COE608: Computer Organization and Architecture

Mid Term Examination

Student Name: [redacted] Student #:

i) Answer all the questions. **Total Marks: 35**
ii) Total time allowed is 50 minutes. Any notes or books are not allowed.
iii) Estimated time for each question is equivalent to the marks assigned to it.
iv) All the questions are not of equal difficulty. Read the questions carefully.

1. (a) Write the VHDL entity of a 3-bit wide Full Adder. (Look up comp Arithmetic lecture) Slide 12

MARKS:

```
library ieee;
use ieee.std_logic_1164all
entity adder_3 is
        port (A, B : in std-logic_vector (2 downto 0);
              C0 : in std-logic;
              S : out std-logic-vector (2 downto 0),
              C3 : out_logic
end adder_3;
```

1. (b) What information about a VHDL design is represented in the Architecture?

MARKS

interna
- Behaviour of the Comp.
- Signals
- inner - Comp

Convert 21.85 ↙ ↓

Question 2

10101

$0.85 \times 2 = 1.7$	1
$0.7 \times 2 = 1.4$	1
$0.4 \times 2 = 0.8$	0
$0.8 \times 2 = 1.6$	1
$0.6 \times 2 = 1.2$	1
$0.2 \times 2 = 0.4$	0
$0.4 \times 2 = 0.8$	0
$0.8 \times 2 = 1.6$	1
$0.6 \times 2 = 1.2$	1
$0.2 \times 2 = 0.4$	0
	0
	1
	1
	⋮

$0.85_{10} = .11011001100110011\ldots$

$21.85_{10} = 10101.110110011 00\ldots \times 2^0$

Normalize: $1.\underbrace{010111011001100\ldots}_{\text{Mantissa}} \times 2^4$

Exponent: $4 + \text{bias} = 4 + 127 = 131$

$131_{10} = 10000011_2$ Sign = 0

∴ Single precision

0	10000011	01011101100110011001100110

sign Exponent Mantissa

Double precision

Exponent: $4 + \text{bias} = 4 + 1023 = 1027$

$1027_{10} = 10000000011_2$ sign = 0

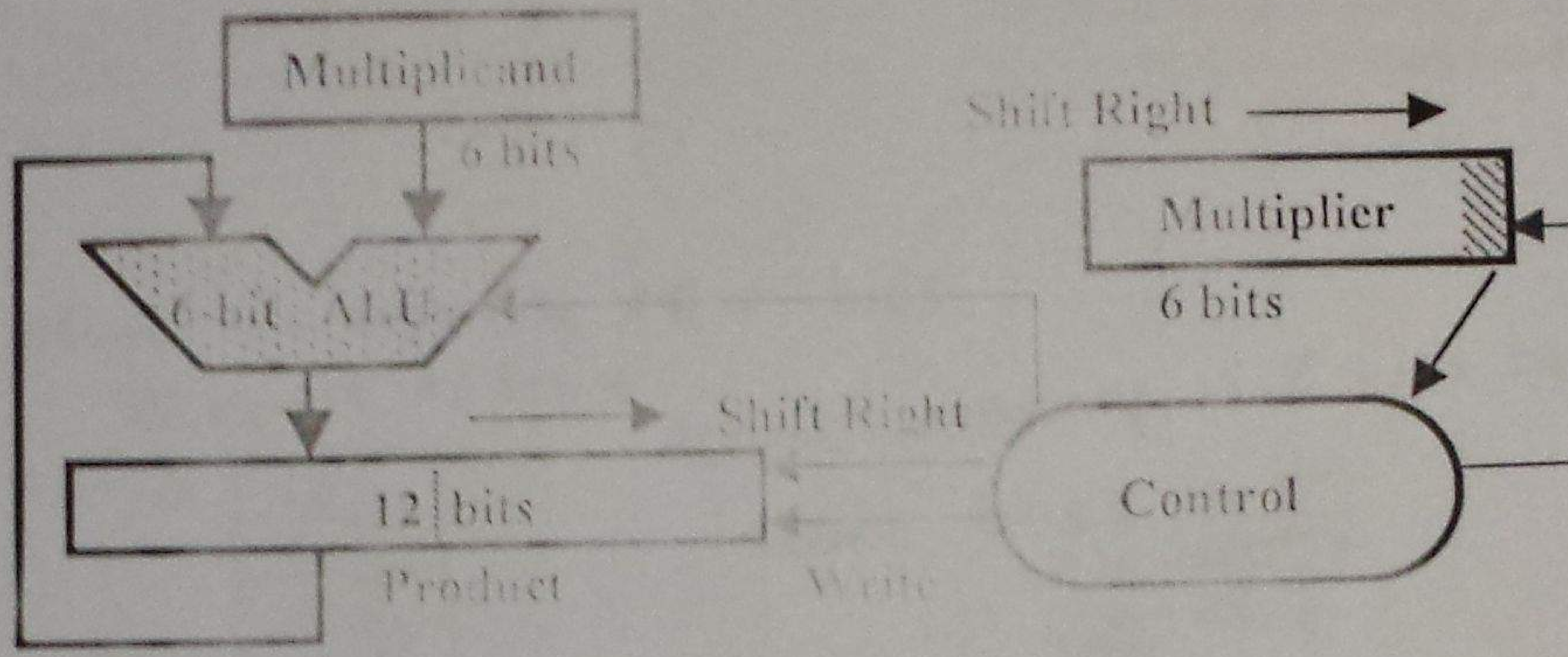
0	10000000011	0101110110011001100110011001100110011001100110011 00110011001

