EECS 370: Introduction to Computer Organization Winter 2005
University of Michigan

---

# EXAM 2

Sec 001
Thursday March 31, 2005
9:40-10:55 a.m.
1311 EECS

## Instructions

1. This is an open books/notes exam.
2. It comprises 17 multiple-choice questions weighted as follows:
   - 14 multiple-choice questions each worth 5 points
   - 3 multiple-choice questions each worth 10 points.
3. Each question has only one correct answer.
4. For each question, indicate your answer on the answer sheet provided.
5. You have 75 minutes to complete the exam.
6. Please make sure you enter your name in the space below AND on the answer sheet.
7. By signing below, you certify that your conduct throughout the exam has been in accordance with the College of Engineering Honor Code.
8. At the end of the exam, please turn in this exam booklet AND your answer sheet.

**Name** _____

**Signature** _____

## Performance

1. [5 points] Consider the single-cycle, multi-cycle, and pipelined versions of LC2K5 discussed in class. All else being equal, one would expect the CPI for all three versions to be:

    a. pipelined ≤ single-cycle ≤ multi-cycle
    b. pipelined ≤ multi-cycle ≤ single-cycle
    c. multi-cycle ≤ pipelined ≤ single-cycle
    d. single-cycle ≤ pipelined ≤ multi-cycle
    e. single-cycle ≤ multi-cycle ≤ pipelined

2. [5 points] The major advantage of the multi-cycle datapath over single-cycle datapath is:

    a. The ability to allow instructions to take shorter cycle time
    b. The ability to allow instructions to take same numbers of clock cycles
    c. The ability to allow a functional unit to be used more than once per instruction in the same clock cycle
    d. The ability to overlap executing different instructions
    e. None of the above

3. [5 points] In which type of datapath does the instruction with the FASTEST execution time determine the cycle time?

    a. In single-cycle datapath
    b. In multi-cycle datapath
    c. In pipelined datapath with forwarding
    d. In pipelined datapath without forwarding
    e. None of the above

4. [5 points] Consider a 5-stage IF|ID|EX|MEM|WB pipeline in which memory accesses take 3 cycles (i.e., given an address, the memory takes 3 cycles to produce the data requested). All other stages can execute in 1 cycle. Assuming perfect branch prediction, full data forwarding, and no load hazards, what is the minimum achievable CPI?

    a. 1
    b. 1.4
    c. 2
    d. 3
    e. 7

5. [5 points] Consider the pipelined LC2K5 datapath as discussed in class. Assume it runs programs with the following instruction mix: 15% add, 15% nand, 20% LW, 10% SW, 40% BEQ. Branch prediction is "speculate always-not-taken and squash". All load and store operations access memory in one cycle. Data hazards are resolved by forwarding whenever possible. 50% of LW instructions are followed by a dependent instruction. 50% of BEQ instructions are taken. What is the CPI?

    a. 1.5
    b. 1.6
    c. 1.7
    d. 1.8
    e. None of the above

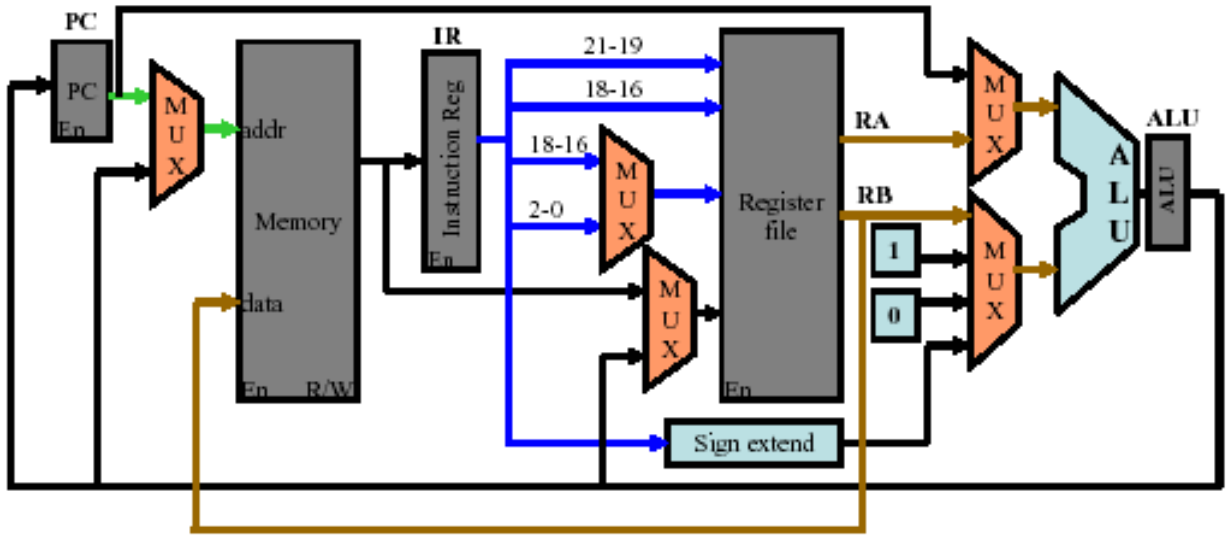
6. [5 points] On some unknown pipeline, we attain a CPI of 1.45 for an application with the following profile.

