Asynchronous Circuit Design

Design Spec.

↓

① - Primitive flow table
- State Diagram

↓

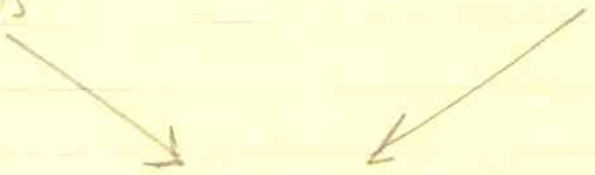
② State Reduction

↓

③ - Critical Race-free state assignment
- Derive transition table

④, ⑤
- Circuit excitation table
- Latches excitation equations

- Next state equations
- Assign unspecified outputs
- Derive output equations



Circuit Logic diagram

Design Example

handset A	bulb A	handset B	bulb B
Replaced	@ OFF	Replaced	OFF
Lifted	ⓑ OFF	Replaced	ON
Replaced	ⓒ ON	Lifted	OFF
Replaced	ⓔ OFF	Lifted	ON
Lifted	ⓕ ON	Replaced	OFF
Lifted	ⓓ OFF	Lifted	OFF

Initially both handsets are replaced. Ignore the case of both switch state simultaneously.

x_1, x_2 - handset status 0-replaced, 1- lifted

z_1, z_2 - bulb ON, OFF

State definition for Moore Model

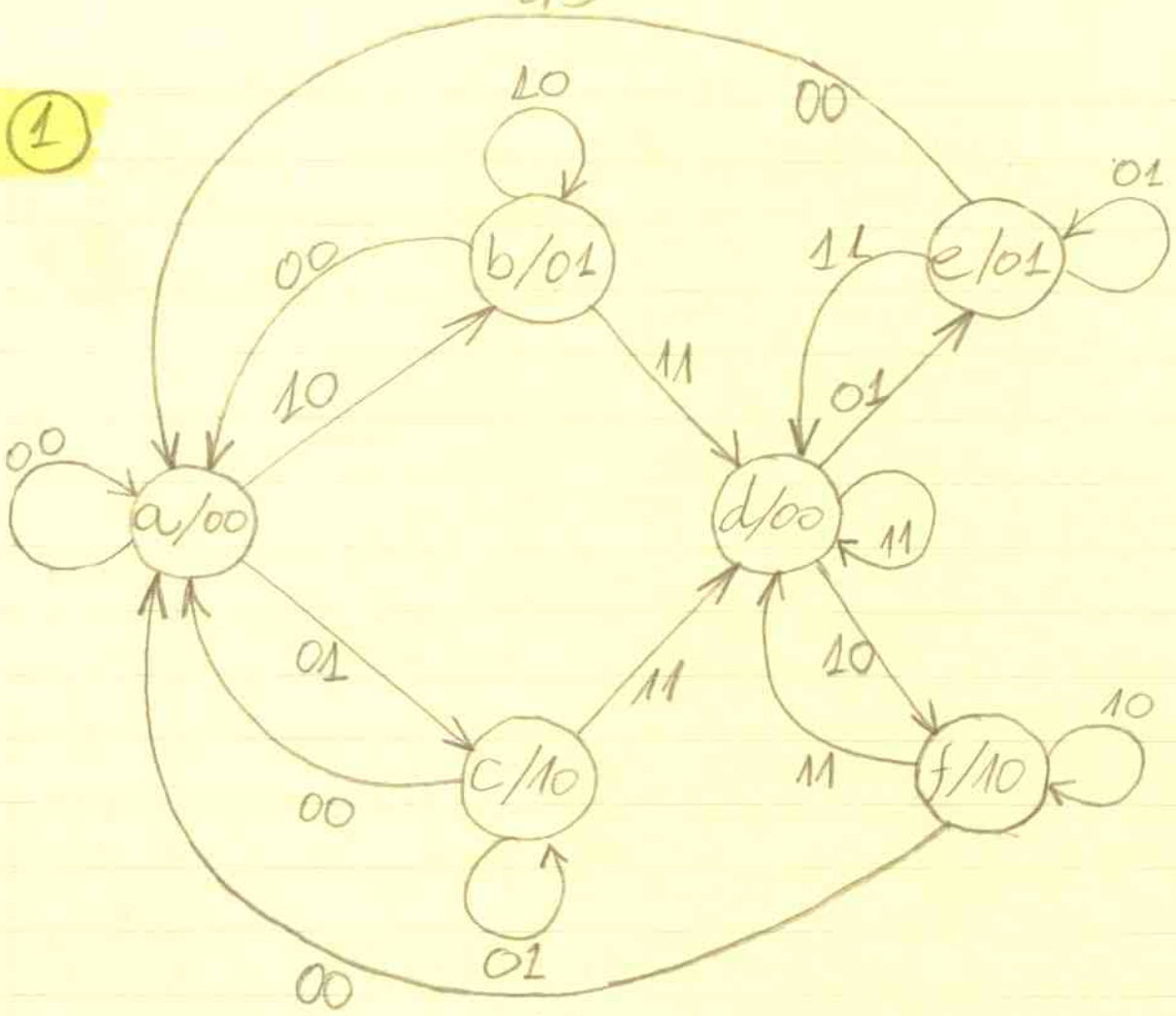
a- both handsets replaced ($z_1 z_2 = 00$)

