Solutions - Midterm Exam

(October 19th @ 5:30 pm)

Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (20 PTS)

a) Complete the following table. The decimal numbers are unsigned: (5 pts.)

Decimal	BCD	Binary	Reflective Gray Code
37	00110111	100101	110111
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: (12 pts.)

	REPRESENTATION		
Decimal	Sign-and-magnitude	1's complement	2's complement
-17	110001	101110	101111
-64	11000000	10111111	1000000
-32	1100000	1011111	100000
-24	111000	100111	101000
41	0101001	0101001	0101001
-36	1100100	1011011	1011100

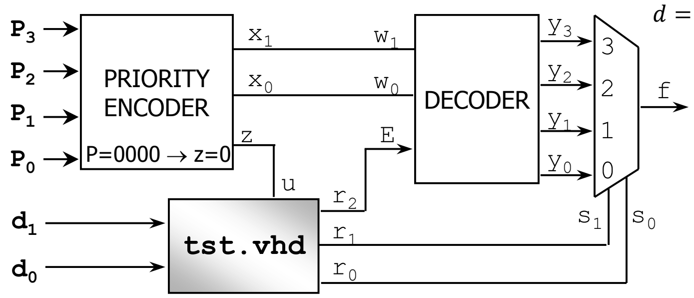
c) Convert the following decimal numbers to their 2's complement representations. (3 pts)

- ✓ -17.125
  +17.125 = 010001.001 ⇒ -17.125= 101110.111
- ✓ 32.375
  +32.375 = 0100000.011

PROBLEM 2 (15 PTS)

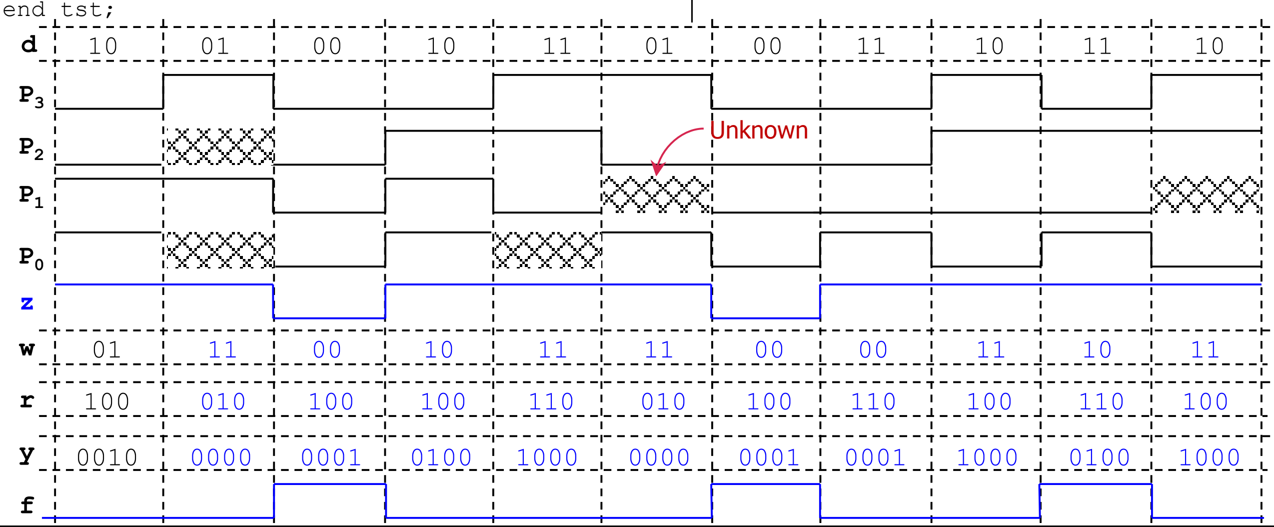
- Complete the timing diagram of the following circuit. The VHDL code (tst.vhd) corresponds to the shaded circuit.

$d = d_1d_0$, $w = w_1w_0$, $r = r_2r_1r_0$, $y = y_3y_2y_1y_0$



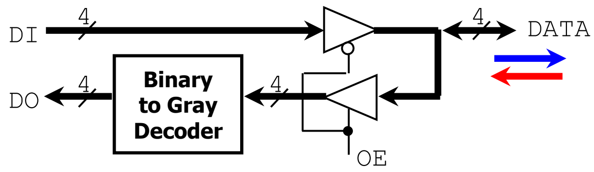

```
library ieee;
use ieee.std_logic_1164.all;
entity tst is
  port (d: in std_logic_vector(1 downto 0);
        r: out std_logic_vector(2 downto 0);
        u: in std_logic);
end tst;
```

