Solutions - Midterm Exam

(February 18th @ 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: (6 pts.)

Decimal	BCD	Binary	Reflective Gray Code
97	10010111	1100001	1010001
51	01010001	110011	101010
98	10011000	1100010	1010011
156	000101010110	10011100	11010010

b) Complete the following table. Use the fewest number of bits in each case: (12 pts.)

	REPRESENTATION		
Decimal	Sign-and-magnitude	1's complement	2's complement
-32	1100000	1011111	100000
-76	11001100	10110011	10110100
-33	1100001	1011110	1011111
69	01000101	01000101	01000101
-64	11000000	10111111	1000000
-19	110011	101100	101101

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

- ✓ -31.3125
  +31.3125 = 011111.0101
  ⇒ 100000.1011
- ✓ 17.375
- ✓ +17.375 = 010001.011

PROBLEM 2 (10 PTS)

- The figure below depicts the entire memory space of a microprocessor. Each memory address occupies one byte. 1KB = 2^{10} bytes, 1MB = 2^{20} bytes, 1GB = 2^{30} bytes
  - ✓ What is the size (in bytes, KB, or MB) of the memory space? What is the address bus size of the microprocessor?

    Address space: 0x000000 to 0xFFFFFF. To represents all these addresses, we require 24 bits. So, the address bus size of the microprocessor is 24 bits. The size of the memory space is 2^{24} = 16 MB

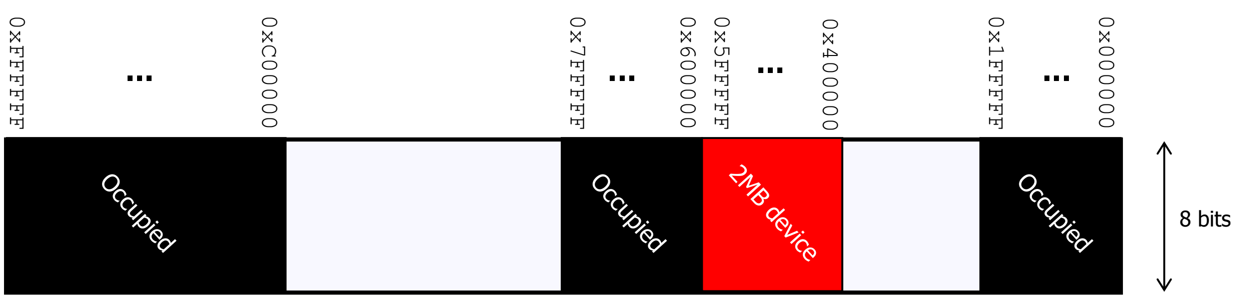
  - ✓ If we have a memory chip of 2 MB, how many bits do we require to address those 2 MB of memory?

    2 MB = 2^{21} bytes. Thus, we require 21 bits to address the memory device.

  - ✓ We want to connect the 2 MB memory chip to the microprocessor. The figure shows all the occupied portions of the memory space. Provide an address range so that 2 MB of memory is properly addressed. You can only use the non-occupied portions of the memory space as shown in the figure below.

    2 MB of memory require 21 bits. The 21-bit address range would be from 0x00000 to 0x1FFFFF. Within the entire 24-bit memory space. Any 24-bit range, where the 21 LSBs go from 0x00000 to 0x1FFFFF, would be valid: this results in 8 valid ranges. However, there are occupied portions in the figure, leaving only four possible ranges:

    - 0x200000 to 0x3FFFFF
    - 0x400000 to 0x5FFFFF → we pick this one!
    - 0x800000 to 0x9FFFFF
    - 0xA00000 to 0xBFFFFF



PROBLEM 3 (12 PTS)

- Given two 4-bit <u>unsigned</u> numbers A, B, sketch the circuit that computes $|A-B| \times 4$. For example: $A = 0011, B = 1010 \rightarrow |A-B| = 7$, $|A-B| \times 4 = 28$. You can only use full adders and logic gates. Make sure your circuit avoids overflow.

