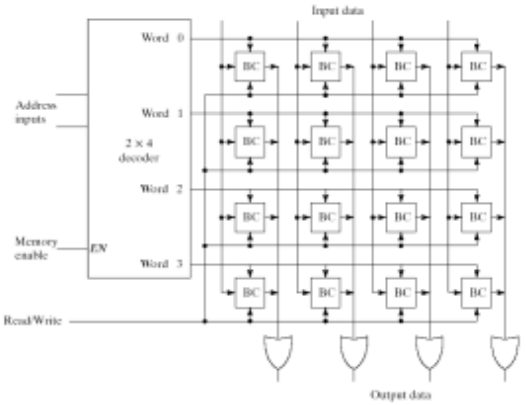
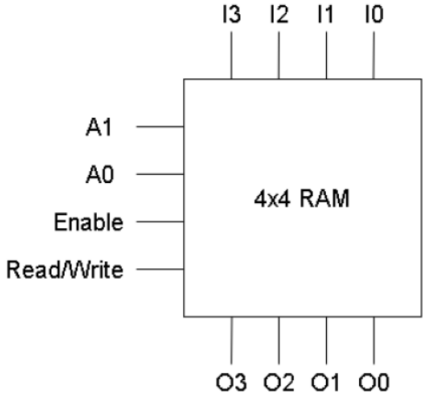
1. (20%) A 4x4 RAM below shows all inputs and outputs. Assuming three-state outputs, construct an 8x8 memory using the four 4x4 RAM units. (Hint: Similar to decoder size extension in decoder with enable input.)

(楊博舜)

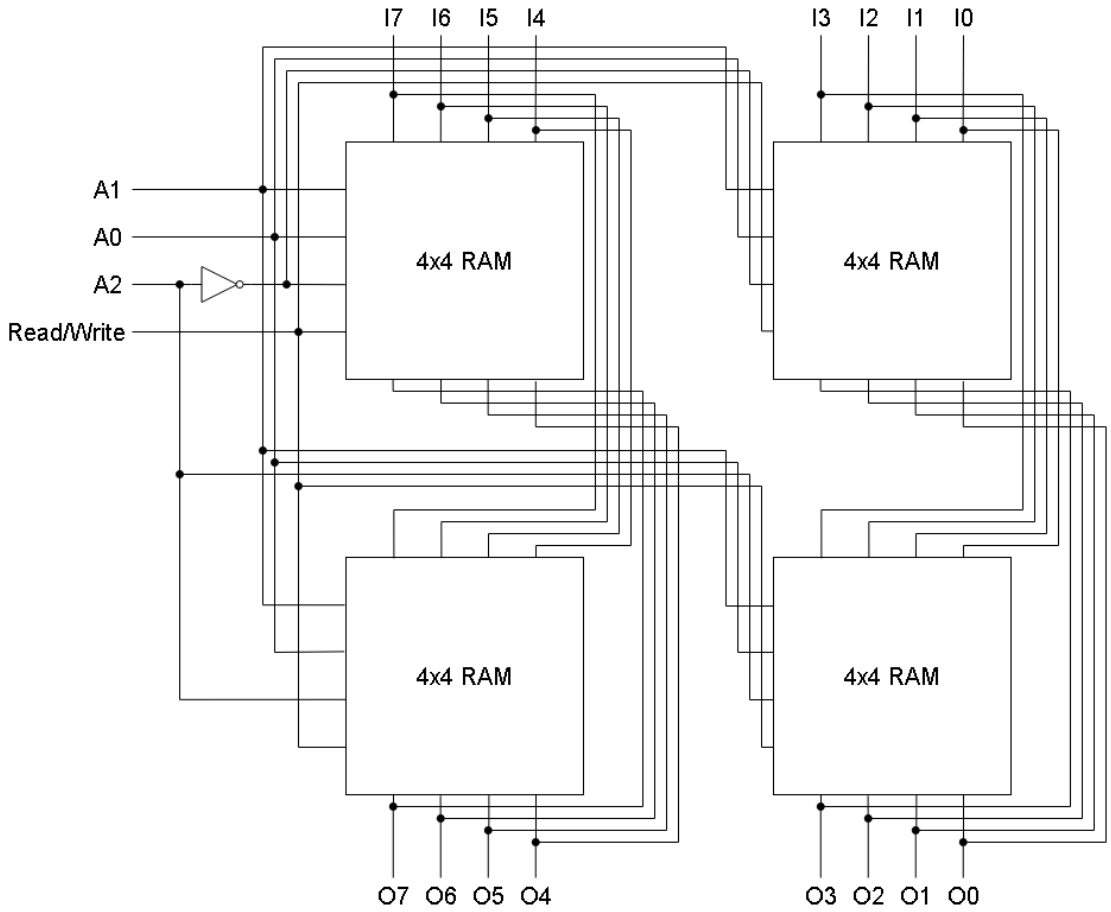


We can express a 4x4 RAM in the following way.



