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

CEG2136: Computer Architecture I / CEG2536: Architecture des Ordinateurs I
MIDTERM EXAMINATION
Professors: Voicu Groza, Fadi Malek and Miguel Garzon
Duration: 1 hour and 20 minutes October 21, 2014, 14:00

SOLUTIONS

Question 1 (30 points)

a) (15 points) Using D-type flip-flops, design a 3-bit Gray code counter which has the following counting sequence:

000 → 001 → 011 → 010 → 110 → 111 → 101 → 100
↑ ↓
← ← ← ← ← ← ← ← ← ← ← ← ← ← ←

Draw the transition table of the counter and derive the excitation equations of the D flip flops' inputs.

Present state S^n			Next state S^{n+1}		
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	0	1	0
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	1

Q_2^+

Q_2 \ Q_1Q_0	00	01	11	10
0	0	0	0	1
1	0	1	1	1

$Q_2^+ = Q_2 Q_0 + Q_1 Q'_0$

Q_1^+

Q_2 \ Q_1Q_0	00	01	11	10
0	0	1	1	1
1	0	0	0	1

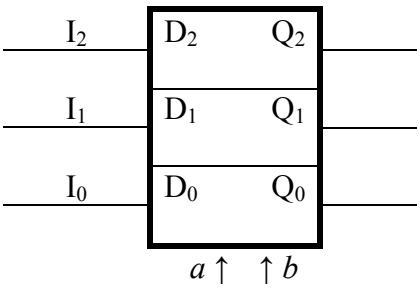
$Q_1^+ = Q'_2 Q_0 + Q_1 Q'_0$

Q_0^+

Q_2 \ Q_1Q_0	00	01	11	10
0	1	1	0	0
1	0	0	1	1

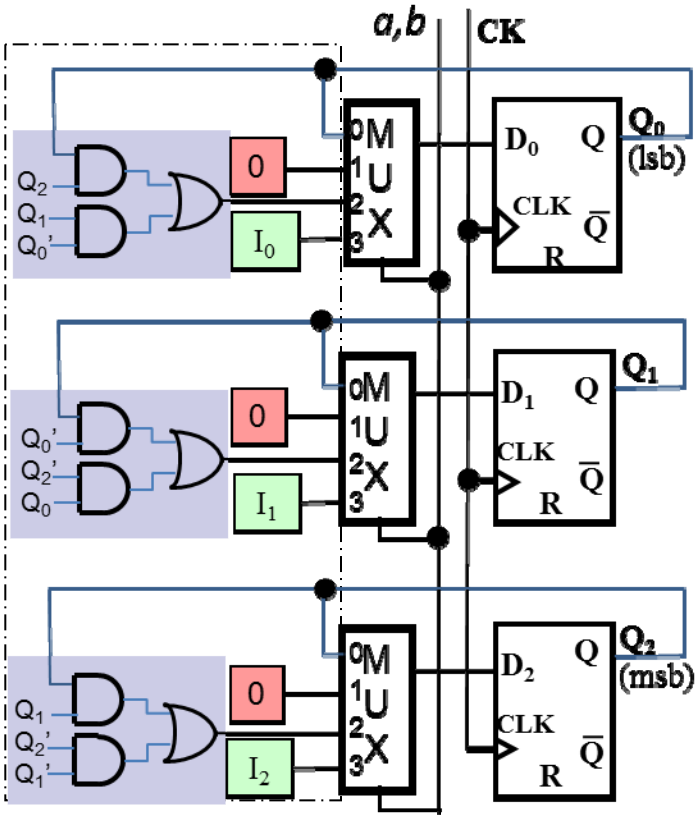
$Q_0^+ = Q'_2 Q'_1 + Q_2 Q_1$

b) (15 points) Design a 3-bit multi-function register whose operation is described in the following table; I_2, I_1, I_0, are register's data inputs, while *a* and *b* are the control bits.