    - R-type: 45%
    - loads: 25%
    - stores: 10%
    - branches: 20%

    Assume that the datapath stalls on branches. Given that 20% of the loads incur a 1-cycle load hazard penalty, what is the branch penalty (in cycles)?

    a. 1
    b. 2
    c. 3
    d. 4
    e. 5

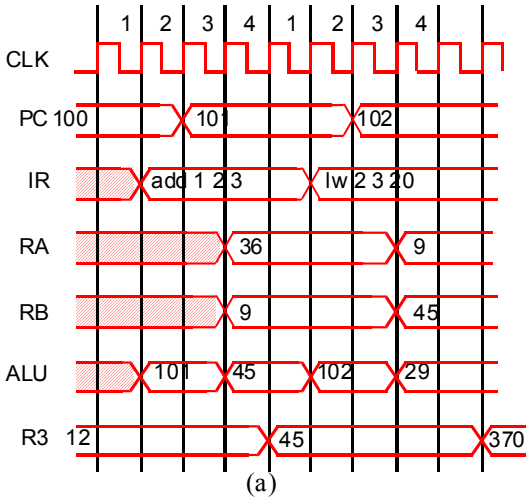
## Multi-Cycle Datapath



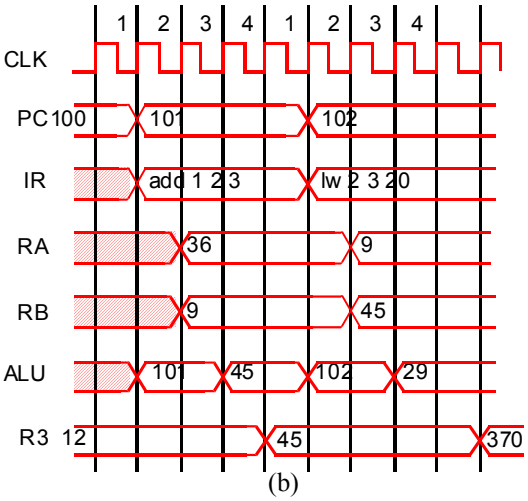

7. [10 points] Consider the execution of the following program on the multi-cycle implementation of LC2K5 shown above:

   add 1 2 3
   lw 2 3 20

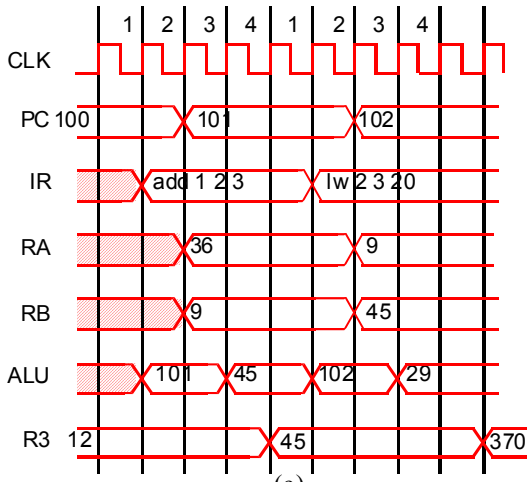
   Assume that the add instruction is at memory location 100, and that the content of memory location 29 is the integer 370. Also assume that the register file is initialized to: R1=36, R2=9, and R3=12. Which of the following timing diagrams corresponds to the execution of the above program? Assume that all registers are positive edge-triggered, and ignore flip-flop delays.


