1 林致佑

(a) 2M x 16

1. 2^10 = 1024 -> take 2^10 as the binary approximation of kilo- (1000 multiplier)
   2M = 2 * 2^20 = 2^21, so 2M x 16 takes **21 address lines** and **16 input and 16 output lines**
2. Number of bits stored in the memories:
   2M x 16: 2M = 2^21 = 2,097,152 words. Each word is 2 bytes = 16 bits, so the total number of bits is **2^21*16**

(b) 32K x 4

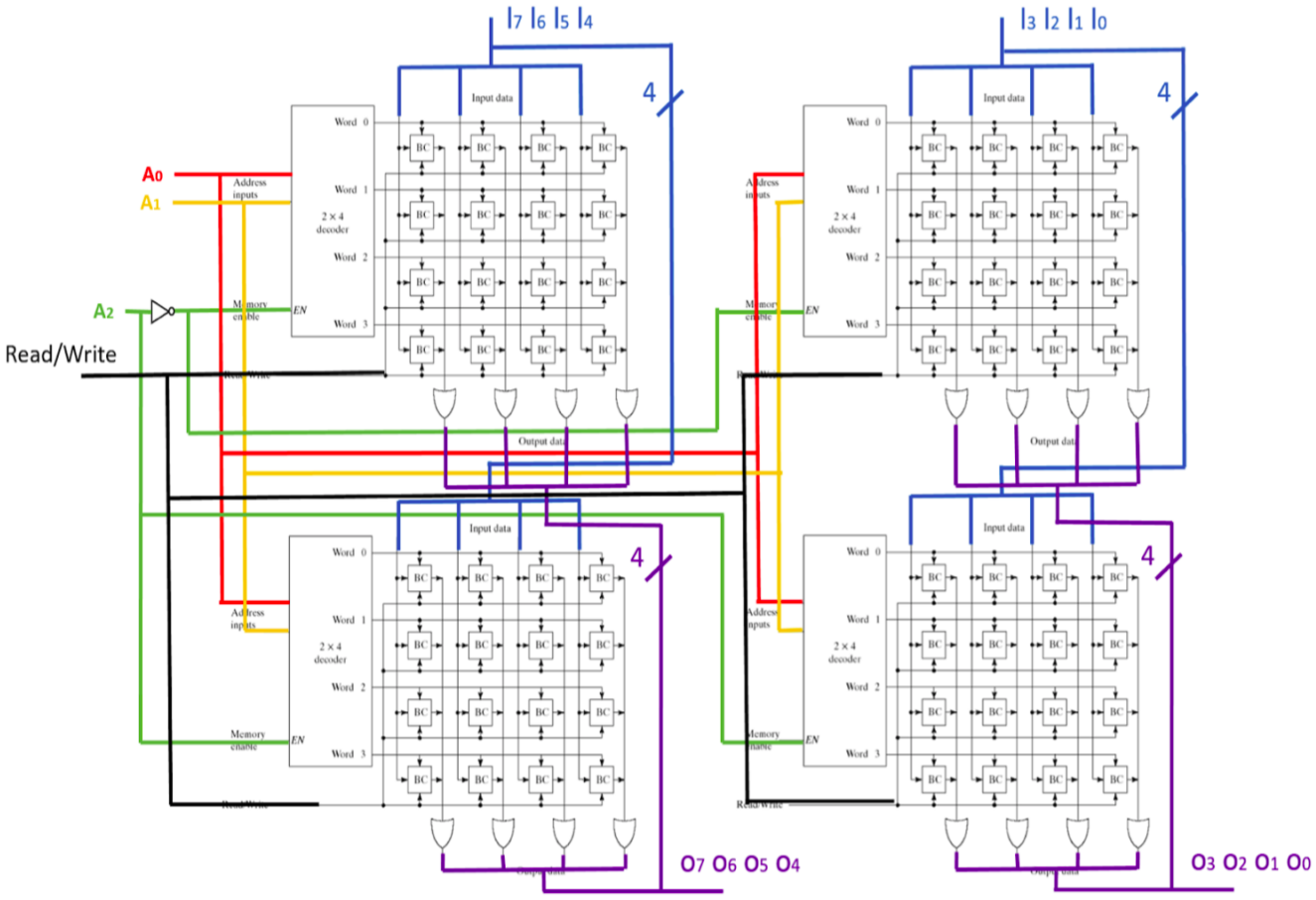
1. 32K = 2^5 * 2^10 = 2^15, so 32K x 4 takes **15 address lines** and **4 input and 4 output lines**
2. Number of bits stored in the memories:
   32K x 4: 32K = 2^15 = 32,768 words. Each word is 4 bits, so the total number of bits is **2^15*4**

