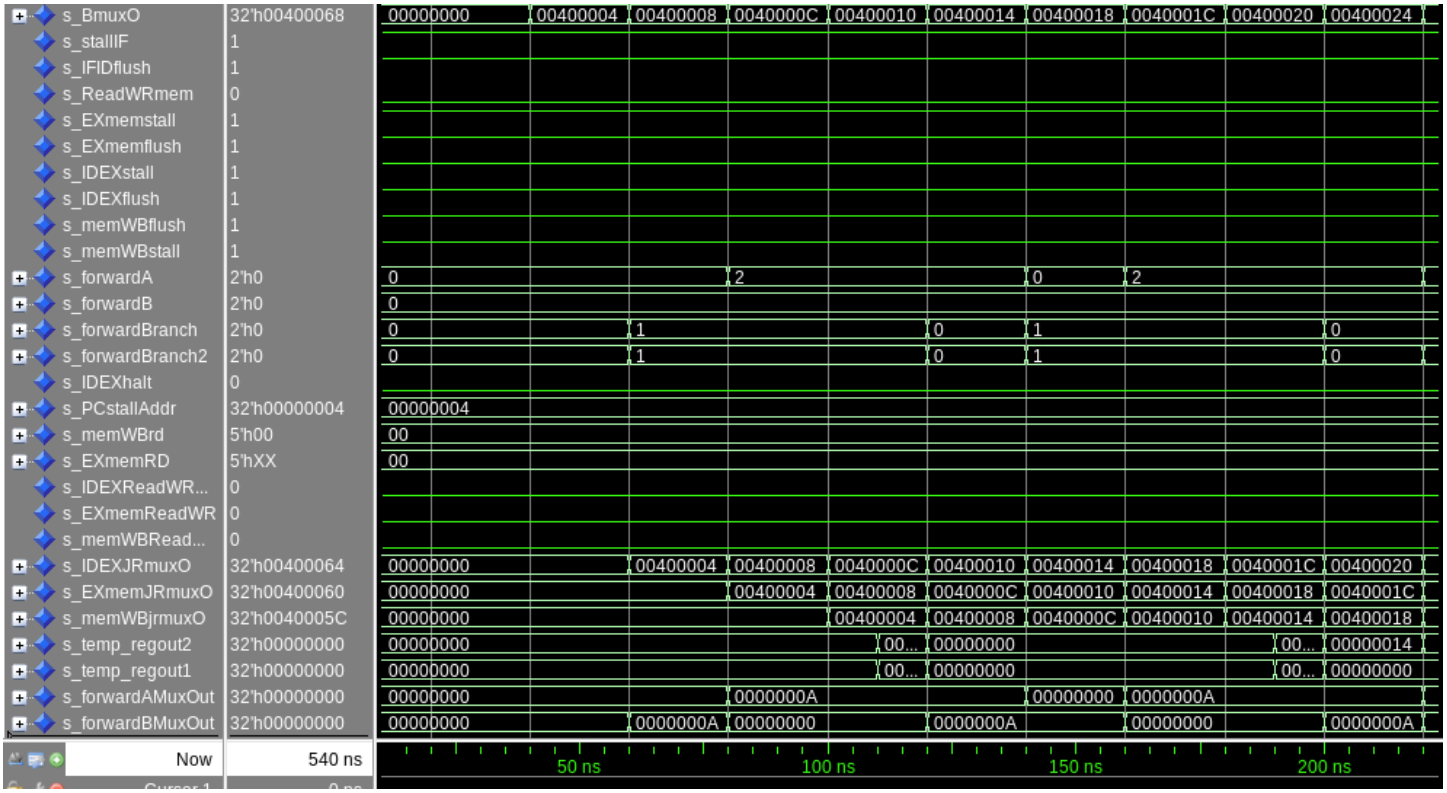
[2.e.i] Create a spreadsheet to track these cases and justify the coverage of your testing approach. Include this spreadsheet in your report as a table.

