

Advanced Digital Circuit Design - Asynchronous Sequential Digital Circuit Design

Assoc. Prof. Dr. Berna Örs Yalçın

Istanbul Technical University
Faculty of Electrical and Electronics Engineering
Department of Electronics and Communication
Engineering
siddika.ors@itu.edu.tr

Asynch. vs. Synch.

- Asynchronous circuits don't use clock pulses
 - state transitions by changes in inputs.
- Storage Elements:
 - Clockless storage elements or
 - Delay elements.
- In many cases, as combinational feedback.
 - → Normally much harder to design.

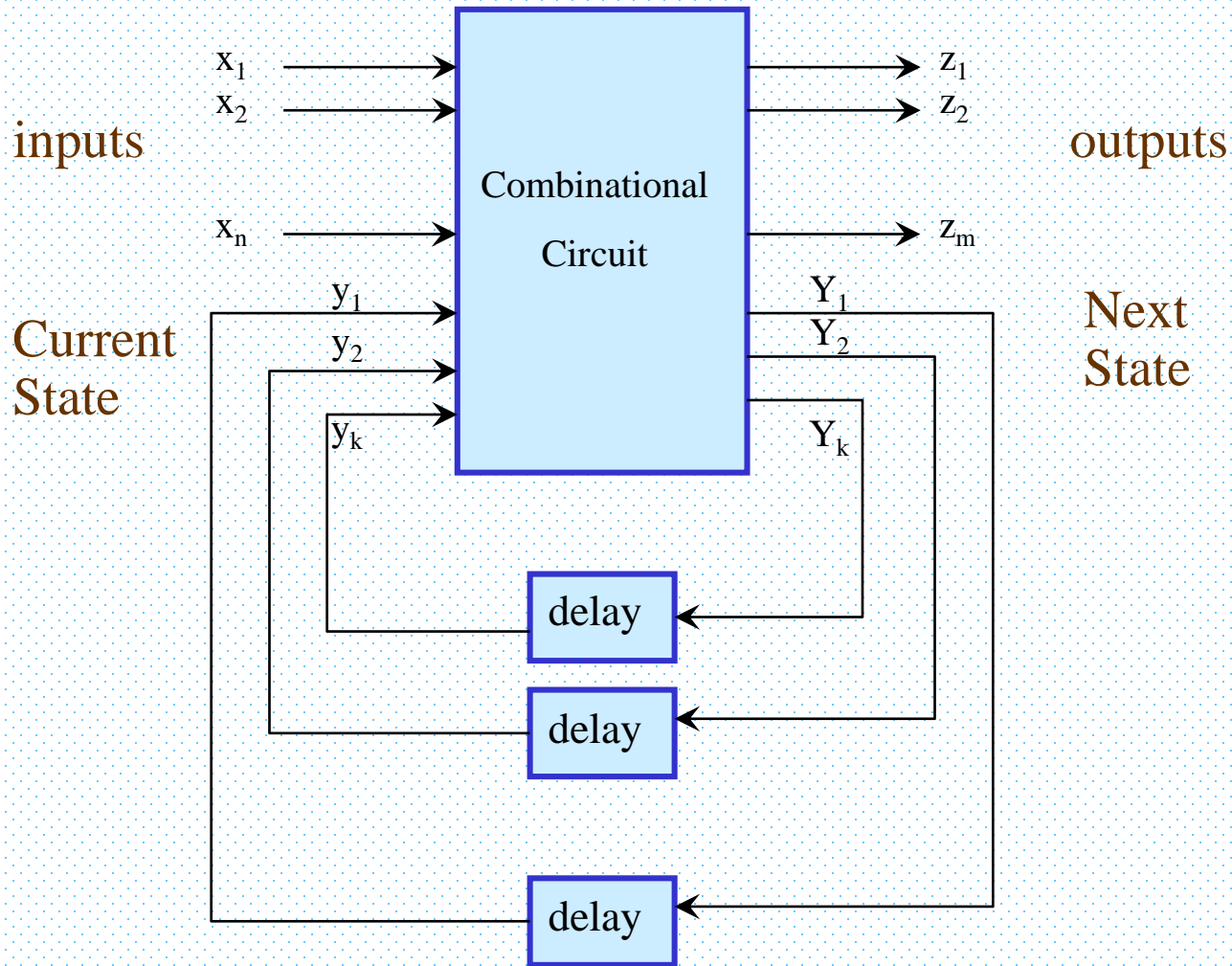
Asynchronous Sequential Circuit

- In a gate-type circuit, the propagation delay that exists in the combinational circuit path from input to output provides sufficient delay along the feedback loop so that no specific delay elements are actually inserted in the feedback path
- **difficult to design:** Timing problems involved in the feedback path
- **must attain a stable state** before the input is changed to a new value

