FFT設計 (CMULT設計)

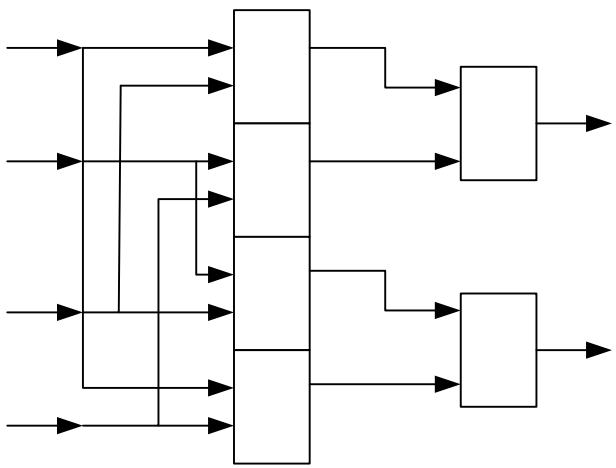
ファイヤー和田　知久
wada@ie.u-ryukyu.ac.jp
琉球大学・工学部・情報工学科　教授
http://www.ie.u-ryukyu.ac.jp/~wada

複素乗算器

- 以下のような複素乗算器を設計する
  - 用途は<14,6,t>なる信号とTwiddle Factor<10,0,t>の複素数どうしの乗算を行う
  - 2つの複素数 a+jb と c+jd の乗算は以下のようになります。

