Com S 321
Fall 2017

NAME: Hosam Abdellatwab

Exam 2

DO NOT OPEN THIS EXAM UNTIL INSTRUCTED TO DO SO

This exam is closed book, closed notes. The use of a calculator is allowed.

NOTE: The datapath handouts given with this exam are different from the ones used in lecture and homework. For problems 4 and 5, students who reproduce values memorized from the lecture handouts will receive ZERO partial credit.

Please write your answers neatly and legibly. Write your name clearly in the space provided at the top of this cover page. Please stop writing when asked to do so.

Problem	Pts	Pts obtained
1	20	20
2	10	10
3	20	20
4	20	17
5	30	7
TOTAL	100	74

1. (20 points)

(a) (6 points) Write a MIPS program fragment **with exactly 3 instructions** for the following high-level language statement:

g = h + g + A [15]; // Assume A is an **int** array with each element taking 4 bytes
t1 t0 s1

Assume that g is in register *$t0*, h is in register *$t1*, and the base address of A is in register *$s1*.

```
lw  $t2, 60($s1)
Add $t3, $t2, $t1
Add $t0, $t0, $t3
```

6

(b) (6 points) Two friends, Harry and David, are arguing. Harry says, "All integers greater than zero and exactly divisible by six have exactly two 1s in their binary representation." David disagrees. He says, "No, but all such numbers have an even number of 1s in their binary representation." Do you agree with Harry or with David, or with neither? Justify your answer with a proof or counterexamples.

Both are wrong since 42 is divisable by 6 & has 3 ones in its binary representation

6

(c) (8 points) Given the bit pattern 1010 1101 1100 1001 1100 0000 0000 0000

what does it represent assuming it is a MIPS instruction?

The format of two possible MIPS instructions for the bit pattern appear below:

Number of bits		6	5	5	16	
LW	rt, disp (rs)	35	rs	rt	16-bit disp	# 35 in binary = 100011
SW	rt, disp (rs)	43	rs	rt	16-bit disp	# 43 in binary = 101011

Your MIPS instruction should specify the operation, the actual register numbers for **rs** and **rt**, and the value of the displacement in decimal.

Useful Powers of 2
$2^{12} = 4{,}096$
$2^{13} = 8{,}192$
$2^{14} = 16{,}384$
$2^{15} = 32{,}768$

opcode = 101011 = 43 = sw

rs = 01110 = 14

rt = 01001 = 9

d = (-ve) 0100 0000 0000 0000 = -16,384

sw 9, -16,384 (14)

8

2. (10 points)

Describe in a sentence or two what the following MIPS program does. Register $a0 is used for input and initially contains **n**, a positive integer. As a function of the input **n**, what does the program put in register $v0 when it reaches the label "done"?

```
begin: addi  $t0, $zero, 0
       addi  $t1, $zero, 0
loop:  slt   $t2, $a0, $t1     # if $a0 < $t1, set $t2 = 1
                               # otherwise set $t2 = 0
       bne   $t2, $zero, done
       add   $t0, $t0, $t1
       addi  $t1, $t1, 1
       j     loop
done:  add   $v0, $t0, $zero
```

a_0	t_0	t_1	t_2	v_0
1	0	0	0	
	0	1	0	
	1	2	1	1
2	1	2	0	
	3	3	1	3
3	3	3	0	
	6	4	1	6
4	6	4	0	
	10	5	1	10
5	10	5	0	
	15	6	1	15

10

v_0 is computing the sum of all integers in t_1 upto n

$2+2+3+4+5+6+\cdots+n$

3. (20 points)

(a) Write a MIPS program to implement **Block Copy** described as follows. Assume that the starting address of the source block is in register $t1, the starting address of the destination block is in $t2, and the number of words to be copied is in $t3 (which is ≥ 0). Furthermore assume the values of these registers and other temporary registers used can be destroyed (so that registers can be used as temporaries to implement Block Copy). Assume also that all the words to be copied are of type **int**.

A(B[0]) = $t1

A(D[0]) = $t2

```
      Addi $t5, $zero, 0 # Counter
      Addi $t6, $t3, 1
Loop: beq  $t5, $t6, Exit
      lw   $t4, 0($t1)
      sw   $t4, 0($t2)
      Addi $t1, $t1, 4
      Addi $t2, $t2, 4
      Addi $t5, $t5, 1
      J Loop
Exit:
```

14

(b) How many instructions are executed in your program to perform a 100-word block copy?