Asynchronous Sequential Circuit

- Because of delays in the wires and the gates, it is impossible to have two or more input variables change at exactly the same instant of time without an uncertainty as to which one changes first.
- Therefore, simultaneous changes of two or more variables are usually prohibited.
- This restriction means that only **one input variable can change at any one time** and the time between two input changes must be longer than the time it takes the circuit to reach a stable state.

Asynch. Sequential Circuit



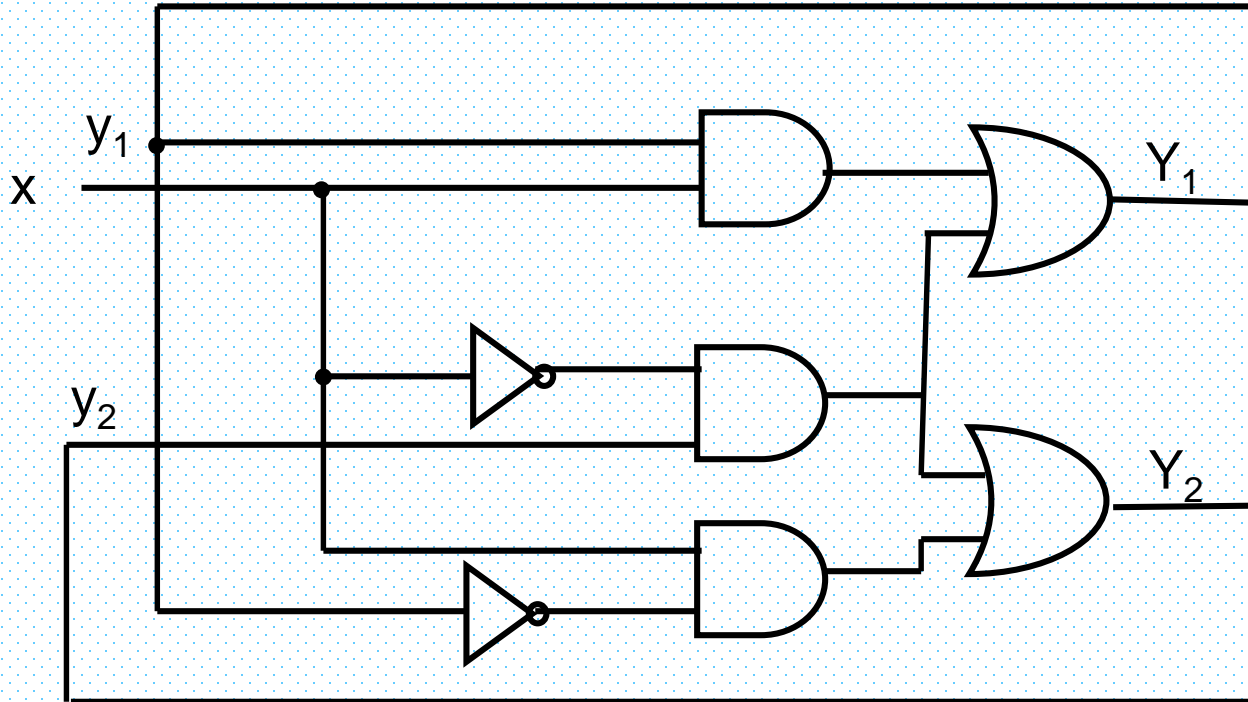
Asynch. Sequential Circuit

- $y_i = Y_i$ in steady state (but may be different during transition)
- Simultaneous change in two (or more) inputs is prohibited.
 - The time between two changes must be less than the time of stability.

Advantages and Disadvantages

- Advantages:
 - Low power
 - High performance
 - No need for clock
- Disadvantages:
 - Complexity of design process.

