Université d'Ottawa
Faculté de génie

École de science informatique
et de génie électrique

uOttawa
L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical Engineering
and Computer Science

COURSE: CEG2136/CEG2536
Computer Architecture I
SEMESTER: Fall 2012

Assignment 2

Q1. In a signed binary system of 8 bits (**N=8**):

a) Find the 2's complement **representation** of the following signed numbers :

$(+63)_{10}$ = (00111111)

$(-115)_{10}$ = (10001101)

b) Find the 2's complement of the following signed numbers and give your results in decimal, too: $(-63)_{10}$ and $(+115)_{10}$

The 2's complement **representation** of $(-63)_{10} = (11000001)_2$

=> **Do the 2's complement of $(-63)_{10}$** = 2's complement of $(11000001)_2$ = $(\mathbf{00111111})_2$ = which is the 2's complement representation of $(\mathbf{+63})_{10}$ (which is the result we expected)

The 2's complement representation of $(+115)_{10}$ = (01110011)

=> **Do the 2's complement of $(+115)_{10}$** = 2's complement of (01110011) = $(\mathbf{10001101})_2$ = 2's complement representation of $(\mathbf{-115})_{10}$

c) Perform the following arithmetic operations using the signed-2's complement representation and provide your results in decimal (including intermediary steps), as well:

1) Say A = $\mathbf{(+115)_{10}}$ and **B** = $\mathbf{(-63)_{10}}$

S = A + B = $(+115)_{10} + (-63)_{10}$

	CY_8	CY_7	CY_6	CY_5	CY_4	CY_3	CY_2	CY_1	CY_0	Base 10
CY:	1	1	0	0	0	0	1	1	0	
A +		0	1	1	1	0	0	1	1	115_{10}+
B		1	1	0	0	0	0	0	1	-63_{10}
S=A+B		0	0	1	1	0	1	0	0	52_{10}

$CY_8 = CY_7 = 1$ => NO overflow

2) Say **C**=$\mathbf{(-115)_{10}}$ and **D**=$\mathbf{(-63)_{10}}$

X = C - D = $(-115)_{10} - (-63)_{10} = (-115)_{10} + [-(-63)]_{10} = (-115)_{10} + (+63)_{10}$

2's complement representation Base 10

	CY_8	CY_7	CY_6	CY_5	CY_4	CY_3	CY_2	CY_1	CY_0	
CY:	0	0	1	1	1	1	1	1	0	
C+		1	0	0	0	1	1	0	1	-115_{10}
- D		0	0	1	1	1	1	1	1	63_{10}
X=C-D		1	1	0	0	1	1	0	0	-52_{10}

As the msb of X = C-D is 1 => X is a negative number (X = –y) with a magnitude
y=-(-y) = -(X) = 2's complement of (X) = 2's complement of (11001100) = 00110100 = 52_{10}

=> X = -y = -52_{10}

$CY_8 = CY_7 = 0$ => NO overflow

Q2. a) Identify the decimal number which is represented next with 32-bit in the IEEE 754 standard: (1 **10001011** **11101**000000000000000000) = $(?)_{10}$

Sign = 1 : => negative ; C = 139, hence Exp = C-127 = 12

1.M = 1.**11101** (where M = mantissa)

-1.11101 * 2^{12} = -1111010000000 = $(-7808)_{10}$

b) Represent (221.390625) in the IEEE 754 standard with 32 bits.

