4点FFT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

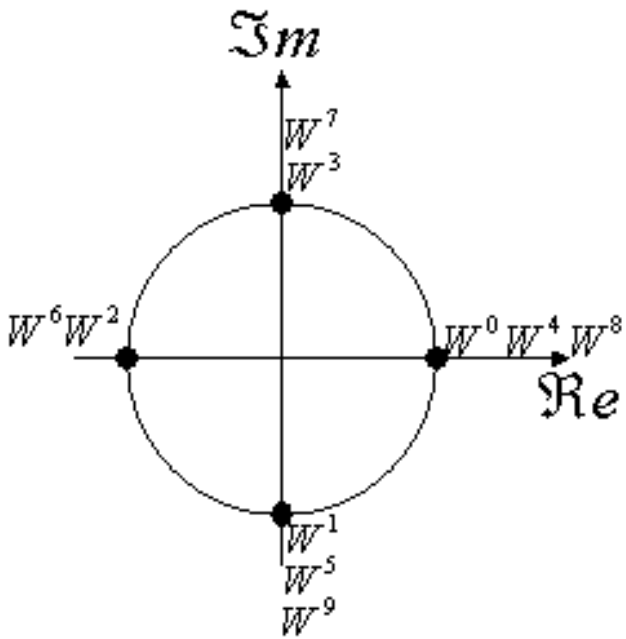
4点FFT

- FFT演算の内容をさらに理解するために、4点FFTを行列表現したものを示します。
- N=4ですので、行列表現では4X4のTwiddle Factorが並ぶ行列ができます。Twiddle Factorは図6右に示すように、W^0, W^1,....の順で複素平面状の半径1の円周を回転します。

$$W = e^{-j\left(\frac{2\pi}{4}\right)}$$

$$X(k) = \sum_{n=0}^{3} x(n) \cdot W^{nk} \quad (k = 0,1,2,3)$$

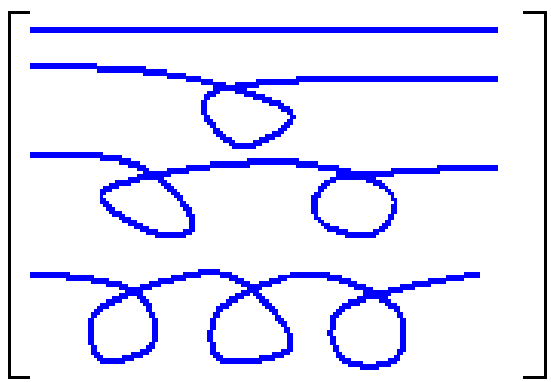
$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$



4点FFT (2)

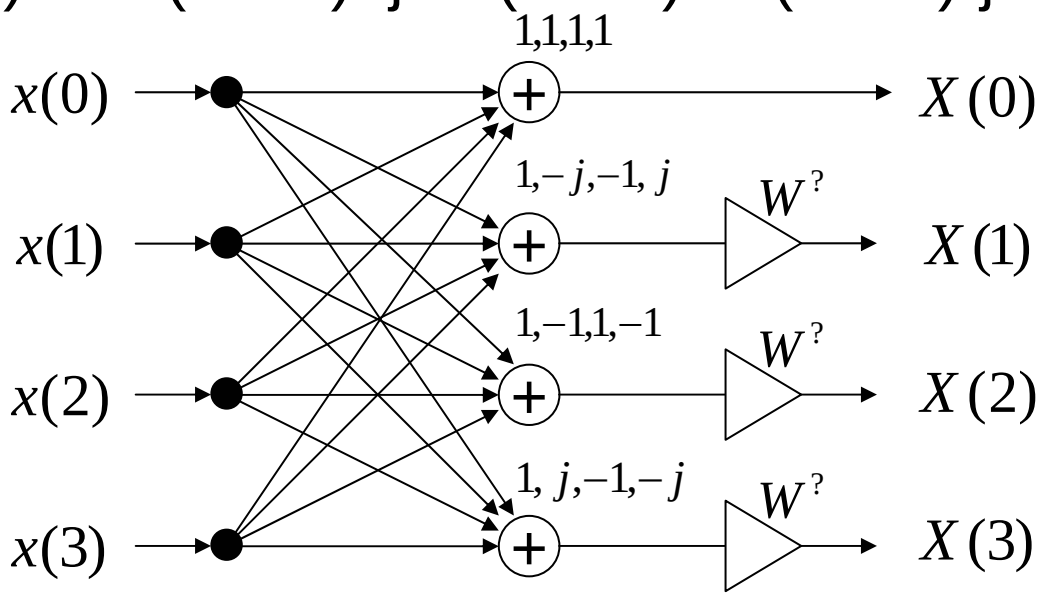
- Twiddle Factorの行列に数値を入れたものを以下に示します。
- 一行目は1が並び、これは同じ値が続く周波数0の回転に対応しています。
- 2行目は、1, -j, -1, jとなり、これは複素平面状での負の方向への1回転を示しています。
- 3行目は1, -1, 1, -1となり、これは複素平面状での2回転を示しています。
- 3行目は、1, j, -1, -jとなり、これは負の方向への3回転ですし、正の方向への1回転と考えることもできます。
- すなわち、FFT演算とは複素数による回転を示す波形の周波数の異なるものそれぞれと、入力信号x(n)の内積計算をしていることになります。

