Digital System Assignment 1

- Full Adder & 4 Bit Adder Design-

Embedded Computer System Lab

Full Adder

- Logic function – Sum
  - Truth Table

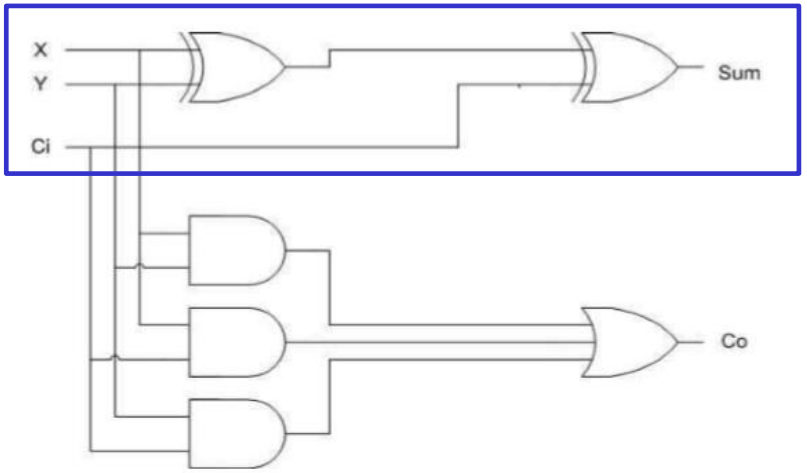
입 력			출 력
X	Y	Cin	Sum
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$\text{Sum} = \bar{X}\bar{Y}C_{in} + \bar{X}Y\overline{C_{in}} + X\bar{Y}\,\overline{C_{in}} + XYC_{in} \qquad C_{out} = XY + XC_{in} + YC_{in}$$

$$= \overline{C_{in}}(\bar{X}Y + X\bar{Y}) + C_{in}(\bar{X}\,\bar{Y} + XY)$$

$$= \overline{C_{in}}(\bar{X}Y + X\bar{Y}) + C_{in}\overline{(\bar{X}Y + X\bar{Y})}$$

$$\text{Sum} = C_{in} \oplus (X \oplus Y)$$



Full Adder

- Logic function – Carry Out
  - Truth Table

입 력			출 력
X	Y	Cin	Cout
0	0	0	0
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	1

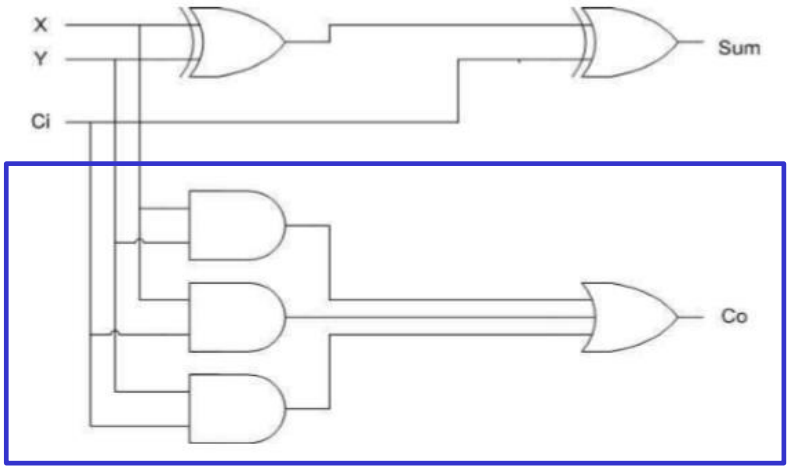
$$\text{Sum} = \bar{X}\bar{Y}C_{in} + \bar{X}Y\overline{C_{in}} + X\bar{Y}\,\overline{C_{in}} + XYC_{in}$$

$$= \overline{C_{in}}(\bar{X}Y + X\bar{Y}) + C_{in}(\bar{X}\bar{Y} + XY)$$

$$= \overline{C_{in}}(\bar{X}Y + X\bar{Y}) + C_{in}\overline{(\bar{X}Y + X\bar{Y})}$$

$$\text{Sum} = C_{in} \oplus (X \oplus Y)$$

$$C_{out} = XY + XC_{in} + YC_{in}$$



Design Assignment 1

- Full Adder Design Source Code (Design Source)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Full_Adder is
    port (x, y, cin : in std_logic;
    sum : out std_logic;
    cout : out std_logic);
end Full_Adder;