1 2 3 4 1 2 3 4
CLK
PC 100 101 102
IR add 1 2 3 lw 2 3 20
RA 36 9
RB 9 45
ALU 101 45 102 29
R3 12 45 370

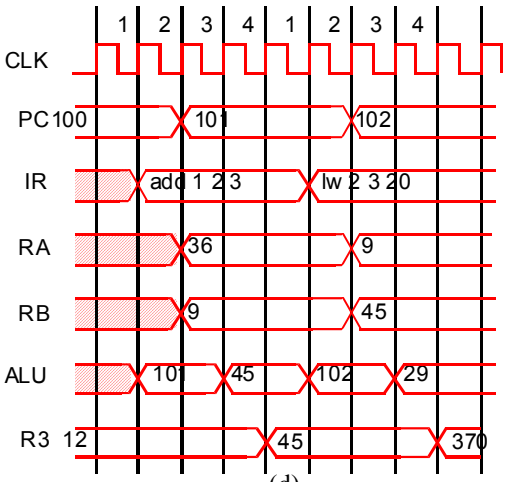
(a)


1 2 3 4 1 2 3 4
CLK
PC 100 101 102
IR add 1 2 3 lw 2 3 20
RA 36 9
RB 9 45
ALU 101 45 102 29
R3 12 45 370

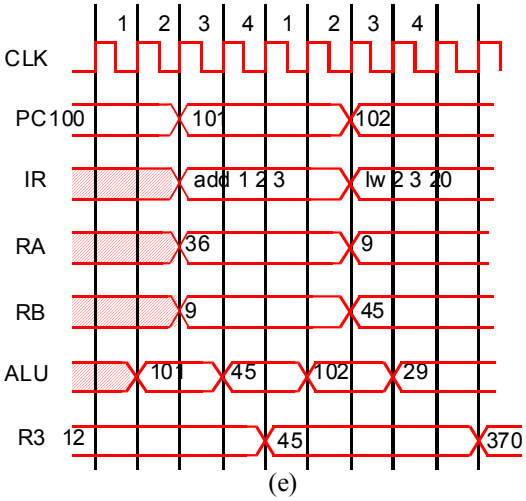
(b)


1 2 3 4 1 2 3 4
CLK
PC 100 101 102
IR add 1 2 3 lw 2 3 20
RA 36 9
RB 9 45
ALU 101 45 102 29
R3 12 45 370

(c)


1 2 3 4 1 2 3 4
CLK
PC 100 101 102
IR add 1 2 3 lw 2 3 20
RA 36 9
RB 9 45
ALU 101 45 102 29
R3 12 45 370

(d)


1 2 3 4 1 2 3 4
CLK
PC 100 101 102
IR add 1 2 3 lw 2 3 20
RA 36 9
RB 9 45
ALU 101 45 102 29
R3 12 45 370

(e)

## Pipelining I

For Questions 8 through 11, consider the following piece of LC2K5 assembly code.

```
add   r1   r2   r3
add   r3   r4   r1
nand  r3   r2   r6
nand  r5   r2   r7
add   r3   r5   r1
lw    r6   r8   100
add   r8   r7   r2
halt
```

8. [5 points] How many dependencies are there in this code?

    a. 5
    b. 6
    c. 7
    d. 8
    e. 9

9. [5 points] Consider the 5-stage IF|ID|EX|Mem|WB LC2K5 pipeline described in lecture with full forwarding to EX from EX|Mem, Mem|WB, and the register file. How many cycles does it take for the code to execute? (Include cycles required to fill and drain the pipeline.)

    a. 12
    b. 13
    c. 14
    d. 15
    e. 16

10. [5 points] Consider a 6-stage version IF|ID|EX1|EX2|Mem|WB of the LC2K5 pipeline in which the Execute stage takes 2 cycles to generate the result of ALU operation. Assume:
    - Full forwarding to EX1 from EX2|Mem, Mem|WB, and the register file.
    - If data are not available to forward, the pipeline stalls.

    How many cycles does it take for the code to execute? (Include cycles required to fill and drain the pipeline.)

