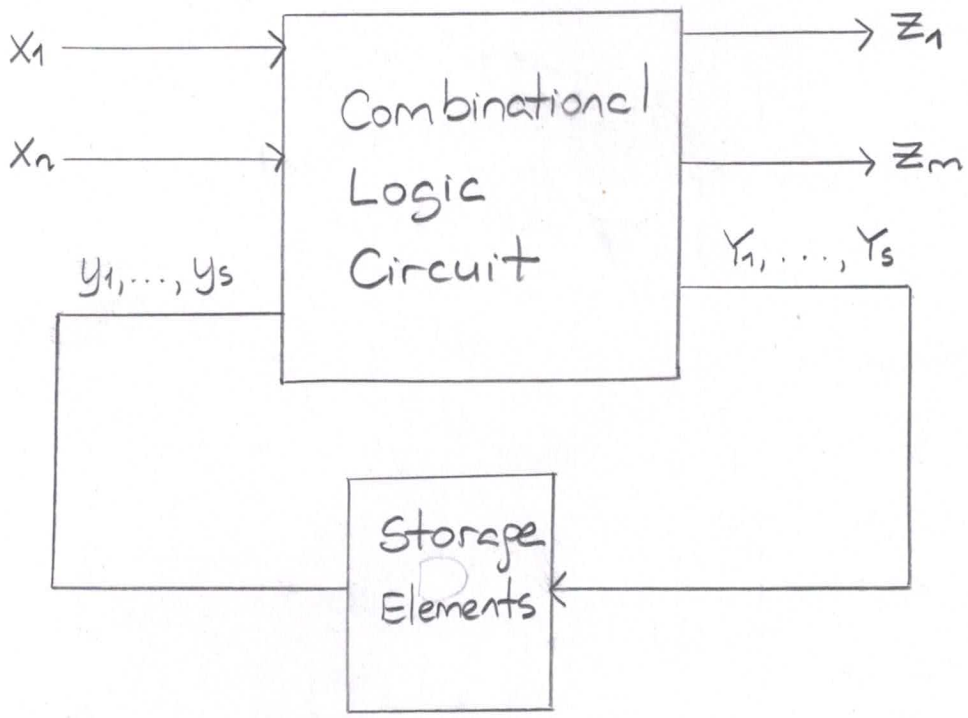
General structure of a sequential circuit

- can be given as follows



Teorem 13. A state diagram with r states can be implemented via formerly given general structure and the number of storage (memory) elements is

$$s = \log_2 \lceil r \rceil$$

proof. The output of the combinational circuit with respect to any input sequence can be selected arbitrarily

