Lab 3

MPEG-2 SIP Design and Realization

Professor Ching-Lung Su
E-mail: kevinsu@yuntech.edu.tw
NYUST/EL

Outline

3.1 Design Environment of the PXA/255 FPGA Module

3.2 IP Design and Realization of IDCT

3.3 IP Functional Verification

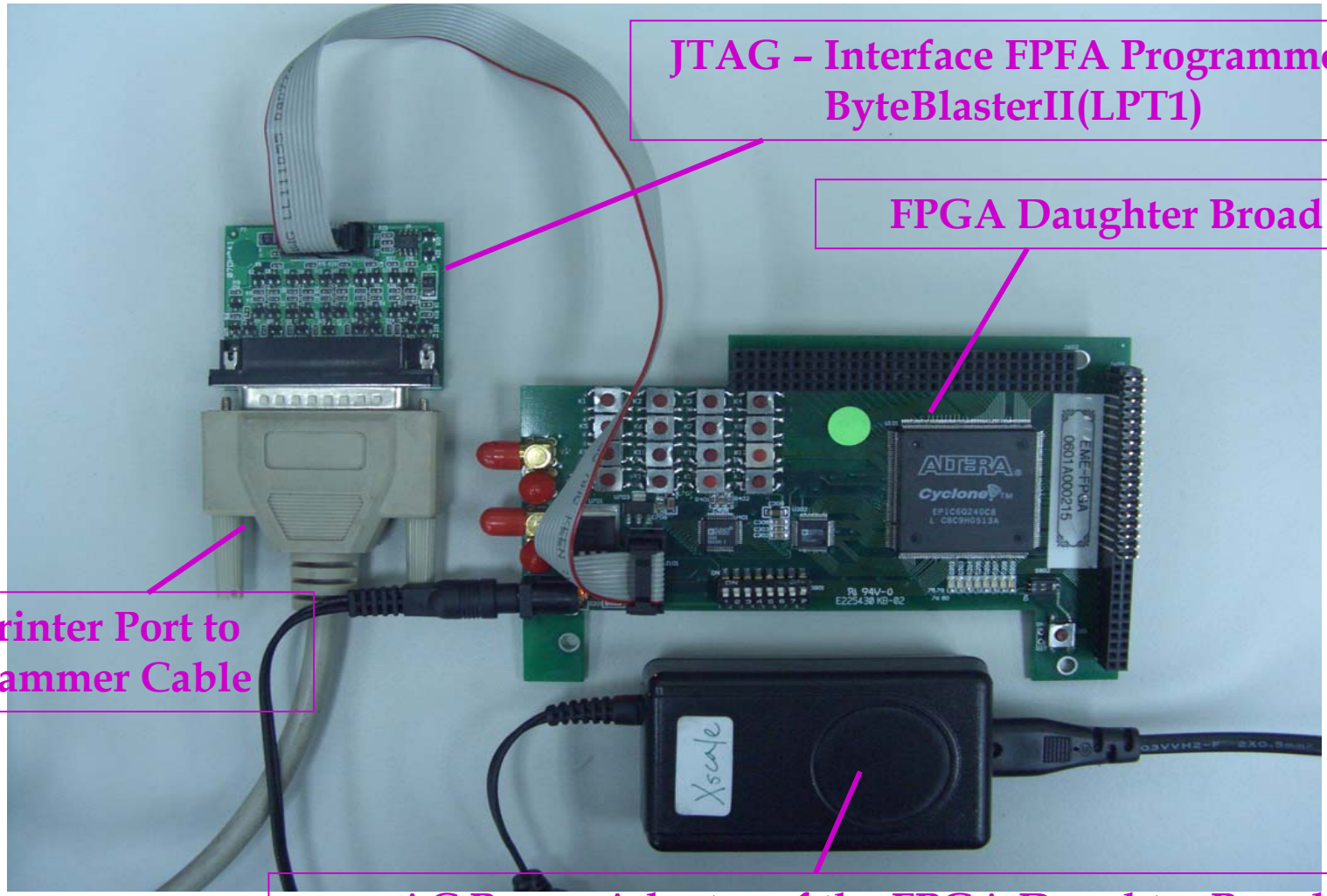
3.1 Design Environment of the FPGA Module

3.1 Design Environment of the FPGA Module

- FPGA Design Environments
- FPGA Daughter Board & Hardware Debug Tools
- Procedures the FPGA Programming

3.1 Design Environment of the FPGA Module

■ FPGA Daughter Board & Hardware Debug Tools



3.1 Design Environment of the FPGA Module

FPGA Design Environments

- FPGA IDE Tool
  - Quartus-II Version 5.1
  - + Service Pack 2
- FPGA Daughter Board
  - Altera EP1C6Q240C8 Specifications:

Feature	EP1C6
LEs	5,980
M4K RAM blocks (128 x 36 bits)	20
Total RAM bits	92,160
PLLs	2
Maximum USER I/O pins	185

3.1 Design Environment of the FPGA Module

FPGA Design Environments

- Hardware Debug Tool
  - JTAG – Interface FPGA Programmer
  - PC Printer Port to Programmer Cable
  - AC Power Adaptor of the FPGA Daughter Board

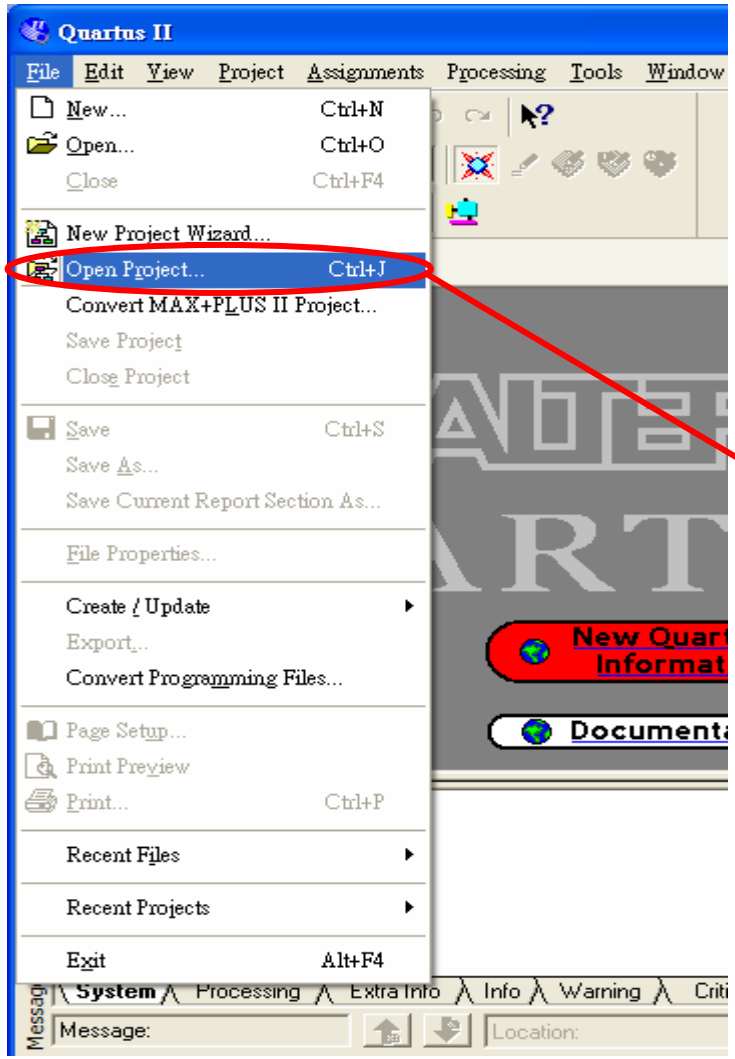
3.1 Design Environment of the FPGA Module

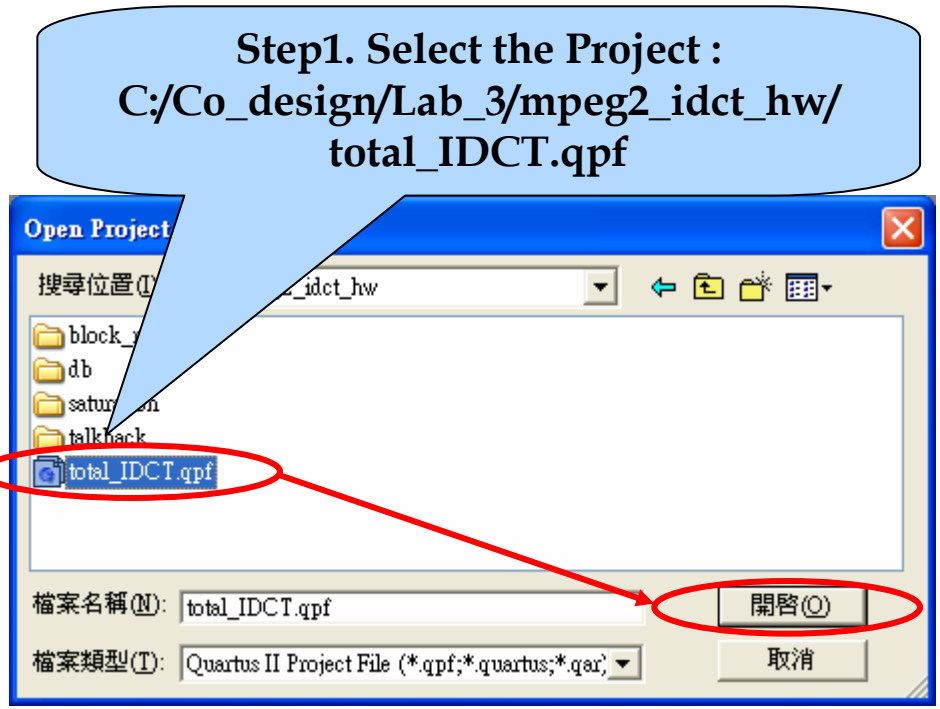
■ Procedures of the FPGA Module Programming

