Solutions - Midterm Exam

(October 8th @ 9:30 am)

Clarity is very important! Show your procedure!

PROBLEM 1 (15 PTS)

- Complete the following table. Use the fewest number of bits on each case.

REPRESENTATION			
Decimal	**Sign-and-magnitude**	**1's complement**	**2's complement**
-63	1111111	1000000	1000001
-26	111010	100101	1100110
-111	11101111	10010000	10010001
45	0101101	0101101	0101101

PROBLEM 2 (15 PTS)

a) Convert the decimal number `136.6875` to i) binary, and ii) hexadecimal.
$136.6875 = (10001000.1011)_2 = 0x88.B$

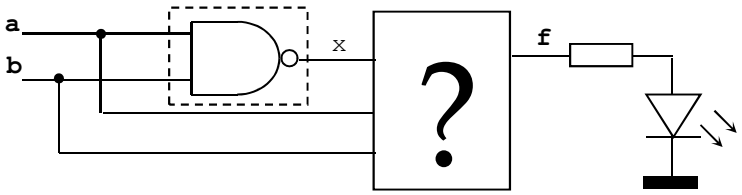
b) Represent the decimal number `136` in i) BCD, and ii) Reflective Gray Code

Decimal	Binary (unsigned)	BCD	Reflective Gray Code
136	10001000	0001 0011 0110	11001100

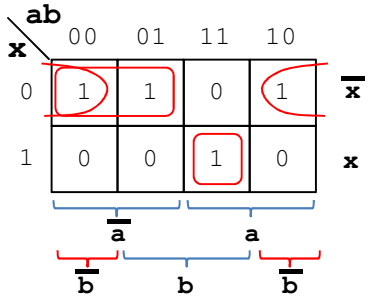
c) What is the minimum number of bits to represent numbers between 12,000 and 13,024?
Numbers between 12000 and 13024: $13024 - 12000 + 1 = 1025$
Minimum number of bits: $\lceil \log_2 1025 \rceil = 11\ bits$

PROBLEM 3 (15 PTS)

Design a circuit (***simplify your circuit!***) that verifies the logical operation of a NAND gate. `f = '1'` (LED ON) if the NAND gate does NOT work properly. Assumption: when the NAND gate is not working, it generates `1`'s instead of `0`'s and vice versa.



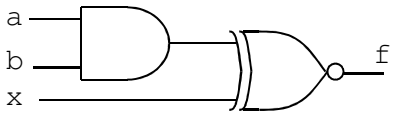
a	b	x	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