$A = a_3a_2a_1a_0$, $B = b_3b_2b_1b_0$

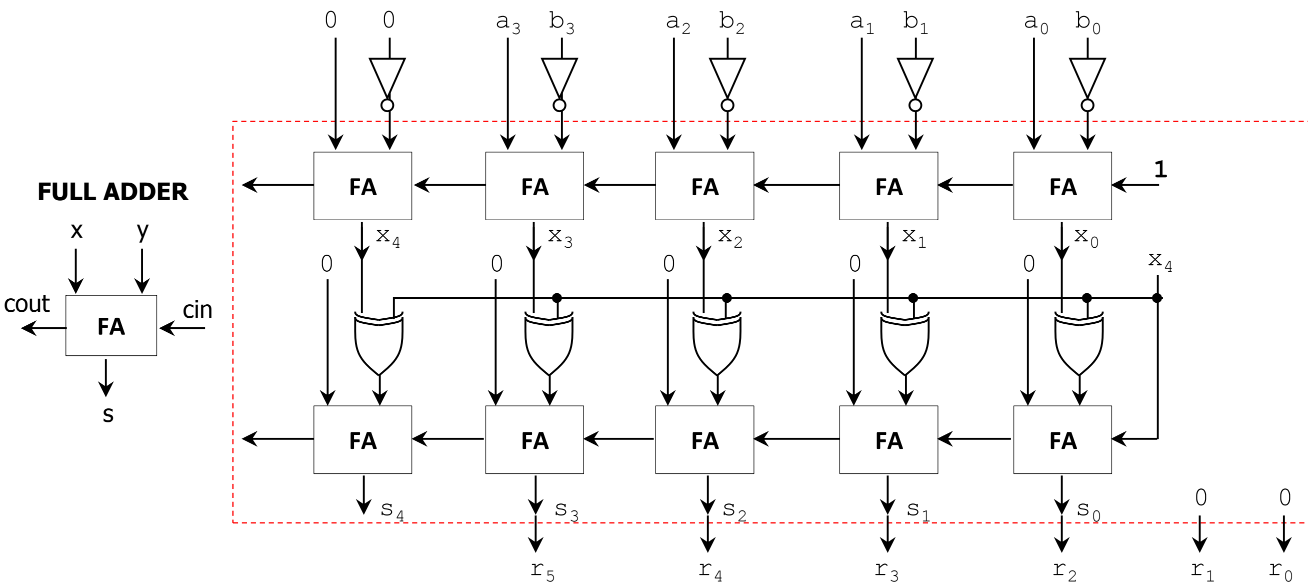
$A, B \in [0,15] \rightarrow A, B$ require 4 bits in unsigned representation. However, to get the proper result of $A - B$, we need to use the 2C representation, where A, B require 5 bits in 2C.

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

✓ $|X| = |A - B| \in [0,15]$ 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 $0 + X$.

✓ $R = |A - B| \times 4 \in [0,60]$ requires 7 bits in 2C. Note that the MSB is always 0. The unsigned result only require 6 bits.



PROBLEM 4 (18 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)

✓ 51 + 27

```
              c6=1 c5=1 c4=0 c3=0 c2=1 c1=1 c0=0
51 = 0x33 =         1    1    0    0    1    1  +
27 = 0x1B =         0    1    1    0    1    1
-----------------------------------------------
Overflow! →    1    0    0    1    1    1    0
```

✓ 19 − 42

```
Borrow out! → b6=1 b5=0 b4=1 b3=0 b2=0 b1=0 b0=0
19 = 0x13 =          0    1    0    0    1    1  -
42 = 0x2A =          1    0    1    0    1    0
------------------------------------------------
                     1    0    1    0    0    1
```

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)

✓ 127 − 76

```
n = 8 bits      c8=1 c7=1 c6=1 c5=1 c4=1 c3=1 c2=0 c1=0 c0=0

c9⊕c8=0         127 = 0 1 1 1 1 1 1 1 +
No Overflow     -76 = 1 0 1 1 0 1 0 0
                ---------------------
                 51 = 0 0 1 1 0 0 1 1

127 - 76 = 51 ∈ [-2^7, 2^7-1] → no overflow
```

✓ −69 − 97

```
n = 8 bits      c8=1 c7=0 c6=1 c5=1 c4=1 c3=1 c2=1 c1=1 c0=0

c7⊕c6=1         -69 = 1 0 1 1 1 0 1 1 +
Overflow!       -97 = 1 0 0 1 1 1 1 1
                ---------------------
                      0 1 0 1 1 0 1 0

-69 -97 = -166 ∉ [-2^7, 2^7-1] → overflow!
```

To avoid overflow: n = 9 bits (sign-extension)

```
c9⊕c8=0         c9=1 c8=1 c7=0 c6=1 c5=1 c4=1 c3=1 c2=1 c1=1 c0=0
No Overflow
                -69 = 1 1 0 1 1 1 0 1 1 +
                -97 = 1 1 0 0 1 1 1 1 1
                -----------------------
               -166 = 1 0 1 0 1 1 0 1 0

-69 -97 = -166 ∈ [-2^8, 2^8-1] → no overflow
```

c) Get the multiplication result of the following numbers that are represented in 2's complement arithmetic with 4 bits. (4 pts)

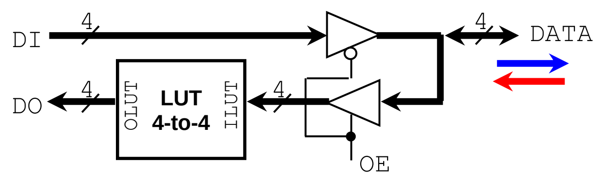
✓ `-7 x 5.`

```
1 0 0 1 x      ➡     0 1 1 1 x
0 1 0 1              0 1 0 1
                   ---------
                     0 1 1 1
                   0 0 0 0
                 0 1 1 1
               0 0 0 0
           ---------------
           0 0 1 0 0 0 1 1
                 ⬇
           1 1 0 1 1 1 0 1
```