7(100) + 3 = 703

6

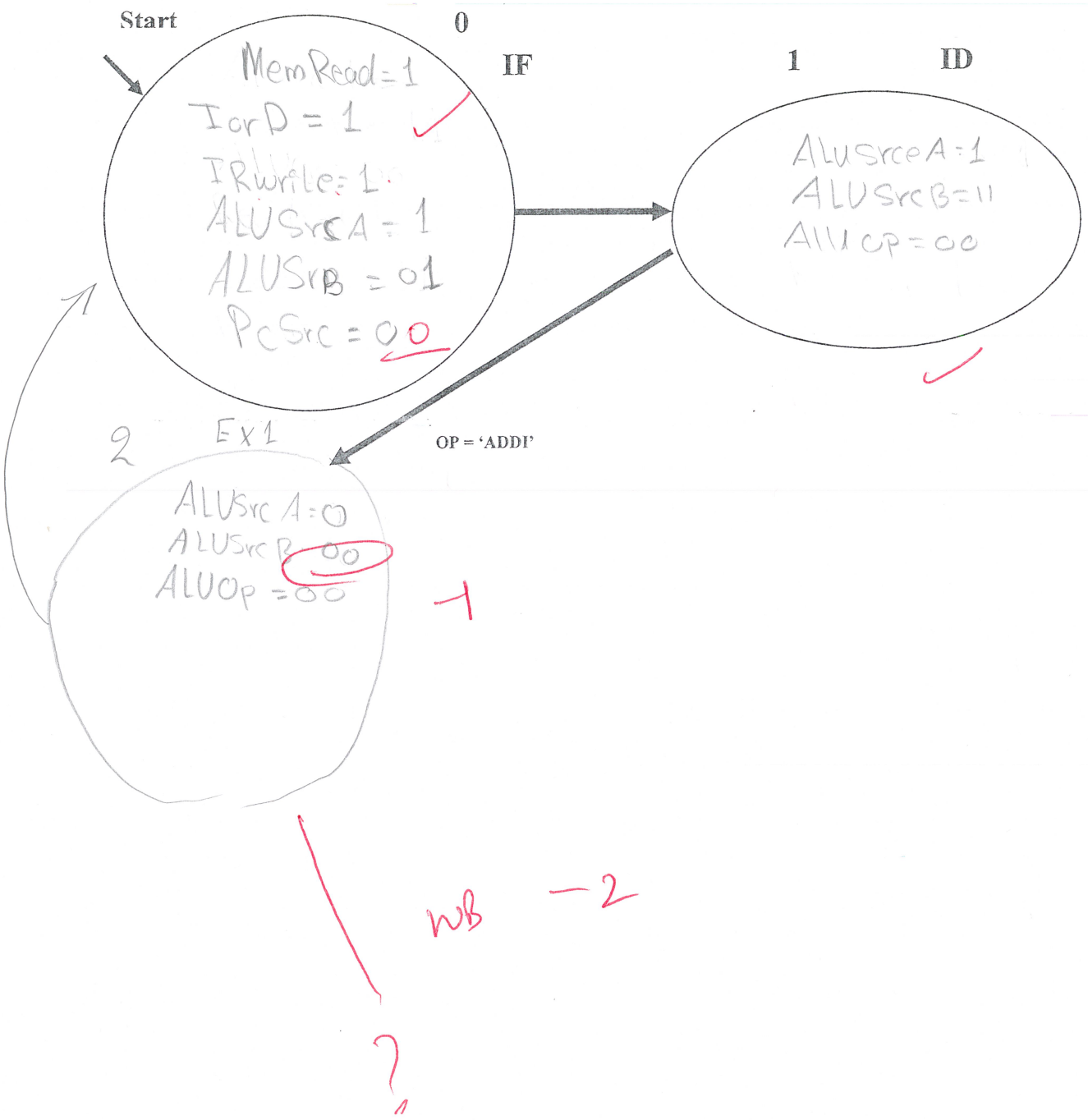
4. (20 points)

(a) Using the diagram given in the handout for the **single cycle** MIPS datapath, complete the control table below with values 0, 1, or D (don't care) for the execution of the instructions: **ADD, LW,** and **SW**.

	ADD	LW	SW
RegDst	0	1	D
AluSrc	1	0	0
MemToReg	1	0	D
RegWrite	1	1	1
MemRead	0	1	0
MemWrite	0	0	1

(b) Using the diagram given in the handout for the MIPS multicycle datapath, specify the **complete finite state machine** for implementation of the **ADDI** instruction. −3

ADDI rt, rs, Imm16 **# R [rt] ← R[rs] + Sign-Extended (Imm16)**
 # rt is bits [20 . . 16] and **rs** is bits [25 . . 21]



5. (30 points) Suppose you wish to implement the following instruction in the MIPS datapath:

DCB rs, Label

DCB stands for Decrement and Branch. As with other branch instructions, **Label** is the branch target address, specified in the low order 16 bits of the instruction word. Register **rs** is in bits [25 . . 21] of the instruction word. This instruction works as follows.

Decrement the contents of register **rs** by 1.
If the result is zero, branch to the target address.
Update the contents of register **rs** with R[rs] – 1.
Assume that DCB is taken, so PC should be updated to the branch target address.

(a) Describe the changes you would make to implement this instruction in the MIPs **single cycle** datapath given in the handout.

I wouldn't change anything since in the single dp we can use the sign extension in order to either decrement or increment.

(b) Specify the values of the following control variables to implement this instruction with the changes you made to the MIPS **single cycle** datapath given in the handout.

Variable	Value
RegDst	1
ALUSrc	0
MemToReg	1 ✓
RegWrite	1 ✓
PCSrc	0 ✓
MemWrite	✗
MemRead	✗
ALUOp (Add or Sub)	sub ✓

(c) Describe the changes you would make to implement this instruction in the MIPS **multicycle** datapath given in the handout.

I won't Change anything since in the multicycle DP we have the abilfty to use multiple registers with sign extensions to achieve our goal

(d) Specify the **complete finite state machine** for the control necessary to implement DCB with the changes you made to the MIPS **multicycle** datapath given in the handout.