# Solutions - Midterm Exam

(October 17th @ 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 |
|---:|---:|---:|---:|
| 32 | 00110010 | 100000 | 110000 |
| 41 | 01000001 | 101001 | 111101 |
| 125 | 000100100101 | 1111101 | 1000011 |

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 |
| -32 | 1100000 | 1011111 | 100000 |
| 26 | 011010 | 011010 | 011010 |
| -31 | 111111 | 100000 | 100001 |
| -1 | 11 | 10 | 1 |
| 0 | 00 | 111 | 0 |

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

✓ -16.375

+16.375 = 010000.011 ⇒ -16.375 = 101111.101

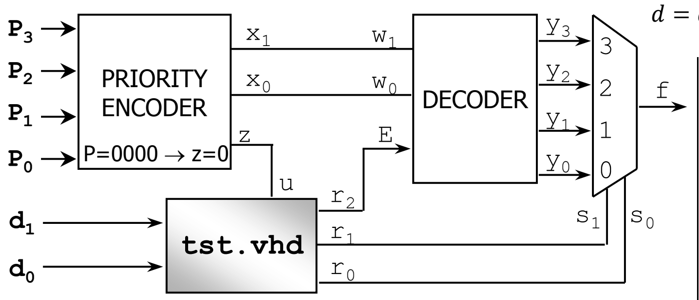
✓ 18.125

+18.125 = 010010.001

## PROBLEM 2 (15 PTS)

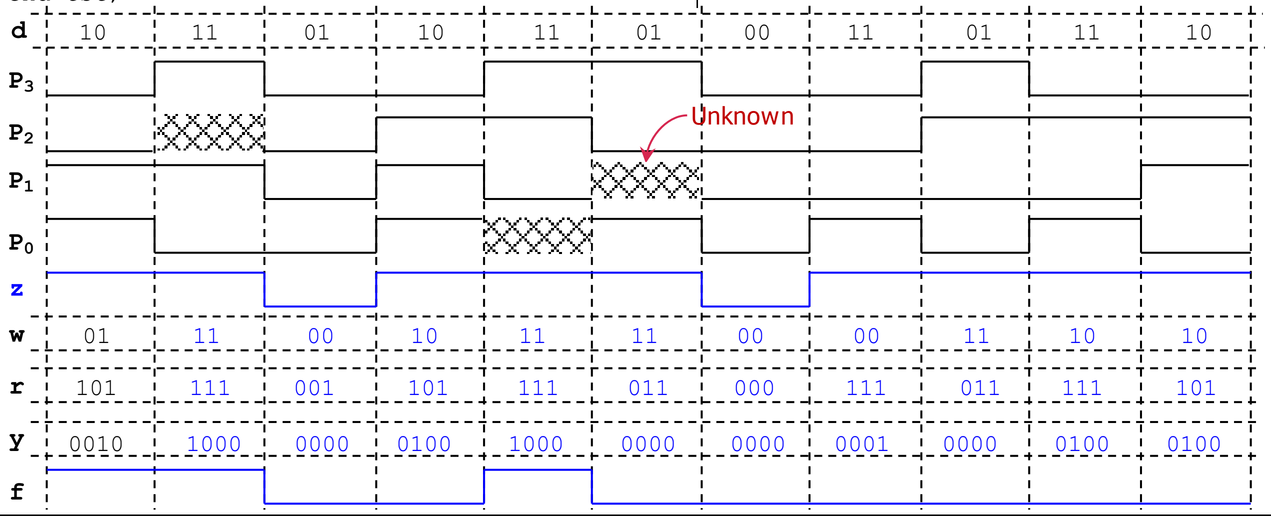
▪ Complete the timing diagram of the following circuit. The VHDL code (`tst.vhd`) corresponds to the shaded circuit.