Analysis



1. Find feedback loops and name feedback variables appropriately.
2. Find boolean expressions of Y_i 's in terms of y_i 's and inputs.

$$Y_1 = x.y_1 + x'.y_2$$
$$Y_2 = x.y_1' + x'.y_2$$

Analysis

3. Draw a map:

- rows: y_i 's
- columns: inputs
- entries: Y_i 's

	X	
$y_1 y_2$	0	1
00	0	0
01	1	0
11	1	1
10	0	1

$$Y_1 = x.y_1 + x'.y_2$$

	X	
$y_1 y_2$	0	1
00	0	1
01	1	1
11	1	0
10	0	0

$$Y_2 = x.y_1' + x'.y_2$$

	X	
$y_1 y_2$	0	1
00	00	01
01	11	01
11	11	10
10	00	10

(Transition Table) $Y_1 Y_2$

Analysis

4. To have a stable state, Y must be $= y$ (circled)

(Transition Table) $Y_1 Y_2$

$y_1 y_2$		x	
		0	1
00	00	01	
01	11	01	
11	11	10	
10	00	10	

- At $y_1 y_2 x = 000$, if $x: 0 \rightarrow 1$
 - then $Y_1 Y_2: 00 \rightarrow 01$
 - then $y_1 y_2 = 01$ (2nd row): stable.

Analysis

- In general, if an input takes the circuit to an unstable state, y_i 's change until a stable state is found.

- **General state of circuit:**

- $y_1 y_2 X$:
 - There are 4 stable states:
 - 000, 011, 110, 101
 - and 4 unstable states.

		X	
		0	1
$y_1 y_2$	00	00	01
	01	11	01
	11	11	10
	10	00	10

State Table

- As synchronous:

present state	next state	
	$X=0$	$X=1$
00	00	01
01	11	01
11	00	10
10	11	10

Flow Table

- As Transition Table (but with symbolic states):

	X	
	0	1
a	a	b
b	c	b
c	c	d
d	a	d

Flow Table: Example 2

- Two states, two inputs, one output.

	$x_1 x_2$			
	00	01	11	10
a	a ,0	a ,0	a ,0	b ,0
b	a ,0	a ,0	b ,1	b ,0

- Each row has more than one stable state.
- If $x_1 = 0$, state is a.
- If $x_1 x_2 = 00 \rightarrow x_1 x_2 = 10$, then state becomes b.
- For $x_1 x_2 = 11$, state is either a or b.
 - If previously in $x_1 x_2 = 01$, keeps state a,
 - If previously in $x_1 x_2 = 10$, keeps state b.
 - Reminder: cannot go from 00 to 11.

Circuit Design

- From flow table to circuit:

- Assign a unique binary value to each state,

		$x_1 x_2$			
		00	01	11	10
a		a,0	a,0	a,0	b,0
b		a,0	a,0	b,1	b,0

		$x_1 x_2$			
		00	01	11	10
y	0	0,0	0,0	0,0	1,0
	1	0,0	0,0	1,1	1,0

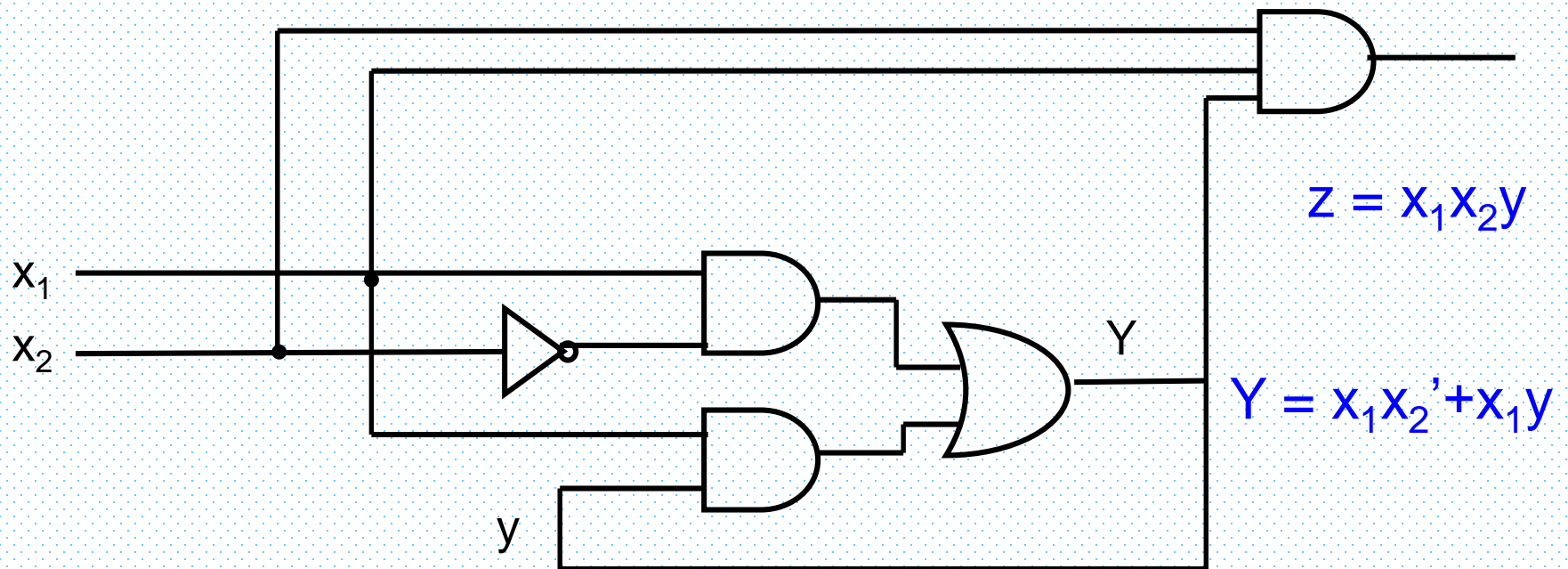
		$x_1 x_2$			
		00	01	11	10
y	0	0	0	0	0
	1	0	0	1	0

		$x_1 x_2$			
		00	01	11	10
y	0	0	0	0	1
	1	0	0	1	1

Map for output $z (=x_1x_2y)$

Map for output $Y (=x_1x_2'+x_1y)$

Circuit Diagram



Race Condition

- If two (or more) state variables change in response to a change in an input, there is a race condition.
 - E.g. from 00 to 11, due to delays
00 → 01 → 11 OR
00 → 10 → 11.

- **Critical Race:**

- If final steady state depends on the order of changes in state vars.

Race: Examples

- Noncritical Cases:

X

y_1y_2	0	1
00	00	11
01		11
11		11
10		11

X

y_1y_2	0	1
00	00	11
01		01
11		01
10		11

$00 \rightarrow 11$
 $00 \rightarrow 01 \rightarrow 11$
 $00 \rightarrow 10 \rightarrow 11$

$00 \rightarrow 01$
 $00 \rightarrow 11 \rightarrow 01$
 $00 \rightarrow 10 \rightarrow 11 \rightarrow 01$

Race: Examples

- Critical Cases:

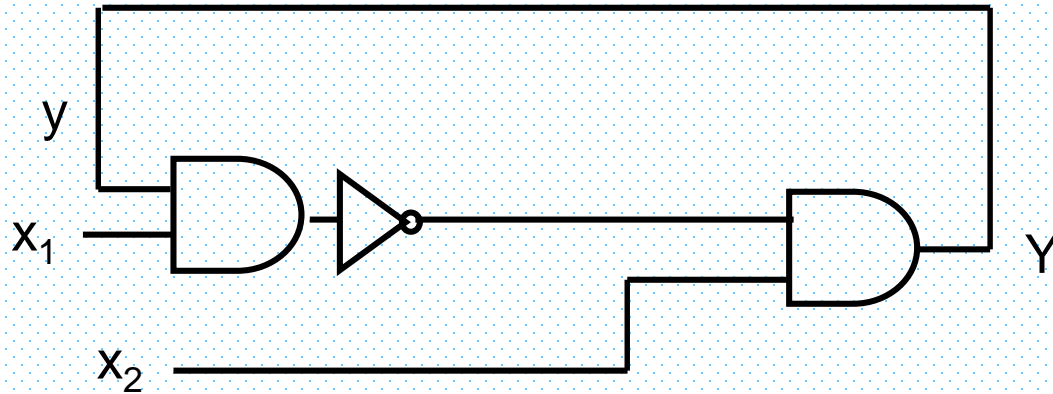
		X	
		0	1
y ₁ y ₂	00	00	11
	01		11
	11		11
	10		10

