Analysis and Synthesis of Sequential Logic Circuits

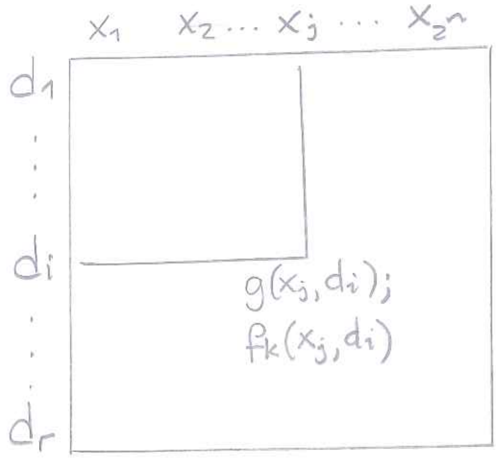
- The sequential circuits are logic circuits
  ↳ whose output depends not only on the present input but also on the history of the input (state)
- A sequential logic circuit thus has storage (memory)
  ↳ while a combinational logic circuit does not
- The input/state of a sequential circuit at $t=i$ uniquely,
  ↳ specify the output at $t=i$ and the state at $t=i+1$, i.e. next state.

Hence;

- Given a sequential circuit, we need to specify
  i) The outputs corresponding to a present state and input sequence
  ii) The next state corresponding to a present state and input sequence
- this relationship can be given in two ways:
  ↳ - state table
  - state diagram

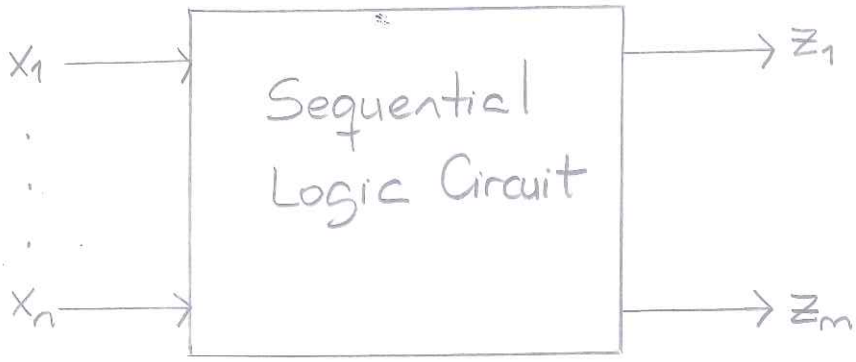
Sequential circuit and state table

- state table is a matrix with r rows and m columns where $m \leq 2^n$ with n denoting number of inputs, r denoting # of states, m denoting # of outputs



State table

- each row represents a single state
- each column represents an input sequence
- the (i,j) entry of matrix involves next state and output information
  - ↳ which are fn.'s of present state, d_i and input sequence, $x_j \triangleq (x_1, \dots, x_n)_j$



$$z_k = f_k(x_1, \dots, x_n, d_i) \quad , k = 1, \dots, m \qquad : \text{outputs}$$

$$d_j = g(x_1, \dots, x_n, d_i) \quad , j = 1, \dots, s \qquad : \text{next state}$$

example.

- Let $n = 1$, $m = 1$ and $r = 6$ for a sequential circuit
  - ↳ whose state table is given as follows

