Case Study
JPEG Encoder

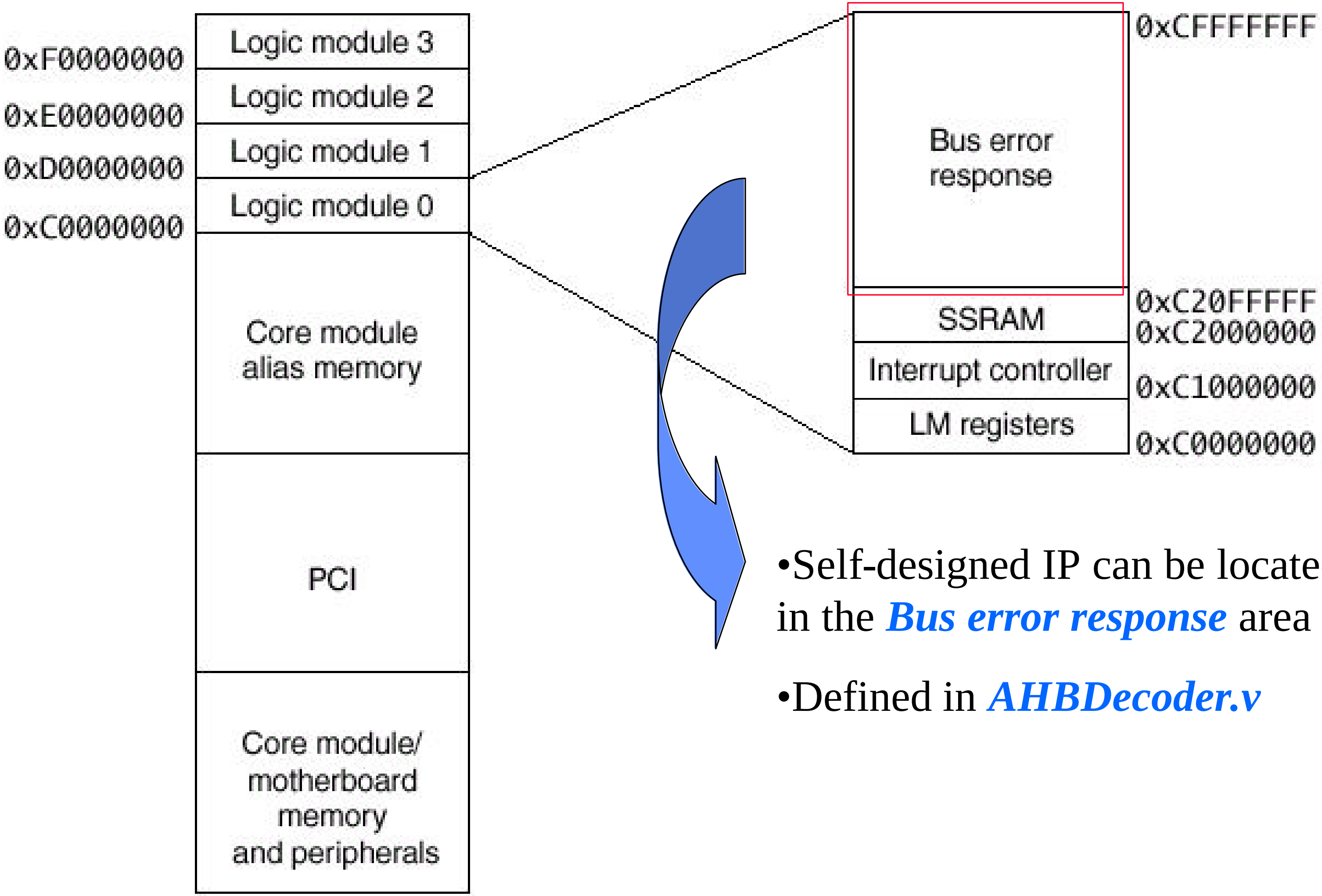
Speaker: **沈文中**

Code Supported by NCTU

Outline

- *How to design an IP with AMBA standard*
- JPEG Spec.
- Lab requirement

Memory definition



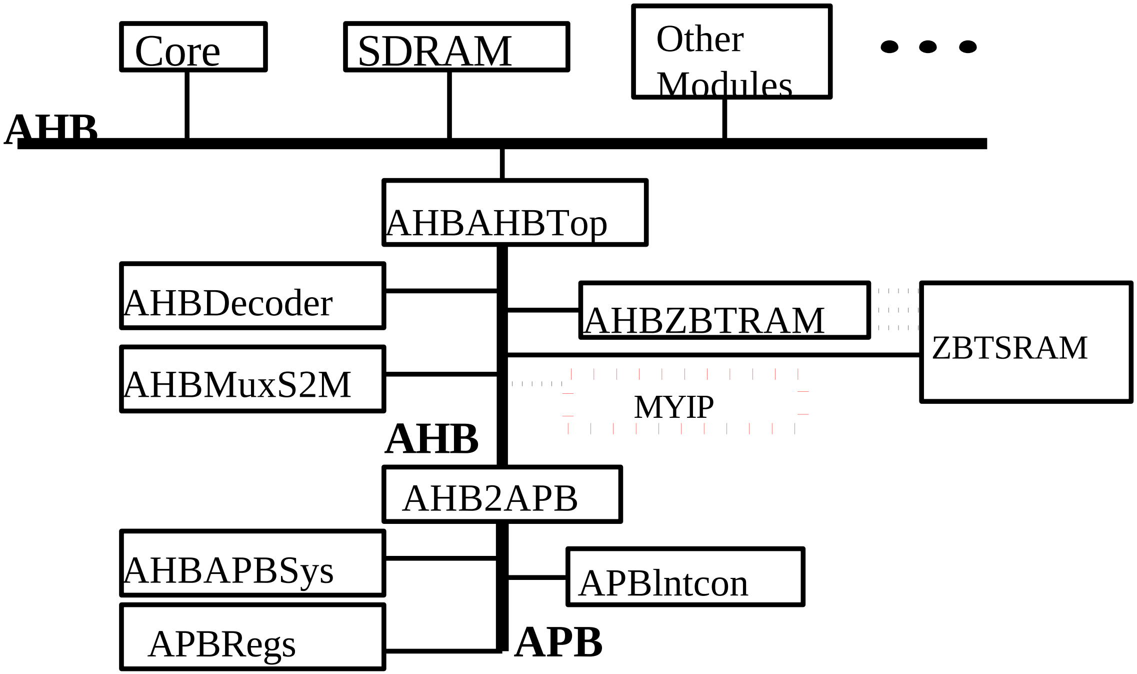
- Self-designed IP can be located in the ***Bus error response*** area
- Defined in ***AHBDecoder.v***