		X	
		0	1
y ₁ y ₂	00	00	11
	01		01
	11		11
	10		10

00 → 11
 00 → 01 → 11
 00 → 10

00 → 11
 00 → 01
 00 → 10

Instability



		$x_1 x_2$			
		00	01	11	10
y	0	0	1	1	0
	1	0	1	0	0

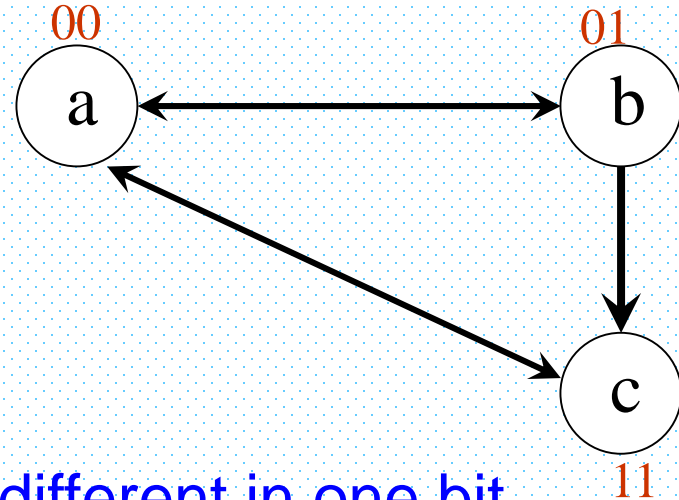
- For $x_1 x_2 = 11$, there is no steady state.
 - → Oscillation.
- For $x_1 x_2 = 11$, $Y = y' \rightarrow$ unstable.

No-Race State Assignment

- Must assign binary values to states such that:
 - one change in an input may not cause two changes in state variables.
 - (because due to delays, one of the variable change sooner and may stay in an unwanted stable state).
 - From a, if $x_1x_2 = 10 \rightarrow 11$, must go to c and stay there.
 - But by the following assignment, it may go to b and stay there.

	$x_1 x_2$			
	00	01	11	10
a	a	b	c	a
b	a	b	b	c
c	a	c	c	c

←



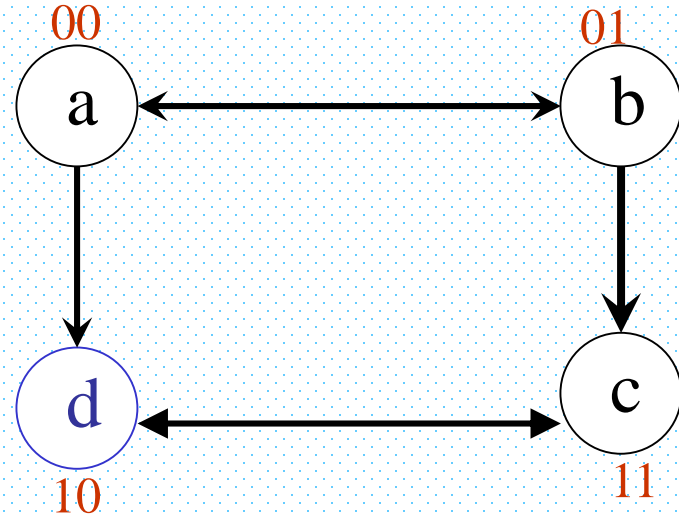
- a and b must be different in one bit,
- a and c must be different in one bit.

No-Race State Assignment

- Impossible \rightarrow add one more row.

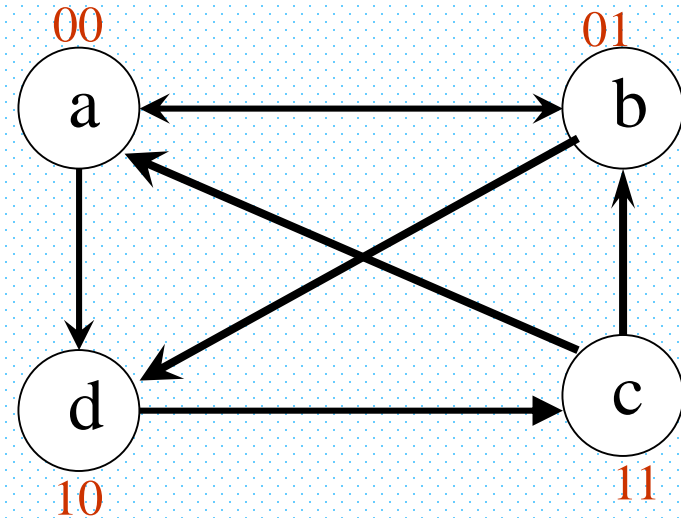
	$x_1 x_2$			
	00	01	11	10
a	a	b	d	a
b	a	b	b	c
c	d	c	c	c
d	a	-	c	-