I3, I2, I1, and I0 are 4 inputs. O3, O2, O1, and O0 are 4 outputs. A1 and A0 are address inputs.

For an 8x8 memory, we need 8 inputs, I7 to I0, 8 outputs, O7 to O0, and 3 address inputs, A2 to A0. For 8-bit words, we can just connect I7~I4 to one 4x4 RAM, and I3 ~I0 to another one. For 8 words, since the outputs are three-state, we can connect the outputs of the upper and the lower half directly, and use “Enable” to control the output states. When we want to get the outputs of the upper half, set “Enable” to 1 for the upper half, and set “Enable” to 0 for the lower half. When we want to get the outputs of the lower half, set “Enable” to 0 for the upper half, and set “Enable” to 1 for the lower half. To achieve it, we can connect A2’ to the upper half, and connect A2 to the lower half. The final result if shown in the following figure.



2. (20%) A 12-bit Hamming code word containing 8 bits of data and 4 parity bits is read from the memory. What is the original 8-bit data word that was written into memory if the 12-bit word read out is as follows: (a) 011001000110 (b) 101110110100.

(呂易縉)

(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)

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

$C = C_1C_2C_4C_8 = 0110 = 6$ ➜ error in bit 6

⇨ 011000000110

⇨ 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)

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

$C = C_1C_2C_4C_8 = 0110 = 6$ ➜ error in bit 6

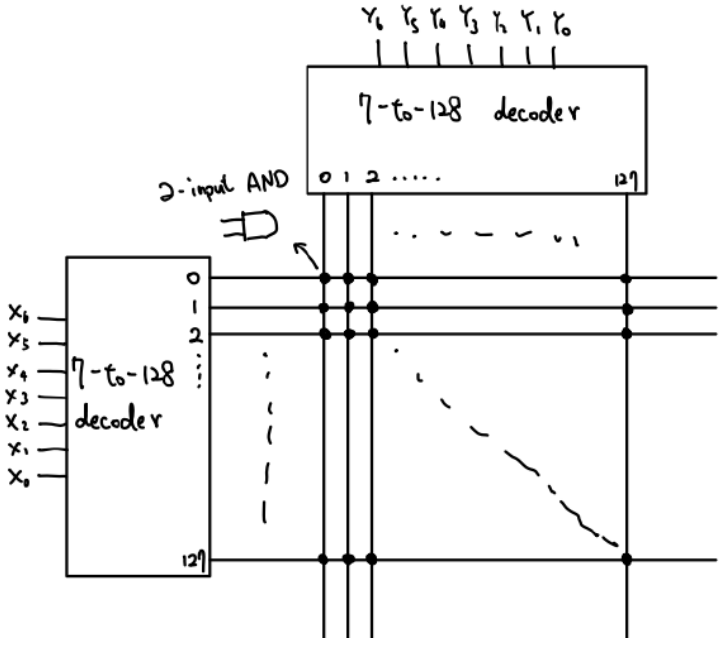
⇨ 101111110100

⇨ Original data = 11110100

3. (14%) A 16K × 6 RAM chip uses coincident decoding by splitting the internal decoder into row select and column select.
   (1) Assuming that the RAM cell array is square, what is the size of each decoder and how many AND gates are required for decoding an address?
   (2) Determine the row and column selection lines that are enabled when the input address is the binary equivalent of $(78)_{10}$.

(林彥岑)

The number of words $= 16\text{k} = 2^4 \times 2^{10} = 2^{14} = 2^7 \times 2^7 = 128 \times 128$



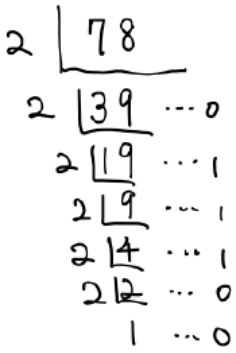
(a)

Each decoder size: $\mathbf{7-to-2^7-line\ decoder}$

There are 128 7-input AND gates in row decoder and column decoder each, and $2^7 \times 2^7$ 2-input AND gates.

Thus, there is a total of **256 7-input AND gates** and **16384 2-input AND gates**.

(b)



$78_{10} = 1001110_2 = 0000000\ \ 1001110_2$

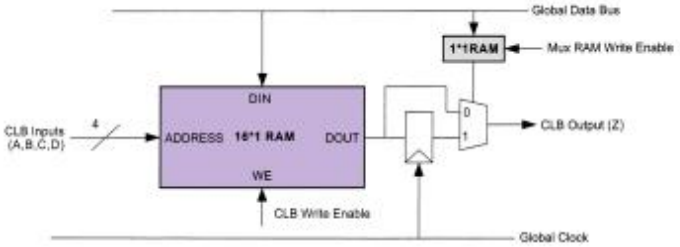
Row selection lines = $\mathbf{0000000_2}$

Column selection lines = $\mathbf{1001110_2}$

4. (30%) A simple FPGA logic cell is shown as figure below:

(1) (10%) Explain how the logic cell can implement a 1-bit JK flip-flop function.

