Solutions - Midterm Exam

(October 15th @ 5:30 pm)

Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (22 PTS)

a) Complete the following table. The decimal numbers are unsigned: (3 pts.)

Decimal	BCD	Binary	Reflective Gray Code
50	01010000	110010	101011
128	000100101000	10000000	11000000

b) Complete the following table. The decimal numbers are signed. Use the fewest number of bits in each case: (15 pts.)

	REPRESENTATION		
Decimal	**Sign-and-magnitude**	**1's complement**	**2's complement**
-17	110001	101110	101111
-16	110000	101111	10000
-1	11	10	1111
41	0101001	0101001	0101001
-37	1100101	1011010	1011011

c) Convert the following decimal numbers to their 2's complement representations. (4 pts)

✓ -9.25
+9.25 = 01001.01 ⇒ -17.25= 10110.11

✓ 8.75
+8.75 = 01000.11

PROBLEM 2 (10 PTS)

- Sketch the circuit that computes $|A-B|$, where A, B are 4-bit signed (2C) numbers. For example, $A = 0101, B = 1101 \rightarrow |A-B| = |5-(-3)| = 8$. You can only use full adders (or multi-bit adders) and logic gates. Your circuit must avoid overflow: design your circuit so that the result and intermediate operations have the proper number of bits.

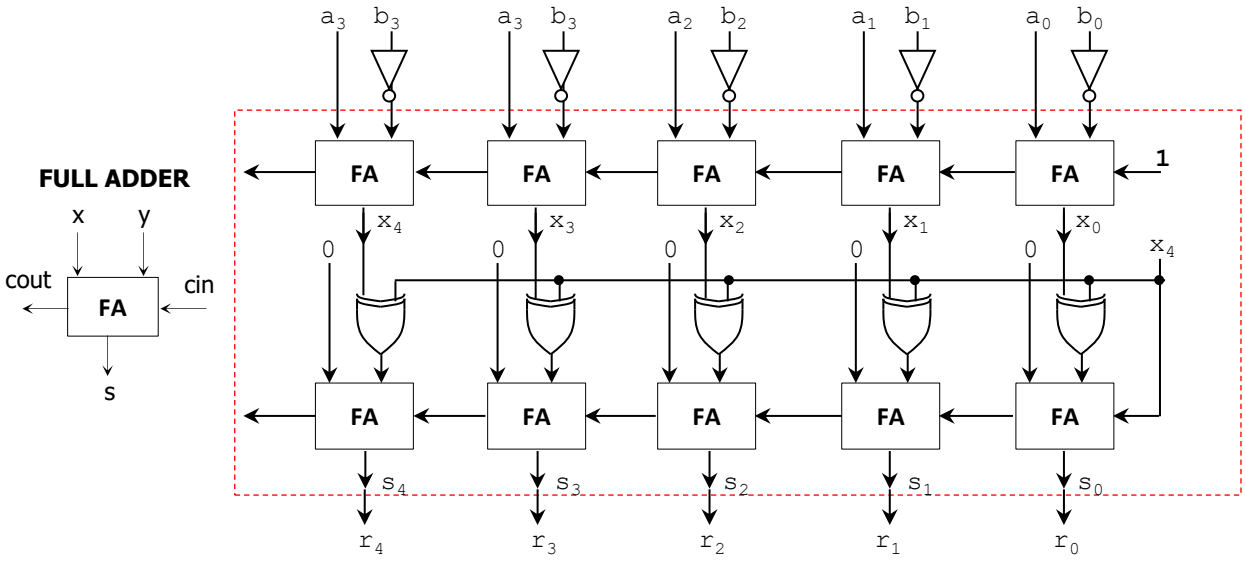
$A = a_3a_2a_1a_0$, $B = b_3b_2b_1b_0$
$A, B \in [-8,7] \rightarrow A, B$ require 4 bits in 2C representation.

✓ $X = A - B \in [-15,15]$ requires 5 bits in 2C. Thus, we need to zero-extend A and B.

✓ $|X| = |A-B| \in [0,15]$ requires 5 bits in 2C. Thus, the second operation $0 \pm X$ only requires 5 bits.

- If $x_4 = 1 \rightarrow X < 0 \rightarrow$ we do $0 - X$.
- If $x_4 = 0 \rightarrow X \geq 0 \rightarrow$ we do $0 + X$.

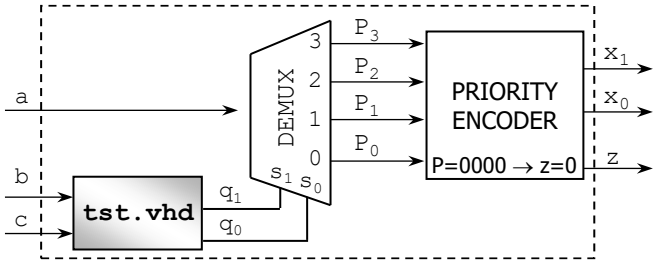
✓ $R = |A-B| \in [0,15]$ requires 5 bits in 2C. Note that the MSB is always 0.