$(221.390625)_{10} = (11011101.011001)_2$ = (1.**1011101011001**) x 2^7

Sign = 0 ; Exp = 7; C = 127 + 7 = 134 = $(10000110)_2$

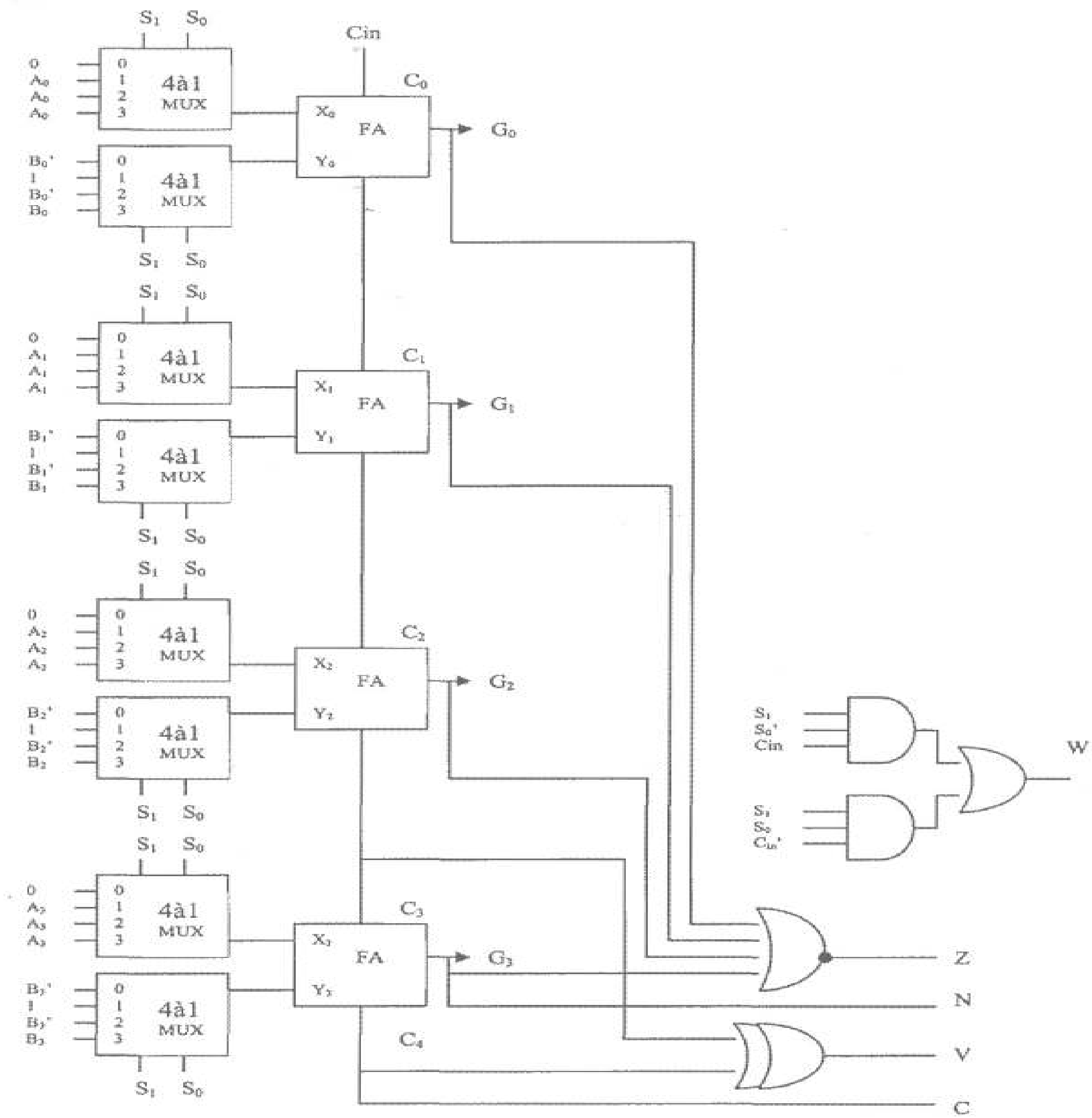
M = **1011101011001**

$(221.390625)_{10}$ = (0 10000110 **1011101011001**0000000000)IEEE754

Q3. Devise a 4-bit arithmetic unit with 2 control inputs (S_1 and S_2) that can perform the following arithmetic operations:

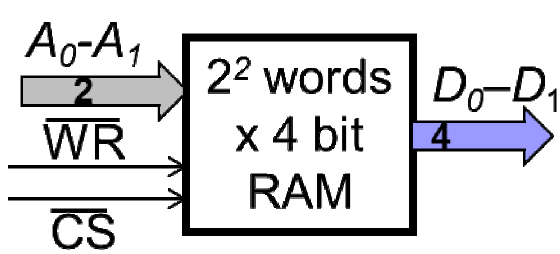
Table 1: Table of arithmetic unit functions.