architecture Behavioral of Full_Adder is
Begin

        Complete the code here

end Behavioral;
```

Design Assignment 1

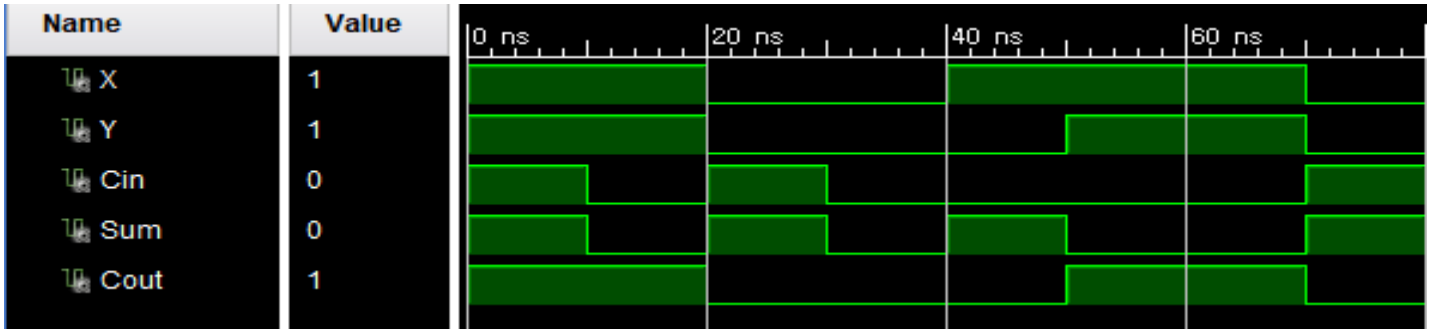
- Full Adder Testbench Code (Simulation Source)

```
entity FA_TB is
end FA_TB;

architecture Behavioral of FA_TB is
    component Full_Adder port (
        x, y, cin : in std_logic;
        sum : out std_logic;
        cout : out std_logic);
    end component;

    signal X : std_logic := '0';
    signal Y : std_logic := '1';
    signal Cin : std_logic := '1';
    signal Sum, Cout : std_logic;

begin
    uut : Full_Adder port map (
        x=>X,
        y=>Y,
        cin=>Cin,
        sum=>Sum,
        cout=>Cout);
```

```
stim_proc : process
        begin
        X <= '1';
        Y <= '1';
        wait for 10ns;

        Cin <= '0';
        wait for 10ns;

        X <= '0';
        Y <= '0';
        Cin <= '1';
        wait for 10ns;

        Cin <= '0';
        wait for 10ns;

        X <= '1';
        Y <= '0';
        wait for 10ns;
    end process;

