Universidad Carlos III de Madrid

Digital Electronics. 1st midterm exam. March, 2014

Groups 65-69-79-95

Surname, Name:

Time: 1h. 40’

Question 1.1 (0.25 points)

Given the decimal integer numbers A = 88 and B = -50.

a) Represent A and B in 2’s complement with the minimum possible number of bits.

b) Using 2’s complement representations for the numbers, perform the operations A-B and A+B. Point out if there is overflow in any of this operations and why.

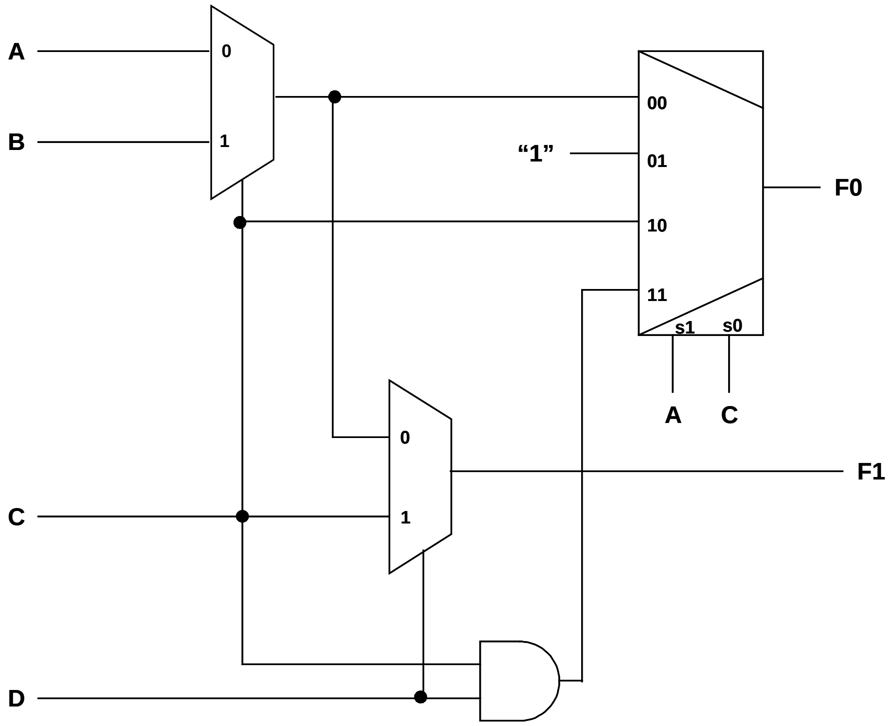
Question 1.2 (0.25 points)

a) Draw the 3-bit Gray’s Code

b) Draw the 3-bit Jonhson’s Code

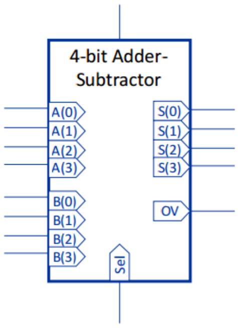
Question 1.3 (0.25 points)

Draw the truth table of the following circuit:



Question 1.4 (0.25 points)

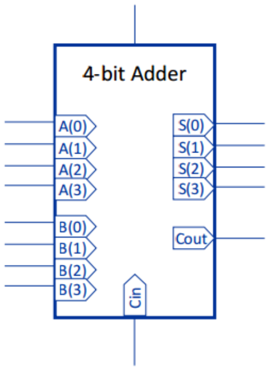
We want to design the following circuit (a 4-bit adder/subtractor).



Where A and B are the data inputs (4-bit), S is the data output (4-bit), Sel is the operation selection (‘0’ means addition and ‘1’ means subtraction), and OV is an overflow indicator (it is active-high, and it is activated when there is overflow in the operation).

To design this circuit we have available the following components:

- Logic gates
- A 4-bit adder like the one of the figure:



-

Where A and B are the data inputs (4-bit), S is the data output (4-bit), and Cin/Cout are the carry-in and carry-out of the 4-bit adder.

Design the circuit using these available components.