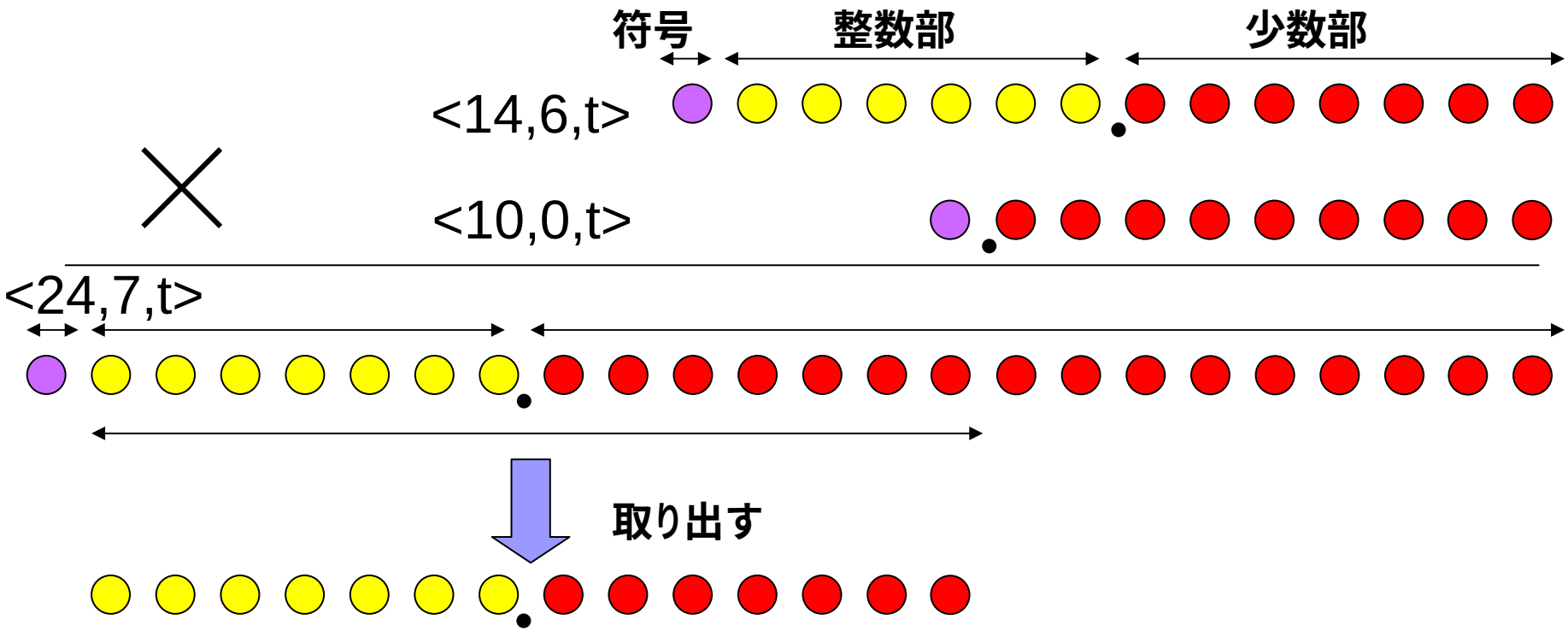
$$(a+jb)\times(c+jd)=(ac-bd)+j(bc+ad)$$

  - したがって、以下のような4つの固定小数点乗算器などで実現できる。

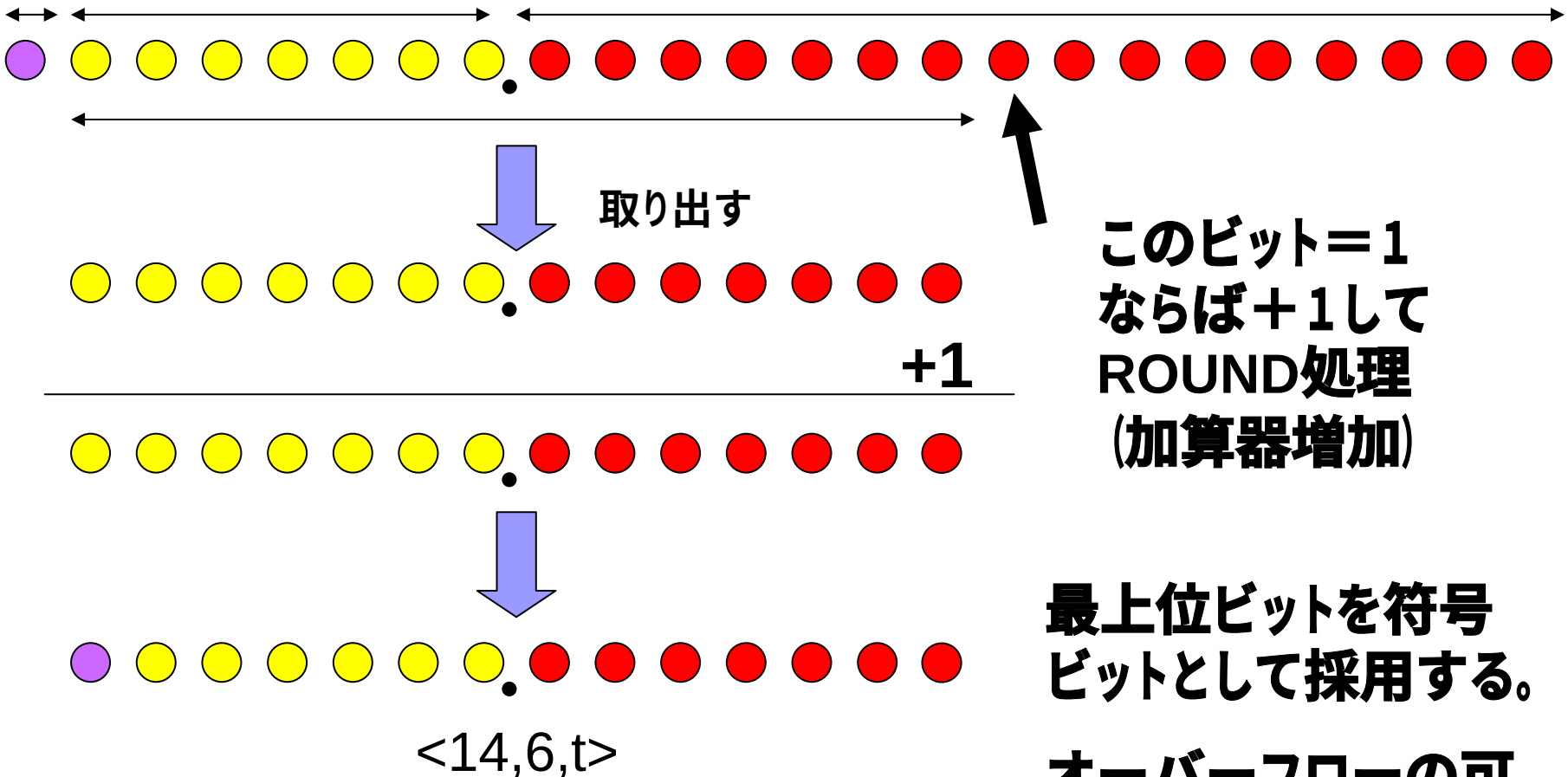


固定小数点乗算器(1)