x_j / d_i	0	1
A	E,0	D,1
B	F,0	D,0
C	E,0	B,1
D	F,0	B,0
E	C,0	F,1
F	B,0	C,0

State Table

- while the sequential circuit is in state B, if we apply the input sequence

$$x^{i}\,x^{i+1}\,x^{i+2}\,x^{i+3}\,x^{i+4} = 10110$$

then the next state and output values are summarized in the following table

t	0	1	2	3	4	
x	1	0	1	1	0	
d_i	B	D	F	C	B	F
z	0	0	0	1	0	

Sequential circuit and state diagram

- In order to specify the behavior of a sequential circuit,

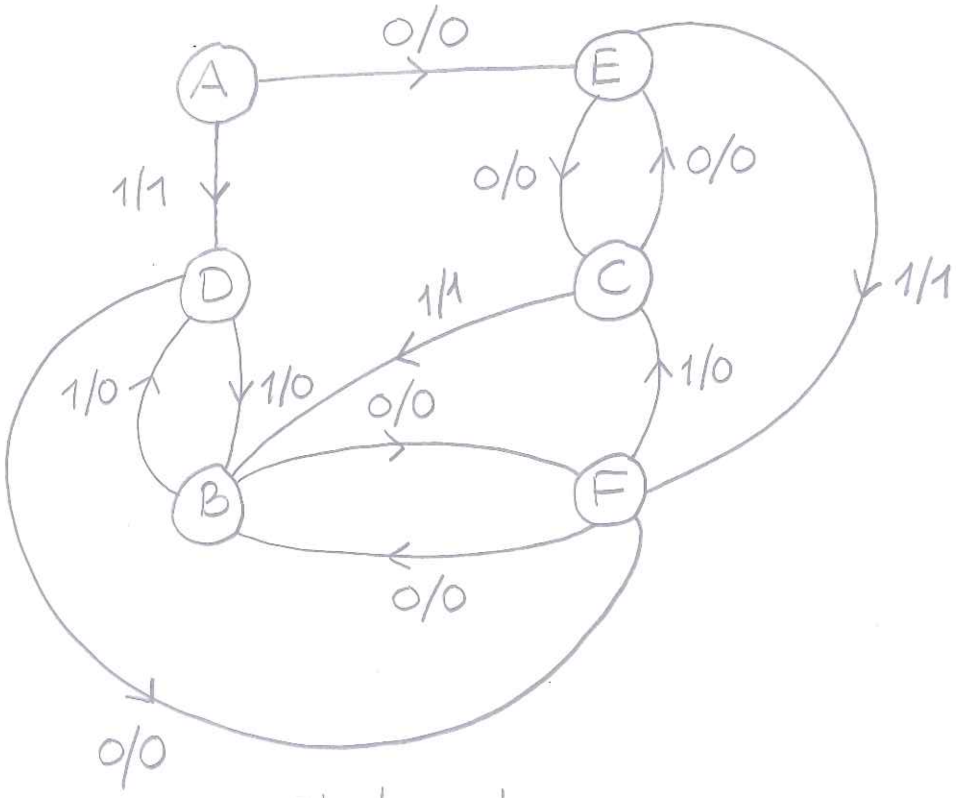
↳ each state is represented with a node

- if the state machine is in state d_i and if the next state becomes d_j when input x_j is applied,
  - ↳ then the nodes d_i and d_j are connected with a topologic element whose direction is from i to j

$$d_i \xrightarrow{x_j / f(x_j, d_i)} d_j$$

- on top of this topologic element, the input x_j and output $z = f(x_j, d_i)$ are specified appropriately

example. Let us consider the previous example and give the state diagram as follows



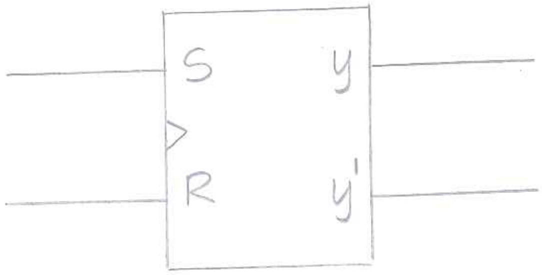
State diagram

Sequential circuit elements

- we investigate several types of flip-flop which is a basic storage element in sequential logic and their describing functions

S-R flip-flop

- shown in block diagram as follows



y \ SR	00	01	11	10
0	0	0	d	1
1	1	0	d	1

$Y = S + R'y$, Y: next state

↳ describing function

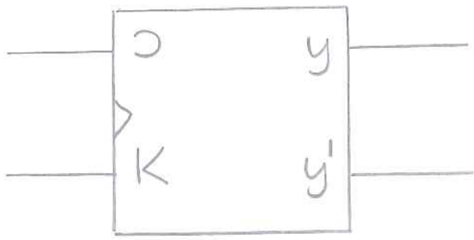
yY	00	01	11	10
	0,d	1,0	d,0	0,1

S,R

↳ inverse describing function

J-K flip-flop

- shown as follows



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

$Y = Jy' + K'y$

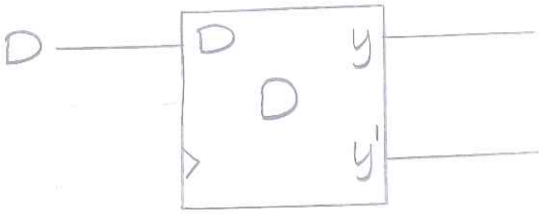
↳ describing function

$y \backslash Y$	00	01	11	10
	0,d	1,d	d,0	d,1

J, K ↳ inverse describing function

D flip-flop

- represented by



$y \backslash D$	0	1
0	0	1
1	0	1

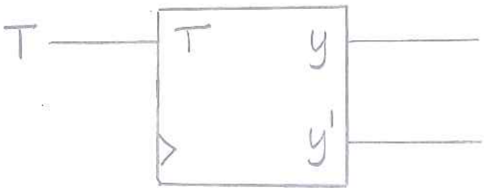
$Y = D$ (describing function)

$y \backslash Y$	00	01	11	10
	0	1	1	0

$D = Y$ (inverse describing function)

T flip-flop

- shown as follows



$y \backslash T$	0	1
0	0	1
1	1	0

$Y = Ty' + T'y = T \oplus y$

↳ describing function

$y \backslash Y$	00	01	11	10
	0	1	0	1

$T = Yy' + Y'y = Y \oplus y$

↳ inverse describing function