end Behavioral;
```

Design Assignment 1

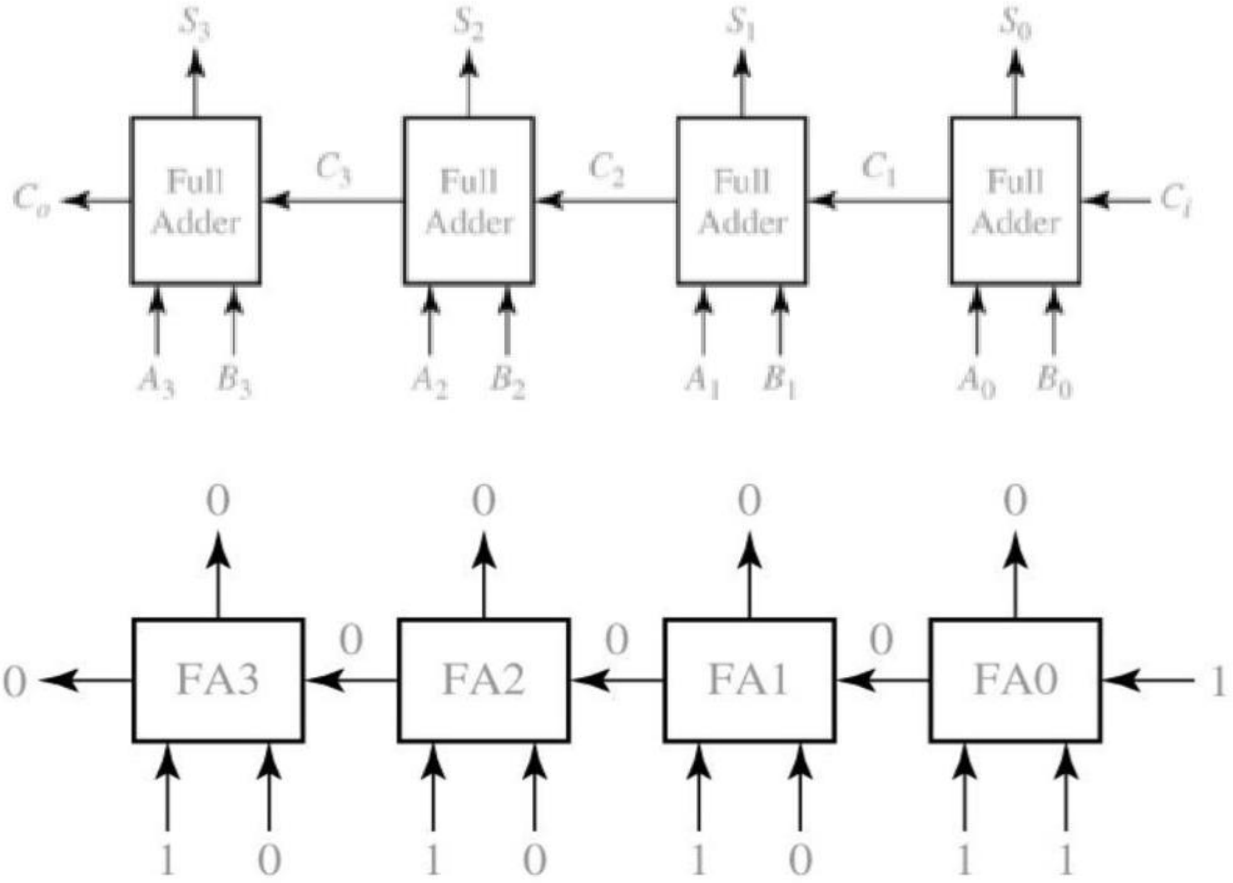
- Result



- X : 1, Y : 1, Cin : 1 -> Sum : 1, Cout : 1
- X : 1, Y : 1, Cin : 0 -> Sum : 0, Cout : 1
- X : 0, Y : 0, Cin : 1 -> Sum : 1, Cout : 0

4 Bit Adder

- Block Diagram
  - 4개의 Full Adder가 직렬로 연결.



Design Assignment 2

- 4 Bit Adder Design Source Code (Design Source)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Adder_4bit is
    port (x, y : in std_logic_vector(3 downto 0);
    cin : in std_logic;
    sum : out std_logic_vector(3 downto 0);
    cout : out std_logic);
end Adder_4bit;

architecture Behavioral of Adder_4bit is
        Complete the code here
Begin
        Complete the code here
end Behavioral;
```

Design Assignment 2

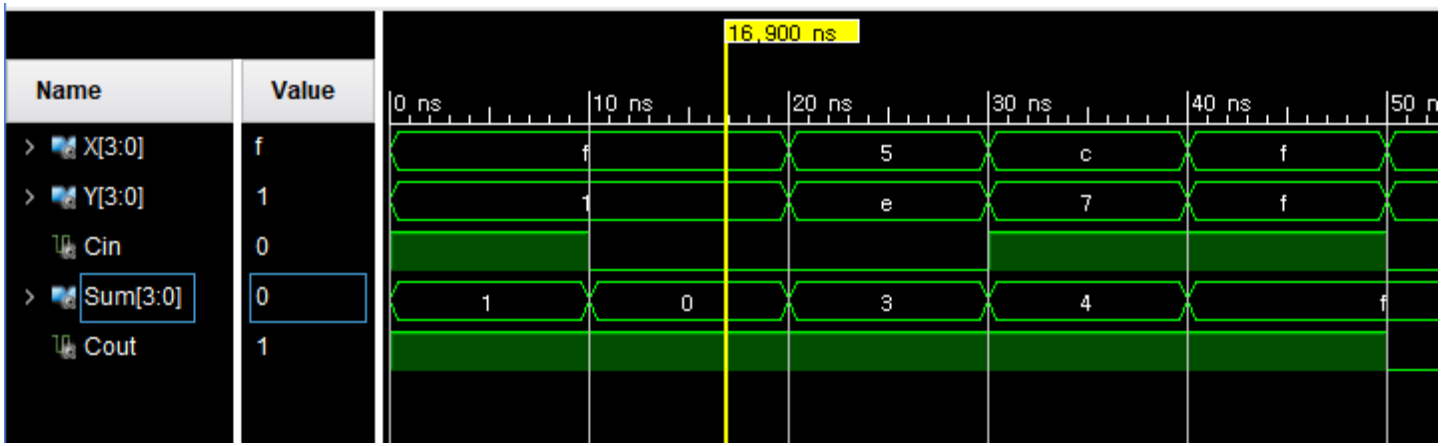
- 4 Bit Adder Testbench Code (Simulation Source)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Adder_4bit_TB is
end Adder_4bit_TB;

architecture Behavioral of Adder_4bit_TB is
    component Adder_4bit
        port (x, y : in std_logic_vector(3 downto 0);
        cin : in std_logic;
        sum : out std_logic_vector(3 downto 0);
        cout : out std_logic);
    end component;

    signal X : std_logic_vector(3 downto 0);
    signal Y : std_logic_vector(3 downto 0);
    signal Cin : std_logic;
    signal Sum : std_logic_vector(3 downto 0);
    signal Cout : std_logic;

begin
    uut : Adder_4bit port map (
        x=>X,
        y=>Y,
        cin=>Cin,
        sum=>Sum,
        cout=>Cout);
```