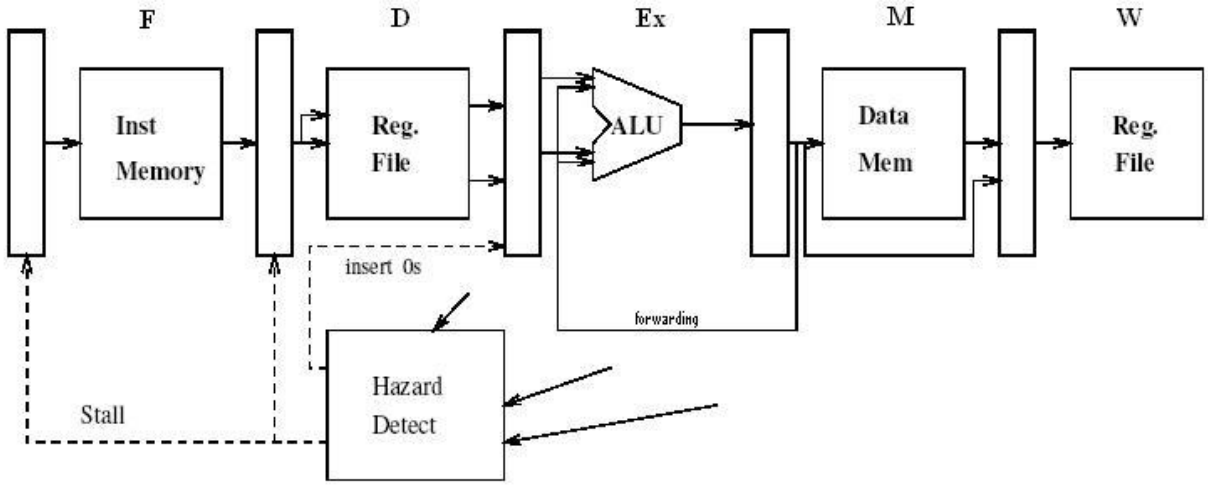
    a. 12
    b. 13
    c. 14
    d. 15
    e. 16

11. [10 points] Now, consider an out-of-order 2-way superscalar 5-stage IF|ID|EX|Mem|WB pipelined version of LC2K5. Assume full forwarding to EX from EX|Mem, Mem|WB, and the register file. Which timeline describes a correct execution of the code on this datapath?

| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| a. | add | IF | ID | EX | M | WB | | | | | |
| | add | IF | ID | EX | M | WB | | | | | |
| | nand | | IF | ID | EX | M | WB | | | | |
| | nand | | IF | ID | EX | M | WB | | | | |
| | add | | | IF | ID | EX | M | WB | | | |
| | lw | | | IF | ID | EX | M | WB | | | |
| | add | | | | IF | ID | EX | M | WB | | |
| | halt | | | | IF | ID | EX | M | WB | | |
| b. | add | IF | ID | EX | M | WB | | | | | |
| | add | | IF | ID | EX | M | WB | | | | |
| | nand | | IF | ID | EX | M | WB | | | | |
| | nand | | | IF | ID | EX | M | WB | | | |
| | add | | | IF | ID | EX | M | WB | | | |
| | lw | | | | IF | ID | EX | M | WB | | |
| | add | | | | IF | ID | EX | M | WB | | |
| | halt | | | | | IF | ID | EX | M | WB | |
| c. | add | IF | ID | EX | M | WB | | | | | |
| | add | | IF | ID | EX | M | WB | | | | |
| | nand | | IF | ID | EX | M | WB | | | | |
| | nand | IF | ID | EX | M | WB | | | | | |
| | add | | | IF | ID | EX | M | WB | | | |
| | lw | | IF | ID | EX | M | WB | | | | |
| | add | | | | IF | ID | EX | M | WB | | |
| | halt | | | | IF | ID | EX | M | WB | | |
| d. | add | IF | ID | EX | M | WB | | | | | |
| | add | | IF | ID | EX | M | WB | | | | |
| | nand | | IF | ID | EX | M | WB | | | | |
| | nand | IF | ID | EX | M | WB | | | | | |
| | add | | | IF | ID | EX | M | WB | | | |
| | lw | | | IF | ID | EX | M | WB | | | |
| | add | | | | | IF | ID | EX | M | WB | |
| | halt | | | | | IF | ID | EX | M | WB | |
| e. | add | IF | ID | EX | M | WB | | | | | |
| | add | | IF | ID | EX | M | WB | | | | |
| | nand | IF | ID | EX | M | WB | | | | | |
| | nand | | IF | ID | EX | M | WB | | | | |
| | add | | | IF | ID | EX | M | WB | | | |
| | lw | | | IF | ID | EX | M | WB | | | |
| | add | | | | | IF | ID | EX | M | WB | |
| | halt | | | | | IF | ID | EX | M | WB | |

## Pipelining II

12. [10 points] The following figure shows a simple pipeline. This pipeline has a register file that WRITES during the FIRST half of the clock cycle and READS during the SECOND half of the clock cycle. Instructions can be stalled only in the Fetch and Decode stages. There is a forwarding path from the output of the Ex|M register to the input of the logic in the Ex stage.




