Design of Clocked Synchronous Sequential Circuits

Design of a sequential circuit starts with the verbal description of the problem (requirements, scenario).

Design process is similar to computer programming.

First, the problem in the physical (real) world should be described and appropriately modeled.

Then the circuit should be designed to solve the problem.

Design of a sequential circuit has the following steps:

1. Verbal description of the problem (**functional requirements** of the circuit). Timing diagrams can be used to avoid uncertainties.
2. The design **model** (Mealy or Moore) of the circuit is determined.
3. The **states** of the FSM are determined.
State transitions according to the inputs and current states are determined.
State transition and output tables are formed. State transition diagrams can be used if they will make the design easier.
State reduction is performed (if applicable). The purpose is to build a correctly functioning machine with the least possible number of states.
This step is similar to computer programming; that is why an intuitional approach is required.

Steps of sequential circuit design (cont'd)

4. **Coding States**: Binary codes are assigned to the states.
If there are **n** states, the number of variables (number of flip-flops) **m** is computed as follows:

$$m = \lceil \log_2 n \rceil$$

Here $\lceil x \rceil$ denotes ceiling function. For example $\lceil 4.1 \rceil = 5$ and $\lceil 4.0 \rceil = 4$

State transition and output table is formed using state variables.

5. Type of **flip-flop** is determined.
6. Using the flip-flop transition tables, inputs of the flip-flops are determined. **Function (F)** that drives flip-flops is obtained.
7. From the output table, **output function (G)** is obtained.
8. **Combinational circuits of the functions** (F and G) are designed and implemented with the minimum cost.

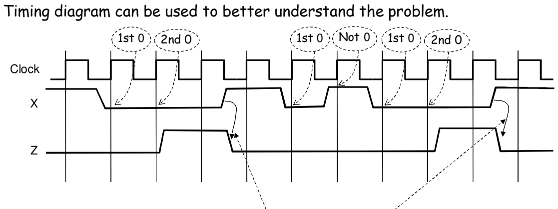
Synchronous Circuit Design Example:

Problem:

A sequential circuit with a single input (X) and single output (Z) will be designed.

After two sequential (consecutive) "0"s at the input, the output should be "1" as long as the input is "0".

Timing diagram can be used to better understand the problem.



Since the output is affected by the input directly (without the active edge of the clock signal), the design must be in the **Mealy model**.

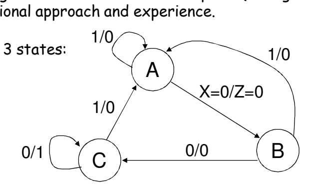
1. Deriving the state transition diagram from the verbal description (timing diagrams). This step requires intuitional approach and experience.

This Machine can be designed with 3 states:

A: No zeros are arrived

B: First zero is arrived

C: Second zero is arrived



2. State, output table

S \ X (S^+,Z)	0	1
A	B,0	A,0
B	C,0	A,0
C	C,1	A,0

Current States — Next States, Outputs

State coding:
A: 00
B: 01 (Alternative coding is possible)
C: 11

State variables:
Q_1 , Q_0

State transition, output table: (in Karnaugh map format)

Q_1Q_0 \ X ($Q_1^+Q_0^+,Z$)	0	1
00	01,0	00,0
01	11,0	00,0
11	11,1	00,0
10	øø,ø	øø,ø

Gray Code

Alternative state coding is possible. For example, A:00, B: 10, C:01.

In this case the internal structure of the circuit will be different. However, the functionality of the circuit will be the same.

3. Transitions of state variables are determined:

Using the state transition table of the circuit, transitions of each state variable (flip-flop) are determined separately.

State transition table

Q_1Q_0 \ X ($Q_1^+Q_0^+,Z$)	0	1
00	01,0	00,0
01	11,0	00,0
11	11,1	00,0
10	øø,ø	øø,ø

Q_1 transitions: ($Q_1 \rightarrow Q_1^+$)

Q_1Q_0 \ X ($Q_1Q_1^+$)	0	1
00	00	00
01	01	00
11	11	10
10	ø	ø

Q_0 transitions: ($Q_0 \rightarrow Q_0^+$)

Q_1Q_0 \ X ($Q_0Q_0^+$)	0	1
00	01	00
01	11	10
11	11	10
10	ø	ø

To simplify notation, symbolic names are assigned to transitions and tables are re-organized with the symbols.

symbol	QQ^+
0	00
α	01
β	10
1	11

Q_1Q_0 \ X ($Q_1Q_1^+$)	0	1
00	0	0
01	α	0
11	1	β
10	ø	ø

Q_1Q_0 \ X ($Q_0Q_0^+$)	0	1
00	α	0
01	1	β
11	1	β
10	ø	ø

Transition of each state variable (flip-flop) for each input value and state is determined.

