Lab_CS322_1_December 2020

Total points 53/100

Email address *

maheeth2000@gmail.com

Roll Number *

1801CS31

Name *

Maheeth Reddy

✓ When the following code is run ; the value of fun(0) is 1/1

```
typedef struct {
   int a[2];
   double d;
} struct_t;

double fun(int i) {
   volatile struct_t s;
   s.d = 3.14;
   s.a[i] = 1073741824;
   return s.d;
}
```

- (●) 3.14 ✓
- ( ) Machine Specific (random value)
- ( ) Segmentation fault
- ( ) None of these
- ( ) Other: _____

✓ When the following code is run ; the value of fun(1) is 1/1

```
typedef struct {
   int a[2];
   double d;
} struct_t;

double fun(int i) {
   volatile struct_t s;
   s.d = 3.14;
   s.a[i] = 1073741824;
   return s.d;
}
```

- (●) 3.14 ✓
- ( ) Machine Specific (random value)
- ( ) Segmentation fault
- ( ) None of these

✗ When the following code is run ; the value of fun(2) is 0/1

```
typedef struct {
   int a[2];
   double d;
} struct_t;

double fun(int i) {
   volatile struct_t s;
   s.d = 3.14;
   s.a[i] = 1073741824;
   return s.d;
}
```

- (●) 3.14 ✗
- ( ) Machine Specific (random value)
- ( ) Segmentation fault
- ( ) None of these

✓ When the following code is run ; the value of fun(3) is 1/1

```
typedef struct {
   int a[2];
   double d;
} struct_t;

double fun(int i) {
   volatile struct_t s;
   s.d = 3.14;
   s.a[i] = 1073741824;
   return s.d;
}
```

- ○ 3.14
- ◉ Machine Specific (random value) ✓
- ○ Segmentation fault
- ○ None of these

✓ When the following code is run ; the value of fun(6) is 1/1

```
typedef struct {
   int a[2];
   double d;
} struct_t;

double fun(int i) {
   volatile struct_t s;
   s.d = 3.14;
   s.a[i] = 1073741824;
   return s.d;
}
```

- ○ 3.14
- ○ Machine Specific (random value)
- ◉ Segmentation fault ✓
- ○ None of these

✗ Microarchitecture: Implementation of the architecture. One of the examples is 0/1

- ◉ The byte-level programs that a processor executes ✗
- ○ A text representation of machine code
- ○ cache sizes and core frequency
- ○ instruction set specification

✓ Give answer as (XXH) 2/2

The following program start at location 0100H.
LXI SP, 00FF
LXI H, 0107
MVI A, 20H
SUB M
The content of accumulator when the program counter reaches 0109H is:

00H ✓

✓ Give answer as (XXH) 1/1

LXI SP, 00FF
LXI H, 0107
MVI A, 20H
SUB M
ORI 40H
ADD M
What will be the result in the accumulator after the last instruction is executed?

60H ✓

✓ In an 8086 Microprocessor, given that the IP = 1230H and CS = 0100H What is the physical address of the code. 1/1

- 1230H
- 1330H
- 22300
- 02230 ✓

✓ Give answer as (XXH) 2/2

```
1          MVI  B,87H
2          MOV  A, B
3  START:  JMP  NEXT
4          MVI  B, 00H
5          XRA  B
6          OUT  PORT 1
7          HLT
8  NEXT:   XRA  B
9          JP   START
10         OUT  PORT 2
11         HLT
```

What is the output at PORT 2

87H ✓

✗ Give answer as (XXXps) ···/1

The multi-cycle has been broken down into 5 steps:
1. Hardware to support an instruction fetch
2. Hardware to support an instruction decode (i.e. a register file read)
3. Hardware to support instruction execution (i.e. the ALU)
4. Hardware to support a memory load or store
5. Hardware to support the write back of the ALU operation back to the register file

Assume that each of the above steps takes the amount of time specified in the table below.
(Fetch:305ps; Decode: 275ps; Execute:280ps; memory:305ps; Write Back:250ps)

Given the times for the datapath stages listed above, what would the *clock period* be for the entire datapath?