2徐浩庭

題目要求為8X8 => I0~I7 為Inputs，O0~O7 為outputs

4X4 RAM為 2-to-4-lines decoder

因此有兩個 Address inputs(A0,A1)，而A2為 Memory enablem用於控制A0及A1，當A2=0時 Address inputs(A0,A1)=0，A2=1時Address inputs(A0,A1)=1。



3呂政和

3. (a) bit position 1 2 3 4 5 6 7 8 9 10 11 12

0 1 1 0 0 1 0 0 0 1 1 0

C_1 = XOR of bits (1, 3, 5, 7, 9, 11)

$= 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 0$

C_2 = XOR of bits (2, 3, 6, 7, 10, 11)

$= 1 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 1$

C_4 = XOR of bits (4, 5, 6, 7, 12)

$= 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 1$

C_8 = XOR of bits (8, 9, 10, 11, 12)

$= 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 0$

$C = C_8 C_4 C_2 C_1 = 0110 = 6 \Rightarrow$ error in bit 6

$\Rightarrow$ 011000000110

$\Rightarrow$ Original data = 10000110

(b)

bit position	1	2	3	4	5	6	7	8	9	10	11	12
	1	0	1	1	1	0	1	1	0	1	0	0

C_1 = XOR of bits (1, 3, 5, 7, 9, 11)

$= 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 0$

C_2 = XOR of bits (2, 3, 6, 7, 10, 11)

$= 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$

C_4 = XOR of bits (4, 5, 6, 7, 12)

$= 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$

C_8 = XOR of bits (8, 9, 10, 11, 12)

$= 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0$

$C = C_8 C_4 C_2 C_1 = 0110 = 6 \Rightarrow$ error in bit 6

$\Rightarrow 101111110100$

Original data = 11110100

4林彦岑

4.

The number of words = 64 k

$= 2^6 \cdot 2^{10}$

$= 2^{16}$

$= 2^8 \cdot 2^8$

(a) $= 256 \cdot 256$

decoder size : **8-to-2^8-line decoder**

for row / column decoder

256 8-input AND gates for row decoder and column decoder

∴ **256x2** 8-input AND gates are required for RAM cell

2^{16} 2-input AND gates are required

(b)

2 | 880
2 | 440 — 0
2 | 220 — 0
2 | 110 — 0
2 | 55 — 0
2 | 27 — 1
2 | 13 — 1
2 | 6 — 1
2 | 3 — 0
1 — 1

$(880)_{10} = (1101110000)_2$

$= (\overbrace{00000011}^{\text{row}}\ \overbrace{01110000}^{\text{column}})_2$

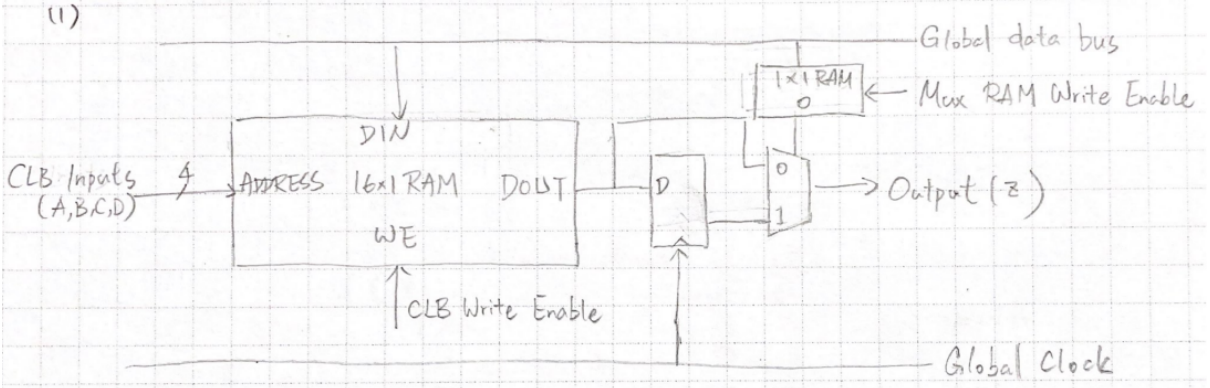
column = $(01110000)_2$

row = $(00000011)_2$

#

5吳東霖

(1)



