Asynchronous Sequential Circuits

$x_1, x_2, \ldots, x_n$ - inputs, $z_1, z_2, \ldots, z_m$ - outputs

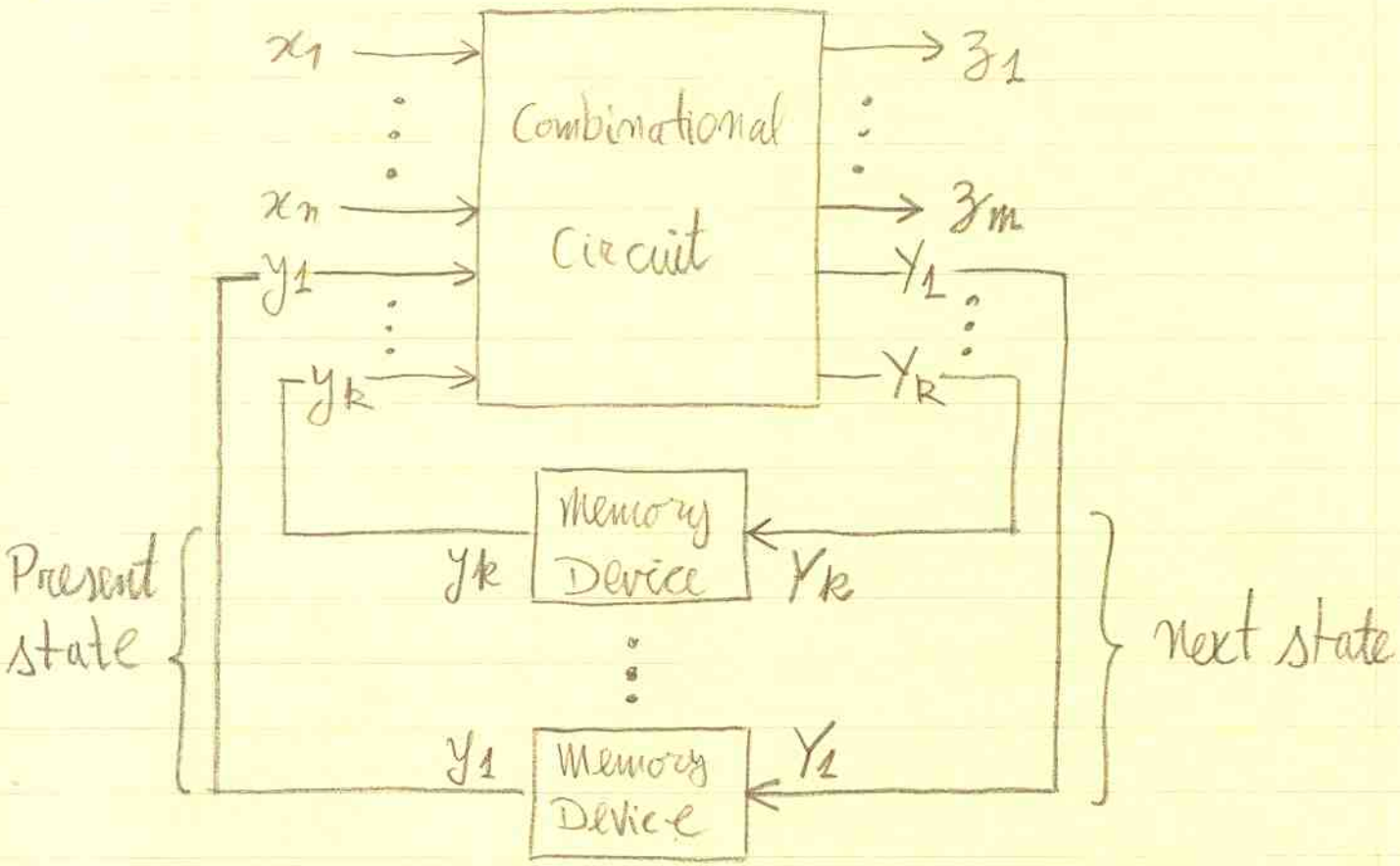
$$z_i = F_i(x_1, \ldots x_n) \quad i = 1, \ldots, m$$

This is **combinational** circuit. Outputs depend only on inputs. Circuit is **memoryless**.