* Computer Airthmetic lecture

Slide 34

Algorithm

3. The block diagram of a 6-bit multiplier with optimal size ALU (adder) and registers is given below.



Assume that the Multiplicand and Multiplier registers are loaded with 6-bit numbers for multiplication and the product register is cleared initially as shown below. Two 6-bit unsigned binary numbers are to be multiplied using shift or add/shift operations. During this process, show the contents of all these registers that change. Determine the contents of these registers after an add/shift or just shift operation and fill up the following table after each operation.

MARKS: 10

Operation Shift and/or Add	Product		Multiplicand	**Multiplier**
Initial Values	0 0 0 0 0 0	0 0 0 0 0 0	0 1 1 0 1 1	0 1 0 1 0 1
ADD	011011	000000	011011	010101
Shift	001101	100000	011011	001010
Shift	000110	110000	011011	000101
ADD	100001	110000	011011	000101
Shift	010000	111000	011011	000010
Shift	001000	011100	011011	000001
ADD	100011	011100	011011	000001
Shift	010001	101110	011011	000000
Shift	001000	110111	011011	000000
			011011	

Mul(0)=1

Mul(0)=0

4. (a) Consider the following C code:

X[13] = m - n

i) For a load-store type 32-bit MIPS CPU, write an optimal assembly language instruction sequence for the above C code. Assume that variables m and n are held in registers $s1 and $s2 respectively while address of X[10] is in register $s0. You are free to use any of the temporary registers $t0-$t7.

