Project #2

2D DCT Design in JPEG Image Compression

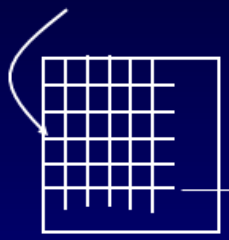
DCT based image compression process

JPEG Encoder Block Diagram

512 x 512 image

8×8 blocks

Source image X



DCT → *Y* → **Quantizer** → *Z* → **Entropy Encoder** → **Compressed Image Data**

$$Y = T \cdot X \cdot T'$$

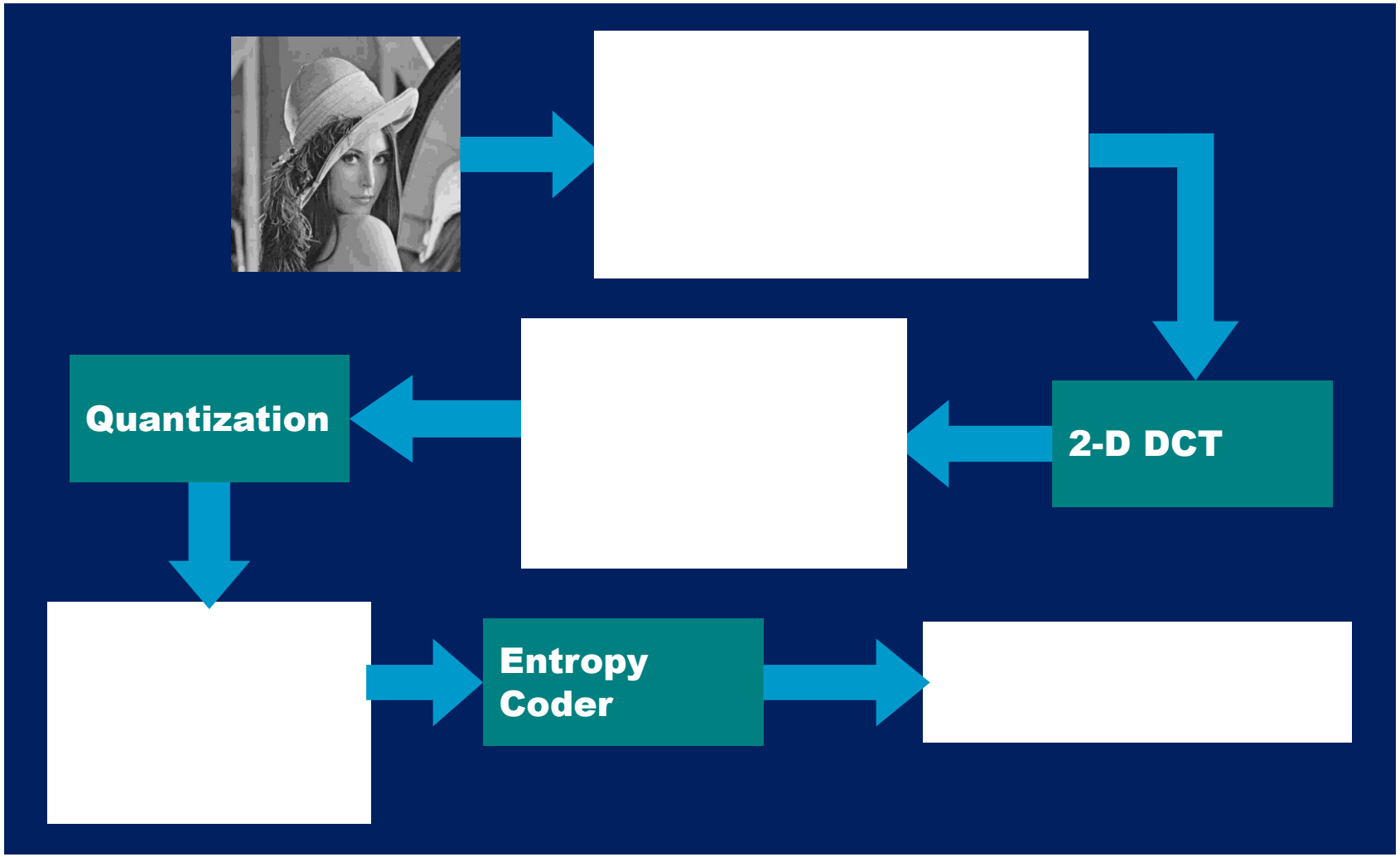
$$\text{Round}\left(\frac{T \cdot Y \cdot T'}{Q}\right)$$

- DCT is used in current international image/video coding standards
  - JPEG, MPEG, H.261, H.263

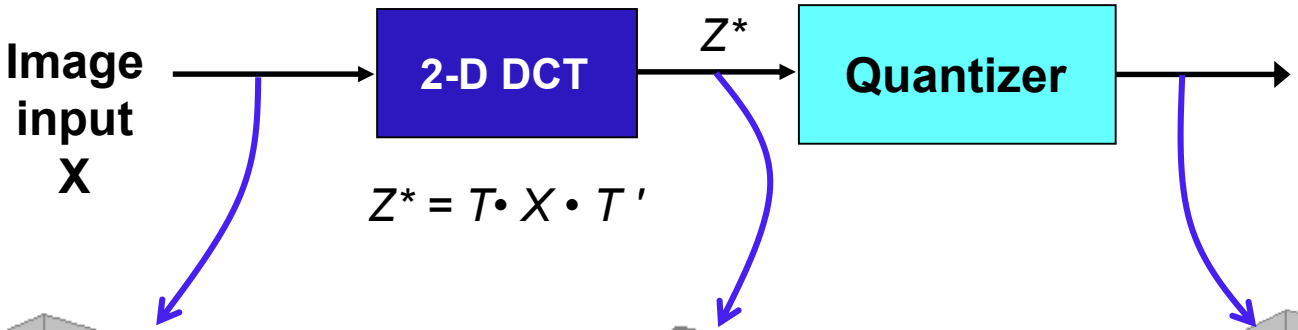
$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

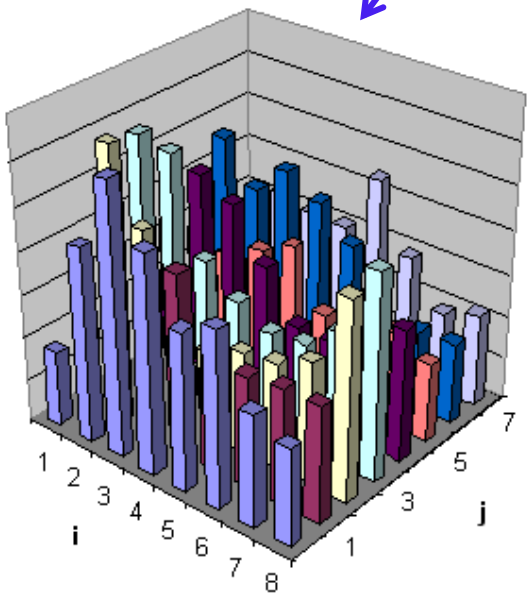
Lossy Compression !