PROBLEM 3 (15 PTS)

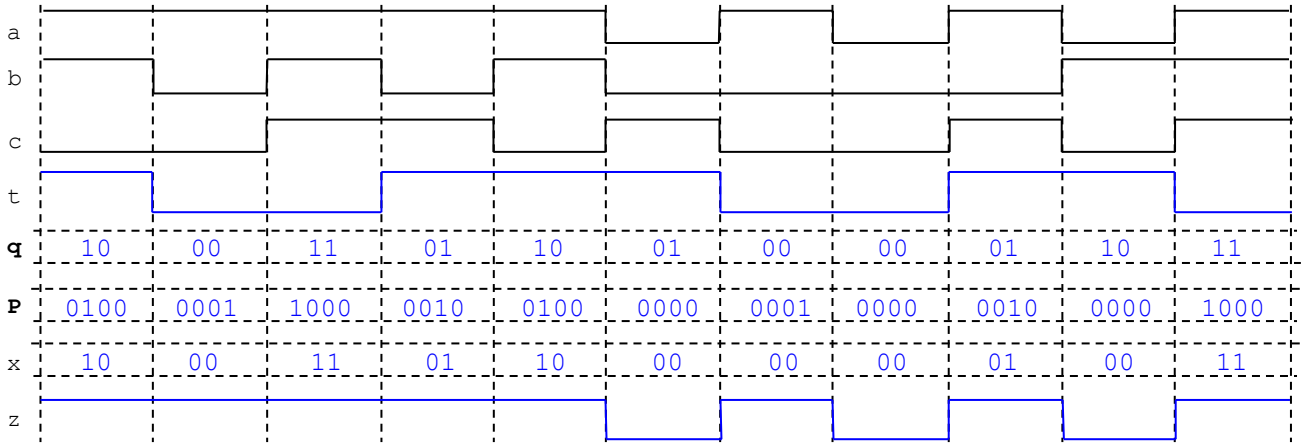
- Complete the timing diagram of the following circuit. The VHDL code (`tst.vhd`) corresponds to the shaded circuit.

$q = q_1q_0$, $P = p_3p_2p_1p_0$, $x = x_1x_0$



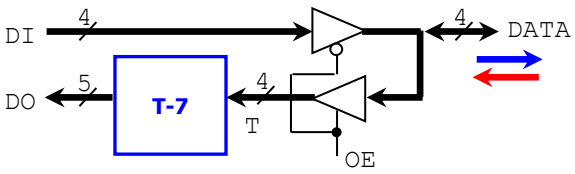

```
library ieee;
use ieee.std_logic_1164.all;

entity tst is
  port (b, c : in std_logic;
        q: out std_logic_vector(1 downto 0));
end tst;
```

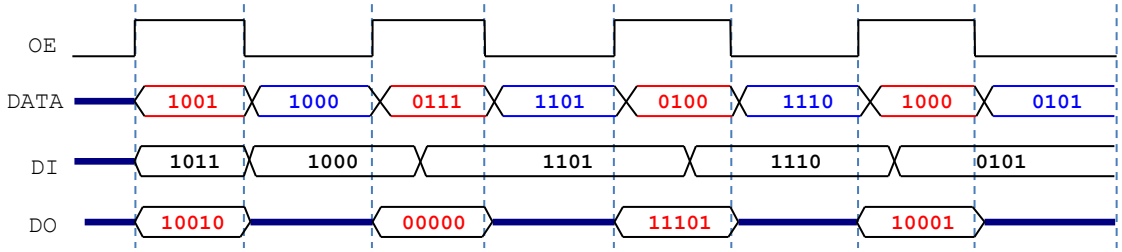
```
architecture bhv of tst is

   signal t: std_logic;

begin

  t <= b xor c;
  process (b,c,t)
  begin
     q <= b & c;
     if t = '0' then
        q <= c & b;
     end if;
  end process;

end bhv;
```



PROBLEM 4 (11 PTS)

- Complete the timing diagram (signals *DO* and *DATA*) of the following circuit. The circuit in the blue box computes the signed operation T-7, with the result having 5 bits. T is a 4-bit signed (2C) number.
  - ✓ Example: if T=1010:
    DO = 1010-0111 = 11010 + 11001
    DO = 10011



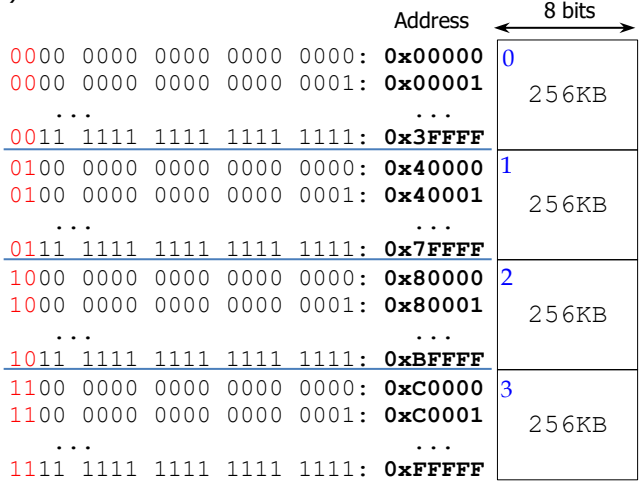


PROBLEM 5 (10 PTS)