1. Open Project
2. FPGA Device Selection
3. Compilation
4. Programming
5. Programming Status Check

3.1 Design Environment of the FPGA Module

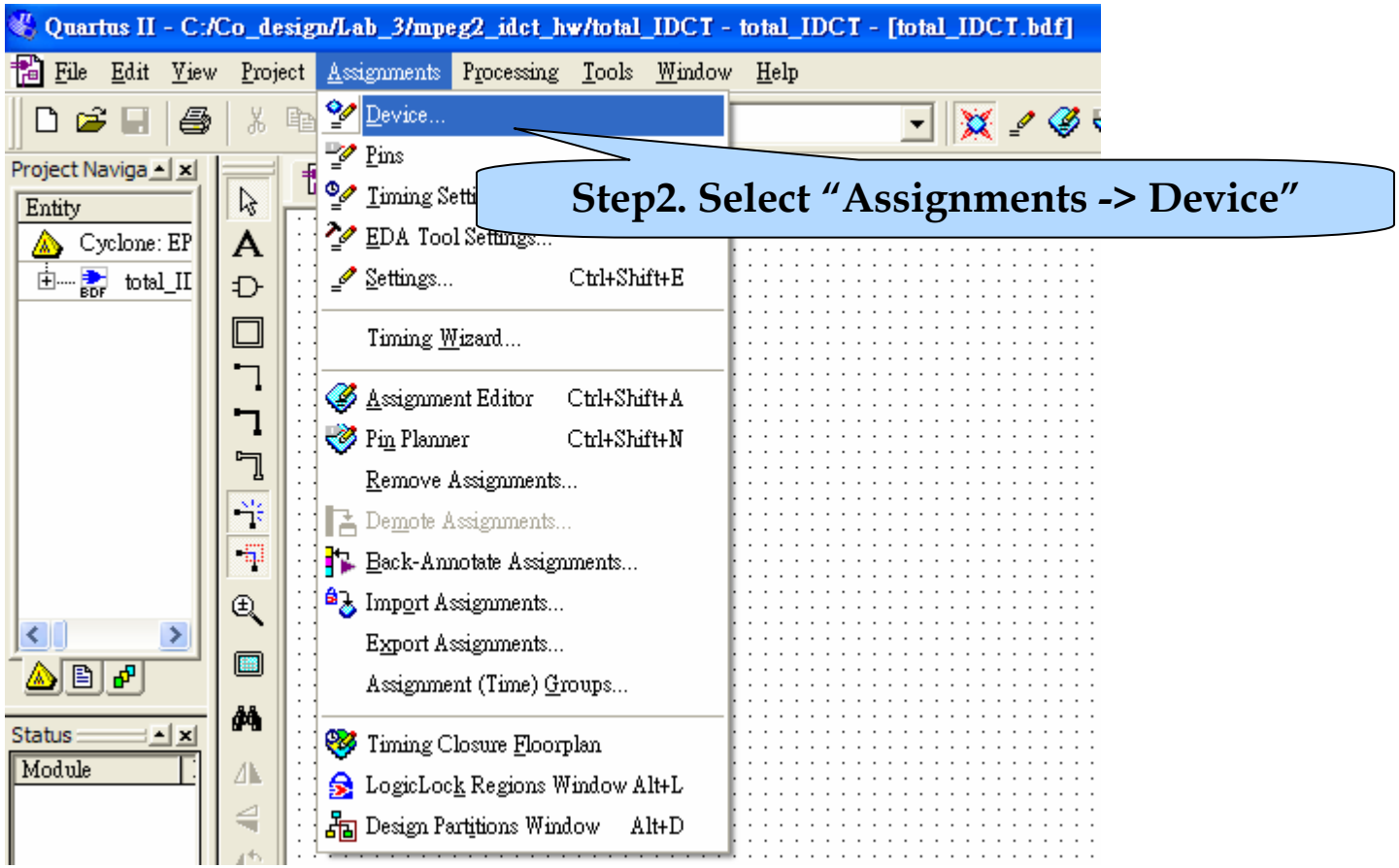
■ Procedure 1 - Open Project





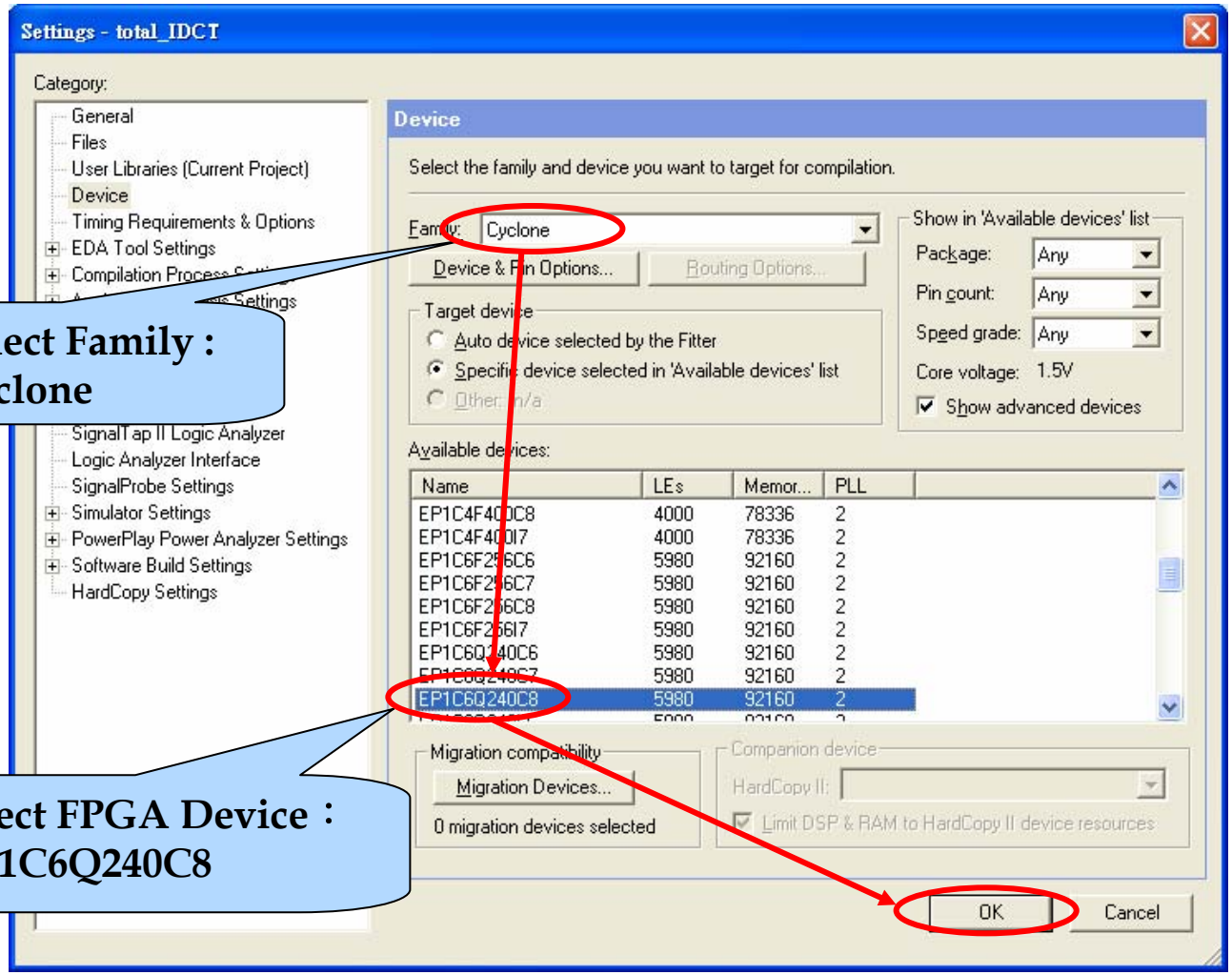
3.1 Design Environment of the FPGA Module

■ Procedure 2 - FPGA Device Selection



3.1 Design Environment of the FPGA Module

■ Procedure 2 - FPGA Device Selection

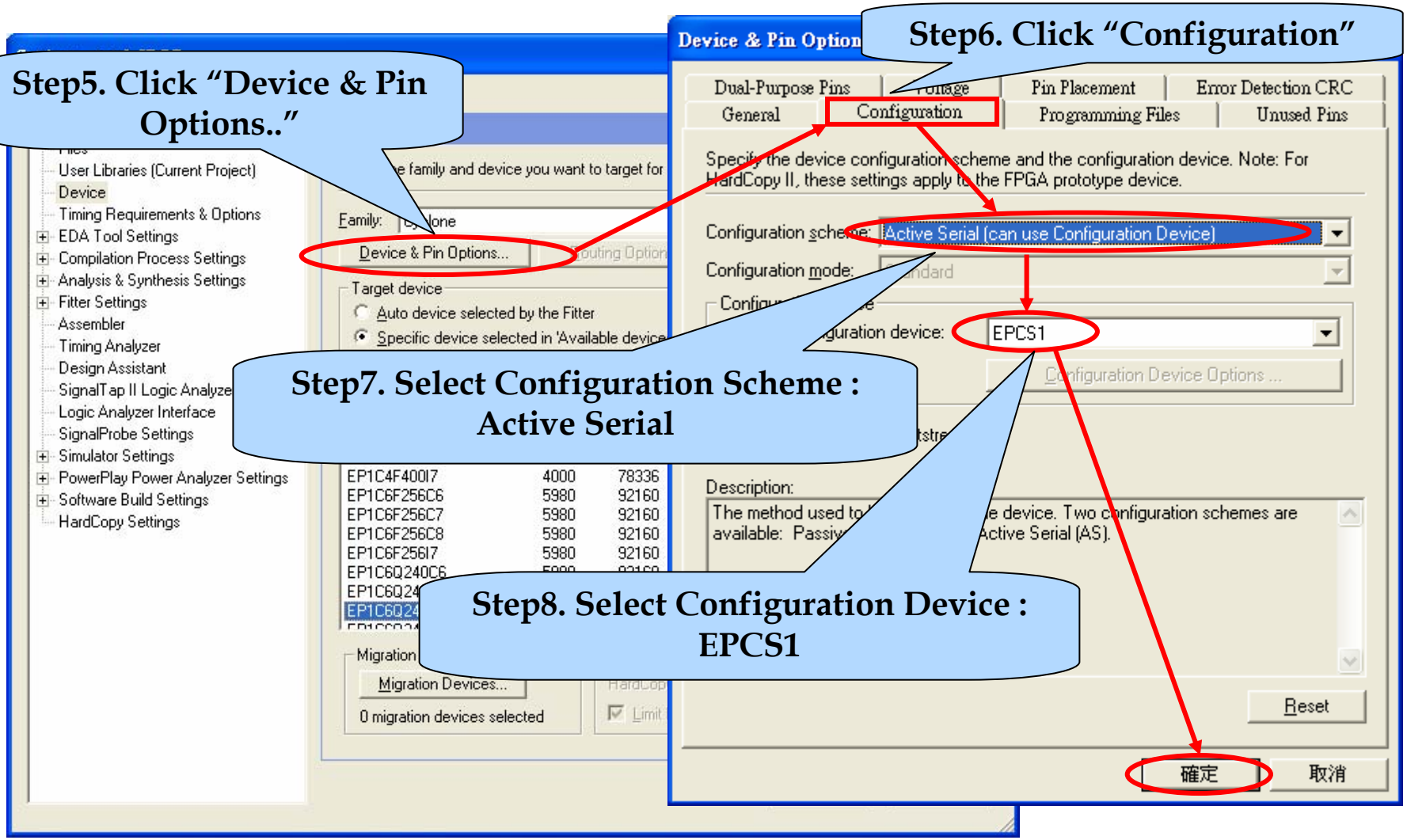


Step3. Select Family : Cyclone

Step4. Select FPGA Device：EP1C6Q240C8

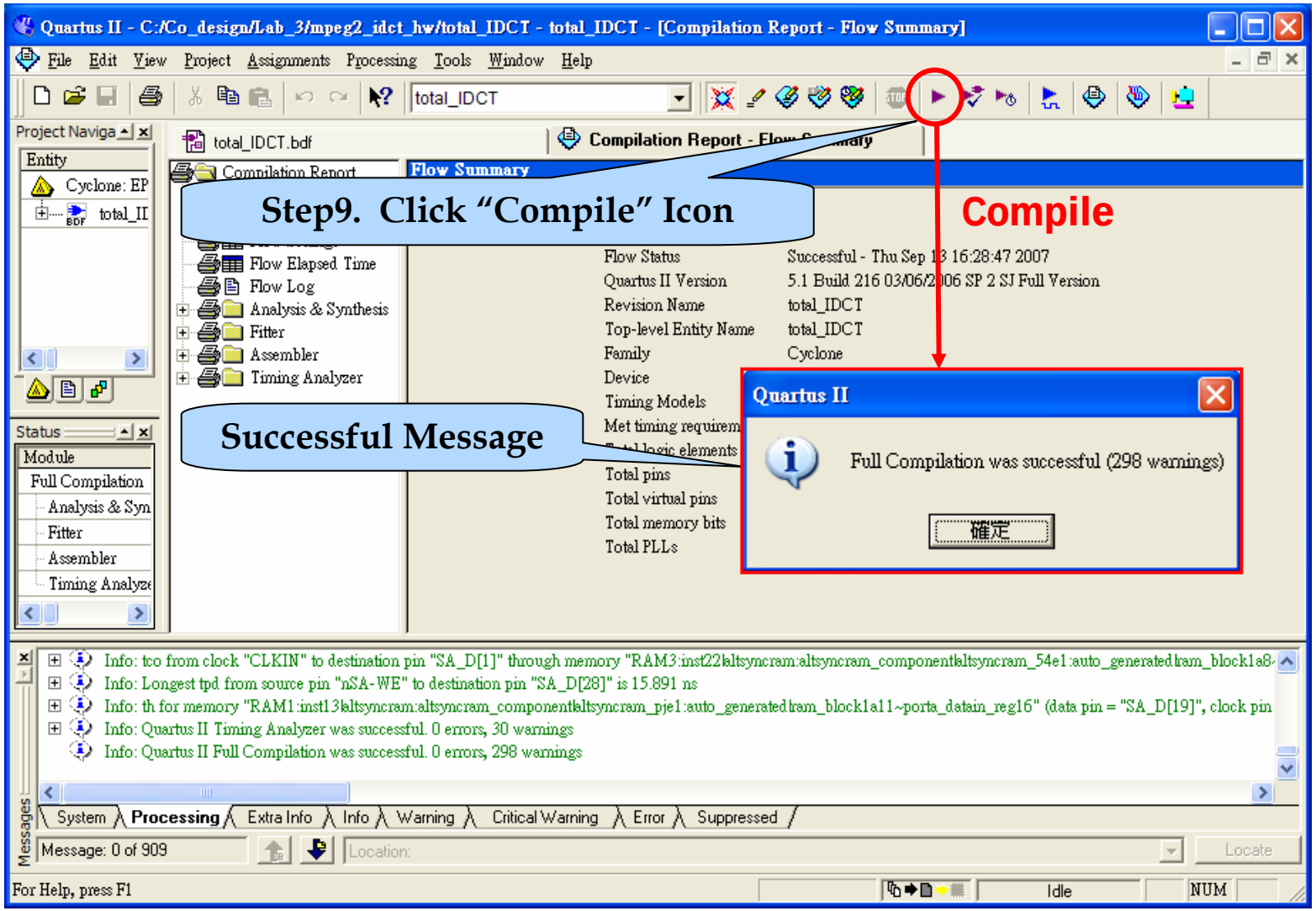
3.1 Design Environment of the FPGA Module