Stage	Description	Examples
IF -> ID	Instruction in IF depends on data ID stage	add $t0, $0, $sp addi $t0, $0, 5

IF -> EX	Instruction in IF depends on data EX stage	sw $t6, 0($t0)
		lw $t2, 0($t0)
		nor $t3, $t2, $t0

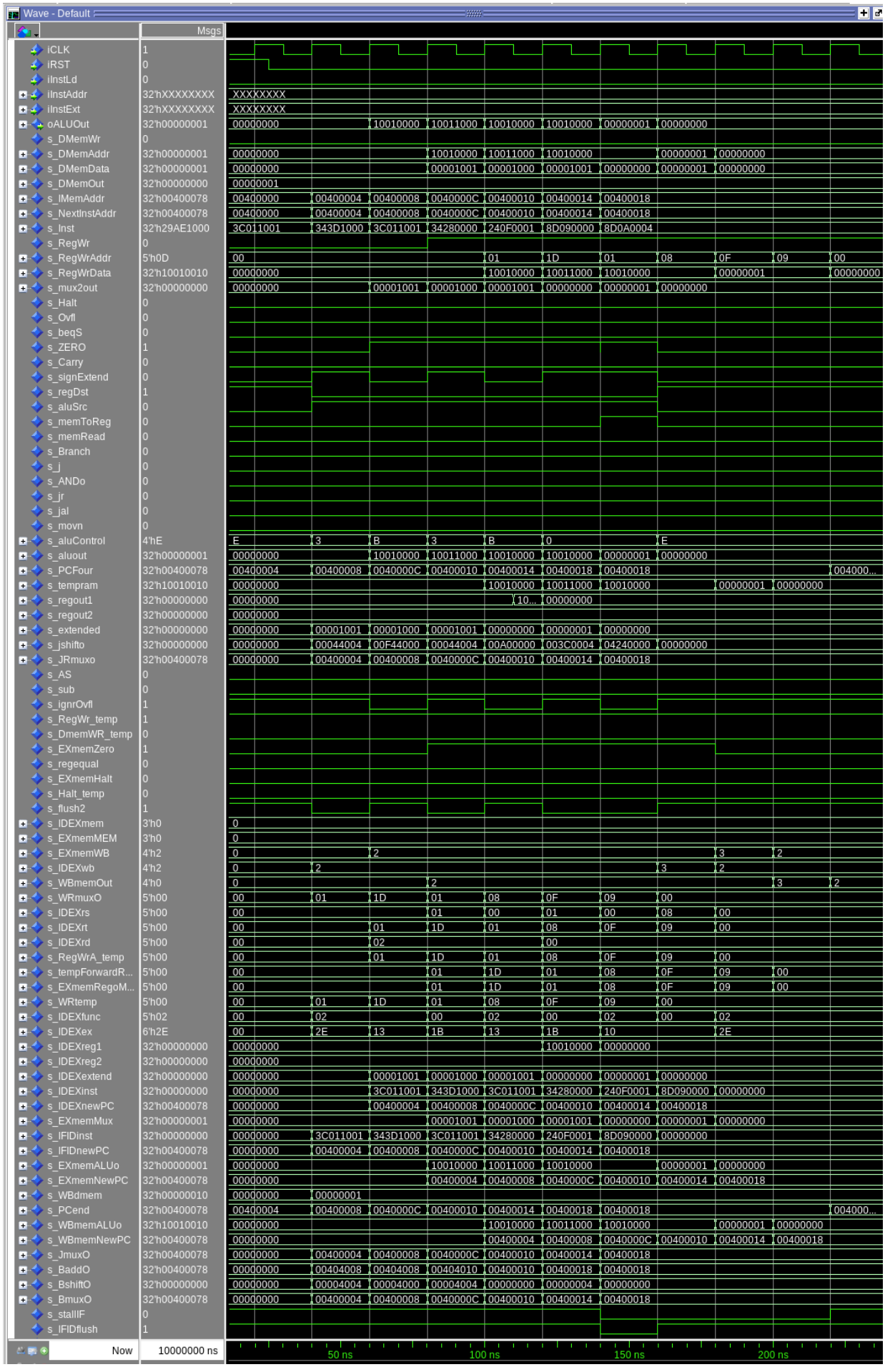
IF -> MEM	Instruction in IF depends on data MEM stage	xor $t4, $t2, $t3 xori $t5, $t4, 0x1100 or $t2, $t3, $0 ori $t4, $t2, 0x11

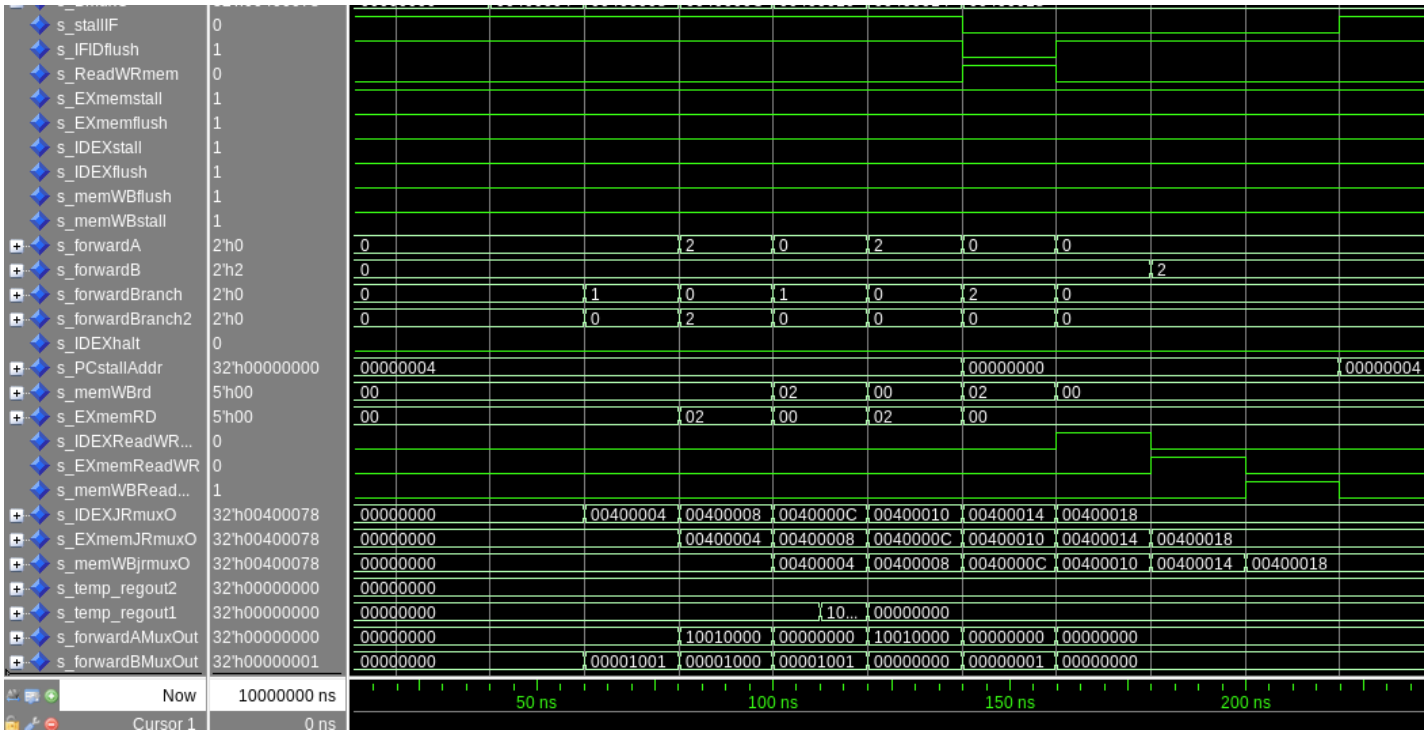

Wave - Default
Msgs
iCLK 1
iRST 0
iInstLd 0
iInstAddr 32'hXXXXXXXX XXXXXXXX
iInstExt 32'hXXXXXXXX XXXXXXXX
oALUOut 32'h00000000 00000000 0000000A 0000000A 00000014 0000000A 0000000A 00000014
s_DMemWr 0
s_DMemAddr 32'h00000000 00000000 0000000A 00000014 0000000A
s_DMemData 32'h00000000 00000000 0000000A 00000000 0000000A 00000000
s_IMemAddr 32'h00400068 00400000 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020 00400024
s_NextInstAddr 32'h00400068 00400000 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020 00400024
s_Inst 32'hXXXXXXXX 2008000A 01084020 200A000A 014A5020 20080050 20090050
s_RegWr 0
s_RegWrAddr 5'h00 00 08 0A
s_RegWrData 32'h00000008 00000000 0000000A 00000014 0000000A
s_mux2out 32'h00000000 00000000 0000000A 00000000 0000000A 00000000 0000000A
s_Halt 1
s_Ovfl 0
s_beqS 0
s_ZERO 0
s_Carry 0
s_signExtend 0
s_regDst 0
s_aluSrc 0
s_memToReg 0
s_memRead 0
s_Branch 0
s_j 0
s_ANDo 0
s_jr 0
s_jal 0
s_movn 0