$$\begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} =$$



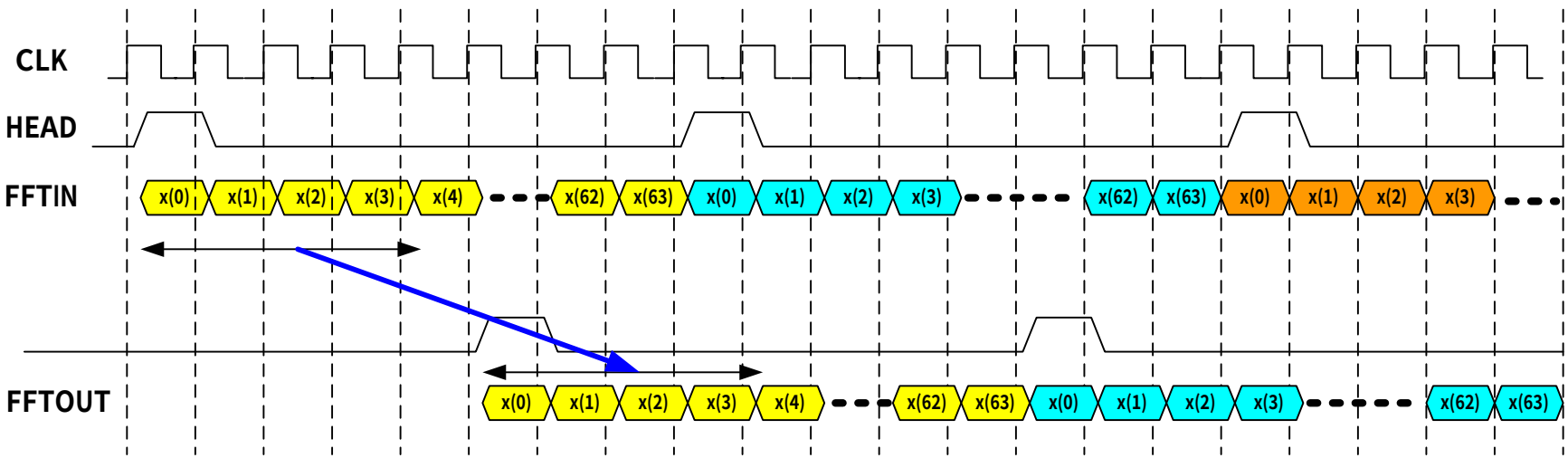
以下の演算をする4FFT回路を設計します。

```
% stage 3  =(4FFT)
for nn = 0:4:60
% nnは0,4,8,…, 60となる意味
   x3(nn+0) = x2(nn+0)  +x2(nn+1)+x2(nn+2)  +x2(nn+3);
   x3(nn+1) = x2(nn+0)-j*x2(nn+1)-x2(nn+2)+j*x2(nn+3);
   x3(nn+2) = x2(nn+0)  -x2(nn+1)+x2(nn+2)  -x2(nn+3);
   x3(nn+3) = x2(nn+0)+j*x2(nn+1)-x2(nn+2)-j*x2(nn+3);
end;
```