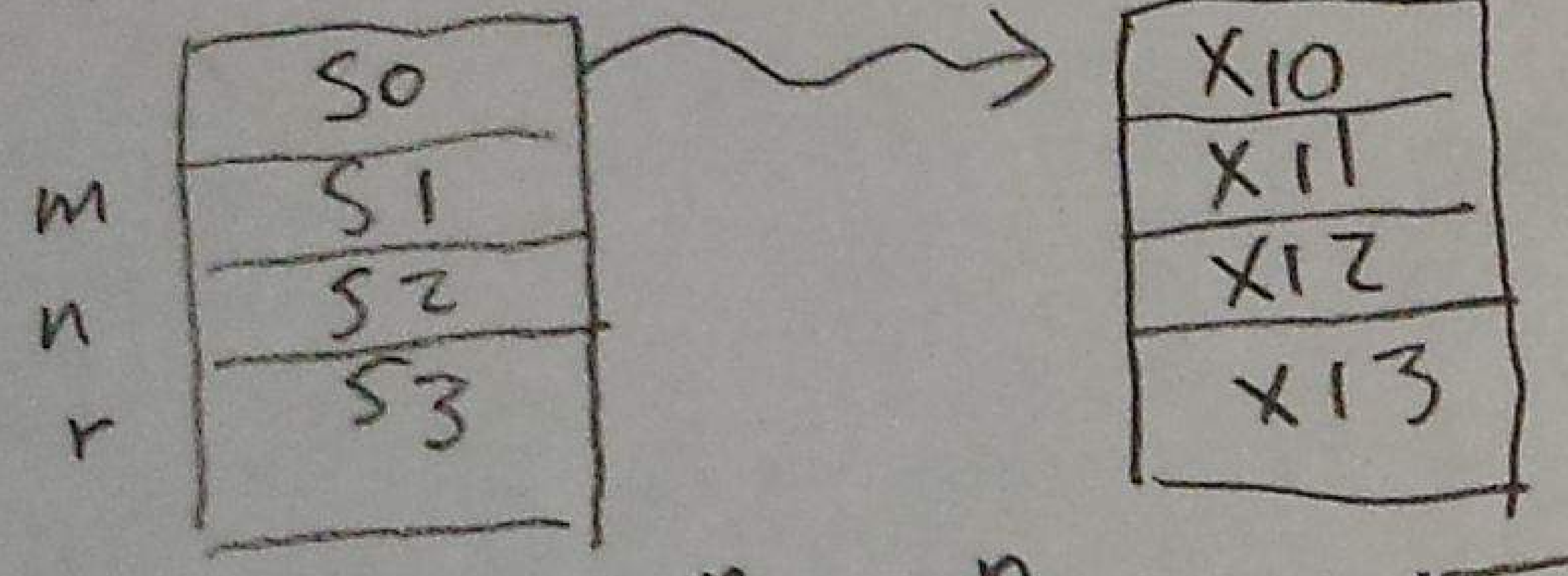
ii) What is the size of the assembly code developed in part (i)?

MARKS: 6 (4+2)

# of instructions in assembly x # of bits

Pick 0003 as Start

Assume memory is 8 bits each (4 locations $\frac{32}{8} = 4$)



Start 0003
4
5
6
⋮
15 → 15 − 3 = 12

```
      m     n
SUB $t0, $s1, $s2;
ADDI $t1, $s0, #12;
ST   $t1, $t0
```

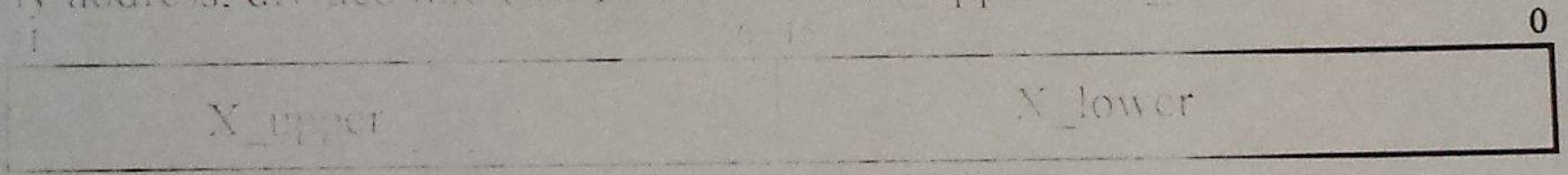
$\boxed{\begin{array}{l}\text{SUB \$t0, \$s1, \$s2;}\\ \text{SW \$t0, 12(\$s0)}\end{array}}$

$\boxed{\text{Alternate code}}$

3 instructions X 32 bits = 96

2 ins x 32 = 64

4. (b) If X is a 32-bit memory address, divided into two 16-bit values X_upper & X_lower as shown below.



Typically following instruction sequence is used to load the data at Address X into a register ($s0).

```
lui     $t0, X_upper
ori     $t0, $t0, X_lower
lw      $s0, 0($t0)
```

↑ From location

Consider the following alternate code that is more efficient

```
lui     $t0, X_upper
lw      $s0, X_lower($t0)
```

Is this code accurate? YES or NO

If YES, justify your answer.

If NO, identify the mistakes and suggest any corrections

MARKS: 5

Code is accurate