CAD

-report5-

055702B　　池野谷克俊

2006 年 12 月 19 日 火曜日

1　課題５ TWIDDLE 生成回路

1.1　問題

以下の ENTITY 仕様で 64FFT 対応の TWIDDLE ファクター生成の組み合わせ回路を作成せよ。(FF は使用禁止!)

W_I= cos(2*π* ADDR /64)

W_Q = -sin (2*π* ADDR/64)

2　解答

以下に作成したプログラム、twiddle.vhd を示す。
(テストベンチは、変更していないので省略する。)

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;

entity TWIDDLE is
  port (
    ADDR : in  std_logic_vector(5 downto 0); -- <6,6,u>
    W_I  : out std_logic_vector(9 downto 0); -- <10,0,t>
    W_Q  : out std_logic_vector(9 downto 0)); -- <10,0,t>
end;

architecture RTL of TWIDDLE is
signal OUTDATA : std_logic_vector(19 downto 0);
begin

W_I <= OUTDATA(19 downto 10);
W_Q <= OUTDATA(9 downto 0);

process(ADDR)begin
  case ADDR is
    when "000000" =>OUTDATA<="01111111110000000000";
    when "000001" =>OUTDATA<="01111111101111001110";
    when "000010" =>OUTDATA<="01111101101110011100";
    when "000011" =>OUTDATA<="01111010101101101011";
    when "000100" =>OUTDATA<="01110110011100111100";
    when "000101" =>OUTDATA<="01110001001100001111";
    when "000110" =>OUTDATA<="01101010101011100100";
    when "000111" =>OUTDATA<="01100110010101111011";
    when "001000" =>OUTDATA<="01011010101010010110";
    when "001001" =>OUTDATA<="01010001011001110100";
    when "001010" =>OUTDATA<="01000111001001010110";
    when "001011" =>OUTDATA<="00111100011000111100";
    when "001100" =>OUTDATA<="00110001001000100111";
    when "001101" =>OUTDATA<="00100101011000010110";
    when "001110" =>OUTDATA<="00011001001000001010";
    when "001111" =>OUTDATA<="00001100101000000010";
    when "010000" =>OUTDATA<="00000000001000000000";
    when "010001" =>OUTDATA<="11110011101000000010";
    when "010010" =>OUTDATA<="11100111001000001010";
    when "010011" =>OUTDATA<="11011010111000010110";
    when "010100" =>OUTDATA<="11001111001000100111";
    when "010101" =>OUTDATA<="11000011111000111100";
    when "010110" =>OUTDATA<="10111001001001010110";
    when "010111" =>OUTDATA<="10101110111001110100";
    when "011000" =>OUTDATA<="10100101101010010110";
    when "011001" =>OUTDATA<="10011101001010111011";
    when "011010" =>OUTDATA<="10010101101011100100";
    when "011011" =>OUTDATA<="10001111001100001111";
```

```
    when "011100" =>OUTDATA<="10001001111100111100";
    when "011101" =>OUTDATA<="10000101101101101011";
    when "011110" =>OUTDATA<="10000010101110011100";
    when "011111" =>OUTDATA<="10000000101111001110";
    when "100000" =>OUTDATA<="10000000000000000000";
    when "100001" =>OUTDATA<="10000000100000110010";
    when "100010" =>OUTDATA<="10000010100001100100";
    when "100011" =>OUTDATA<="10000101100010010101";
    when "100100" =>OUTDATA<="10001001110011000100";
    when "100101" =>OUTDATA<="10001111000011110001";
    when "100110" =>OUTDATA<="10010101100100011100";
    when "100111" =>OUTDATA<="10011101000101000101";
    when "101000" =>OUTDATA<="10100101100101101010";
    when "101001" =>OUTDATA<="10101110110110001100";
    when "101010" =>OUTDATA<="10111001000110101010";
    when "101011" =>OUTDATA<="11000011110111000100";
    when "101100" =>OUTDATA<="11001111000111011001";
    when "101101" =>OUTDATA<="11011010110111101010";
    when "101110" =>OUTDATA<="11100111000111110110";
    when "101111" =>OUTDATA<="11110011100111111110";
    when "110000" =>OUTDATA<="00000000000111111111";
    when "110001" =>OUTDATA<="00001100100111111110";
    when "110010" =>OUTDATA<="00011001000111110110";
    when "110011" =>OUTDATA<="00100101010111101010";
    when "110100" =>OUTDATA<="00110001000111011001";
    when "110101" =>OUTDATA<="00111100010111000100";
    when "110110" =>OUTDATA<="01000111000110101010";
    when "110111" =>OUTDATA<="01010001010110001100";
    when "111000" =>OUTDATA<="01011010100101101010";
    when "111001" =>OUTDATA<="01100011000101000101";
    when "111010" =>OUTDATA<="01101010100100011100";
    when "111011" =>OUTDATA<="01110001000011110001";
    when "111100" =>OUTDATA<="01111011001011000100";
    when "111101" =>OUTDATA<="01111010100010010101";
    when "111110" =>OUTDATA<="01111101100001100100";
    when "111111" =>OUTDATA<="01111111100000110010";
    when others =>OUTDATA<="XXXXXXXXXXXXXXXXXXXX";
  end case;