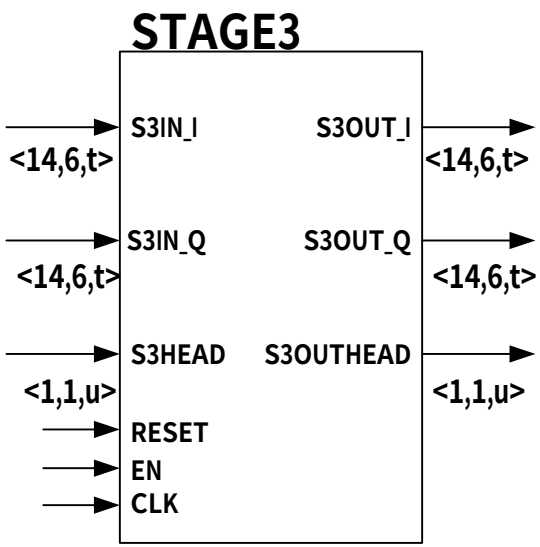


動作波形

- HEAD信号を先頭に、６４点が入力される。
- 4点ごとに前頁で示した、演算を行う。



4 FFTのENTITY



RESET: ‘1’でリセット

EN: ‘1’の期間のみ動作

固定小数点フォーマット

- 今回の課題では0.0625のような小数点以下の数をデジタル回路で取り扱う必要があります。たとえば4ビットの2進数"0111"ですが、小数点をどこの位置に置くかで表す値が変わってきます。たとえば、"01.11"ならば10進数表現では＋1.75、"0111."ならば、10進表現では＋7です。
- また、4ビットの2進数ですが、それが符号無し数すなわち正または0の数を表しているか、2の補数表現で正または負の数を表しているかを区別する必要があります。"11.10"は符号無し数であれば、10進表現で+3.50となり、2の補数表現であれば、10進表現で-0.50となります。
- すなわち、同じ4ビットの2進数でも、「小数点の位置」と「符号無し表現か2の補数表現か」を明確にしないとまったく異なった数に対応してしまいます。
- ここではSignal Processing Workbenchで用いられている固定小数点の属性表記方法を解説し、用います。
- <8,2,t>と表記すると、その信号は
- 8 ： 全体のビット数が8ビット
  2 ： 整数ビットが2ビット
  t ： two's complement ということで2の補数表現、すなわち最上位ビットは符号ビットとなる
- という意味です。したがって、"01101111"なる8ビットの数の属性が<8,2,t>とすると、

0	1	1	0	1	1	1	1
符号ビット	整数部		小数部				

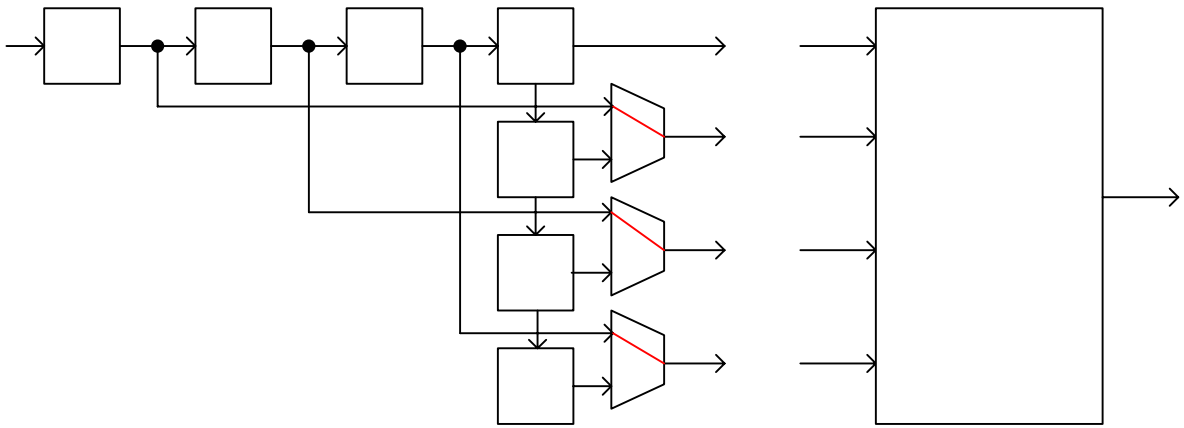
- 01101111符号ビット整数部小数部ということになり、小数部は5ビットとなり、10進表現では＋3.46875に対応します。

