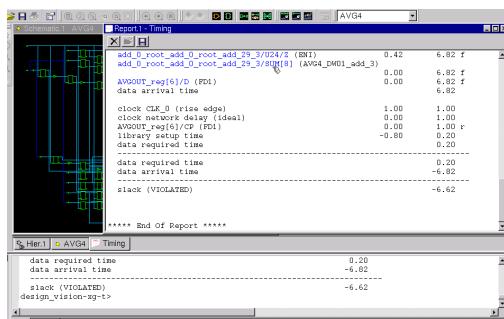


氏名：津波古正輝
学籍番号：e075739A
提出日：12月23日(火曜日)

HW5

1) クロックの周期のターゲット値を1として、再合成する。(なるべく、最小サイクル時間の小さい回路を合成せよ)



図：period を 1

slack が-6.62 と表示され、VIOLATION となつた。

2) この時の最小の動作サイクル時間はいくらか？

$$1 - (-6.62) = 7.62 \text{ より、} \\ 7.62.$$

3) クロック = 50 ns の場合と、面積および最小サイクルタイムで比較を行え！
面積はクロック=7.62の方が大きくなつた。
速度はクロック=7.62の方が圧倒的に小さい。

```
class (File: /usr/local/synopsys/z-2007.03/libraries/s
Number of ports: 17
Number of nets: 85
Number of cells: 46
Number of references: 5

Combinational area: 336.000000
Noncombinational area: 280.000000
Net Interconnect area: undefined (Wire load has zero
Total cell area: 616.000000
Total area: undefined
```

図：最小クロックの面積

```
class (File: /usr/local/synopsys/z-2007.03/libraries/syn/cle
Number of ports: 17
Number of nets: 79
Number of cells: 43
Number of references: 4

Combinational area: 283.000000
Noncombinational area: 280.000000
Net Interconnect area: undefined (Wire load has zero net a
Total cell area: 563.000000
Total area: undefined

***** End Of Report *****
```

図：50ns クロックの面積

4) 回路を改造して、16ポイントの平均を計算する回路を設計せよ。

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;

entity AVG4 is
    port(CLK      : in  std_logic;
          FMINPUT :
          in  std_logic_vector(7 downto 0);
          AVGOUT   :
          out std_logic_vector(7 downto 0));
end AVG4;

architecture RTL of AVG4 is
signal FF1, FF2, FF3, FF4, FF5,
       FF6, FF7, FF8, FF9, FF10,
       FF11, FF12, FF13, FF14,
       FF15, FF16
       : std_logic_vector(7 downto 0);
signal SUM:
std_logic_vector(11 downto 0);

begin
-- SHIFT REGISTER
process(CLK) begin
    if (CLK'event and CLK = '1') then
        FF1 <= FMINPUT;
        FF2 <= FF1;
        FF3 <= FF2;
        FF4 <= FF3;
        FF5 <= FF4;
        FF6 <= FF5;
        FF7 <= FF6;
        FF8 <= FF7;
        FF9 <= FF8;
        FF10 <= FF9;
end if;
end process;

-- SUM
Sum <= signed(FF1(7)&FF1(7)&FF1(7)&FF1(7)&FF1(7)&FF1(7)&FF1(7)&FF1(7)&FF1(7)&FF1(7)&FF1(7)&FF1(7)&FF1(7)&FF1(7)&FF1(7)&FF1(7));
+signed(FF2(7)&FF2(7)&FF2(7)&FF2(7)&FF2(7)&FF2(7)&FF2(7)&FF2(7)&FF2(7)&FF2(7)&FF2(7)&FF2(7)&FF2(7)&FF2(7)&FF2(7)&FF2(7)&FF2(7));
+signed(FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7)&FF3(7));
+signed(FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7)&FF4(7));
+signed(FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7)&FF5(7));
+signed(FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7)&FF6(7));
+signed(FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7)&FF7(7));
+signed(FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7)&FF8(7));
+signed(FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7)&FF9(7));
+signed(FF10(7)&FF10(7)&FF10(7)&FF10(7)&FF10(7)&FF10(7)&FF10(7)&FF10(7)&FF10(7)&FF10(7)&FF10(7)&FF10(7)&FF10(7)&FF10(7)&FF10(7)&FF10(7)&FF10(7));
+signed(FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7)&FF11(7));
+signed(FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7)&FF12(7));
+signed(FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7)&FF13(7));
+signed(FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7)&FF14(7));
+signed(FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7)&FF15(7));
+signed(FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7)&FF16(7));
-- DIVIDE BY 16 (SHIFT 4 bit), OUTPUT REGISTER
process(CLK) begin
    if (CLK'event and CLK='1') then
        AVGOUT <= SUM(11 downto 4);
    end if;
end process;
end RTL;

```