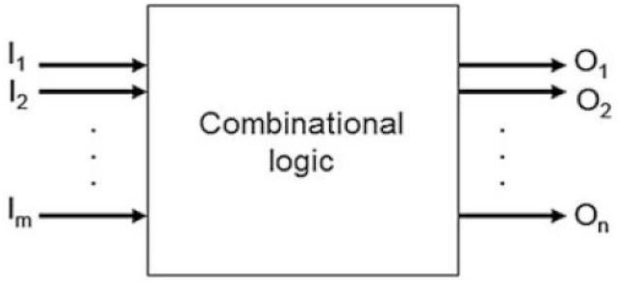
CEG3155: DGD 1

Review

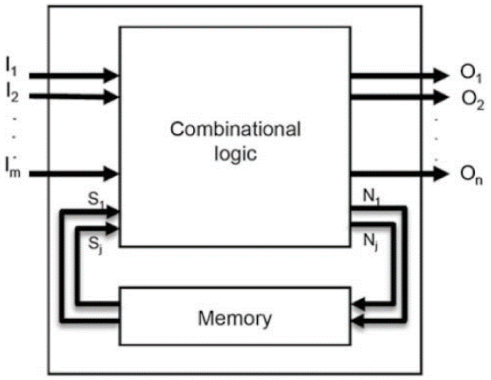
Combinational Logic

- Outputs depend only on the present inputs, i.e. no memory elements.
- Examples:
  - Multiplexers, demultiplexers, encoders, and decoders.
  - Adders, subtractors, multipliers, shifters, comparators, and ALUs.



Sequential Logic

- Outputs depend not only on the present inputs but also on past inputs.
- A memory element is used to store the state of the previous inputs.
- Sequential circuits are usually designed using a Finite State Machine (FSM).
- Examples:
  - Flip-flops, and counters.



IEEE 1164 Standard

- A technical standard published by the IEEE in 1993. (https://perso.telecom-paristech.fr/guilley/ENS/20171205/TP/tp_syn/doc/IEEE_VHDL_1164-1993.pdf)
- It describes the logic values to be used in Electronic Design Automation (EDA) tools.
- Defines the following primary data types:

Character	Value
'U'	Uninitialized
'X'	Strong unknown value
'0'	Strong zero
'1'	Strong one
'Z'	High impedance
'W'	Weak unknown
'L'	Weak zero
'H'	Weak one
'-'	Don't care

- In VHDL, this standard is embodied in the std_logic_1164 package.

Questions

Question 1

Consider the function $f(x1, x2, x3) = m(2, 3, 4, 6, 7)$, use the truth table below to identify the Sum-of-Products (SoP) and Product-of-Sums (PoS).

	$x1$ $x2$ $x3$	Minterms	Maxterms	$f(x1, x2, x3)$
0	0 0 0	m0 = $\overline{x1}\ \overline{x2}\ \overline{x3}$	M0 = $x1 + x2 + x3$	0
1	0 0 1	m1 = $\overline{x1}\ \overline{x2}\ x3$	M1 = $x1 + x2 + \overline{x3}$	0
2	0 1 0	m2 = $\overline{x1}\ x2\ \overline{x3}$	M2 = $x1 + \overline{x2} + x3$	1
3	0 1 1	m3 = $\overline{x1}\ x2\ x3$	M3 = $x1 + \overline{x2} + \overline{x3}$	1
4	1 0 0	m4 = $x1\ \overline{x2}\ \overline{x3}$	M4 = $\overline{x1} + x2 + x3$	1
5	1 0 1	m5 = $x1\ \overline{x2}\ x3$	M5 = $\overline{x1} + x2 + \overline{x3}$	0
6	1 1 0	m6 = $x1\ x2\ \overline{x3}$	M6 = $\overline{x1} + \overline{x2} + x3$	1
7	1 1 1	m7 = $x1\ x2\ x3$	M7 = $\overline{x1} + \overline{x2} + \overline{x3}$	1

Identify the SoP, and PoS terms, and prove that $f(x1, x2, x3)$ can be expressed by SoP or PoS.

Solution

Sum-of-products:

$$f(x1, x2, x3) = m(2, 3, 4, 6, 7) = m2 + m3 + m4 + m6 + m7$$

$$f(x1, x2, x3) = \overline{x1}\ x2\ \overline{x3} + \overline{x1}\ x2\ x3 + x1\ \overline{x2}\ \overline{x3} + x1\ x2\ \overline{x3} + x1\ x2\ x3$$

Product-of-sums:

$$f(x1, x2, x3) = \prod(M2, M3, M4, M6, M7) = M2\ M3\ M4\ M6\ M7$$

$$f(x1, x2, x3) = (x1 + \overline{x2} + x3)(x1 + \overline{x2} + \overline{x3})(\overline{x1} + x2 + x3)(\overline{x1} + \overline{x2} + x3)(\overline{x1} + \overline{x2} + \overline{x3})$$