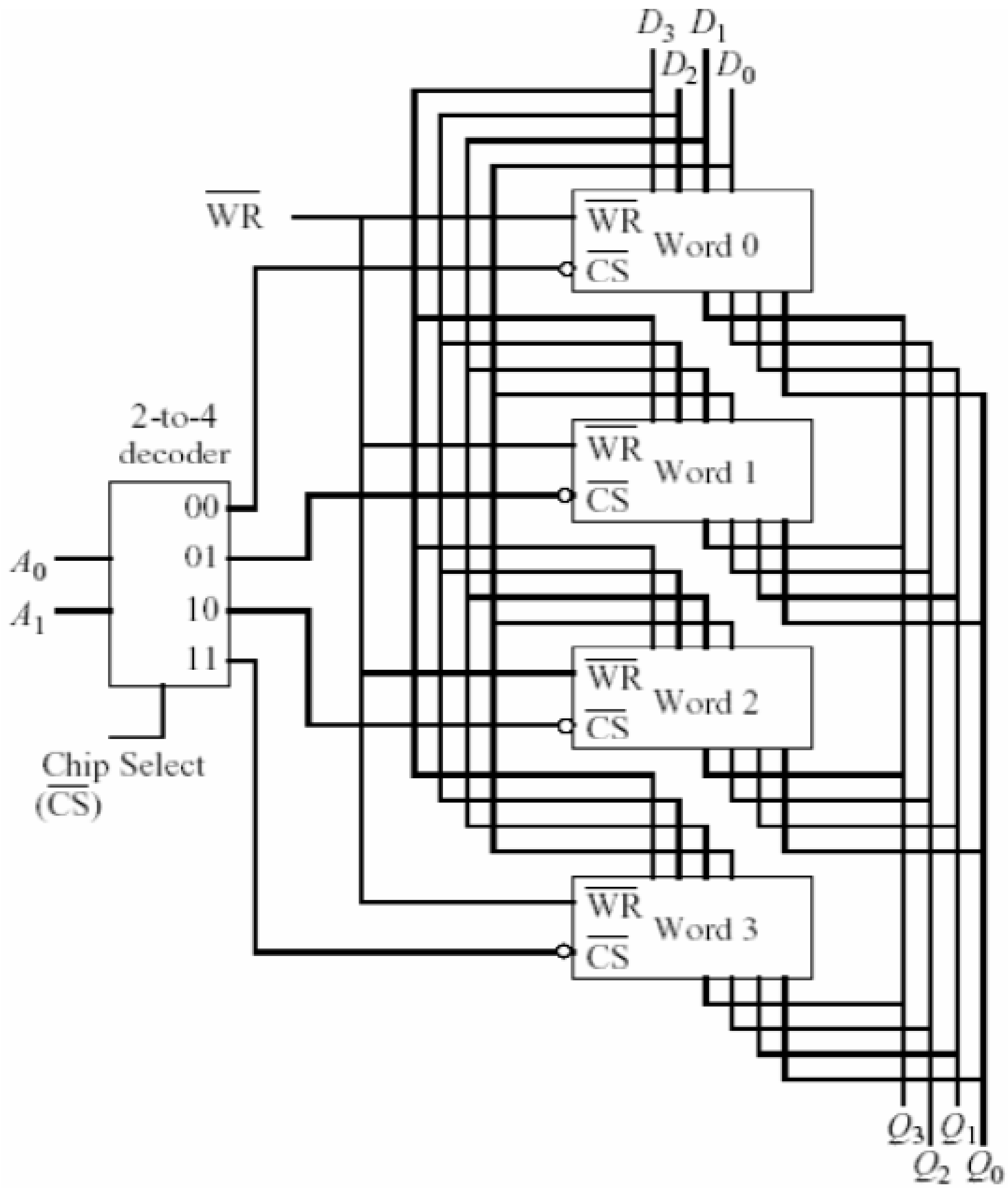
S_1S_0 (Control inputs)	G (output)	
	$C_{in} = 0$	$C_{in} = 1$
00	G = B′ (1's complement)	G = B′ + 1 (2's complement)
01	G = A − 1 (decrement)	G = A = A-0001+0001 (transfer)
10	G = dddd (invalid operation) = = A + B′	G = A + B′ + 1 (subtraction)
11	G = A + B (addition)	G = dddd (invalid operation) = A + B +1



V can also be implemented as $X3Y3G3' + X3'Y3'G3$

Q4. Draw the logic diagram of a RAM of 4 words of 4 bits (4 x 4). Use four 4-bit registers, digital components, and any simple logic gates you may need.