780ps ✗

✗ At the end of program, register A contains (Progream starts at address 200H) 0/1

In an 8085 microprocessor, the following program is executed

Address location – Instruction

2000HXRA A
2001HMVI B,04H
2003HMVI A, 03H
2005HRAR
2006HDCR B
2007HJNZ 2005
200AHHLT
At the end of program, register A contains

- ○ 60H
- ◉ 30H ✗
- ○ 06H
- ○ 03H

✓ The set of instructions that precede the RET instruction in the subroutine are 2/2

A portion of the main program to call a subroutine SUB in an 8085 environment is given below.

```
:
:
LXI D,DISP
LP : CALL SUB
:
```

It is desired that control be returned to LP+DISP+3 when the RET instruction is executed in the subroutine. The set of instructions that precede the RET instruction in the subroutine are

- ○ POP D DAD H PUSH D
- ○ POP H DAD D INX H INX H INX H PUSH H
- ◉ POP H DAD D PUSH H ✓
- ○ XTHL INX D INX D INX D XTHL

✓ In a x86-32 architecture pointers point to locations in memory that are multiples of 32 bits apart. This statement is 1/1

- ○ True
- ◉ False ✓

✓ X86 assembly store the return value is always in ..... when a function is finished 2/2

- ○ ebx
- ◉ eax ✓
- ○ ecx
- ○ esp
- ○ Other: _____

✓ A RET instruction is equivalent to 1/1

- ○ pop sp
- ◉ pop ip ✓
- ○ pop a
- ○ push sp

✗ Reading memory from the heap is slower than reading from a local variable allocated on the stack. This statement is 0/2

- (●) True ✗
- ( ) False

✓ On a write hit, a cache that is write-back will immediately write a value from the cache back to memory 2/2

- ( ) True
- (●) False ✓

✓ We use dynamic memory because storing data on the stack requires knowing the size of that data at compile time 2/2

- (●) True ✓
- ( ) False

✓ The two address lines, along with CS signal, determine the selection of a particular port or control register in an 8255. 1/1

- (●) True ✓
- ( ) False

✓ In a x86-64 architecture pointers point to locations in memory that are multiples of 64 bits apart. 1/1

- ○ True
- ◉ False ✓

✓ Assuming the following dynamic instruction frequency for a program running on the single-cycle MIPS processor Add- 25%; addi- 25%; beq- 10%; lw-25%; sw- 15% In what fraction of all cycles is the data memory accessed (either read or written)? Give answer XX% 2/2

40% ✓

✗ Assuming the following dynamic instruction frequency for a program running on the single-cycle MIPS processor Add- 25%; addi- 25%; beq- 10%; lw-25%; sw- 15% In what fraction of cycles is the sign extend circuit used?? Give answer XX% ···/3

65% ✗

✕ GIve answer as XXXXps ···/1

The multi-cycle and pipelined datapaths have been broken down into 5 steps:
1. Hardware to support an instruction fetch
2. Hardware to support an instruction decode (i.e. a register file read)
3. Hardware to support instruction execution (i.e. the ALU)
4. Hardware to support a memory load or store
5. Hardware to support the write back of the ALU operation back to the register file

Assume that each of the above steps takes the amount of time specified in the table below.

Fetch	Decode	Execute	Memory	Write Back
305 ps	275 ps	280 ps	305 ps	250 ps

In a pipelined datapath, assuming no hazards or stalls, how many seconds will it take to execute 1 instruction?

1200ps ✕

✕ For non-pipelined processor: what is the cycle time for processor (a)? (Give answer in the format: XXXXps) ···/1

The 5 stages of the processor have the following latencies:

	Fetch	Decode	Execute	Memory	Writeback
a.	300ps	400ps	350ps	550ps	100ps
b.	200ps	150ps	100ps	190ps	140ps

Assume that when pipelining, each pipeline stage costs 20 ps extra for the registers between pipeline stages.