Question 2

Use algebraic manipulation to show that for three input variables $x1, x2, x3$:

$$\sum m(1, 2, 3, 4, 5, 6, 7) = x1 + x2 + x3$$

Solution

$$\sum m(1, 2, 3, 4, 5, 6, 7) = m1 + m2 + m3 + m4 + m5 + m6 + m7$$

$$\sum m(1, 2, 3, 4, 5, 6, 7) = m1 + m2 + m3 + m4 + m5 + m6 + m7$$

$$\sum m(1, 2, 3, 4, 5, 6, 7) = \overline{x1}\ \overline{x2}\ x3 + \overline{x1}\ x2\ \overline{x3} + \overline{x1}\ x2\ x3 + x1\ \overline{x2}\ \overline{x3} + x1\ \overline{x2}\ x3 + x1\ x2\ \overline{x3} + x1\ x2\ x3$$

$$\sum m(1, 2, 3, 4, 5, 6, 7) = \overline{x1}\ (\overline{x2}\ x3 + x2\ \overline{x3} + x2\ x3) + x1(\overline{x2}\ \overline{x3} + \overline{x2}\ x3 + x2\ \overline{x3} + x2\ x3)$$

$$\sum m(1, 2, 3, 4, 5, 6, 7) = \overline{x1}\ (\overline{x2}\ x3 + x2) + x1$$

$$\sum m(1,2,3,4,5,6,7) = x1 + \overline{x1}\,(\overline{x2}\,x3 + x2 + x2x3)$$

$$\sum m(1,2,3,4,5,6,7) = x1 + \overline{x1}\,(x2 + x3)$$

$$\sum m(1,2,3,4,5,6,7) = x1 + \overline{x1}\,x2 + \overline{x1}\,x3$$

Using Absorption rule (A + AB = A), we write x1 = x1 + x1x2

$$\sum m(1,2,3,4,5,6,7) = x1 + x1\,x2 + \overline{x1}\,x2 + \overline{x1}\,x3$$

$$\sum m(1,2,3,4,5,6,7) = x1 + x2 + \overline{x1}\,x3$$

$$\sum m(1,2,3,4,5,6,7) = x1 + x1\,x3 + x2 + \overline{x1}\,x3$$

$$\sum m(1,2,3,4,5,6,7) = x1 + x2 + x3$$

Question 3

Perform the following operations in binary, where numbers are signed and represented in 2's complement form. Then, get the corresponding decimal number.

a) 00000110 + 00001101
b) 11111010 – 11110011
c) 1100 + 1010

Solution

a)

$$\begin{array}{r} 00000110 \\ +00001101 \\ \hline 00010011 \end{array}$$

b)

$$\begin{array}{r} 11111010 \\ -11110011 \\ \hline 11111010 \\ +00001101 \\ \hline 1\,00000111 \end{array}$$

$$Discard\ the\ left\ most\ bit, no\ overlow$$

c)

$$\begin{array}{r} 1100 \\ +1010 \\ \hline 1\ 0010 \end{array}$$

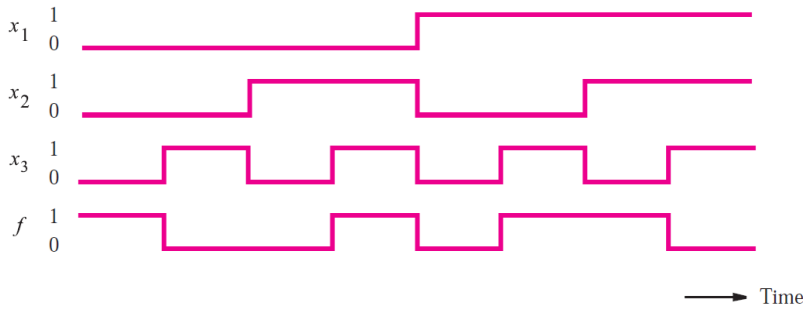
$$Overflow$$

In general, overflow happens if the size of the data type cannot accommodate the result with the following conditions:

- *If both operands are positive and the result is negative.*
- *If both operands are negative and the result is positive.*

Question 4

For the timing diagram in the Figure, synthesize the function $f(x1, x2, x3)$ in the simplest sum-of-products form.



