→ real medical condition called "Anterograde Amnesia"

(*) Movie memento.

A man, suffering from short-term memory loss, uses notes & tattoos to hunt down his wife's killer.

CpE111 Introduction to Computer Engineering

Dr. Minsu Choi

CH 9: Sequential Logic Networks

UMR UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.

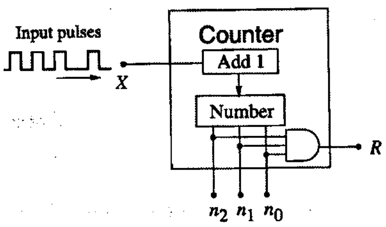
Sequential Logic Networks?

- A sequential logic network is a digital system where the output is determined by both the present input and the result of a previous event.
- Ex) 3-bit counter

So, SLN has ability to memorize the previous state.

If count is in range of 0~6, then R=0

If " 7, then output will reset to 0 on next pulse.

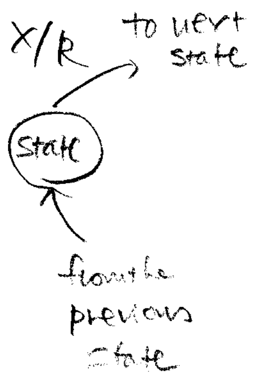


→ R=1 } reset

=) 000 → 001 → . . . → 111 → 000

X= Input that Causes transition

R= output

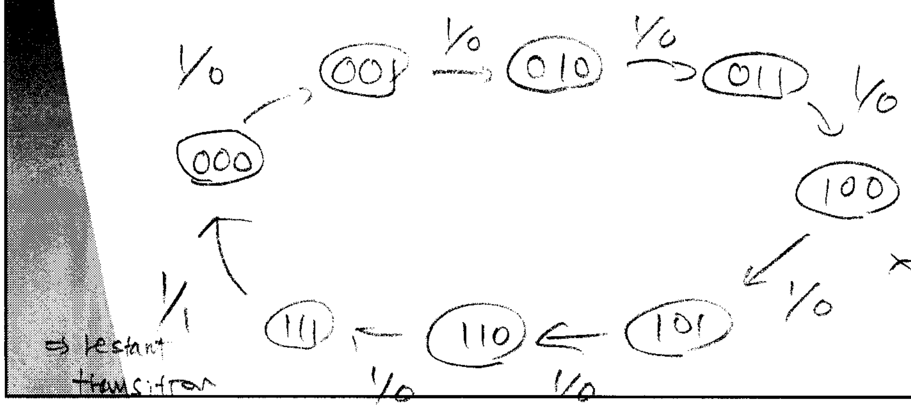


=) symbolic representation of a state

Formal Description of the System: State Diagram

- We define each possible value of the word as a distinct state of the machine.

ex) 3-bit counter



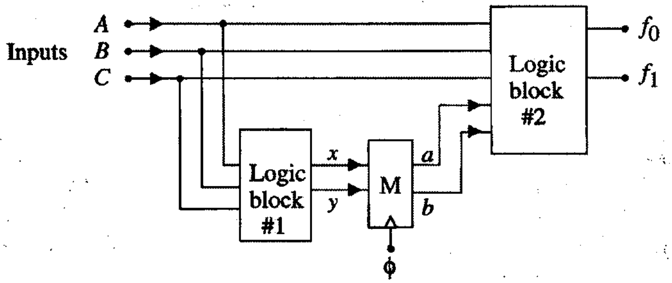
← State diagram.

Sequential Network Requirements

- A combinational logic network to perform Boolean operations.
- A memory element that stores the result of an earlier event.
- If we require a clock signal to synchronize the events, it is called "synchronous logic network", also.
- Clocking signal convention: T = 1 is assumed.
- (t-1) to t called previous cycle, t to (t+1) called present cycle and (t+1) to (t+2) called next clock cycle.

→ We now assume that 1 is the unit clock period

Ex) General Sequential Network #1



- Two combinational logic blocks #1 and #2 and a state element (a clocked memory) M.