DCT based image compression process

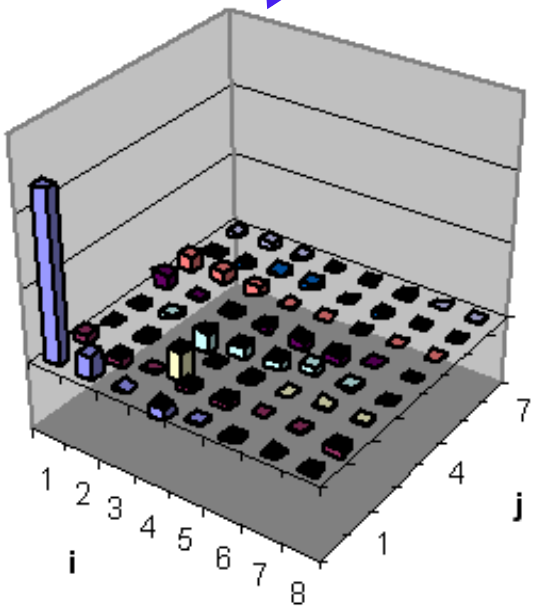


DCT based image compression process

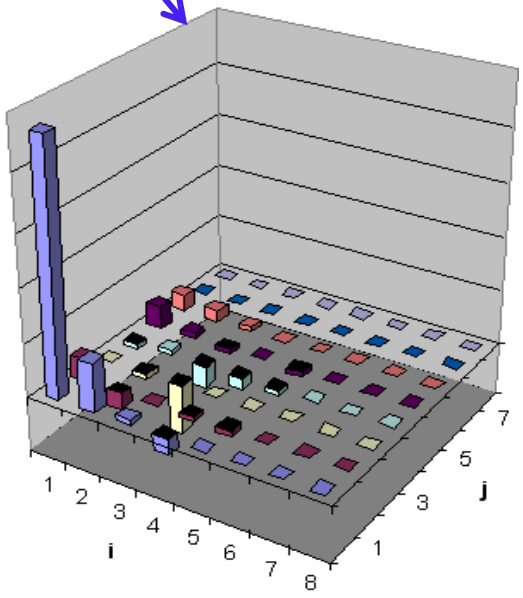




Spatial domain



Frequency domain



After Quantization

Project #2

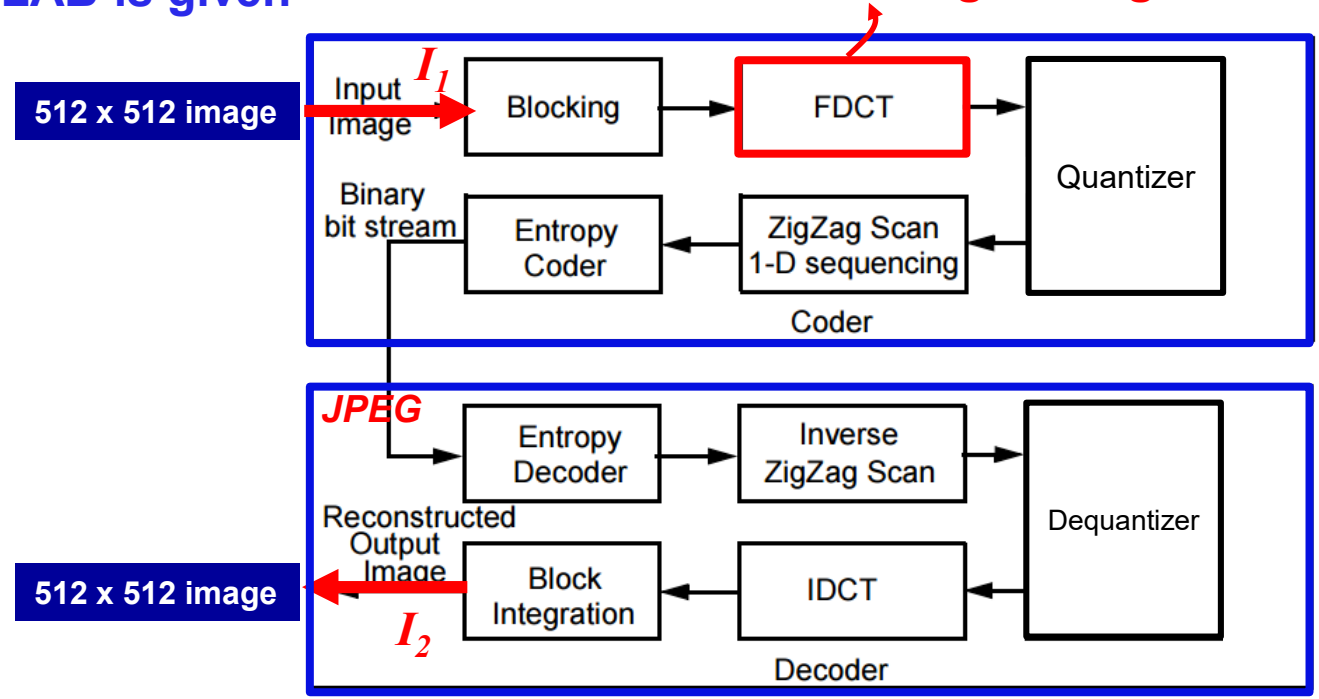
Please make a hardware for 2D DCT operations in JPEG Compression

Project #2

JPEG CODEC Architecture

MATLAB is given

Need to do the Verilog coding !



MATLAB environments are given including Data IO between Verilog and MATLAB simulation

Discrete Cosine Transform

What is Discrete Cosine Transform?

$$X(k) = e(k) \sum_{n=0}^{N-1} x(n) \cos[\frac{(2n+1)\pi k}{2N}], \quad k = 0, 1, \cdots, N-1$$

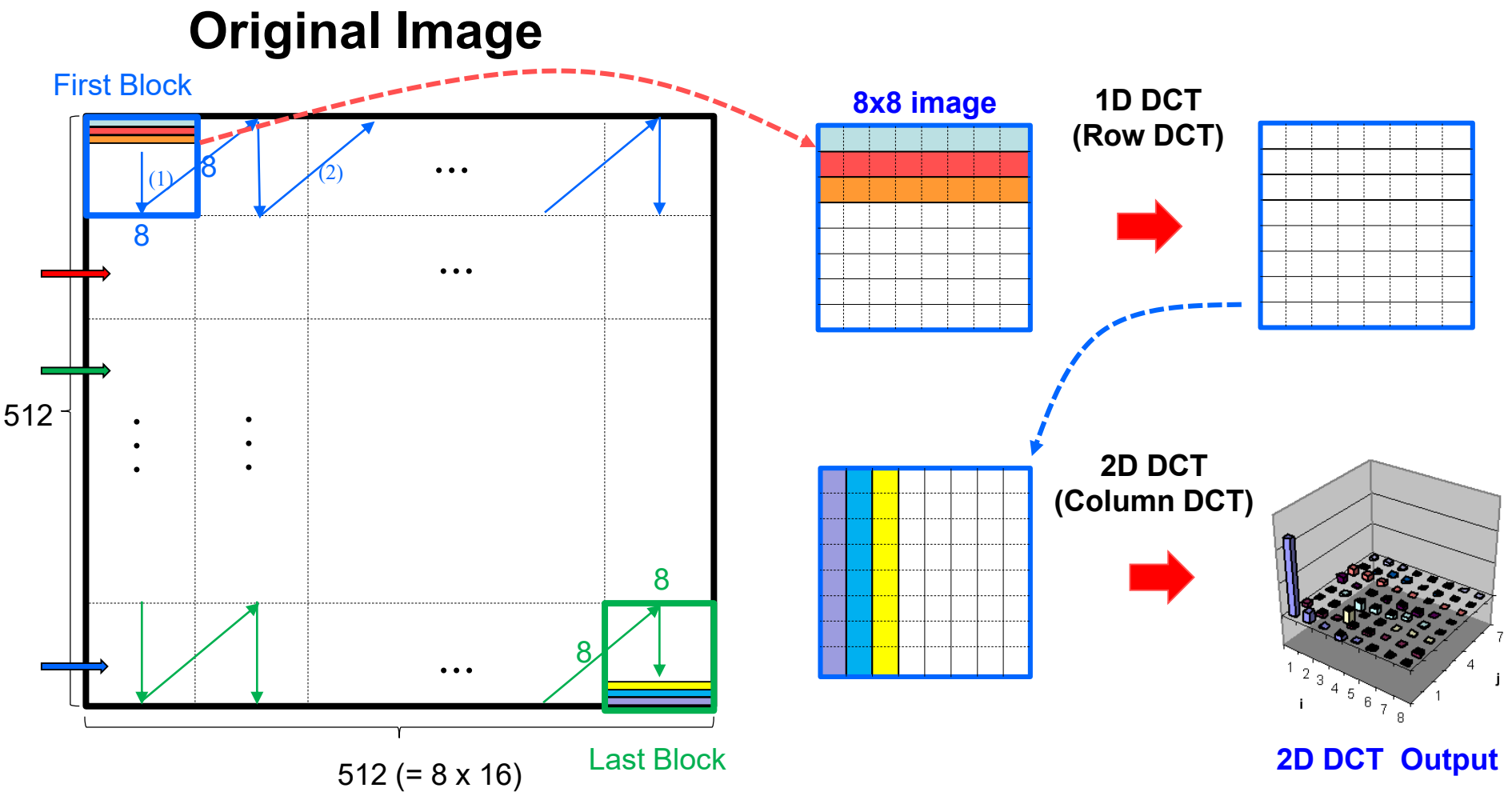
$$e(k) = \begin{cases} \frac{1}{\sqrt{2}}, & if\ k = 0, \\ 1, & otherwise. \end{cases}$$

ex) 8-point DCT presented in matrix multiplication

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} c_4 & c_4 & c_4 & c_4 & c_4 & c_4 & c_4 & c_4 \\ c_1 & c_3 & c_5 & c_7 & c_9 & c_{11} & c_{13} & c_{15} \\ c_2 & c_6 & c_{10} & c_{14} & c_{18} & c_{22} & c_{26} & c_{30} \\ c_3 & c_9 & c_{15} & c_{21} & c_{27} & c_1 & c_7 & c_{13} \\ c_4 & c_{12} & c_{20} & c_{28} & c_4 & c_{12} & c_{20} & c_{28} \\ c_5 & c_{15} & c_{25} & c_3 & c_{13} & c_{23} & c_1 & c_{11} \\ c_6 & c_{18} & c_{30} & c_{10} & c_{22} & c_2 & c_{14} & c_{26} \\ c_7 & c_{21} & c_3 & c_{17} & c_{31} & c_{13} & c_{27} & c_9 \end{bmatrix} \cdot \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix}$$

$$where\ c_i = \frac{1}{2} \times \cos\frac{i\pi}{16}$$

Discrete Cosine Transform



Discrete Cosine Transform

DCT:

$$X(k) = e(k) \sum_{n=0}^{N-1} x(n) \cos[\frac{(2n+1)\pi k}{2N}], \quad k = 0, 1, \cdots, N-1$$

$Z = Tx^t$

Note the symmetry of the DCT coef. matrix

$$T = \begin{bmatrix} c_4 & c_4 & c_4 & c_4 & c_4 & c_4 & c_4 & c_4 \\ c_1 & c_3 & c_5 & c_7 & -c_7 & -c_5 & -c_3 & -c_1 \\ c_2 & c_6 & -c_6 & -c_2 & -c_2 & -c_6 & c_6 & c_2 \\ c_3 & -c_7 & -c_1 & -c_5 & c_5 & c_1 & c_7 & -c_3 \\ c_4 & -c_4 & -c_4 & c_4 & c_4 & -c_4 & -c_4 & c_4 \\ c_5 & -c_1 & c_7 & c_3 & -c_3 & -c_7 & c_1 & -c_5 \\ c_6 & -c_2 & c_2 & -c_6 & -c_6 & c_2 & -c_2 & c_6 \\ c_7 & -c_5 & c_3 & -c_1 & c_1 & -c_3 & c_5 & -c_7 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

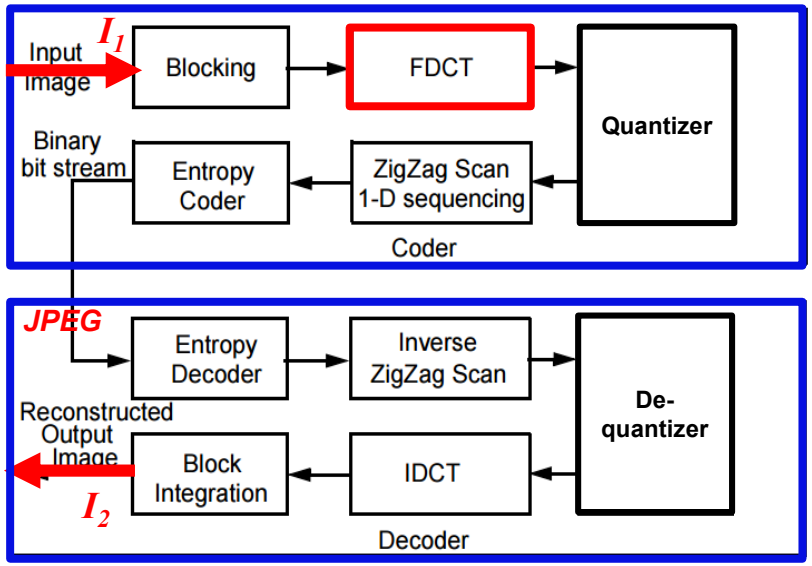
$C_0 = 0.500000000000000$
$C_1 = 0.490392640201615$
$C_2 = 0.461939766255643$
$C_3 = 0.415734806151273$
$C_4 = 0.353553390593274$
$C_5 = 0.277785116509801$
$C_6 = 0.191341716182545$
$C_7 = 0.097545161008064$

Quantization is decided by student

$$c_k = \frac{1}{2} \times \cos\frac{k\pi}{16}$$

Project #2

JPEG CODEC Architecture



2D DCT

Large quantization bit-width, Large Area, Large Power

➡

Good image !