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



```
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;
```

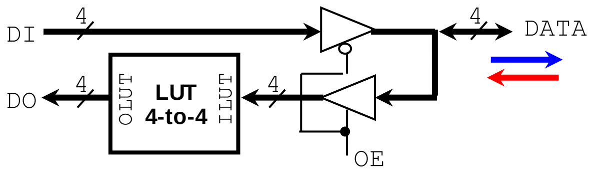
```
architecture bhv of tst is

begin

  process (d, u)
  begin

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

  end process;

end bhv;
```



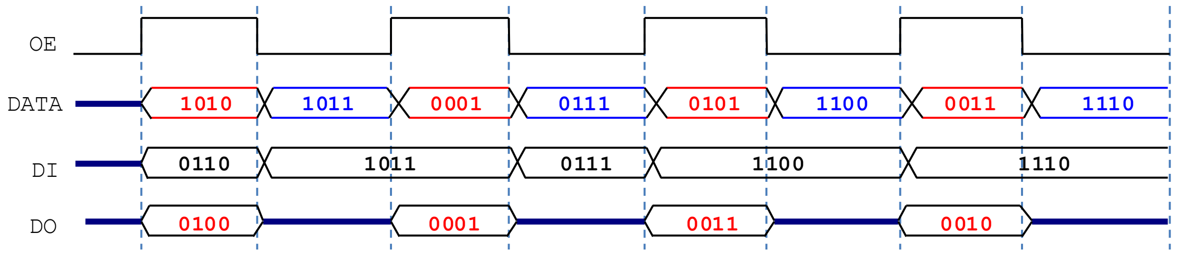
## PROBLEM 3 (10 PTS)

- 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(1)) = 1
ceil(sqrt(5)) = 3
ceil(sqrt(3)) = 2
```



## PROBLEM 4 (11 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? (2 pts.)

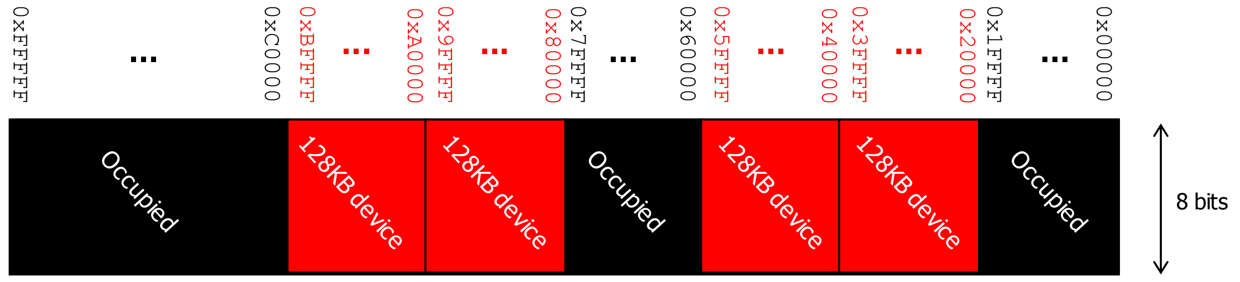
    Address space: `0x00000` to `0xFFFFF`. To represent all these addresses, we require 20 bits. So, the address bus size of the microprocessor is 20 bits. The size of the memory space is then $2^{20}$ = 1 MB.

  - ✓ If we have a memory chip of 128 KB, how many bits do we require to address those 128 KB of memory? (1 pt.)

    128 KB memory device: 128KB = $2^{17}$ bytes. Thus, we require 17 bits to address the memory device.

  - ✓ We want to connect the 128 KB memory chip to the microprocessor. For optimal implementation, we must place those 128 KB in an address range where every address shares some MSBs. Provide a list of all the possible address ranges that the 128 KB memory chip can occupy. You can only use the non-occupied portions of the memory space as shown below.

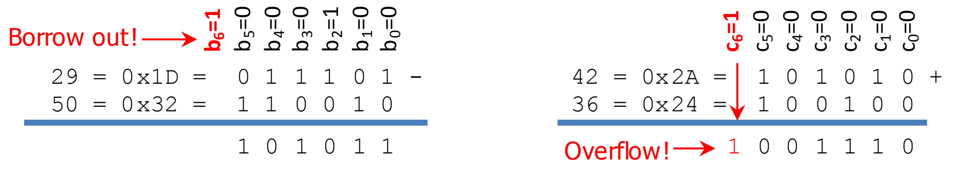
    ▫ 0x20000 to 0x3FFFF ▫ 0x40000 to 0x5FFFF ▫ 0x80000 to 0x9FFFF ▫ 0xA0000 to 0xBFFFF



## 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)

✓ 29 − 50 ✓ 42 + 36


Borrow out!
$b_6=1$ $b_5=0$ $b_4=0$ $b_3=0$ $b_2=1$ $b_1=0$ $b_0=0$
29 = 0x1D = 0 1 1 1 0 1 −
50 = 0x32 = 1 1 0 0 1 0
1 0 1 0 1 1
$c_6=1$ $c_5=0$ $c_4=0$ $c_3=0$ $c_2=0$ $c_1=0$ $c_0=0$
42 = 0x2A = 1 0 1 0 1 0 +
36 = 0x24 = 1 0 0 1 0 0
Overflow!
1 0 0 1 1 1 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)

✓ -79 + 62

**n = 8 bits** $c_8=0$ $c_7=0$ $c_6=1$ $c_5=1$ $c_4=0$ $c_3=0$ $c_2=0$ $c_1=0$ $c_0=0$

$c_8 \oplus c_7=0$
No Overflow