AMBA IP design

- ❑ Software part
  - – Write a function to control hardware
  - – Delay number of clocks by NOOP (asm) instruction
- ❑ Hardware part
  - – Add MYIP.v into top module
  - – Change ahbdecoder.v
  - – Change AHBMuxS2M.v
  - – Change ahbahbtop.v

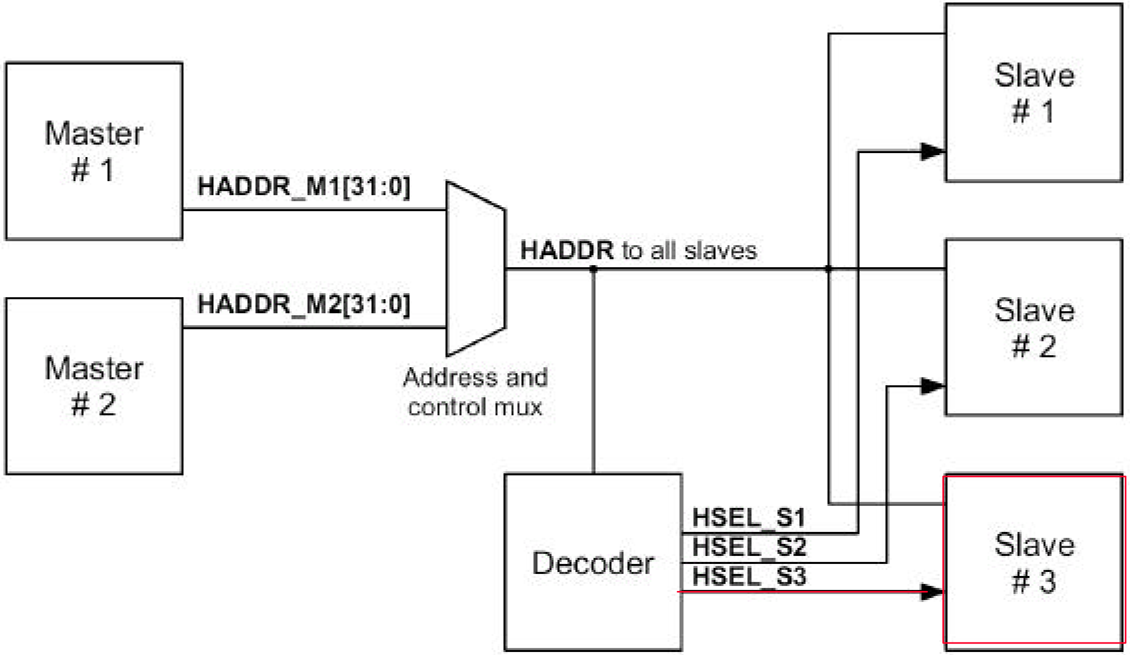
IP design-HW

- ❑ Add MYIP.v into top module
  - – Write your own module in AMBA interface



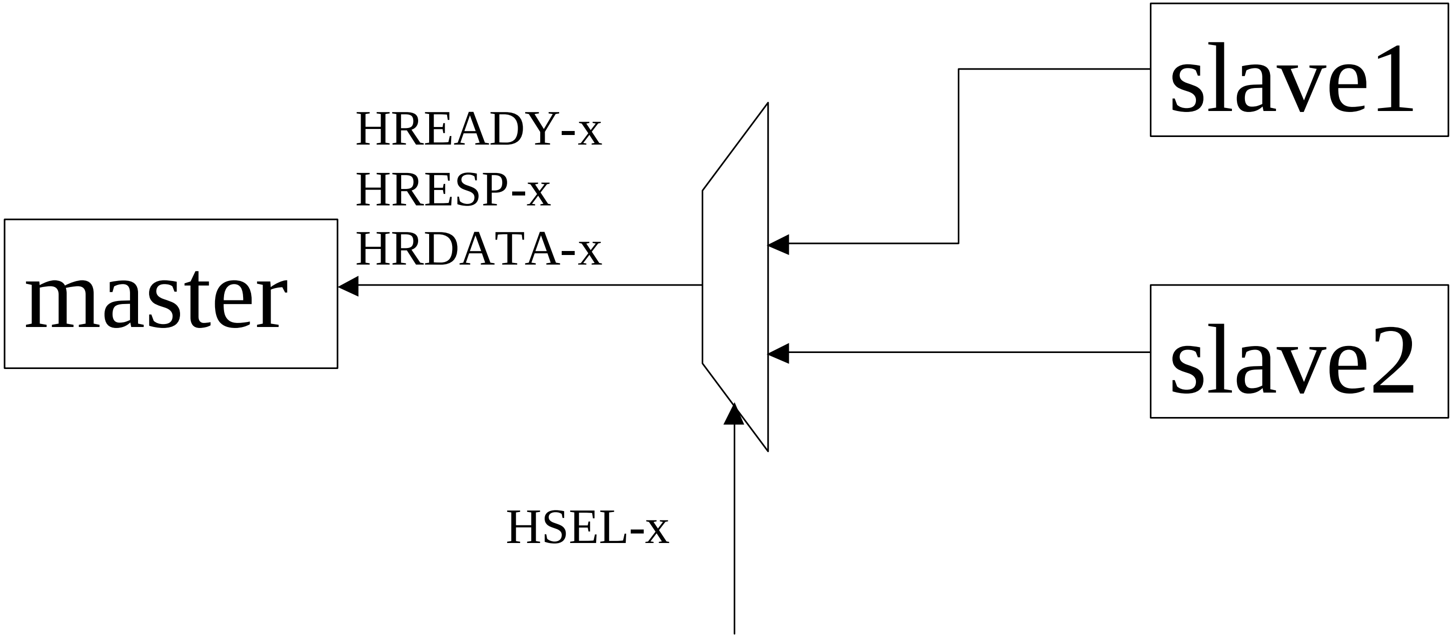
IP design-HW

- ❑ Change ahbdecoder.v
  - – Add **HSELMYIP** signal to select your own slave IP to response
  - – Address are defined in decoder