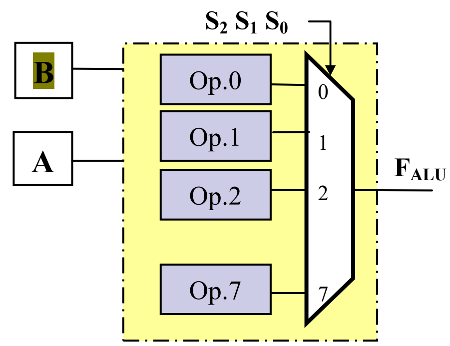


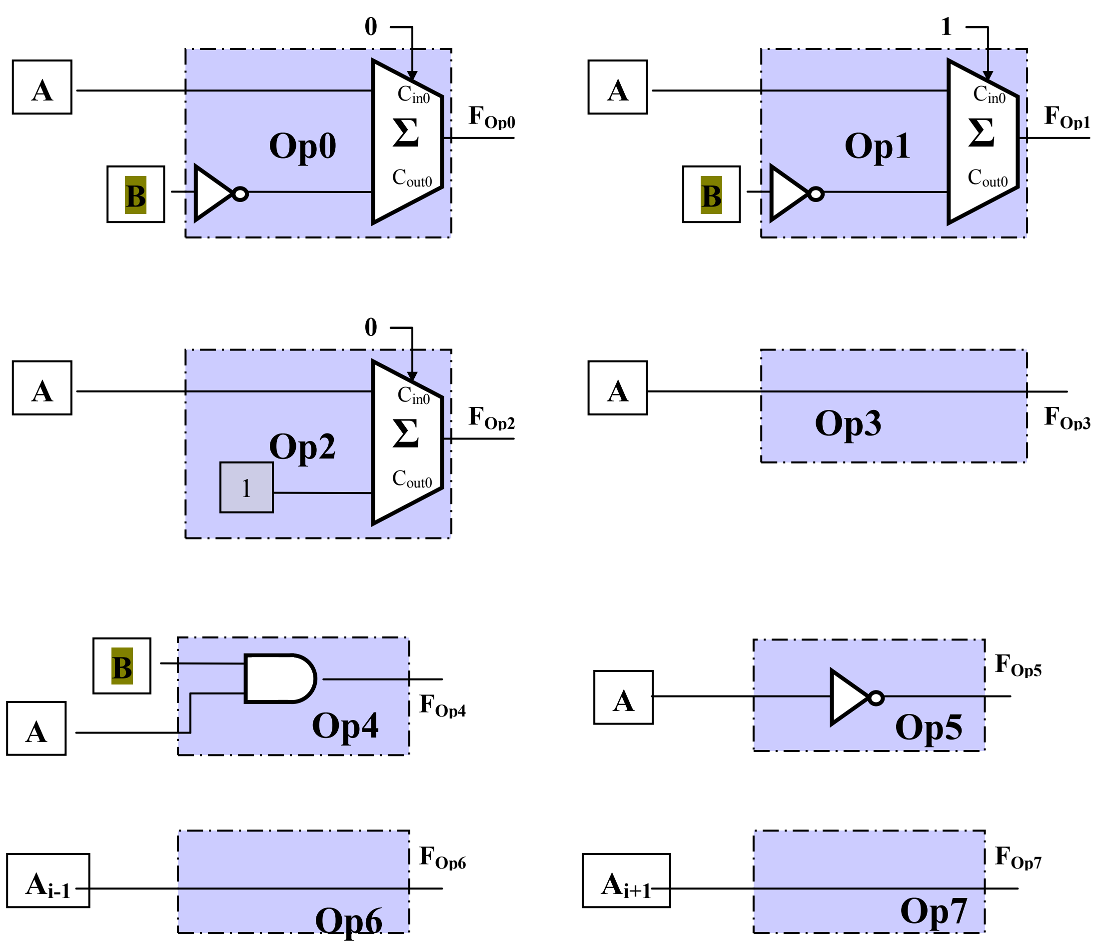

D3 D1
D2 D0
WR
WR Word 0
CS
2-to-4
decoder
00
01
10
11
A0
A1
Chip Select
(CS)
WR Word 1
CS
WR Word 2
CS
WR Word 3
CS
Q3 Q1
Q2 Q0

Q6. Design a 3-bit arithmetic logic unit (ALU) which performs the micro-operations described in Table 2. The operands are signed numbers represented in 2's complement format and are stored in two 3-bit registers $A = A_2A_1A_0$ and $B = B_2B_1B_0$; results are provided at the ALU's output $F = F_2F_1F_0$

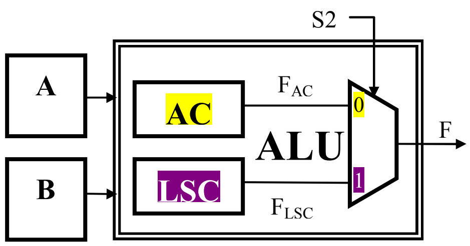
I. ALU architecture consists in an 8-to-1 MUX and 8 "smaller" ALU's (modules Op.0 – Op.7) which implement each operation described by a row of the given table.