- \rightarrow d is an intermediate (unstable) state.
- \rightarrow - means any value can be assigned (Except d=10).



Example 2

	x_1		x_2	
	00	01	11	10
a	b	a	d	a
b	b	d	b	a
c	c	a	b	c
d	c	d	d	c



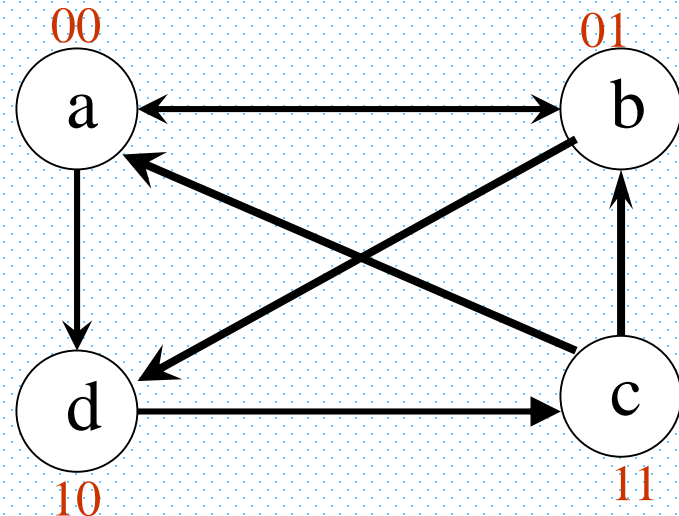
- If there were no diagonal transition, it would be possible
- Impossible → add some more rows.

Example 2

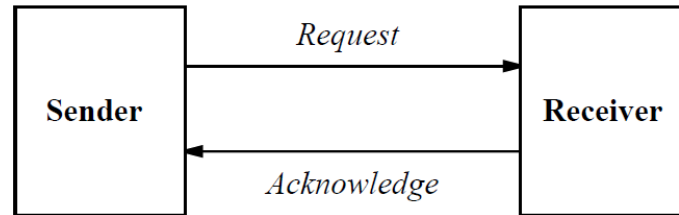
	$x_1 x_2$			
	00	01	11	10
$a = 000$	b	a	e	a
$b = 001$	b	d	b	a
$c = 011$	c	g	b	c
$g = 010$	-	a	-	-
110	-	-	-	-
$f = 111$	c	-	-	c
$d = 101$	f	d	d	f
$e = 100$	-	-	d	-

	$y_1 y_2$			
	00	01	11	10
y_3				
0	a	b	c	g
1	e	d	f	0

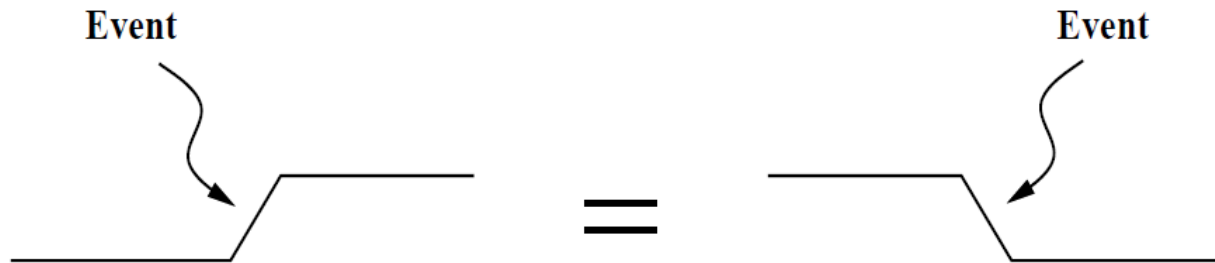
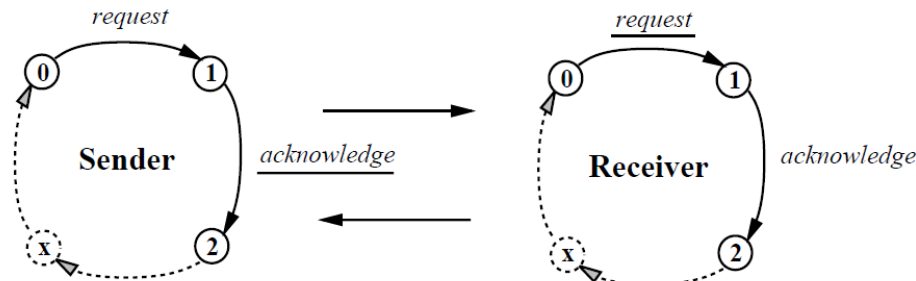
- b is adjacent to a, c, d
- c → a through g
- a → d through e
- d → c through f