- <8,2,u>と表記すると、その信号は
- 8 ： 全体のビット数が8ビット
  2 ： 整数ビットが2ビット
  u ： unsignedということで、符号無し数すなわち負の数を表さない
- という意味です。したがって、同じ8ビットの数"01101111"の属性が<8,2,u>とすると、

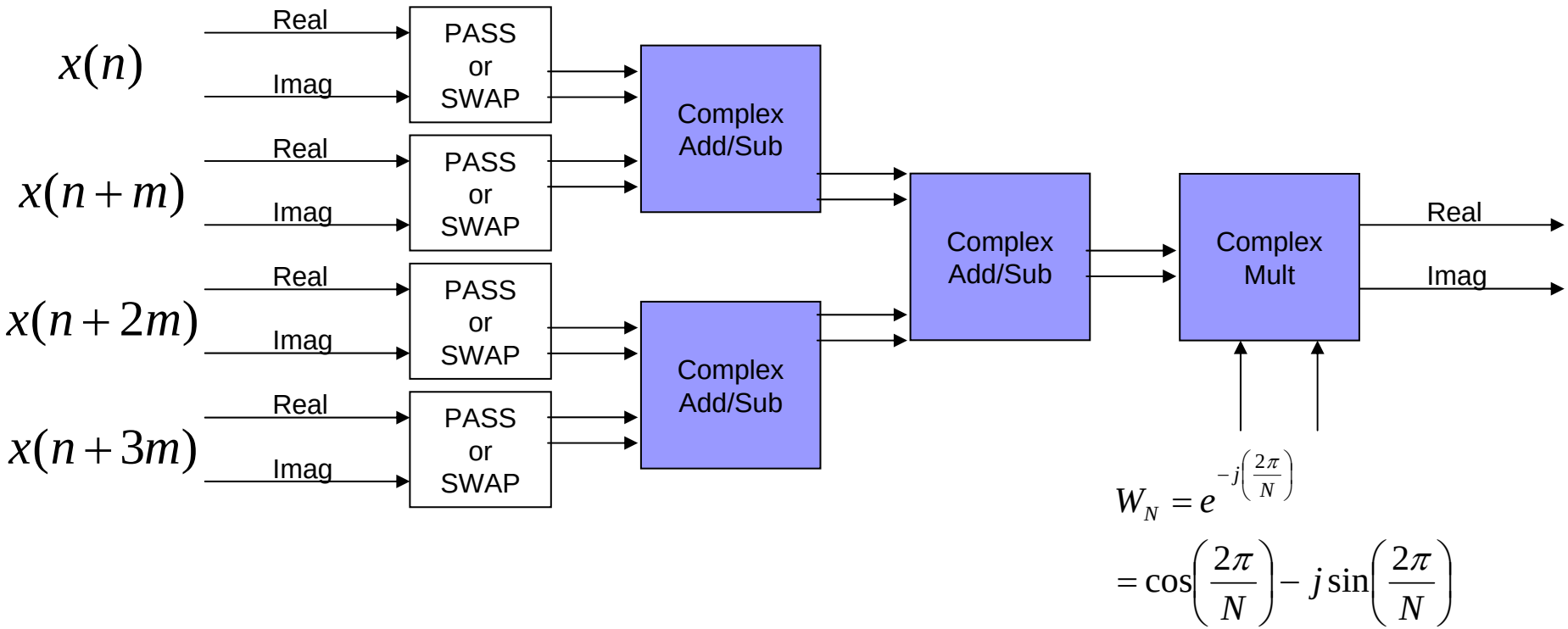
0	1	1	0	1	1	1	1
整数部		小数部					

- 01101111整数部小数部ということになり、小数部は6ビットとなり、10進表現では＋1.734375に対応します。
- 整数部のビット数が多いほど表現できる最大数すなわちレンジが大きくなり、小数部のビット数が多いほど表現できる最小数が小さくなり、解像度が向上します。