- **固定小数点数**<14,6,t>**と**<10,0,t>**を乗算する**
  - **(1) 14ビットの数と10ビットの数を乗算すると24ビット長になる**
    しかし、ここでは<14,6,t>**の出力を得る**



固定小数点乗算器(2)



このビット＝1ならば＋1してROUND処理（加算器増加）

最上位ビットを符号ビットとして採用する。

オーバーフローの可能性ある。

固定小数点乗算器VHDLコード(multar.vhd)

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

entity MULTAR is
  port(in1    : in  std_logic_vector(13 downto 0);  -- <14,6,t>
       in2    : in  std_logic_vector(9  downto 0);  -- <10,0,t>
       outp   : out std_logic_vector(13 downto 0)); -- <14,6,t>
end MULTAR;

architecture RTL of MULTAR is
begin
  MULTAR : process(in1,in2)
    variable  var_tprod  : std_logic_vector(23 downto 0); -- <24,7,t>
    variable  var_round  : std_logic_vector(13 downto 0); -- <14,6,t>
  begin
    var_tprod := signed(in1) * signed(in2);
    if (var_tprod(8) = '1') then
       var_round := signed(var_tprod(22 downto 9)) + '1';
    else
       var_round := var_tprod(22 downto 9);
    end if;
    outp <= var_round;
  end process MULTAR;
end RTL;
```

新変数
variable

同一processの中で、すぐに計算結果を参照したい場合には、variableにしないと値が参照できない。

新変数 variable (1)

- Process文の中だけで使用できるvariableという変数がある
- 特徴
  - そのprocessの中だけで使用し、processの外からは参照できない
  - これまで信号の代入は A <= B+C; のように <= をもちいたが、variableでは v_A := v_B + v_C; のように := で代入する。
  - 信号代入 <= は実際の代入はprocces文を出るときに行われるので、
    B<=C;
    A<=B;
    では、CはAにそのprocess中に代入されないが、
  - Variable 代入 := ではその行の実行の瞬間に、代入が行われるので、前の行の計算結果を即座に次の行で利用できる。
    v_B := C;
    A <= v_B;
    では、CはAにそのprocess中に代入される。

新変数 variable（2）

