**Question 1**

N =

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | 0 |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

HA

FA

FA

FA

HA

HA

Area = 3 Half Adders + 3 Full Adders + AND Gate

Delay = 3 + 3 + AND Gate delayImplementation of 1011:

1

HA

FA

FA

FA

HA

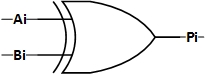
HA

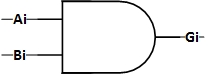
P = =

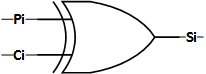
Check

**Question 2**

Let

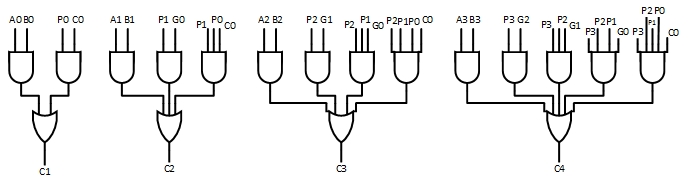




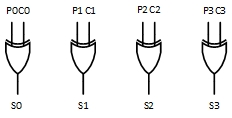


and

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

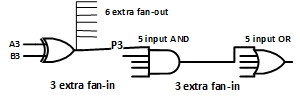


**Carry Output**



**Sum Output**

Delay associated with is the highest. The path is shown below.



**Critical path**

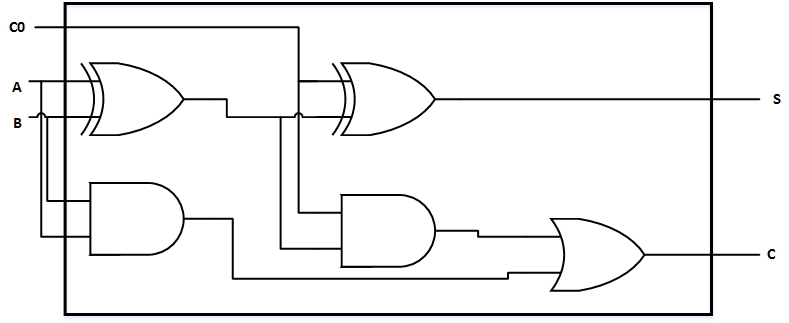
In comparison, the 4-bit Carry Ripple Adder:

FA

FA

FA

FA

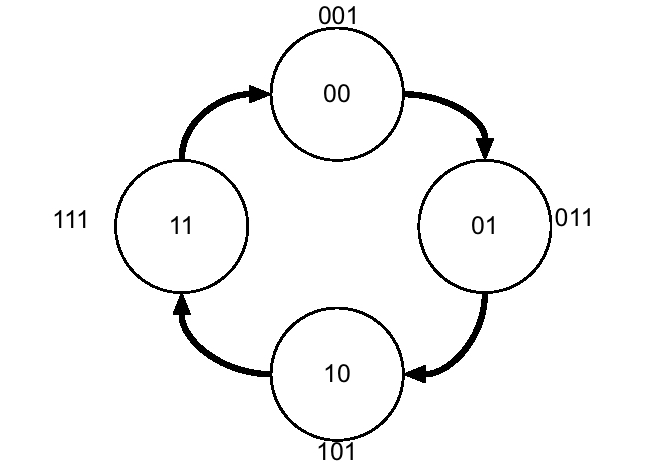


**Question 3**

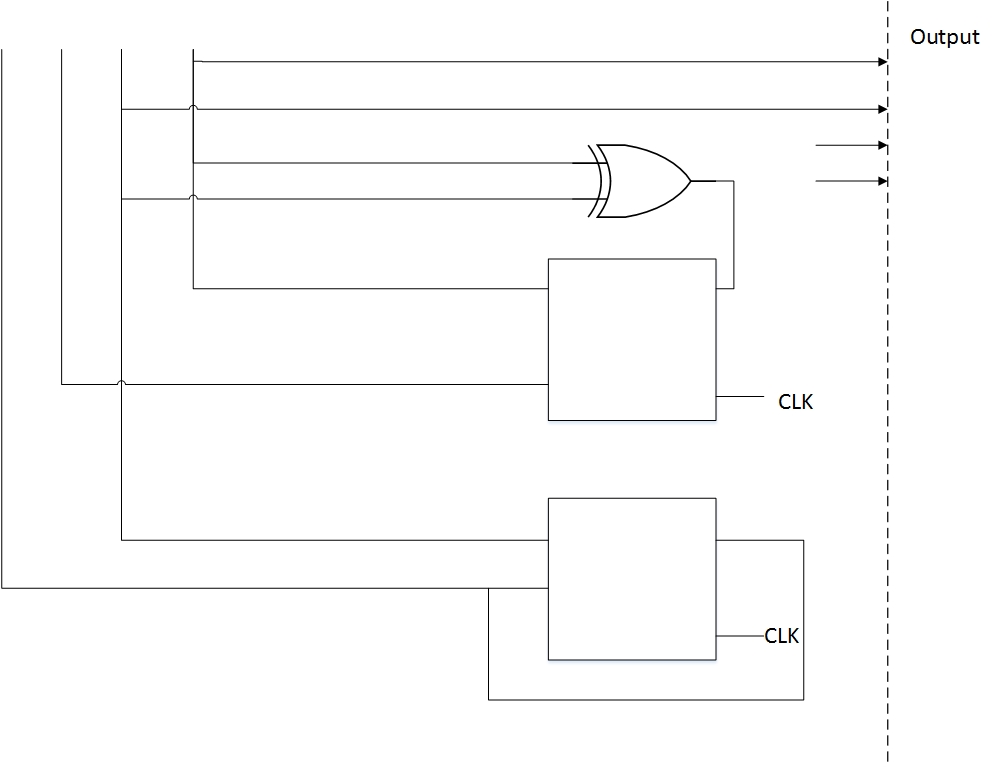
You may design a 8-state FSM with don’t cares or go for a 4-state FSM with decoder for the output which is the simplest as we follow:

State diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| State | | Next State | | Output | | |
|  |  |  |  |  |  |  |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 |



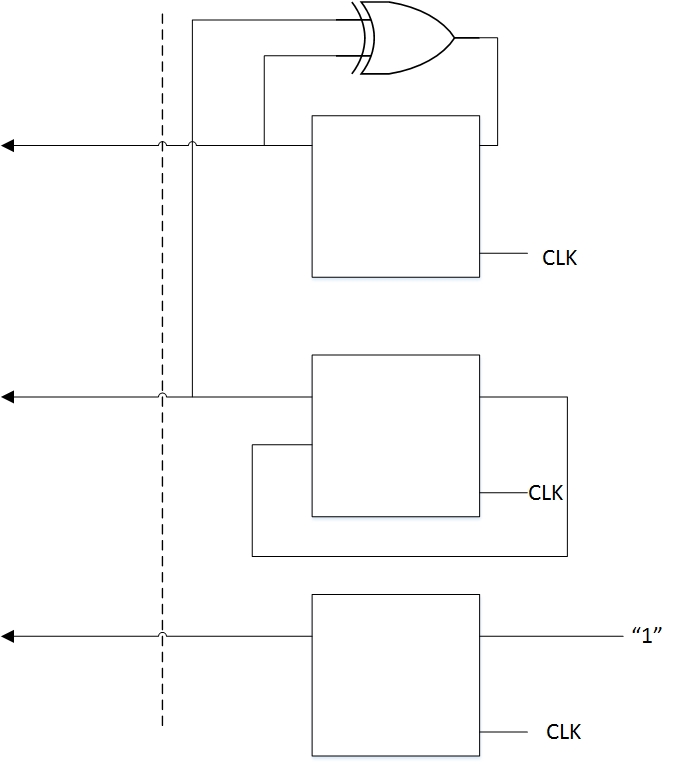
From state table directly the next states and the outputs can be read as:



Alternatively you might want to design an 8 bit FSM with don’t care states. A ,….. states 000, 010, 100, 110 never happen.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 0 | 0 | 0 |  |  |  |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 |  |  |  |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 |  |  |  |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 |  |  |  |
| 1 | 1 | 1 | 0 | 0 | 1 |

Giving



uts = states