s_aluControl 4'h0 E 0
s_aluout 32'h00000000 00000000 0000000A 0000000A 00000014 0000000A 0000000A 00000014
s_PCFour 32'h0040006C 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020 00400024 00400028
s_tempram 32'h00000008 00000000 0000000A 00000014 0000000A
s_regout1 32'h00000000 00000000 00... 00000000 00... 00000000
s_regout2 32'h00000000 00000000 00... 00000000 00... 00000014
s_extended 32'h0000XXXX 00000000 0000000A 00004020 0000000A 00005020 00000050
s_jshifto 32'h0XXXXXXX 00000000 00200028 04210080 00280028 05294080 00200140
s_JRmuxo 32'h00400068 00000000 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020 00400024
s_AS 0
s_sub 0
s_ignrOvfl 0
s_RegWr_temp 0
s_DmemWR_temp 0
s_EXmemZero 0
s_regequal 0
s_EXmemHalt 0
s_Halt_temp 0
s_flush2 1
s_IDEXmem 3'h0 0
s_EXmemMEM 3'h0 0
s_EXmemWB 4'h0 0 2
s_IDEXwb 4'h0 0 2
s_WBmemOut 4'h0 0 2
s_WRmuxO 5'hXX 00 08 0A 08
s_IDEXrs 5'hXX 00 08 00 0A
s_IDEXrt 5'hXX 00 08 0A
s_IDEXrd 5'hXX 00 08 00 0A
s_RegWrA_temp 5'hXX 00 08 0A
s_tempForwardR... 5'hXX 00 08 0A
s_EXmemRegoM... 5'hXX 00 08 0A
s_WRtemp 5'hXX 00 08 0A 08
s_IDEXfunc 5'h00 00 02 00
s_IDEXex 6'h00 00 2E 10 20 10 20
s_IDEXreg1 32'h00000000 00000000 0000000A 00000000 0000000A
s_IDEXreg2 32'h00000000 00000000 0000000A 00000000 0000000A
s_IDEXextend 32'h0000XXXX 00000000 0000000A 00004020 0000000A 00005020
s_IDEXinst 32'hXXXXXXXX 00000000 2008000A 01084020 200A000A 014A5020