1650ps ✕

✗ Pipelined processor: What is the cycle time for processor (a)? (Give answer in the format: XXXXps) ···/2

The 5 stages of the processor have the following latencies:

	Fetch	Decode	Execute	Memory	Writeback
a.	300ps	400ps	350ps	550ps	100ps
b.	200ps	150ps	100ps	190ps	140ps

Assume that when pipelining, each pipeline stage costs 20 ps extra for the registers between pipeline stages.

0520ps ✗

✗ For non-pipelined processor: what is the cycle time for processor (b)? (Give answer in the format: XXXps) ···/1

The 5 stages of the processor have the following latencies:

	Fetch	Decode	Execute	Memory	Writeback
a.	300ps	400ps	350ps	550ps	100ps
b.	200ps	150ps	100ps	190ps	140ps

Assume that when pipelining, each pipeline stage costs 20 ps extra for the registers between pipeline stages.

800ps ✗

✓ For pipelined processor: what is the cycle time for processor (b)? (Give answer in the format: XXXps) 1/1

The 5 stages of the processor have the following latencies:

	Fetch	Decode	Execute	Memory	Writeback
a.	300ps	400ps	350ps	550ps	100ps
b.	200ps	150ps	100ps	190ps	140ps

Assume that when pipelining, each pipeline stage costs 20 ps extra for the registers between pipeline stages.

220ps ✓

✗ Consider the sequence of 8085 instructions given below:LXI H, 9258; MOV A, M; CMA; MOV M, A; which one of the following is performed by this sequence? 0/3

- ○ Contents of location 9258 are moved to the accumulator
- ◉ Contents of location 9258 are compared with the contents of accumulator ✗
- ○ Contents of location 9258 are complemented and stored in location 9258
- ○ Contents of location 5892 are complemented and stored in location 5892

✗ SI = displacement of 5th element 0/3

```
table1 DW 10 DUP (0)
```

fill in the blanks in the following code:

```
mov SI, _______ ; SI = displacement of 5th element
```

- ○ 5
- ◉ 4 ✗
- ○ 8
- ○ 10
- ○ Other: _____

✓ compare 5th and 4th elements (fill the second blank). Assume prior instructions are executed. 3/3

Given the following declaration of table1

```
table1 DW 10 DUP (0)
```

fill in the blanks in the following code:

```
mov SI, _______ ; SI = displacement of 5th element

mov AX,table1[SI]
cmp AX, _______ ; compare 5th and 4th elements
```

- ○ cmp AX, [SI-6]
- ○ cmp AX, [SI-4]
- ○ cmp AX, [SI-5]
- ◉ cmp AX, [SI-2] ✓

✗ The time delay when the routine is called 0/3

Given the following delay subroutine, determine the time delay when the routine is called.

```
Delay:  mov AX, 0x20
        mov CX,AX
Repeat: dec CX
        jz repeat
        iret
```

Assume that no. of clk cycles as follows: mov (1 cycle); dec (1, 2 if the result is 0); And goto, iret (2). The processor operates at 1 MHz.

- ○ 99us
- ◉ 90us ✗
- ○ 110us
- ○ 200us

✕ Answer for Q1: legal/ illegal 0/1

Assume that

j DW 10 ; i DW 20; Check the following instructions legal/ illegal Complete the table. Answer legal/ illegal .

	instruction	comment
1	MOV AX, BL	Q1
2	MOV AL, BL	Q2

6

3	MOV AH, BL	Q3
4	MOV j, AL	Q4
5	MOV AL, j	Q5
6	MOV j, j	Q6
7	ADD 2, AX	Q7
8	ADD AX, 2	Q8
9	MOV AL, j	Q9
10	MOV AL, Byte PTR j	Q10