$f = \overline{x}\,\overline{b} + \overline{x}\,\overline{a} + xab$

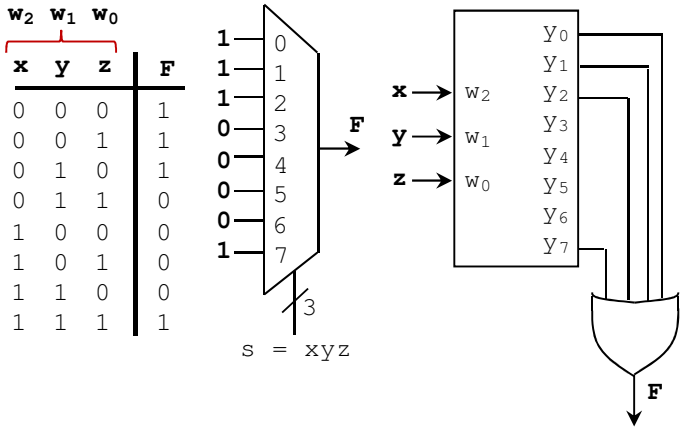
$f = \overline{x}\overline{(ab)} + xab$

$f = \overline{x \oplus (ab)}$

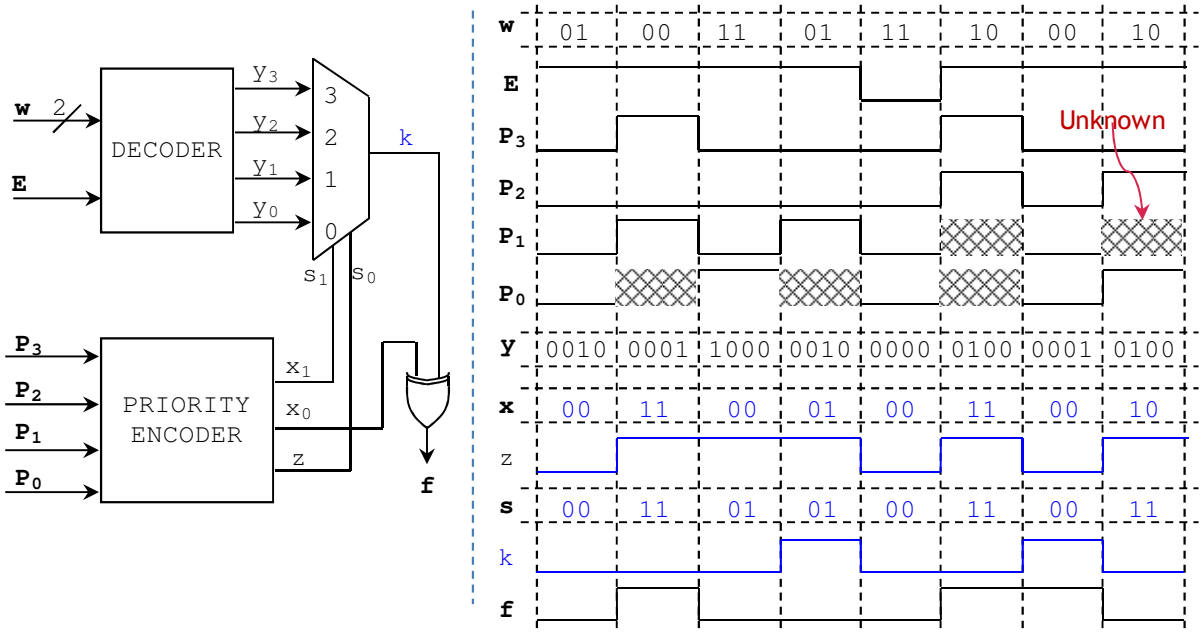


PROBLEM 4 (20 PTS)

- Implement the following function using: i) a decoder (and gates) and ii) a multiplexer:
  - ✓ $F(X,Y,Z) = \prod(M_3, M_4, M_5, M_6)$



- Complete the timing diagram of the circuit shown below:



PROBLEM 5 (15 PTS)

- Perform the following operations using the 2's complement arithmetic. For each case, provide the summands and the result in 2's complement representation. Use the minimum number of bits to represent the summands and the result so that overflow is avoided.

✓ −26 + 45

n = 7 bits

c_7=1 c_6=1

```
−26 = 1100110 +
+45 = 0101101
------------
+19 = 0010011
```

overflow = $c_7 \oplus c_6$=0 -> no overflow

+19 ∈ $[-2^6, 2^6-1]$ -> no overflow

✓ −63 − 15

n = 7 bits

c_7=1 c_6=0

```
−63 = 1000001 +
−15 = 1110001
------------
      0110010
```

overflow = $c_7 \oplus c_6$=1 -> overflow!

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

c_8=1 c_7=1

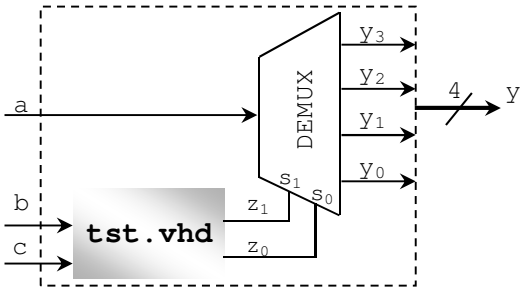
```
−63 = 11000001 +
−15 = 11110001
------------
−78 = 10110010
```