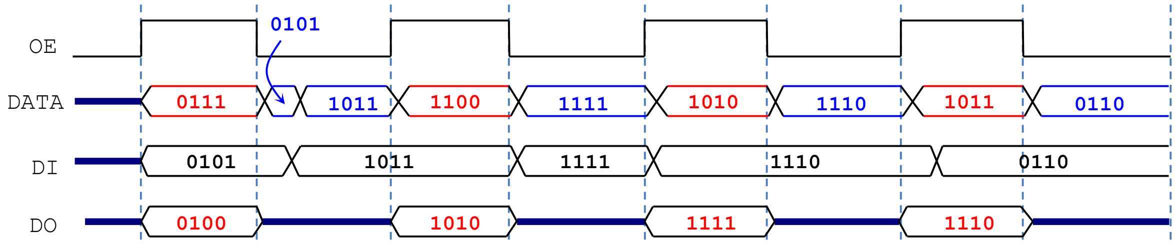
```
architecture bhv of tst is

begin

  process (d, u)
  begin

     r <= '1'&d;
     if u = '1' then
        r <= d&'0';
     end if;

  end process;

end bhv;
```



PROBLEM 3 (8 PTS)

- Complete the timing diagram (signals *DO* and *DATA*) of the following circuit. The 4-bit binary to gray decoder treats input data as unsigned numbers.





PROBLEM 4 (12 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. We want to connect four 256 KB memory chips to this microprocessor.
  - ✓ What is the address bus size (number of bits of the address) of the microprocessor? (1 pt).

    Size of memory space: 1 MB = 2^{20} bytes. Thus, we require 20 bits to address the memory space.

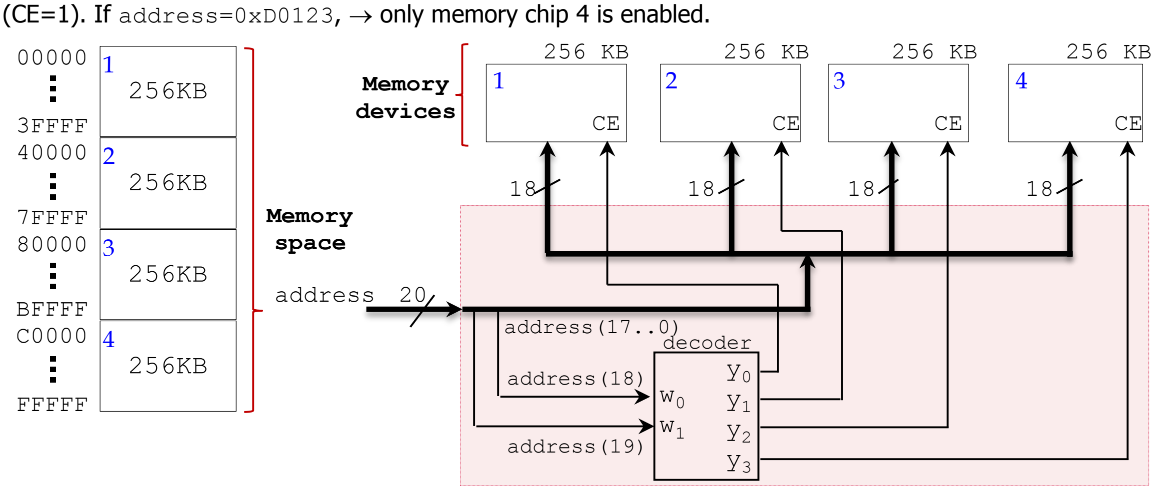
  - ✓ For a memory chip of 256 KB, how many bits do we require to address 256 KB of memory? (1 pt).

    256 KB memory device: 256KB = 2^{18} bytes. Thus, we require 18 bits to address the memory device.

  - ✓ Complete the address ranges (lowest to highest, in hexadecimal) for each of the memory chips in the figure. (4 pts).

Address (binary)	Address (hex)	Chip (8 bits)
0000 0000 0000 0000 0000	0x00000	1: 256KB
0000 0000 0000 0000 0001	0x00001	
...	...	
0011 1111 1111 1111 1111	0x3FFFF	
0100 0000 0000 0000 0000	0x40000	2: 256KB
0100 0000 0000 0000 0001	0x40001	
...	...	
0111 1111 1111 1111 1111	0x7FFFF	
1000 0000 0000 0000 0000	0x80000	3: 256KB
1000 0000 0000 0000 0001	0x80001	
...	...	
1011 1111 1111 1111 1111	0xBFFFF	
1100 0000 0000 0000 0000	0xC0000	4: 256KB
1100 0000 0000 0000 0001	0xC0001	
...	...	
1111 1111 1111 1111 1111	0xFFFFF	

  - ✓ Sketch the circuit that: i) addresses the memory chips, and ii) enables only one memory chip (via CE: chip enable) when the address falls in the corresponding range. Example: if `address=0x5FFFF`, → only memory chip 2 is enabled (CE=1). If `address=0xD0123`, → only memory chip 4 is enabled.