b- A lifted ($z_1 z_2 = 01$) c- B lifted ($z_1 z_2 = 10$)

d- the counterpart lifted ($z_1 z_2 = 00$)

e - A replaced ($z_1 z_2 = 01$), f- B replaced ($z_1 z_2 = 10$)

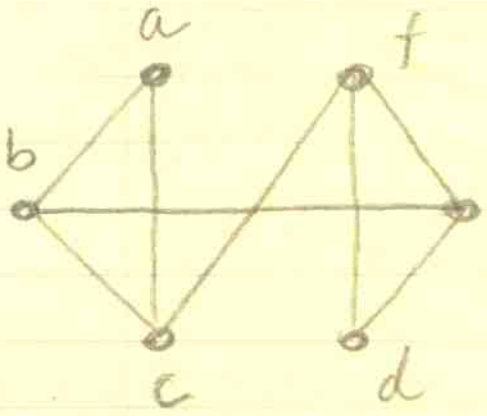
Step (1)



PS	NS x_1x_2 00	01	11	10	outputs z_1z_2
a	(a)	c	—	b	00
b	a	—	d	(b)	01
c	a	(c)	d	—	10
d	—	e	(d)	f	00
e	a	(e)	d	—	01
f	a	—	d	(f)	10

Step (2) Simplification

- No redundant states (**homework**)
- Better merging with Mealy (due to less conflict chance at outputs of stable states).



(a,b,c), (e,f,g)
Smallest cover

PS	NS/output Y_1Y_2 Input x_1x_2 00	01	11	10
a	(a/00)	(a/10)	b/-	(a/01)
b	a/-	(b/01)	(b/00)	(b/10)

Step (3) State assignment - trivial, no races

y	Y/Y_1Y_2 x_1x_2 00	01	11	10
0	(0/00)	(0/10)	1/-	(0/01)
1	0/-	(1/01)	(1/00)	(1/10)

Step ④ Equation derivation

Y – next state

y \ x_1x_2	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$Y = x_1x_2 + y(x_1 + x_2)$$

Output – assignment of unspecified outputs

z_1

x_1x_2	00	01	11	10
0	0	1	0	0
1	0	0	0	1

z_2

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

Unspecified must be set both to 0 in order

to avoid spikes of 1 at unstable states

Here is the rule:

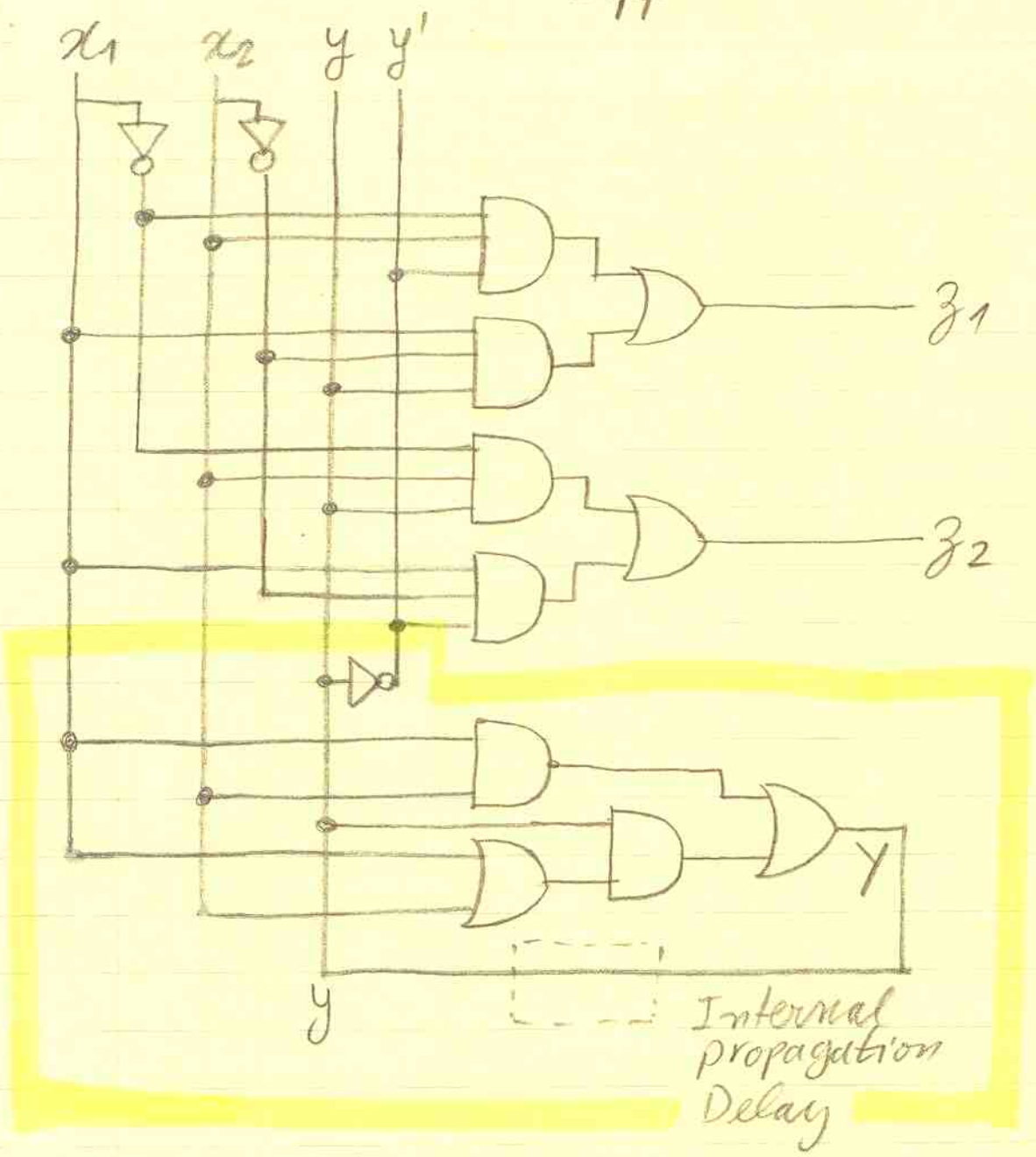
① If the value of some output doesn't change between two stable states, it must be set the same at each unstable states through which transition occurs.

② If the value of any output does change value in transition between two stable states, it can be assigned X (don't care) at each unstable state through which transition occurs.

$$Z_1 = x_1' x_2 y' + x_1 x_2' y$$

$$Z_2 = x_1' x_2 y + x_1 x_2' y$$

At this point fully functional circuit can be implemented where the internal signal propagation delay suffices for feedback loop delay.



Step (5) Implementation with latches

No general procedure to determine which type of latch yields minimum logic to generate S and R. In any case, Q and Q' are used for y and y'

① The logic equation of Y was found as

⊛ $Y = x_1 x_2 + y(x_1 + x_2)$

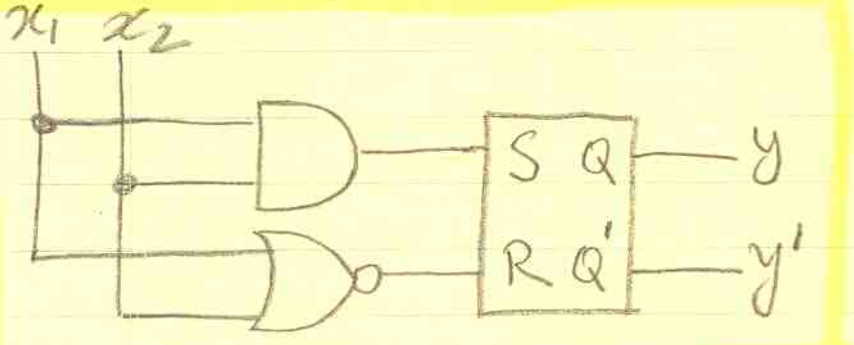
② Characteristic equation of SR Latch is

