CprE 381 – Project Part C
MIPS Pipeline Processor

Section B – Group 10
Victor Lin & Lianhao Qu

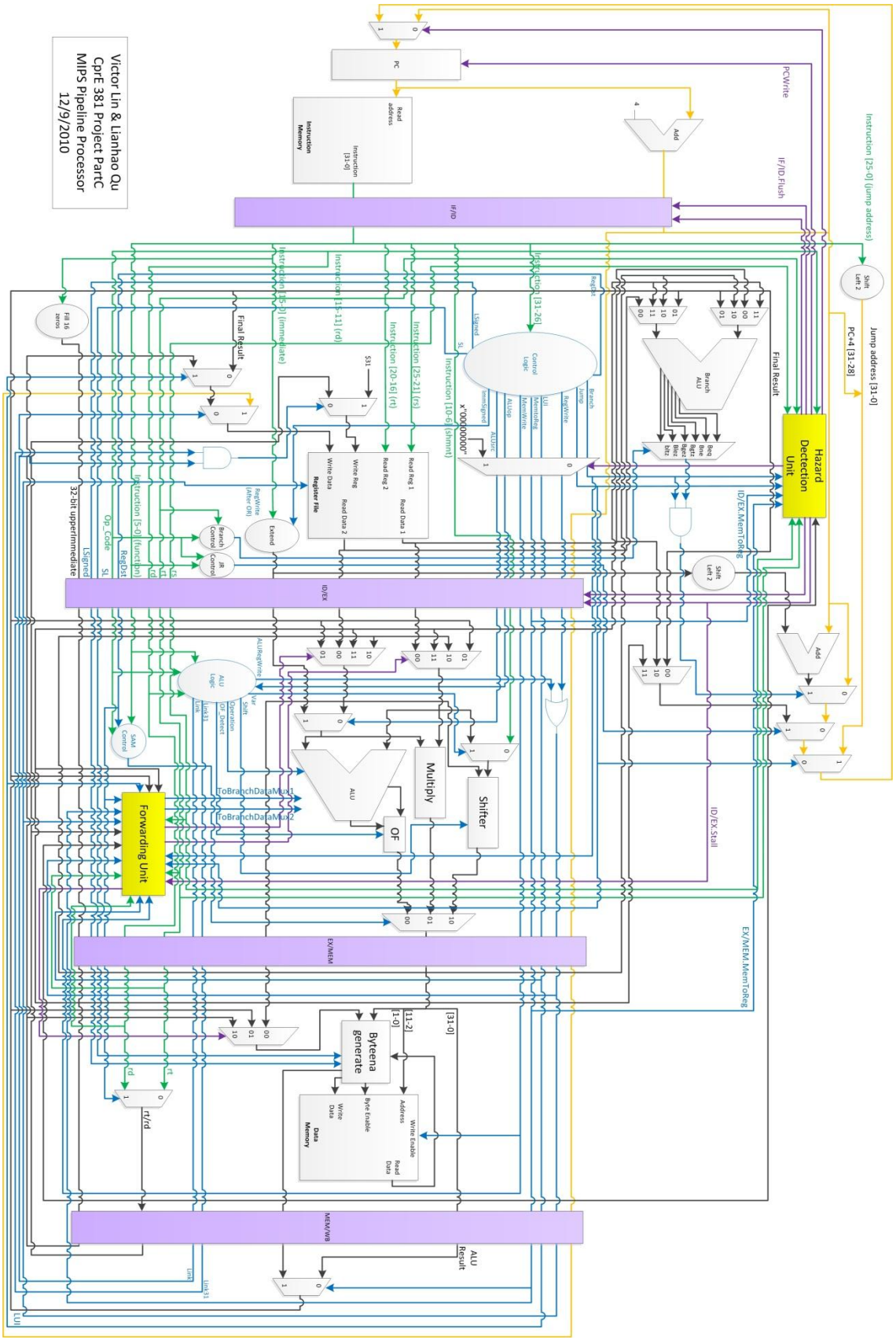
Introduction:

Following project part B, we are going to further design a pipelined processor based on our previous design.

Discussion:

1. Pipelined Registers
    a. The simple schematic about stalling or flushing a register looks something like the attached register file picture. Generally, flushing behaves like a reset that clears all the individual registers. However, stalling is the opposite of write enable. When we want to stall the registers, we have to DISABLE the write enable, and vice versa.
    b. The pipeline registers and testbench (FourRegisters_tb.vhd) are included in the zip file attached.

2. Data Dependencies
    With the 45 instructions that we were asked to implement, the attaching spreadsheet shows some relationships between the previous instruction and the following instruction. All possible scenarios are considered as we created the 45*45 list.
3. Hazard Detection and Forwarding Units
    These VHDL files are also included in the zip file.
4. MIPS Pipelined Processor
    This is the major part of the project. We combined all the VHDL codes that we've created since the beginning of the semester. The entire schematic of our pipelined processor is like the following (We've also attached a full-size picture to make it easier to see).


Victor Lin & Lianhao Qu
CprE 381 Project PartC
MIPS Pipeline Processor
12/9/2010
PC
Instruction Memory
Add
IF/ID
PCWrite
IF/ID.Flush
Instruction [25-0] (jump address)
Shift Left 2
Jump address [31-0]
PC+4 [31-28]
Hazard Dectection Unit
ID/EX.MemToReg
Final Result
Control Logic
Branch ALU
Register File
Extend
ID/EX
ALU
Multiply
Shifter
OF
Forwarding Unit
ToBranchDataMux1
ToBranchDataMux2
ID/EX.Stall
EX/MEM
EX/MEM.MemToReg
Byteena generate
Data Memory
MEM/WB
ALU Result
rt/rd

5. Testing
    a. The simple testbench application Test.txt that contains the MIPS instructions is included. Also, like the Project part B, we've also included the BubbleSort and MergeSort algorithms.

Conclusion

After this project, we are all very proud of the work that we've done during the entire semester, and we surely learned a lot from those exercises. We really hope these assignments that we've done could be extremely helpful in our future career.