CprE 381: Computer Organization and Assembly-Level Programming

Project Part 2 Report

Team Members: Sam Burrell

Justin Jaeckel

____

____

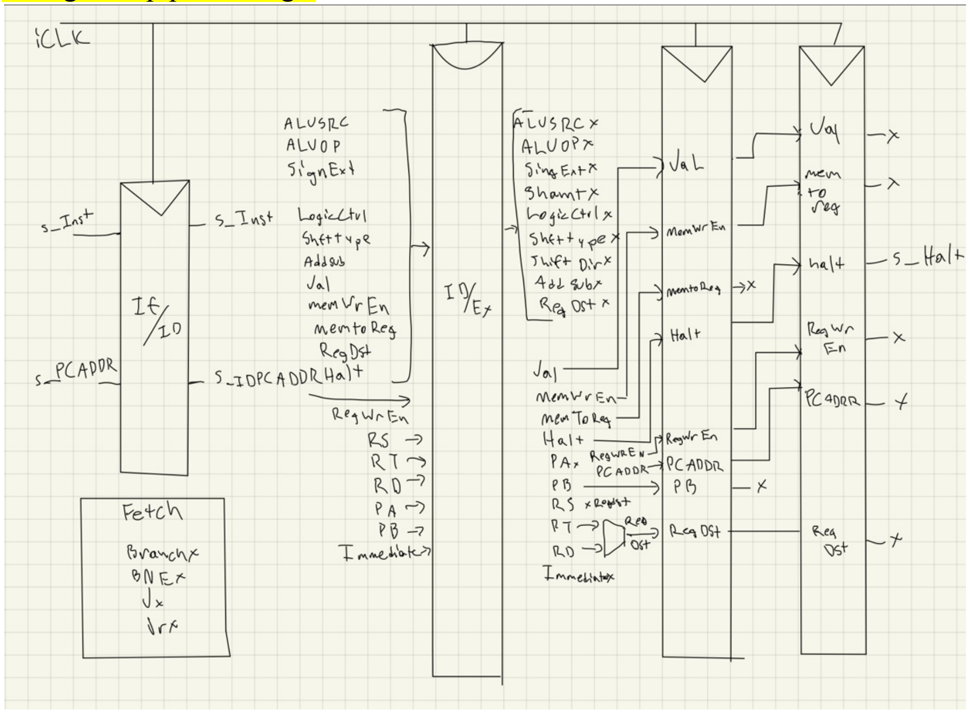
____

____

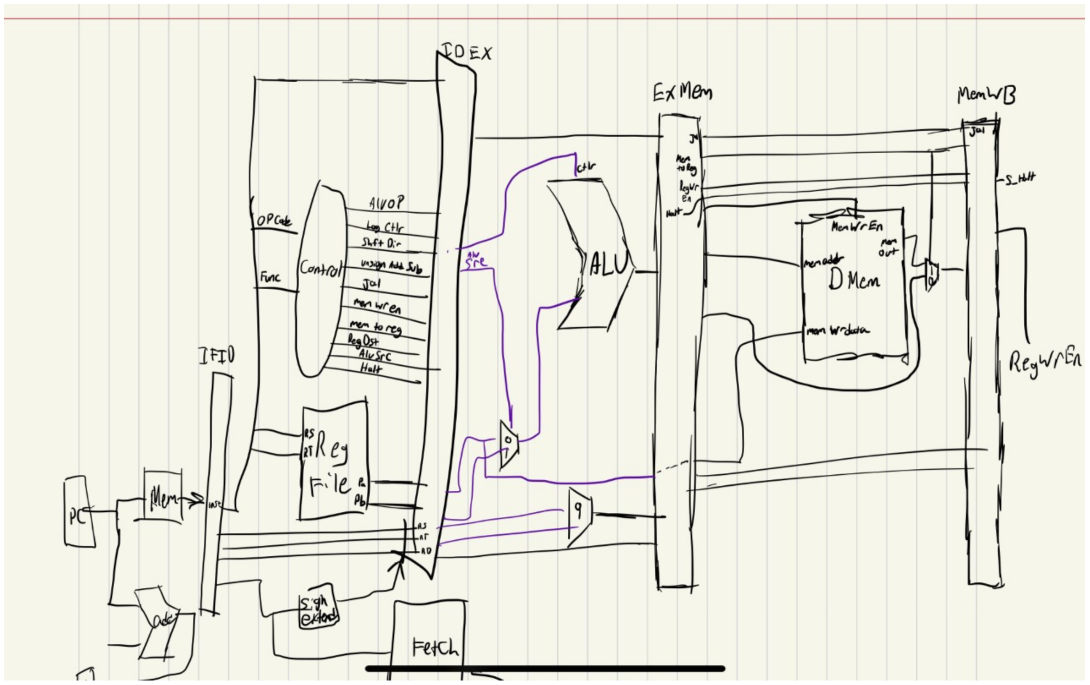
Project Teams Group #: 8-2

Refer to the highlighted language in the project 1 instruction for the context of the following questions.