◉ legal ✕

○ illegal

✓ Answer for Q2: legal/ illegal 1/1

Assume that

j DW 10 ; i DW 20; Check the following instructions legal/ illegal Complete the table. Answer legal/ illegal .

	instruction	comment
1	MOV AX, BL	Q1
2	MOV AL, BL	Q2

6

3	MOV AH, BL	Q3
4	MOV j, AL	Q4
5	MOV AL, j	Q5
6	MOV j, j	Q6
7	ADD 2, AX	Q7
8	ADD AX, 2	Q8
9	MOV AL, j	Q9
10	MOV AL, Byte PTR j	Q10

- (●) legal ✓
- ( ) illegal

✓ Answer for Q3: legal/ illegal 1/1

Assume that

j DW 10 ; i DW 20; Check the following instructions legal/ illegal Complete the table. Answer legal/ illegal .

	instruction	comment
1	MOV AX, BL	Q1
2	MOV AL, BL	Q2

6

3	MOV AH, BL	Q3
4	MOV j, AL	Q4
5	MOV AL, j	Q5
6	MOV j, j	Q6
7	ADD 2, AX	Q7
8	ADD AX, 2	Q8
9	MOV AL, j	Q9
10	MOV AL, Byte PTR j	Q10

- (●) legal ✓
- ( ) illegal

✓ Answer for Q4: legal/ illegal 1/1

Assume that

j DW 10 ; i DW 20; Check the following instructions legal/ illegal Complete the table. Answer legal/ illegal .

	instruction	comment
1	MOV AX, BL	Q1
2	MOV AL, BL	Q2

6

3	MOV AH, BL	Q3
4	MOV j, AL	Q4
5	MOV AL, j	Q5
6	MOV j, j	Q6
7	ADD 2, AX	Q7
8	ADD AX, 2	Q8
9	MOV AL, j	Q9
10	MOV AL, Byte PTR j	Q10

- (●) legal ✓
- ( ) illegal

✕ Answer for Q5: legal/ illegal 0/1

Assume that

j DW 10 ; i DW 20; Check the following instructions legal/ illegal Complete the table. Answer legal/ illegal .

	instruction	comment
1	MOV AX, BL	Q1
2	MOV AL, BL	Q2

6

3	MOV AH, BL	Q3
4	MOV j, AL	Q4
5	MOV AL, j	Q5
6	MOV j, j	Q6
7	ADD 2, AX	Q7
8	ADD AX, 2	Q8
9	MOV AL, j	Q9
10	MOV AL, Byte PTR j	Q10

- ○ legal
- ◉ illegal ✕

✓ Answer for Q6: legal/ illegal 1/1

Assume that

j DW 10 ; i DW 20; Check the following instructions legal/ illegal Complete the table. Answer legal/ illegal .

	instruction	comment
1	MOV AX, BL	Q1
2	MOV AL, BL	Q2

6

3	MOV AH, BL	Q3
4	MOV j, AL	Q4
5	MOV AL, j	Q5
6	MOV j, j	Q6
7	ADD 2, AX	Q7
8	ADD AX, 2	Q8
9	MOV AL, j	Q9
10	MOV AL, Byte PTR j	Q10

- ○ legal
- ◉ illegal ✓

✓ Answer for Q7: legal/ illegal 1/1

Assume that

j DW 10 ; i DW 20; Check the following instructions legal/ illegal Complete the table. Answer legal/ illegal .

	instruction	comment
1	MOV AX, BL	Q1
2	MOV AL, BL	Q2

6

3	MOV AH, BL	Q3
4	MOV j, AL	Q4
5	MOV AL, j	Q5
6	MOV j, j	Q6
7	ADD 2, AX	Q7
8	ADD AX, 2	Q8
9	MOV AL, j	Q9
10	MOV AL, Byte PTR j	Q10