overflow = $c_8 \oplus c_7$=0 -> no overflow

−78 ∈ $[-2^7, 2^7-1]$ -> no overflow

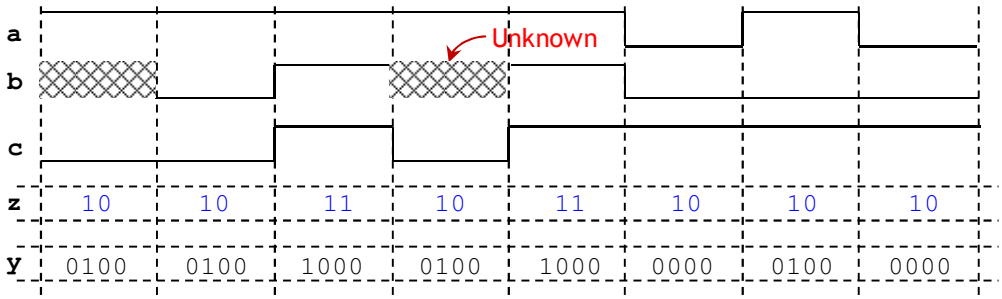
PROBLEM 6 (10 PTS)

- The following VHDL code corresponds to the shaded circuit. Complete the timing diagram:



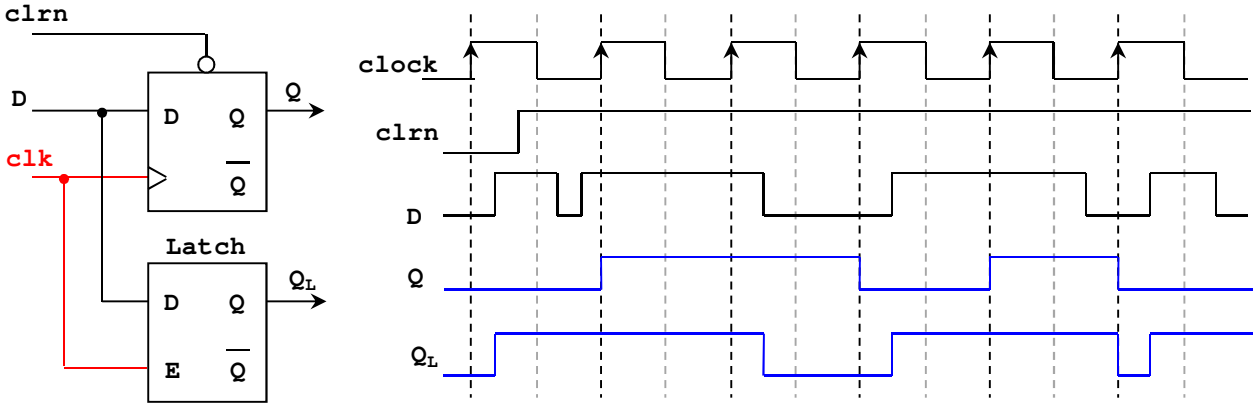

```
library ieee;
use ieee.std_logic_1164.all;

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

```
architecture bhv of tst is
  signal x, y: std_logic;

begin

  process (b,c)
  begin
     z <= c & b;
     if c = '0' then
        z <= "10";
     end if;
  end process;

end bhv;
```



PROBLEM 7 (10 PTS)

- Complete the timing diagram of the circuit shown below:



BONUS PROBLEM (+5 PTS)

- A microprocessor has a 24-bit address line. We connect a memory chip to the microprocessor. The memory chip addresses are assigned the range `0x800000` to `0xBFFFFF`. What is the minimum number of bits required to represent addresses in that individual memory chip? Explain your procedure.

- If we convert the numbers to binary, we notice that the addresses in the range `0x800000` to `0xBFFFFF` require 24 bits.

```
0x800000: 1000 0000 0000 0000 0000 0000
                                    ...
                                    ...
                                    ...
0xBFFFFF: 1011 1111 1111 1111 1111 1111
```

- However, we notice that all the addresses in the range `0x800000` to `0xBFFFFF` share the same first two MSBs: `10`. Therefore, if we were to address data ONLY in the individual memory chip, we do not need those two bits ⇒ we need ***22 bits***.
- Another way to put it: We only need ***22 bits*** for the address line of that individual memory device.