

الرقم التسلسلي:

رقم التسجيل:

الاسم:

Instructions: Time 70 minutes. Open book and notes exam. No electronics. Please answer all problems in the space provided and limit your answer to the space provided. Numbers are in hexadecimal. **No questions are allowed.**

<Good Luck>

Q1. At the end of executing the following MIPS instruction sequence, specify the contents of the following registers.

<5 points>

```
lui $s0, 0x3232
ori $s1, $s0, 0x2323
addi $t0, $zero, 4
slt $s2, $t0, $s0
sll $s3, $s2, 3
sw $t0, 0($s1)
lw $s4, 0($s1)
```

Register \$s0 = 0x32320000Register \$s1 = 0x32322323Register \$s2 = 1Register \$s3 = 8Register \$s4 = 4

Q2. For the following C language loop, what is the corresponding MIPS assembly code? Assume that all variables are one-word signed integers. Also assume that the compiler maps *i* to \$s0 and maps the starting address of Array A to Register \$s1.

<5 points>

```
for (i=9; i>=0; i--)
    A[i] = 0;
```

```
addi $s0, $zero, 9
Loop:
    slt $t0, $s0, $zero
    bne $t0, $zero, End
    sll $t1, $s0, 2
    addu $t1, $t1, $s1
    sw $zero, 0($t1)
    addi $s0, $s0, -1
    j Loop
End:
```

Q3. The following screen capture shows the QtSpim after loading a simple MIPS program and execution to the point shown.

<5 points>

```

QtSpim
File Simulator Registers Text Segment Data Segment Window Help
[Icons]
Int Regs [16] Text
PC = 400024
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000ff10
HI = 0
LO = 0
R0 [r0] = 0
R1 [at] = 0
R2 [v0] = 4
R3 [v1] = 0
R4 [a0] = 1
R5 [a1] = 7ffff730
R6 [a2] = 7ffff738
R7 [a3] = 0
User Text Segment [00400000]..[00440000]
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) # a1
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4 #
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4 #
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # syscall
[00400024] 20100002 addi $16, $0, 2 ; 7: addi $s0, $zero, 2
[00400028] 20110003 addi $17, $0, 3 ; 8: addi $s1, $zero, 3
[0040002c] 02309020 add $18, $17, $16 ; 9: add $s2, $s1, $s0
[00400030] 02309824 and $19, $17, $16 ; 10: and $s3, $s1, $s0
[00400034] 02110018 mult $16, $17 ; 11: mult $s0, $s1
[00400038] 0810000e j 0x00400038 [end] ; 13: j end
  
```

Specify the contents of the following registers after executing the “Single Step” function five times.

Register \$s2= 5

Register \$s3= 2

Register HI = 0

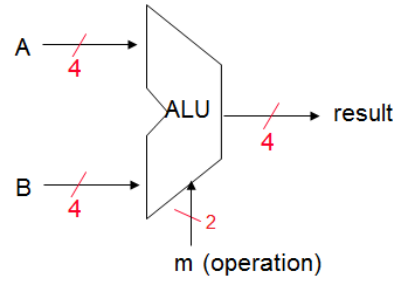
Register LO = 6

Register PC = 0x00400038

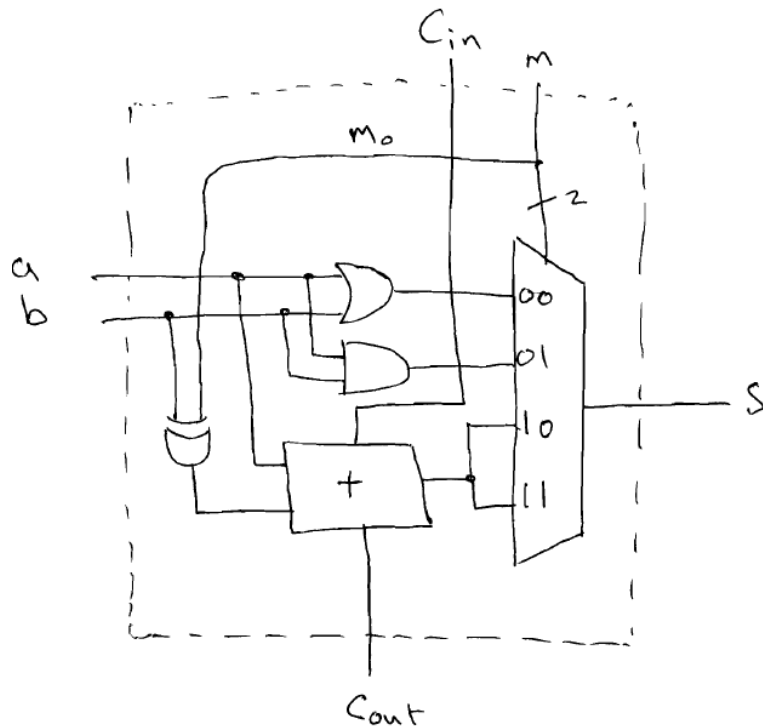
Q4. It is required to design a 4-bit ALU that can perform the operations specified in the following table. This ALU has the interface specified to the right.