(2) (20%) How to use the logic cell as building blocks to implement a 3-bit binary up counter?



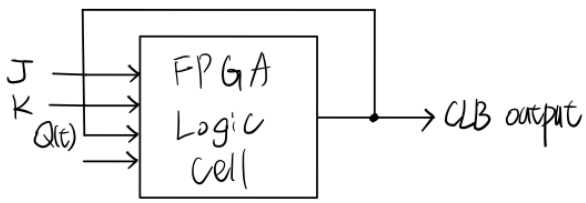
(徐浩庭)

(1)

$Q(t+1) = J(t)\,Q'(t) + K'(t)\,Q(t)$

(Don't care)

J	K	Q	X	
A	B	C	D	DOUT
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

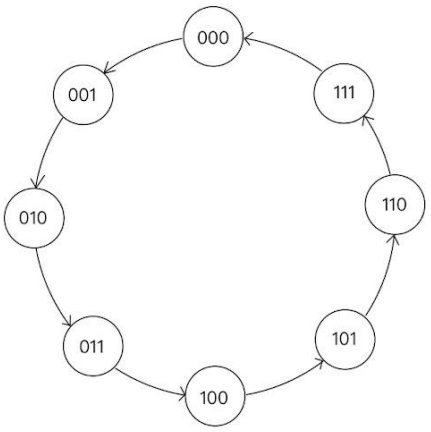


Connect J to CLB input A, Q(t+1) to DOUT, K to CLB input B, CLB output (z) to CLB input C, and ‘‘1x1 RAM’’ is ‘‘1’’ order to create a sequential circuit.

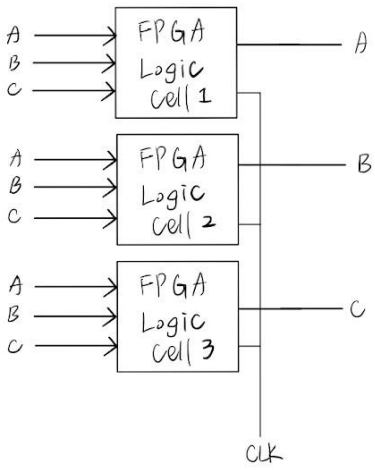
(2)

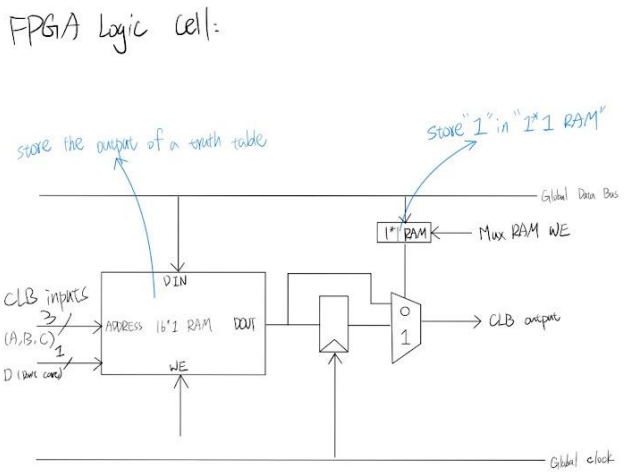
(2)

State Diagram



Present state				Next state		
A(t)	B(t)	C(t)	D (Don't care) X	A(t+1)	B(t+1)	C(t+1)
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	0	1	1
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	1	0	1
1	0	0	1	1	0	1
1	0	1	0	1	1	0
1	0	1	1	1	1	0
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	0	0	0
1	1	1	1	0	0	0





A,B,C are 3 different output from logic cell and we stored these truth tables into the 16*1 RAM separately. When ‘‘1x1 RAM’’ is ‘‘1’’, we can get 3 bit up counter.

5. (16%) Tabulate the truth table for an 8x4 ROM that implements the Boolean functions.

(1) A(X, Y, Z) = Σm(2, 4, 5)
(2) D(X, Y, Z) = Σm(1, 4, 6, 7)

(林致佑)

X	Y	Z	A	D	F0	F1
0	0	0	0	0	X	X
0	0	1	0	1	X	X
0	1	0	1	0	X	X
0	1	1	0	0	X	X
1	0	0	1	1	X	X
1	0	1	1	0	X	X
1	1	0	0	1	X	X
1	1	1	0	1	X	X