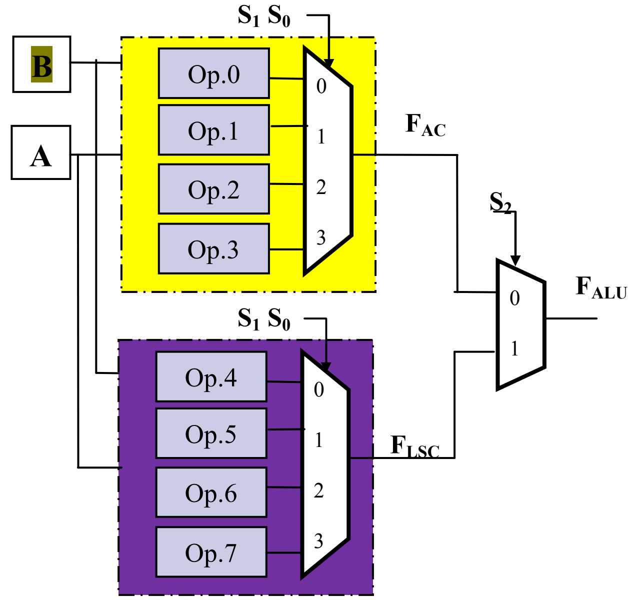
S2 S1 S0		Operation
0 0 0	Op0	F = A−B−1 =
0 0 1	Op1	F = A – B =
0 1 0	Op2	F = A – 1 =
0 1 1	Op3	F = A
1 0 0	Op4	F = A ^ B
1 0 1	Op5	F = A'
1 1 0	Op6	F = ashl A
1 1 1	Op7	F = ashr A





II. An equivalent, more structured ALU architecture uses S_2 to form two partitions of the operations: arithmetic (S_2=0) and logic & shifting (S_2=1)



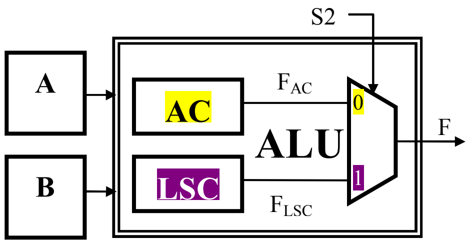


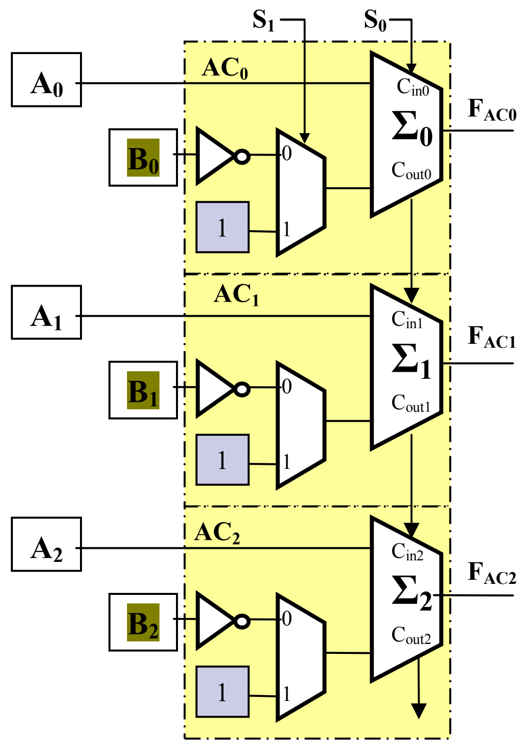
III. Since one single operation is performed at some point in time, it makes sense to try to minimize the number of circuits by using the same adder for all arithmetic operations, if possible, and not 3, one for each operation, like in the initial design (step 1 or 2). Inspecting the table, one can observe:

		Opcode			
Select S2 S1 S0	S2= 0=arithmetic 1=shift/logic	S1 Op	S0 Carry In if arith	Operation	Description
0 0 0	0	0	0	F = A−B−1 = =A+(B'+1)-1= = A + B'	Subtraction with borrow
0 0 1			1	F = A – B = A+(B'+001) = A+B'+001	Subtraction
0 1 0		1	0	F = A – 1 = = A – 001 = = A + 111	Decrement
0 1 1			1	F = A = = A-001+001= = A+111 + 001	Transfer
1 0 0	1	0	0	F = A ^ B	Masking
1 0 1		0	1	F = A'	Complementation
1 1 0		1	0	F = ashl A	Arithmetic left shift (multiplication

					by 2)
1 1 1		1	1	F = ashr A	Arithmetic right shift (division by 2)