$Q^+ = R'Q + S \quad (SR=0)$

Namely ⊛⊛ $Y = R'y + S$

Comparing ⊛ with ⊛⊛ yields:

$R' = x_1 + x_2 \Rightarrow R = x_1' x_2', \; S = x_1 x_2$

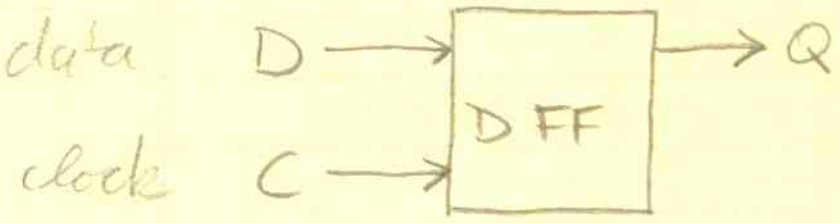


must validate that SR=0

$SR = x_1 x_2 x_1' x_2' = 0$ indeed

Design example

Positive Edge-Triggered Flip-Flop



Data is transferred to Q only on positive edge (0→1 transition) of clock

D=1 ⇒ Q=1 on change of C from 0→1

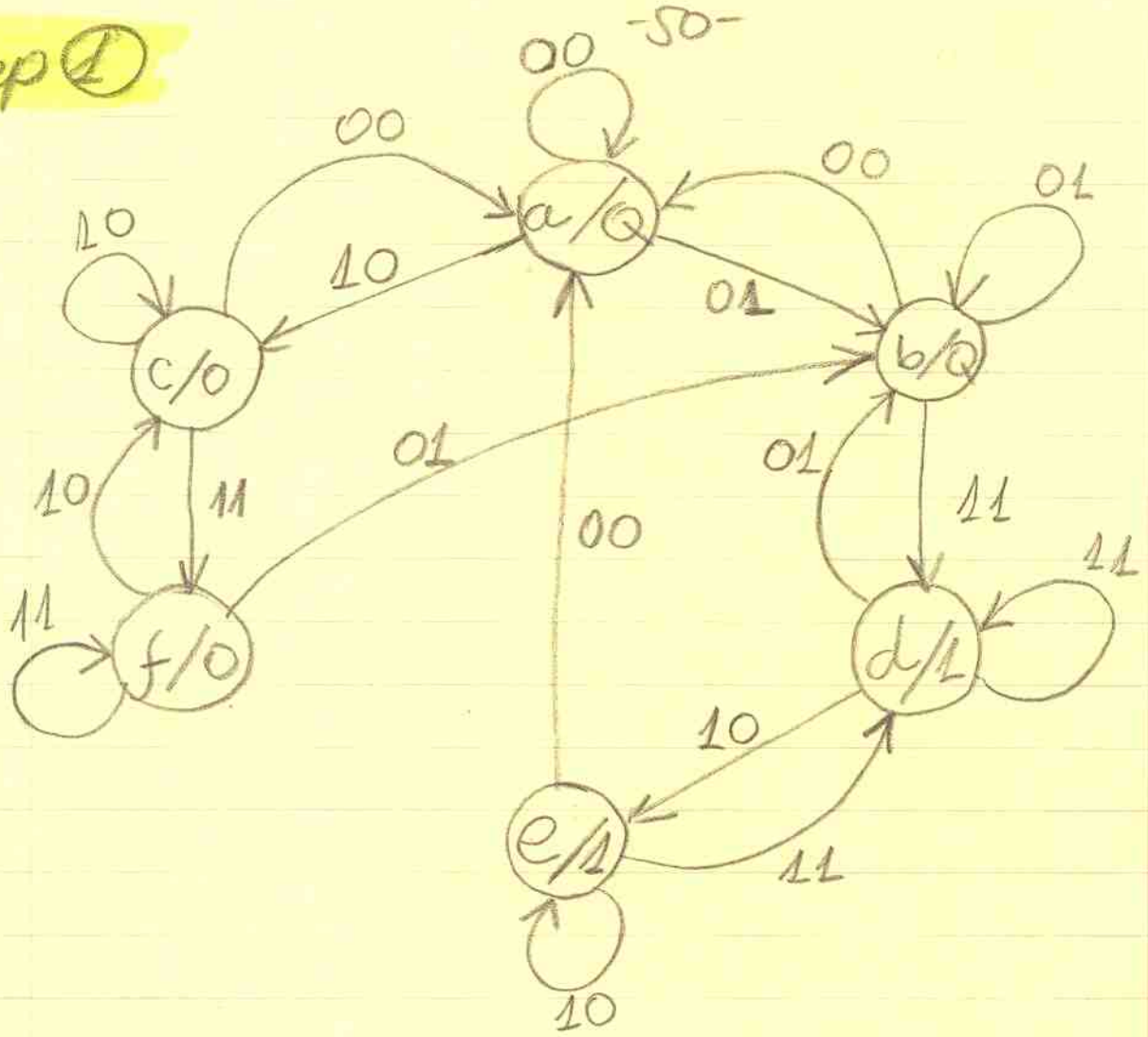
D=0 ⇒ Q=0 ↙

There are 2 inputs CD and 1 output Q

Moore model

Let us start with a state where output is either 0 or 1, hence Q

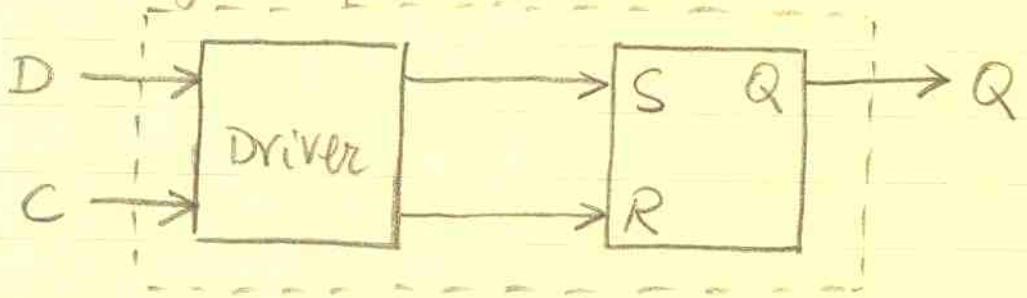
Step ①



Primitive state table

	CD 00	01	11	10	Q^+
a	ⓐ	b	–	c	Q
b	a	ⓑ	d	–	Q
c	a	–	f	ⓒ	0
d	–	b	ⓓ	e	1
e	a	–	d	ⓔ	1
f	–	b	ⓕ	c	0

Design implementation with SR Latch



We'll generate S and R from flow table to drive an active-low (NAND) latch

S	R	Q^+
1	1	Q
1	0	0
0	1	1
0	0	Indeterminate

Step ② reducing table

- No redundant states
- Merging yields (a,b), (c,f) (d,e)
  A B C

PS	NS Input (CD) 00	01	11	10	Output Q+
A	(A)	(A)	C	B	Q
B	A	A	(B)	(B)	0
C	A	A	(C)	(C)	1

There are 3 states, hence 2 state variables, hence 2 memory elements (2 SR latches)

State assignment A=11, B=10, C=01 is races-free and corresponds to the valid inputs of SR active-low characteristic table (00 is invalid)

PS Y_1Y_2	NS (Y_1Y_2) CD 00	01	11	10
11	(11)	(11)	01	10
10	11	11	(10)	(10)
01	11	11	(01)	(01)
00	X	X	X	X

Y_1 and Y_2 4 variable K-map

↗ Invalid, will never happen due to state assignment

Step (4) - Derive equations, use K-map

$$Y_1 = C' + y_1 y_2' + D' y_1$$

$$Y_2 = C' + y_1' + D y_2$$

We can build now the circuits for Y_1 and Y_2 with internal propagation delay for memory and connect to SR latch to generate Q.

