

# Design of Sequential Circuit using Registers & ROM

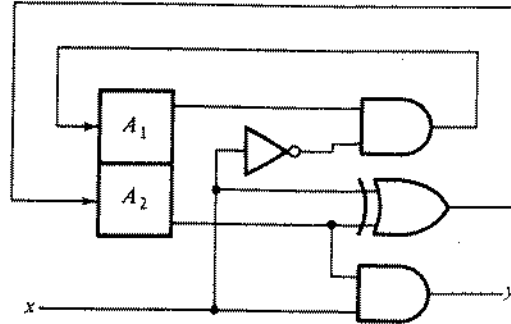
$$A_1(t+1) = \Sigma(4, 6)$$

$$A_2(t+1) = \Sigma(1, 2, 5, 6)$$

$$y(A_1, A_2, x) = \Sigma(3, 7)$$

| Present state |       | Input |  | Next state |       | Output |
|---------------|-------|-------|--|------------|-------|--------|
| $A_1$         | $A_2$ | $x$   |  | $A_1$      | $A_2$ | $y$    |
| 0             | 0     | 0     |  | 0          | 0     | 0      |
| 0             | 0     | 1     |  | 0          | 1     | 0      |
| 0             | 1     | 0     |  | 0          | 1     | 0      |
| 0             | 1     | 1     |  | 0          | 0     | 1      |
| 1             | 0     | 0     |  | 1          | 0     | 0      |
| 1             | 0     | 1     |  | 0          | 1     | 0      |
| 1             | 1     | 0     |  | 1          | 1     | 0      |
| 1             | 1     | 1     |  | 0          | 0     | 1      |

(a) State table



(b) Logic diagram

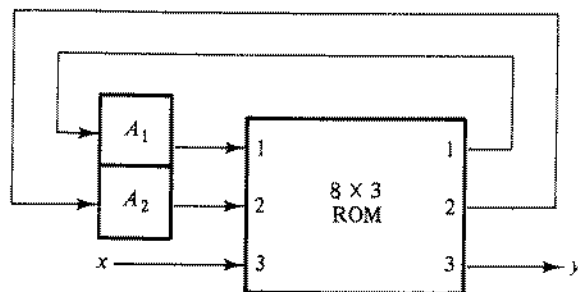
$$A_1(t+1) = A_1x'$$

$$A_2(t+1) = A_2 \oplus x$$

$$y = A_2x$$

ROM truth table

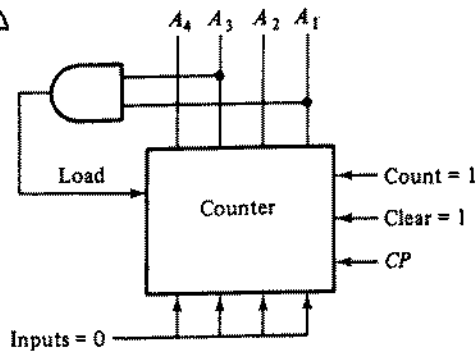
| Address |   |   | Outputs |   |   |
|---------|---|---|---------|---|---|
| 1       | 2 | 3 | 1       | 2 | 3 |
| 0       | 0 | 0 | 0       | 0 | 0 |
| 0       | 0 | 1 | 0       | 1 | 0 |
| 0       | 1 | 0 | 0       | 1 | 0 |
| 0       | 1 | 1 | 0       | 0 | 1 |
| 1       | 0 | 0 | 1       | 0 | 0 |
| 1       | 0 | 1 | 0       | 1 | 0 |
| 1       | 1 | 0 | 1       | 1 | 0 |
| 1       | 1 | 1 | 0       | 0 | 1 |



### Example

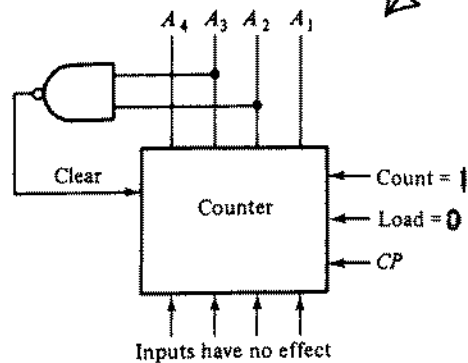
Modulo 6 Counter with parallel load which can be used to generate a sequence of six counts.

Detects 0101,  
load input enabled  
all 0's input is loaded

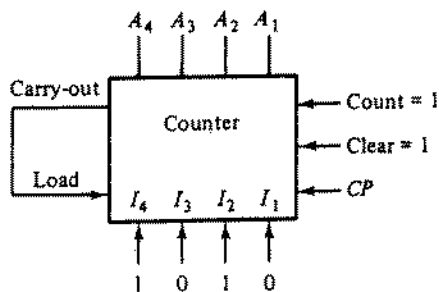


(a) Binary states 0, 1, 2, 3, 4, 5.

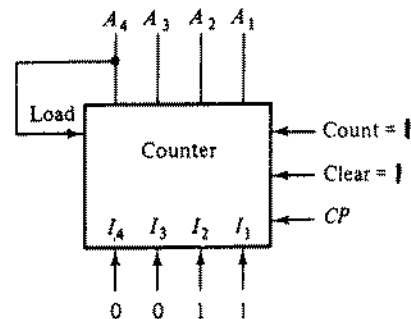
Detects 0110.  
As soon as 0110 occurs,  
counter is cleared.



(b) Binary states 0, 1, 2, 3, 4, 5.



(c) Binary states 10, 11, 12, 13, 14, 15.



(d) Binary states 3, 4, 5, 6, 7, 8.

Generates count sequence  
10, 11, 12, 13, 14, 15.  
When carry is generated  
after 1111,  
counter is loaded with  
1010

Generates count sequence  
3, 4, 5, 6, 7, 8.  
When 1000 is detected,  
0011 is loaded