```
architecture RTL of REI is
signal X, Y, Z, C : unsigned (3 downto 0);
signal A, B       : unsigned(3 downto 0);
begin
process (X, Y, Z, C) begin
  C <= Z;
  A <= X + C ;
  C <= Y;
  B <= X + C;
end process;
end RTL;
```

この例ではCにY,Zが代入されるが、それらは、processから出る時の一度なので、Cは前回のYの値を保持しているので、

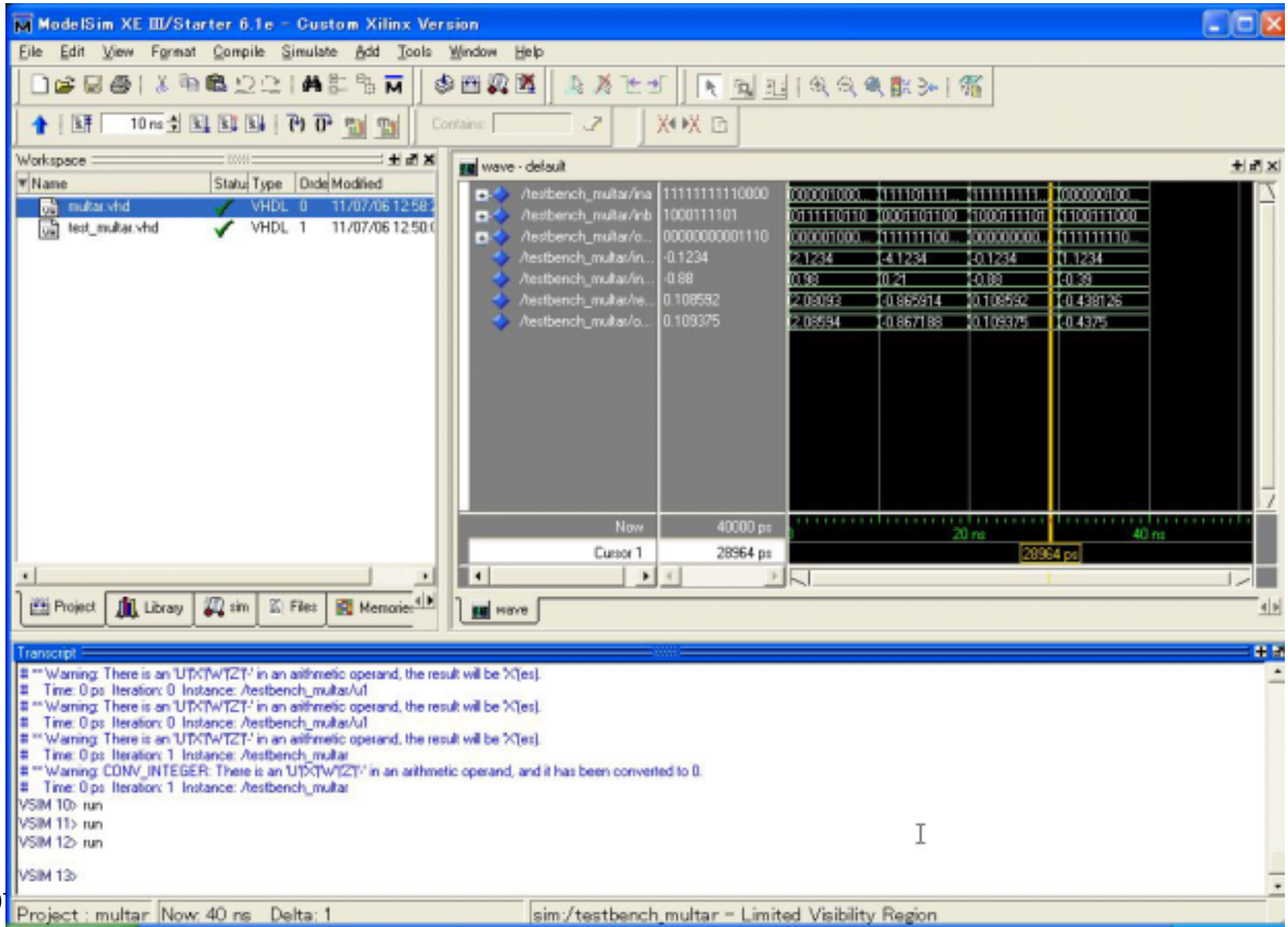
```
A <= X + Y
B <= X + Y
```