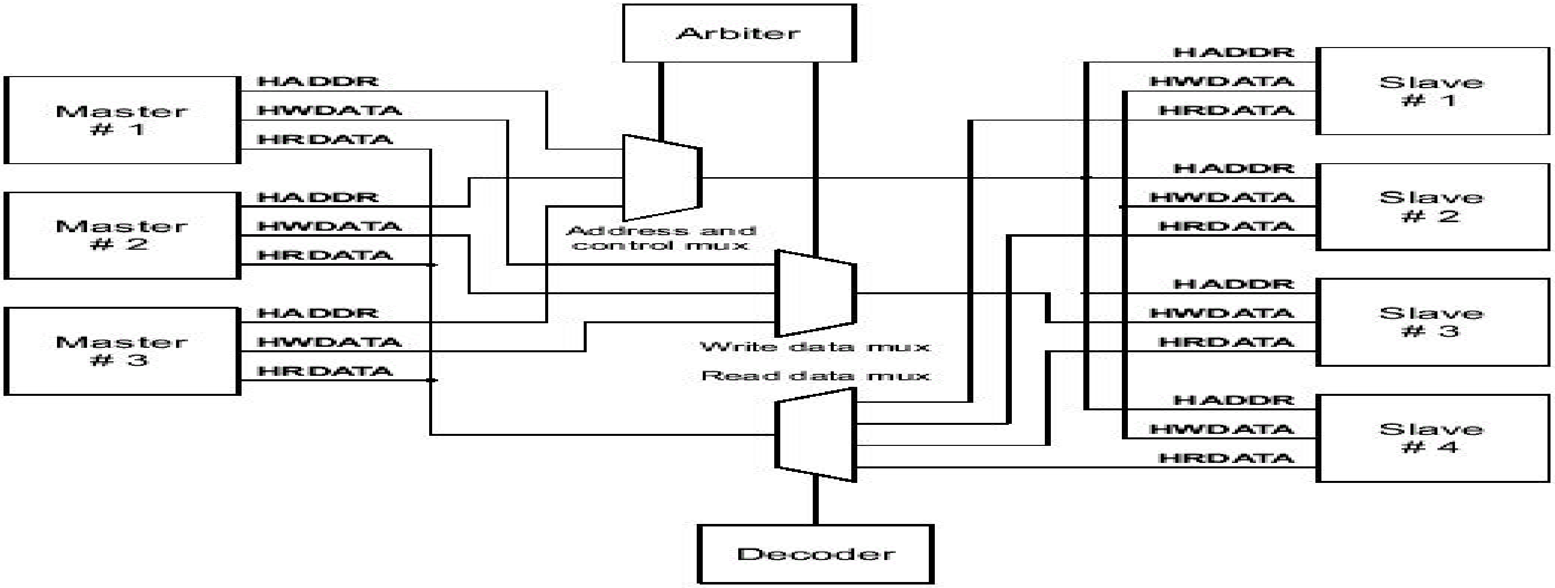
IP design-HW

- ❑ Change AHBMuxS2M.v
  - – Use mux to select slave which can use HRDATA

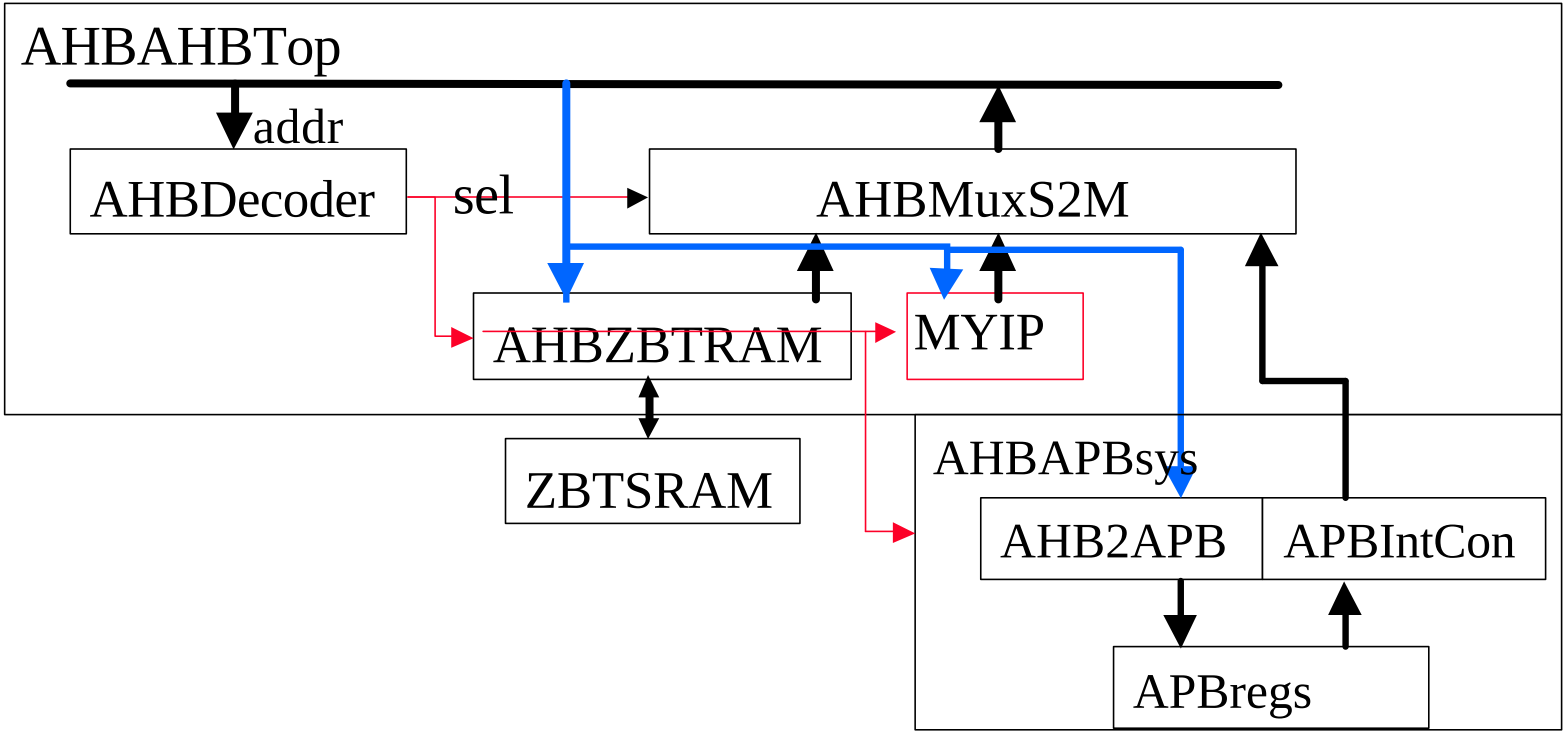


IP design-HW

- ❑ Change ahbahbtop.v
  - – Add your module in AMBA Bus
  - – Connect the above net connection



Architecture



Outline