Clock	*a*	*b*	Operation
↑	0	0	No change = stores current state
↑	0	1	Synchronous reset/clear (0 ➔ $Q_2Q_1Q_0$)
↑	1	0	Gray code counter: advance to the next state
↑	1	1	Parallel loading external inputs $I_2\ I_1\ I_0$ ➔ $Q_2Q_1Q_0$

a	*b*	Function	Transition Equations (preliminary steps)	Excitation Equations D_i (final results - derived for each case separately)
0	0	f^0: Store register's content	$Q_i\,(n+1) = Q_i\,(n)$; $i = \{0,1,2\}$	$D_i = Q_i\,(n)$; $i = \{0,1,2\}$
0	1	f^1: Synchronous clear	$Q_i\,(n+1) = 0$; $i = \{0,1,2\}$	$D_i\,(n+1) = 0$; $i = \{0,1,2\}$
1	0	f^2: Gray code counter: advance to the next state	$Q_2^+ = Q_2\,Q_0 + Q_1\,Q'_0$ $Q_1^+ = Q'_2\,Q_0 + Q_1\,Q'_0$ $Q_0^+ = Q'_2\,Q'_1 + Q_2\,Q_1$	$D_2 = Q_2\,Q_0 + Q_1\,Q'_0$ $D_1 = Q'_2\,Q_0 + Q_1\,Q'_0$ $D_0 = Q'_2\,Q'_1 + Q_2\,Q_1$
1	1	f^3: Loading external inputs, $I_2\ I_1\ I_0$	$Q_i\,(n+1) = I_i\,(n)$; $i = \{0,1,2\}$	$D_i = I_i\,(n)$; $i = \{0,1,2\}$



Question 2 (30 points)

Two's complement representation is used for signed numbers in this question.
The 8-bit registers AR, BR, CR and DR initially have the following values:
AR = 0 1100001; BR = 1 0110000;
CR = 1 0001110; DR = 0 1000011.

An 8-bit adder is employed to perform the following operations:
CR = AR + BR
DR = AR – BR

1. **(10 points) Determine the 8-bit values in each register after the execution of these micro-operations.**

CR = AR + BR

		64	32	16	8	4	2	1
Carry:	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0
	1	1	1	0	0	0	0	0	
AR		0	1	1	0	0	0	0	1
BR		1	0	1	1	0	0	0	0
CR		**0**	0	0	1	0	0	0	1

Verification in decimal:
= 97
= - 80
17

No verification is required here!

BR = 1 0 1 1 0 0 0 0 < 0 ➔ BR = -|BR|
|BR| = - BR = 2's compl (1 0 1 1 0 0 0 0) = 0 1 0 1 0 0 0 0 = (64+16) = 80_{10}
BR = -|BR| = - 80_{10}

DR = AR - BR = AR + (- BR) = AR + 2's compl (BR) = 0 1 1 0 0 0 0 1 + 2's compl (1 0 1 1 0 0 0 0)
= 0 1 1 0 0 0 0 1 + 0 1 0 1 0 0 0 0

Carry:	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0
	0	1	0	0	0	0	0	0	
AR		0	1	1	0	0	0	0	1
-BR		**0**	1	0	1	0	0	0	0
DR		**1**	0	1	1	0	0	0	1

Verification in decimal:
= 97
= 80
BAD : DR<0 ↔ 177?

No verification is required here

Conclusion:
AR = 0 1 1 0 0 0 0 1;
BR = 1 0 1 1 0 0 0 0;
CR = 0 0 0 1 0 0 0 1;
DR = 1 0 1 1 0 0 0 1

2. **(10 points) Convert to decimal all four binary numbers stored in these registers after the execution of the micro-operations.**

AR = 0 1 1 0 0 0 0 1 = + 64+32+1 = +97

BR = 1 0 1 1 0 0 0 0 < 0 ➔ BR = -|BR|
|BR| = - BR = 2's compl (1 0 1 1 0 0 0 0) = 0 1 0 1 0 0 0 0 = (64+16) = 80_{10}
BR = -|BR| = - 80_{10}

CR = 0 0 0 1 0 0 0 1 = 16+1= +17

DR = 1 0 1 1 0 0 0 1 < 0 ➔ DR = -|DR|
|BR| = - BR = 2's compl (1 0 1 1 0 0 0 1) = 0 1 0 0 1 1 1 1 = 64+15 = 79_{10}
DR = -|DR| = - 79_{10}

(4 points) Represent in BCD the decimal number equivalent to the binary number stored in register AR.