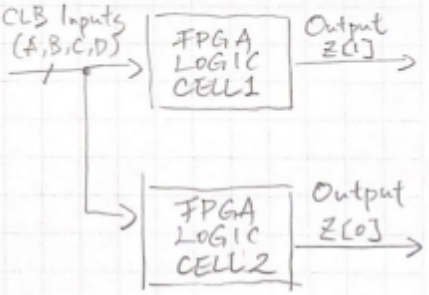
Inputs A, B, C would first send to the "16 × 1 RAM". There is also a lookup table (LUT) inside that RAM to determine the output "DOUT". The LUT (16-1 mux) is shown below.

LOGIC CELL 1

Address	ABCD	carry DOUT(Z[1])
0	000X	0
1	000X	0
2	001X	0
3	001X	0
4	010X	0
5	010X	0
6	011X	1
7	011X	1
8	100X	0
9	100X	0
10	101X	1
11	101X	1
12	110X	1
13	110X	1
14	111X	1
15	111X	1

LOGIC CELL 2

Address	ABCD	Sum DOUT(Z[0])
0	000X	0
1	000X	0
2	001X	1
3	001X	1
4	010X	1
5	010X	1
6	011X	0
7	011X	0
8	100X	1
9	100X	1
10	101X	0
11	101X	0
12	110X	0
13	110X	0
14	111X	1
15	111X	1



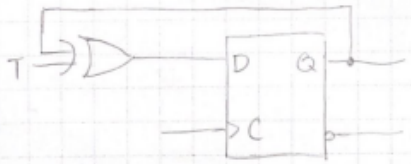
Since we have two functions ($f(A,B,C)=Z[1]$, $f(A,B,C)=Z[0]$), we need two FPGA logic cells for carry and sum respectively.

Full adder is a combinational logic so the "1×1 RAM" would select "0". Finally, we can get the output Z[0] or Z[1].

During the process, "CLB Write Enable" is off.

"CLB Write Enable" =1 only happens if we wanna change the RAM CLB function..

(2)



The LUT of TFF is shown below.

ADDRESS	(T) A	(Q(t)) B	C	D	(Q(t+1)) DOUT
0	0	0	X	X	0
1	0	0	X	X	0
2	0	0	X	X	0
3	0	0	X	X	0
4	0	1	X	X	1
5	0	1	X	X	1
6	0	1	X	X	1
7	0	1	X	X	1
8	1	0	X	X	1
9	1	0	X	X	1
10	1	0	X	X	1
11	1	0	X	X	1
12	1	1	X	X	0
13	1	1	X	X	0
14	1	1	X	X	0
15	1	1	X	X	0

Connect T to CLB input A, Q(t+1) to DOUT, CLB output (Z) to CLB input B.

The whole process is smilar to (1), but this time the "1×1 RAM" is "1" order to create a sequential circuit.

6呂依凡

6.

8×4 ROM $\Rightarrow 2^3 \times 4$ ROM

$A(X,Y,Z) = \sum m(1,3,5) = m_1 + m_3 + m_5$

$D(X,Y,Z) = \sum m(2,3,5,6,7) = m_2 + m_3 + m_5 + m_6 + m_7$

	X	Y	Z	A	D	f_0	f_1
m_0	0	0	0	0	0	0	0
m_1	0	0	1	1	0	0	0
m_2	0	1	0	0	1	0	0
m_3	0	1	1	1	1	0	0
m_4	1	0	0	0	0	0	0
m_5	1	0	1	1	1	0	0
m_6	1	1	0	0	1	0	0
m_7	1	1	1	0	1	0	0
						not	use

登記分數:陳謙謙