- ❑ How to design an IP with AMBA standard
- ❑ *JPEG Spec.*
- ❑ Lab requirement

JPEG spec.

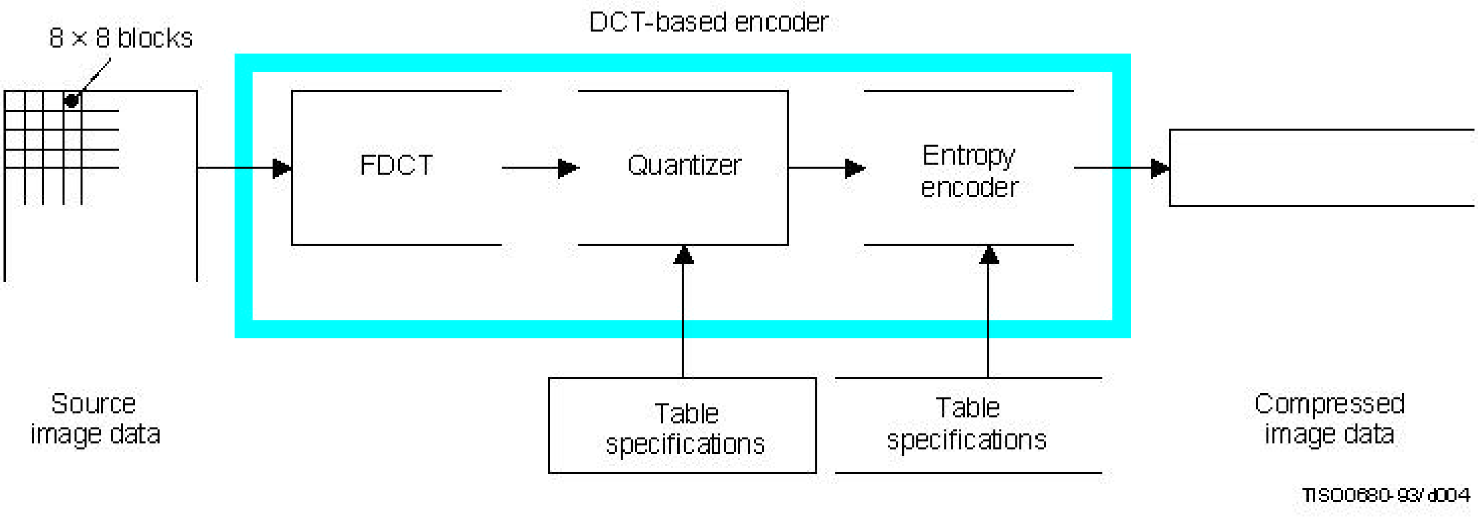
□ Target :JPEG baseline sequential DCT-based encoder

□ Content

- RGB->YUV
- DCT (discrete cosine transform)
- DQT (quantization)
- Zig-Zag scan
- Entropy coding (run length coding+Huffman coding)