■ Procedure 2 - FPGA Device Selection



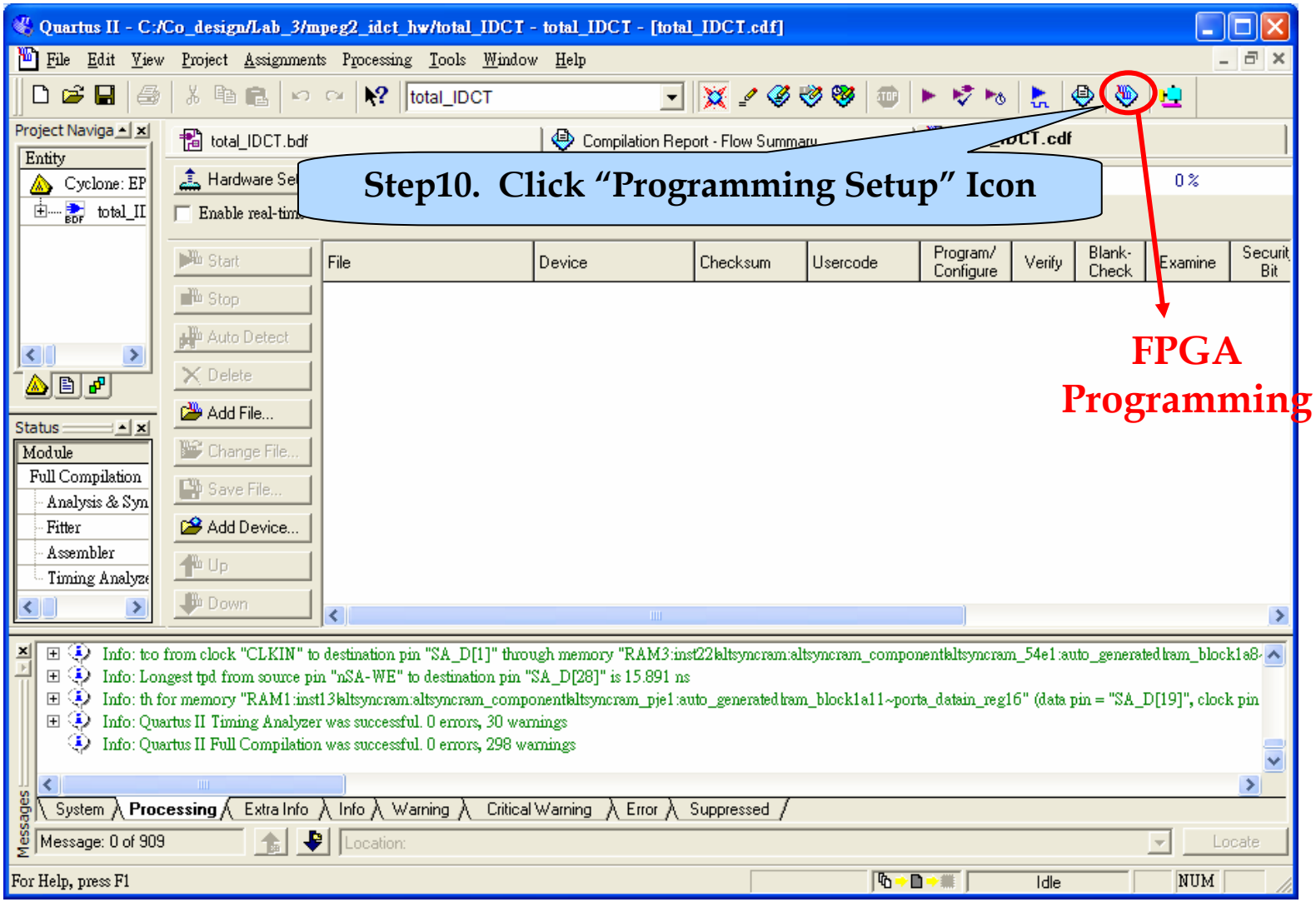
3.1 Design Environment of the FPGA Module

■ Procedure 3 - Compilation



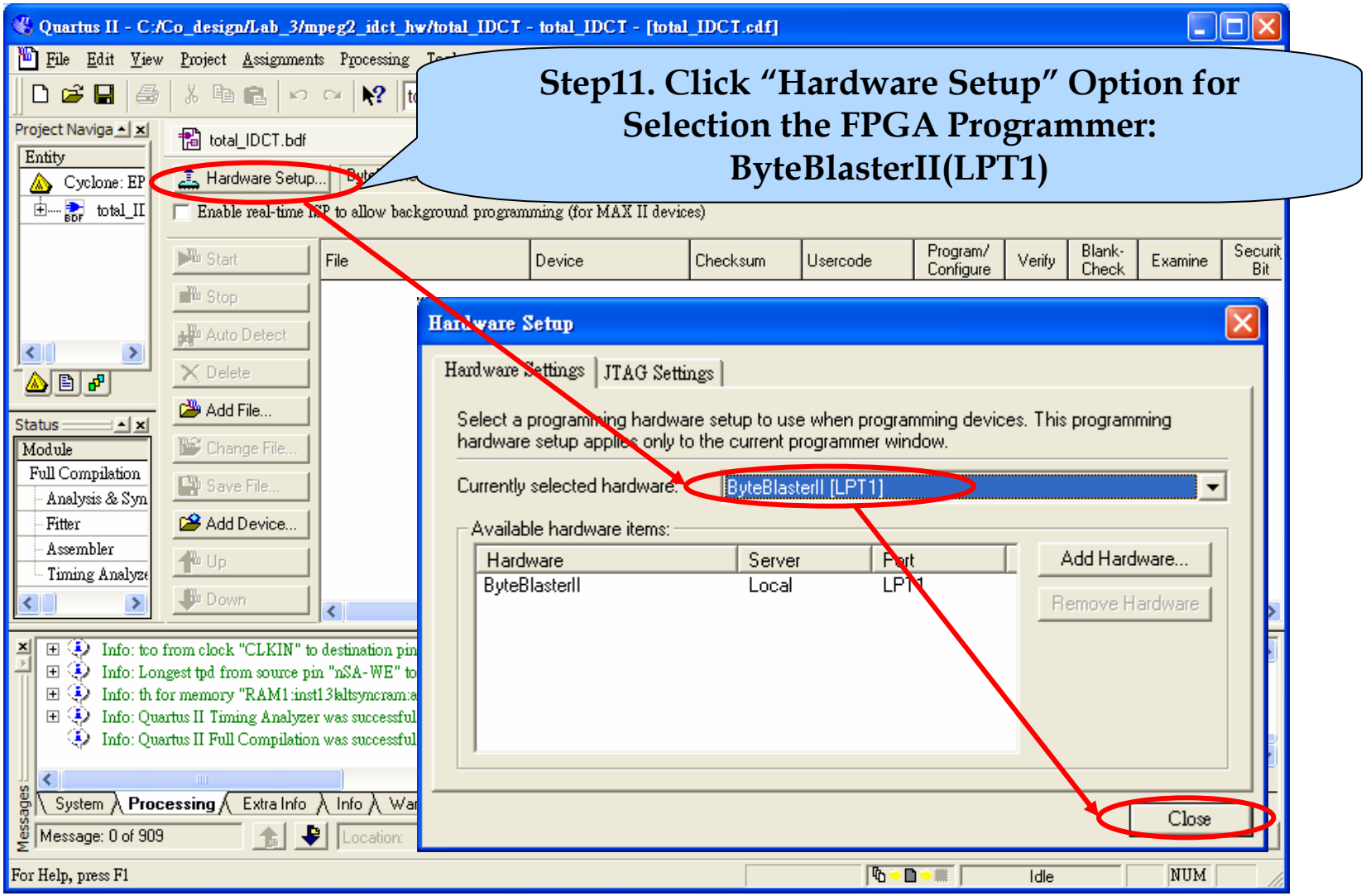
3.1 Design Environment of the FPGA Module

■ Procedure 4 - Programming



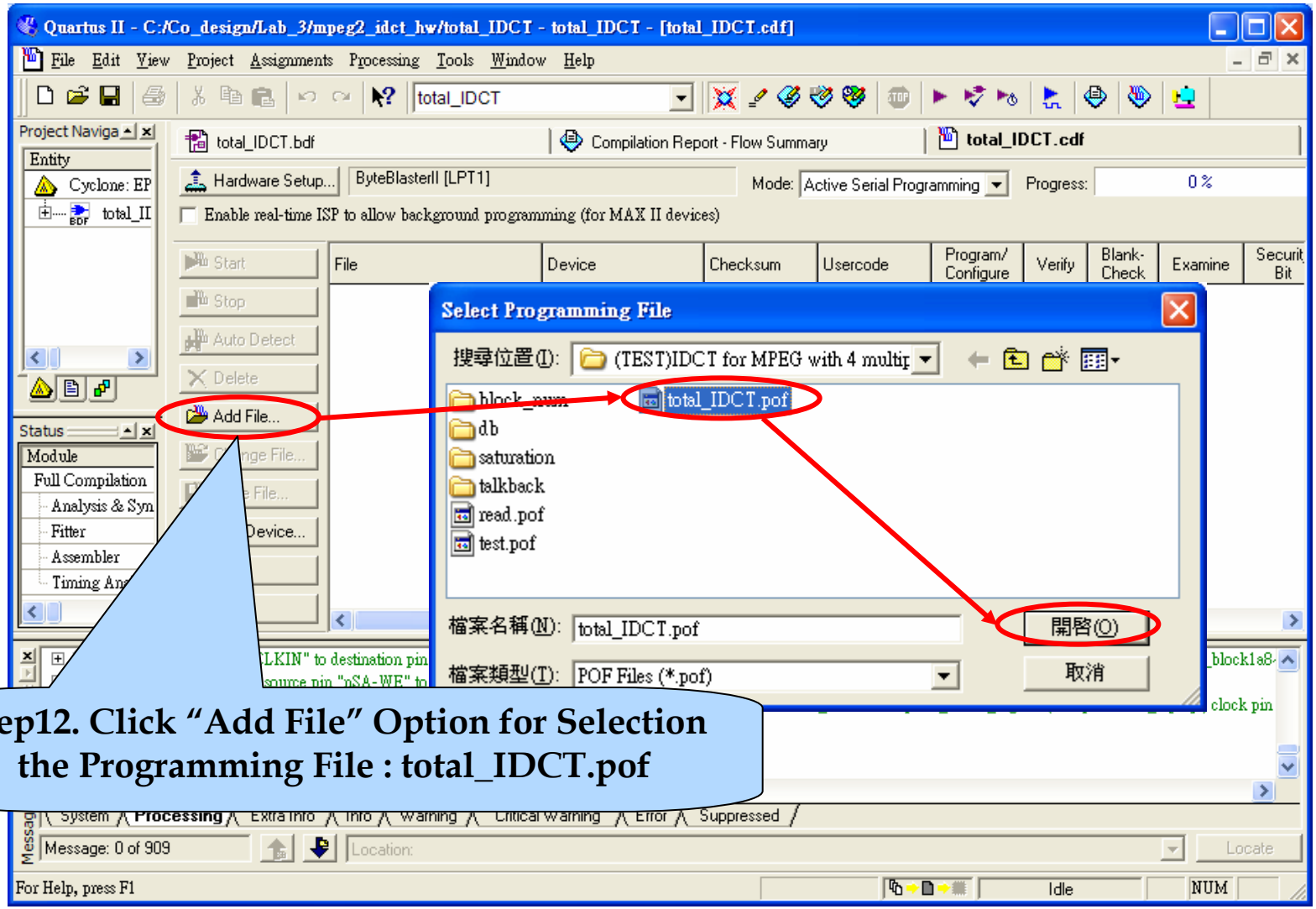
3.1 Design Environment of the FPGA Module