↳ thus the resulting truth table can be implemented

- This means that if the present inputs and present states are specified

↳ the corresponding output and next state can be implemented as desired

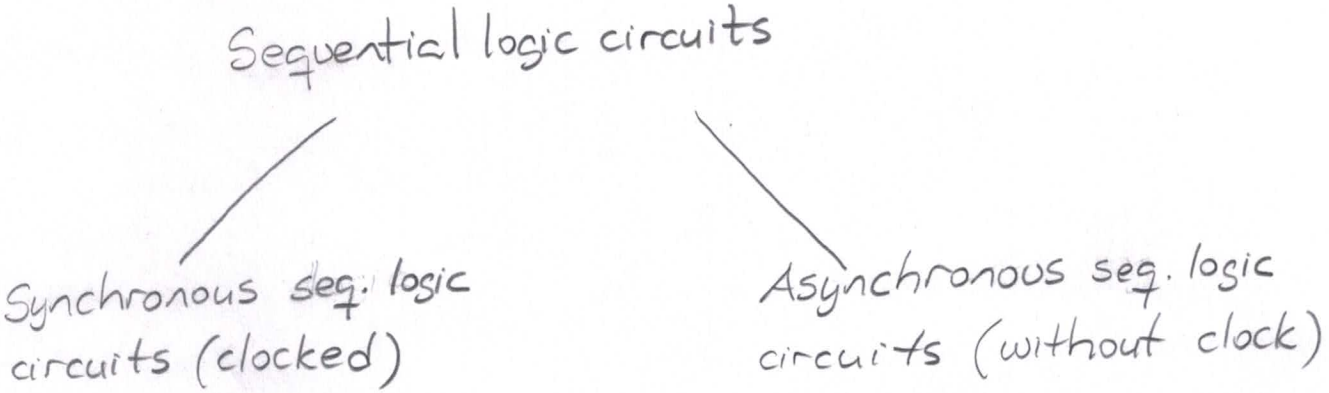
Namely,

- any state diagram can be implemented

Corollary. The sequential circuit synthesis is reduced to a combinational circuit synthesis with (n+s) inputs and (m+s) outputs.

Classification of sequential logic circuits

- can be given as follows



- asynchronous sequential logic circuits have some advantages

↳
1. No clock signal required
2. Operates fast if no transitional states exist
3. Enables the operation of two synchronous sequential circuits with different clock frequency

- In sequential logic circuits, some undesired outputs

may arise

↳ because of feedback