The following sequence of instructions is executed on this pipeline.

```
$r3 = $r2 + $r1                 // ADD1
$r2 = Mem($r3 + $r1)            // LOAD
$r2 = $r2 + $r1                 // ADD2
$r4 = $r3 - $r1                 // SUB
$r1 = $r2 + $r1                 // ADD3
```

Due to data dependencies, some pipeline registers will have to be stalled, and bubbles would be inserted in the execution stages in certain cycles. From the following set of tables, choose the one which represents a correct execution sequence.

(Note: F=fetch, D=decode, Ex=execute, M=memory access, W=write back. The numbers in the first row indicate the clock cycle.)

a.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADD1 | F | D | Ex | M | W | | | | | | | | |
| LOAD | | F | D | Ex | M | W | | | | | | | |
| ADD2 | | | F | D | D | D | Ex | M | W | | | | |
| SUB | | | | F | F | F | D | Ex | M | W | | | |
| ADD3 | | | | | | | F | D | D | Ex | M | W | |

b.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADD1 | F | D | Ex | M | W | | | | | | | | |
| LOAD | | F | D | D | Ex | M | W | | | | | | |
| ADD2 | | | F | D | D | D | Ex | M | W | | | | |
| SUB | | | | F | F | F | D | D | Ex | M | W | | |
| ADD3 | | | | | | | F | D | D | Ex | M | W | |

c.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADD1 | F | D | Ex | M | W | | | | | | | | |
| LOAD | | F | D | Ex | M | W | | | | | | | |
| ADD2 | | | F | D | D | Ex | M | W | | | | | |
| SUB | | | | F | F | D | Ex | M | W | | | | |
| ADD3 | | | | | | F | D | D | Ex | M | W | | |

d.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADD1 | F | D | Ex | M | W | | | | | | | | |
| LOAD | | F | D | Ex | M | W | | | | | | | |
| ADD2 | | | F | D | D | D | Ex | M | W | | | | |
| SUB | | | | F | F | F | D | Ex | M | W | | | |
| ADD3 | | | | | | | F | F | D | D | Ex | M | W |

e.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADD1 | F | D | Ex | M | W | | | | | | | | |
| LOAD | | F | D | Ex | M | W | | | | | | | |
| ADD2 | | | F | D | D | Ex | M | W | | | | | |
| SUB | | | | F | F | F | D | Ex | M | W | | | |
| ADD3 | | | | | | | F | D | D | Ex | M | W | |

13. [5 points] Now assume that the register file is modified so that READS occur in the FIRST half of the clock cycle and WRITES occur in the SECOND half of the clock cycle. What additional forwarding path would be required for correct execution, without any additional stalls?

    a. Output of M|W register to the input of the D|Ex register.
    b. Output of M|W register to the input of the F|D register.
    c. Output of Ex|M register to the input of the D|Ex register.
    d. Output of Ex|M register to the input of the F|D register.
    e. Output of M|W register to the input of the Ex|M register.

14. [5 points] Refer back to the original datapath in Question 12. Assume that a forwarding path is introduced from the output of the M|W register to the inputs of the ALU with suitable MUXes. From the following five options, select the correct reordered instruction sequence that eliminates **all** bubbles.

    a. LOAD
       ADD1
       SUB
       ADD2
       ADD3

    b. ADD1
       ADD2
       SUB
       LOAD
       ADD3

    c. ADD1
       SUB
       LOAD
       ADD2
       ADD3

    d. LOAD
       ADD1
       ADD2
       SUB
       ADD3

    e. ADD1
       LOAD
       SUB
       ADD2
       ADD3

## Caches

15. [5 points] Consider a program in which 20% of the instructions are memory loads/stores. Assume a main memory access time of 50 nsec, a clock rate of 500MHz, and a CPI of 2 for a machine with a perfect cache (i.e., data and instructions are always found in first-level cache). What is the CPI if instruction cache miss rate is 3% and data cache miss rate is 10%?

    a. 1
    b. 3.25
    c. 2.5
    d. 2.75
    e. 1.5

For Questions 16 and 17 assume you are given a cache with a block size of 32 bytes, a 32-bit address, and a data capacity of 8K bytes:

16. [5 points] If the cache is direct-mapped, how many sets are in this cache?

    a. 256
    b. 128
    c. 64
    d. 32
    e. 20

17. [5 points] If the cache is 2-way set-associative, how many bits are in the tag?

    a. 16
    b. 18
    c. 19
    d. 20
    e. 21