s_IDEXnewPC 32'h00400064 00000000 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020
s_EXmemMux 32'h00000000 00000000 0000000A 00000000 0000000A 00000000
s_IFIDinst 32'hXXXXXXXX 00000000 2008000A 01084020 200A000A 014A5020 20080050
s_IFIDnewPC 32'h00400068 00000000 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020 00400024
s_EXmemALUo 32'h00000000 00000000 0000000A 00000014 0000000A
s_EXmemNewPC 32'h00400060 00000000 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C
s_PCend 32'h0040006C 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020 00400024 00400028
s_WBmemALUo 32'h00000008 00000000 0000000A 00000014 0000000A
s_WBmemNewPC 32'h0040005C 00000000 00400004 00400008 0040000C 00400010 00400014 00400018
s_JmuxO 32'h00400068 00000000 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020 00400024
s_BaddO 32'h004XXXXX 00000000 0040002C 00410088 0041008C 00410090 0040003C 00414098 0041409C 004140A0 00400164
s_BshiftO 32'h000XXXXX 00000000 00000028 00010080 00000028 00014080 00000140
s_BmuxO 32'h00400068 00000000 00400004 00400008 0040000C 00400010 00400014 00400018 0040001C 00400020 00400024
s_stallIF 1
s_IFIDflush 1
s_ReadWRmem 0
s_EXmemstall 1



[2.e.ii] Create a spreadsheet to track these cases and justify the coverage of your testing approach. Include this spreadsheet in your report as a table.