- thus the operation of flip-flops in a synchronous sequential logic circuit

↳ controlled with clock pulses

- two such control methods are known as

↳ - master-slave flip-flop
- edge-trisserred flip-flop

Analysis of sequential logic circuits

- The analysis of a given sequential logic circuit

↳ has the goal of determining its state table or state diagram

Step 1. The output functions corresponding to the combinational logic circuit part, namely, the functions with (n+s) variables belonging to the outputs and flip-flop inputs

↳ are expressed in accordance with the logic circuit

Step 2. The flip-flop input functions are substituted into the describing equations ($Y = Dy' + K'y$, $Y = S + R'y$, $Y = Ty' + T'y$, $Y = D$) and

$Y_1, \ldots, Y_s$ are determined.

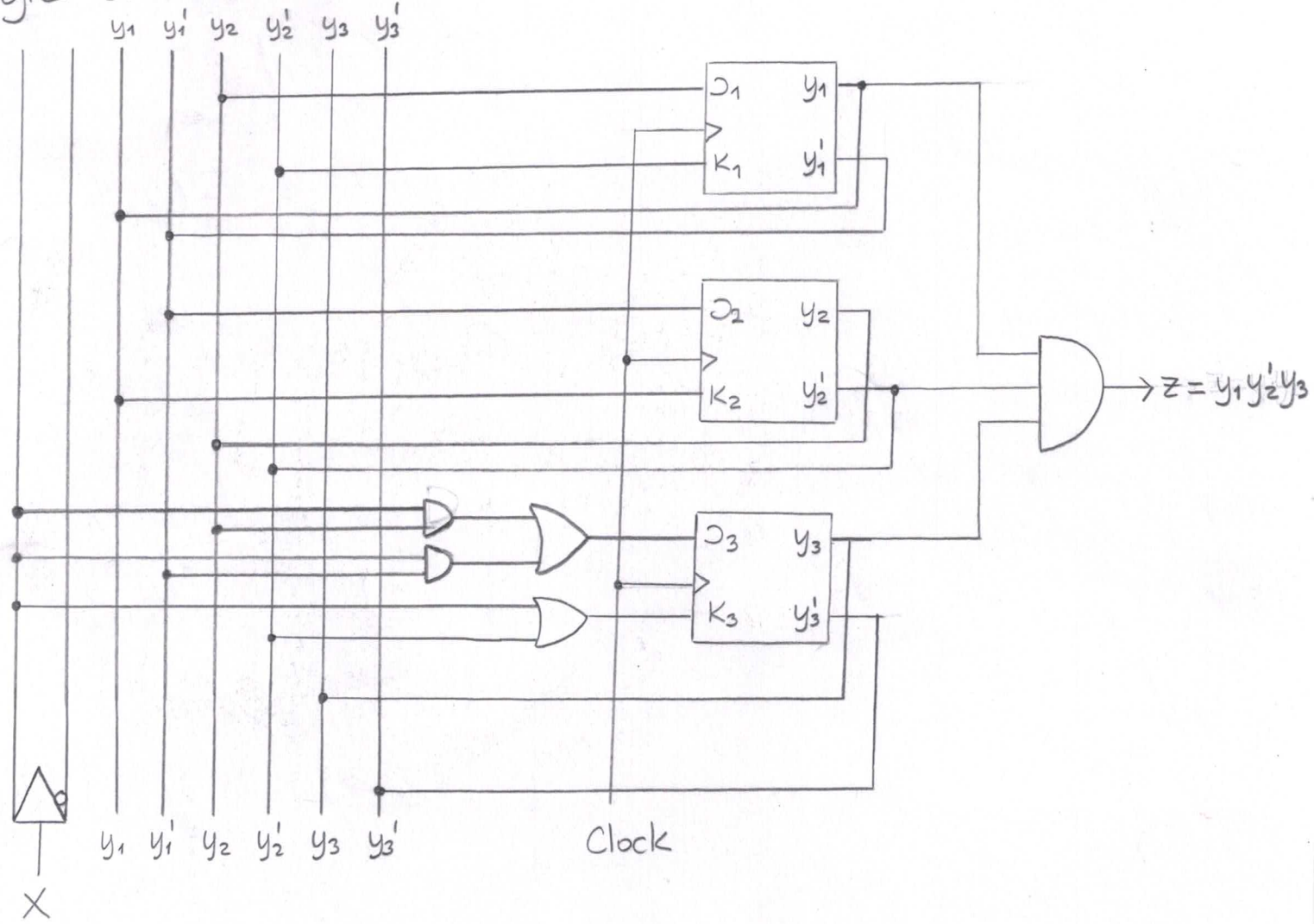
Hence;