Problem (1 points)

Given the following logic function:

$$f(a,b.c.d) = \overline{a}cd + \overline{b}(a + c + \overline{d})$$

a) Find the most simplified logic expression as a sum of products
b) Find the most simplified logic expression as a product of sums
c) Implement the logic function with only 2-input NOR gates.
d) Implement f with a 4:16 decoder and additional logic gates.
e) Implement f with a MUX4 (multiplexer with 4 data inputs) and additional logic gates.

* **Question 1**: a) $A = 88$

88 | 2
0 44 | 2
0 22 | 2
0 11 | 2
1 5 | 2
1 2 | 2
0 1

$\Rightarrow 88_{10} = 1011000_2 = \boxed{01011000_{2C}}$ 8 bits

$B = -50$

50 | 2
0 25 | 2
1 12 | 2
0 6 | 2
0 3 | 2
1 1

$\Rightarrow 50_{10} = 110010_2 = 0110010_{2C}$

$\Rightarrow$ Therefore $-50_{10} = \boxed{1001110_{2C}}$ 7 bits

b) $A - B = 88 - (-50) = 88 + 50$

$$
\begin{array}{rl}
 & 11 \\
 & 01011000 \quad (+88) \\
+ & \boxed{0}0110010 \quad (+50) \\
\hline
 & \boxed{1}\boxed{0001010} \quad (+138)
\end{array}
$$

Adding zeros because this is a positive number

There is overflow, because the sum of 2 positive numbers cannot be equal to a negative number. We can solve this using 9 bits for the operation

$A + B = 88 - 50$

$$
\begin{array}{rl}
 & 111 \\
 & 01011000 \quad (+88) \\
 & \boxed{1}1001110 \quad (-50) \\
\hline
\cancel{} & \boxed{0}\boxed{0100110} \quad (+38)
\end{array}
$$

Adding ones because this is a positive number

There is no overflow, because the result sign is perfecty ok

* **Question 2**: a) 3-bit Gray's code

0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

1-bit Gray's code

2-bit Gray's code

zeros above

Mirror

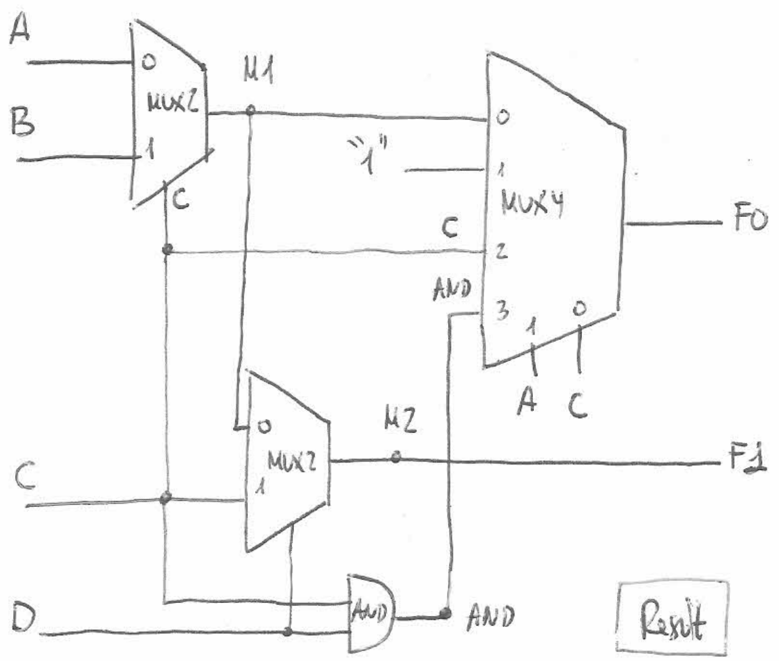
ones below

Mirror

b) 3-bit Johnson's code

0	0	0
0	0	1
0	1	1
1	1	1
1	1	0
1	0	0

* Question 3 :



Result → M2 = F1

Result → F0

A	B	C	D	M1	M2 = F1	AND	F0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1
0	0	1	1	0	1	1	1
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	1	1	0	1
0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0
1	0	0	1	1	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	0	1	1	1
1	1	0	0	1	1	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	1	0	0
1	1	1	1	1	1	1	1