AR = 0 1 1 0 0 0 0 1_2 = 97_{10} = (1 0 0 1 0 1 1 1) $_{BCD}$

3. **(6 points) Is there any overflow? Justify your answer. How would a computer detect overflows in these operations?**

No overflow in CR = AR + BR since AR>0 and BR<0 cannot generate OFL
Overflow possible in DR = AR – BR because AR>0 and – BR>0, and it is overflow since 97+80 = 177 > 127, the largest positive number with 8 bits

Computer sees in
CR: OFL = $c_8 \oplus c_7 = 1 \oplus 1 = 0$
DR: OFL = $c_8 \oplus c_7 = 0 \oplus 1 = 1$

Question 3 (40 points)

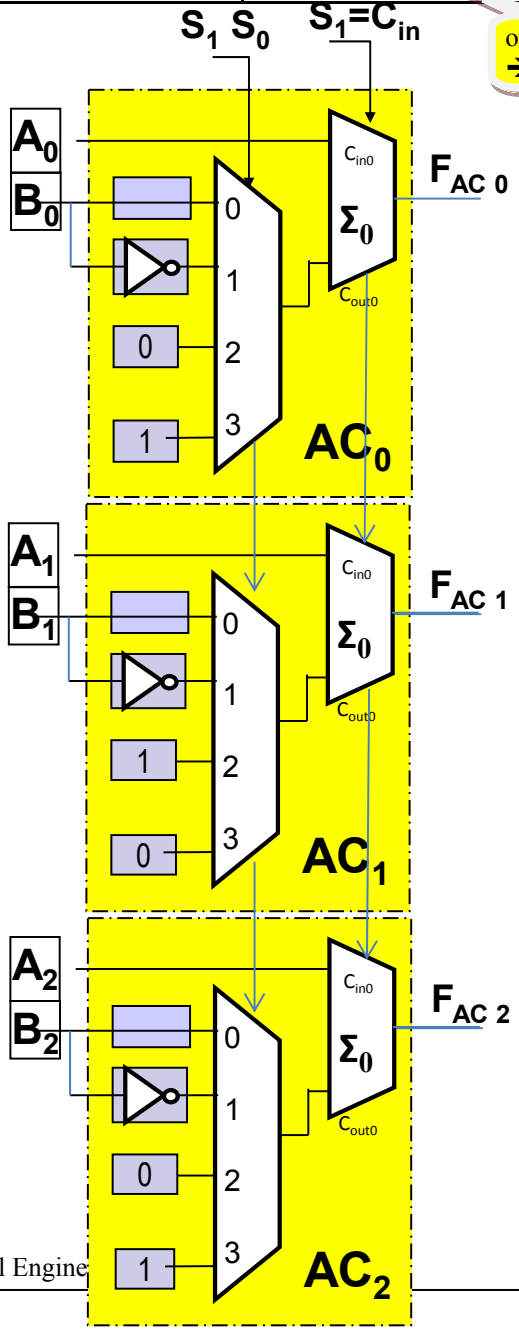
In this problem you have to design a 3-bit arithmetic and logical unit (ALU) which executes the micro-operations described in Table 1. The ALU takes its operands from two 3-bit registers $A=A_2A_1A_0$ and $B=B_2B_1B_0$, and returns an output $F = F_2F_1F_0$. The micro-operation to be executed is specified by the code bits $S_2\,S_1\,S_0$. In the case of arithmetic operations, assume that the contents of A and B are signed numbers in 2's complement representation.