Given times t_n, t_{n-1}, the outputs at t_{n-1} depend only on inputs at t_{n-1}.

In contrast the outputs of a circuit with memory at t_{n-1} depend on inputs at t_{n-1} and outputs at t_n. This dependency requires the usage of **feedback memory devices**.

Circuits with memory are called **sequential circuits** (sequential machine, finite automaton, finite state machine)



$(y_1, \ldots, y_k)$ collectively define present state

$(Y_1, \ldots, Y_k)$ " " next state

With k state variables (memory devices) a maximum of 2^k distinct states can be defined

$$Y_i = G_i(x_1, \ldots, x_n, y_1, \ldots, y_k) \quad i = 1, \ldots, k$$

$$z_i = F_i(x_1, \ldots, x_n, y_1, \ldots, y_k) \quad i = 1, \ldots, m$$ Mealy

$$z_i = F_i(y_1, \ldots, y_k) \quad i = 1, \ldots, m$$ Moore

Every Mealy circuit with k states and m outputs is equivalent to a Moore circuit with mk+1 states atmost and m outputs

Classification of Sequential Circuits

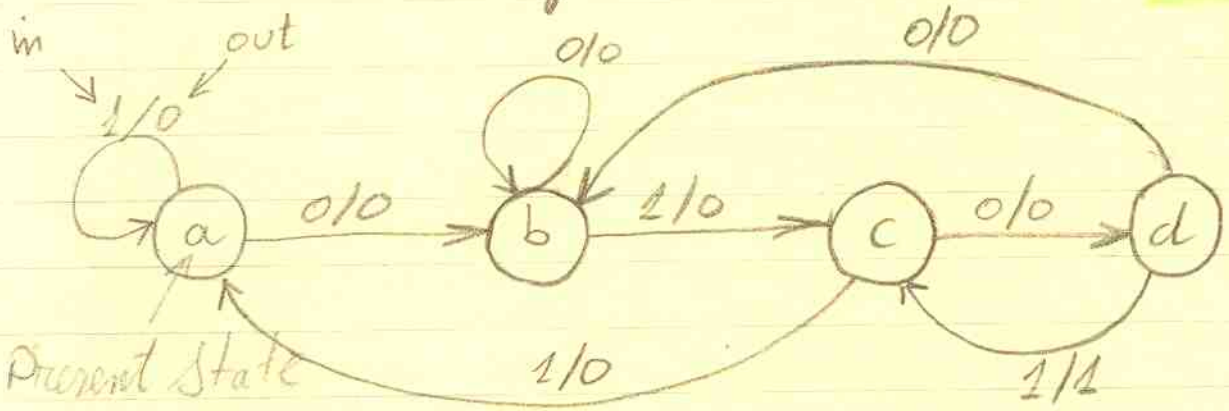
If the states can change only at discrete instants of time circuit is **Synchronous**. We use **clock pulses** generated by **clock generator**. They are input to memory devices which change state only in response to clock pulse and once for each pulse.

Asynchronous circuit can change state in response to input changes only. It must become **stable** befor next change of input. **Only one** input is allowed to change at a time

If all next states and outputs are specified for every inputs combination, sequential circuit is completely specified. Otherwise they are incompletely specified.