4. Determining the input functions of the flip-flops:

D flip-flops will be used in this example.

In the previous (3.) step, transitions were determined for all flip-flops.

In this step, the input values of the flip-flops for a required transition will be investigated.

The transition table of the flip-flop will be used for this purpose.

D flip-flop transition table:

symbol	QQ^+	D
0	00	0
α	01	1
β	10	0
1	11	1

This table shows the inputs of the D flip-flop for each transition.

Different types of flip-flops have different transition tables.

Transition table of D flip-flop is simple. The input value of the D flip-flop is equal to the next value of its state variable.

The required input values of the flip-flops are calculated using the transition tables.

Q_1 transitions ($Q_1 \rightarrow Q_1^+$) :

Q_1Q_0 \ X ($Q_1Q_1^+$)	0	1
00	0	0
01	α	0
11	1	β
10	ø	ø

Q_0 transitions ($Q_0 \rightarrow Q_0^+$) :

Q_1Q_0 \ X ($Q_0Q_0^+$)	0	1
00	α	0
01	1	β
11	1	β
10	ø	ø

D flip-flop transition table:

symbol	QQ+	D
0	00	0
α	01	1
β	10	0
1	11	1

Input of D_1 :

Q_1Q_0 \ X (D_1)	0	1
00	0	0
01	1	0
11	1	0
10	ø	ø

Input of D_0 :

Q_1Q_0 \ X (D_0)	0	1
00	1	0
01	1	0
11	1	0
10	ø	ø

To obtain expressions easily, tables are formed as Karnaugh maps.

Rows and columns are in Gray code.

$D_1 = X'Q_0$ $\quad D_0 = X'$ $\quad \{D_1, D_0\} = \mathbf{F}$(Input "X" , State "$Q_i$")

The function **F** that drives the inputs of the flip-flops is obtained.

5. Using the output table, output function (G) is obtained.

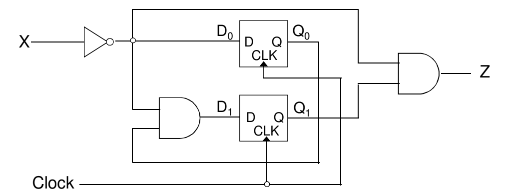
Q_1Q_0 \ X (Z)	0	1
00	0	0
01	0	0
11	1	0
10	ø	ø

During the design of functions F and G, design methods for combinational circuits (prime implicants, prime implicant chart, minimization) that are covered in the first part of the course should be used.

There is no need to minimize the functions in this example because they are simple.

$Z = X'Q_1$ $\quad Z = \mathbf{G}$(Input "X" , State "Q_i")

6. Implementation and drawing of the designed circuit using logical gates.



Example: Same circuit designed with JK flip-flops

The first three steps are the same.

4. Positive edge triggered JK flip-flops will be used.

JK flip-flop transition table:

symbol	QQ+	J	K
0	00	0	ø
α	01	1	ø
β	10	ø	1
1	11	ø	0

Using JK flip-flops instead of D flip-flops can produce simpler logical functions for the next state.

As the functions in this example are already simple, no further simplification is achieved.

The transitions of state variables were determined from the state transition table in step 3.

Q_1Q_0 \ X ($Q_1^+Q_0^+$,Z)	0	1
00	01,0	00,0
01	11,0	00,0
11	11,1	00,0
10	øø,ø	øø,ø

Q_1 transitions ($Q_1 \rightarrow Q_1^+$):

Q_1Q_0 \ X ($Q_1Q_1^+$)	0	1
00	0	0
01	α	0
11	1	β
10	ø	ø

Q_0 transitions ($Q_0 \rightarrow Q_0^+$):

Q_1Q_0 \ X ($Q_0Q_0^+$)	0	1
00	α	0
01	1	β
11	1	β
10	ø	ø

The required input values of the flip-flops are calculated using the transition tables.