■ Procedure 4 - Programming



3.1 Design Environment of the FPGA Module

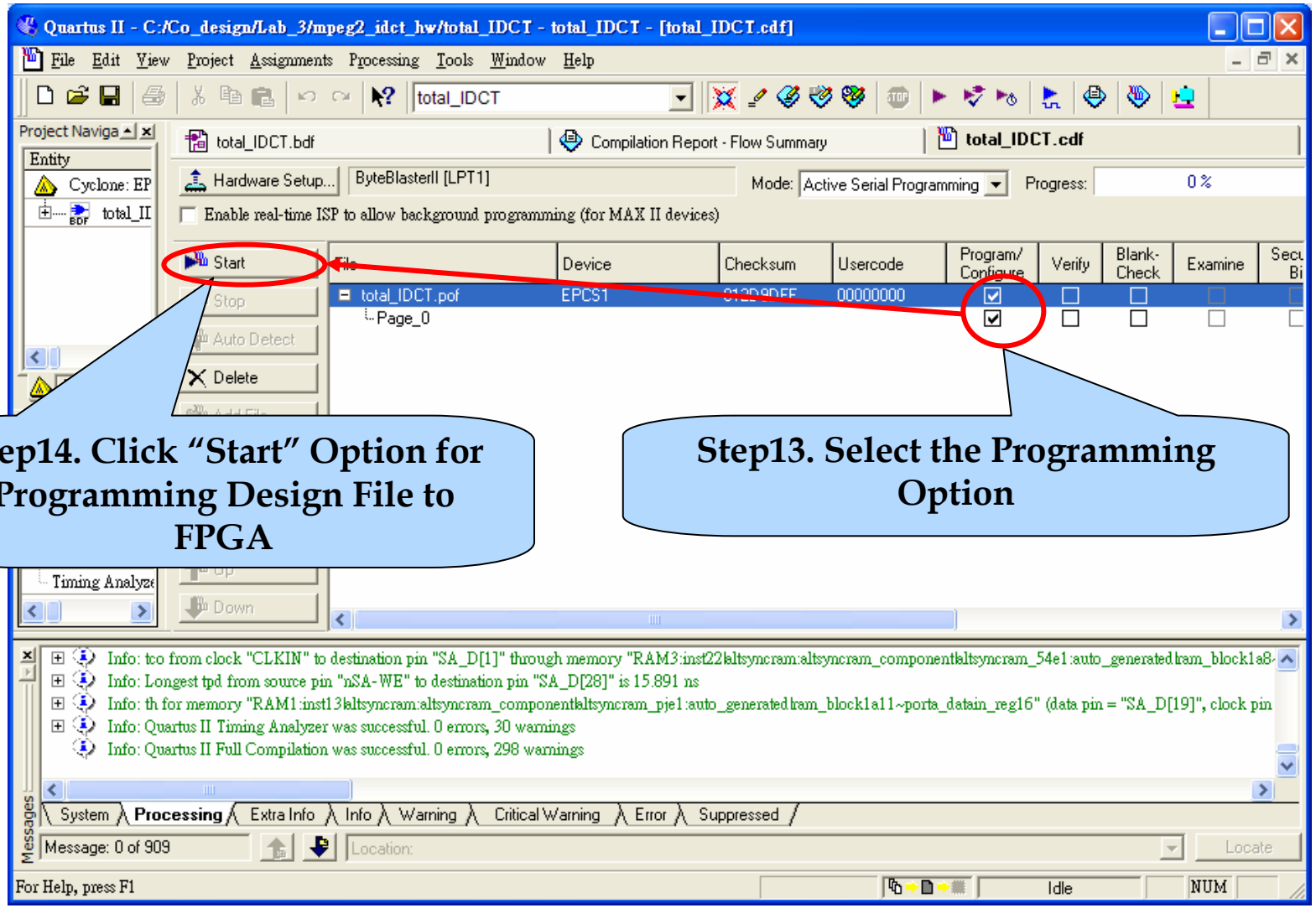
■ Procedure 4 - Programming



Step12. Click "Add File" Option for Selection the Programming File : total_IDCT.pof

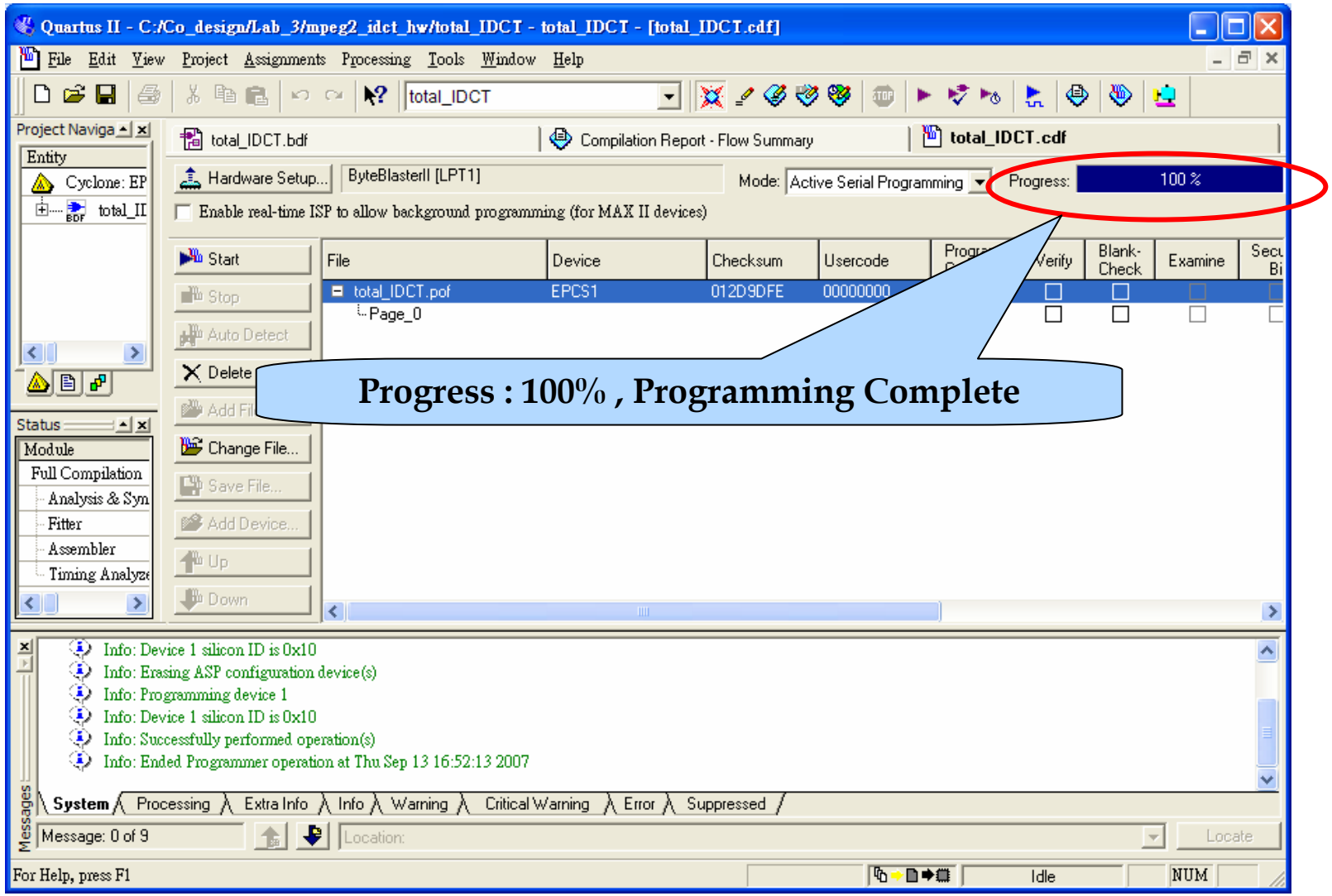
3.1 Design Environment of the FPGA Module

■ Procedure 4 - Programming



3.1 Design Environment of the FPGA Module

■ Procedure 5 – Programming Status Check



3.2 IP Design and Realization of IDCT

3.1 Design Environment of the PXA/255 FPGA Module

3.2 IP Design and Realization of IDCT

3.3 IP Functional Verification

3.2 IP Design and Realization of IDCT

- Introduction to IDCT
- IDCT IP Design
- IDCT IP Realization

3.2 IP Design and Realization of IDCT

■ Introduction to IDCT

◆ **2D Inverse Discrete Cosine Transform (IDCT) of an *NxN* Sample Block**

$$Y = A^T X A$$

Where X is a matrix of samples, Y is a matrix of coefficients and A is an *NxN* transform matrix.

$$A_{ij} = C_i \cos\frac{(2j+1)i\pi}{2N} \quad where\, C_i = \sqrt{\frac{1}{N}}\,(i=0),\ C_i = \sqrt{\frac{2}{N}}\,(i>0)$$

$$Y_{ij} = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} C_x C_y X_{xy} \cos\frac{(2j+1)y\pi}{2N}\cos\frac{(2i+1)x\pi}{2N}$$

i, j, x, y ：Matrix coordinate (range 0 ~ N-1)

3.2 IP Design and Realization of IDCT

Introduction to IDCT

- The operation complexity of 2D IDCT is very high, 2D IDCT of one *NxN*, there is the operation amount of N^4 .

  (Total operation：64 multiplication and 64 addition)

- For the amount of operation decreased, so has proposed the fast algorithms of IDCT.
  - IDCT is the orthogonal transform, we can dismantle two-dimensional IDCT into 2 one-dimensional IDCTs.