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



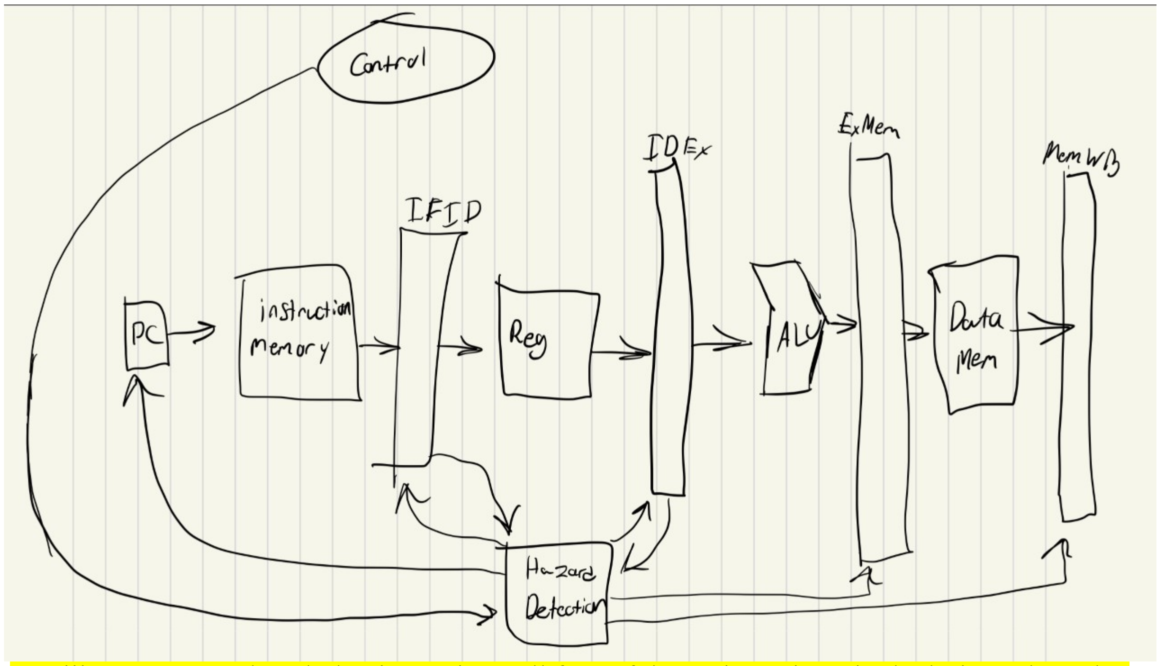
[1.b.ii] high-level schematic drawing of the interconnection between components.



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

[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.

[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.

[2.b.i] list which instructions produce values, and what signals (i.e., bus names) in the pipeline these correspond to.
Add, addi, addiu, and, andi, lui, lw, nor, xor, xori,or,ori,slt,slti,sll,srl,sra,sw,sub,subu,jal
they all are on the ALUOut signal
[2.b.ii] List which of these same instructions consume values, and what signals in the pipeline these correspond to.
Add, addi, addiu, and, andi, lui, lw, nor, xor,
xori,or,ori,slt,slti,sll,srl,sra,sw,sub,subu,jal,bew,bne,j,jr
most consume rs, and rt

[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.

DM/WB = ID/EX(alu) or with Stall

ID/EX = ALU
PC and IF/ID get stalled so ID/EX can be flushed
IF/ID flushed for Jump and PC, Stalled for JR
Branch = PC and IF/ID stalled, ID/EX Flushed

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

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

Branch instructions can cause an update as well as jump instructions. Any others would be specifically declared.

[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.

IF/ID: Load use Stall, Jump Jal Flush, Jr Stall, Branch Stall
ID/EX: Load Use, Jr, Branch Flush
EX/MEM: Alu Data
MEM/WB: Location for reg