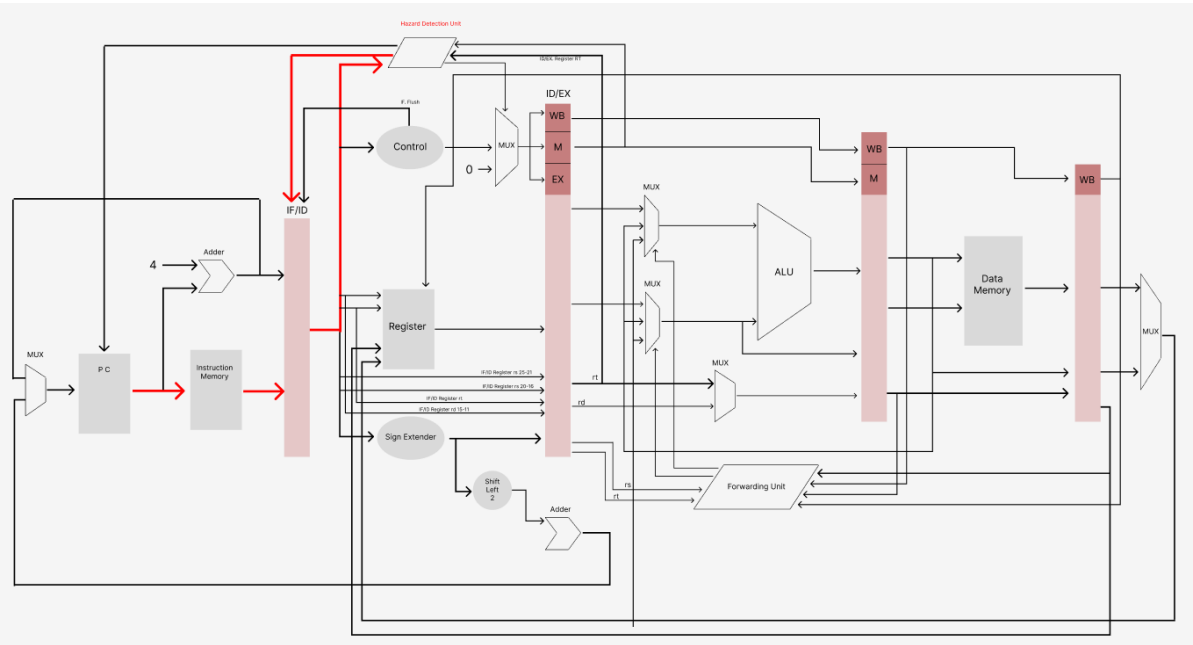
Instruction	Description	Example
Jump	Completes alu instructions and jumps to a new set of instructions that are sw in after, variable change	addi $t1, $t1, 0x1 ori $t7, $t7, 0x1 j after
Branch	branches after lw, conditional change	lw $t5, 0x10($t0) beq $t7, 0x1, odd


Wave - Default
Msgs
iCLK 1
iRST 0
iInstLd 0
iInstAddr 32'hXXXXXXXX XXXXXXXX
iInstExt 32'hXXXXXXXX XXXXXXXX
oALUOut 32'h00000001 00000000 10010000 10011000 10010000 10010000 00000001 00000000
s_DMemWr 0
s_DMemAddr 32'h00000001 00000000 10010000 10011000 10010000 00000001 00000000
s_DMemData 32'h00000001 00000000 00001001 00001000 00001001 00000000 00000001 00000000
s_DMemOut 32'h00000000 00000001
s_IMemAddr 32'h00400078 00400000 00400004 00400008 0040000C 00400010 00400014 00400018
s_NextInstAddr 32'h00400078 00400000 00400004 00400008 0040000C 00400010 00400014 00400018
s_Inst 32'h29AE1000 3C011001 343D1000 3C011001 34280000 240F0001 8D090000 8D0A0004
s_RegWr 0
s_RegWrAddr 5'h0D 00 01 1D 01 08 0F 09 00
s_RegWrData 32'h10010010 00000000 10010000 10011000 10010000 00000001 00000000
s_mux2out 32'h00000000 00000000 00001001 00001000 00001001 00000000 00000001 00000000
s_Halt 0
s_Ovfl 0
s_beqS 0
s_ZERO 1
s_Carry 0
s_signExtend 0
s_regDst 1
s_aluSrc 0
s_memToReg 0
s_memRead 0
s_Branch 0
s_j 0
s_ANDo 0
s_jr 0
s_jal 0
s_movn 0
s_aluControl 4'hE E 3 B 3 B 0 E
s_aluout 32'h00000001 00000000 10010000 10011000 10010000 10010000 00000001 00000000
s_PCFour 32'h00400078 00400004 00400008 0040000C 00400010 00400014 00400018 004000...
