CS 250 Spring 2017
Purdue University
Practice Midterm Exam 02

1. The register file is used in two stages of the 5-stage pipelined version of Figure 6.9.
A True
B False

2. Which instruction is the most likely to cause a structural hazard in the EX stage of a traditional 5-stage pipeline?
A load
B branch
C sign-magnitude integer compare
D floating point divide
E BCD addition

3. Pipelining is a form of hardware parallelism.
A True
B False

4. **Two versions of this question.** Assume that a five-stage instruction execution pipeline (IF, ID, EX, MEM, WB stages) is capable of hazard detection, but it is not capable of forwarding, so it stalls instead. The register file for this pipeline **can/can not** be written to and then read from during one clock cycle. How many clock cycles will it take to execute the following two instruction assembly language sequence?

```
sub $6, $7, $8        ; $6 ← $7 – $8
add $5, $6, $7        ; $5 ← $6 + $7
```

Clock	1	2	3	4	5	6	7	8
sub	IF	ID	EX	MEM	WB			
add		IF	ID	stall	stall	EX	MEM	WB

A 5
B 6
C 7
D 8
E None of the above

5. Which of the following lists presents its terms in order from least abstract to most abstract?
A register, full adder, Boolean logic, S’R’ latch.
B “bare metal” computer, silicon wafer, silicon chip, wire.
C transistor, gate, function unit, machine instruction execution.
D multiplexer, decoder, data path, address.

E None of the above is in order from least abstract to most abstract

6. What is the **most/least** complex of the combinatorial circuits in a pipelined processor?
   A The multiplexer in the WB stage.
   B The largest of the pipeline stage interface registers.
   C The ALU.
   D The sign extender.
   E No one of the above is always the most complex

7. An 8-byte data item is stored in the byte-addressed memory of a 32-bit computer at address 0x004488cc. To load this item into registers R1 and R2 so that R1 contains the most significant bytes and R2 the least significant bytes, the processor should
   A Execute two load word instructions, one at address 0x004488cc and one at 0x004488d0.
   B Execute two load word instructions, one at address 0x004488c8 followed by one at 0x004488cc.
   C Execute eight load word instructions, one each from addresses 0x004488cc through 0x004488d3.
   D It is not possible to load this data item into two registers as described
   E None of the above is correct

8. Designers are working on a new, pipelined version of the processor with the four instructions shown in text Figure 6.2 and the circuit of Figure 6.9. The have decided to save money and instead of having a dedicated sign extension unit in the ID stage, the ALU in the EX stage will be used to perform sign extension as a dedicated operation. These designers have
   A Increased the speed of the ID stage.
   B Created a structural hazard for memory access.
   C Forgotten that there is a limit on the number of different circuits (add, subtract, etc.) that may be placed in the ALU.
   D Created a data hazard when reading offset values.
   E None of the above is correct

9. What happens to instruction(s) in a pipeline that were fetched earlier than an instruction that is stalled?
   A Those instruction(s) stall, also.
   B Those instruction(s) do not stall.
   C Those instruction(s) do not stall if they have no dependence with the stalled instruction.
   D Those instruction(s) stall if they have a true data dependence with the stalled instruction.
   E Those instruction(s) are replaced with NOP instruction(s).

10. Consider the following three-instruction assembly code sequence.

```
          store r5, r7, 4          ; memory[r7 + 4] ← r5
```

```
add r7, r7, r8          ; r7 ← r7 + r8
load r3, r6, –16        ; r3 ← memory[r6 – 16]
```

In addition the instruction schedule shown above, *how many more* valid schedules exist for this instruction sequence?

A 0
B 1
C 2
D 3! = 3 x 2 x 1
E None of the above

11. Consider this code.

```
for (i=1000; i>=1; i=i-1) {
        x[i] = x[i+1] + s;
}
```

Can this loop be unrolled 4 times and, after performing or not performing the loop unrolling, how many basic blocks will the code have?

A Cannot be unrolled 4 times; will have 1 basic block
B Cannot be unrolled 4 times; will have 1000 basic blocks
C Can be unrolled 4 times; will have 1 basic block.
D Can be unrolled 4 times; will have 4 basic blocks
E Can be unrolled 4 times; will have 250 basic blocks

12. Which instruction causes a control hazard in a pipelined processor?

A load R1, R2, 16
B add R1, R1, R3
C store R1, R2, 32
D subi R3, R3, 8
E bne R2, R3, target

13. Loop unrolling

A Increases the number of instructions in instruction memory
B Reduces the number of instructions fetched
C Increases basic block size
D All of the above
E None of the above

14. Refer to the following code the answer the next three questions.

```
        loadi   r5, 16          ; r5 ← 16  (load immediate value into register)
loop:   load    r1, r5, 40      ; r1 ← memory[r5 + 40]
        add     r1, r1, r2      ; r1 ← r1 + r2
        addi    r5, r5, -4      ; r5 ← r5 – 4
        store   r1, r5, 40      ; memory[r5 + 40] ← r1
        bne     r5, 0, loop     ; branch to loop if r5 != 0  (r5 not equal to zero)
```

If the loop is unrolled 2 times, the number of instructions in the resulting loop body

A must be 10
B need only be 8
C need only be 6
D Cannot be determined until the instructions are scheduled
E None of the above

15. As the code above executes, there is a true data dependence from the store to the load.
A True
B False

Part II: Short Answer Questions
Legibly store your answers to these questions on this exam document. Keep your answers short and to the point.