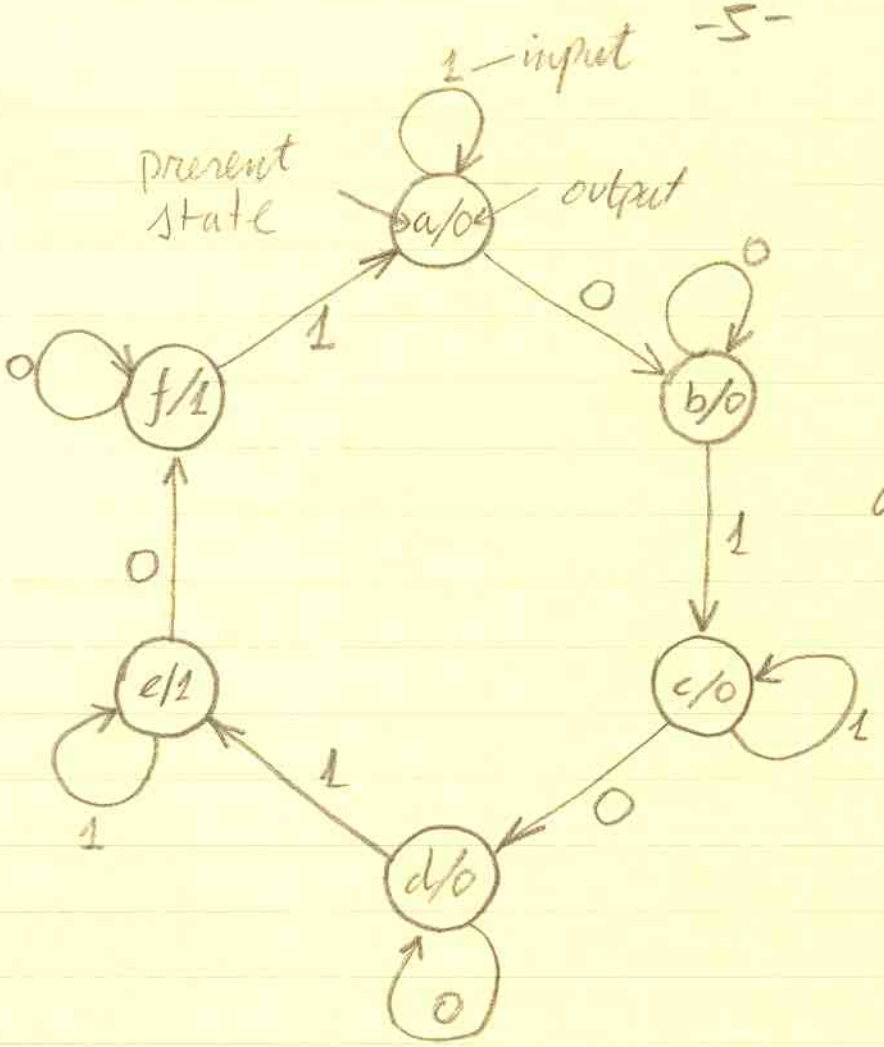
Representation of Sequential Circuits

Synchronous Mealy



How it is working?

Apply 0101 to a → b and output 0 → c and output 0 → d and output 0 → c and output 1. The response of 0101 is 0001.



Moore

asynchronous.

The response of 0101 sequence input to a is 0001.

The above Mealy is Synchronous as it may change state without input change. Moore above is asynchronous since it change state only in response to input change.

In synchronous circuit all events are synchronized to clock pulses. Hence a 0 input entering into a state is a different event than 0 input leaving that state.

State table

Mealy

Present state (PS)	next state / output (NS / output) z	
	x=0	x=1
a	b/0	a/0
b	b/0	c/0
c	d/0	a/1
d	b/0	c/1

For n inputs it has 2^n columns.