s_tempram 32'h10010010 00000000 10010000 10011000 10010000 00000001 00000000
s_regout1 32'h00000000 00000000 10... 00000000
s_regout2 32'h00000000 00000000
s_extended 32'h00000000 00000000 00001001 00001000 00001001 00000000 00000001 00000000
s_jshifto 32'h00000000 00000000 00044004 00F44000 00044004 00A00000 003C0004 04240000 00000000
s_JRmuxo 32'h00400078 00000000 00400004 00400008 0040000C 00400010 00400014 00400018
s_AS 0
s_sub 0
s_ignrOvfl 1
s_RegWr_temp 1
s_DmemWR_temp 0
s_EXmemZero 1
s_regequal 0
s_EXmemHalt 0
s_Halt_temp 0
s_flush2 1
s_IDEXmem 3'h0 0
s_EXmemMEM 3'h0 0
s_EXmemWB 4'h2 0 2 3 2
s_IDEXwb 4'h2 0 2 3 2
s_WBmemOut 4'h0 0 2 3 2
s_WRmuxO 5'h00 00 01 1D 01 08 0F 09 00
s_IDEXrs 5'h00 00 01 00 01 00 08 00
s_IDEXrt 5'h00 00 01 1D 01 08 0F 09 00
s_IDEXrd 5'h00 00 02 00
s_RegWrA_temp 5'h00 00 01 1D 01 08 0F 09 00
s_tempForwardR... 5'h00 00 01 1D 01 08 0F 09 00
s_EXmemRegoM... 5'h00 00 01 1D 01 08 0F 09 00
s_WRtemp 5'h00 00 01 1D 01 08 0F 09 00
s_IDEXfunc 5'h02 00 02 00 02 00 02 00 02
s_IDEXex 6'h2E 00 2E 13 1B 13 1B 10 2E
s_IDEXreg1 32'h00000000 00000000 10010000 00000000
s_IDEXreg2 32'h00000000 00000000
s_IDEXextend 32'h00000000 00000000 00001001 00001000 00001001 00000000 00000001 00000000
s_IDEXinst 32'h00000000 00000000 3C011001 343D1000 3C011001 34280000 240F0001 8D090000 00000000
s_IDEXnewPC 32'h00400078 00000000 00400004 00400008 0040000C 00400010 00400014 00400018
s_EXmemMux 32'h00000001 00000000 00001001 00001000 00001001 00000000 00000001 00000000
s_IFIDinst 32'h00000000 00000000 3C011001 343D1000 3C011001 34280000 240F0001 8D090000 00000000
s_IFIDnewPC 32'h00400078 00000000 00400004 00400008 0040000C 00400010 00400014 00400018
s_EXmemALUo 32'h00000001 00000000 10010000 10011000 10010000 00000001 00000000
s_EXmemNewPC 32'h00400078 00000000 00400004 00400008 0040000C 00400010 00400014 00400018
s_WBdmem 32'h00000010 00000000 00000001
s_PCend 32'h00400078 00400004 00400008 0040000C 00400010 00400014 00400018 00400018 004000...
s_WBmemALUo 32'h10010010 00000000 10010000 10011000 10010000 00000001 00000000
s_WBmemNewPC 32'h00400078 00000000 00400004 00400008 0040000C 00400010 00400014 00400018
s_JmuxO 32'h00400078 00000000 00400004 00400008 0040000C 00400010 00400014 00400018
s_BaddO 32'h00400078 00000000 00404008 00404008 00404010 00400010 00400018 00400018
s_BshiftO 32'h00000000 00000000 00004004 00004000 00004004 00000000 00000004 00000000
s_BmuxO 32'h00400078 00000000 00400004 00400008 0040000C 00400010 00400014 00400018
s_stallIF 0
s_IFIDflush 1
Now 10000000 ns
50 ns
100 ns
150 ns
200 ns



[2.f] report the maximum frequency your hardware-scheduled pipelined processor can run at and determine what your critical path is (specify each module/entity/component that this path goes through).

Our hardware-scheduled pipelined processor can run at a maximum frequency of 43.09 MHz in 23.2 nanoseconds. The critical path starts at the id/ex pipeline register and goes to the alu then ends at the pc reg. Red lines in the diagram below denote critical routes.