```
architecture RTL of REI is
signal X, Y, Z : unsigned (3 downto 0);
signal A, B    : unsigned(3 downto 0);
begin
process (X, Y, Z)
variable C : unsigned ( 3 downto 0);
begin
  C := Z;
  A <= X + C ;
  C := Y;
  B <= X + C;
end process;
end RTL;
```

この例ではCはvariableであり、その行で値が代入されるので、

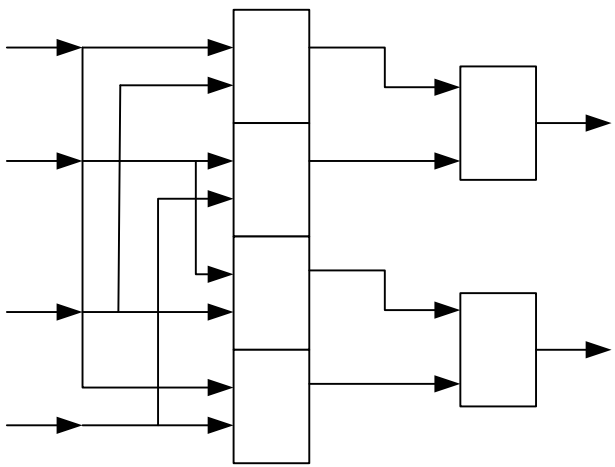
```
A <= X + Z
B <= X + Y
```

MULTAR動作波形


ModelSim XE III/Starter 6.1e - Custom Xilinx Version
File Edit View Format Compile Simulate Add Tools Window Help
Workspace
Name
multar.vhd VHDL 0
test_multar.vhd VHDL 1
wave - default
/testbench_multar/ina 1111111110000
/testbench_multar/inb 1000111101
/testbench_multar/o... 00000000001110
/testbench_multar/in... -0.1234
/testbench_multar/in... -0.88
/testbench_multar/re... 0.108582
/testbench_multar/o... 0.109375
Now 40000 ps
Cursor 1 28964 ps
20 ns
40 ns
Project Library sim Files Memories
wave
Transcript
** Warning: There is an 'UTXTWTZT-' in an arithmetic operand, the result will be 'X'(es).
Time: 0 ps Iteration: 0 Instance: /testbench_multar/u1
** Warning: There is an 'UTXTWTZT-' in an arithmetic operand, the result will be 'X'(es).
Time: 0 ps Iteration: 0 Instance: /testbench_multar/u1
** Warning: There is an 'UTXTWTZT-' in an arithmetic operand, the result will be 'X'(es).
Time: 0 ps Iteration: 1 Instance: /testbench_multar
** Warning: CONV_INTEGER: There is an 'UTXTWTZT-' in an arithmetic operand, and it has been converted to 0.
Time: 0 ps Iteration: 1 Instance: /testbench_multar
VSIM 10> run
VSIM 11> run
VSIM 12> run
VSIM 13>
Project : multar Now: 40 ns Delta: 1
sim:/testbench_multar - Limited Visibility Region

複素乗算器の設計

- 課題1で作成した乗算器を用いて、複素乗算器を設計できる。
- コードを次ページに示す。



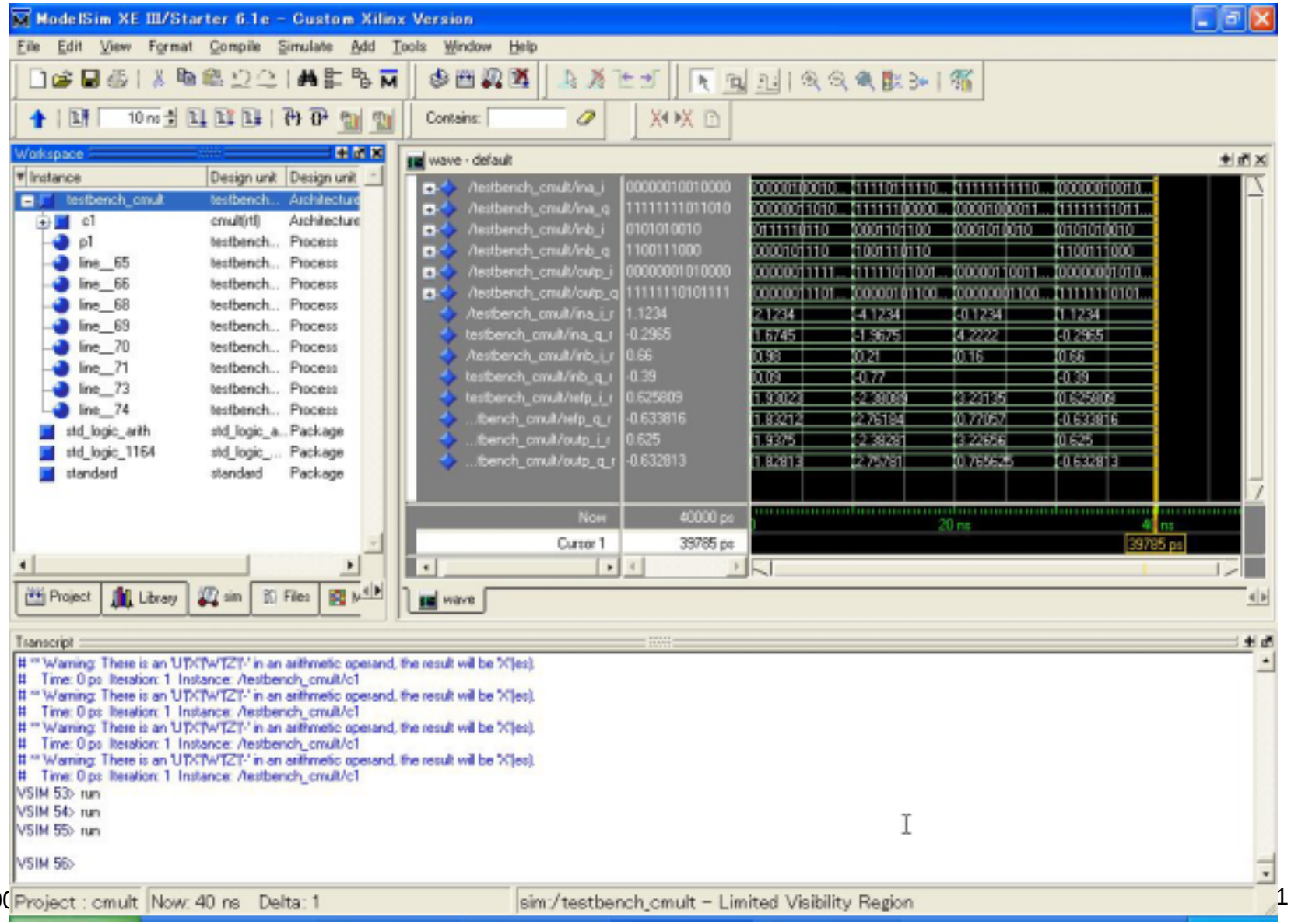
複素乗算器VHDLコード (cmult.vhd)

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

entity CMULT is