Two Phase Signalling Protocol

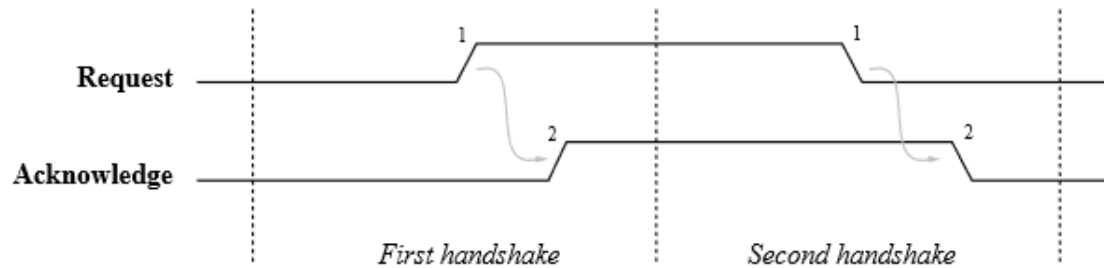
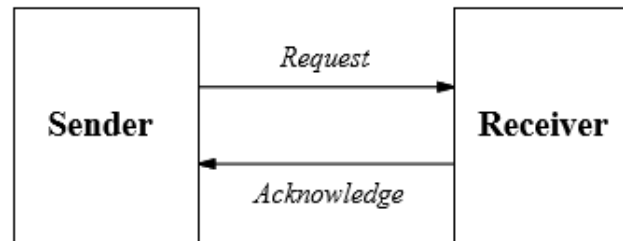


(a) Handshake at the structural level

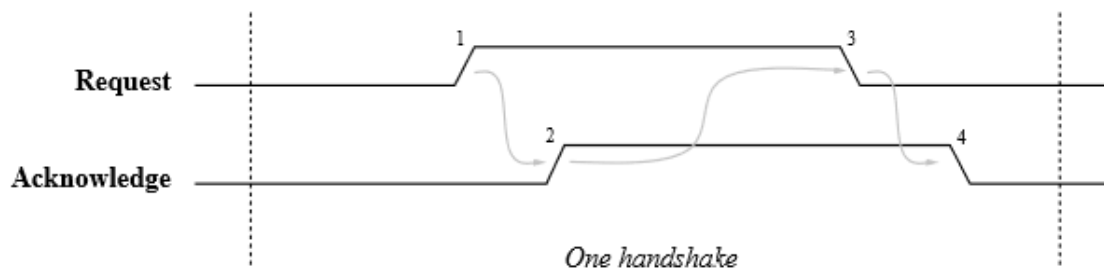


The only thing that is important is that there is an event on the signal, that it changes value, not what it changes value to.