F0 = M1 when A=0, C=0
F0 = 1 when A=0, C=1
F0 = C when A=1, C=0
F0 = AND when A=1, C=1

M1 = A when C=0
M1 = B when C=1

F1 = M1 when D=0
F1 = C when D=1

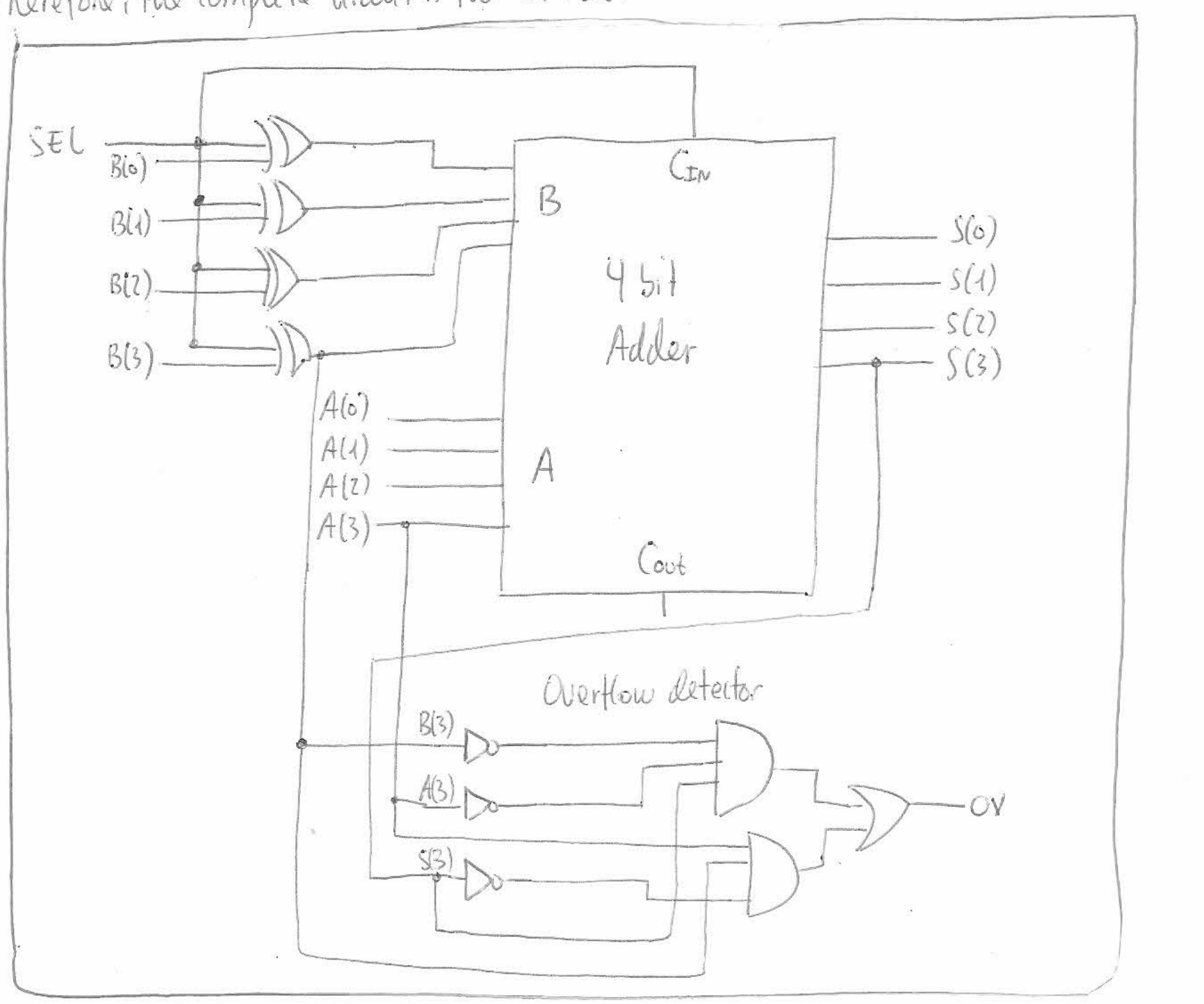
AND = C * D

* Question 4 : + You have to get the adder / substractor circuit using the chip given. The trick is to add the neccesary gates to this chip in order to get the desired circuit.