Instead, let's implement with Latches

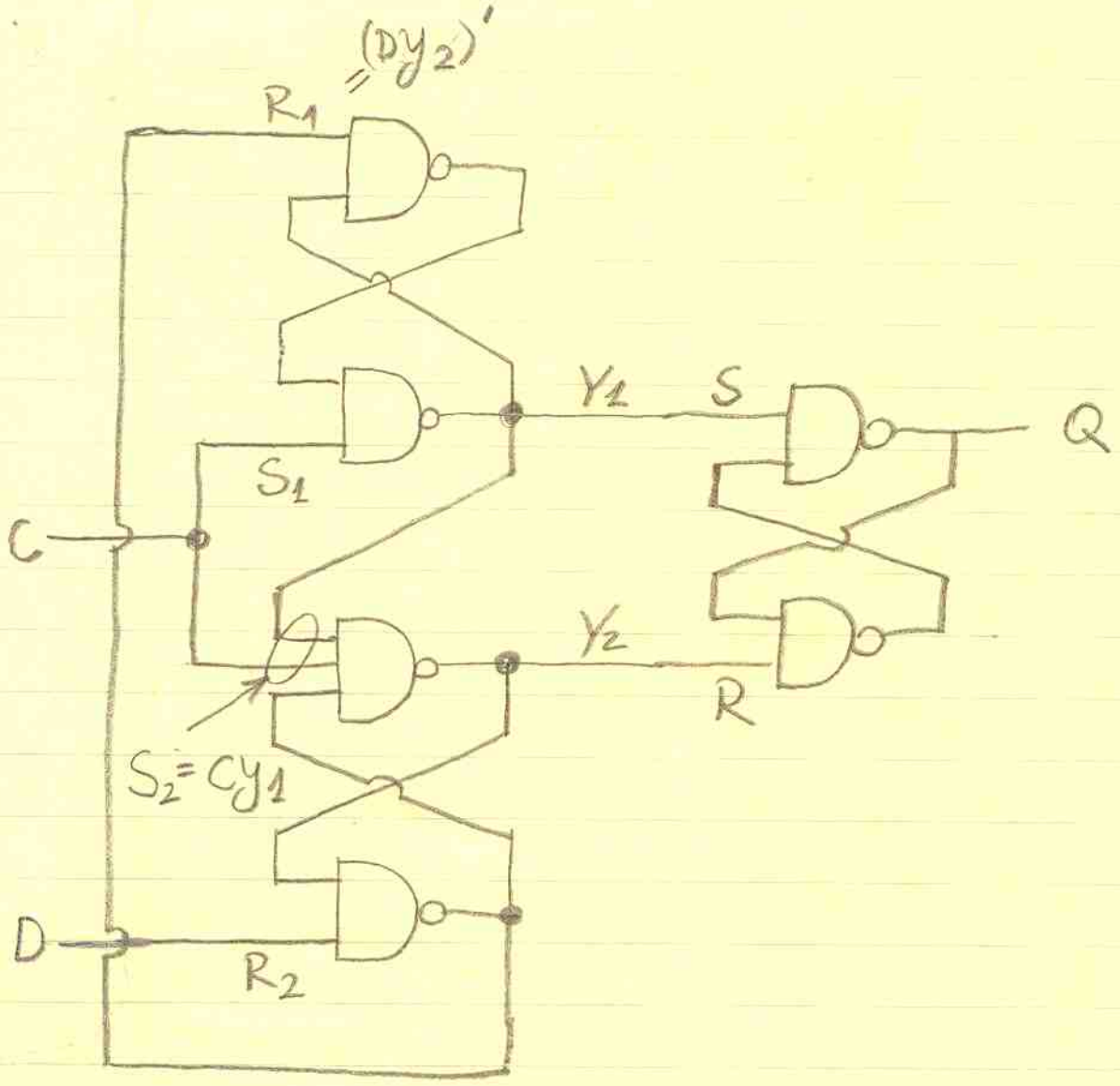
Step (5) use latches

active-low characteristic equation

$$Q^+ = RQ + S' = (S(RQ)')'$$

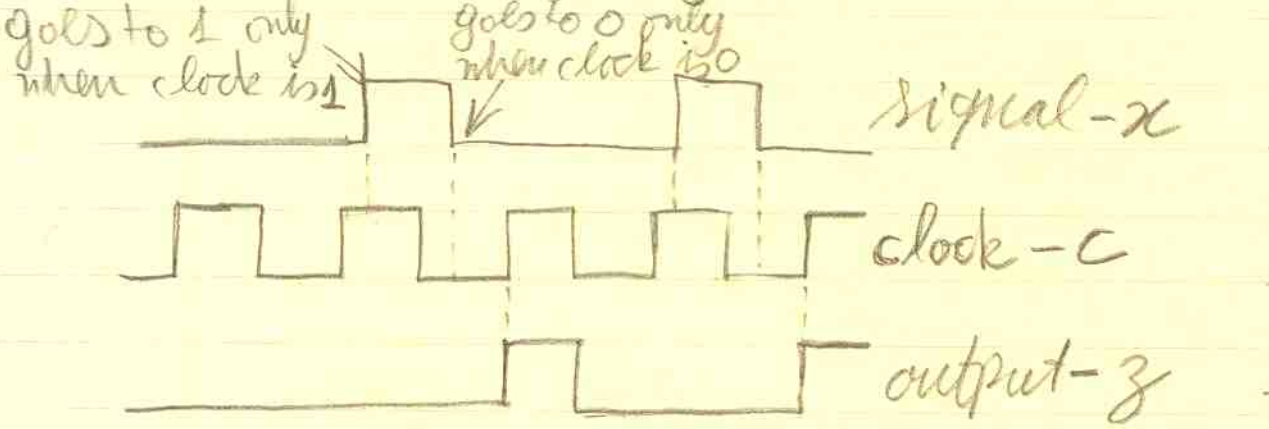
$$Y_1 = \{C[(Dy_2)' y_1]'\}' \Rightarrow \begin{matrix} S_1 = C \\ R_1 = (Dy_2)' \end{matrix}$$

$$Y_2 = [Cy_1(Dy_2)']' \Rightarrow \begin{matrix} S_2 = Cy_1 \\ R_2 = D \end{matrix}$$



When drawing, start with the building blocks alone. Then identify ID of each. Then complete all internal connections to provide proper excitations.

Example - Aligning a signal to clock



data pulse is narrow enough, its period are larger than clock