4 FFTのARCHITECTURE (1)

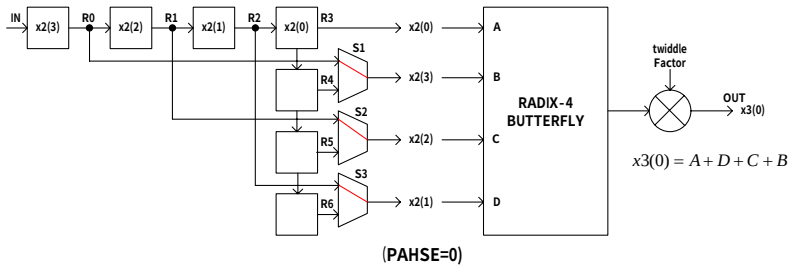


Radix-4 computation unit

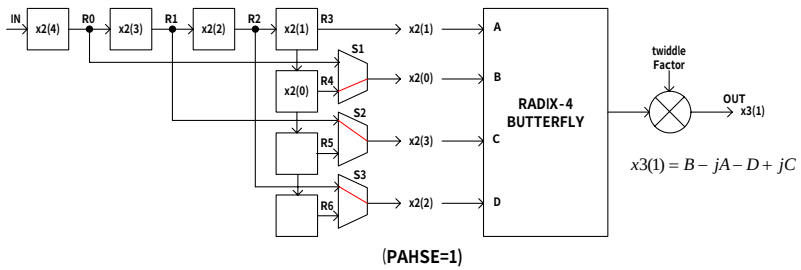


- complex add : (a+bj)+(c+dj)= (a+c)+(b+d)j
  - 1 complex add = 2 real add
- Complex mult : (a+bj)(c+dj) = (ac-bd)+(bc+ad)j
  - 1 complex mult = 4 real mult + 2 real add

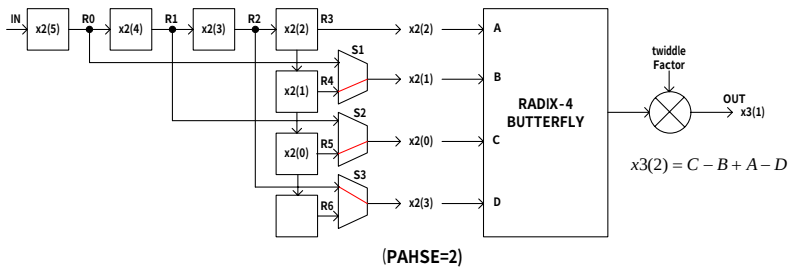
4 FFTのARCHITECTURE (1)



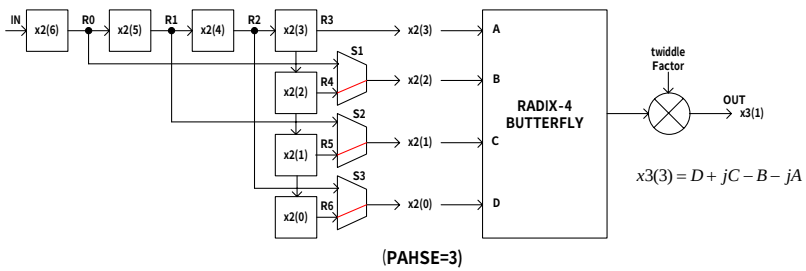
(PAHSE=0)



(PAHSE=1)