- ○ legal
- ◉ illegal ✓

✓ Answer for Q8: legal/ illegal 1/1

Assume that

j DW 10 ; i DW 20; Check the following instructions legal/ illegal Complete the table. Answer legal/ illegal .

	instruction	comment
1	MOV AX, BL	Q1
2	MOV AL, BL	Q2

6

3	MOV AH, BL	Q3
4	MOV j, AL	Q4
5	MOV AL, j	Q5
6	MOV j, j	Q6
7	ADD 2, AX	Q7
8	ADD AX, 2	Q8
9	MOV AL, j	Q9
10	MOV AL, Byte PTR j	Q10

- (●) legal ✓
- ( ) illegal

✓ Answer for Q9: legal/ illegal 1/1

Assume that

j DW 10 ; i DW 20; Check the following instructions legal/ illegal Complete the table. Answer legal/ illegal .

	instruction	comment
1	MOV AX, BL	Q1
2	MOV AL, BL	Q2

6

3	MOV AH, BL	Q3
4	MOV j, AL	Q4
5	MOV AL, j	Q5
6	MOV j, j	Q6
7	ADD 2, AX	Q7
8	ADD AX, 2	Q8
9	MOV AL, j	Q9
10	MOV AL, Byte PTR j	Q10

- ○ legal
- ◉ illegal ✓

✓ Answer for Q10: legal/ illegal 1/1

Assume that

j DW 10 ; i DW 20; Check the following instructions legal/ illegal Complete the table. Answer legal/ illegal .

	instruction	comment
1	MOV AX, BL	Q1
2	MOV AL, BL	Q2

6

3	MOV AH, BL	Q3
4	MOV j, AL	Q4
5	MOV AL, j	Q5
6	MOV j, j	Q6
7	ADD 2, AX	Q7
8	ADD AX, 2	Q8
9	MOV AL, j	Q9
10	MOV AL, Byte PTR j	Q10

- (●) legal ✓
- ( ) illegal

✓ question a: What is the value in AX after this instruction? 2/2

Refer to this code fragment to answer the following questions:

```
.data
stuff dw 4 dup (?)
var1 dw 0
var2 dw 1
var3 dw 23Fh
inx dw ?
string db 'ABCD'
array db 0,1,2,3,4,5,6,7
db 8,9,10,11,12,13,14,15

.code
        mov ax,seg @data
        mov ds,ax
        mov ax,var2             ;question a
        mov bx,ax
        mov al, string[bx]      ;question b, c
        mov cx,1122h            ;question d
        mov dl, cl
        mov dh,ah
        mov var1,dx             ;question e, f
        mov ah, dl
        mov array[8],ax
        leasi, array[bx+2]      ;question g
        mov al,[si]             ;question h
ends
```

- (●) 0001h ✓
- ( ) 0042h
- ( ) 0011h
- ( ) all of the above

✓ question b: What is the value in AX after this instruction? 2/2

Refer to this code fragment to answer the following questions:

```
.data
stuff dw 4 dup (?)
var1 dw 0
var2 dw 1
var3 dw 23Fh
inx dw ?
string db 'ABCD'
array db 0,1,2,3,4,5,6,7
db 8,9,10,11,12,13,14,15

.code
        mov ax,seg @data
        mov ds,ax
        mov ax,var2             ;question a
        mov bx,ax
        mov al, string[bx]      ;question b, c
        mov cx,1122h            ;question d
        mov dl, cl
        mov dh,ah
        mov var1,dx             ;question e, f
        mov ah, dl
        mov array[8],ax
        leasi, array[bx+2]      ;question g
        mov al,[si]             ;question h
ends
```