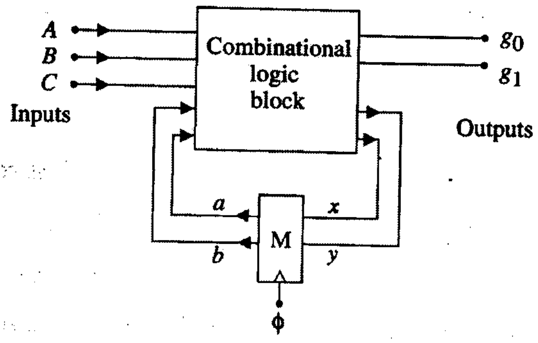
$$a(t) = x(t-1)$$

$$b(t) = y(t-1)$$

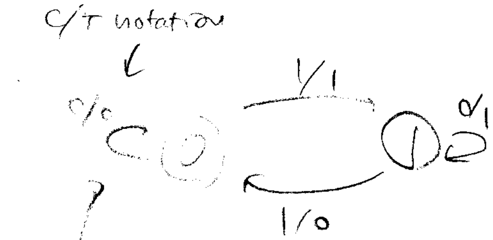
$$\Rightarrow f_0 = f_0(\overbrace{A,B,C}^{\text{current input}}, \overbrace{a,b}^{\text{previous state}})$$

$$f_1 = f_1(A,B,C,a,b)$$

Ex) General Sequential Network #2



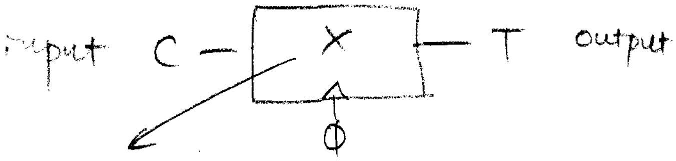
- 1 combinational logic block & 1 memory element and feedback wires.



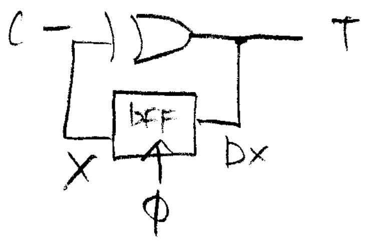
⇒ C acts as a control bit.
(if C=0 then No state change
if C=1 " state change.

Single State Variable Networks

- Ex1)



T(t) = C(t) ⊕ X(t)



⇒ circuit diagram.

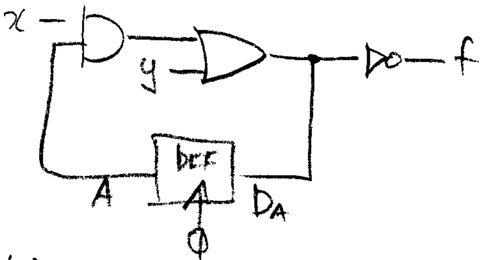
present value		output	Next value
C(t)	X(t)	T(t)	X(t+1)
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

X(t+1) = Dx(t) = T(t)
←T(t) is mean X(t+1)

⇒ state table.

Continued,

- Ex2)



⇒ $f = \overline{x \cdot A + y}$

$D_A = \bar{f}$

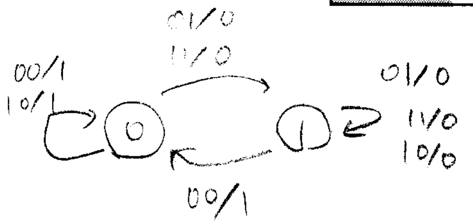
$= x \cdot A + y$.

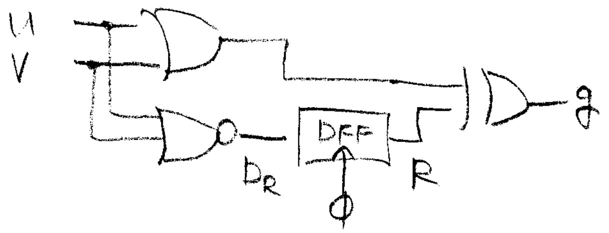
$A(t+1) = D_A(t)$

state table)

present state	present inputs		present output	Next state
A(t)	x(t)	y(t)	f(t)	A(t+1)
0	0	0	1	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

state diagram.





$\Rightarrow g = (u+v) \oplus R$

$D_R(t) = \overline{u(t) + v(t)}$

$= R(t+1)$

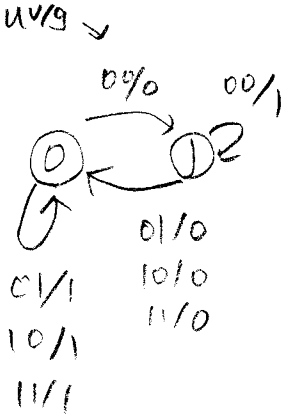
Continued,

state table

- Ex3)

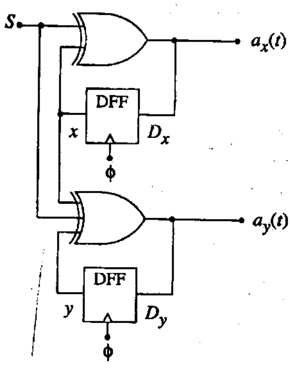
	present values			next state
R(t)	u(t)	v(t)	g(t)	R(t+1)
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

state diagram.



