Design of sequential circuits (continued)

- once the verbal description of the problem is given
  - ↳ the state diagram is derived
- the # of states required is desired to be minimum
  - ↳ which provides a minimum number of storage elements to be used in the logic circuit of the task to be implemented
- this is so called as minimal realization problem in analog and discrete-time circuits

state reduction

- the number of states is minimized by using the method of state reduction

Namely;

- the state diagram corresponding to a circuit which accomplishes the task
  - ↳ with minimum states is obtained
- the state reduction will NOT be investigated within the scope of this course

state coding

- if the number of states in the state diagram is n, then the states are encoded

↳ by matching a binary code with

$s := \lceil \log_2 n \rceil$ bits

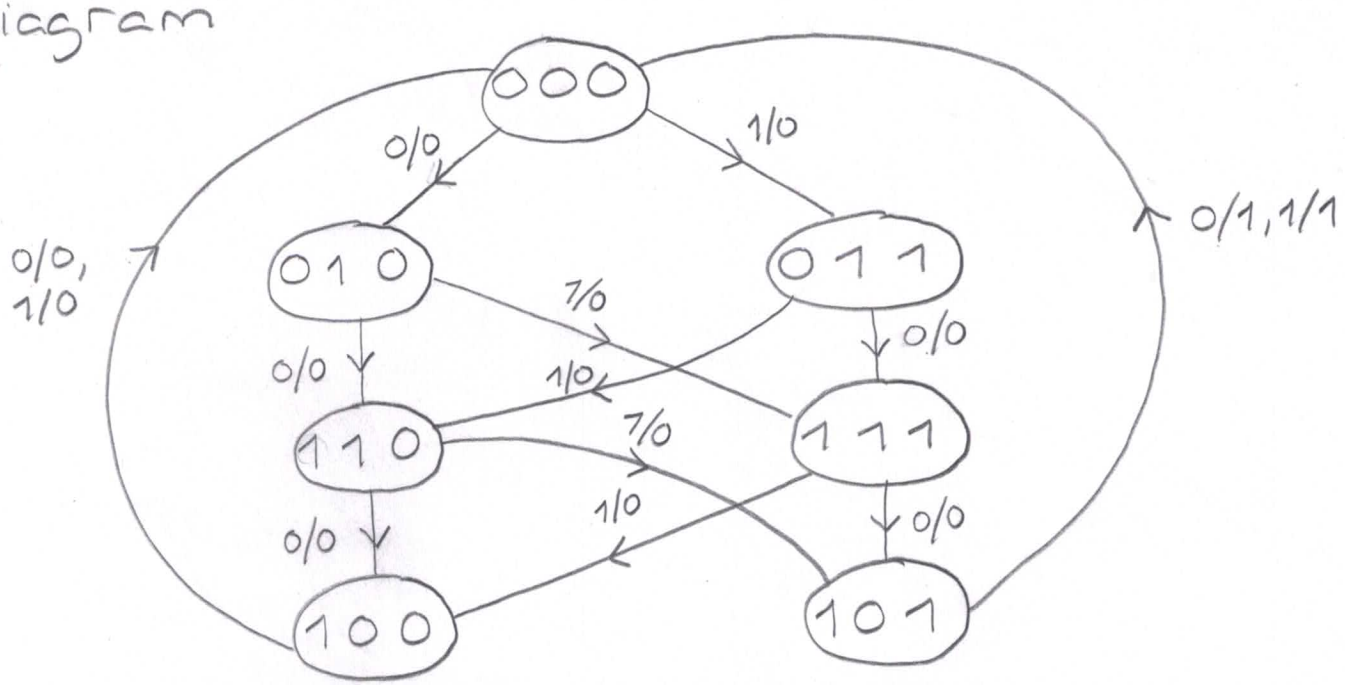
- if the combinational logic circuit part will not be implemented using PROM then

↳ an appropriate way of encoding minimizes the complexity of the design

However;

- in this course we will NOT investigate the optimal form of state encoding within the scope of this course

Example. Consider the following state diagram



Design the sequential logic circuit using JK type flip-flops.

- the 1st step is to tabulate the flip-flop inputs and outputs
  - ↳ based on the state diagram
- this is so called as the truth table corresponding to the state diagram

x	y_1	y_2	y_3	Y_1	Y_2	Y_3	J_1	K_1	J_2	K_2	J_3	K_3	z
0	0	0	0	0	1	0	0	d	1	d	0	d	0
1	0	0	0	0	1	1	0	d	1	d	1	d	0
0	0	1	0	1	1	0	1	d	d	0	0	d	0
1	0	1	0	1	1	1	1	d	d	0	1	d	0
0	0	1	1	1	1	1	1	d	d	0	d	0	0
1	0	1	1	1	1	0	1	d	d	0	d	1	0
0	1	1	0	1	0	0	d	0	d	1	0	d	0
1	1	1	0	1	0	1	d	0	d	1	1	d	0
0	1	1	1	1	0	1	d	0	d	1	d	0	0
1	1	1	1	1	0	0	d	0	d	1	d	1	0
0	1	0	0	0	0	0	d	1	0	d	0	d	0
1	1	0	0	0	0	0	d	1	0	d	0	d	0
0	1	0	1	0	0	0	d	1	0	d	d	1	1
1	1	0	1	0	0	0	d	1	0	d	d	1	1