```
 62 = 0 0 1 1 1 1 1 0 +
-79 = 1 0 1 1 0 0 0 1
---------------------
-17 = 1 1 1 0 1 1 1 1
```

$-62 + 79 = -17 \in [-2^7, 2^7-1] \rightarrow$ no overflow

✓ -26 - 52

**n = 7 bits** $c_7=1$ $c_6=0$ $c_5=0$ $c_4=1$ $c_3=1$ $c_2=0$ $c_1=0$ $c_0=0$

$c_7 \oplus c_6=1$
Overflow!

```
-52 = 1 0 0 1 1 0 0 +
-26 = 1 1 0 0 1 1 0
-------------------
      0 1 1 0 0 1 0
```

$-52\ -26 = -78 \notin [-2^6, 2^6-1] \rightarrow$ overflow!

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

$c_8 \oplus c_7=0$
No Overflow

$c_8=1$ $c_7=1$ $c_6=0$ $c_5=0$ $c_4=1$ $c_3=1$ $c_2=0$ $c_1=0$ $c_0=0$

```
-52 = 1 1 0 0 1 1 0 0 +
-26 = 1 1 1 0 0 1 1 0
5
---------------------
      1 0 1 1 0 0 1 0
```

$-52\ -26 = -78 \in [-2^7, 2^7-1] \rightarrow$ no overflow

c) Perform binary multiplication of the following numbers that are represented in 2's complement arithmetic. (3 pts)

✓ 7 x -8

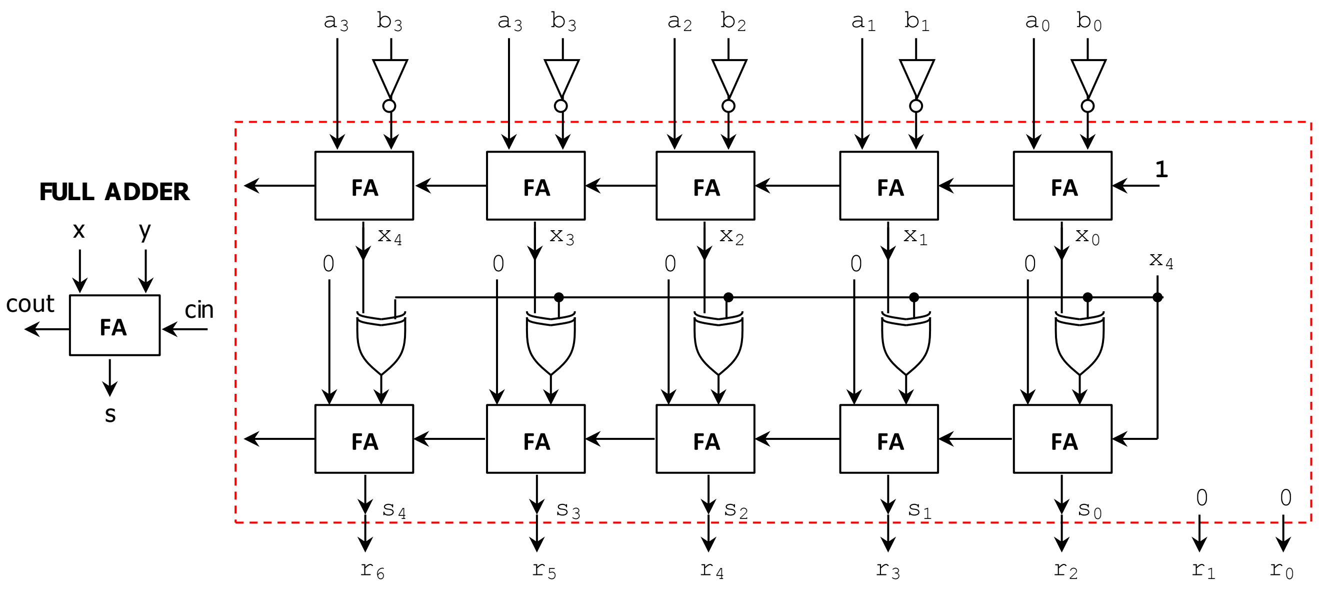
```
0 1 1 1 x          0 1 1 1 x
1 0 0 0            1 0 0 0
                   -------
                   0 0 0 0
                 0 0 0 0
               0 0 0 0
             0 1 1 1
             -------------
             0 1 1 1 0 0 0