Moore

PS	NS		output z
	x=0	x=1	
a	b	a	0
b	b	c	0
c	d	c	0
d	d	e	0
e	f	e	1
f	f	a	1

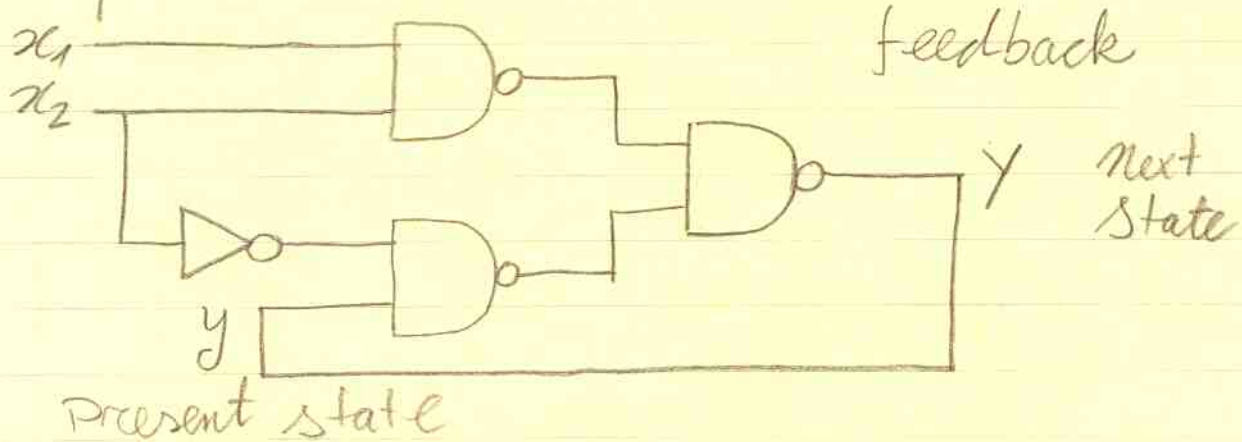
State table of incompletely specified state machine

PS	NS / output Input (x_1x_2) 00	01	11	10
a	-/-	c/1	b/-	e/1
b	e/0	-/-	-/-	-/-
c	f/0	f/1	-/-	-/-
d	a/-	-/-	e/-	b/1
e	-/-	f/0	d/1	a/0
f	c/0	-/-	c/1	b/0

State Machine Analysis

Start with a circuit schematics or Boolean equations and results in state machine diagram or state table

Example



$$Y = x_1 x_2 + x_2' y$$

The Boolean expression describes the next state as a function of inputs and present state

next state table

PS	NS (Y) Input $(x_1 x_2)$			
y	00	01	11	10
0	(0)	(0)	1	(0)
1	(1)	0	(1)	(1)

Assume $y=1$, $x_1 x_2 = 00 \Rightarrow Y=1$

change $x_1 x_2 = 01 \Rightarrow Y=0$, y is still 1,

$y \neq Y$, circuit is unstable, after a while

$y=0 \Rightarrow Y=0$ circuit is stable

The entries of next state table are called states. The encircled ones are stable

Inputs are allowed to change only when circuit is in stable state.

Problem: Let circuit be in $y=0$, $x_1x_2=00$ $\Rightarrow y=0$. Change $x_1x_2=11$, two inputs change. According to table $y \Rightarrow 1$. However, in practice $00 \rightarrow 01\,(10) \rightarrow 11$ due to uneven delay and will first "lock in" stable state ⓪ and only after a while will change to ①

Proper operation mode of asynchronous state machine: fundamental-mode circuit

- change input one at a time
- change input only at stable state

Analysis Example

2-input 3-state variables 1-output

$Y_1 = y_2 x_1 x_2' + y_2 x_1' x_2 + y_2 y_3 x_1' + y_1 y_2 + y_1 y_3$

$Y_2 = y_1' y_3' x_1' x_2 + y_1 y_2 + y_2 x_1' + y_2 x_2' + y_1 y_3 + y_1' y_3 x_1 + y_1' y_3 x_2$