Multi-State Variable Networks

- If we have n state variables, then 2^n states are needed when we draw a state diagram.
- Ex)



$a_x(t) = S(t) \oplus x(t$

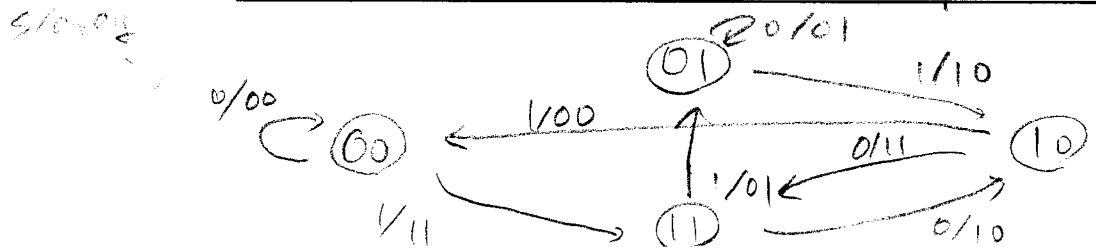
$a_y(t) = S(t) \oplus x(t)$
$\oplus\, y(t)$

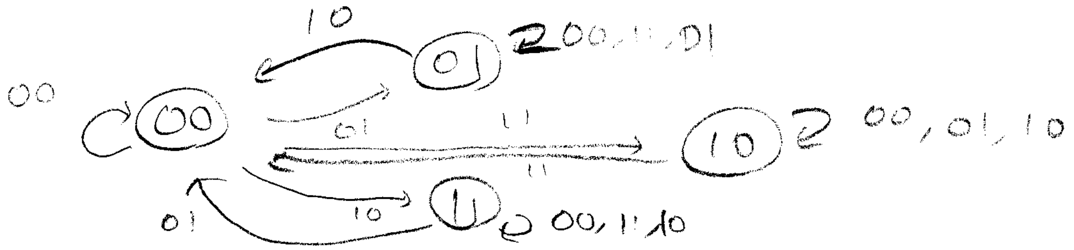
&

$x(t+1) = D_x(t) = a_x($

$y(t+1) = D_y(t) = a_y(t$

present values					next states	
s(t)	x(t)	y(t)	$a_x(t)$	$a_y(t)$	x(t+1)	y(t+1)
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	1	1	1	1
0	1	1	1	0	1	0
1	0	0	1	1	1	1
1	0	1	1	0	1	0
1	1	0	0	0	0	0
1	1	1	0	1	0	1





State Diagram to State Table

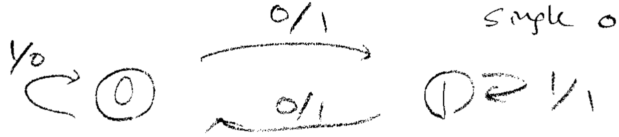
⇒ For each state, consider outgoing transitions.

Inputs	present state	Next state
X Y	A B	A B
0 0	0 0	0 0
0 1	0 0	0 1
1 0	0 0	1 1
1 1	0 0	1 0
	⋮	

You need to consider 4 cases since you have 2 input bits.

Sequential Network Design

- We start with a listing of the desired transition characteristics (state table, etc) and use the information to create the network. → single input var x(t)
single output var f(t)
single state variable A(t).
- Ex)



present values			Next	FF in
A(t)	x(t)	f(t)	A(t+1)	DA(t)
0	0	1	1	1
0	1	0	0	0
1	0	1	0	0
1	1	1	1	1