PSNR : 36.5 dB

Small quantization bit-width, Small Area, Small Power

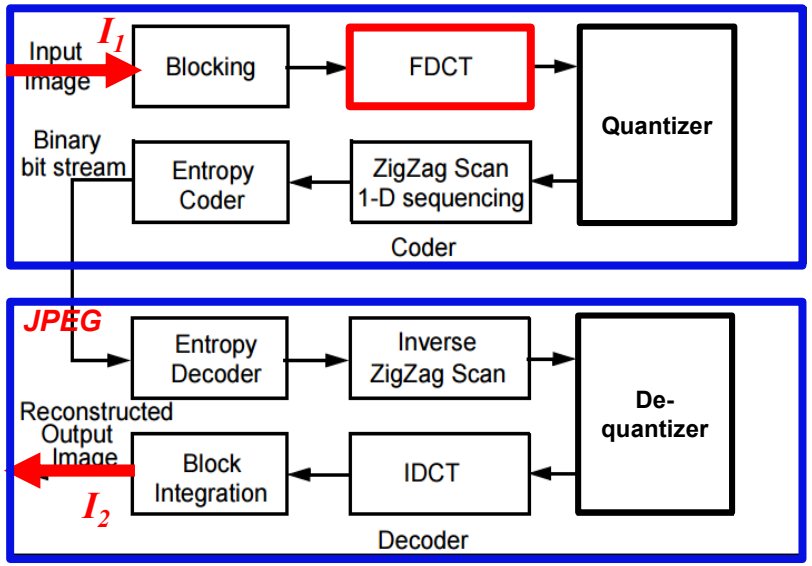
➡

Poor image !

PSNR : 24.6 dB

Project #2

For Image Quality Measure: PSNR !



$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N}$$

$$PSNR = 10 \log_{10}\left(\frac{R^2}{MSE}\right)$$

where M, N = 512, R = 255.

PSNR : 36.5 dB

PSNR : 24.6 dB

PSNR Measure Process is already in the Matlab code

Project #2

Please design a lowest cost

(minimum area and minimum power)

2D DCT hardware while satisfying the PSNR value of 29.5 dB !

Some Design Tips ?

Project HINTS: Simplification ?

DCT:

$$X(k) = e(k) \sum_{n=0}^{N-1} x(n) \cos[\frac{(2n+1)\pi k}{2N}], \quad k = 0, 1, \cdots, N-1$$

$Z = Tx^t$ **Note the symmetry of the DCT coef. matrix**

$$T = \begin{bmatrix} c_4 & c_4 & c_4 & c_4 \\ c_1 & c_3 & c_5 & c_7 \\ c_2 & c_6 & -c_6 & -c_2 \\ c_3 & -c_7 & -c_1 & -c_5 \\ c_4 & -c_4 & -c_4 & c_4 \\ c_5 & -c_1 & c_7 & c_3 \\ c_6 & -c_2 & c_2 & -c_6 \\ c_7 & -c_5 & c_3 & -c_1 \end{bmatrix} \begin{bmatrix} c_4 & c_4 & c_4 & c_4 \\ -c_7 & -c_5 & -c_3 & -c_1 \\ -c_2 & -c_6 & c_6 & c_2 \\ c_5 & c_1 & c_7 & -c_3 \\ c_4 & -c_4 & -c_4 & c_4 \\ -c_3 & -c_7 & c_1 & -c_5 \\ -c_6 & c_2 & -c_2 & c_6 \\ c_1 & -c_3 & c_5 & -c_7 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

$$c_k = \frac{1}{2} \times \cos\frac{k\pi}{16}$$

Even DCT

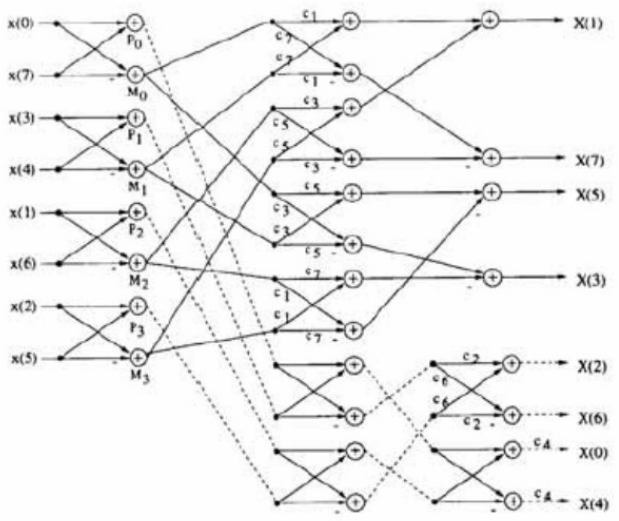
$$\begin{bmatrix} z_0 \\ z_2 \\ z_4 \\ z_6 \end{bmatrix} = \begin{bmatrix} c_4 & c_4 & c_4 & c_4 \\ c_2 & c_6 & -c_6 & -c_2 \\ c_4 & -c_4 & -c_4 & c_4 \\ c_6 & -c_2 & c_2 & -c_6 \end{bmatrix} \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \end{bmatrix}$$

Odd DCT

$$\begin{bmatrix} z_1 \\ z_3 \\ z_5 \\ z_7 \end{bmatrix} = \begin{bmatrix} c_1 & c_3 & c_5 & c_7 \\ c_3 & -c_7 & -c_1 & -c_5 \\ c_5 & -c_1 & c_7 & c_3 \\ c_7 & -c_5 & c_3 & -c_1 \end{bmatrix} \begin{bmatrix} x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix}$$

Sub-expression (Computation) Sharing ?

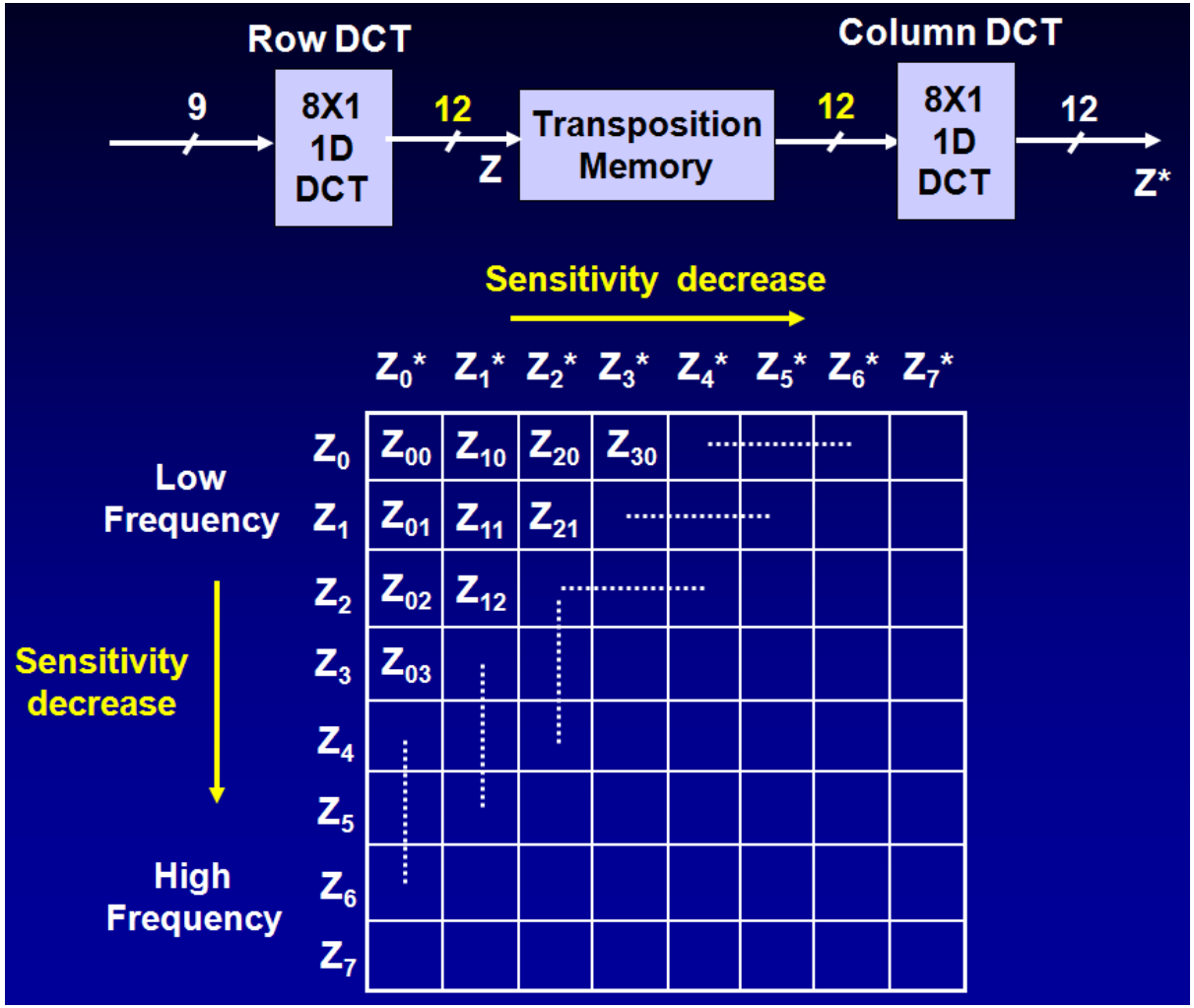
$$T = \begin{bmatrix} c_4 & c_4 & c_4 & c_4 \\ c_1 & c_3 & c_5 & c_7 \\ c_2 & c_6 & -c_6 & -c_2 \\ c_3 & -c_7 & -c_1 & -c_5 \\ c_4 & -c_4 & -c_4 & c_4 \\ c_5 & -c_1 & c_7 & c_3 \\ c_6 & -c_2 & c_2 & -c_6 \\ c_7 & -c_5 & c_3 & -c_1 \end{bmatrix} \begin{bmatrix} c_4 & c_4 & c_4 & c_4 \\ -c_7 & -c_5 & -c_3 & -c_1 \\ -c_2 & -c_6 & c_6 & c_2 \\ c_5 & c_1 & c_7 & -c_3 \\ c_4 & -c_4 & -c_4 & c_4 \\ -c_3 & -c_7 & c_1 & -c_5 \\ -c_6 & c_2 & -c_2 & c_6 \\ c_1 & -c_3 & c_5 & -c_7 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$



Multiple fixed Coefficients exist !!!