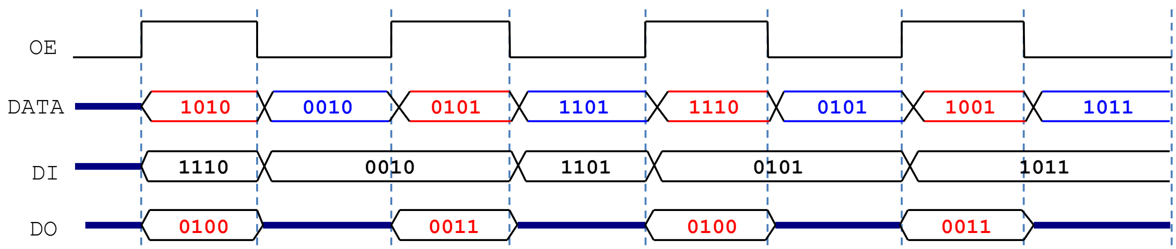
PROBLEM 5 (10 PTS)

- Given the following circuit, complete the timing diagram (signals DO and $DATA$).
  The LUT 4-to-4 implements the following function: $OLUT = \lceil sqrt(ILUT) \rceil$. For example: $ILUT = 1100 \rightarrow OLUT = 0100$

Input data to LUT is treated as an unsigned number.

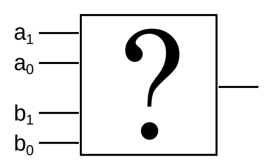


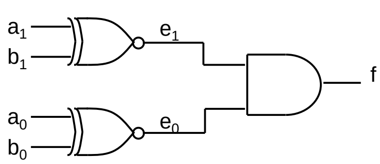

```
ceil(sqrt(10))= 4
ceil(sqrt(5))= 3
ceil(sqrt(14)) = 4
ceil(sqrt(9))= 3
```



PROBLEM 6 (15 PTS)

a) We want to design a circuit that determines whether two 2-bit numbers $A = a_1a_0, B = b_1b_0$ are equal: $f = 1\ if\ A = B, f = 0\ if\ A \neq B$. Sketch this circuit using logic gates. (4 pts)





b) Implement the previous circuit using ONLY 2-to-1 MUXs (AND, OR, NOT, XOR gates are not allowed). (11 pts)

$f(a_1, b_1, a_0, b_0) = \left(\overline{a_1 \oplus b_1}\right)\left(\overline{a_0 \oplus b_0}\right)$

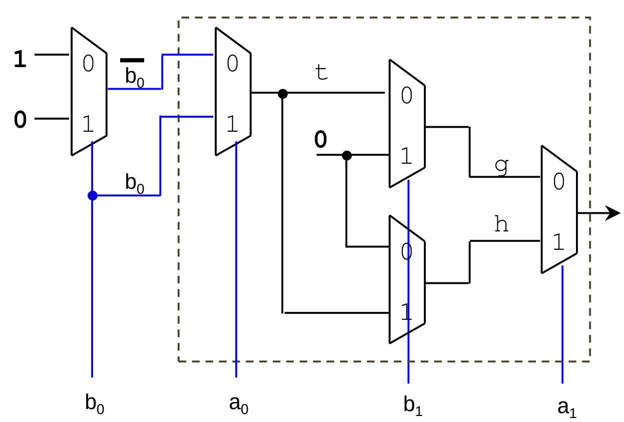
$f = \overline{a_1}f(0, b_1, a_0, b_0) + a_1 f(1, b_1, a_0, b_0) = \overline{a_1}\left(\overline{b_1}\left(\overline{a_0 \oplus b_0}\right)\right) + a_1\left(b_1\left(\overline{a_0 \oplus b_0}\right)\right) = \overline{a_1}g(b_1, a_0, b_0) + a_1 h(b_1, a_0, b_0)$

$g(b_1, a_0, b_0) = \overline{b_1}\left(\overline{a_0 \oplus b_0}\right) + b_1(0)$

$h(b_1, a_0, b_0) = \overline{b_1}(0) + b_1\left(\overline{a_0 \oplus b_0}\right)$

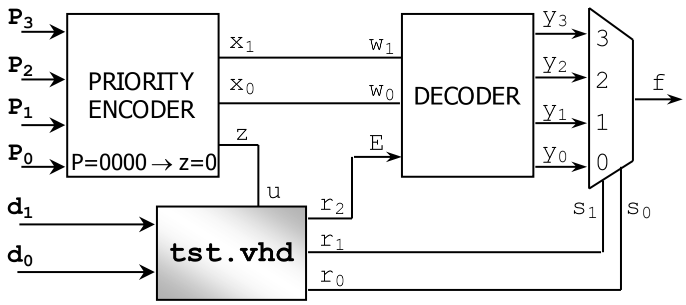
$t(a_0, b_0) = \left(\overline{a_0 \oplus b_0}\right) = \overline{a_0}\left(\overline{b_0}\right) + a_0(b_0)$

Also: $\overline{b_0} = \overline{b_0}(1) + b_0(0)$



PROBLEM 7 (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 <= '0'&d;
     if u = '1' then
        r <= d&'1';
     end if;

  end process;

end bhv;
```