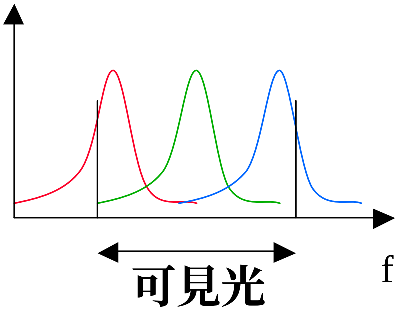
JPEG encoder overview

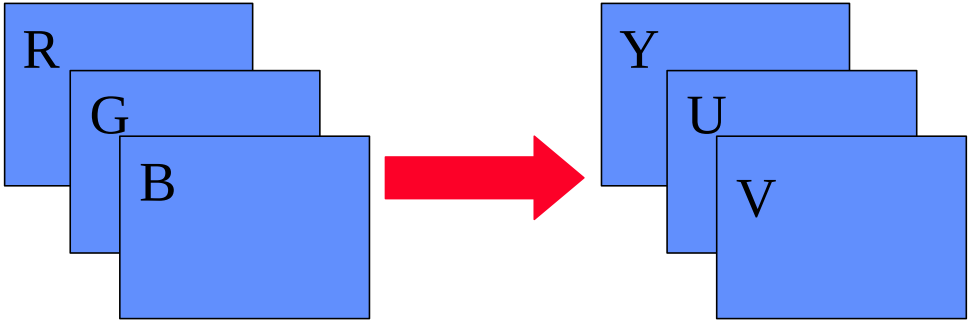
- ❑ Split into 8x8 pixels per block
- ❑ Use FDCT, Quantizer, and Entropy encoder to compress data



RGB->YUV

- Y = 0.299 R + 0.587 G + 0.114 B
- Cb = - 0.1687 R - 0.3313 G + 0.5 B + 128
- Cr = 0.5 R - 0.4187 G - 0.0813 B + 128

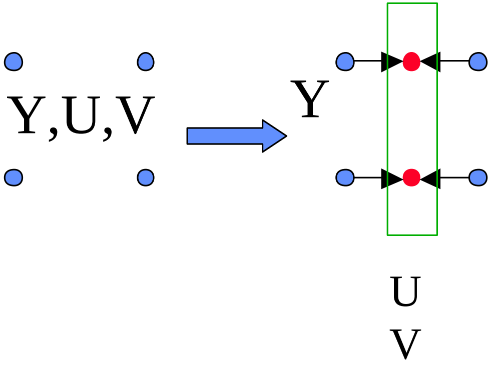




- Y-Luminance(亮度) Cb,Cr-Chrominance domain(色差)

YUV arrange

- ❑ Order of block is from left to right, and then from up to down.
- ❑ Each block obeys the order of Y->U->V
- ❑ YUV can be sub-sampled as 4:2:2 ratio



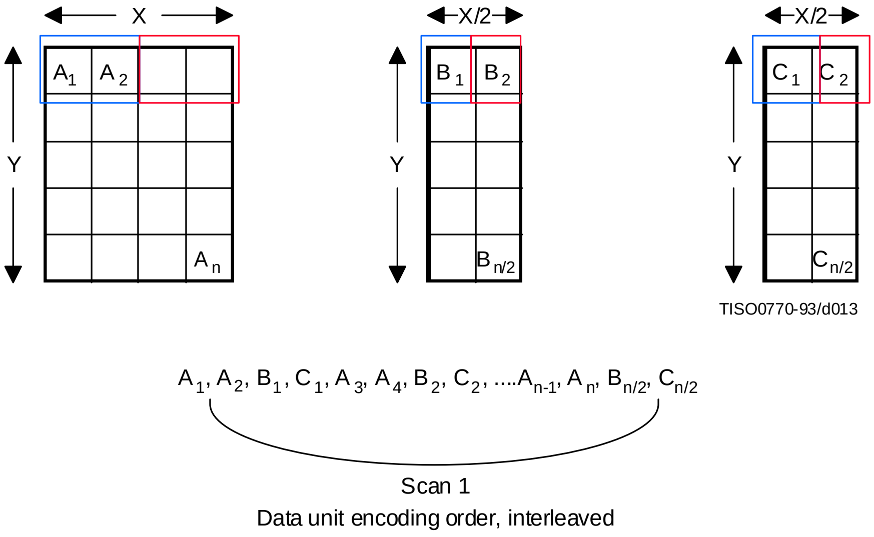
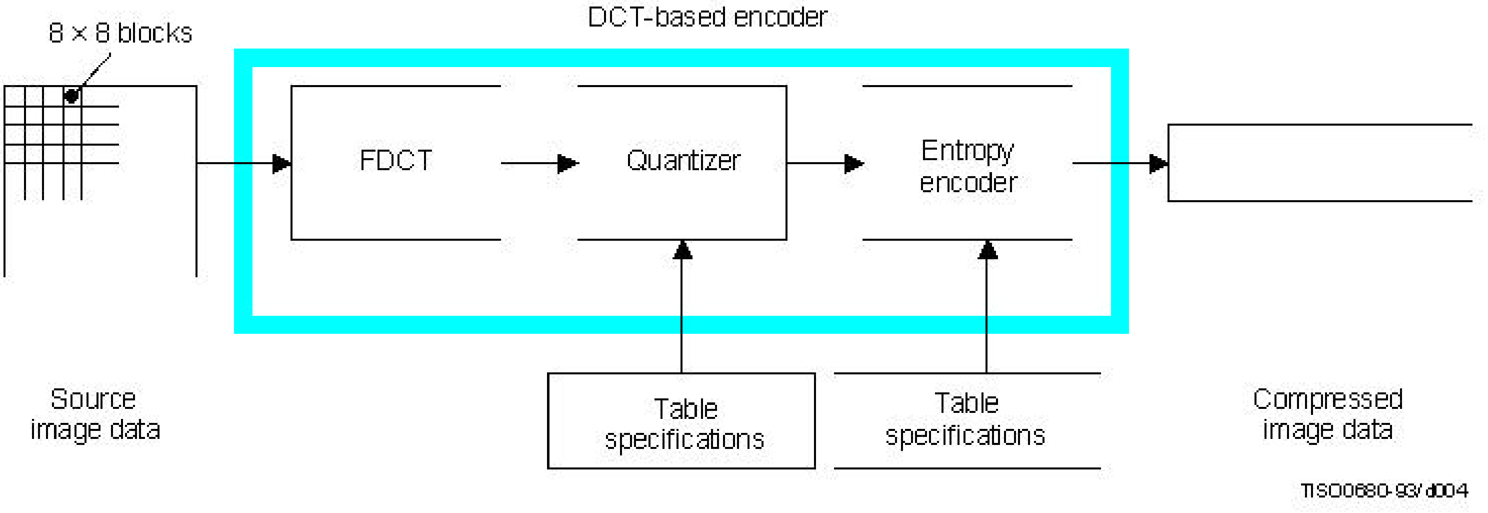