Solution

x1	*x2*	*x3*	*f*
0	*0*	*0*	*1*
0	*0*	*1*	*0*
0	*1*	*0*	*0*
0	*1*	*1*	*1*
1	*0*	*0*	*0*
1	*0*	*1*	*1*
1	*1*	*0*	*1*
1	*1*	*1*	*0*

Using minterms, we get:

$$f(x1, x2, x3) = m(0, 3, 5, 6) = \overline{x1}\,\overline{x2}\,\overline{x3} + \overline{x1}\,x2\,x3 + x1\,\overline{x2}\,x3 + x1\,x2\,\overline{x3}$$

Question 5

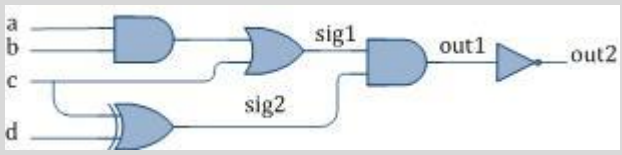
Draw the circuit described by the following VHDL code.

```
architecture SomeCode of SomeEntity is
    signal sig1, sig2, out1: std_logic;
begin

    out2 <= not out1;
    sig2 <= (c xor d);
    out1 <= (sig1 and sig2);
    sig1 <= (a and b) or c;

end SomeCode;
```

Solution



Question 6

Write VHDL code to implement the function $f(x1, x2, x3) = m(0, 1, 3, 4, 5, 6)$.

Solution

Perform simplification using Karnaugh map:

$$f(x1, x2, x3) = m(0, 1, 3, 4, 5, 6) = \overline{x1} + \overline{x2}$$

question6.vhd

```
1   library ieee;
2   use ieee_std_1164.all;
3
4   entity question6 is
5       port(
6           x1, x2, x3: in std_logic;
7           f: out std_logic
8       );
9   end question6;
10
11  architecture q6 of question6 is
12  begin
13      f <= (not x1) or (not x2);
14  end q6;
```

Question 7

Design 1-bit full adder in VHDL:

a) Show timing diagrams of the sum and carry-out bits for all possible values of inputs.
b) Show structural implementation of it.

Solution

a) Full-adder

$$S = \bar{A}\bar{B}C_{IN} + \bar{A}B\bar{C}_{IN} + A\bar{B}\bar{C}_{IN} + ABC_{IN}$$

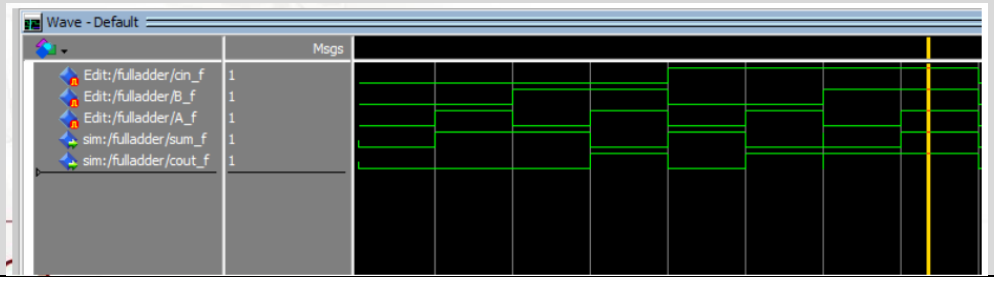
$$C_{OUT} = BC_{IN} + AC_{IN} + AB$$

$$S = A \oplus [B \oplus C_{IN}]$$

Input bit for number A	Input bit for number B	Carry bit input C_{IN}	Sum bit output S	Carry bit output C_{OUT}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

fulladder.vhd

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity fulladder is
5       port(
6           A_f, B_f, cin_f : in std_logic;
7           sum_f, cout_f   : out std_logic
8       );
9   end fulladder;
10
11  architecture fulladder_behav of fulladder is
12  begin
13      sum_f <= (A_f xor B_f) xor cin_f;
14      cout_f <= (A_f and B_f) or (A_f and cin_f) or (B_f and cin_f);
15
16  end fulladder_behav;
17
```



b) Structural design of a full-adder using two half-adders:

halfadder.vhd

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity halfadder is
5      port(
6          A, B        : in std_logic;
7          sum, cout   : out std_logic
8      );
9  end halfadder;
10
11 architecture halfadder_behav of halfadder is
12 begin
13     sum <= A xor B;
14     cout <= A and B;
15 end halfadder_behav;
16
```

fulladder.vhd

```
1  library ieee;
2  use ieee_std_1164.all;
3
4  entity fulladder is
5      port(
6          A_f, B_f, cin_f: in std_logic;
7          sum_f, cout_f: out std_logic
8      );
9  end fulladder;
10
11 architecture fulladder_struct of fulladder is
12     signal sum, cout1, cout2: std_logic;
13
14     component halfadder is
15     port(
16         A, B: in std_logic;
17         sum, cout: out std_logic
18     );
19     end component;
20 begin
21     H1: halfadder port map(A->A_f, B->B_f, sum, cout1);
22     H2: halfadder port map(A->sum, B->cout1, sum_f, cout2);
23
24     cout_f <= cout1 or cout2;
25 end fulladder_struct;
```