➔ **Low Power & Area DCT architecture.**

Project HINTS: Sensitivity Differences

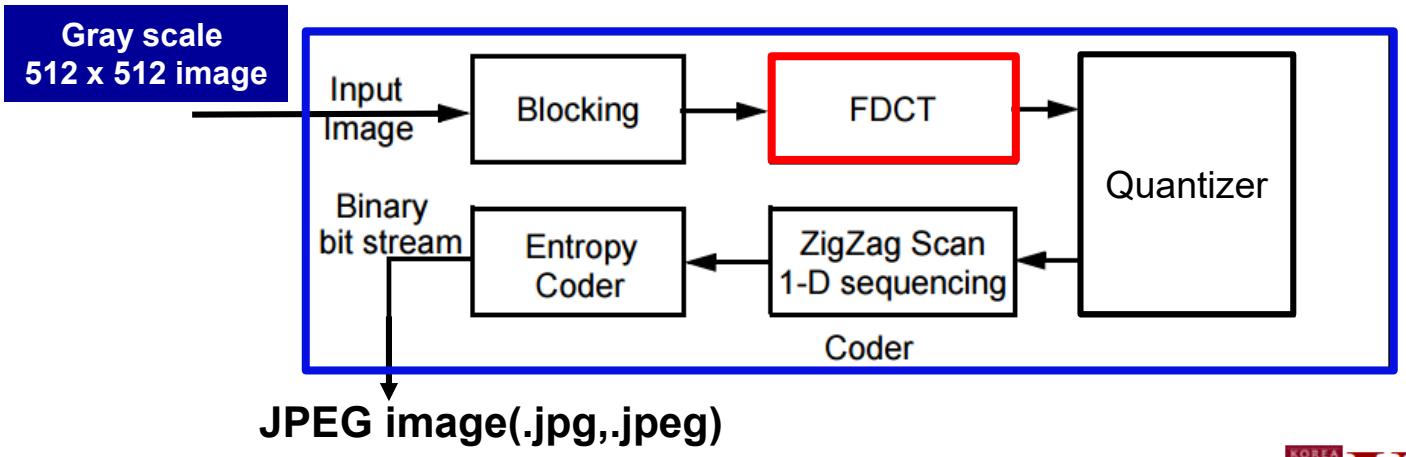


Project HINTS: Sensitivity Differences

Project 2 Process

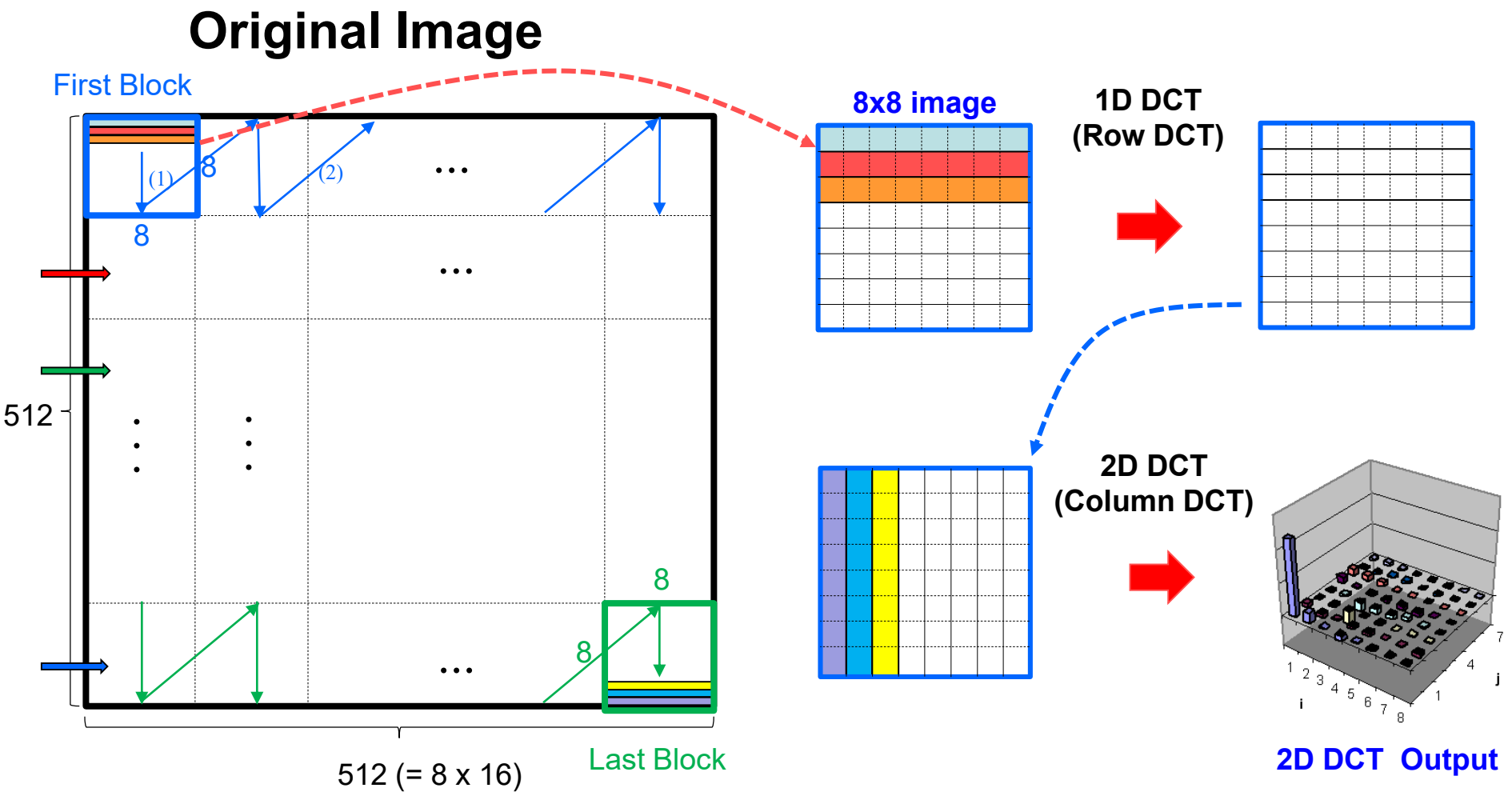
A simple JPEG compressor:

- **Cut an image up (512x512) into blocks of 8x8 pixels**
- **Run each block through an 8x8 2D-DCT**
- **DCT basis quantization is designer's choice !**
- **Internal Node Quantization is the designer's choice !**
- **Whole Simulation environment is given !**
- **Two types of MATLAB files !**

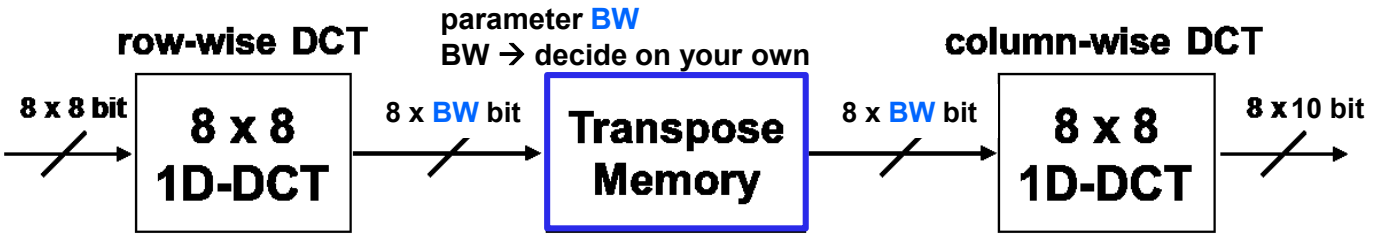


Design with Verilog
Use given MATLAB

Discrete Cosine Transform



2D-DCT Implementation



Input
(row-wise 8 pixels)

1st WRITE	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	$x_{1,5}$	$x_{1,6}$	$x_{1,7}$	$x_{1,8}$
2nd WRITE	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	$x_{2,5}$	$x_{2,6}$	$x_{2,7}$	$x_{2,8}$
3rd WRITE	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$	$x_{3,5}$	$x_{3,6}$	$x_{3,7}$	$x_{3,8}$
4th WRITE	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$	$x_{4,5}$	$x_{4,6}$	$x_{4,7}$	$x_{4,8}$
5th WRITE	$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	$x_{5,4}$	$x_{5,5}$	$x_{5,6}$	$x_{5,7}$	$x_{5,8}$
6th WRITE	$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	$x_{6,4}$	$x_{6,5}$	$x_{6,6}$	$x_{6,7}$	$x_{6,8}$
7th WRITE	$x_{7,1}$	$x_{7,2}$	$x_{7,3}$	$x_{7,4}$	$x_{7,5}$	$x_{7,6}$	$x_{7,7}$	$x_{7,8}$
8th WRITE	$x_{8,1}$	$x_{8,2}$	$x_{8,3}$	$x_{8,4}$	$x_{8,5}$	$x_{8,6}$	$x_{8,7}$	$x_{8,8}$
	1st READ	2nd READ	3rd READ	4th READ	5th READ	6th READ	7th READ	8th READ

Transpose memory Verilog module is given !

IF YOU CAN IMPROVE : BONUS !!!