X
X/2
X/2
Y
Y
Y
A1
A2
An
B1
B2
Bn/2
C1
C2
Cn/2
TISO0770-93/d013
A1, A2, B1, C1, A3, A4, B2, C2, ...An-1, An, Bn/2, Cn/2
Scan 1
Data unit encoding order, interleaved

Figure 13 – Interleaved order for components with different dimensions

DCT

- ❑ Encoding procedure use 8x8 pixels as a compressing unit
- ❑ FDCT is composed of 2 times of 1D-DCT



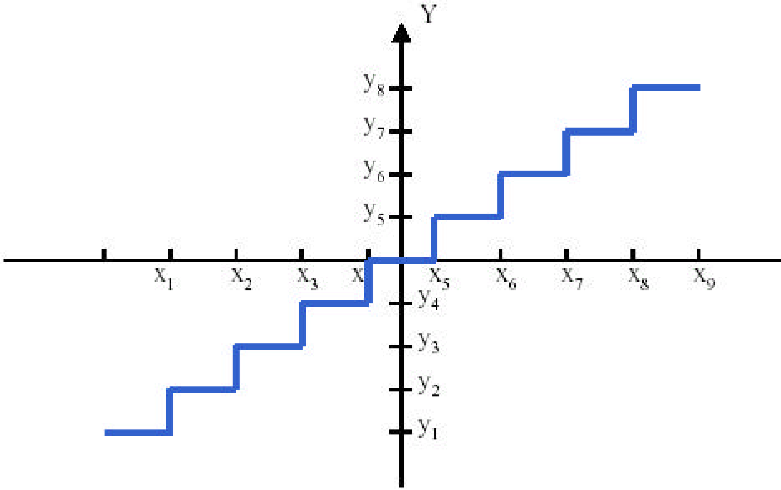
Quantization

- Less bits is needed to code the quantization coefficient
- static BYTE std_luminance_qt[64] = {