             1 0 0 1 0 0 0
```

## PROBLEM 6 (11 PTS)

- Sketch the circuit that computes $|A-B| \times 4$, where $A, B$ are 4-bit signed (2's complement) numbers. For example: $A = 1001, B = 0111 \rightarrow |A-B| \times 4 = 14 \times 4 = 56$. You can only use full adders and logic gates. Your circuit must avoid overflow.

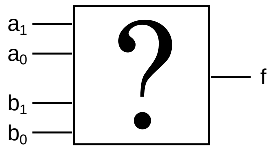
$A = a_3a_2a_1a_0$, $B = b_3b_2b_1b_0$
$A, B \in [-8,7] \rightarrow A, B$ require 4 bits in 2C representation.
✓ $X = A - B \in [-15,15]$ requires 5 bits in 2C. Thus, we need to sign-extend $A$ and $B$.
✓ $|X| = |A-B| \in [0,15]$ requires 5 bits in 2C. Thus, the second operation $0 \pm X$ only requires 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 $0 + X$.

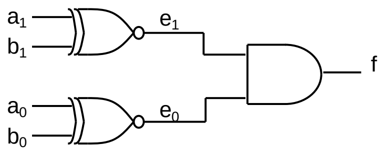
✓ $R = |A-B| \times 4 \in [0,60]$ requires 7 bits in 2C. Note that the MSB is always 0.



## PROBLEM 7 (16 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). (12 pts)

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

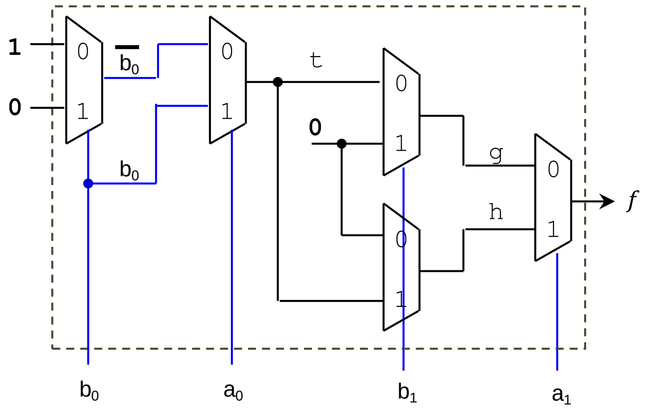
$f = \overline{a_1}f(0, b_1, a_0, b_0) + a_1 f(1, b_1, a_0, b_0) = \overline{a_1}\left(\overline{b_1}\left(\overline{a_0 \oplus b_0}\right)\right) + a_1\left(b_1\left(\overline{a_0 \oplus b_0}\right)\right) = \overline{a_1}g(b_1, a_0, b_0) + a_1 h(b_1, a_0, b_0)$

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

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

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

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


1
0
0
1
b̄0
b0
0
1
t
0
0
1
0
1
g
h
0
1
f
b0
a0
b1
a1