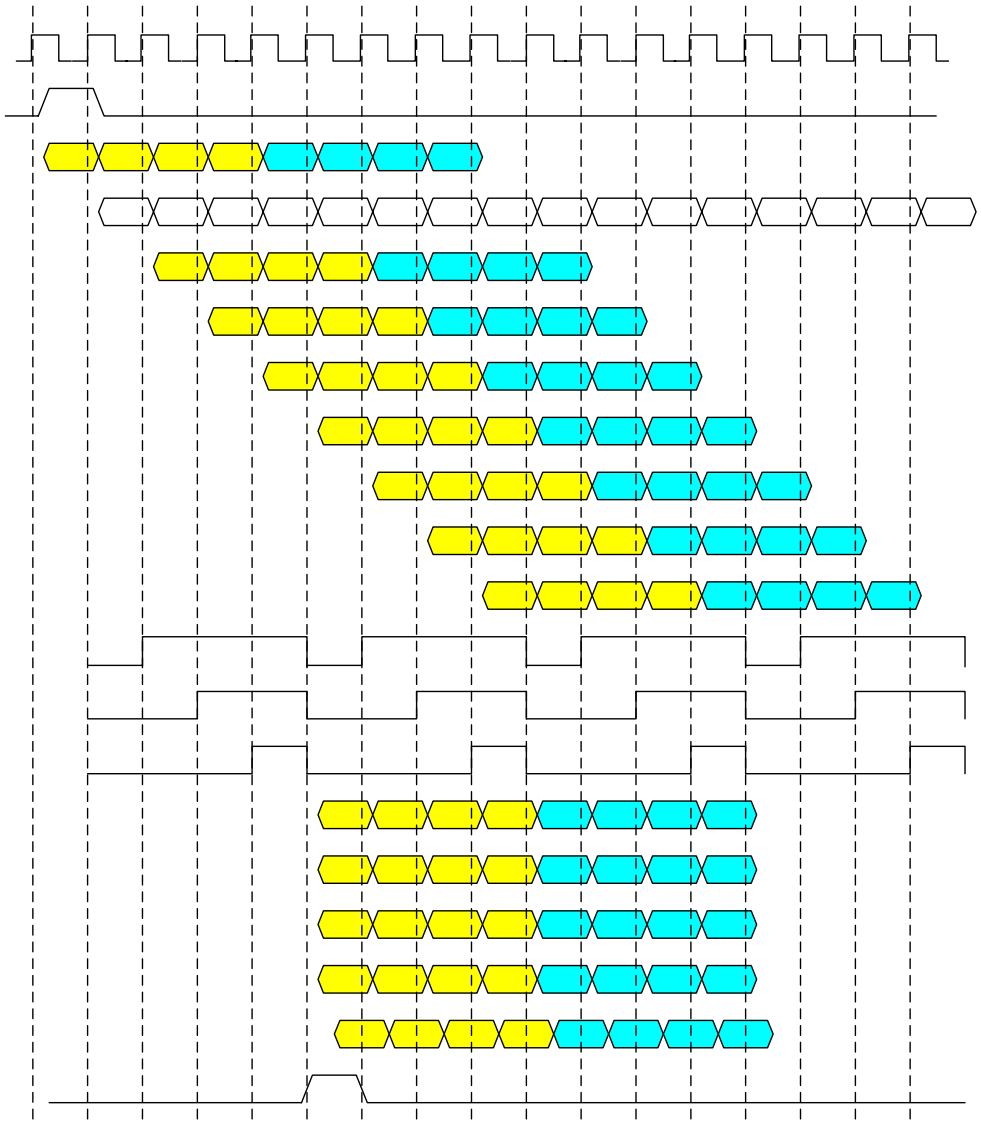


(PAHSE=2)