- ○ 0001h
- ◉ 0042h ✓
- ○ 1042h
- ○ None of the above

✗ question c: What is the value in AX after this instruction? 0/1

Refer to this code fragment to answer the following questions:

```
.data
stuff dw 4 dup (?)
var1 dw 0
var2 dw 1
var3 dw 23Fh
inx dw ?
string db 'ABCD'
array db 0,1,2,3,4,5,6,7
db 8,9,10,11,12,13,14,15

.code
        mov ax,seg @data
        mov ds,ax
        mov ax,var2             ;question a
        mov bx,ax
        mov al, string[bx]      ;question b, c
        mov cx,1122h            ;question d
        mov dl, cl
        mov dh,ah
        mov var1,dx             ;question e, f
        mov ah, dl
        mov array[8],ax
        leasi, array[bx+2]      ;question g
        mov al,[si]             ;question h
ends
```

- ○ 0001h
- ◉ 0042h ✗
- ○ 0011h
- ○ None of the above

✓ question d: What is the effective address (offset) used for the memory address in this instruction? 2/2

Refer to this code fragment to answer the following questions:

```
.data
stuff dw 4 dup (?)
var1 dw 0
var2 dw 1
var3 dw 23Fh
inx dw ?
string db 'ABCD'
array db 0,1,2,3,4,5,6,7
db 8,9,10,11,12,13,14,15

.code
        mov ax,seg @data
        mov ds,ax
        mov ax,var2             ;question a
        mov bx,ax
        mov al, string[bx]      ;question b, c
        mov cx,1122h            ;question d
        mov dl, cl
        mov dh,ah
        mov var1,dx             ;question e, f
        mov ah, dl
        mov array[8],ax
        leasi, array[bx+2]      ;question g
        mov al,[si]             ;question h
ends
```

- ○ 01h
- ◉ 11h ✓
- ○ 10h
- ○ None of the above

✗ question e: What value will be stored into memory by this instruction? 0/1

Refer to this code fragment to answer the following questions:

```
.data
stuff dw 4 dup (?)
var1 dw 0
var2 dw 1
var3 dw 23Fh
inx dw ?
string db 'ABCD'
array db 0,1,2,3,4,5,6,7
db 8,9,10,11,12,13,14,15

.code
        mov ax,seg @data
        mov ds,ax
        mov ax,var2             ;question a
        mov bx,ax
        mov al, string[bx]      ;question b, c
        mov cx,1122h            ;question d
        mov dl, cl
        mov dh,ah
        mov var1,dx             ;question e, f
        mov ah, dl
        mov array[8],ax
        leasi, array[bx+2]      ;question g
        mov al,[si]             ;question h
ends
```

- ○ 0001h
- ◉ 0002h ✗
- ○ 2222
- ○ 0022h

✓ question f: What is the effective address (offset) used for the memory address in this instruction? 2/2

Refer to this code fragment to answer the following questions:

```
.data
stuff dw 4 dup (?)
var1 dw 0
var2 dw 1
var3 dw 23Fh
inx dw ?
string db 'ABCD'
array db 0,1,2,3,4,5,6,7
db 8,9,10,11,12,13,14,15

.code
        mov ax,seg @data
        mov ds,ax
        mov ax,var2             ;question a
        mov bx,ax
        mov al, string[bx]      ;question b, c
        mov cx,1122h            ;question d
        mov dl, cl
        mov dh,ah
        mov var1,dx             ;question e, f
        mov ah, dl
        mov array[8],ax
        leasi, array[bx+2]      ;question g
        mov al,[si]             ;question h
ends
```

- ○ 0007h
- ◉ 0008h ✓
- ○ 000Ah
- ○ 000Bh

✗ question g: What is the value in SI after this instruction? 0/1

Refer to this code fragment to answer the following questions:

```
.data
stuff dw 4 dup (?)
var1 dw 0
var2 dw 1
var3 dw 23Fh
inx dw ?
string db 'ABCD'
array db 0,1,2,3,4,5,6,7
db 8,9,10,11,12,13,14,15

.code
        mov ax,seg @data
        mov ds,ax
        mov ax,var2             ;question a
        mov bx,ax
        mov al, string[bx]      ;question b, c
        mov cx,1122h            ;question d
        mov dl, cl
        mov dh,ah
        mov var1,dx             ;question e, f
        mov ah, dl
        mov array[8],ax
        leasi, array[bx+2]      ;question g
        mov al,[si]             ;question h
ends
```