  port(AIN_I     : in  std_logic_vector(13 downto 0);  -- <14,6,t>
       AIN_Q     : in  std_logic_vector(13 downto 0);  -- <14,6,t>
       BIN_I     : in  std_logic_vector(9  downto 0);  -- <10,0,t>
       BIN_Q     : in  std_logic_vector(9  downto 0);  -- <10,0,t>
       YOUT_I    : out std_logic_vector(13 downto 0);  -- <14,6,t>
       YOUT_Q    : out std_logic_vector(13 downto 0)); -- <14,6,t>
end CMULT;

architecture RTL of CMULT is
-- component declaration
component MULTAR
  port(in1     : in  std_logic_vector(13 downto 0);  -- <14,6,t>
       in2     : in  std_logic_vector(9  downto 0);  -- <10,0,t>
       outp    : out std_logic_vector(13 downto 0)); -- <14,6,t>
end component;
signal S1, S2, S3, S4 : std_logic_vector(13 downto 0);  -- <14,6,t>
begin
M1: MULTAR port map (AIN_I, BIN_I, S1);
M2: MULTAR port map (AIN_Q, BIN_Q, S2);
M3: MULTAR port map (AIN_Q, BIN_I, S3);
M4: MULTAR port map (AIN_I, BIN_Q, S4);
YOUT_I <= signed(S1) - signed(S2);
YOUT_Q <= signed(S3) + signed(S4);
end RTL;
```

CMULT動作波形


ModelSim XE III/Starter 6.1e - Custom Xilinx Version
File Edit View Format Compile Simulate Add Tools Window Help
Workspace
Instance
Design unit
testbench_cmult
c1
p1
line__65
line__66
line__68
line__69
line__70
line__71
line__73
line__74
std_logic_arith
std_logic_1164
standard
Project
Library
sim
Files
wave - default
/testbench_cmult/ina_i
/testbench_cmult/ina_q
/testbench_cmult/inb_i
/testbench_cmult/inb_q
/testbench_cmult/outp_i
/testbench_cmult/outp_q
1.1234
-0.2965
0.66
-0.39
0.625809
-0.633816
0.625
-0.632813
Now
40000 ps
Cursor 1
39785 ps
20 ns
wave
Transcript
** Warning: There is an 'UTXTWTZT-' in an arithmetic operand, the result will be 'X'(es).
Time: 0 ps Iteration: 1 Instance: /testbench_cmult/c1
VSIM 53> run
VSIM 54> run
VSIM 55> run
VSIM 56>
Project : cmult
Now: 40 ns Delta: 1
sim:/testbench_cmult - Limited Visibility Region