Output
(column-wise 8 pixels)

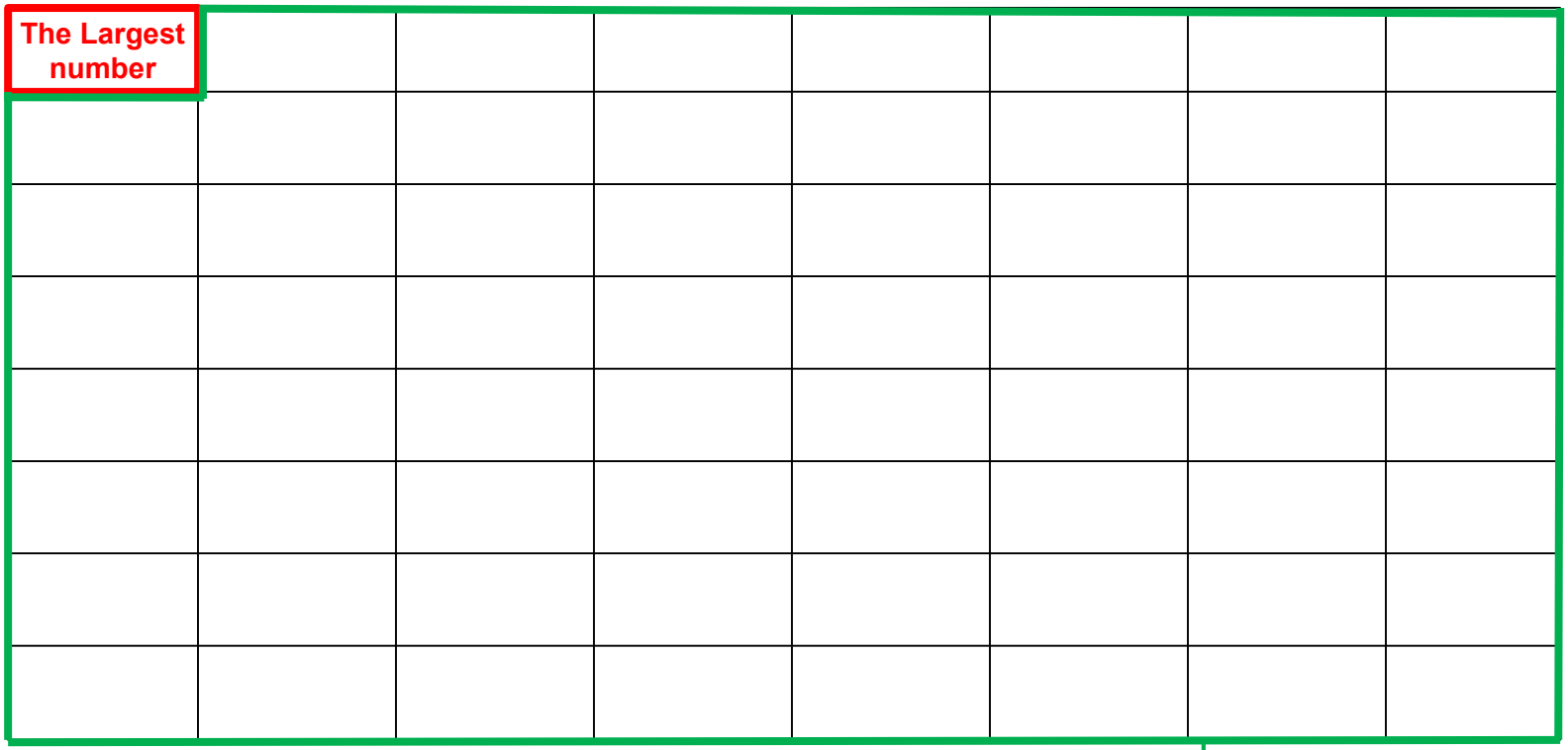
Truncation of the 2D-DCT output

How to make 10 bit output from the 2D DCT (8 x 8)