1. A 64-bit RISC computer has byte-addressed memory. At what address should we expect to find the default next instruction? ____________________

2. How many different bit strings correspond to machine language **ADD/LOAD** instructions for the format of text Figure 6.2? _______.

3. Branch prediction has a positive __________ when correct, and likely a negative one when incorrect.

4. Register renaming can remove ___________________ from assembly language code.

5. A zero-address computer executes a subtract instruction:
Difference = Subtrahend – Minuend
The Minuend value is sourced from the _______________________________ and

the Difference destination is ___________________. (Be specific)

6. Computer Y is a 3-address architecture for which operands and results reside in a 4 GByte memory. Computer Z is a 3-address computer for which operands and results reside in the register file. What is the most significant difference between formats of the ADD instructions for computers Y and Z?.

7. Two assembly language instructions are defined as follows.

```
jump offset(register)   ; PC ← offset + contents of register
brr regA, regB, offset  ; if contents of regA = contents of regB, then
                        ;     PC ← PC + offset else  PC ← PC + 4
```

Which instruction, jump or brr, is easier to use when writing program loops and why?

8. What, if any, constrains are there on the execution order of the following assembly language instructions?

```
label01:        add R1, R3, R7          # R1 = R3 + R7
```

```
label02:        sub R4, R4, R6
label03:        add R9, R1, R2
label04:        shift R12, 5
```

9. Consider the processor circuit below and the instruction lw defined as follows.
   lw r4, r3   ; r4 ← memory[r3]
   Show how to modify Figure 6.9 to perform the lw ~~instruction~~ memory access during the EX stage. Clearly draw in any necessary new data path(s) and clearly mark with an “x” any data paths that must be removed.

10. Consider pipelining the circuit of text Figure 6.9 into five stages: IF, ID, EX, MEM, and WB. On this figure, use a thin line, rather than a thin rectangle, to show the location of the ID/EX pipeline register. Your line must cut through all the signal paths shown in Figure 6.9 that connect to this register. You may ignore any signals not shown in 6.9 that would be part of an actual ID/EX register.

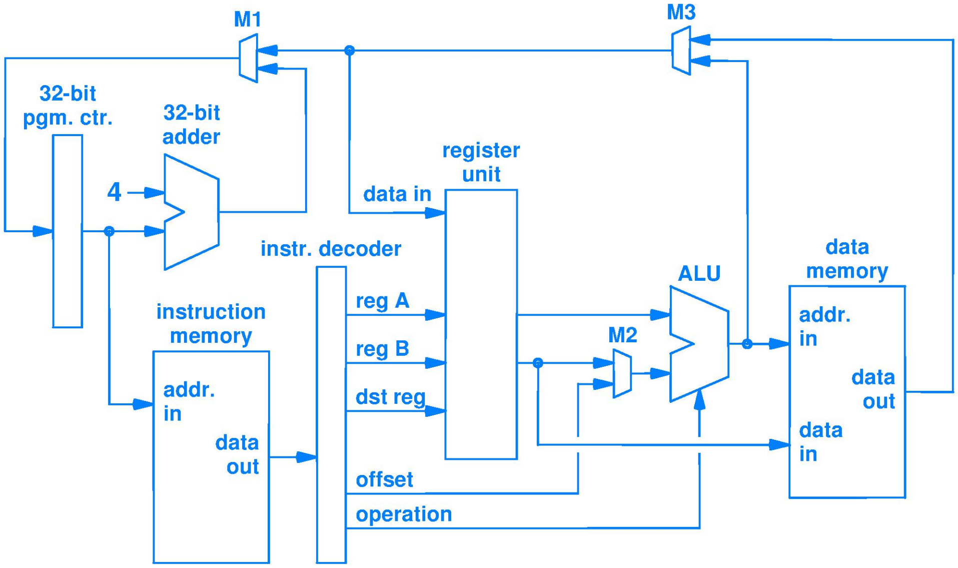

Figure 6.9 Illustration of data paths including data memory.

16. Refer to Figure 6.9 for the following set of questions (a) through (g). If asked to specify parts of Figure 6.9 please use *exactly* the same names as shown in the figure. For example, write **32-bit pgm. ctr.** or **M2** if one of these is a component you wish to name; write **dst reg** if that is a signal path you wish to name.

    A With respect to execution of the four instructions shown in Figures 6.1 and 6.2, list all the components in Figure 6.9 that output pointers only.

    B With respect to execution of the four instructions shown in Figures 6.1 and 6.2, list all the components in Figure 6.9 that sometimes output a pointer and sometimes output a value.

C After an **add** instruction the value of **32-bit pgm. ctr.** will be updated to what new value?

D The instruction format of Figure 6.2 can use addressing mode 3 in Figure 7.6 with a memory up to what maximum size?