Two versus Four Phase Signalling

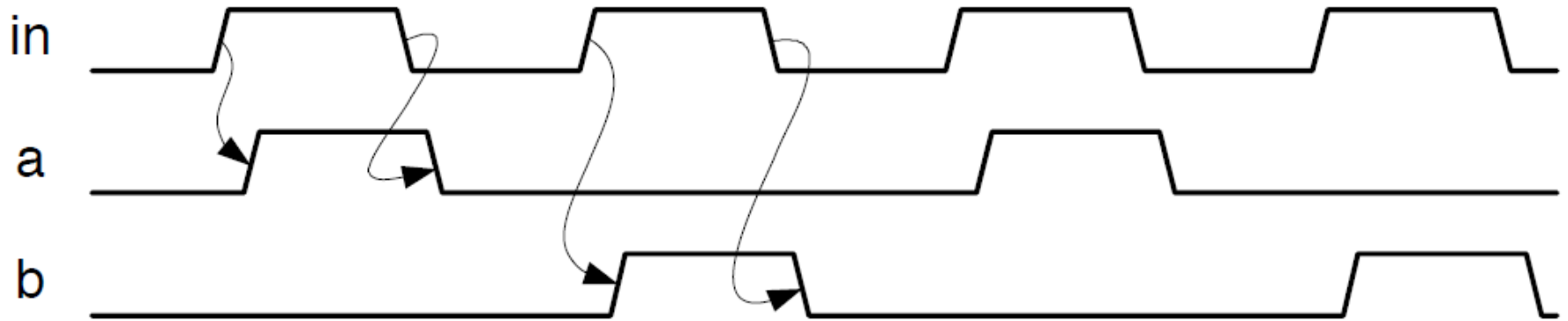
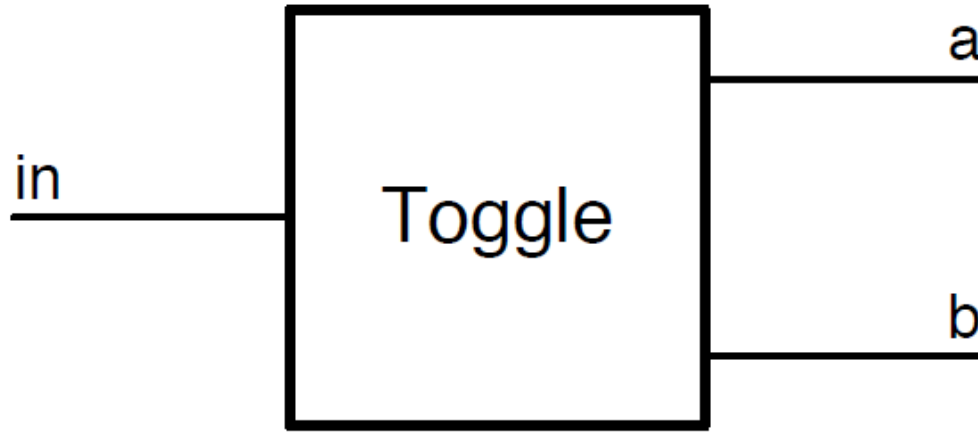


(a) Two Phase Protocol

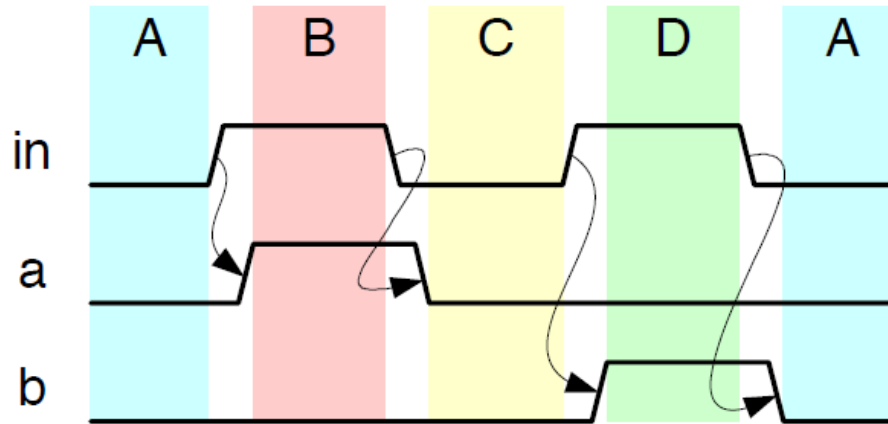


(a) Four Phase Protocol

Toggle Circuit



Toggle Circuit Flow Table



State	Next (in)		Out (a,b)
	0	1	
A	(A)	B	00
B	C	(B)	10
C	(C)	D	00
D	A	(D)	01

Karnaugh Map of the State Variables

State	Code	Next (in)		Out (a,b)
		0	1	
A	00	(A)	B	00
B	01	C	(B)	10
C	11	(C)	D	00
D	10	A	(D)	01