Truncate 2 LSB(Least Significant Bit)
Of the DC output

xxx10010.10xx

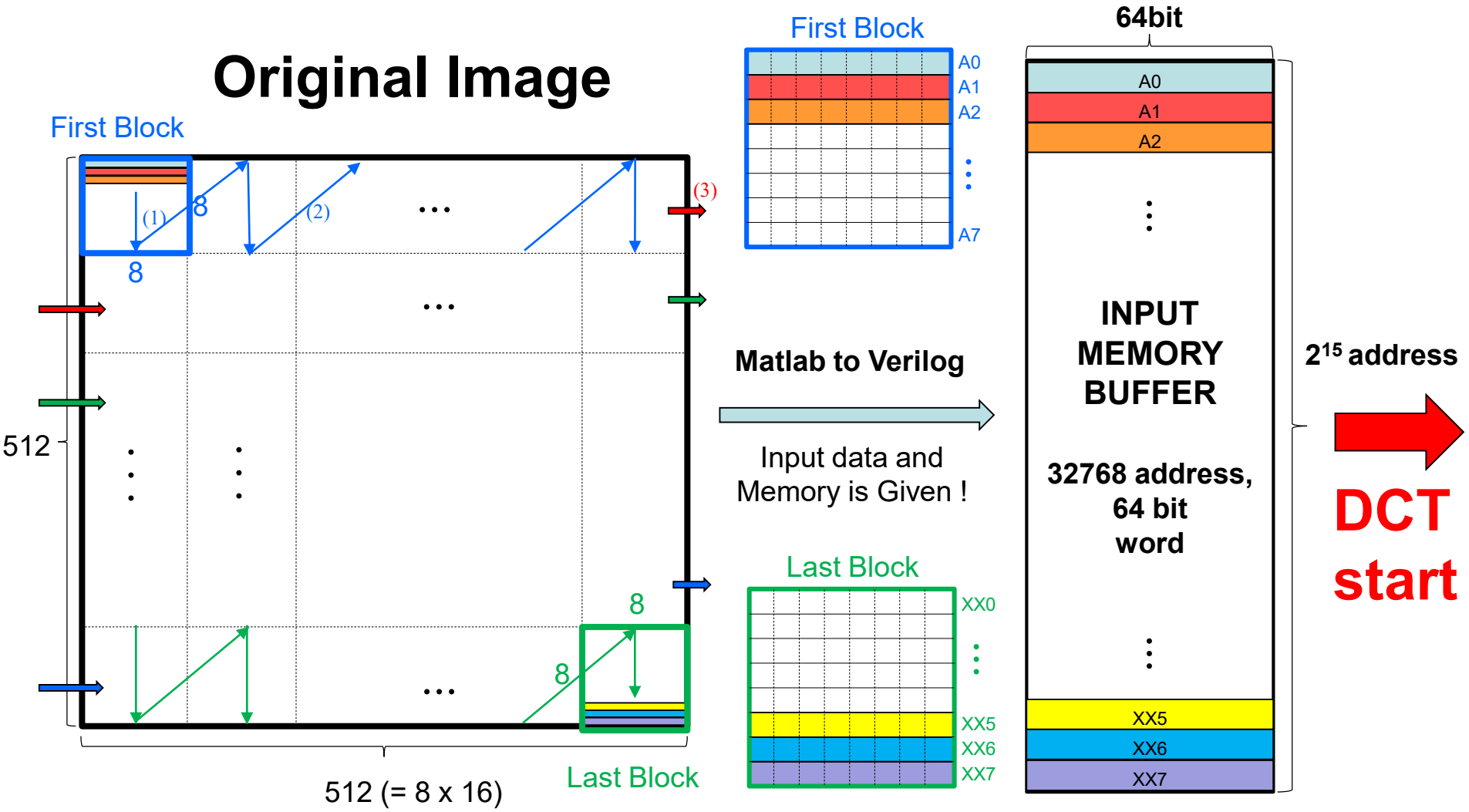
10 bit



10 bit integer output of 2D-DCT

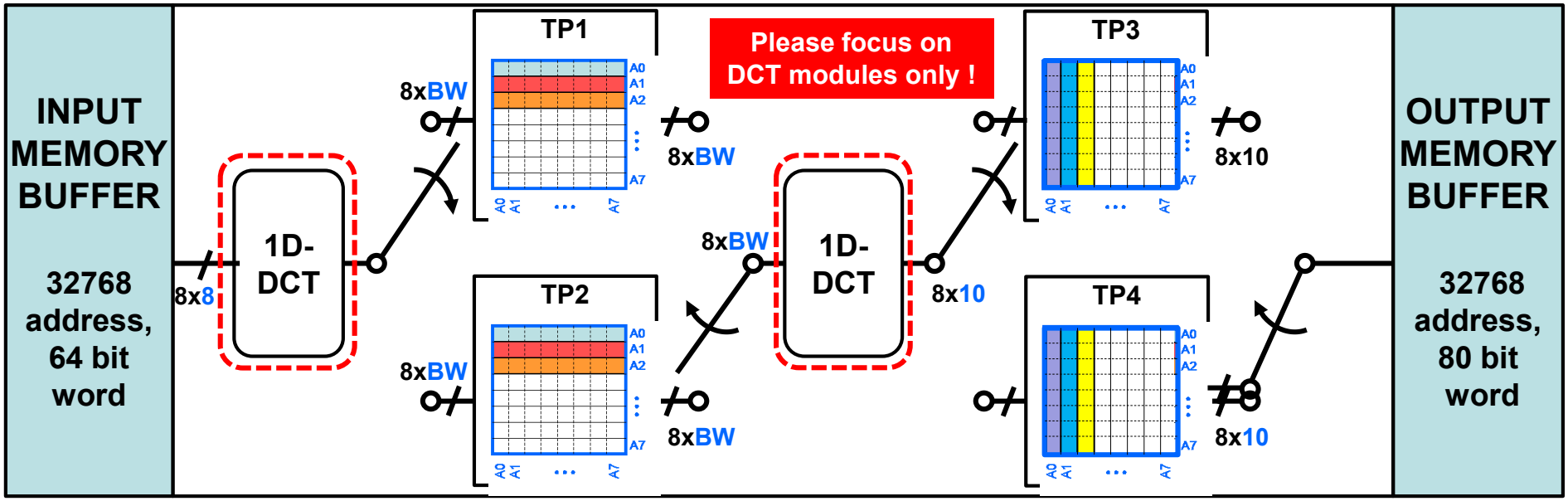
xxx10010.10xx

Input memory buffer data structure



Overall 2D DCT process

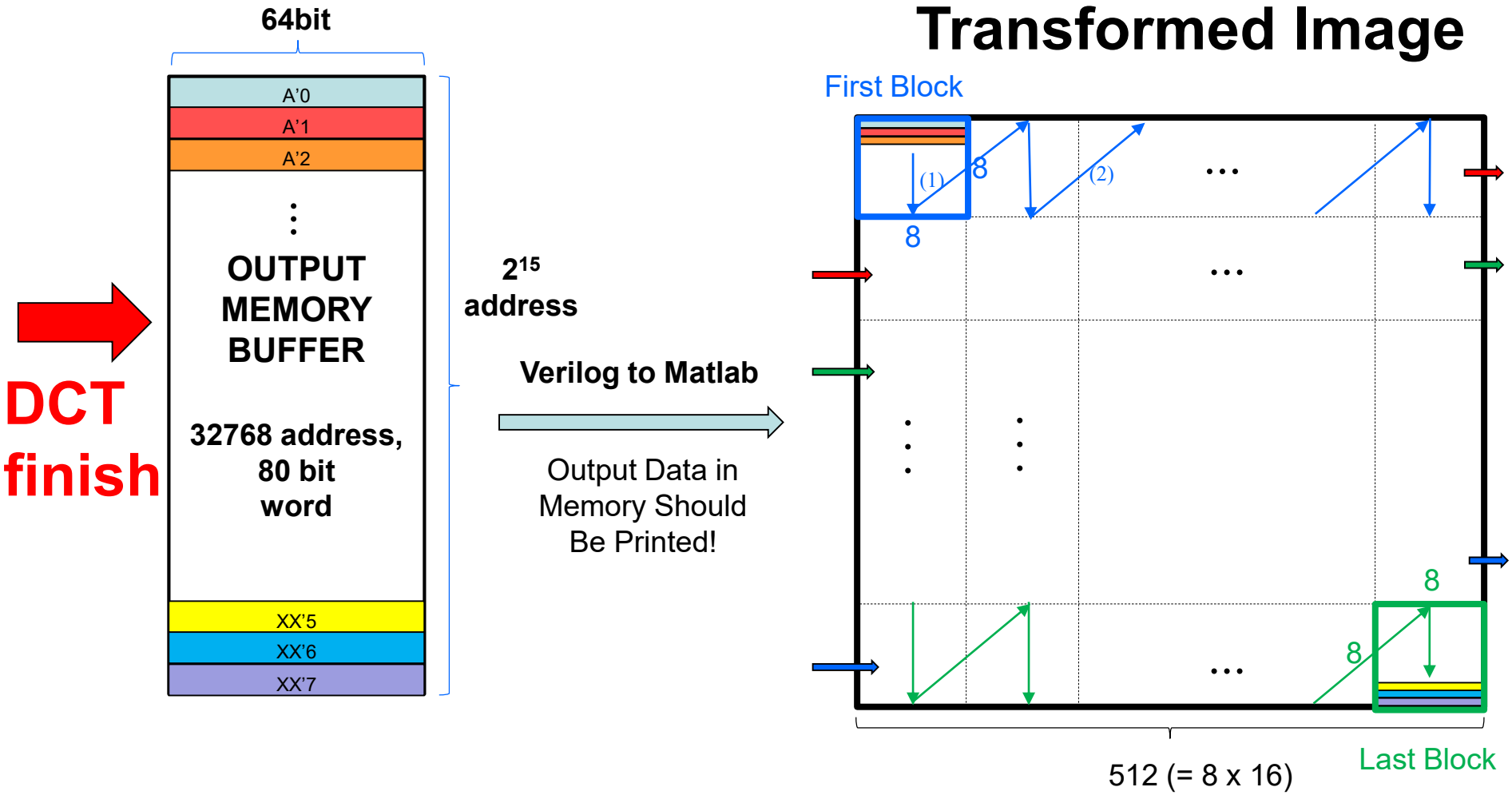
Design your 2D DCT architecture !!



1) Image data is stored in the input buffer !!
Control the input / output memory buffers !!

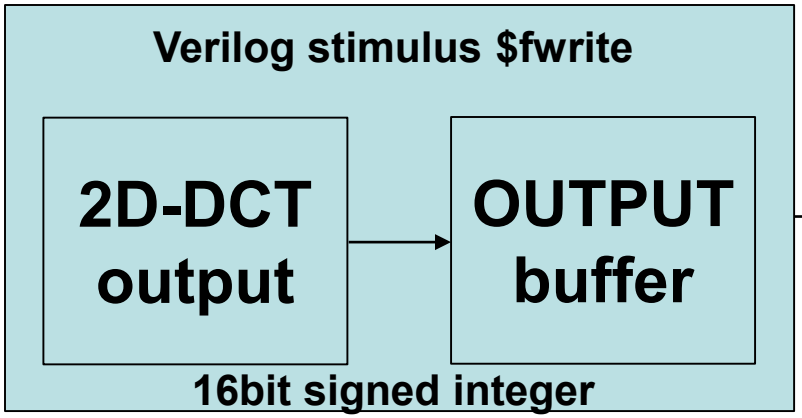
2) Transpose Memory module is given !!!
But, if you can optimize, BONUS !!

Output memory buffer data structure



Data IO between Verilog and MATLAB

- **Data format for Data IO**



txt file

MATLAB JPEG encoding (zigzag + entropy coding) + JPEG decoding

```
integer f;
integer i;

initial
begin
        f = $fopen("DCT_image_1.txt","w");

        @(negedge reset); //Wait for reset to be released
        @(posedge clk);   //Wait for fisrt clock out of reset

        for (i = 0; i<32768; i=i+1)
        begin
                write[i] <= TEST.memory.Mem[i];
                $display("DATA %x", TEST.memory.Mem[i]);
                $fwrite(f,"%x\n",   TEST.memory.Mem[i]);
        end

        #100
        $fclose(f);
        $finish;
end
```

PSNR value
tiff image file

About Project #2 Evaluation

Given Files

1. Input image vectors are stored in the input buffer (8 images, 512 x 512 pixels)
2. MATLAB codes (Data IO, JPEG encoding / decoding)
3. Stimulus files, SRAM memory files, Transpose memory
4. Synthesis Environment

Evaluation condition

Please use minimum hardware for 2D-DCT while satisfying the minimum required PSNR value (29.5 dB)

- PSNR values of all 8 images should be over 29.5dB
- Target frequency is given as 100MHz

About Project #2 Evaluation

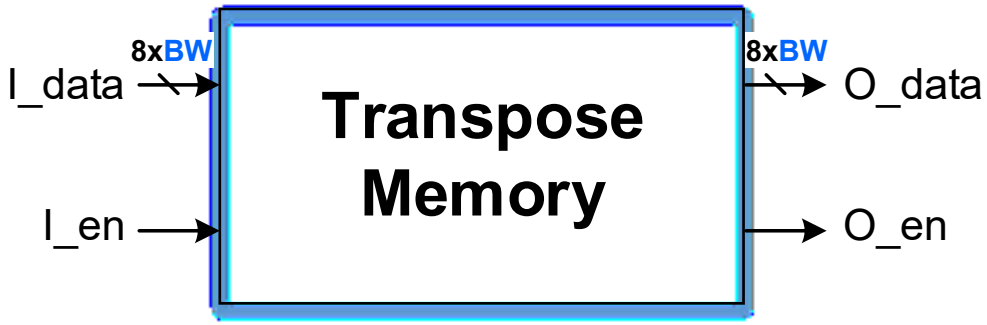
8 of tiff images are given.
each of these should satisfy the PSNR standard.

Project 2: 2D-DCT Design

- **Submit the report and Prepare for presentation**
  - Describe the 2D-DCT hardware architecture
    - Describe the memory usage of your design
  - Show timing diagrams of your 2D-DCT design
  - Show timing & area report & .v file
    - Please use report_area –hierarchy for area report
  - Check the critical path delay, Area & Power
- **Presentation day : June 19th**
  - Describe your design in detail and show the above results
- **Question – TA's and toto9900@korea.ac.kr**

2)Transpose Memory (TPmem.v)

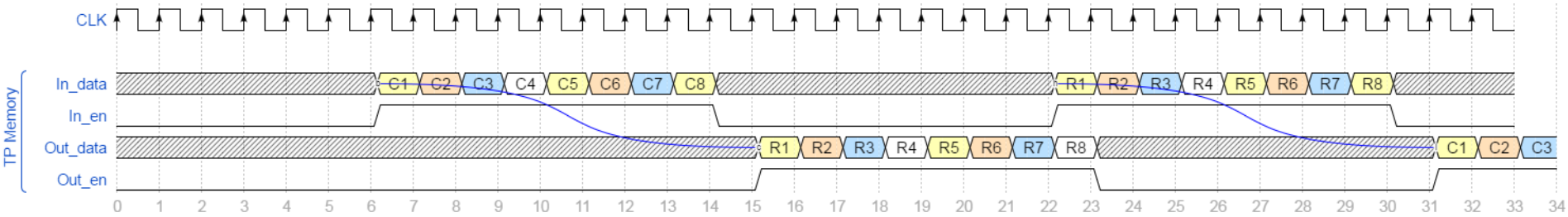
- **Transpose Memory for the 2D-DCT is given.**
  - It is designed for 8 by 8 data blocks (1 data = BW bit)
- **Bit Width**
  - **parameter BW in Verilog code**
  - **Input**
    - **Input data: 8xBW bit**
    - **Input enable : 1bit**
  - **Output**
    - **Output data: 8xBW bit**
    - **Output enable : 1bit**



- **Positive edge triggered(CLK)**
- **It is made of number of registers and MUXs**

2)Transpose Memory (TPmem.v)