- A microprocessor has a memory space of 1 MB. Each memory address occupies one byte. 1KB = 2^{10} bytes, 1MB = 2^{20} bytes, 1GB = 2^{30} bytes.
  a) What is the address bus size (number of bits of the address) of the microprocessor?
  Size of memory space: 1 MB = 2^{20} bytes. Thus, we require 20 bits to address the memory space.

  b) What is the range (lowest to highest, in hexadecimal) of the memory space for this microprocessor? (1 pt.)
  With 20 bits, the address range is `0x00000` to `0xFFFFF`.

  c) The figure to the right shows four memory chips that are placed in the given positions:
  ✓ Complete the address ranges (lowest to highest, in hexadecimal) for each of the memory chips. (8 pts)



PROBLEM 6 (15 PTS)

a) Perform the following additions and subtractions of the following unsigned integers. Use the fewest number of bits n to represent both operators. Indicate every carry (or borrow) from c_0 to c_n (or b_0 to b_n). For the addition, determine whether there is an overflow. For the subtraction, determine whether we need to keep borrowing from a higher bit. (6 pts)

✓ `37 + 41`

	c_6=1	c_5=0	c_4=0	c_3=0	c_2=0	c_1=1	c_0=0
37 = 0x25 =	1	0	0	1	0	1	+
41 = 0x29 =	1	0	1	0	0	1	
Overflow! →	1	0	0	1	1	1	0

✓ `37 - 41`

	Borrow out! →	b_6=1	b_5=1	b_4=1	b_3=0	b_2=0	b_1=0	b_0=0
37 = 0x25 =		1	0	0	1	0	1	-
41 = 0x29 =		1	0	1	0	0	1	
		1	1	1	1	0	0	

b) For the decimal numbers in the figure, perform the signed (2C) 8-bit addition. The operands must be represented in 2's complement arithmetic with 8 bits. Also, complete all the carries and summation bits. Indicate the corresponding decimal number of the 8-bit result.

Decimal values		c_8	c_7	c_6	c_5	c_4	c_3	c_2	c_1	c_0	
		1	1	1	1	1	1	1	0	0	
-41	=		1	1	0	1	0	1	1	1	+
58	=		0	0	1	1	1	0	1	0	
17	=		0	0	0	1	0	0	0	1	

Does this 8-bit operation incur in overflow? Yes ~~No~~
Value of the overflow bit: $c_8 \oplus c_7 = 0$
Value of carry out bit: $c_8 = 1$

c) Perform binary multiplication of the following numbers (they must be represented in 2's complement arithmetic). (3 pts)
✓ `-7 x 9`

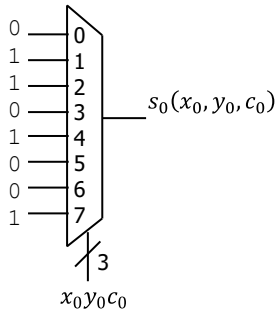
```
  1 0 0 1 x          1 0 0 1 x
0 1 0 0 1            0 1 1 1
                   ---------
                     1 0 0 1
                   1 0 0 1
                 1 0 0 1
               0 0 0 0
             ---------------
             0 1 1 1 1 1 1
                   ↓
             1 0 0 0 0 0 1
```

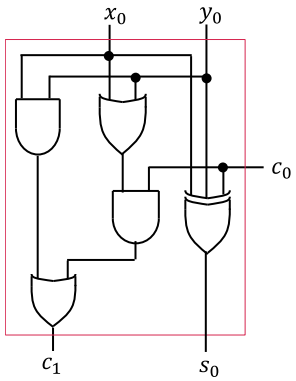
PROBLEM 7 (17 PTS)

- Given the circuit in the figure:
  - ✓ Implement s_0 using ONLY an 8-to-1 MUX. (5 pts.)

$s_0(x_0, y_0, c_0) = x_0 \oplus y_0 \oplus c_0$

x_0	y_0	c_0	c_1	s_0
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1





  - ✓ Implement c_1 using ONLY 2-to-1 MUXs (AND, OR, NOT, XOR gates are not allowed). (12 pts)

$c_1(x_0, y_0, c_0) = x_0 y_0 + x_0 c_0 + y_0 c_0$

$c_1(x_0, y_0, c_0) = \overline{x_0} c_1(0, y_0, c_0) + x_0 c_1(1, y_0, c_0) = \overline{x_0}(y_0 c_0) + x_0(y_0 + c_0 + y_0 c_0)$
$c_1(x_0, y_0, c_0) = \overline{x_0} g(y_0, c_0) + x_0 h(y_0, c_0)$

$g(y_0, c_0) = \overline{y_0} g(0, c_0) + y_0 g(1, c_0) = \overline{y_0}(0) + y_0(c_0)$
$h(y_0, c_0) = \overline{y_0} h(0, c_0) + y_0 h(1, c_0) = \overline{y_0}(c_0) + y_0(1)$