$Y_3 = y_1 y_2 x_2' + y_1' y_2' x_2 + y_1' x_1 x_2' + y_1 y_3 + y_3 x_1'$

$Z = y_1 x_1 + y_2' x_2'$

PS	NS ($Y_1 Y_2 Y_3$) / output Input ($x_1 x_2$)			
$y_1 y_2 y_3$	00	01	11	10
000	(000)/1	011/0	001/0	111/1
001	011/1	(001)/0	011/0	011/1
011	111/0	111/0	010/0	111/0
010	(010)/0	110/0	000/0	111/0
110	111/0	(110)/0	(110)/1	111/1
111	(111)/0	(111)/0	(111)/1	(111)/1
101	111/1	111/0	111/1	111/1
100	000/1	000/0	000/1	000/1

$y_1 y_2 y_3 = 000$, $x_1 x_2 = 01 \Rightarrow Y_1 Y_2 Y_3 = 011$

Problem: two state variables are **changing!**

1. Y_3 changes first $\Rightarrow y_1 y_2 y_3 = 001$ circuit will look in $Y_1 Y_2 Y_3 = 001$

2. Y_2 changes first $\Rightarrow y_1 y_2 y_3 = 010 \Rightarrow$ unstable,
   $PS = 010 \Rightarrow NS = 110 \Rightarrow$ stable $Y_1 Y_2 Y_3 = 110$

3. $Y_2 Y_3$ change simultaneously $\Rightarrow y_1 y_2 y_3 = 011$
   $\Rightarrow$ unstable, $PS = 011 \Rightarrow NS = 111 \Rightarrow$ stable
   $Y_1 Y_2 Y_3 = 111$

Next state depends on relative propagation delay

Race condition exists when **more than one** state variable changes. Race is **Critical** if circuit reaches more than one stable state

$PS = 000$ and $x_1 x_2 = 01$ is **Critical race** condition.

Noncritical races: the outcoming state is independent of the order of state variable change.

Example: PS = 010, $x_1x_2 = 10$ NS = 111 in table. Both $y_1y_2y_3 = 110$ and $y_1y_2y_3 = 011$ result in same NS = 111.

Buzzer: totaly unstable operation

Example: PS = 001, $x_1x_2 = 11 \Rightarrow$

$Y_1Y_2Y_3 = 011, 010, 000, 001, 011, \ldots$

Question: Let S be the number of states. Show that for any buzzer there exists some $q \leq S$ and $N > 0$ such that if S_i is a state then $S_i = S_{i+q}$ $i \geq N$, namely, a buzzer is periodic.

Flow Table

We symbols rather than binary code of states. We are interested in state transition rather than specific codes.

a = 000, b = 001, c = 010, d = 110, e = 111

PS	NS / output, Input 00 (races!)	01	11	10
a	A @/1	b, d, e/0	—	e/1
b	e/1	B ⓑ/0	Buzzer	—
c	C ©/0	d/0	—	e/0
d	e/0	D @/0	E @/1	e/1
e	F ⓔ/0	F ⓔ/0	G ⓔ/1	G ⓔ/1