```
  stim_proc : process
        begin
        X <= "1111";
        Y <= "0001";
        Cin <= '1';
        wait for 10ns;

        Cin <= '0';
        wait for 10ns;

        X <= "0101";
        Y <= "1110";
        wait for 10ns;

        X <= "1100";
        Y <= "0111";
        Cin <= '1';
        wait for 10ns;

        X <= "1111";
        Y <= "1111";
        wait for 10ns;

        X <= "0110";
        Y <= "1001";
        Cin <= '0';
        wait for 10ns;
    end process;

end Behavioral;
```

Design Assignment 2

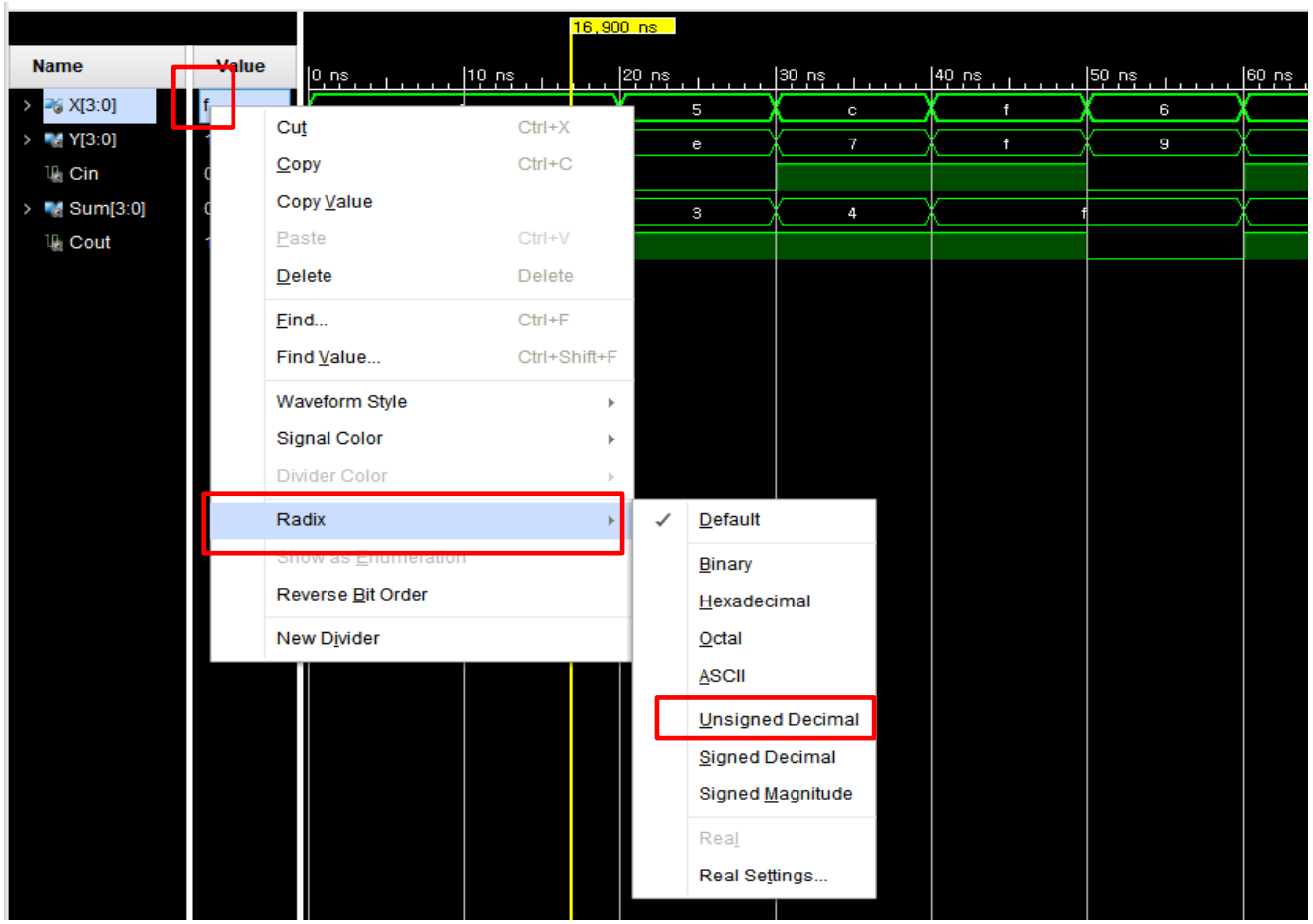
- Result



- X : f(1111) + Y : 1(0001) + Cin (1) = Cout(1) / Sum(0001)
- X : f(0101) + Y : e(1110) + Cin (0) = Cout(1) / Sum(0011)
- 16진수로 되어 있어 결과 값을 한 눈에 이해하기 어려울 수 있음

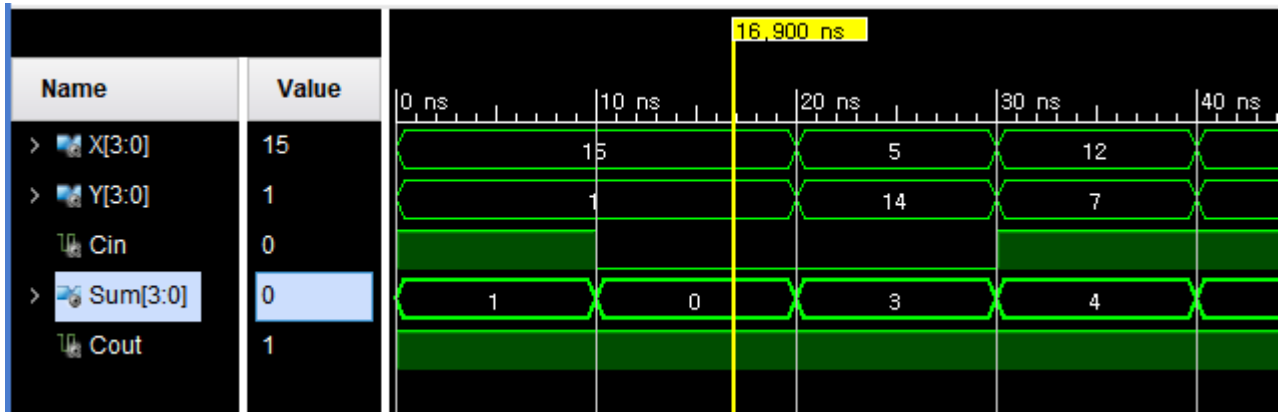
Design Assignment 2

- Radix : unsigned decimal



Design Assignment 2

- Result



- X : 15(1111) + Y : 1(0001) + Cin (1) = Cout(1) / Sum : 1(0001)