[Synthesis_hw: timing.txt]

Open | *timingHW(1).txt ~/Downloads | Save

```
# CprE 381 toolflow Timing dump

FMax: 43.09mhz Clk Constraint: 20.00ns Slack: -3.21ns

The path is given below

 ====================================================================
 From Node    : IDEX:id_ex|dffg_N:RegInst|s_Q[30]
 To Node      : PC:pcReg|s_OUT[2]
 Launch Clock : iCLK
 Latch Clock  : iCLK
 Data Arrival Path:
 Total (ns)  Incr (ns)     Type  Element
 ==========  ========= ==  ====  ===================================
      0.000     0.000           launch edge time
      3.026     3.026  R        clock network delay
      3.258     0.232     uTco  IDEX:id_ex|dffg_N:RegInst|s_Q[30]
      3.258     0.000 FF  CELL  id_ex|RegInst|s_Q[30]|q
      3.789     0.531 FF    IC  Mux0~0|dataa
      4.195     0.406 FR  CELL  Mux0~0|combout
      4.399     0.204 RR    IC  Mux0~2|datad
      4.538     0.139 RF  CELL  Mux0~2|combout
      5.024     0.486 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:1:MUXI|g_or|o_F~0|datad
      5.174     0.150 FR  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:1:MUXI|g_or|o_F~0|combout
      5.378     0.204 RR    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:1:MUXI|g_or|o_F~1|datad
      5.517     0.139 RF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:1:MUXI|g_or|o_F~1|combout
      5.754     0.237 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:2:MUXI|g_or|o_F~0|datad
      5.879     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:2:MUXI|g_or|o_F~0|combout
      6.128     0.249 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:3:MUXI|g_or|o_F~0|datad
      6.253     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:3:MUXI|g_or|o_F~0|combout
      6.879     0.626 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:4:MUXI|g_or|o_F~0|datad
      7.004     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:4:MUXI|g_or|o_F~0|combout
      7.256     0.252 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:5:MUXI|g_or|o_F~0|datad
      7.381     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:5:MUXI|g_or|o_F~0|combout
      7.637     0.256 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:6:MUXI|g_or|o_F~0|datac
      7.918     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:6:MUXI|g_or|o_F~0|combout
      8.168     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:7:MUXI|g_or|o_F~0|datad
      8.293     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:7:MUXI|g_or|o_F~0|combout
      8.544     0.251 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:8:MUXI|g_or|o_F~0|datad
      8.669     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:8:MUXI|g_or|o_F~0|combout
      8.919     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:9:MUXI|g_or|o_F~0|datad
      9.044     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:9:MUXI|g_or|o_F~0|combout
      9.293     0.249 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:10:MUXI|g_or|o_F~0|datad
      9.418     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:10:MUXI|g_or|o_F~0|combout
      9.672     0.254 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:11:MUXI|g_or|o_F~0|datac
      9.953     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:11:MUXI|g_or|o_F~0|combout
     10.204     0.251 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:12:MUXI|g_or|o_F~0|datad
     10.329     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:12:MUXI|g_or|o_F~0|combout
     10.579     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:13:MUXI|g_or|o_F~0|datad
     10.704     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:13:MUXI|g_or|o_F~0|combout
     10.961     0.257 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:14:MUXI|g_or|o_F~0|datac
     11.242     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:14:MUXI|g_or|o_F~0|combout
     11.490     0.248 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:15:MUXI|g_or|o_F~0|datad
     11.615     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:15:MUXI|g_or|o_F~0|combout
     12.030     0.415 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:16:MUXI|g_or|o_F~0|datad
     12.155     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:16:MUXI|g_or|o_F~0|combout
     12.407     0.252 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:17:MUXI|g_or|o_F~0|datad
     12.532     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:17:MUXI|g_or|o_F~0|combout
     12.787     0.255 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:18:MUXI|g_or|o_F~0|datac