    (Total operation：32 multiplication and 32 addition)
    - First time, do the every column
    - Second time, do the every row
  - The operation complexity is reduced from N^4 to $2N^3$

3.2 IP Design and Realization of IDCT

■ 2D IDCT Design

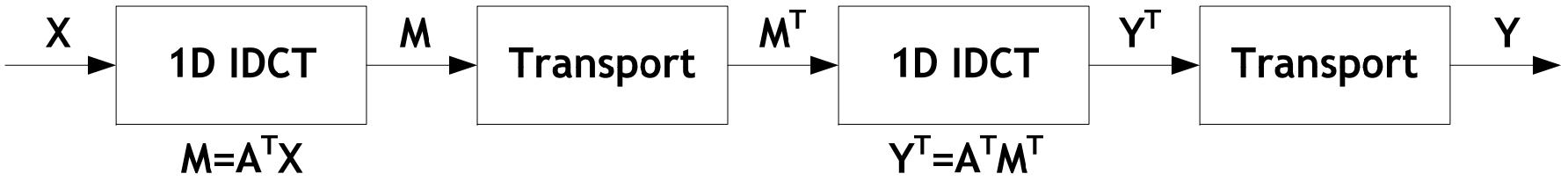
- **2D IDCT**

$$Y = A^T X A$$

- **Can be separated into two 1D-IDCT with Transpose Operations**

$$LET \quad A^T \mathbf{X} = \mathbf{M}$$

$$Then \quad Y = MA, \quad Y^T = (MA)^T, \quad Y^T = A^T M^T$$



3.2 IP Design and Realization of IDCT

8x8 2D IDCT Design Example

- **Block Size : 8x8, N=8**

$$M = A^T X,\ M(i,j) = c(j)\sum_{k=0}^{7} x(i,k)\cos[\frac{(2k+1)j\pi}{16}]$$

$$Y = MA,\quad Y(i,j) = c(i)\sum_{k=0}^{7} M(v,j)\cos[\frac{(2v+1)i\pi}{16}]$$

$$\begin{cases} C(k) = \sqrt{\frac{1}{8}}\ (k=0) \\ C(k) = \frac{1}{2}\ (k>0) \end{cases}$$

3.2 IP Design and Realization of IDCT

8x8 2D IDCT Design Example

- IDCT Coefficients
  - 7 values of the IDCT coefficients)

Coefficient	Real Value	×256 (Scale to 8-bits Int)	Variables
$\sqrt{1/4}\cos(1\pi/16)$	0.490393	125	a
$\sqrt{1/4}\cos(2\pi/16)$	0.461940	118	b
$\sqrt{1/4}\cos(3\pi/16)$	0.415735	106	c
$\sqrt{1/4}\cos(4\pi/16)$	0.353553	90	d
$\sqrt{1/4}\cos(5\pi/16)$	0.277785	71	e
$\sqrt{1/4}\cos(6\pi/16)$	0.191342	48	f
$\sqrt{1/4}\cos(7\pi/16)$	0.097545	24	g

3.2 IP Design and Realization of IDCT

■ 8x8 2D IDCT Design Example

◆ **Coefficients Matrix of 8x8 2D IDCT**

$$A^T = \begin{bmatrix} d & a & b & c & d & e & f & g \\ d & c & f & -g & -d & -a & -b & -e \\ d & e & -f & -a & -d & g & b & c \\ d & g & -b & -e & d & c & -f & -a \\ d & -g & -b & e & d & -c & -f & a \\ d & -e & -f & a & -d & -g & b & -c \\ d & -c & f & g & -d & a & -b & e \\ d & -a & b & -c & d & -e & f & -g \end{bmatrix}$$

3.2 IP Design and Realization of IDCT

■ 8x8 2D IDCT Design Example

◆ **2D 8x8 Coefficients Matrix**

$$\begin{bmatrix} Y0 \\ Y1 \\ Y2 \\ Y3 \\ Y4 \\ Y5 \\ Y6 \\ Y7 \end{bmatrix} = \begin{bmatrix} d & a & b & c & d & e & f & g \\ d & c & f & -g & -d & -a & -b & -e \\ d & e & -f & -a & -d & g & b & c \\ d & g & -b & -e & d & c & -f & -a \\ d & -g & -b & e & d & -c & -f & a \\ d & -e & -f & a & -d & -g & b & -c \\ d & -c & f & g & -d & a & -b & e \\ d & -a & b & -c & d & -e & f & -g \end{bmatrix} \begin{bmatrix} X0 \\ X1 \\ X2 \\ X3 \\ X4 \\ X5 \\ X6 \\ X7 \end{bmatrix}$$

$$Y = A^T X$$

Total Operations：64 multiplications + 64 additions

3.2 IP Design and Realization of IDCT

■ 8x8 2D IDCT Design Example

◆ **Outputs of 1D - IDCT：**

$$Y0 = dX0 + aX1 + bX2 + cX3 + dX4 + eX5 + fX6 + gX7$$
$$Y1 = dX0 + cX1 + fX2 - gX3 - dX4 - aX5 - bX6 - eX7$$
$$Y2 = dX0 + eX1 - fX2 - aX3 - dX4 + gX5 + bX6 + cX7$$
$$Y3 = dX0 + gX1 - bX2 - eX3 + dX4 + cX5 - fX6 - aX7$$
$$Y4 = dX0 - gX1 - bX2 - eX3 + dX4 - cX5 - fX6 + aX7$$
$$Y5 = dX0 - eX1 - fX2 + aX3 - dX4 - gX5 + bX6 - cX7$$
$$Y6 = dX0 - cX1 + fX2 + gX3 - dX4 + aX5 - bX6 + eX7$$
$$Y7 = dX0 - aX1 + bX2 - cX3 + dX4 - eX5 + fX6 - gX7$$

3.2 IP Design and Realization of IDCT

■ 8x8 2D IDCT Design Example

◆ **Reorganization 1：**

$$Y0 + Y7 = (dX0 + bX2 + dX4 + fX6) \times 2$$
$$Y1 + Y6 = (dX0 + fX2 - dX4 - bX6) \times 2$$
$$Y2 + Y5 = (dX0 - fX2 - dX4 + bX6) \times 2$$
$$Y3 + Y4 = (dX0 - bX2 + dX4 - fX6) \times 2$$

$$\frac{1}{2}\begin{bmatrix} Y0+Y7 \\ Y1+Y6 \\ Y2+Y5 \\ Y3+Y4 \end{bmatrix} = \begin{bmatrix} d & b & d & f \\ d & f & -d & -b \\ d & -f & -d & b \\ d & -b & d & -f \end{bmatrix} \begin{bmatrix} X0 \\ X2 \\ X4 \\ X6 \end{bmatrix}$$

3.2 IP Design and Realization of IDCT

■ 8x8 2D IDCT Design Example

◆ **Reorganization 2：**

$$Y0 - Y7 = (aX1 + cX3 + eX5 + gX7) \times 2$$
$$Y1 - Y6 = (cX1 - gX3 - aX5 - eX7) \times 2$$
$$Y2 - Y5 = (eX1 - aX3 + gX5 + cX7) \times 2$$
$$Y3 - Y4 = (gX1 - eX3 + cX5 - aX7) \times 2$$

$$\frac{1}{2}\begin{bmatrix} Y0 - Y7 \\ Y1 - Y6 \\ Y2 - Y5 \\ Y3 - Y4 \end{bmatrix} = \begin{bmatrix} a & c & e & g \\ c & -g & -a & -e \\ e & -a & g & c \\ g & -e & c & -a \end{bmatrix} \begin{bmatrix} X1 \\ X3 \\ X5 \\ X7 \end{bmatrix}$$

3.2 IP Design and Realization of IDCT

■ 8x8 2D IDCT Design Example

- **Fast Algorithm: Decomposed 2D 8x8 IDCT to two 1D 4x4 IDCTs**

$$\begin{bmatrix} Y0 \\ Y1 \\ Y2 \\ Y3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} Y0+Y7 \\ Y1+Y6 \\ Y2+Y5 \\ Y3+Y4 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} Y0-Y7 \\ Y1-Y6 \\ Y2-Y5 \\ Y3-Y4 \end{bmatrix}$$

$$\begin{bmatrix} Y7 \\ Y6 \\ Y5 \\ Y4 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} Y0+Y7 \\ Y1+Y6 \\ Y2+Y5 \\ Y3+Y4 \end{bmatrix} - \frac{1}{2}\begin{bmatrix} Y0-Y7 \\ Y1-Y6 \\ Y2-Y5 \\ Y3-Y4 \end{bmatrix}$$