- Input x goes to 1 only when clock C is 1
- Input x goes to 0 only when clock C is 0
- output is synchronized to next clock and results a pulse of one clock, regardless of the input pulse width

a- System in "reset" both clock and data are 0. Output is 0

b- System gets ready to accept input Pulse as clock goes to 1. Output still 0

c- Input data changed to 1 while

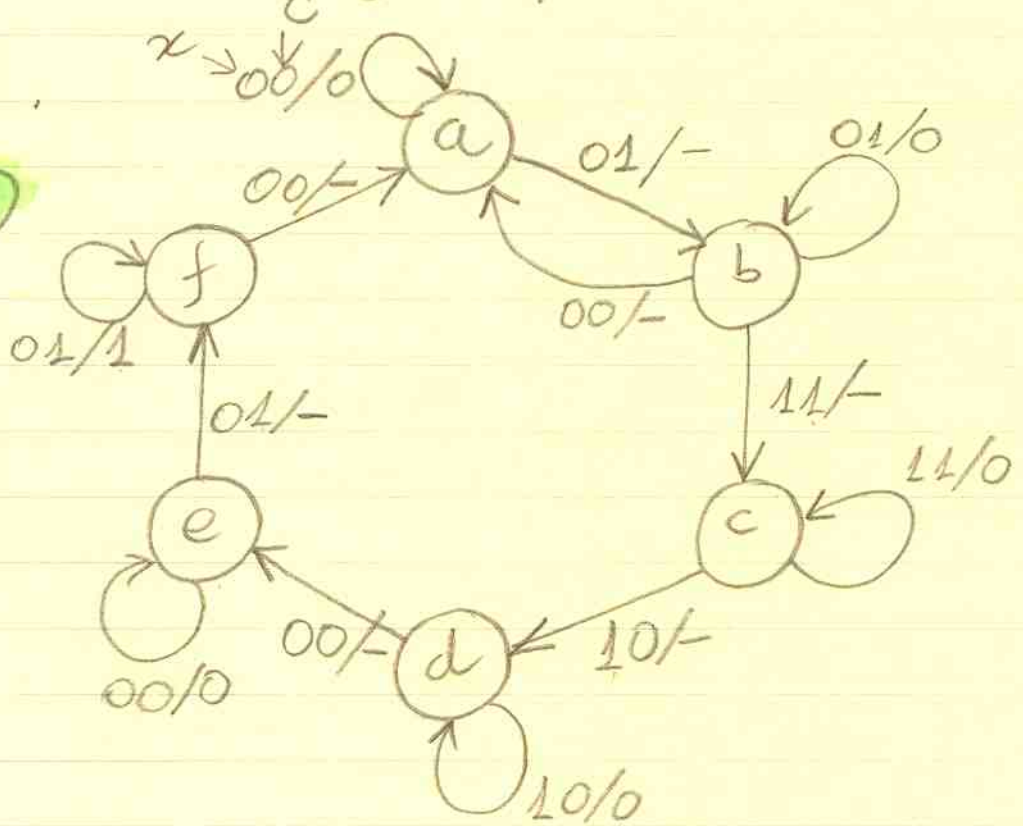
clock is 1. Output still 0

d- clock pulse ended and data still 1

e- data pulse ended (went to 0) while clock was 0. System gets ready to generate pulse in sync with next clock.

f- clock arrives, a pulse is issued, output is 1.