- ○ 0017h
- ◉ 0027h ✗
- ○ 0018h
- ○ 0019h

✕ question h: What is the value in AX after this instruction? 0/3

Refer to this code fragment to answer the following questions:

```
.data
stuff dw 4 dup (?)
var1 dw 0
var2 dw 1
var3 dw 23Fh
inx dw ?
string db 'ABCD'
array db 0,1,2,3,4,5,6,7
db 8,9,10,11,12,13,14,15

.code
        mov ax,seg @data
        mov ds,ax
        mov ax,var2             ;question a
        mov bx,ax
        mov al, string[bx]      ;question b, c
        mov cx,1122h            ;question d
        mov dl, cl
        mov dh,ah
        mov var1,dx             ;question e, f
        mov ah, dl
        mov array[8],ax
        leasi, array[bx+2]      ;question g
        mov al,[si]             ;question h
ends
```

- ○ 2204h
- ◉ 2200h ✕
- ○ 2201h
- ○ None of the above

✓ question a: What is the value in AX after this instruction? 1/1

Refer to this code fragment to answer the following questions:

```
.data
stuff dw 4 dup (?)
var1 dw 0
var2 dw 1
var3 dw 23Fh
inx dw ?
string db 'ABCD'
array db 0,1,2,3,4,5,6,7
db 8,9,10,11,12,13,14,15

.code
        mov ax,seg @data
        mov ds,ax
        mov ax,var2             ;question a
        mov bx,ax
        mov al, string[bx]      ;question b, c
        mov cx,1122h            ;question d
        mov dl, cl
        mov dh,ah
        mov var1,dx             ;question e, f
        mov ah, dl
        mov array[8],ax
        leasi, array[bx+2]      ;question g
        mov al,[si]             ;question h
ends
```

- ○ R-type
- ◉ sw ✓
- ○ addi
- ○ None of the above

✓ Opcode part of MIPS instruction has x bits (give answer in decimal) 1/1

6 ✓

✓ The value of Q1 1/1

For the following sequence of MIPS instructions , the values registers after the code was executed.

```
li $t0,4
li $t1,7
li t2, 3
sub $t3, $t1, $t2
beq $to, $t3, next
add $S0, $zeo, $t3
j end

next: add $s0, $t1, $t2
end:
```

Register Name	t0	t1	t2	t3	s0
Register Value	Q1	Q2	Q3	Q4	Q5

- ○ 10
- ◉ 4 ✓
- ○ 5
- ○ None of these
- ○ Other: _____

✓ The value of Q2 1/1

For the following sequence of MIPS instructions , the values registers after the code was executed.

```
li $t0,4
li $t1,7
li t2, 3
sub $t3, $t1, $t2
beq $to, $t3, next
add $S0, $zeo, $t3
j end

next: add $s0, $t1, $t2
end:
```

Register Name	t0	t1	t2	t3	s0
Register Value	Q1	Q2	Q3	Q4	Q5

- ○ 10
- ○ 4
- ◉ 7 ✓
- ○ None of these
- ○ Other: _____

✕ The value of Q4 0/1

For the following sequence of MIPS instructions , the values registers after the code was executed.

```
li $t0,4
li $t1,7
li t2, 3
sub $t3, $t1, $t2
beq $to, $t3, next
add $S0, $zeo, $t3
j end

next: add $s0, $t1, $t2
end:
```