→ current FF input

Continued,

K-map to find f & DA

f

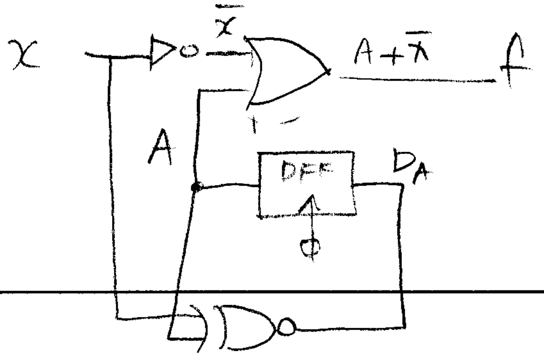
A \ x	0	1
0	1	0
1	1	1

$f = A + \bar{x}$

DA

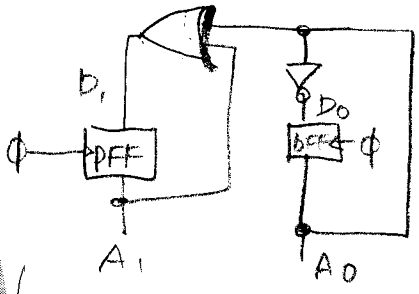
A \ x	0	1
0	1	0
1	0	1

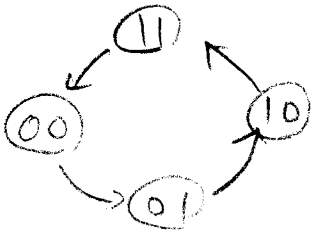
$D_A = \bar{A}\bar{x} + Ax = \overline{A \oplus x}$



Binary Counters

- Ex) 2-bit counter

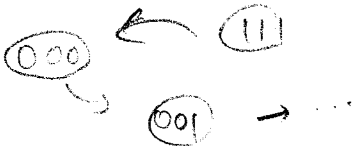




$D_0 = \bar{A_0}$, $D_1 = A_1 \oplus A_0$

$A_1(t+1) = D_1(t)$, $A_0(t+1) = D_0(t)$

present		Next		Inputs to FFs	
A_1	A_0	A_1	A_0	D_1	D_0
0	0	0	1	0	1
0	1	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	0



∈ 8 States.

⇒ we need three FFs

Three inputs : $D_2\ D_1\ D_0$
" outputs : $A_2\ A_1\ A_0$

⇒ $A_2(t+1) = D_2(t)$
$A_1(t+1) = D_1(t)$
$A_0(t+1) = D_0(t)$

Continued,

- Ex) 3-bit counter

present $A_2 A_1 A_0$	Next $A_2 A_1 A_0$	Inputs $D_2 D_1 D_0$
0 0 0	0 0 1	
0 0 1	0 1 0	
0 1 0	0 1 1	→ same
0 1 1	1 0 0	
1 0 0	1 0 1	
1 0 1	1 1 0	
1 1 0	1 1 1	
1 1 1	0 0 0	

K-map :

A_2 \ A_1A_0	00	01	11	10
0			1	
1	1	1		1

D_2 map

A_2 \ A_1A_0	00	01	11	10
0		1		1
1		1		1

D_1 map

A_2 \ A_1A_0	00	01	11	10
0	1			1
1	1			1

D_0 map.

Continued,

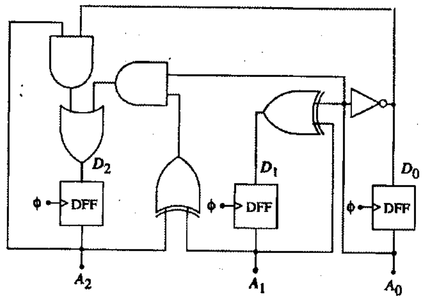
$D_2 = A_2\bar{A_0} + A_2\bar{A_1}A_0 + \bar{A_2}A_1A_0$

$= A_2\bar{A_0} + A_0(A_2 \oplus A_1)$

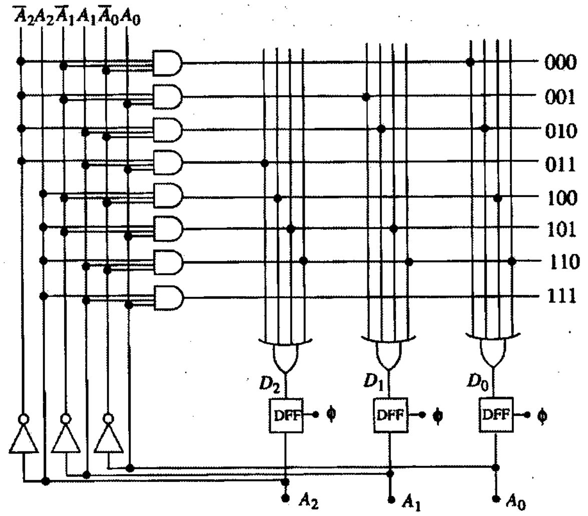
$D_1 = \bar{A_1}A_0 + A_1\bar{A_0} = A_1 \oplus A_0$.

$D_0 = \bar{A_0}$

Design



PLA Implementation of 3-Bit Counter (using SOP forms)



$D_2 = \sum m(3,4,5,6$

$D_1 = \sum m(1 2 5 6)$

$D_0 = \sum m(0,2,4,6)$

Program Completed

University of Missouri-Rolla

© 2003 Curators of University of Missouri

UMR UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.