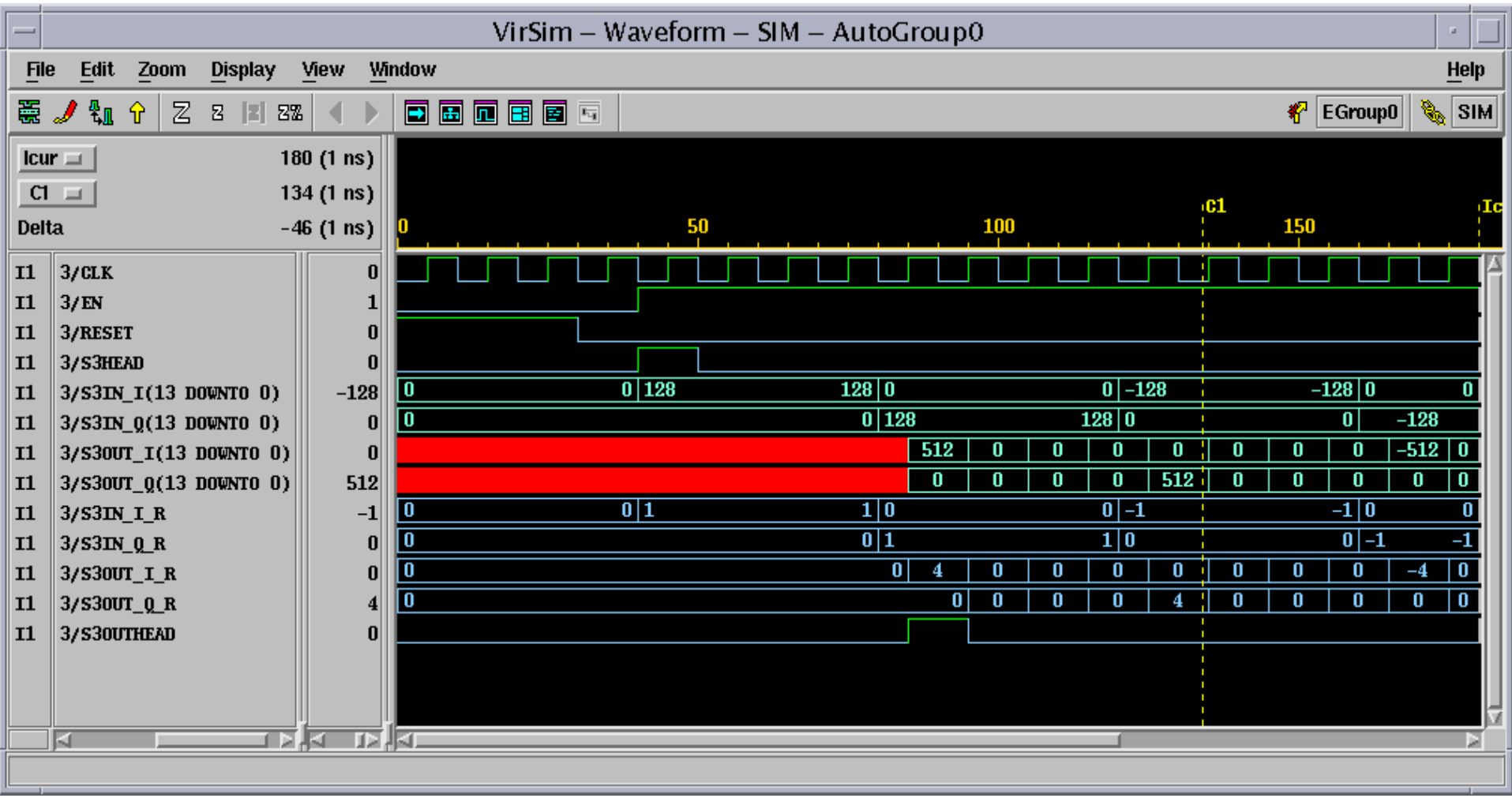


(PAHSE=3)

4FFTのARCHITECTURE (動作波形)



STAGE3 operation



宿題7　4FFT設計

1. 以下を参考に、4FFT回路を設計せよ！
2. テストベンチで計算されている、4FFTの入力値と出力値の関係をレポートに報告せよ！
   - テンプレート配布
     - http://www.ie.u-ryukyu.ac.jp/~wada/cad06/stage3/stage3.vhd
   - テストベンチ配布
     - http://www.ie.u-ryukyu.ac.jp/~wada/cad06/stage3/test_stage3.vhd
   - 追加説明図面、回答例のプリント配布
     - http://www.ie.u-ryukyu.ac.jp/~wada/cad06/stage3/FFT4.pdf