<10 marks>

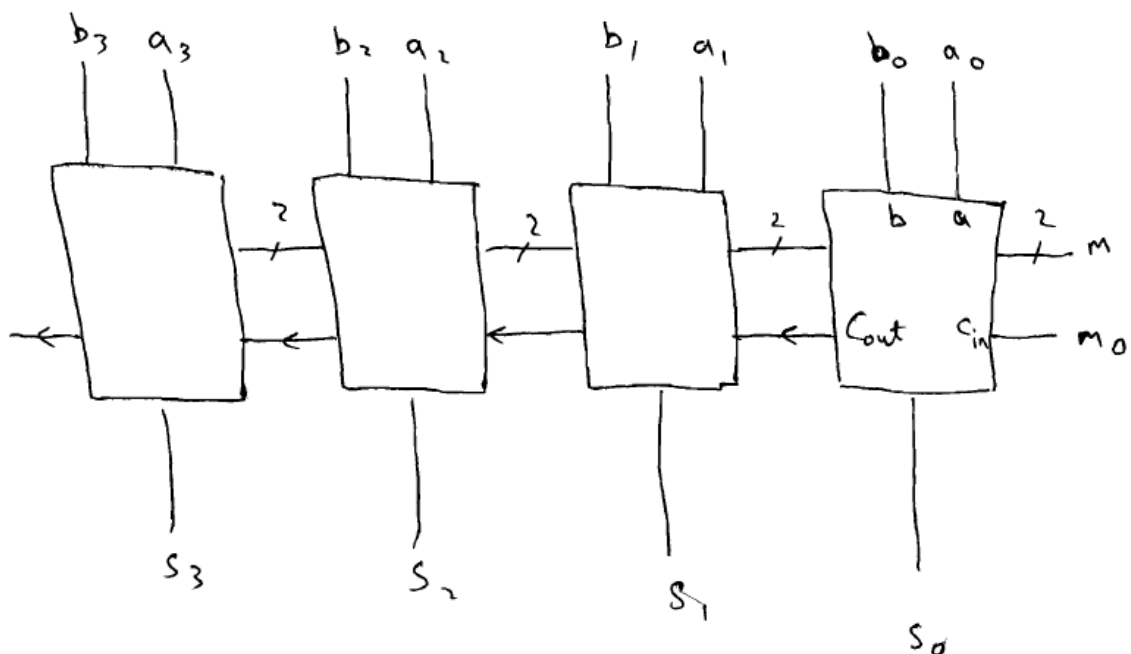
Operation	Function
00	or
01	and
10	add
11	sub



a) Design a one-bit ALU slice that performs these four operations. Use full adder, multiplexer, and basic logic gates as your building blocks.

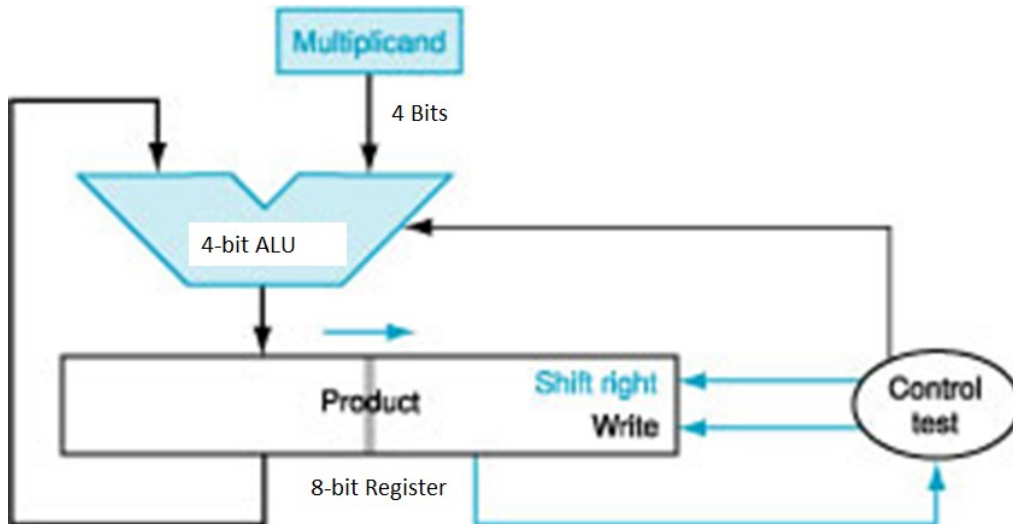


b) Connect four of these one-bit slices to get the required 4-bit ALU.



Q5. Show the contents of the two registers of the optimized multiplication hardware shown below when multiplying the multiplicand 1010_2 by the multiplier 1110_2 over the 4 multiplication steps.

<5 points>



Cycle	Multiplicand	Product
0	1010	0000 1110
1	1010	0000 0111
2	1010	1010 0111 0101 0011
3	1010	1111 0011 0111 1001
4	1010	10001 1001 1000 1100