- **Waveform of Transpose memory**



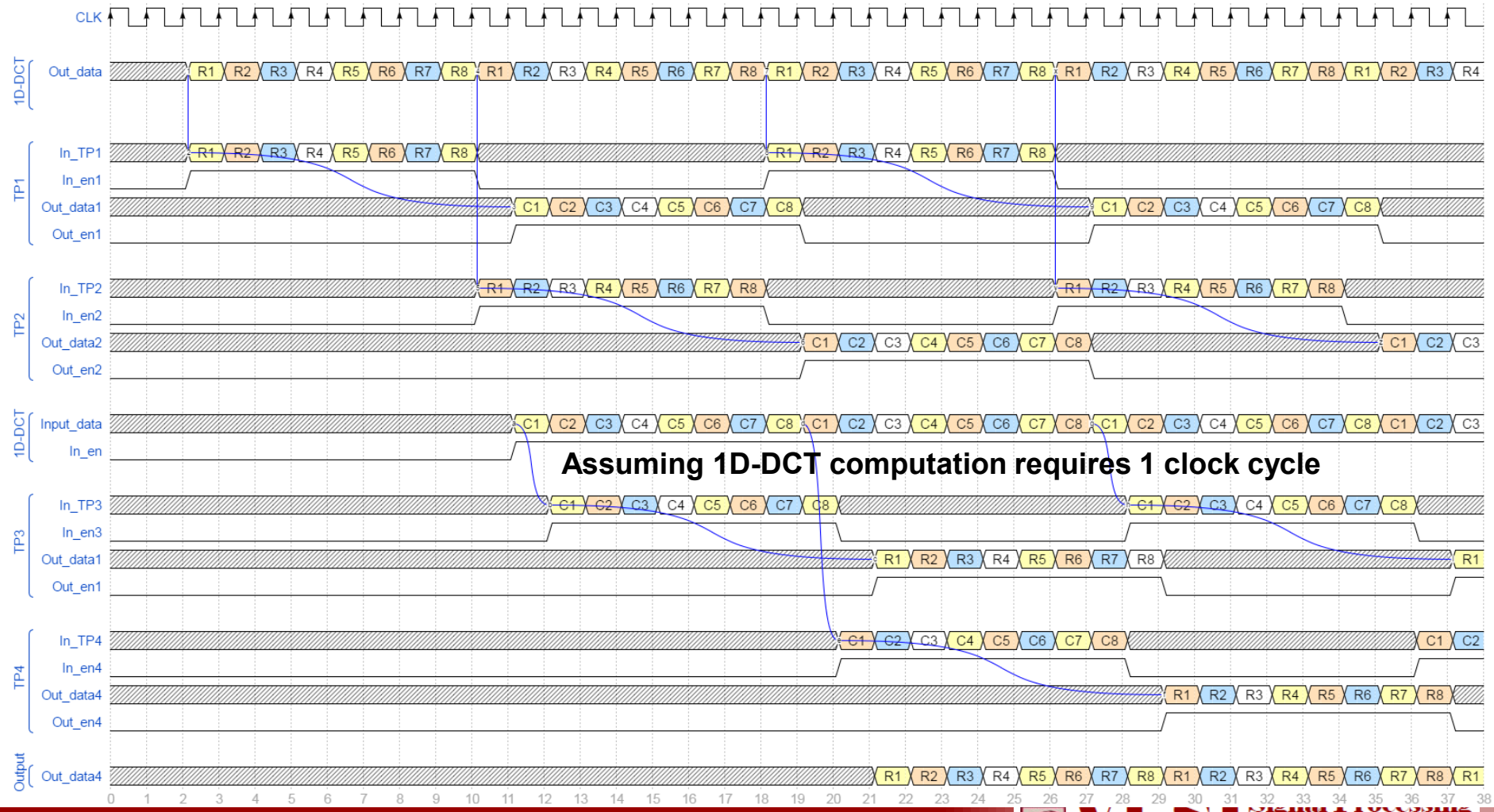
– **Row-wise data input → Column-wise data output**
Column-wise data input → Row-wise data output

– **The operations should be done in the following order**
1 pair : 8 clock cycles of write ➔ 8 clock cycles of read

– **No sequential write operations over 8 clock cycles**
→ operational error occurs

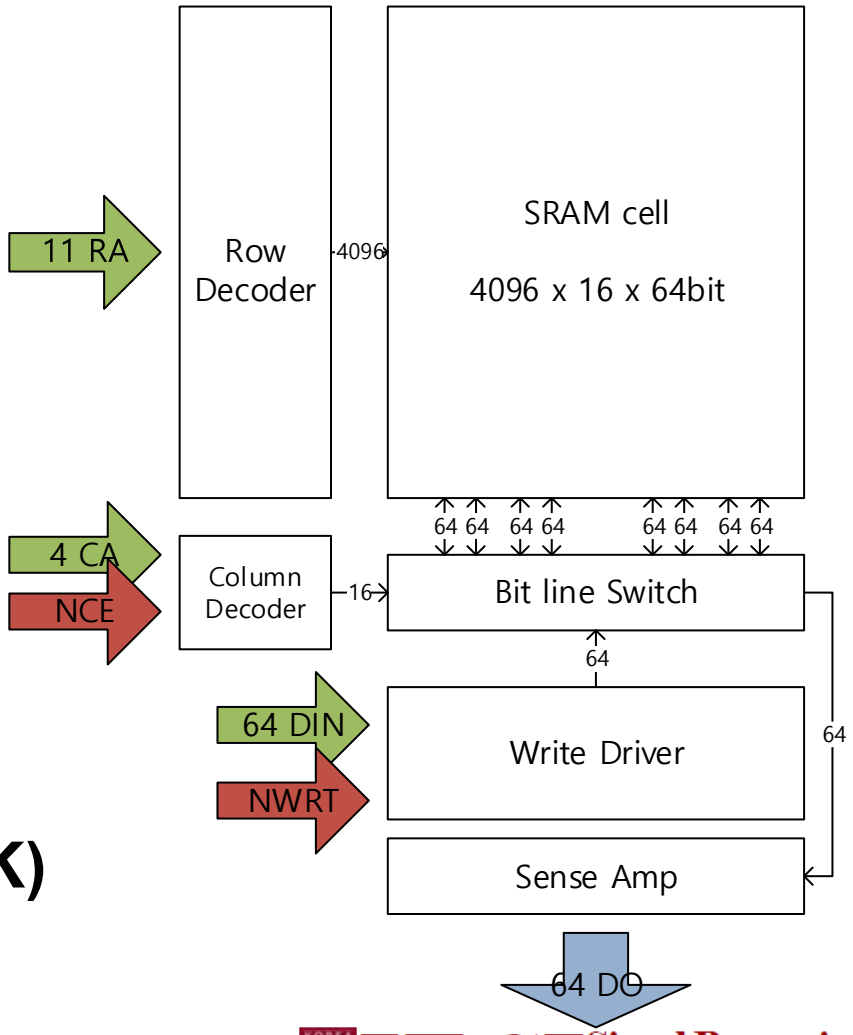
Project 2: 2D-DCT Design

Waveform of the 2D DCT architecture !!



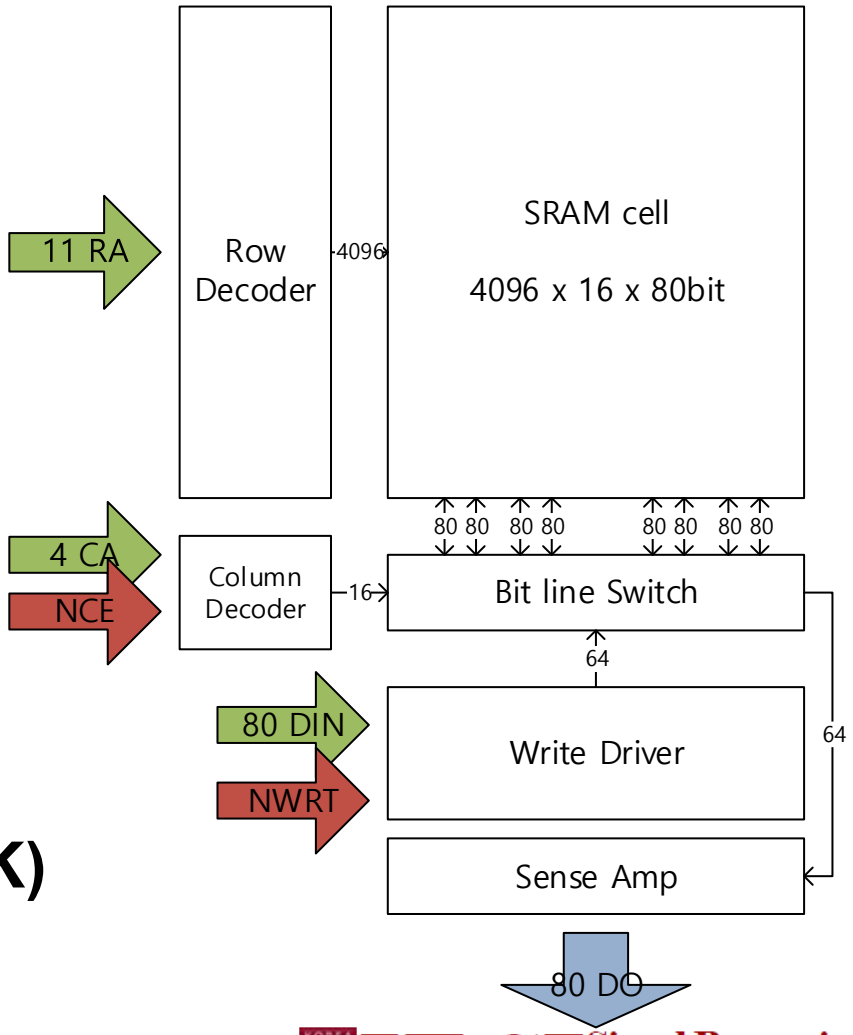
SRAM32768x64.v SRAM model

- 1 port SRAM(Read or Write)
- Bit Width
  - Input
    - Row Address : 11 bit
    - Column Address : 4 bit
    - Data In : 64 bit
  - Output
    - Data Out : 64bit
  - Control Signal
    - NWRT : 1bit (0 → Write, 1 → Read)
    - NCE : 1bit (0 → Enable, 1 → Disable)
- Positive Edge triggered(CLK)



SRAM32768x80.v SRAM model

- 1 port SRAM(Read or Write)
- Bit Width
  - Input
    - Row Address : 11 bit
    - Column Address : 4 bit
    - Data In : 80 bit
  - Output
    - Data Out : 80 bit
  - Control Signal
    - NWRT : 1bit (0 → Write, 1 → Read)
    - NCE : 1bit (0 → Enable, 1 → Disable)
- Positive Edge triggered(CLK)



Read Verilog output file using Matlab

```
% Load DCT output text file from verilog (512x512 pixel)
% Each pixel has 8bit data (0~255)
DCT_image_32768x1 = fopen(sprintf('DCT_image_%d.txt',image_number),'r');
DCT_image_64b = fscanf(DCT_image_32768x1,'%lx',[32768 1]);

x=1;
for k= 1:64
    for i= 1:64
        for j = 1 : 8
            DCT_image_temp( 8*(k-1)+j , 8*(i-1)+1 ) =  DCT_image_64b(x,1) / 2^56;
            DCT_image_temp( 8*(k-1)+j , 8*(i-1)+2 ) = (DCT_image_64b(x,1) - (DCT_image_64b(x,1)/2^56)*2^56) / 2^48;
            DCT_image_temp( 8*(k-1)+j , 8*(i-1)+3 ) = (DCT_image_64b(x,1) - (DCT_image_64b(x,1)/2^48)*2^48) / 2^40;
            DCT_image_temp( 8*(k-1)+j , 8*(i-1)+4 ) = (DCT_image_64b(x,1) - (DCT_image_64b(x,1)/2^40)*2^40) / 2^32;
            DCT_image_temp( 8*(k-1)+j , 8*(i-1)+5 ) = (DCT_image_64b(x,1) - (DCT_image_64b(x,1)/2^32)*2^32) / 2^24;
            DCT_image_temp( 8*(k-1)+j , 8*(i-1)+6 ) = (DCT_image_64b(x,1) - (DCT_image_64b(x,1)/2^24)*2^24) / 2^16;
            DCT_image_temp( 8*(k-1)+j , 8*(i-1)+7 ) = (DCT_image_64b(x,1) - (DCT_image_64b(x,1)/2^16)*2^16) / 2^8;
            DCT_image_temp( 8*(k-1)+j , 8*(i-1)+8 ) =  DCT_image_64b(x,1) - (DCT_image_64b(x,1)/2^8)*2^8;
            x = x+1;
        end
    end
end

for j = 1:512
    for i = 1:512
        DCT_image(i,j) = typecast(uint8(DCT_image_temp(i,j)),'int8');
    end
end
```