PROBLEM 5 (17 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 byte. (6 pts)

✓ 17 + 50

```
                  c6=1 c5=1 c4=0 c3=0 c2=0 c1=0 c0=0

50 = 0x32 =            1    1    0    0    1    0  +
17 = 0x11 =            0    1    0    0    0    1
------------------------------------------------
Overflow! →       1    0    0    0    0    1    1
```

✓ 37 - 41

```
Borrow out! →     b6=1 b5=1 b4=1 b3=0 b2=0 b1=0 b0=0

37 = 0x25 =            1    0    0    1    0    1  -
41 = 0x29 =            1    0    1    0    0    1
------------------------------------------------
                       1    1    1    1    0    0
```

b) Perform the following operations, where numbers are represented in 2's complement. Indicate every carry from c_0 to c_n. For each case, use the fewest number of bits to represent the summands and the result so that overflow is avoided. (8 pts)

✓ -36 + 50

```
n = 7 bits        c7=1 c6=1 c5=1 c4=0 c3=0 c2=0 c1=0 c0=0

c7⊕c6=0       -36 =    1    0    1    1    1    0    0  +
No Overflow    50 =    0    1    1    0    0    1    0
--------------------------------------------------------
               14 =    0    0    0    1    1    1    0

-36 + 50 = 14 ∈ [-2^6, 2^6-1] → no overflow
```

✓ -41 - 24

```
n = 7 bits        c7=1 c6=0 c5=0 c4=0 c3=0 c2=0 c1=0 c0=0

c7⊕c6=1       -41 =    1    0    1    0    1    1    1  +
Overflow!     -24 =    1    1    0    1    0    0    0
--------------------------------------------------------
                       0    1    1    1    1    1    1

-41 -24 = -65 ∉ [-2^6, 2^6-1] → overflow!
```

To avoid overflow: n = 8 bits (sign-extension)

```
c8⊕c7=0          c8=1 c7=1 c6=0 c5=0 c4=0 c3=0 c2=0 c1=0 c0=0
No Overflow
             -41 =      1    1    0    1    0    1    1    1  +
             -24 =      1    1    1    0    1    0    0    0
--------------------------------------------------------------
                        1    0    1    1    1    1    1    1

-41 -24 = -65 ∈ [-2^7, 2^7-1] → no overflow
```

c) Perform binary multiplication of the following numbers that are 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 6 (10 PTS)

- Given two 4-bit unsigned numbers A, B, sketch the circuit that computes $|A-2B|$. For example: $A = 1010, B = 1110 \rightarrow |A-2B| = |10-2\times 14| = 18$. You can only use full 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.

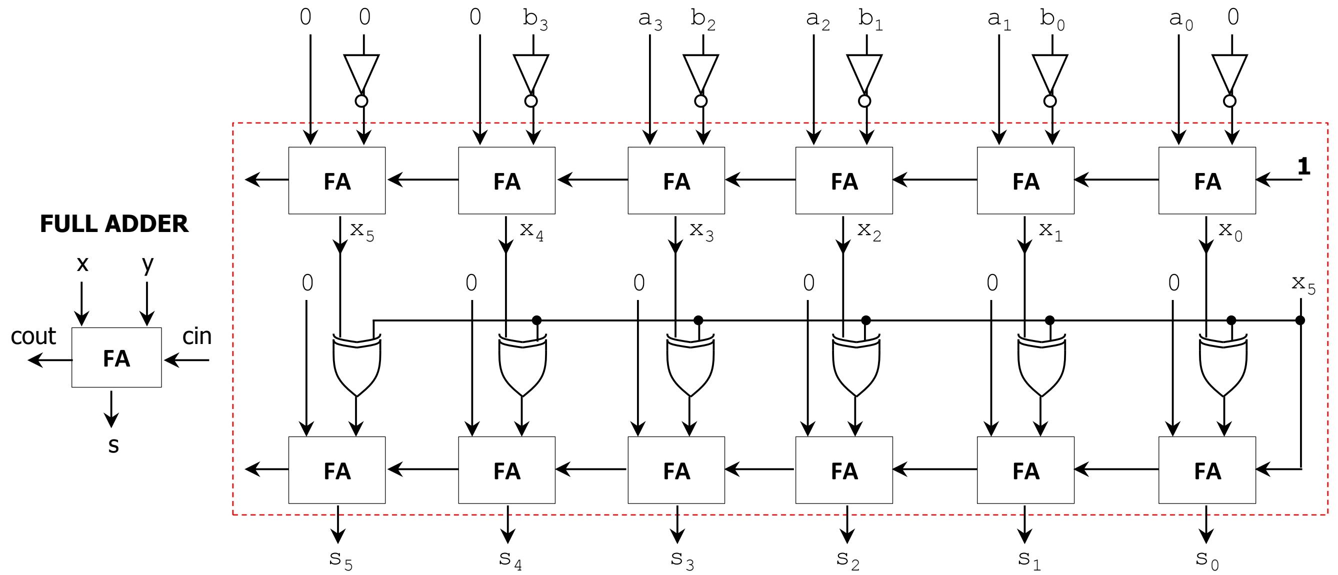
$A = a_3a_2a_1a_0$, $B = b_3b_2b_1b_0$: unsigned numbers
$A = 0a_3a_2a_1a_0$, $B = 0b_3b_2b_1b_0$: signed numbers (2C)