Register Name	t0	t1	t2	t3	s0
Register Value	Q1	Q2	Q3	Q4	Q5

- ○ 10
- ◉ 4 ✕
- ○ 3
- ○ None of these
- ○ Other: _____

✓ The value of Q4 1/1

For the following sequence of MIPS instructions , the values registers after the code was executed.

```
li $t0,4
li $t1,7
li t2, 3
sub $t3, $t1, $t2
beq $to, $t3, next
add $S0, $zeo, $t3
j end

next: add $s0, $t1, $t2
end:
```

Register Name	t0	t1	t2	t3	s0
Register Value	Q1	Q2	Q3	Q4	Q5

- ○ 10
- ◉ 4 ✓
- ○ 3
- ○ None of these
- ○ Other: _____

✓ The value of Q5 1/1

For the following sequence of MIPS instructions , the values registers after the code was executed.

```
li $t0,4
li $t1,7
li t2, 3
sub $t3, $t1, $t2
beq $to, $t3, next
add $S0, $zeo, $t3
j end
next: add $s0, $t1, $t2
end:
```

Register Name	t0	t1	t2	t3	s0
Register Value	Q1	Q2	Q3	Q4	Q5

- (●) 10 ✓
- ( ) 4
- ( ) 3
- ( ) None of these

✗ Suppose one of the following control signals in the multicycle MIPSprocessor has a stuck-at-1 fault, meaning that the signal is always 1, regardless of its intended value. What instructions would malfunction? For Signal: ALUOp0 0/3

- ( ) R-type, addi
- ( ) lw, sw, addi, R-type
- ( ) lw, sw, addi,
- (●) All instructions ✗

✗ Suppose one of the following control signals in the multicycle MIPSprocessor has a stuck-at-1 fault, meaning that the signal is always 1, regardless of its intended value. What instructions would malfunction? For Signal: MemtoReg 0/3

- ○ R-type, addi
- ○ addi only
- ○ lw
- ◉ All instructions ✗

✓ Suppose one of the following control signals in the multicycle MIPSprocessor has a stuck-at-1 fault, meaning that the signal is always 1, regardless of its intended value. What instructions would malfunction? For Signal: PCSrc 3/3

- ○ R-type, addi
- ○ addi only
- ○ lw
- ◉ All instructions ✓

✕ Your friend, the crack circuit designer, has offered to redesign one of the units in the multicycle MIPS processor to be much faster. Using the delays from Table, which unit should she work on to obtain the greatest speedup of the overall processor? 0/3

Element	Parameter	Delay (ps)
register clk-to-Q	t_{pcq}	30
register setup	t_{setup}	20
multiplexer	t_{mux}	25
ALU	t_{ALU}	200
memory read	t_{mem}	250
register file read	t_{RFread}	150
register file setup	$t_{RFsetup}$	20

- ○ register file
- ○ ALU
- ○ All parts
- ◉ memory ✕

✕ Your friend, the crack circuit designer, has offered to redesign one of the··· /3
units in the multicycle MIPS processor to be much faster. Using the delays from Table, which unit should she work on to obtain the greatest speedup of the overall processor? How fast should it be?

Element	Parameter	Delay (ps)
register clk-to-Q	t_{pcq}	30
register setup	t_{setup}	20
multiplexer	t_{mux}	25
ALU	t_{ALU}	200
memory read	t_{mem}	250
register file read	t_{RFread}	150
register file setup	$t_{RFsetup}$	20

100 ✕

✗ Suppose the multicycle MIPS processor has the component delaysgiven in Table . P. Hacker designs a new register file that has 40% less power but twice as much delay. Should she switch to the slower but lower power register file for her multicycle processor design? 0/3

Element	Parameter	Delay (ps)
register clk-to-Q	t_{pcq}	30
register setup	t_{setup}	20
multiplexer	t_{mux}	25
ALU	t_{ALU}	200
memory read	t_{mem}	250
register file read	t_{RFread}	150
register file setup	$t_{RFsetup}$	20

- ○ No
- ○ Yes
- ◉ It does not matter ✗
- ○ Other: _____