Coefficient Quantization in Matlab

< Top of the matlab test file >

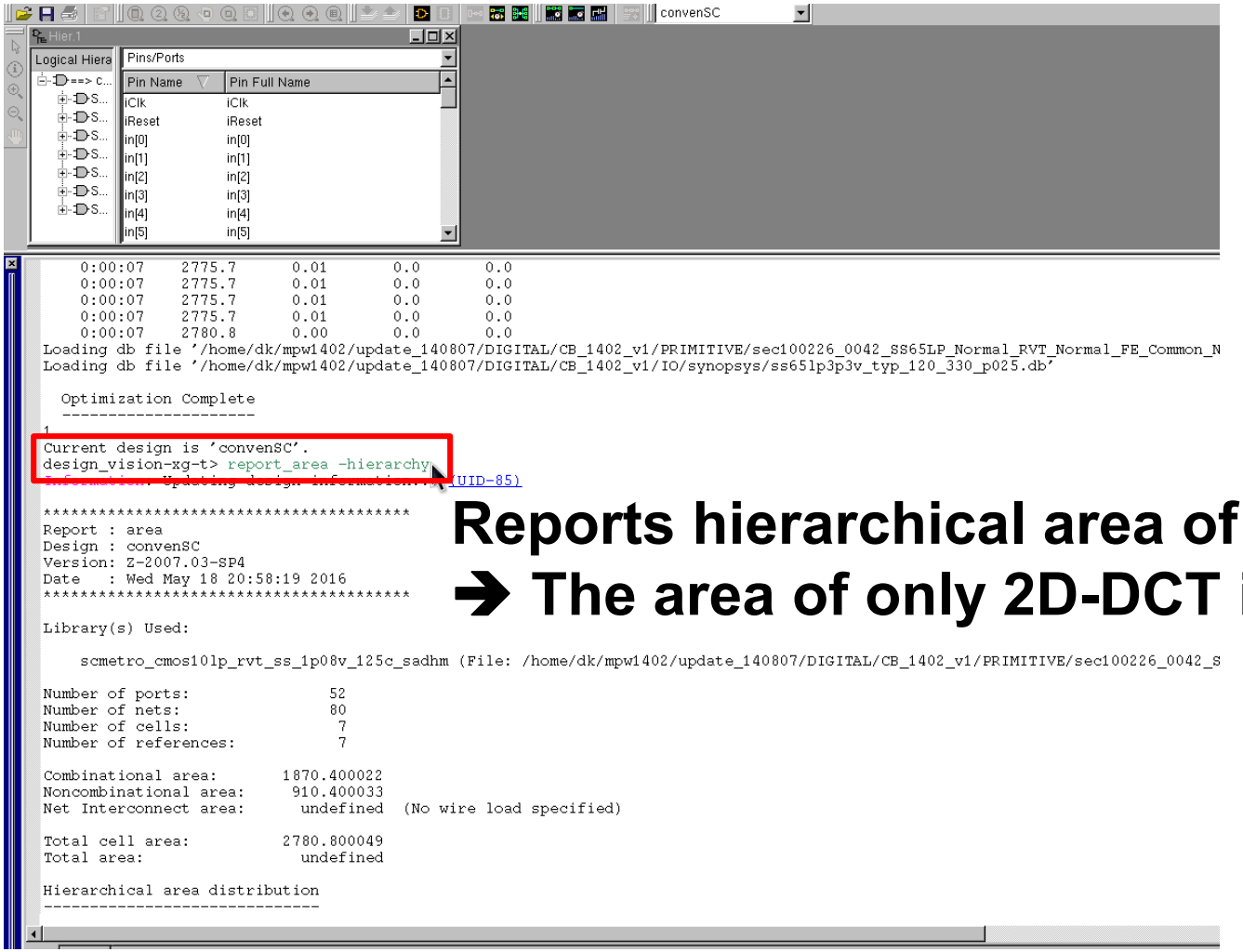
```
%------------------------Generation of DCT Bases Vector Matrix ----------------------

% The number of Bits for Quantization
% You can "adjust this number" to improve the qualities of images.

DCT_quantization_bit = 4;
T = func_DCTquant(DCT_quantization_bit);
```

You can change this number if you need

Inside of Quantization function (func_DCTquant.m)

```
function  T_quant = func_DCTquant(num_bin)
%% num_bin : The DCT Quantization bit allocation
%% Each DCT coefficients
a = 0.5*cos(pi/16);
b = 0.5*cos(2*pi/16);
c = 0.5*cos(3*pi/16);
d = 0.5*cos(4*pi/16);
e = 0.5*cos(5*pi/16);
f = 0.5*cos(6*pi/16);
g = 0.5*cos(7*pi/16);
%coefficient matrix
  T=[ d  d  d  d  d  d  d  d;
      a  c  e  g -g -e -c -a;
      b  f -f -b -b -f  f  b;
      c -g -a -e  e  a  g -c;
      d -d -d  d  d -d -d  d;
      e -a  g  c -c -g  a -e;
      f -b  b -f -f  b -b  f;
      g -e  c -a  a -c  e -g];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Change from Decimal to Binary number %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:8
   for j = 1:8
       T_bi(i,j,:) = func_Dec2Bin_mag(T(i,j), num_bin);
   end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Again Change from Binary to Decimal number %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:8
   for j = 1:8
       num_int = 0;
       T_quant(i,j) = func_Bin2Dec_mag(T_bi(i,j,:), num_int, num_bin);
   end
end
```

Report_area –hierarchy (1)



Reports hierarchical area of 2D-DCT

➔ The area of only 2D-DCT is shown

Report_area –hierarchy (2)

```
Hierarchical area distribution
------------------------------

                                 Global cell area          Local cell area
                                 ------------------  ----------------------------
Hierarchical cell                Absolute   Percent  Combi-     Noncombi-  Black
                                 Total      Total    national   national   boxes   Design
-------------------------------  ---------  -------  ---------  ---------  ------  ---------------
convenSC                         2780.7871    100.0     0.0000     0.0000  0.0000  convenSC
STAGE[0].NUM_PE[0].CONVPE         395.8402     14.2    36.1600     0.0000  0.0000  mergedPE_Q5_0
STAGE[0].NUM_PE[0].CONVPE/D0[0]     8.6400      0.3     0.0000     8.6400  0.0000  dff_100
STAGE[0].NUM_PE[0].CONVPE/D0[1]     8.6400      0.3     0.0000     8.6400  0.0000  dff_99
STAGE[0].NUM_PE[0].CONVPE/D0[2]     8.6400      0.3     0.0000     8.6400  0.0000  dff_98
STAGE[0].NUM_PE[0].CONVPE/D0[3]     8.6400      0.3     0.0000     8.6400  0.0000  dff_97
STAGE[0].NUM_PE[0].CONVPE/D0[4]     8.6400      0.3     0.0000     8.6400  0.0000  dff_96
STAGE[0].NUM_PE[0].CONVPE/D1[0]     8.6400      0.3     0.0000     8.6400  0.0000  dff_95
STAGE[0].NUM_PE[0].CONVPE/D1[1]     8.6400      0.3     0.0000     8.6400  0.0000  dff_94
STAGE[0].NUM_PE[0].CONVPE/D1[2]     8.6400      0.3     0.0000     8.6400  0.0000  dff_93
STAGE[0].NUM_PE[0].CONVPE/D1[3]     8.6400      0.3     0.0000     8.6400  0.0000  dff_92
STAGE[0].NUM_PE[0].CONVPE/D1[4]     8.6400      0.3     0.0000     8.6400  0.0000  dff_91
STAGE[0].NUM_PE[0].CONVPE/F0[0]     8.6400      0.3     0.0000     8.6400  0.0000  dff_104
STAGE[0].NUM_PE[0].CONVPE/F0[1]     8.6400      0.3     0.0000     8.6400  0.0000  dff_103
STAGE[0].NUM_PE[0].CONVPE/F0[2]     8.6400      0.3     0.0000     8.6400  0.0000  dff_102
STAGE[0].NUM_PE[0].CONVPE/F0[3]     8.6400      0.3     0.0000     8.6400  0.0000  dff_101
STAGE[0].NUM_PE[0].CONVPE/F0[4]     8.6400      0.3     0.0000     8.6400  0.0000  dff_0
STAGE[0].NUM_PE[0].CONVPE/PEMERGE  230.0801      8.3   149.1200     0.0000  0.0000  PE_merged_B5_0
STAGE[0].NUM_PE[0].CONVPE/PEMERGE/MOD_PE
                                   80.9600      2.9     1.2800     0.0000  0.0000  PE_B5_0
STAGE[0].NUM_PE[0].CONVPE/PEMERGE/MOD_PE/B0
                                    6.7200      0.2     6.7200     0.0000  0.0000  add_sub_half_0
STAGE[0].NUM_PE[0].CONVPE/PEMERGE/MOD_PE/PE_WIRING[1].B1
                                   18.2400      0.7    18.2400     0.0000  0.0000  add_sub_full_0
STAGE[0].NUM_PE[0].CONVPE/PEMERGE/MOD_PE/PE_WIRING[2].B1
                                   18.2400      0.7    18.2400     0.0000  0.0000  add_sub_full_27
STAGE[0].NUM_PE[0].CONVPE/PEMERGE/MOD_PE/PE_WIRING[3].B1
                                   18.2400      0.7    18.2400     0.0000  0.0000  add_sub_full_26
STAGE[0].NUM_PE[0].CONVPE/PEMERGE/MOD_PE/PE_WIRING[4].B1
                                   18.2400      0.7    18.2400     0.0000  0.0000  add_sub_full_25
STAGE[0].NUM_PE[1].CONVPE         388.8002     14.0    29.1200     0.0000  0.0000  mergedPE_Q5_6
STAGE[0].NUM_PE[1].CONVPE/D0[0]     8.6400      0.3     0.0000     8.6400  0.0000  dff_85
```

Hierarchy

Top module

Sub module1

Sub module2