-the state equations of the circuit are obtained

$$Y_i = f_i(x_1, \ldots, x_n, y_1, \cdots, y_s), \quad i = 1, \ldots, s$$
$$z_j = f_j(x_1, \ldots, x_n, y_1, \cdots, y_s), \quad j = 1, \ldots, m$$

Step 3. Using state equations, the state table and state diagram are obtained.

Example. Let us consider the following sequential logic circuit



Step 1. Let us write eqn.'s for J_i, K_i $(i=1,2,3)$ and z

$$J_1 = y_2 \;,\; K_1 = y_2' \;,\; J_2 = y_1' \;,\; K_2 = y_1 \;,$$

$$J_3 = x y_1' + x y_2 \;,\; K_3 = x + y_2' \;,\; z = y_1 y_2' y_3$$

Step 2. Let us find equations for Y_i

$$Y_1 = J_1 y_1' + K_1' y_1 = y_2 y_1' + y_2 y_1 = y_2 (y_1' + y_1) = y_2$$

$$Y_2 = J_2 y_2' + K_2' y_2 = y_1' y_2' + y_1' y_2 = y_1' (y_2' + y_2) = y_1'$$

$$Y_3 = J_3 y_3' + K_3' y_3 = (x y_1' + x y_2) y_3' + (x + y_2')' y_3$$

$$= x y_1' y_3' + x y_2 y_3' + x' y_2 y_3 = x y_3' (y_1' + y_2) + x' y_2 y_3$$

Hence, the state equations are given by

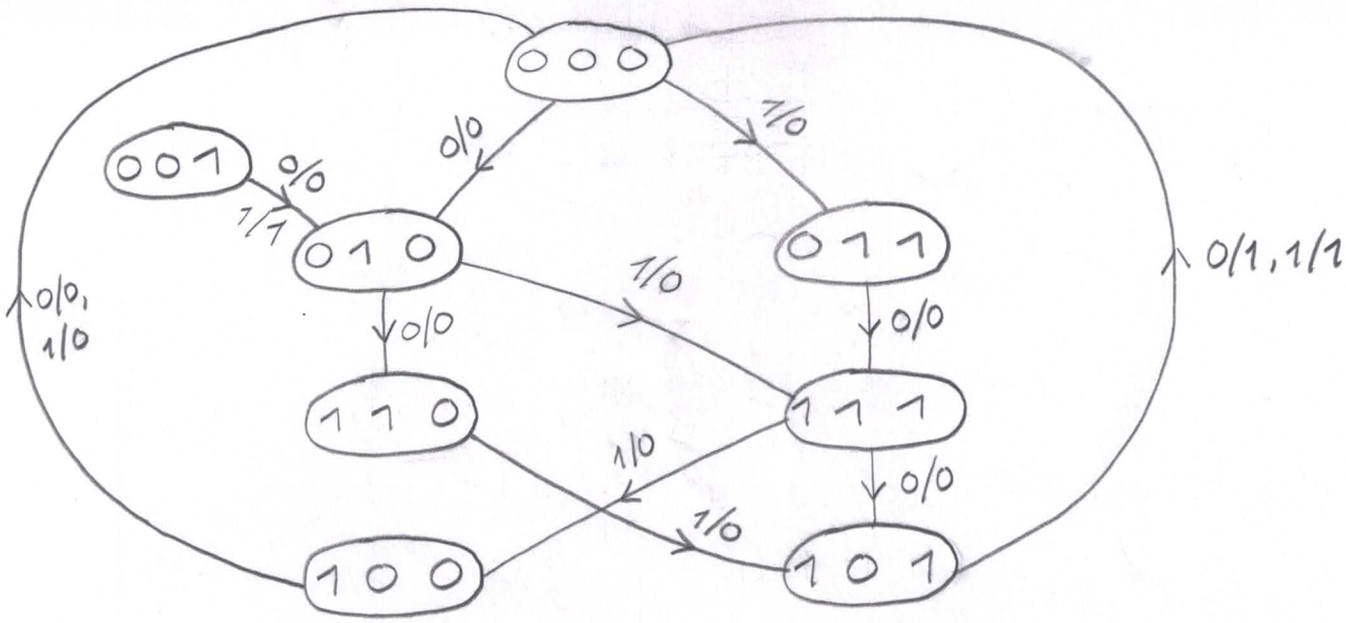
$$Y_1 = y_2 \;,\; Y_2 = y_1' \;,\; Y_3 = x y_3' (y_1' + y_2) + x' y_2 y_3$$

-then the state table is tabulated as follows

$y_1 y_2 y_3$ \ x	0	1
0 0 0	010, 0	011, 0
0 0 1	010, 0	010, 0
0 1 0	110, 0	111, 0
0 1 1	111, 0	110, 0
1 0 0	000, 0	000, 0
1 0 1	000, 1	000, 1
1 1 0	100, 0	101, 0
1 1 1	101, 0	100, 0

$Y_1 Y_2 Y_3, z$

-and based on the state table, we can draw the state diagram as follows



- Note that this circuit corresponds to a parity bit generator

Definition . If the circuit can not return back to the used states when it goes to an unused state, then it is said to be a locked type of circuit or machine

- The parity bit generator circuit is NOT of locked type
  - (L) because when it goes to the unused state 001, it can return back to a used state

Synthesis of sequential logic circuits

- There exist different kinds of methods for the design of sequential logic circuits
- We will introduce one of such methods whose intermediate steps are outlined as follows:

1. Description of the problem statement verbally
2. Construction of the state diagram
3. Reduction of the state diagram
4. Assignment of codes to the states
5. Tabulation of the truth table corresponding to the combinational logic circuit part and construction of the circuit diagram