     13.068     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:18:MUXI|g_or|o_F~0|combout
     13.318     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:19:MUXI|g_or|o_F~0|datad
     13.443     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:19:MUXI|g_or|o_F~0|combout
     13.699     0.256 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:20:MUXI|g_or|o_F~0|datac
     13.980     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:20:MUXI|g_or|o_F~0|combout
     14.235     0.255 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:21:MUXI|g_or|o_F~0|datac
     14.516     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:21:MUXI|g_or|o_F~0|combout
     14.768     0.252 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:22:MUXI|g_or|o_F~0|datad
     14.893     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:22:MUXI|g_or|o_F~0|combout
     15.142     0.249 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:23:MUXI|g_or|o_F~0|datad
     15.267     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:23:MUXI|g_or|o_F~0|combout
     15.518     0.251 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:24:MUXI|g_or|o_F~0|datad
     15.643     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:24:MUXI|g_or|o_F~0|combout
     15.901     0.258 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:25:MUXI|g_or|o_F~0|datac
     16.182     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:25:MUXI|g_or|o_F~0|combout
     16.437     0.255 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:26:MUXI|g_or|o_F~0|datac
     16.718     0.281 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:26:MUXI|g_or|o_F~0|combout
     16.968     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:27:MUXI|g_or|o_F~0|datad
     17.093     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:27:MUXI|g_or|o_F~0|combout
     17.342     0.249 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:28:MUXI|g_or|o_F~0|datad
     17.467     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:28:MUXI|g_or|o_F~0|combout
     17.717     0.250 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:29:MUXI|g_or|o_F~0|datad
     17.842     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:29:MUXI|g_or|o_F~0|combout
     18.093     0.251 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:30:MUXI|g_or|o_F~0|datad
     18.218     0.125 FF  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:30:MUXI|g_or|o_F~0|combout
     18.445     0.227 FF    IC  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:31:MUXI|g_xor2|o_F|datad
     18.595     0.150 FR  CELL  mainALU|g_addSub|g_fulladder|\G_NBit_MUX:31:MUXI|g_xor2|o_F|combout
     19.010     0.415 RR    IC  mainALU|g_mux9|Mux0~11|datac
     19.297     0.287 RR  CELL  mainALU|g_mux9|Mux0~11|combout
     19.502     0.205 RR    IC  mainALU|g_mux9|Mux0~12|datad
     19.657     0.155 RR  CELL  mainALU|g_mux9|Mux0~12|combout
     19.860     0.203 RR    IC  mainALU|g_mux9|Mux0~13|datad
     20.015     0.155 RR  CELL  mainALU|g_mux9|Mux0~13|combout
     20.912     0.897 RR    IC  s_temp_regout2[31]~52|datad
     21.067     0.155 RR  CELL  s_temp_regout2[31]~52|combout
     22.319     1.252 RR    IC  s_regequal~1|datab
     22.705     0.386 RF  CELL  s_regequal~1|combout
     22.933     0.228 FF    IC  s_regequal~3|datad
     23.058     0.125 FF  CELL  s_regequal~3|combout
     23.285     0.227 FF    IC  s_regequal~4|datad
     23.410     0.125 FF  CELL  s_regequal~4|combout
     23.638     0.228 FF    IC  s_regequal~2|datad
     ...
    23.788      0.150 FR  CELL  s_regequal~2|combout
    24.486      0.698 RR    IC  comb~2|datad
    24.625      0.139 RF  CELL  comb~2|combout
    25.565      0.940 FF    IC  prePCmux|\G_NBit_MUX:2:MUXI|g_org1|o_F~0|datad
    25.715      0.150 FR  CELL  prePCmux|\G_NBit_MUX:2:MUXI|g_org1|o_F~0|combout
    25.919      0.204 RR    IC  prePCmux|\G_NBit_MUX:2:MUXI|g_org1|o_F~2|datad
    26.058      0.139 RF  CELL  prePCmux|\G_NBit_MUX:2:MUXI|g_org1|o_F~2|combout
    26.058      0.000 FF    IC  pcReg|s_OUT[2]|d
    26.162      0.104 FF  CELL  PC:pcReg|s_OUT[2]
Data Required Path:
Total (ns)  Incr (ns)     Type  Element
==========  ========= ==  ====  ===================================
    20.000    20.000           latch edge time
    22.924     2.924  R        clock network delay
    22.956     0.032           clock pessimism removed
    22.936    -0.020           clock uncertainty
    22.954     0.018     uTsu  PC:pcReg|s_OUT[2]
Data Arrival Time  :    26.162
Data Required Time :    22.954
Slack              :    -3.208 (VIOLATED)
====================================================================
```