Hierarchical design
ALU Block Diagram



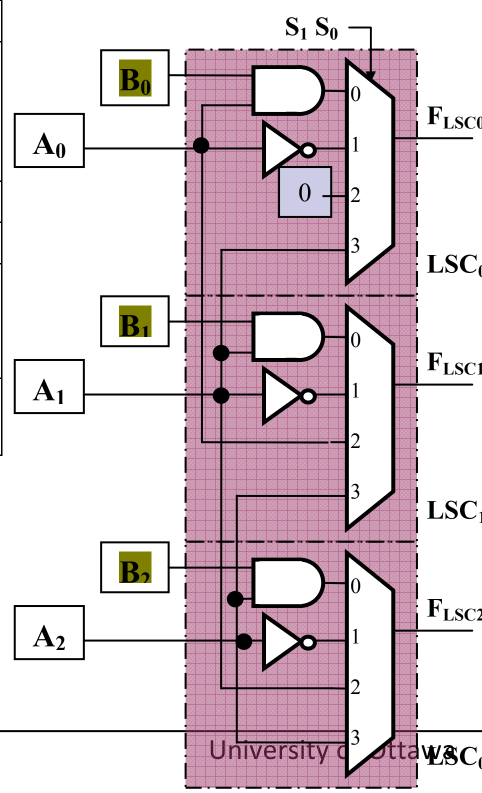


1) Draw a detailed logic diagram of the ***arithmetic unit*** (AC) of the ALU, using any type of gates you may need.
2) State variable W is set to 1 if an overflow occurs in the arithmetic unit, and stays at 0 otherwise. Find a simplified logic expression of W.

$$W = \left(C_{out2} \cdot \overline{C_{out1}} + \overline{C_{out2}} \cdot C_{out1}\right) \cdot \overline{S_2} = \left(C_{out2} \oplus C_{out1}\right) \cdot \overline{S_2}$$

3) Draw a detailed logic diagram of the ***logic unit*** (LSC) of the ALU, using AND, OR and NOT gates.

		Opcode			
Select S2 S1 S0	S2 =1 shift/ logic	S1	S0	Operation	Description
1 0 0		0	0	F = A ^ B	Masking
1 0 1	1	0	1	F = A'	Complementation
1 1 0		1	0	F = ashl A	Arithmetic left shift (multiplication by 2)
1 1 1		1	1	F = ashr A	Arithmetic right shift (division by 2)



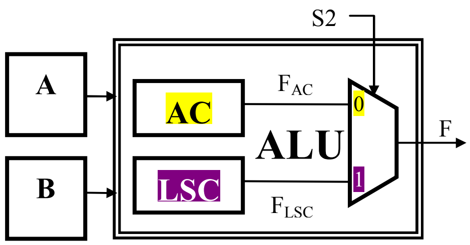
4) State variable V is set to 1 if an overflow is triggered by a shift in the logic unit, and stays at 0 otherwise. Find a simplified logic expression of V.

$$V = \left(A_2 \cdot \overline{A_1} + \overline{A_2} \cdot A_1\right) \cdot S_1 \cdot \overline{S_0}$$

5) State variable T is set to 1 if an overflow occurs after performing any micro-operation defined by Table 2, and stays at 0 otherwise. Find a simplified logic expression of T.

$$T = V \cdot S_2 + W \cdot \overline{S_2}$$

6) Considering the *arithmetic unit* and the *logic unit* being represented as blocks, draw the diagram of the ALU.



CONCLUSIONS

All three solutions (I, II and III) are logically correct, but of course, the last one provides an optimized/ minimized implementation.

Q7. Using full adders (represented as block-diagrams) draw the logic diagram of a circuit that is capable of performing additions and subtractions on two 4-bit unsigned numbers $A = A_3A_2A_1A_0$ and $B = B_3B_2B_1B_0$. Results has to be provided as 5-bit signed numbers $S = S_4S_3S_2\,S_1S_0$ represented in sign-magnitude format. The circuit functionality is controlled by bit M as follows:

- If M = 0, then S = A – B
- If M = 1, then S = A + B