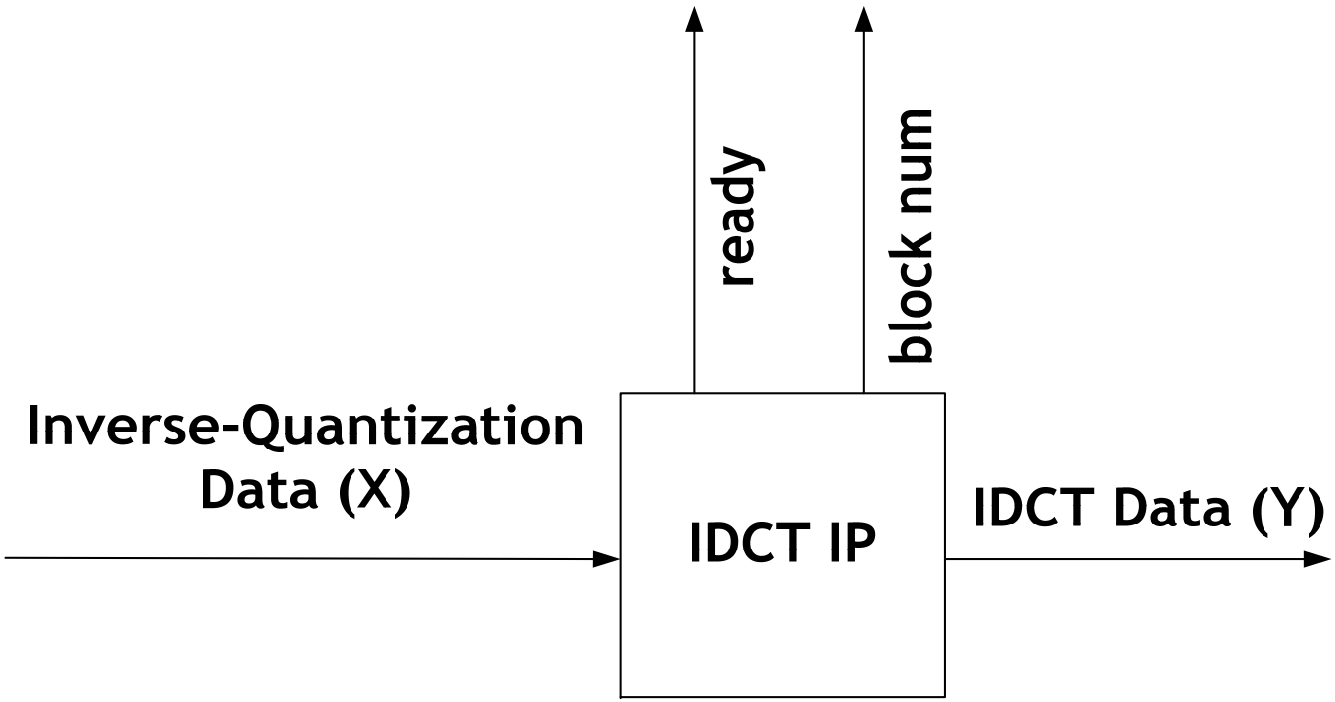
3.2 IP Design and Realization of IDCT

■ IDCT IP Realization

- IDCT IP Architecture
- IDCT IP Block Function
- IDCT IP Circuit Design

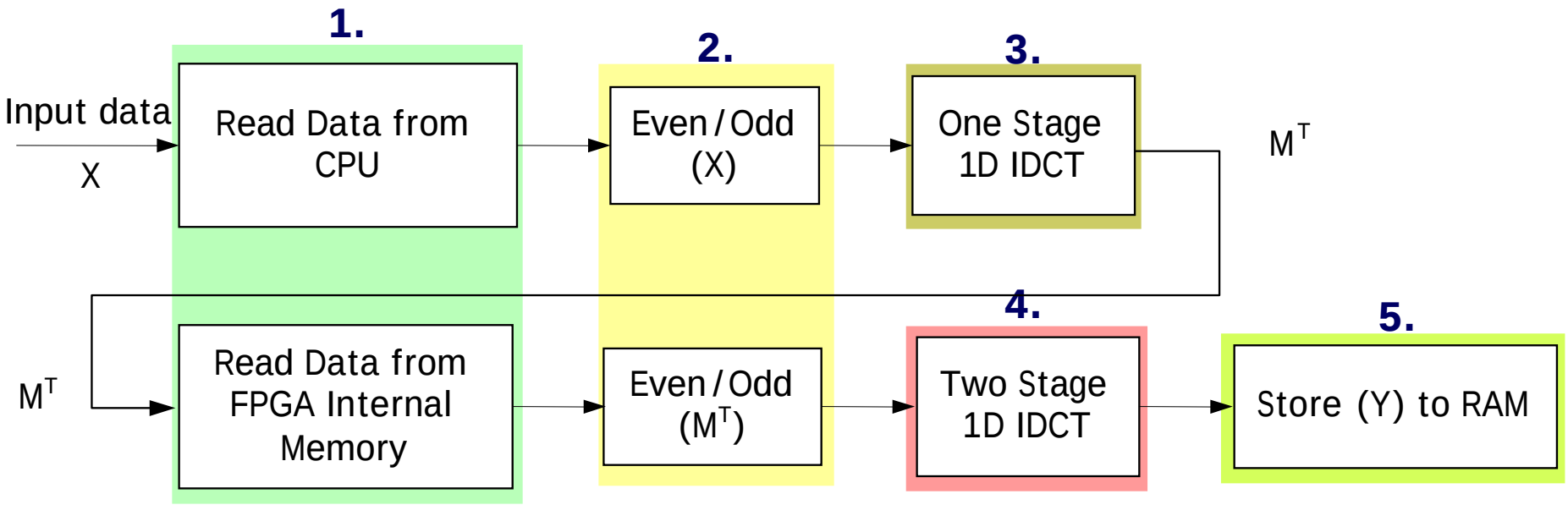
3.2 IP Design and Realization of IDCT

Input / Output of IDCT IP



3.2 IP Design and Realization of IDCT

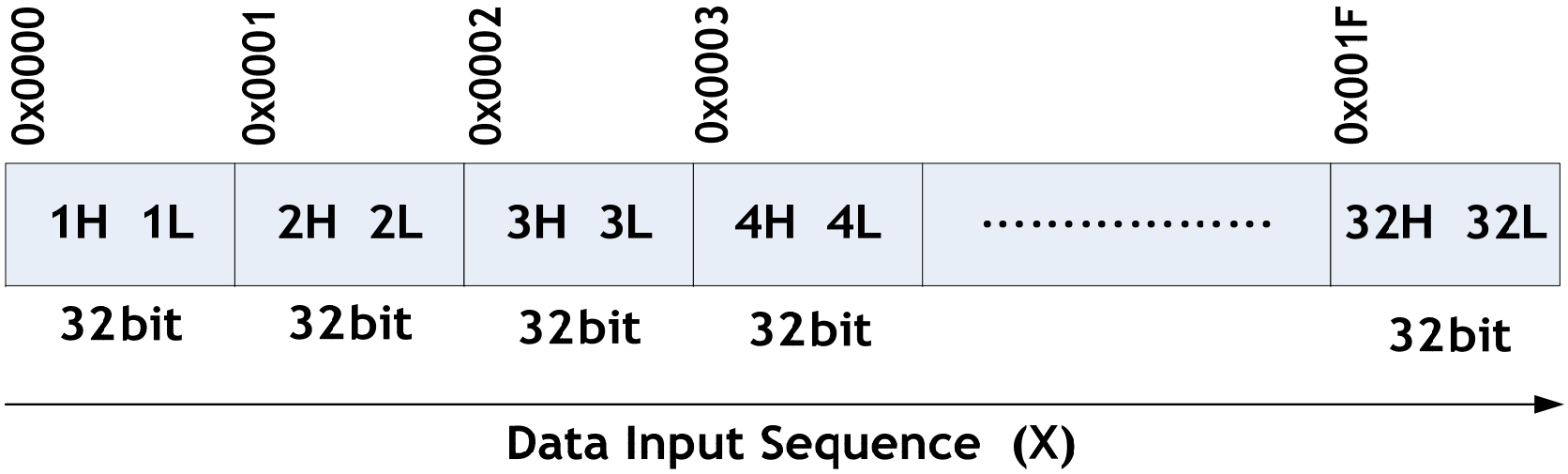
IDCT IP Architecture



3.2 IP Design and Realization of IDCT

■ IDCT IP Block Function

◆ Data Sequence of the IQ Coefficients from CPU to FPGA



L : [11:0] bit data
H : [27:16] bit data

3.2 IP Design and Realization of IDCT

■ IDCT IP Block Function

◆ Relations of the Input Sequence and 8x8 IDCT Coefficients

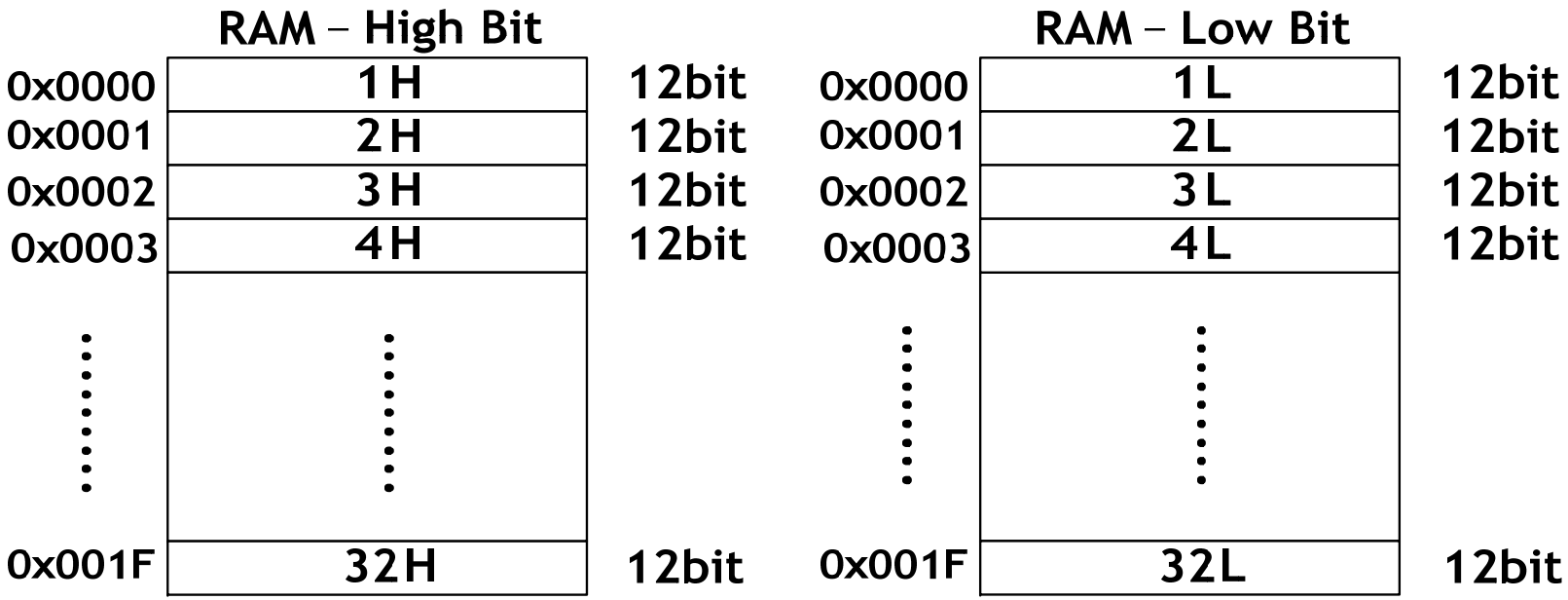
$$\begin{bmatrix} 1L & 5L & 9L & 13L & 17L & 21L & 25L & 29L \\ 1H & 5H & 9H & 13H & 17H & 21H & 25H & 29H \\ 2L & 6L & 10L & 14L & 18L & 22L & 26L & 30L \\ 2H & 6H & 10H & 14H & 18H & 22H & 26H & 30H \\ 3L & 7L & 11L & 15L & 19L & 23L & 27L & 31L \\ 3H & 7H & 11H & 15H & 19H & 23H & 27H & 31H \\ 4L & 8L & 12L & 16L & 20L & 24L & 28L & 32L \\ 4H & 8H & 12H & 16H & 20H & 24H & 28H & 32H \end{bmatrix}$$

- L：16 LSB bits of a 32-bit data access
- H：16 MSB bits of a 32-bit data access

3.2 IP Design and Realization of IDCT

■ IDCT IP Block Function

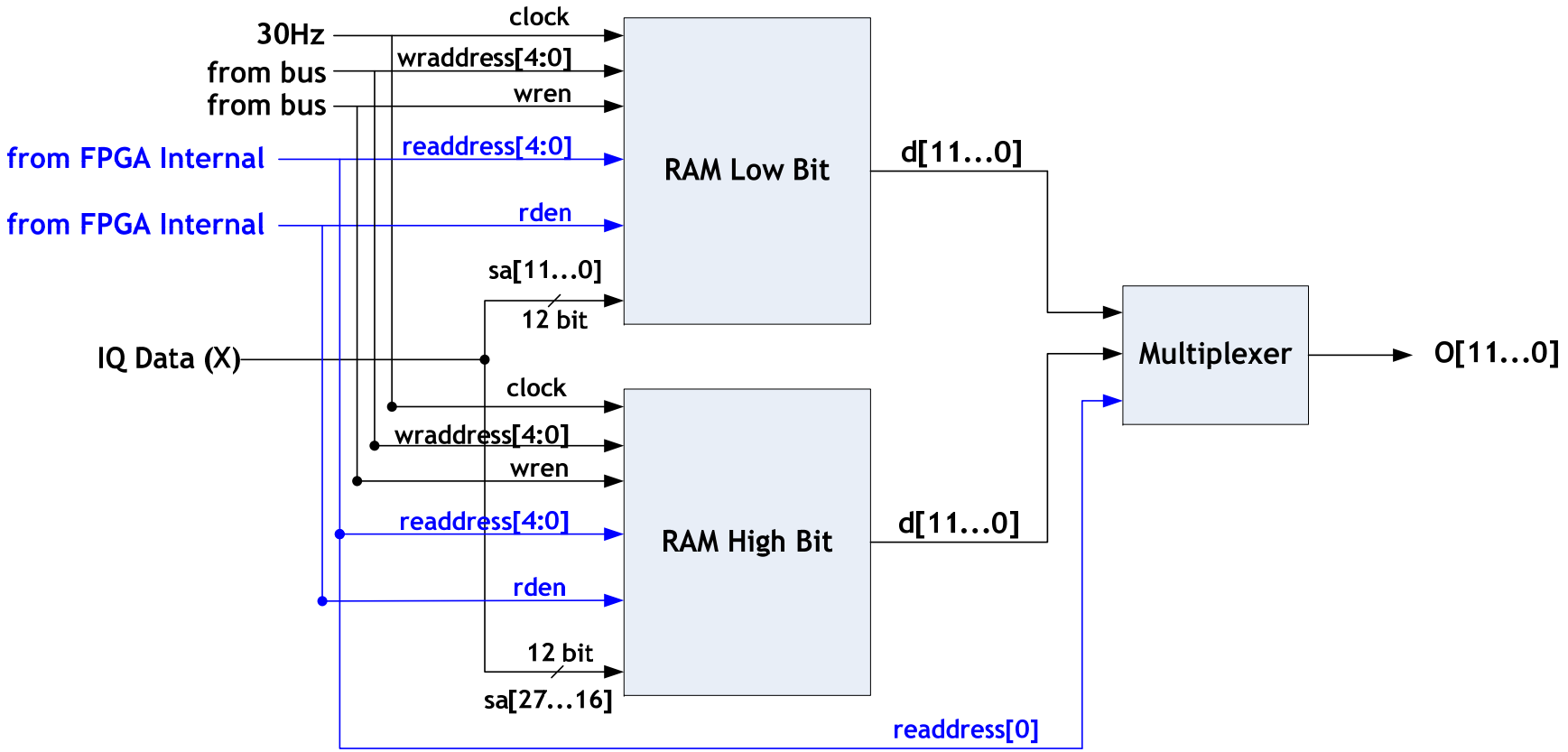
◆ IQ Data Location in the FPGA Internal Memory



3.2 IP Design and Realization of IDCT

■ IDCT IP Block Function

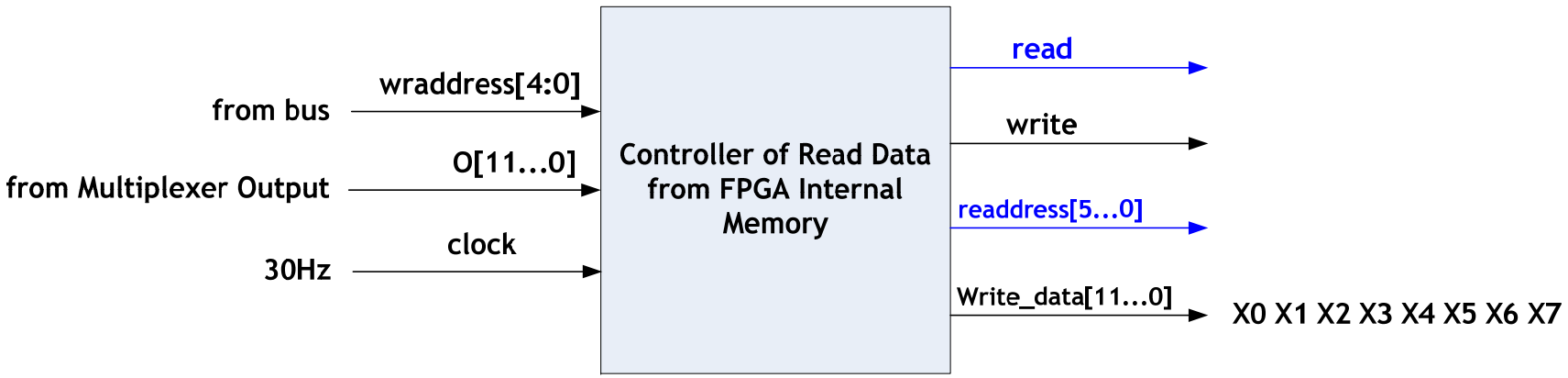
- Multiplexer to read and select the data