+ To substract in 2-complement, first you have to get the 2-complement of a variable (for example, using the inversion of this variable and adding "1"
+ Now you have to add A and B using the select input to decide if you want to add A and B (addition) or to add A in 2-complement with B (substraction)
+ You can obtain all of this using the SEL input and every bit of B with 4 XOR gates (connecting the 4 XOR outputs to the 4 B bits in the 4-bit adder) and besides connecting the SEL input to the C_{IN} input of the 4-bit adder, so that:
  - → If SEL=0, B is not inverted, $C_{IN}=0$ and the operation is A+B, but
  - → If SEL=1, B is inverted, $C_{IN}=1$ and the operation is A−B
+ To obtain the OV circuit, it is necessary to do the truth table with the most significant bits of the Operands A(3), B(3) and S(3), considering that there is overflow when the sign changes, that means S(3)=1 (negative number) when A(3)=0 and B(3)=0 (positive numbers), or S(3)=0 (positive number) when A(3)=1 and B(3)=1 (negative numbers):

$$OV = \overline{A(3)}\cdot\overline{B(3)}\cdot S(3) + A(3)\cdot B(3)\cdot\overline{S(3)}$$

+ Therefore, the complete circuit is the next one:



Problem: $f = f(a,b,c,d) = \bar{a}\cdot c\cdot d + \bar{b}(a+c+\bar{d}) = \bar{a}cd + a\bar{b} + \bar{b}c + \bar{b}\bar{d}$

- $\bar{a}\cdot c\cdot d = \bar{a}\cdot c\cdot d\,(b+\bar{b}) = \bar{a}bcd + \bar{a}\bar{b}cd$
- $a\bar{b} = a\bar{b}(c+\bar{c})(d+\bar{d}) = (a\bar{b}c + a\bar{b}\bar{c})(d+\bar{d}) = a\bar{b}cd + a\bar{b}c\bar{d} + a\bar{b}\bar{c}d + a\bar{b}\bar{c}\bar{d}$
- $\bar{b}c = \bar{b}c(a+\bar{a})(d+\bar{d}) = (a\bar{b}c + \bar{a}\bar{b}c)(d+\bar{d}) = a\bar{b}cd + a\bar{b}c\bar{d} + \bar{a}\bar{b}cd + \bar{a}\bar{b}c\bar{d}$
- $\bar{b}\bar{d} = \bar{b}\bar{d}(a+\bar{a})\cdot(c+\bar{c}) = (a\bar{b}\bar{d} + \bar{a}\bar{b}\bar{d})(c+\bar{c}) = a\bar{b}c\bar{d} + a\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}\bar{c}\bar{d}$

Therefore: $f = \bar{a}bcd + \bar{a}\bar{b}cd + a\bar{b}cd + a\bar{b}c\bar{d} + a\bar{b}\bar{c}d + a\bar{b}\bar{c}\bar{d} +$
$\cancel{a\bar{b}cd} + \cancel{a\bar{b}c\bar{d}} + \cancel{\bar{a}\bar{b}cd} + \bar{a}\bar{b}c\bar{d} + \cancel{a\bar{b}c\bar{d}} + \cancel{a\bar{b}\bar{c}\bar{d}} + \cancel{\bar{a}\bar{b}c\bar{d}} + \bar{a}\bar{b}\bar{c}\bar{d}$

So:

truth table

a	b	c	d	f	position
0	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	1	2
0	0	1	1	1	3
0	1	0	0	0	4
0	1	0	1	0	5
0	1	1	0	0	6
0	1	1	1	1	7
1	0	0	0	1	8
1	0	0	1	1	9
1	0	1	0	1	10
1	0	1	1	1	11
1	1	0	0	0	12
1	1	0	1	0	13
1	1	1	0	0	14
1	1	1	1	0	15

and

ab \ cd	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

a) Sum of products → 1st canonical form:

ab \ cd	00	01	11	10
00	1		1	1
01			1	
11				
10	1	1	1	1

→ $b\bar{d}$

→ $\bar{a}\cdot c\cdot d$

→ $a\bar{b}$

$$\boxed{f = b\bar{d} + a\bar{b} + \bar{a}cd}$$

b) Product of sums → 2nd canonical form:

ab \ cd	00	01	11	10
00	1		1	1
01			1	
11				
10	1	1	1	1

→ $a+c+\bar{d}$

→ $\bar{b}+d$

→ $\bar{a}+\bar{b}$

$$\boxed{f = (\bar{a}+\bar{b})\cdot(\bar{b}+d)\cdot(a+c+\bar{d})}$$

c) We use the first canonical form:

$$f = b\bar{c} + a\bar{b} + \bar{a}cd = \overline{\overline{b\bar{c}}} + \overline{\overline{a\bar{b}}} + \overline{\overline{\bar{a}cd}} = \overline{\overline{\overline{\bar{b}+c} + \overline{\bar{a}+b} + \overline{a+\bar{c}+\bar{d}}}}$$

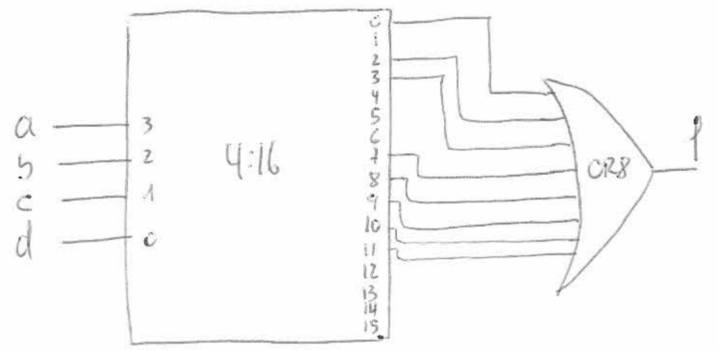
- The not gate is implemented with a NOR gate with the same inputs =>
- The NOR with 3 inputs could be implemented as, for example, $(a+\bar{c}) + \bar{d}$ and the AND could be implemented as NOR + NOT
- Therefore the solution is:

$$\boxed{f = \overline{\overline{\overline{\bar{b}+c} + \overline{\bar{a}+b} + \overline{\overline{\overline{(a+\bar{c})}} + \bar{d}}}}}$$

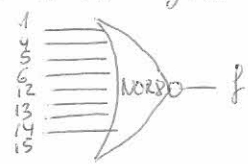
NOT, NOR1, NOR2, NOT, NOR3, NOT, NOR6, NOR4, NOR5

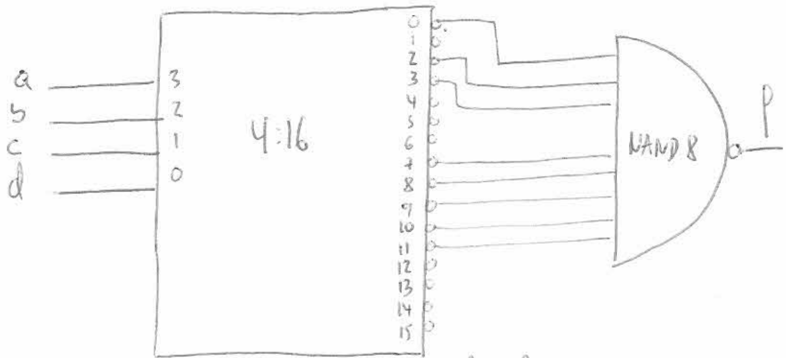
d) $f = \sum_4 (0, 2, 3, 7, 8, 9, 10, 11)$, so looking at the truth table there are 4 solutions

with high-active output decoder

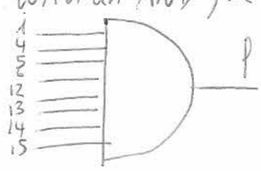


the dual case would be to connect the outputs 1, 4, 5, 6, 12, 13, 14, 15 with a NOR gate





the dual case would be to connect the outputs 1, 4, 5, 6, 12, 13, 14, 15 with an AND gate.



with low-active output decoder

e) Looking at the truth table: