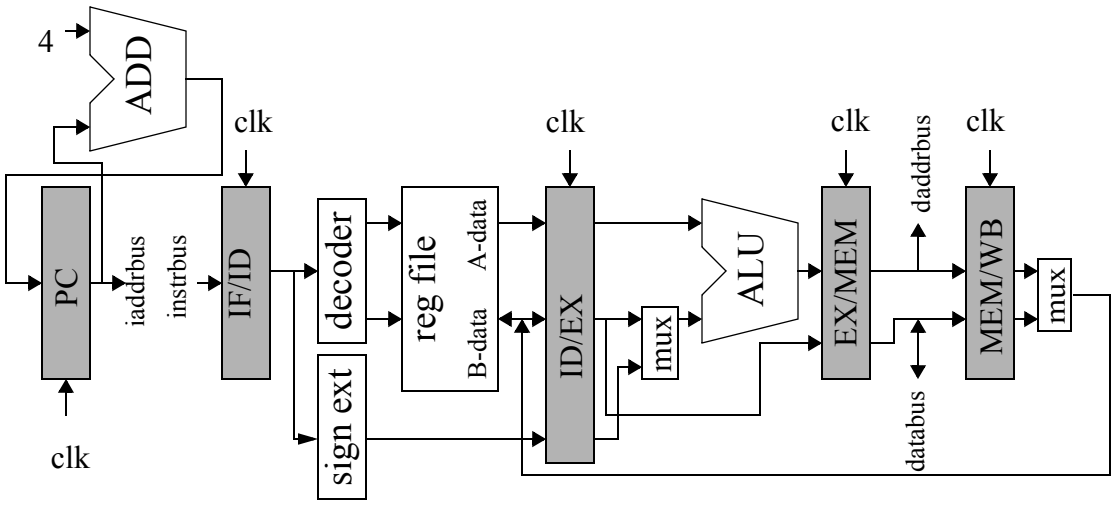
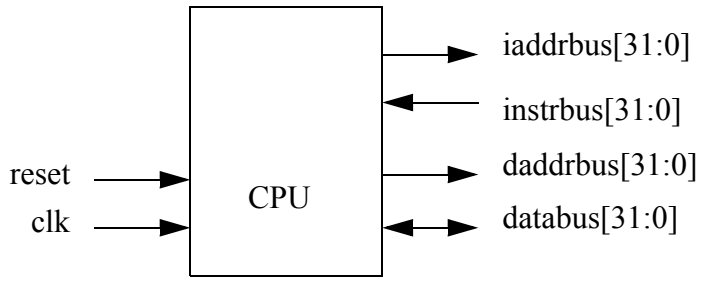
ECEN 5253
Fall 2006
Design Project 5

Due Fri., Dec. 1, 1700 hrs.

Add instruction and data memory interfaces to the design of project 4 to make it a fully pipelined 32 bit CPU capable of doing load and stores as shown in fig. 6.11 on p. 388 of Patterson and Hennessy. It should run at a 100 MHz clock rate (a full clock cycle is 10 nsec. and a half cycle is 5 nsec.). You will not need the branch logic, operand forwarding logic or load interlock logic for this project. You need only what is shown below.



To facilitate testing of your design, the instruction cache and data cache will not be modeled. Instead the instruction and data busses will be made available as external terminals and the test bench will simulate the behavior of the memory. Your CPU design should have the terminals shown below.



Note that the databus must be bi-directional.

The reset input should clear the PC (set the PC register to all 0's). It is acceptable for all of the other registers to remain undefined until something is clocked into them.

You will need to change the decode logic so that the following instructions are implemented. These are the same op codes as project 4 except for the load and store instructions.

TABLE 1. Subset of instruction codes to be implemented.

Code	Func	Name	Format
000011		ADDI	I
000010		SUBI	I
000001		XORI	I
001111		ANDI	I
001100		ORI	I
011110		LW	I
011111		SW	I
000000	000011	ADD	R
000000	000010	SUB	R
000000	000001	XOR	R
000000	000111	AND	R
000000	000100	OR	R

The instruction formats should be as in Patterson and Hennessy, inside front cover, except that only the I and R formats are implemented and the logic operations are slightly different. All other op-codes should be treated as NOPs.

Design Strategy.

1. The ID, EX and WB stages are the same as in project 4, except that the b-operand from the register file must be passed through the EX stage into the EX/MEM pipeline register to drive the databus for stores.
2. The controller (in ID) must be redesigned to pass control signals for MEM through the ID/EX and EX/MEM pipeline registers and pass control signals for WB through the ID/EX, EX/MEM and MEM/WB pipeline registers. Control signals for EX should be the same.
3. The instruction address must be produced by your CPU. This requires your CPU to put the next instruction address onto the iaddrbus at the beginning of each clock cycle. Since there are no branch or jump instructions, the PC should be incremented by 4 every clock cycle.
4. Operand forwarding and stalling on the load hazard will not be required. The compiler is assumed to insert NOP's where necessary to avoid hazards. Note that any instruction that puts its result into register 0 does nothing and is a NOP. Your processor already should work correctly in this case.

5. Use tri-state buffers to control the bi-directional data bus so that a load instruction drives the databus with the value in the EX/MEM pipeline register. A load instruction should leave the databus floating so that the data memory can drive the databus and the value gets clocked into the MEM/WB pipeline register. The databus must be left floating when it is not used by load or store instructions.

You must turn in your Verilog structural files, a list of Verilog structural files named cpu5.ver and your high level file must be named cpu5.v. Other files can be named as you wish. When using the automatic grader, be sure to use the correct class and assignment name (spelling and case is important!).

class: ecen5253
assignment: proj5