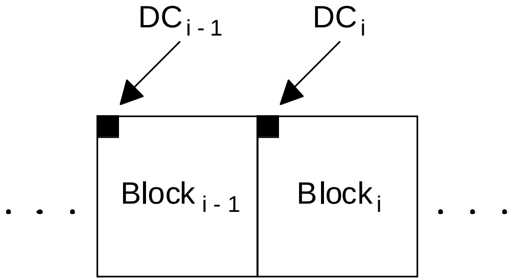
```
16,  11,  10,  16,  24,  40,  51,  61,
12,  12,  14,  19,  26,  58,  60,  55,
14,  13,  16,  24,  40,  57,  69,  56,
14,  17,  22,  29,  51,  87,  80,  62,
18,  22,  37,  56,  68, 109, 103,  77,
24,  35,  55,  64,  81, 104, 113,  92,
49,  64,  78,  87, 103, 121, 120, 101,
72,  92,  95,  98, 112,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

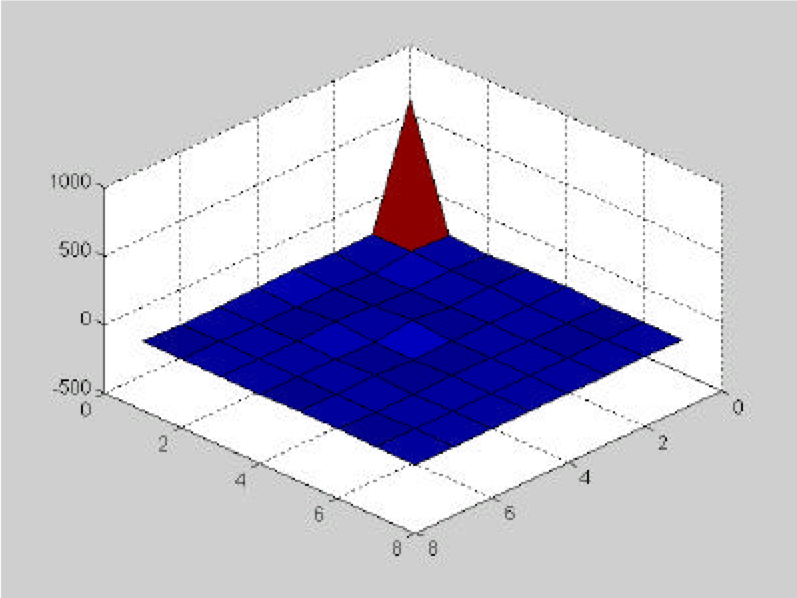
- static BYTE std_chrominance_qt[64] =

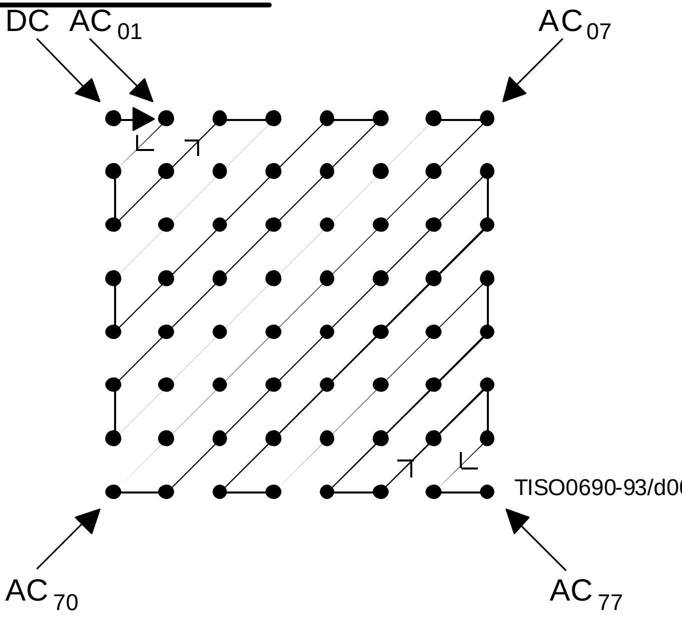
```
17,  18,  24,  47,  99,  99,  99,  99,
18,  21,  26,  66,  99,  99,  99,  99,
24,  26,  56,  99,  99,  99,  99,  99,
47,  66,  99,  99,  99,  99,  99,  99,
99,  99,  99,  99,  99,  99,  99,  99,
99,  99,  99,  99,  99,  99,  99,  99,
99,  99,  99,  99,  99,  99,  99,  99,
99,  99,  99,  99,  99,  99,  99,  99
};
```



Zig-Zag scan







Zig-zag order

zed coefficients for entropy encoding

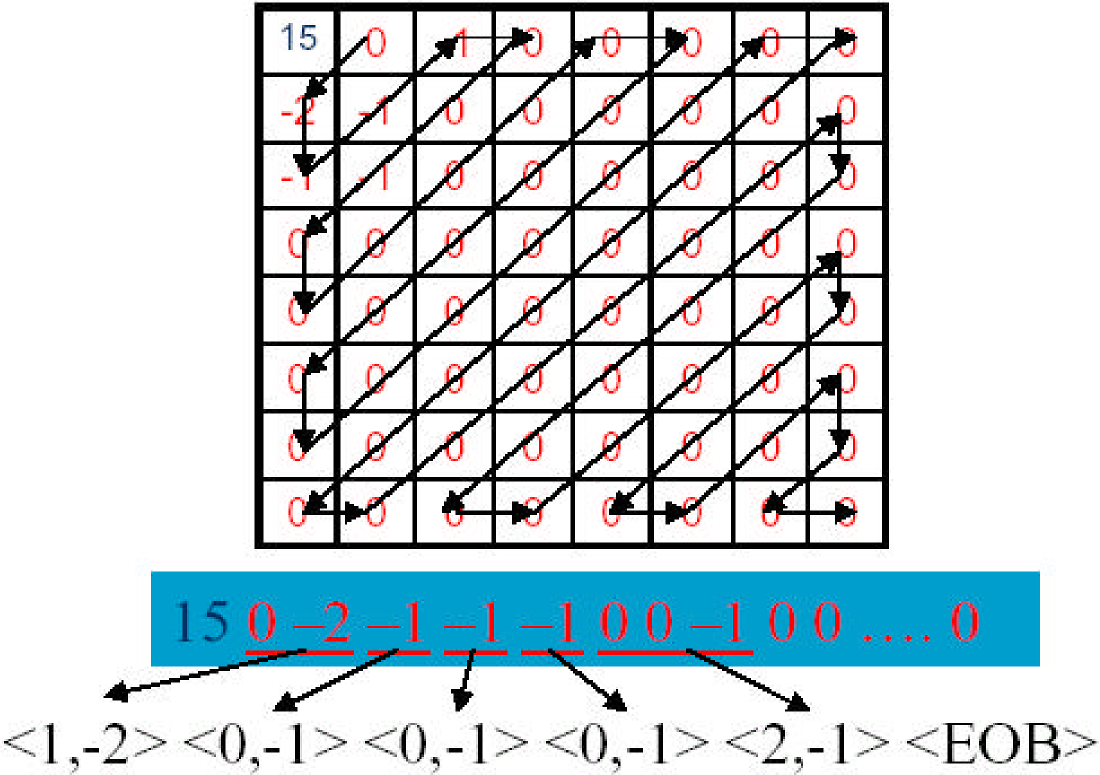
Entropy Coding

- ***Entropy coding*** transfer the quantized data to compressed data
- Entropy coding is composed of ***Run Length Coding*** and ***Variable Length Coding (Huffman Coding)***

□ ***15,0,-2,-1,-1,-1,0,0,…***	Quantized data
□ ***->15,(1,-2)(0,-1)(0,-1)(0,-1),..***	After RLC
□ ***->101111…***	After VLC

AC Run Length Coding

- ❑ *Zero* is the value occurs at highest frequency in quantized data

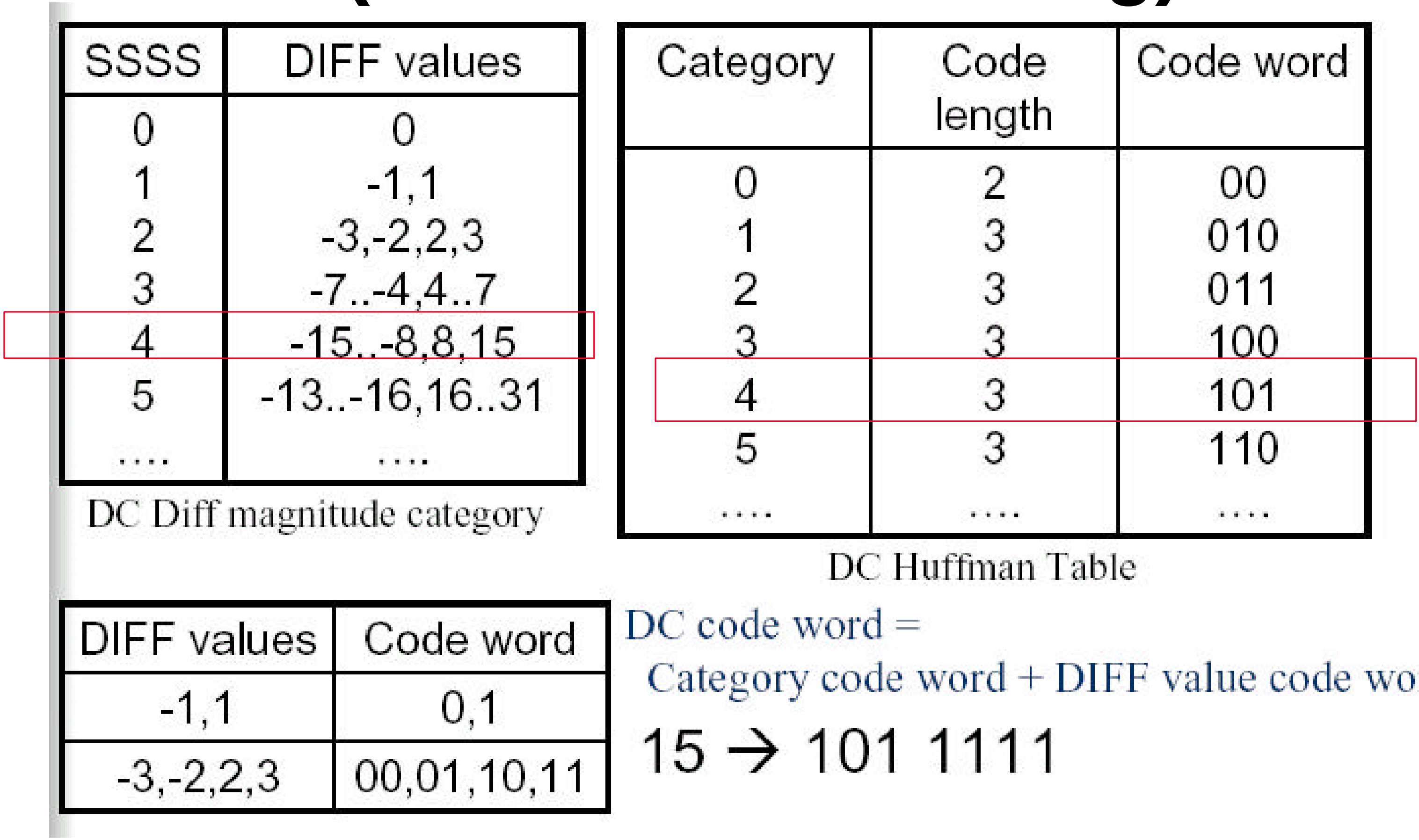