3.2 IP Design and Realization of IDCT

■ IDCT IP Block Function

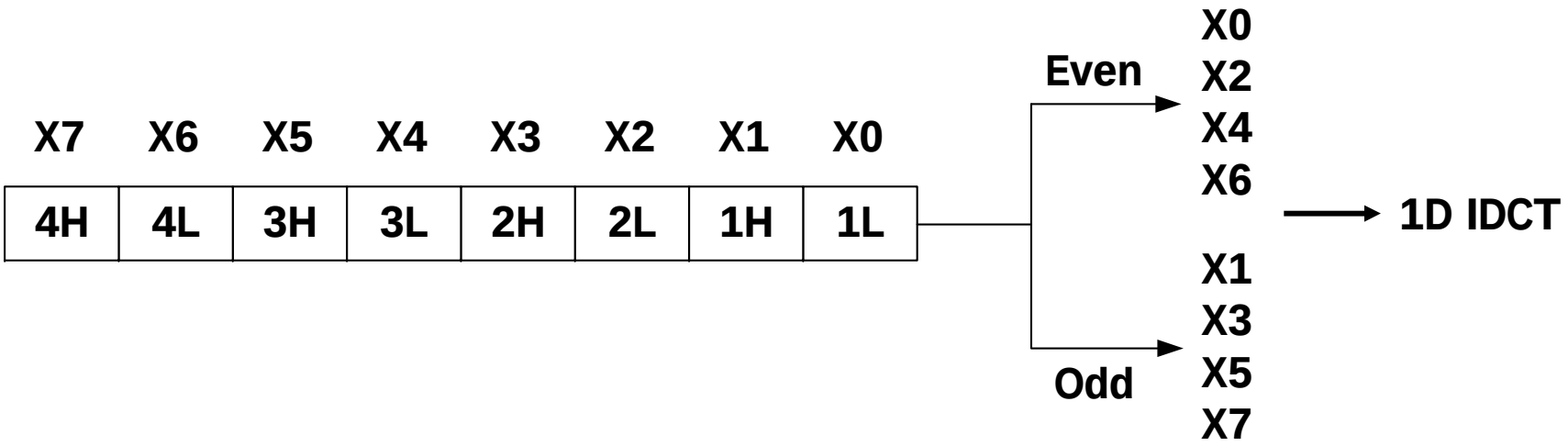
- Controller of the data read from the FPGA internal memory



3.2 IP Design and Realization of IDCT

■ IDCT IP Block Function

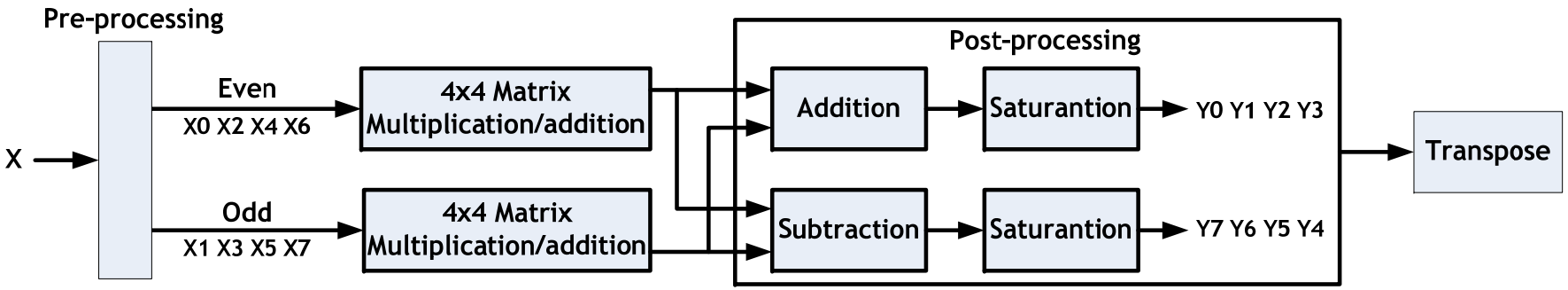
◆ Even/Odd IQ Coefficients of Decomposition



