

ENEE646 Fall 2000

Midterm Examination

Name: _____ Answer key _____
Student ID: _____

Time allotted: **1 hour 10 minutes.**
Maximum score: **50 points**

There are 7 questions, each with subparts.

1. For each subpart (i) to (iii) below, circle **one** correct answer from among the four given. **(2 points*3 = 6 points)**

(i) A program runs in 0.15 seconds on an 800Mhz processor with 40% processor cycles spent on the program. The program executes 37 million instructions. The average program IPC is:

- (a) 0.31
- (b) 0.77
- (c) 1.30
- (d) 3.24

Answer: b

(ii) In the best-case implementation of a 5-stage DLX pipeline, a branch mispredict results in how many cycles wasted?

- (a) 0
- (b) 1
- (c) 2
- (d) 3

Answer: b

(iii) In a fully bypassed 5-stage DLX pipeline with a data cache latency of 1 cycle, a truly register-dependent instruction immediately following a load must stall for how many cycles?

- (a) 0
- (b) 1
- (c) 2
- (d) 3

Answer: b

2. For each subpart (i) to (vii) below, circle **all** correct answers from among the four given - note that more than one answer may be correct!

(2 points*7 = 14 points)

(i) Assuming current processor technology trends, it is expected that in the future the following will increase:

- (a) Cross-chip wire delay in cycles.
- (b) Cross-chip wire delay.
- (c) Memory latency in processor cycles.
- (d) The ratio of on-chip silicon area for computation versus memory.

Answers: a,c

(ii) The advantages of RISC ISAs over CISC ISAs include the following:

- (a) They are more amenable to pipelining.
- (b) On average, they need fewer executed instructions for the same program.
- (c) Compilers tend to exploit them better.
- (d) They offer greater flexibility in addressing modes.

Answers: a,c

(iii) The following are defined by most ISAs:

- (a) The maximum amount of addressable memory.
- (b) The kinds of addressing modes.
- (c) The latencies of different operations.
- (d) The length of pipeline interlocks due to hazards.

Answers: a, b

(iv) The advantages of general-purpose register architectures over stack architectures include:

- (a) More compact instruction encoding.
- (b) On average, fewer memory instructions executed for the same source program.
- (c) Inherently have a larger amount of processor state
- (d) Are easier to compile to.

Answers: b

(v) For correctness, instructions executed along a branch-predicted path before the outcome is known must not:

- (a) Read operands.
- (b) Commit results.
- (c) Signal terminating exceptions.
- (d) Execute out-of-order.

Answers: a

(vi) Tomasulo's method is intended to enable:

- (a) Multiple instructions to issue per cycle.
- (b) Out-of-order execution.
- (c) Out-of-order issue.
- (d) Overlapping of functional unit latencies.

Answers: b, d

(vii) To handle synchronous interrupts, the processor hardware must:

- (a) Complete the instruction causing the interrupt before handling the interrupt.
- (b) Know the location of the interrupt handler for each kind of interrupt.
- (c) Provide hardware support for saving and restoring processor state.
- (d) Know the number and nature of interrupts.

Answers: d

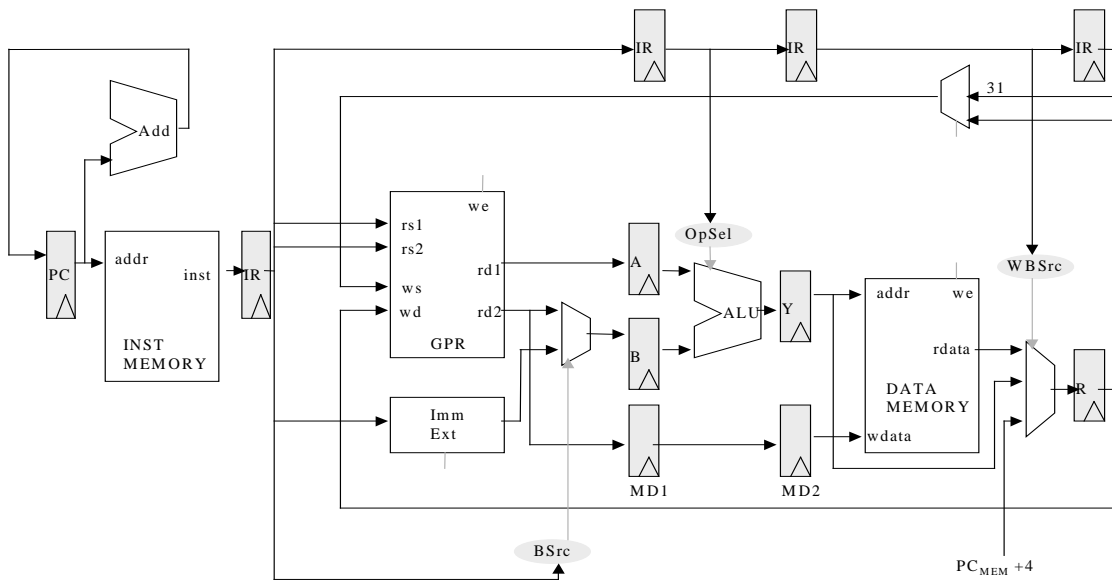
3. In Scoreboarding, for each kind of dependence given, write down the stage (issue/read operands/execute/write result) that checks for that dependence:

(3 points)

True dependence: _____ Read operands _____ stage
Output dependence: _____ Issue _____ stage
Anti-dependence: _____ Write result _____ stage

4. The following picture shows a DLX integer pipeline with some control signals missing. Fill in the blanks in the specifications below of the WBSrc and OpSel control signals. WBSrc is a 3-way multiplexor select, and OpSel is the operation performed specification for the ALU. Assume that all DLX integer instructions are classified as one of the following: ALU, ALUi, LW, SW, BNEZ, BEQZ, J, JR, JAL, JALR. (ALUi are ALU operations with one immediate operand).