Step 1



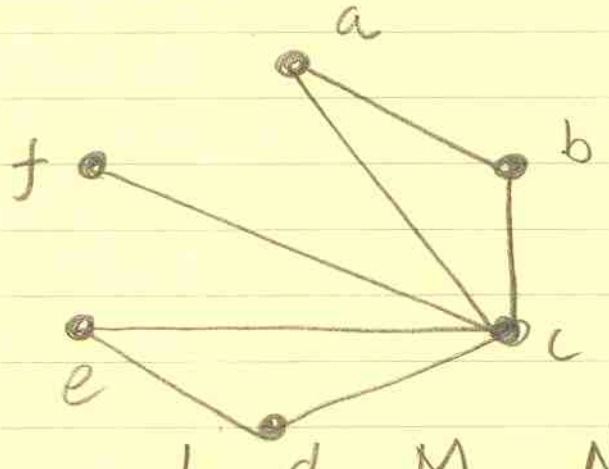
Mealy

Primitive flow table

PS	NS/output (z) Input (xC) 00	01	11	10
a	(a/0)	b/–	–/–	–/–
b	a/–	(b/0)	c/–	–/–
c	–/–	–/–	(c/0)	d/–
d	e/–	–/–	–/–	(d/0)
e	(e/0)	f/–	–/–	–/–
f	a/–	(f/1)	–/–	–/–

Step ② - State reduction

- no equivalent states
- Merging



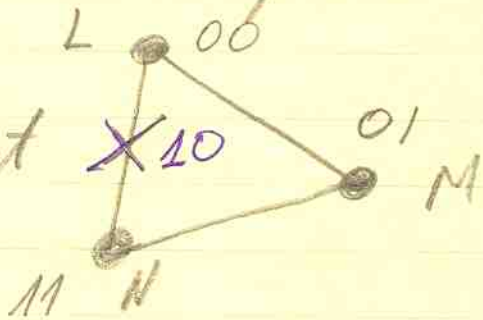
(a,b,c), (d,e), (f) ; (a,b), (c,d,e), (f)

(a,b), (c,f), (d,e)

PS	00	01	11	10
L	(L/0)	(L/0)	M/-	-/-
M	(M/0)	N/-	(M/0)	(M/0)
N	L/-	(N/1)	-/-	-/-

Step (3) - state assignment

State assignment graph



No race-free assigment. We'll add another row with unstable state

			Y_1Y_2/Z	
y_1y_2	00	01	11	10
00	(00/0)	(00/0)	01/-	-/-
01	(01/0)	11/-	(01/0)	(01/0)
11	10/-	(11/1)	-/-	-/-
10	00/-	-/-	-/-	-/-

Step ④ Next state and output equations

y_1y_2 \ xG	00	01	11	10
00	0	0	0	X
01	0	1	0	0
11	1	1	X	X
10	0	X	X	X

K-map for Y_1

(*) $Y_1 = y_1y_2 + y_2x'G$

y_1y_2 \ xG	00	01	11	10
00	0	0	1	X
01	1	1	1	1
11	0	1	X	X
10	0	X	X	X

These have been set to 1 for Y_1. Since next is 11, they must stay 0, otherwise unwanted stable states 11 will be created

(**) $Y_2 = y_1'y_2 + y_2x'G + y_1'x$

$= y_2(y_1' + x'G) + y_1'x$

$Z = y_2$

Step ⑤ Implementation with latches

We'll use active-High latches

$Q^+ = R'Q + S \quad (SR=0)$

from (*) it follows:

$R_1 = y_2'$, $S_1 = y_2 x'G$, and indeed

$S_1 R_1 = y_2' y_2 x'G = 0$

from (**) it follows

$R_2 = (y_1' + x'G)' = y_1(x+C')$, $S_2 = y_1' x$ and

indeed, $S_2 R_2 = y_1' x y_1 (x+C') = 0$