- X : 5(0101) + Y : 14(1110) + Cin (0) = Cout(1) / Sum : 3(0011)

Submission

- 제출
  - i-campus에 파일 업로드 (보고서 + 소스코드를 하나로 압축해 제출)
  - 압축된 파일명 : 학번_이름_HW1.xxx
  - Late Penalty
    - 1s ~ 1 day : -10%
    - 1day ~ 2day : -30%
    - 2day ~ 3day : -50%
    - 3day ~ : Cannot submit
  - 문의사항 : 조교 고준성(myun243@skku.edu)

Submission

- 제출 유의사항 (미흡시 불이익)
  - 워드 프로그램에 설계한 VHDL 코드와 waveform을 첨부한 보고서
    - 설계한 VHDL 코드와 설명
      - 설계의 핵심이 되는 내용은 반드시 보고서에 설명
      - 설계한 방법과 이를 바탕으로 한 Adder 동작 원리 상세히 설명
      - TestBench 결과에만 맞게 설계 방향을 틀어서 진행한 경우, 대량 감점
        TestBench 이외의 추가 경우의 수로 시뮬레이션 진행
    - 주어진 TB를 이용한 시뮬레이션 결과 waveform과 설명 및 분석
    - 보고서 파일명 : 학번_이름_HW1.xxx
  - 직접 작성한 VHDL 소스코드 파일 제출
    - Source code 파일명 : 학번_이름_FA.vhd / 학번_이름_4_Bit_Adder.vhd
    - Testbench code 파일명 : 학번_이름_FA_tb.vhd / 학번_이름_4_Bit_Adder_tb.vhd
    - 소스코드의 핵심 내용은 반드시 주석으로 설명할 것
  - 보고서와 소스코드 파일을 하나로 압축해 제출