(6 points)



As an example, here is the specification of the BSrc signal:

$$\begin{aligned} \text{BSrc} &= \text{case Opcode}_{ID} \\ \text{ALU} &\Rightarrow \text{rd2} \\ \text{ALUi, LW, SW} &\Rightarrow \text{Imm Ext} \end{aligned}$$

Now fill in the blanks below:

$$\begin{aligned} \text{WBSrc} &= \text{case Opcode}_{MEM} \\ \underline{\text{ALU, ALUi}} &\Rightarrow \text{Y} \\ \underline{\text{LW}} &\Rightarrow \text{rdata} \\ \underline{\text{JAL, JALR}} &\Rightarrow \text{PC}_{MEM} + 4 \end{aligned}$$

$$\begin{aligned} \text{OpSel} &= \text{case Opcode}_{EX} \\ \text{ALU, ALUi} &\Rightarrow \underline{\text{OP}_{EX}} \\ \text{LW, SW} &\Rightarrow \underline{+} \\ \text{BEQZ} &\Rightarrow \underline{\text{REG} = 0?} \\ \text{BNEZ} &\Rightarrow \underline{\text{REG} \neq 0?} \end{aligned}$$

5. For parts (i) to (iii) below, consider the following code sequence running on a 5-stage DLX pipeline:

(2 points * 3 = 6 points)

SW 6(R5), R4
LW R1, 3(R2)

- (i) Is it possible for the two instructions to ever have a dependence? Is yes, under what condition?

Yes. Memory dependence when $R5 + 6 == R2 + 3$ at runtime.

- (ii) Assume that the latency of the data cache is one cycle to complete a memory operation. Will the above code execute correctly without stalling if the data cache were implemented as a simple memory bank with no associated logic? If no, then what associated logic would allow stall-free execution? (1-2 sentences)?

Yes

- (iii) Same question as (ii) above, but with a two-cycle data cache latency:

No. Need bypass paths in memory that forward values from outstanding stores to load requests.

6. Consider the following code sequence for a loop:

(11 points)

```
Loop: LD      F0, 0(R5)
      MULD   F4, F4, F0
      SUBI   R5, R5, #8
      BNEZ   R5, Loop
```

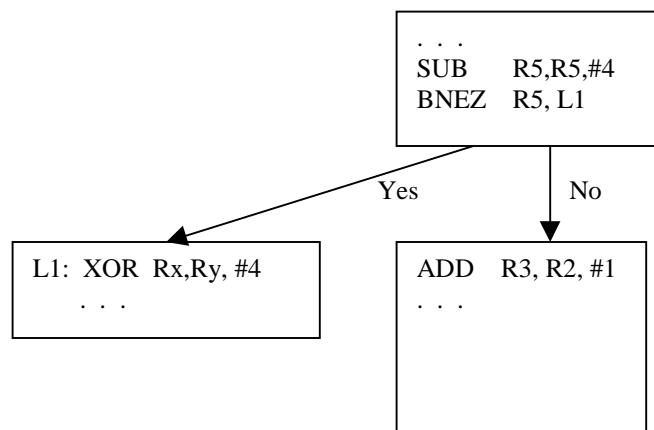
By filling out table below, show the cycle-by-cycle execution of the different stages of each instruction in the first two iterations of the loop on Tomasulo hardware. The subscript on the instructions in the first column of the table is the iteration number. Show the behavior for the first 12 cycles only. Assume the following:

- There are no structural hazards on reservation stations, buffers or functional units in the first two iterations.
- There is only one common data bus (CDB). In case of a bus conflict, the longer-running functional unit takes precedence on it.
- Multiply functional units have a 5-cycle latency; the integer ALU has a 1-cycle latency.
- The data cache has a 3-cycle execution latency, including address computation.
- A branch target buffer is present, and predicts the first two branches correctly.
- The write result stage writes its value at the beginning of its cycle, allowing waiting units to begin execution immediately.

Ex = Execute, Wr= Write Result. Use NW for a null write-result (insts with no destination reg). Leave Ex cycles waiting for input as blank in the table.

Instruction ↓	Cycle →											
	1	2	3	4	5	6	7	8	9	10	11	12
LD ₁	Issue	Ex	Ex	Ex	Wr							
MULD ₁		Issue			Ex	Ex	Ex	Ex	Ex	Wr		
SUBI ₁			Issue	Ex		Wr						
BNEZ ₁				Issue		Ex	NW					
LD ₂					Issue	Ex	Ex	Ex	Wr			
MULD ₂						Issue				Ex	Ex	Ex
SUBI ₂							Issue	Ex			Wr	
BNEZ ₂								Issue			Ex	NW

7. Consider the following fragment of the control-flow graph (CFG) of a program:
(2 points * 2 = 4 points)



A trace-scheduling compiler wants to move the ADD instruction to just before the BNEZ instruction. Assume that it is assured that the ADD instruction will not raise an exception.

(i) Give the constraints on the values of x and y that, just by themselves, will ensure that the move is legal.

$$y \neq 3, x == 3$$

(ii) Assume that the legality constraints in (i) are violated, and x and y cannot be changed. If the compiler has an unused register Rnew available, describe how the code can be modified to make the move legal.

Compiler-rename R3 in result of ADD to Rnew : ADD Rnew, R2, #1.

Also change uses of that value of R3 to Rnew.