Remark. If D type flip-flops are employed, then the columns for D inputs are not listed in the table
  - ↳ since D input information is available in the column of Y

- if the combinational logic circuit part will be implemented using two-level AND-OR circuit,

↳ then the flip-flop input functions and output function are optimized separately

- employing Karnaugh map method, $\mathcal{J}_i$, K_i, $i = 1,2,3$ are minimized in sum of products (SOP) form

x \ y_2y_3	00	01	11	10
0	0	d	1	1
1	0	d	1	1

$\mathcal{J}_1 = y_2$

x \ y_2y_3	00	01	11	10
0	1	1	0	0
1	1	1	0	0

$K_1 = y_2'$

x \ y_1y_3	00	01	11	10
0	1	d	0	0
1	1	d	0	0

$\mathcal{J}_2 = y_1'$

x \ y_1y_3	00	01	11	10
0	0	0	1	1
1	0	0	1	1

$K_2 = y_1$

x \ y_1y_2	00	01	11	10
0	0	0	0	0
1	1	1	1	0

$\mathcal{J}_3 = xy_1' + xy_2$

x \ y_1y_2	00	01	11	10
0	d	0	0	1
1	d	1	1	1

$K_3 = x + y_2'$

y_2y_3 \ xy_1	00	01	11	10
00	0	0	0	0
01	d	1	1	d
11	0	0	0	0
10	0	0	0	0

$z = y_2'y_3$

Completion of lacking states and checking locked-type circuit

- note that in the former example, we have 7 states

However;

- if we analyze the sequential circuit, then
  - ↳ we find that the state diagram involves 8 states
- in general, let n denote the number of states which are encoded by s bits, then if

$$n < 2^s$$

- we need to define the lacking $(2^s - n)$ states
  - ↳ and check whether the circuit is of locked type or not
- For example, in our case, the lacking state is 001, i.e.

$y_1 = 0$, $y_2 = 0$, $y_3 = 1$

$J_1 = 0$, $K_1 = 1$ ⇓ $Y_1 = 0$ $x = 0,1$

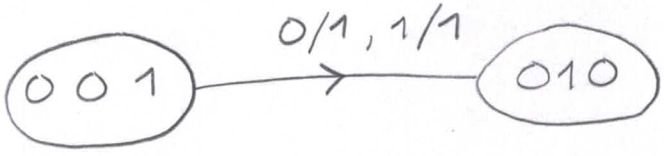
$J_2 = 1$, $K_2 = 0$ ⇓ $Y_2 = 1$ $x = 0,1$

$J_3 = 0$, $K_3 = 1$, $x = 0$ ⇓ $Y_3 = 0$ $x = 0$

$J_3 = 1$, $K_3 = 1$, $x = 1$ ⇓ $Y_3 = 0$ $x = 1$

, $z = y_2' y_3 = 1 \cdot 1 = 1$

Hence ;



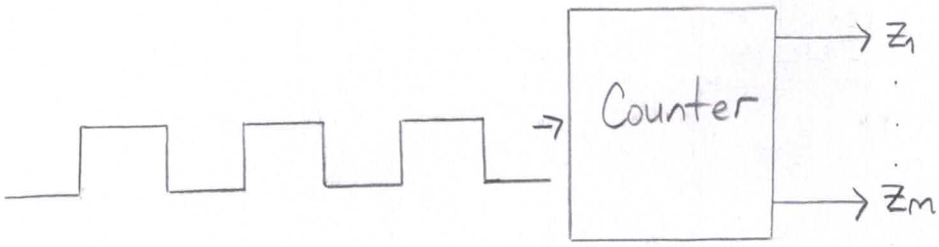
- As a result, we see that the circuit is NOT of a locked type

Special sequential logic circuits

- Memory elements, registers, and counters are some examples of sequential logic circuits
  - ↳ which are often used in digital systems
- within the scope of this course, we investigate only counters

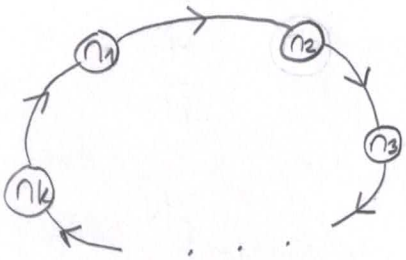
Counters

- In general, a counter is a sequential circuit with a single input and n outputs
- the input of the circuit is specified by the pulse that will be counted,
- and the output of the circuit is specified by the outputs of the flip-flops involved



"general structure of a counter"

- at the time instant of i, the output is equivalent to $n_i = y_{n-1}\, y_{n-2} \cdots y_1\, y_0$



$$0 \leq n_i \leq 2^k - 1$$

Definition. Let n_i denote the output at time instant of i. If $n_{i+1} = n_i + 1$, then the counter is so called as "up" counter. Moreover, if the initial output is 0, i.e. $n_0 = 0$ and if the output, n_k returns back to 0, the counter is referred to as "modulo-n_{k+1}" counter. If $n_{i+1} = n_i - 1$ then the counter is called "down" counter.

Definition. If the pulse input to be counted is applied to the inputs of all flip-flops simultaneously, then the counter is referred to as a "synchronous" counter. Otherwise, the counter is called "asynchronous" counter.