3.2 IP Design and Realization of IDCT

■ IDCT IP Block Function

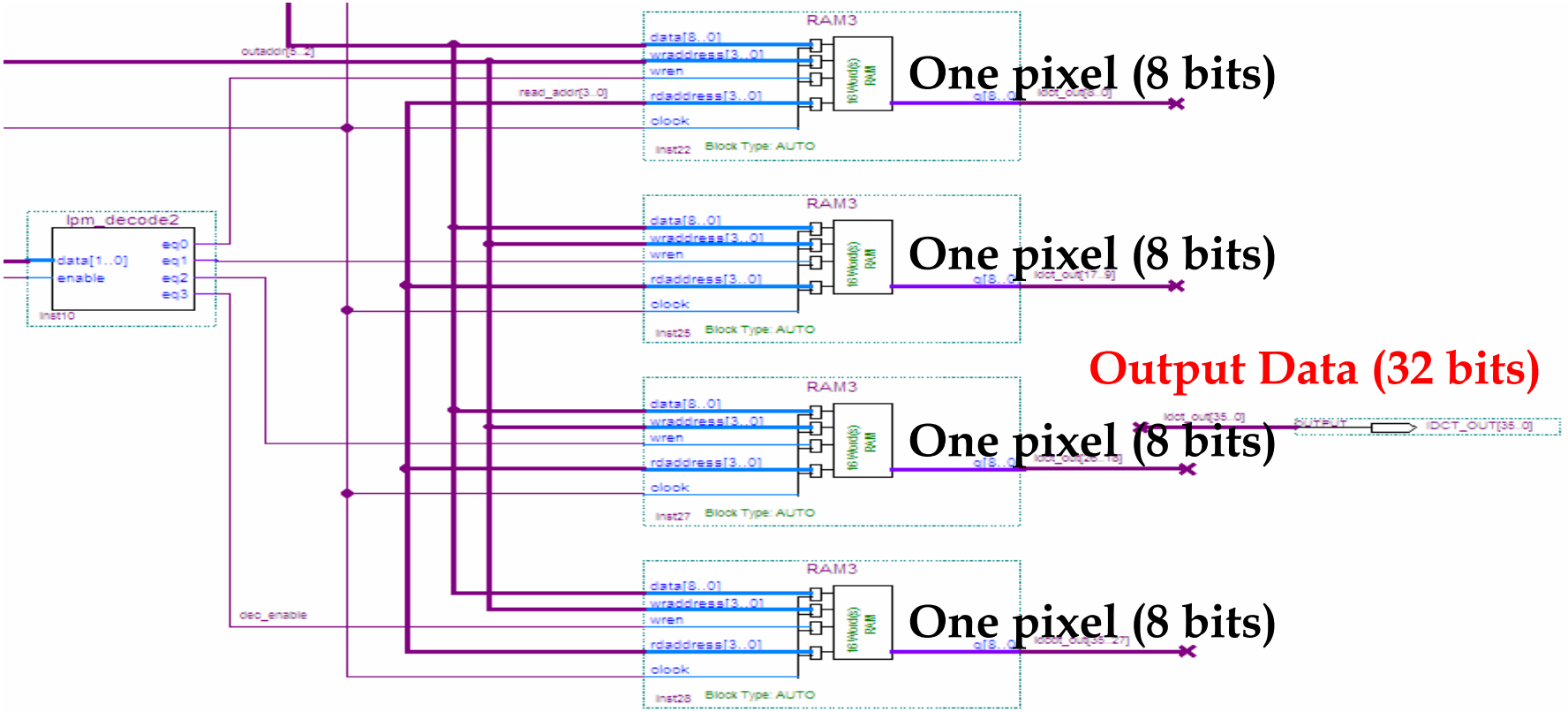
◆ 4x4 1D IDCT



3.2 IP Design and Realization of IDCT

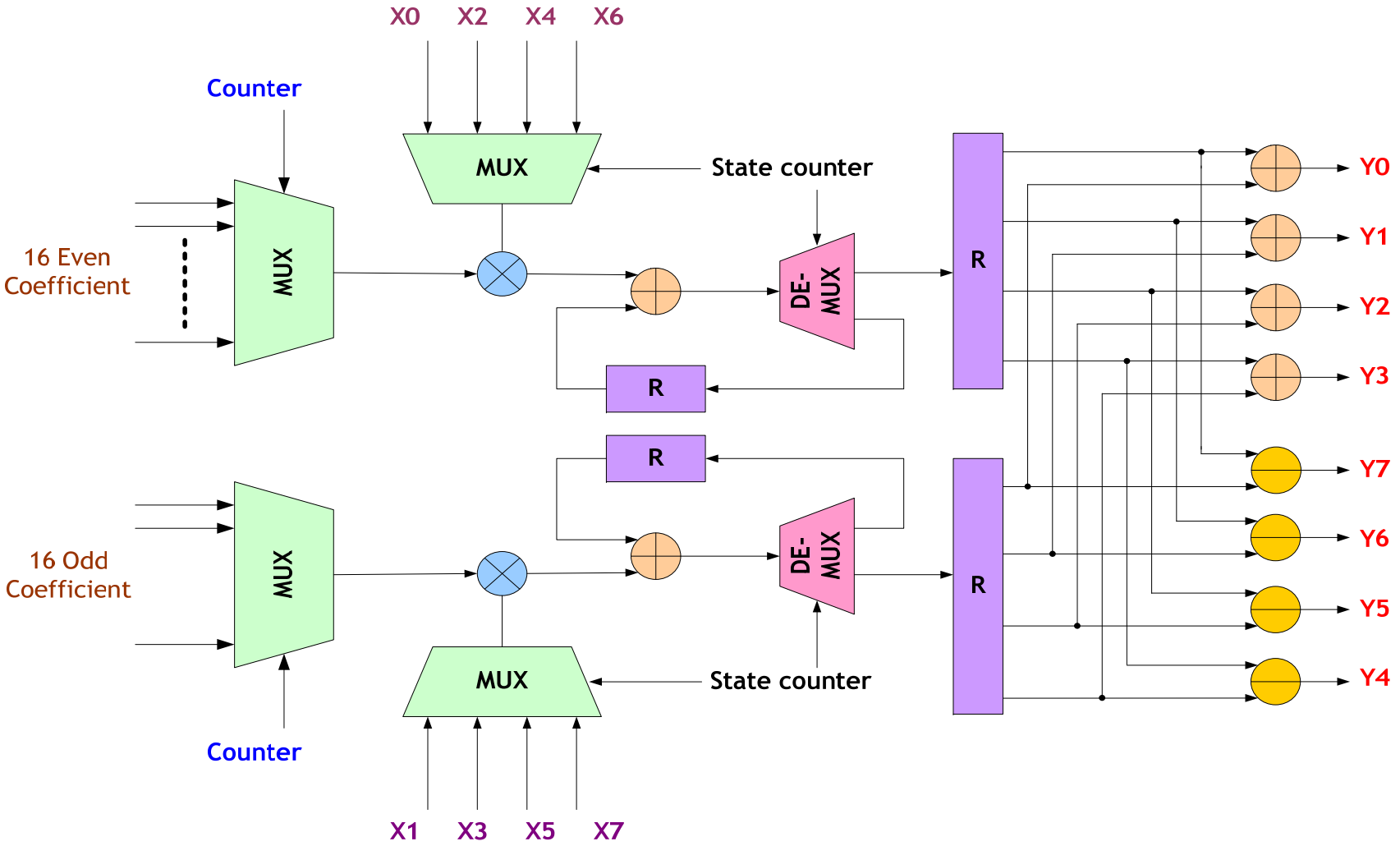
IDCT IP Block Function

- Store Output Data to RAM
  - One pixel = 8 bits



3.2 IP Design and Realization of IDCT

■ IDCT IP Circuit Design



3.3 IP Functional Verification

3.1 Design Environment of the PXA/255 FPGA Module

3.2 IP Design and Realization of IDCT

3.3 IP Functional Verification

3.3 IP Functional Verification

■ 2D IDCT IP Verification

Step1: Dump the IDCT I/P and O/P Data from the MPEG-2 Reference Software

Step2: Create Input Waveform into Quartus-II

Step3: Check of the Quartus-II Output Waveform

3.3 IP Functional Verification

■ Dump IDCT I/P and O/P Data from Reference SW

◆ File Name：idct.c

```
-----------------------------------------------------------------------
static void idctcol(blk)
short *blk;
{
FILE *IDCT_OUT;
int x0,x1,x2,x3,x4,x5,x6,x7,x8;
IDCT_OUT=fopen("c:\IDCT_OUT.txt","a");
x0 = (blk[8*0]<<8) + 8192;
/* first stage */
x8 = W7*(x4+x5) + 4;
x4 = (x8+(W1-W7)*x4)>>3;
x5 = (x8-(W1+W7)*x5)>>3;
x8 = W3*(x6+x7) + 4;
x6 = (x8-(W3-W5)*x6)>>3;
x7 = (x8-(W3+W5)*x7)>>3;
/* second stage */
x8 = x0 + x1;
x0 -= x1;
x1 = W6*(x3+x2) + 4;
x2 = (x1-(W2+W6)*x2)>>3;
x3 = (x1+(W2-W6)*x3)>>3;
x1 = x4 + x6;
x4 -= x6;
-----------------------------------------------------------------------
```

3.3 IP Functional Verification

■ Dump IDCT I/P and O/P Data from Reference SW

```
--------------------------------
x6 = x5 + x7;
x5 -= x7;
/* third stage */
x7 = x8 + x3;
x8 -= x3;
x3 = x0 + x2;
x0 -= x2;
x2 = (181*(x4+x5)+128)>>8;
x4 = (181*(x4-x5)+128)>>8;
/* fourth stage */
blk[8*0] = iclp[(x7+x1)>>14];
blk[8*1] = iclp[(x3+x2)>>14];
blk[8*2] = iclp[(x0+x4)>>14];
blk[8*3] = iclp[(x8+x6)>>14];
blk[8*4] = iclp[(x8-x6)>>14];
blk[8*5] = iclp[(x0-x4)>>14];
blk[8*6] = iclp[(x3-x2)>>14];
blk[8*7] = iclp[(x7-x1)>>14];
--------------------------------
```

```
----------------------------------------
fprintf(IDCT_OUT,"idct_out1=%d\n",blk[8*0]);
fprintf(IDCT_OUT,"idct_out2=%d\n",blk[8*1]);
fprintf(IDCT_OUT,"idct_out3=%d\n",blk[8*2]);
fprintf(IDCT_OUT,"idct_out4=%d\n",blk[8*3]);
fprintf(IDCT_OUT,"idct_out5=%d\n",blk[8*4]);
fprintf(IDCT_OUT,"idct_out6=%d\n",blk[8*5]);
fprintf(IDCT_OUT,"idct_out7=%d\n",blk[8*6]);
fprintf(IDCT_OUT,"idct_out8=%d\n",blk[8*7]);
fclose(IDCT_OUT);
}
----------------------------------------
```

3.3 IP Functional Verification

■ Dump IDCT I/P Data from Reference SW

- **Simulation Result of the MEPG-2 Reference Software**
  - **IQ_data (X)**

[0]	224
[1]	-380
[2]	-403
[3]	-220
[4]	0
[5]	135
[6]	72
[7]	0
[8]	100
[9]	-100
[10]	55
[11]	0
[12]	0
[13]	0
[14]	0
[15]	0
[16]	71
[17]	-55
[18]	-32
[19]	67
[20]	0
[21]	0
[22]	0
[23]	0
[24]	55
[25]	-27
[26]	-32
[27]	33
[28]	0
[29]	-42
[30]	0
[31]	0
[32]	-27
[33]	0
[34]	101

3.3 IP Functional Verification

Dump IDCT I/P Data from Reference SW

- Simulation Result of the MEPG-2 Reference Software
  - IDCT_out (Y)

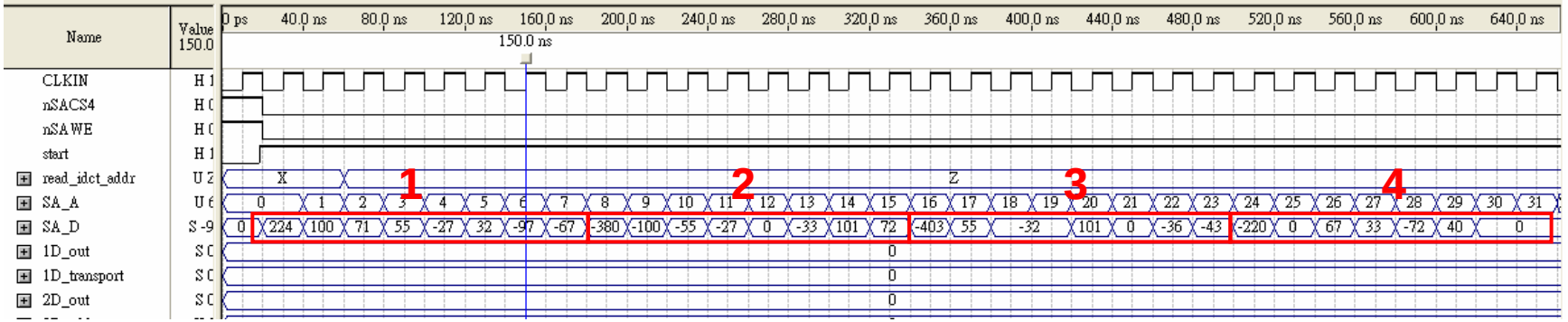
```
idct_out1=-98
idct_out2=-128
idct_out3=-116
idct_out4=-107
idct_out5=-131
idct_out6=-124
idct_out7=-132
idct_out8=-109
idct_out1=-84
idct_out2=-72
idct_out3=-96
idct_out4=-83
idct_out5=-70
idct_out6=-77
idct_out7=-79
idct_out8=-100
idct_out1=59
idct_out2=79
idct_out3=65
idct_out4=78
idct_out5=96
idct_out6=64
idct_out7=67
idct_out8=72
idct_out1=121
```

3.3 IP Functional Verification

■ Create the Quartus-II Input Waveform

◆ Created Input Waveform (1)

■ IQ_data (X)



● 8x8 IQ_data (X)

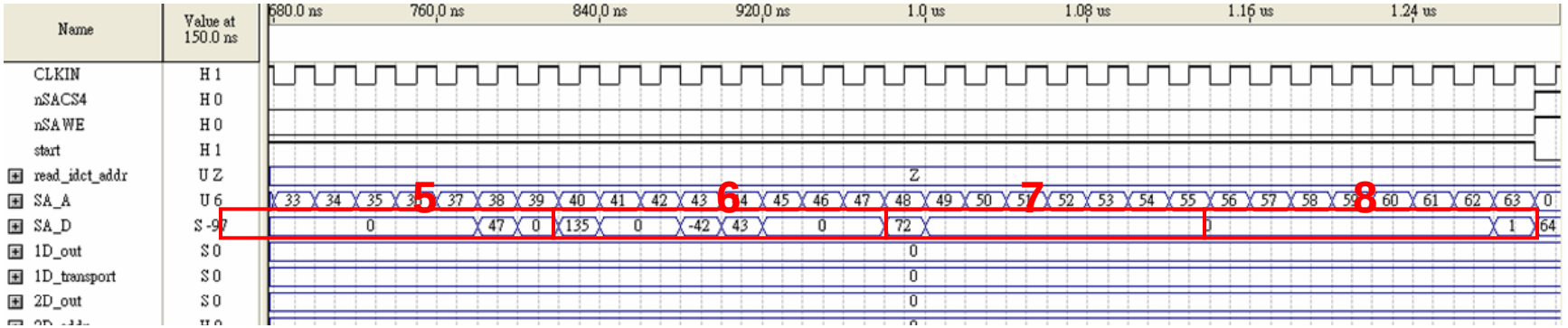
1	2	3	4				
224	-380	-403	-220	0	135	72	0
100	-100	55	0	0	0	0	0
71	-55	-32	67	0	0	0	0
55	-27	-32	33	0	-42	0	0
-27	0	101	-72	0	43	0	0
32	-33	0	40	0	0	0	0
-97	101	-36	0	47	0	0	0
-67	72	-43	0	0	0	0	1

3.3 IP Functional Verification

■ Create the Quartus-II Input Waveform

◆ Created Input Waveform (2)

■ IQ_data (X)



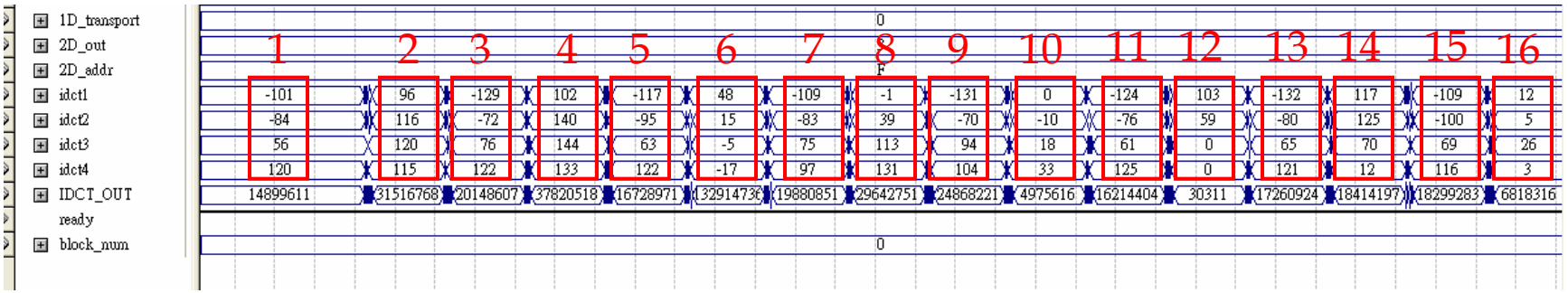
● 8x8 IQ_data (X)

				5	6	7	8
224	-380	-403	-220	0	135	72	0
100	-100	55	0	0	0	0	0
71	-55	-32	67	0	0	0	0
55	-27	-32	33	0	-42	0	0
-27	0	101	-72	0	43	0	0
32	-33	0	40	0	0	0	0
-97	101	-36	0	47	0	0	0
-67	72	-43	0	0	0	0	1

3.3 IP Functional Verification

Create the Quartus-II Input Waveform

- **Simulation Output of 2D IDCT**
  - **IDCT_out (Y)**



3.3 IP Functional Verification

■ Check of the Quartus-II Output Waveform

◆ Comparisons

224	-380	-403	-220	0	135	72	0
100	-100	55	0	0	0	0	0
71	-55	-32	67	0	0	0	0
55	-27	-32	33	0	-42	0	0
-27	0	101	-72	0	43	0	0
32	-33	0	40	0	0	0	0
-97	101	-36	0	47	0	0	0
-67	72	-43	0	0	0	0	1

IDCT Transform →

1	3	5	7	9	11	13	15
-98	-128	-116	-107	-131	-124	-132	-109
-84	-72	-96	-83	-70	-77	-79	-100
59	79	65	78	96	64	67	72
121	123	123	98	106	126	122	118
97	103	49	0	0	104	119	13
116	141	15	39	-10	60	127	5
120	146	-5	115	17	-1	69	24
116	134	-17	133	34	1	12	4
2	4	6	8	10	12	14	16

Check OK !