Mealy

unstable passing through non existing state 011

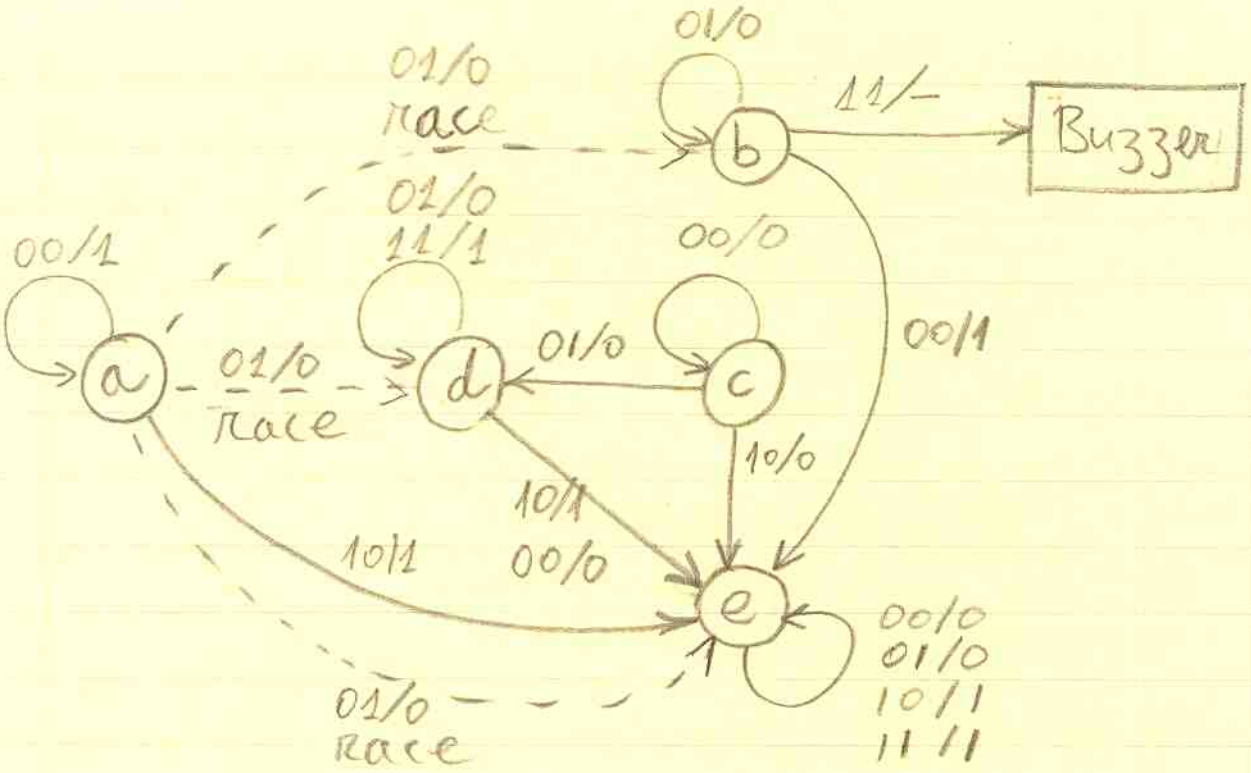
Conversion to Moore: recall that output is a part of the state, hence a Mealy state yielding different output translates into distinct Moore states.

A = 000 under $x_1x_2 = 00$
B = 001 under $x_1x_2 = 01$
C = 010 under $x_1x_2 = 00$
D = 110 under $x_1x_2 = 01$
E = 110 under $x_1x_2 = 11$
F = 111 under $x_1x_2 = 00, 01$
G = 111 under $x_1x_2 = 11, 10$

PS	NS: Input (x_1x_2) 00	01	11	10	output z
A	(A)	B, D, F	—	G	1
B	F	(B)	Buzzer	—	0
C	(C)	D	—	F	0
D	F	(D)	E	—	0
E	—	D	(E)	G	1
F	(F)	(F)	G	G	0
G	F	F	(G)	(G)	1

[illegible] fundamental mode

due to fundamental mode



a, c - initial states - once leaving they are never entered.

e - final state - once entered, never left.

Primitive Flow Tables

This is the initial flow table occurring in design. It has only one stable state at every row. Unspecified next state occur either because of fundamental mode

or because some input is not allowed.

	NS / output ($Y_1 Y_2$)			
	Input ($x_1 x_2$)			
	00	01	11	10
a	(a/00)	b/-	-/-	e/-
b	c/-	(b/10)	f/-	-/-
c	(c/00)	h/-	-/-	d/-
d	a/-	-/-	f/-	(d/11)
e	g/-	-/-	f/-	(e/11)
f	-/-	h/-	(f/01)	e/-
g	(g/00)	h/-	-/-	e/-
h	c/-	(h/10)	i/-	-/-
i	-/-	j/-	(i/01)	e/-
j	a/-	(j/-)	i/-	-/-

Unspecified outputs are left so if Mealy implementation is of interest. In Moore implementation all outputs are set equal in a row defined by the stable state.