stage3.vhdコードの注意事項

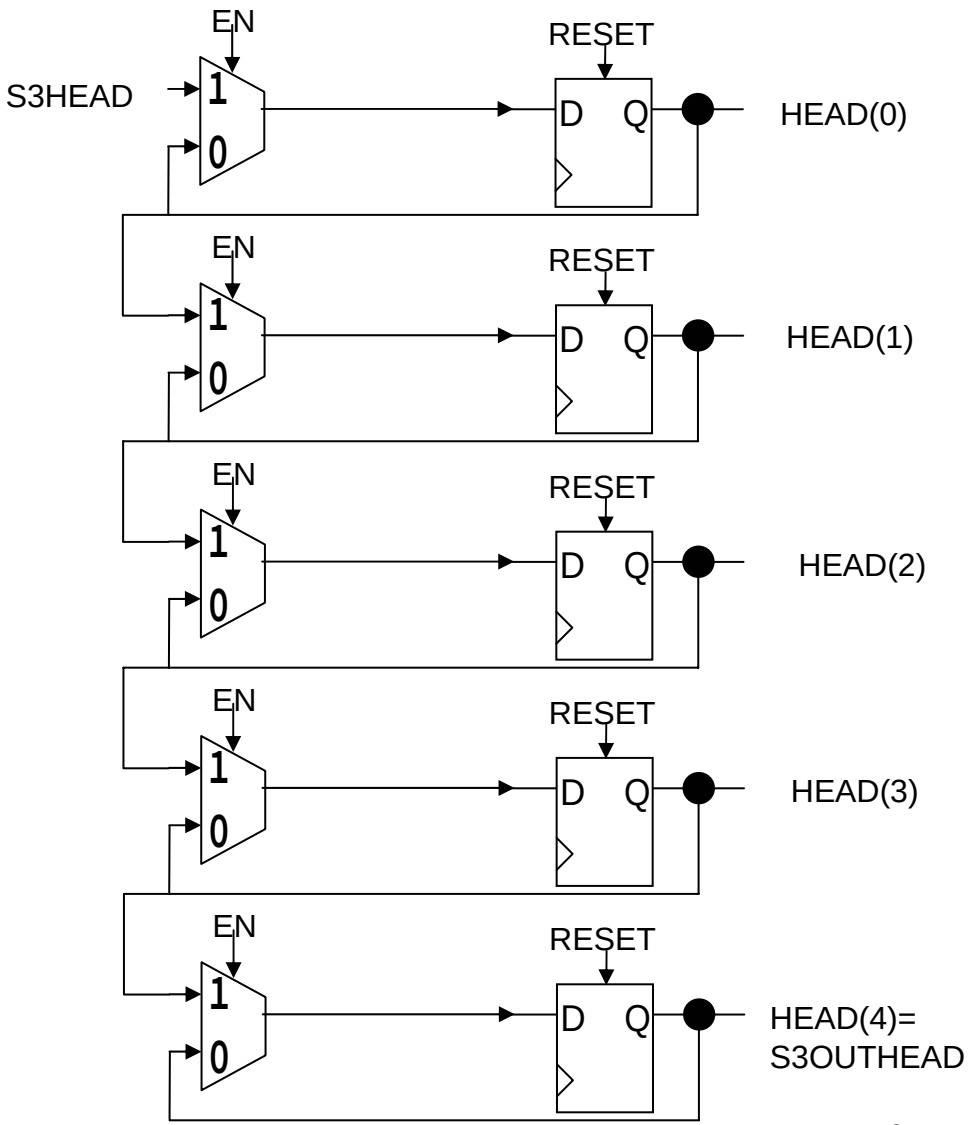
1. unsigned と signed 演算を行うために、ライブラリは IEEE.std_logic_arith.allを使用し、加算の時にunsigned, signedを指定。
2. 複素数jを乗じての加算は、I/Qの交代と符号変化で対応
3. サイズ7のアレイを使用、アレイのインデックスは整数である。

test_stage3.vhdコードの注意事項

1. 実数型realを使用
2. realのアレイを入力データとして使用
3. 整数→std_logic_vector変換関数使用
   - ライブラリ IEEE.std_logic_arith.allに含まれる
4. std_logic_vector →整数変換関数使用
5. 整数→実数変換real使用

人間コンパイラー (1)

```
-- OUTHEAD GENERATION
   process ( CLK , RESET) begin
     if (RESET = '1') then
        HEAD(0) <= '0';
        HEAD(1) <= '0';
        HEAD(2) <= '0';
        HEAD(3) <= '0';
        HEAD(4) <= '0';
     elsif (CLK'event and CLK = '1' ) then
       if ( EN = '1' ) then
          HEAD(0) <= S3HEAD;
          HEAD(1) <= HEAD(0);
          HEAD(2) <= HEAD(1);
          HEAD(3) <= HEAD(2);
          HEAD(4) <= HEAD(3);
       end if;
     end if;
 end process;
   S3OUTHEAD <= HEAD(4);
```



人間コンパイラー（2）

```
-- 64 CYCLE COUNTER
    process ( CLK, RESET ) begin
    if (RESET = '1') then
        COUNT <= "000000";
    elsif (CLK'event and CLK = '1' ) then
        if ( EN = ‘1’ ) then
            if (S3HEAD = '1') then
                COUNT <= "000000";
            else
                COUNT <= unsigned(COUNT) + '1';
            end if;
        end if;
    end if;
    end process;
```