15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

15 0 –2 –1 –1 –1 0 0 –1 0 0 0

<1,-2> <0,-1> <0,-1> <0,-1> <2,-1> <EOB>

Variable Length Coding (DC Huffman Coding)



SSSS	DIFF values
0	0
1	-1,1
2	-3,-2,2,3
3	-7..-4,4..7
4	-15..-8,8,15
5	-13..-16,16..31
….	….

DC Diff magnitude category

Category	Code length	Code word
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
….	….	….

DC Huffman Table

DIFF values	Code word
-1,1	0,1
-3,-2,2,3	00,01,10,11

DC code word =
Category code word + DIFF value code word

15 → 101 1111

Variable Length Coding (AC Huffman Coding)

SSSS	AC coefficients
1	-1,1
2	-3,-2,2,3
3	-7..-4,4..7
4	-15..-8,8,15
5	-13..-16,16..31
6	-127..-64,64..127
….	….

AC coefficient magnitude category

Run/Size	Code length	Code word
0/0 (EOB)	4	1010
0/1	2	00
..	..	..
1/2	5	11011
..	..	..
2/1	5	11100
….	….	….

AC Huffman Table

AC code word = Run/Size code word + AC coefficient code word

<1,-2>	<0,-1>	<0,-1>	<0,-1>	<2,-1>	<EOB>
11011 01	00 0	00 0	00 0	11100 0	1010

Outline

- ❑ How to design an IP with AMBA standard
- ❑ JPEG Spec.
- ❑ *Lab requirement*

Lab requirement

- Try to explain the given dct IP content
- Try to partition the hw/sw part for implement
- Implement the JPEG encoder
- Compare the timing constraint of your own design with the one of reference code

Reference

- □ lm_xcv600e_revc.pdf
- □ DUI146C_LM600_UG.pdf
- □ Code Example supplied by NCTU
- □ JPEG Still Image Specification