end process;
end RTL;
```

scirocco を用いて動作波形を出力してみた。

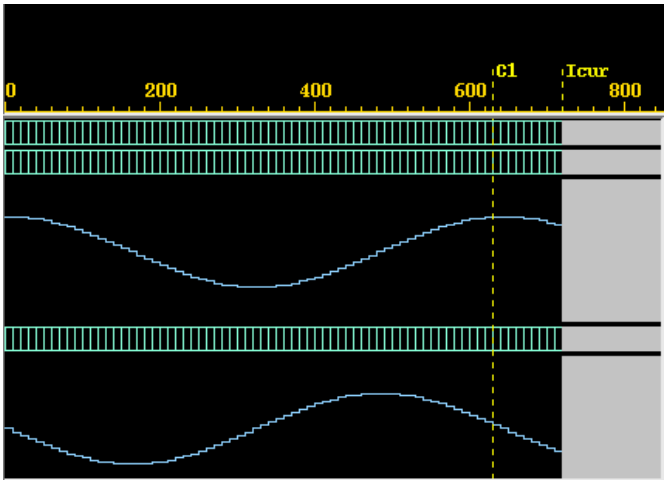

図 1: 動作波形

以下に回路図と面積と速度のレポートを示す。

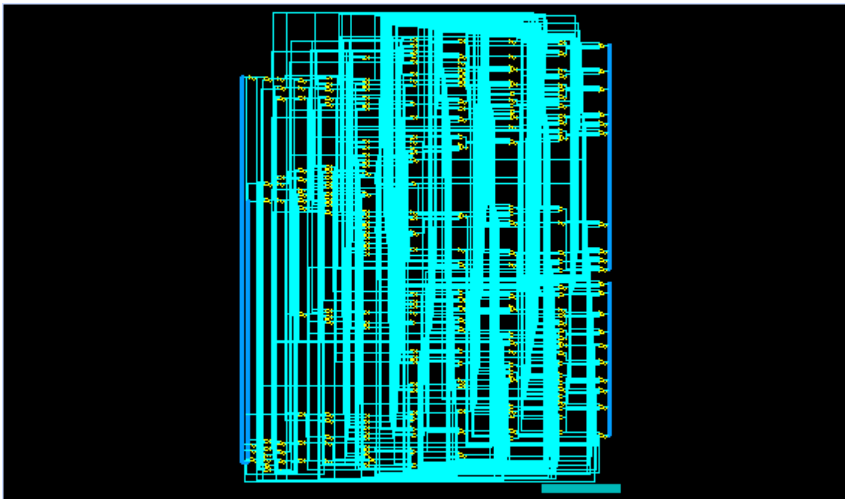

図 2: 回路図

```
****************************************
Report : area
Design : TWIDDLE
Version: 2003.06
Date   : Thu Dec 14 15:34:59 2006
****************************************

Library(s) Used:

    class (File: /usr/local/synopsys/U-2003.06-dc/libraries/syn/class.db)

Number of ports:               26
Number of nets:               246
Number of cells:              240
Number of references:          14

Combinational area:        359.000000
Noncombinational area:       0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:           359.000000
Total area:                 undefined
1
design_analyzer> report_timing -path full -delay max -max_paths 1 -nworst 1

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : TWIDDLE
Version: 2003.06
Date   : Thu Dec 14 15:34:59 2006
****************************************

Operating Conditions:
Wire Load Model Mode: top

  Startpoint: ADDR[2] (input port)
  Endpoint: W_I[8] (output port)
  Path Group: (none)
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  TWIDDLE            05x05                 class

  Point                                    Incr       Path
  -----------------------------------------------------------
  input external delay                     0.00       0.00 f
  ADDR[2] (in)                             0.00       0.00 f
  U32/Z (IV)                               1.27       1.27 r
  U35/Z (NR2)                              0.89       2.16 f
  U225/Z (IV)                              0.75       2.91 r
  U36/Z (NR2)                              0.54       3.45 f
  U185/Z (ND2)                             1.34       4.79 r
  U106/Z (ND4)                             0.65       5.44 f
  U70/Z (OR2)                              1.00       6.44 f
  U83/Z (NR4)                              2.40       8.84 r
  U138/Z (AN4)                             1.18      10.02 r
  U95/Z (AN4)                              1.36      11.38 r
  U17/Z (ND4)                              0.51      11.89 f
  W_I[8] (out)                             0.00      11.89 f
  data arrival time                                  11.89
  -----------------------------------------------------------
  (Path is unconstrained)


1
design_analyzer>
```