$A, B \in [0,15] \rightarrow 2B \in [0,30]$ requires 6 bits in 2C.

✓ $X = A - 2B \in [-30,15]$ requires 6 bits in 2C. Thus, the operation $A-2B$ requires 6 bits (we sign-extend A).

✓ $|X| = |A-2B| \in [0,30]$ requires 6 bits in 2C. Thus, the second operation $0 \pm X$ only requires 6 bits.

- If $x_5 = 1 \rightarrow X < 0 \rightarrow$ we do $0 - X$.
- If $x_5 = 0 \rightarrow X \geq 0 \rightarrow$ we do $0 + X$.



PROBLEM 7 (18 PTS)

- Sketch the circuit that implements the following Boolean function: $f(a,b,c,d) = (a \oplus \bar{b})(c \oplus d)$
  - ✓ Using <u>ONLY</u> 2-to-1 MUXs (AND, OR, NOT, XOR gates are not allowed). (12 pts)

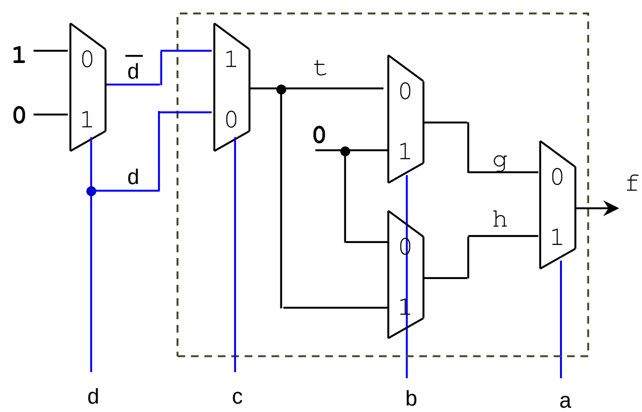
$f(a,b,c,d) = \bar{a}f(0,b,c,d) + af(1,b,c,d) = \bar{a}\left(\bar{b}(c \oplus d)\right) + a\left(b(c \oplus d)\right) = \bar{a}g(b,c,d) + ah(b,c,d)$

$g(b,c,d) = \bar{b}g(0,c,d) + bg(1,c,d) = \bar{b}(c \oplus d) + b(0)$

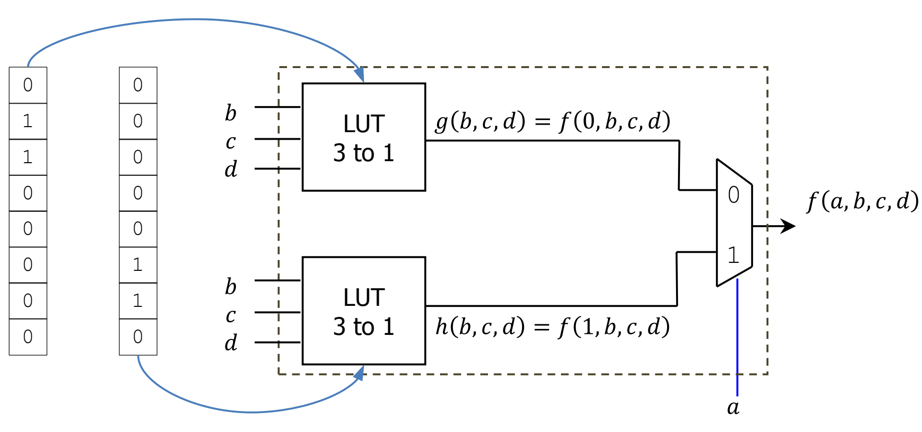
$h(b,c,d) = \bar{b}h(0,c,d) + bh(1,c,d) = \bar{b}(0) + b(c \oplus d)$

$t(c,d) = c \oplus d = \bar{c}t(0,d) + ct(1,d) = \bar{c}(d) + c(\bar{d})$

Also: $\bar{d} = \bar{d}(1) + d(0)$



  - ✓ Using two 3-to-1 LUTs and a 2-to-1 MUX. Specify the contents of each of the 3-to-1 LUTs. (6 pts)



a	b	c	d	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0