CS418	Advanced Computer Architecture	Spring 1995

Homework 4: Due 2 Feb

Reading

Thur	26	Jan	H&P 6.1–6.5
Tue	31	Jan	skim H&P 6.6–6.9.
Thur	02	Feb	H&P Appendix E

Note

- If you are confused about a feature or design decision for a hypothetical microprocessor described in a homework (*e.g.* how many registers does it have?), use the information for the DLX described in H&P, **and** say what you are assuming.
- When drawing "execution tables" for pipelines:
  - If a value does not change from one clock-cycle to the next, simply draw a horizontal line, as has been done for the registers from time 2–5 in Figure 3.
  - For each stage, show the computation that it performs (*e.g.* `R7=S1&S2` for the Fetch-Operands stage in cycle 3 of Figure 2), and *not* just the instruction in the stage.
  - The gray region at the left hand side of the tables represent computations that happened in the past, which we are not concerned with.
  - Additional, blank execution tables will be available on the Web pages.

General Information for This Homework

- All of the problems refer to the pipeline shown in Figure 1.
- All "normal" arithmetic and logical instructions (*e.g.* ADD, SUB, XOR, MULT, *etc.*) use the Execute stage and then go directly to WriteBack. Divide (DIV) instructions use Execute, and then loop through the Divide stage twice, before going to WriteBack.
- As with all pipeline stages, each stage can contain *at most one instruction.*
- "BY" is the register for the bypass from the output of Execute to its input.
- "S1" and "S2" are the registers where the register file puts the values currently stored in the registers corresponding to the operands in the FetchOps stage (*e.g.* if FO has `ADD R5, R3, R2`, then S1 has the value currently stored in `R3` and S2 has the value currently stored in `R2`.
- There is no bypass path from the output of Divide to the input of Execute. Divide instructions *must* be written to the register file before they can be used.

- The block diagram of the pipeline does not show the functional units for loads and stores, because this homework does not mention those instructions.

Problem 1 (9 Points) Given the following assembly code:

```
; initial values:  R3=A, R4=B, R5=C, R6=D, R7=E
1000 :  AND R7, R3, R4  ; F = A&B
1004 :  XOR R3, R5, R6  ; G = C^D
1008 :  SUB R7, R3, R4  ; H = G-B
100C :  ADD R7, R3, R7  ; I = G+H
```

(a) (4 Points) **Assuming that the bypass path in Figure 1 does not exist**, complete the execution table in Figure 3.

For each clock-cycle show the values stored/produced at the end of the clock-cycle for:

- (PC) Program counter
- (FI, DI, FO, EX, WB) Output of each stage in the pipeline
- (S1, S2) Source operands produced by register file
- (R3–R7) Registers

(b) (4 Points) **Using the bypass path**, complete the execution table in Figure 2.

For each clock-cycle show the values stored/produced at the end of the clock-cycle for:

- (PC) Program counter
- (FI, DI, FO, EX, WB) Output of each stage in the pipeline
- (BY) Execute unit bypass path
- (S1, S2) Source operands produced by register file
- (R3–R7) Registers

Problem 2 (8 Points)

Given the following assembly code:

```
; initial values:  R2=A, R3=B, R4=C, R5=D, R6=E, R7=F
1000 :  ADD R5, R4, R6
1004 :  DIV R5, R5, R4
1008 :  SUB R5, R3, R4
100C :  ADD R4, R2, R3
1010 :  XOR R5, R6, R4
```

(a) (1 Point) Comment the code by giving a unique character to each value, as was done in lecture. (NOTE: Be nice to the TA and assign characters in alphabetical order.)

(b) (5 Points) Complete the execution table in Figure 4.

For each clock-cycle show the values stored/produced at the end of the clock-cycle for:

- (PC) Program counter
- (FI, DI, FO, EX, WB) Output of each stage in the pipeline
- (BY) Execute unit bypass path
- (S1, S2) Source operands produced by register file
- (R3–R7) Registers

(c) (2 Points) What would be the costs and performance gains, if any, for adding a bypass path from the output of the Divide stage to the input of the Execute stage?

Problem 3 (1 Point)

How much time did you spend on each of these problems? What was the educational value of each problem? How well did the lectures and text book prepare you for this homework? (Which concepts needed better explanation?)

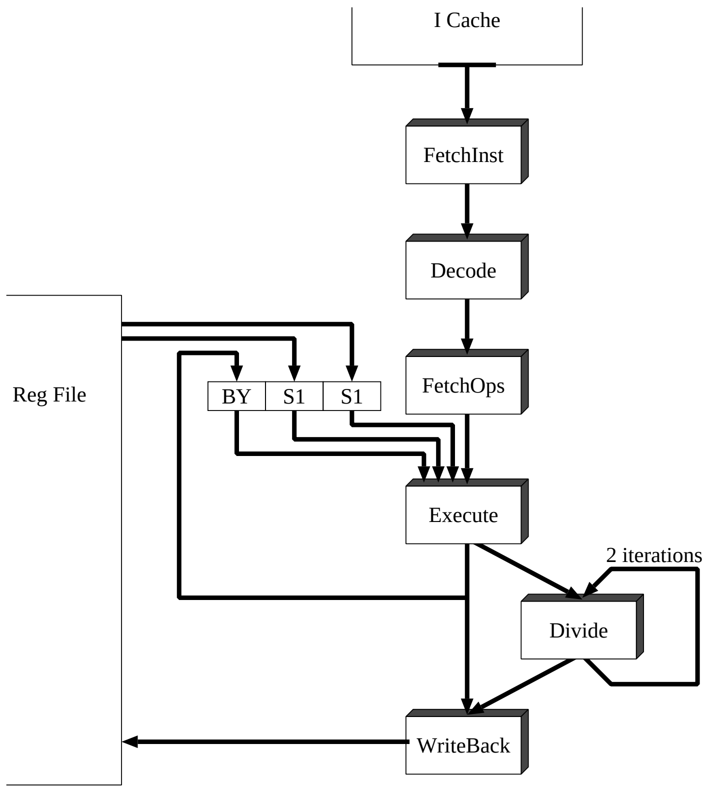

Figure 1: Block Diagram of Microprocessor Pipeline

	0	1	2	3	4	5	6	7	8	9	10	11
PC	1000	1004										
FI		AND R7,R3,R4										
DI			R7=R3&R4									
FO				R7=S1&S2								
EX					R7=F							
WB						R7=F						
S1				A								
S2				B								
R3			A									
R4			B									
R5			C									
R6			D									
R7			Q			F						

	0	1	2	3	4	5	6	7	8	9	10	11
PC	1000	1004										
FI		AND R7,R3,R4										
DI			R7=R3&R4									
FO				R7=S1&S2								
EX					R7=F							
WB						R7=F						
BY					F							
S1				A								
S2				B								
R3			A									
R4			B									
R5			C									
R6			D									
R7			E			F						

	0	1	2	3	4	5	6	7	8	9	10	11
PC	1000											
FI		ADD R5,R4,R6										
DI												
FO												
EX												
DIV												
WB												
BY												
S1												
S2												
R3			A									
R4			B									
R5			C									
R6			D									