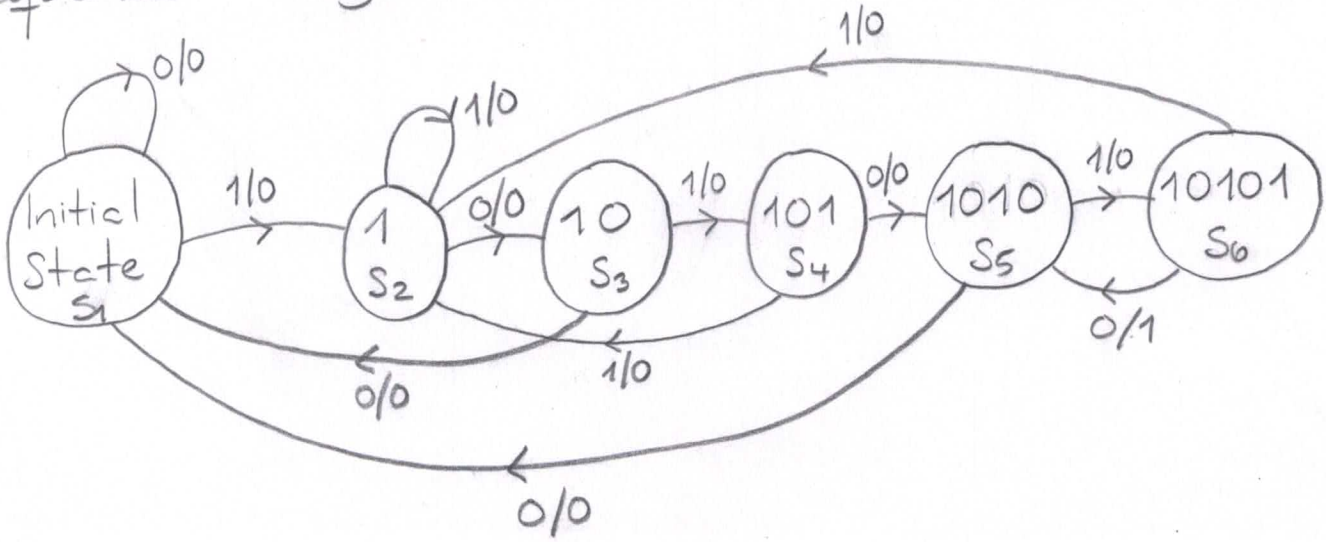
Definition. The circuit whose output is equal to 1 if the last k inputs correspond to a certain sequence, otherwise 0, is referred to a sequence recognizer (detector).

Example. If the last 6 inputs

$x_1 x_2 x_3 x_4 x_5 x_6 = 101010$ then $z = 1$ otherwise

$z = 0$

- we can find out the state diagram of this sequence recognizer as follows



- it is an expected outcome that there exist six different states

  (1) since the length of the sequence is six

Note that ;

- we have the following observation:

  - When state machine is in state S_6, if the input is 0, the output is 1 as the sequence is recognized. However, the state machine goes to state S_5 instead of initial state S_1

  Because;

  - if the 2 inputs that come after the sequence is recognized are 1,0 then the last 6 inputs become 101010 once again and output becomes 1 again.

Theorem 14. The reduced expressions for the pair of inputs $J_i(x_1, \dots, x_n, y_1, \dots, y_s)$ and $K_i(x_1, \dots, x_n, y_1, \dots, y_s)$ of a JK type flip-flop do not involve the variable y_i.

Proof. We have

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

J, K

-it follows from the inverse describing function that we have

$$Y = Jy' + K'y$$

$$\Rightarrow \quad J = \begin{cases} Y, & y = 0 \\ d, & y = 1 \end{cases}, \quad K = \begin{cases} Y', & y = 1 \\ d, & y = 0 \end{cases}$$

Hence ;

-the reduced SOP expression for J_i is obtained as

$$J_i = A_1 y_i' + A_2 y_i' + \cdots + A_k y_i' + d(A_1 y_i + \cdots + A_k y_i)$$

-if we choose $d = 1$, then we set

$$J_i = A_1 + \cdots + A_k$$

(L) the reduced expression does not involve y_i

Similarly ;

$$K_i = B_1 y_i + \cdots + B_m y_i + d(B_1 y_i' + \cdots + B_m y_i')$$

and $d = 1 \Rightarrow K_i = B_1 + \cdots + B_m$

(L) no y_i available in the reduced expression

Remark. Theorem 15 outlines the advantage of JK type flip-flops as it simplifies to set reduced