S_2	S_1	S_0	Operation Description
0	0	0	F← A + B Addition
0	0	1	F← A + 2 Increment by 2
0	1	0	F← A - B Subtraction
0	1	1	F← A - 2 Decrement by 2
1	0	0	F← (A ∨ B)' NOR
1	0	1	F← (A ⊕ B)' Exclusive-NOR
1	1	0	F← ashl A Arithmetic shift- left
1	1	1	F← ashr A Arithmetic shift- right

A. (10 points) Draw a detailed logic diagram of the ALU's arithmetic unit.

S_1	S_0	Operation Description		Carry
0	0	F← A + B Addition	$A_2A_1A_0$+ $B_2\,B_1\,B_0$	
0	1	F← A + 2 Increment by 2	$A_2A_1A_0$+ 0 1 0	
1	0	F← A - B Subtraction	$A_2A_1A_0$+ $B_2'B_1'B_0'$	+001
1	1	F← A - 2 Decrement by 2	$A_2A_1A_0$+ 1 0 1	+001

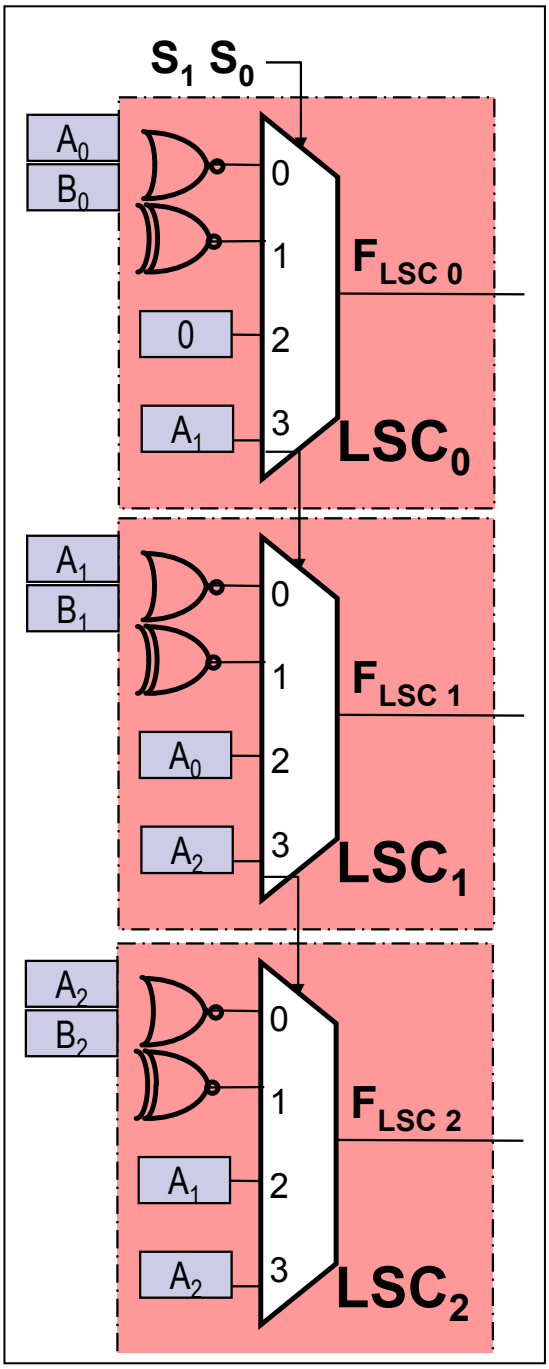
or 110 + 000 ➔ $C_{in}=S_1S_0'$



(2 points) A Boolean variable W is used to show whether one of the arithmetic operations described in Table 1 has caused an overflow. In the case of an overflow, W is set to 1 otherwise W is set to 0. Give a Boolean expression for W.

$W = c_{in3} \oplus c_{out3}$

B. (10 points) Draw the logic diagram of the logic and shift unit of the ALU.



(3 points) A Boolean variable V is used to determine whether the arithmetic shift operations have caused an overflow. In the case of an overflow, V is set to 1, and V is reset to 0 otherwise. Find a Boolean expression of V.

$V = S_1 S_0' (A_2 \oplus A_1)$

C. (5 points) A Boolean variable T is used to signalize if any of the operations described in Table 1 causes an overflow. In the case of an overflow, T is set to 1, and otherwise T is set to 0. Find a Boolean expression of T.

$T = S_2' W + S_2 V$

(10 points) Use bloc diagrams of the arithmetic and logic & shift units in order to draw the block diagram of the complete ALU, including the overflow detection bit T.