Q_1 transitions:

Q_1Q_0 \ X ($Q_1Q_1^+$)	0	1
00	0	0
01	α	0
11	1	β
10	ø	ø

Q_0 transitions:

Q_1Q_0 \ X ($Q_0Q_0^+$)	0	1
00	α	0
01	1	β
11	1	β
10	ø	ø

Transition table for JK flip-flop:

symbol	QQ+	J	K
0	00	0	ø
α	01	1	ø
β	10	ø	1
1	11	ø	0

Q_1Q_0 \ X (J_1)	0	1
00	0	0
01	1	0
11	ø	ø
10	ø	ø

Q_1Q_0 \ X (K_1)	0	1
00	ø	ø
01	ø	ø
11	0	1
10	ø	ø

Q_1Q_0 \ X (J_0)	0	1
00	1	0
01	ø	ø
11	ø	ø
10	ø	ø

Q_1Q_0 \ X (K_0)	0	1
00	ø	ø
01	0	1
11	0	1
10	ø	ø

$J_1 = X'Q_0$ $\quad K_1 = X$ $\quad J_0 = X'$ $\quad K_0 = X$

$\{J_1, K_1, J_0, K_0\} = F(X, Q_1, Q_0)$

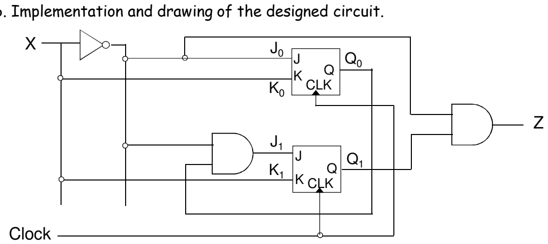
The function F that drives the inputs of the flip-flops is obtained.

5. The output function (G) is determined using output table.

Q_1Q_0 \ X (Z)	0	1
00	0	0
01	0	0
11	1	0
10	ø	ø

$Z = X'Q_1$

6. Implementation and drawing of the designed circuit.



Transition tables for flip-flops:

Transition tables for different types of flip-flops are given below.

Transition table for SR flip-flop :

symbol	QQ+	S	R
0	00	0	ø
α	01	1	0
β	10	0	1
1	11	ø	0

Transition table for JK flip-flop :

symbol	QQ+	J	K
0	00	0	ø
α	01	1	ø
β	10	ø	1
1	11	ø	0

Transition table for D flip-flop:

symbol	QQ+	D
0	00	0
α	01	1
β	10	0
1	11	1

Transition table for T flip-flop :

symbol	QQ+	T
0	00	0
α	01	1
β	10	1
1	11	0

Synchronous Circuit Design Example 2: Moore Model

Design using the Moore model has the same design stages that are already shown. Important points are:

- outputs depend ONLY on the states,
- because of this, each state corresponds to a single output.

Problem:

A synchronous sequential circuit with 2 inputs (X,Y) and a single output (Z) will be designed.
If the number of incoming 1s to the input is a multiple of 4, the output of the circuit is 1. Otherwise the output should be 0. If there is no incoming 1 (the number of 1s is zero), the output should be 1.

Solution:

The circuit should perform *modulo 4* operation, and if the result of the operation is 0, the output should be 1. This FSM can be implemented with 4 states:

1.	Modulo 0: S0	Output = 1	number of incoming 1s mod 4 = 0
2.	Modulo 1: S1	Output = 0	number of incoming 1s mod 4 = 1
3.	Modulo 2: S2	Output = 0	number of incoming 1s mod 4 = 2
4.	Modulo 3: S3	Output = 0	number of incoming 1s mod 4 = 3

State/output table:

Meaning	S \ XY (S^+)	00	01	11	10	Z
Modulo 0	S0	S0	S1	S2	S1	1
Modulo 1	S1	S1	S2	S3	S2	0
Modulo 2	S2	S2	S3	S0	S3	0
Modulo 3	S3	S3	S0	S1	S0	0

State coding:
S0: 00
S1: 01
S2: 11
S3: 10

State variables:
Q1, Q0

Coded state/output table:

Q1Q0 \ XY ($Q1^+Q0^+$)	00	01	11	10	Z
00	00	01	11	01	1
01	01	11	10	11	0
11	11	10	00	10	0
10	10	00	01	00	0

D1 (Q0·X'·Y, Q1'·X·Y, Q1·X'·Y', Q0·X·Y'):

Q1 Q0 \ XY	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	1	1	0	1
10	1	0	0	0

D0 (Q1'·X'·Y, Q1'·X·Y', Q0·X'·Y', Q0'·X·Y):

Q1 Q0 \ XY	00	01	11	10
00	0	1	1	1
01	1	1	0	1
11	1	0	0	0
10	0	0	1	0

The characteristic function of the D flip-flop is used ($Q^+=D$) thus $D_1=Q_1^+$, $D_0=Q_0^+$

D1= Q0·X'·Y + Q1'·X·Y + Q1·X'·Y' + Q0·X·Y'
D0= Q1'·X'·Y + Q1'·X·Y' + Q0·X'·Y' + Q0'·X·Y

Z= Q1' · Q0'

Using Multiplexers for Synchronous Circuit Implementation

If a synchronous sequential circuit is designed using D flip-flops, simpler implementations are possible, if the inputs of the flip-flops are driven with multiplexers.

In this method,

- Input of each D flip-flop is driven by a separate multiplexer.
- The state variables (flip-flop outputs) are connected to the selection inputs of the multiplexers.

Therefore, each multiplexer selects its inputs according to the current state.

- The inputs of the multiplexer should have the necessary values that produces the next state of the machine.
- The inputs of the multiplexers are obtained from the rows of the state transition table.

The same circuit designed in the previous example will be redesigned using multiplexers.

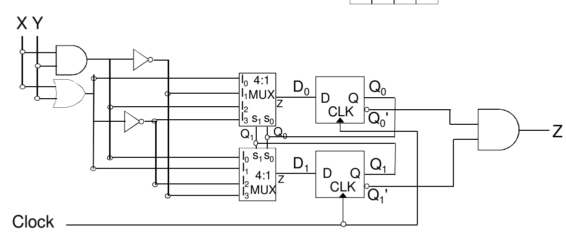
D input values of the flip-flop: (From the previous example)

D1:

Q1 Q0 \ XY	00	01	11	10	to multiplexer:
00	0	0	1	0	$X \cdot Y$
01	0	1	1	1	$X + Y$
11	1	1	0	1	$\overline{(X \cdot Y)}$
10	1	0	0	0	$\overline{(X + Y)}$

D0:

Q1 Q0 \ XY	00	01	11	10	to multiplexer :
00	0	1	1	1	$X + Y$
01	1	1	0	1	$\overline{(X \cdot Y)}$
11	1	0	0	0	$\overline{(X + Y)}$
10	0	0	1	0	$X \cdot Y$



Counter Design

Counters that count at each active clock signal with a certain sequence can be designed as a synchronous sequential circuit.

Moore model is appropriate for counter design.

Each output value of the counter can be treated as a different state.

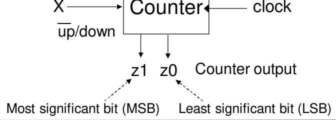
Outputs can be obtained directly from the state variables O=S.
It means that the design must be in Moore Model.

Example:

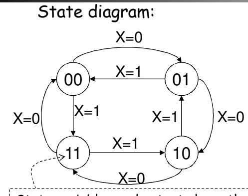
Design the counter shown below that has a single control input (X).

Counter should count at each rising edge of the clock signal in the natural order of 0-1-2-3. The counter should go back to 0 after 3.

When X=0 it should count up, if X=1 it should count down.



State diagram:



State variables and outputs have the same values. O = S

State table:

Q_1Q_0 \ X ($Q_1^+Q_0^+$)	0	1
00	01	11
01	10	00
11	00	10
10	11	01

Rows are replaced according to the Gray code so that state table can also be used as a Karnaugh diagram.

Designing the counter using D flip-flops:

Recall:
$Q^+=D$

Therefore
$D_1=Q_1^+$
$D_0=Q_0^+$

D_1:

Q_1Q_0 \ X	0	1
00	0	1
01	1	0
11	0	1
10	1	0

D1 = X'·(Q1⊕Q0) + X·(Q1⊕Q0)'
D1 = X⊕Q1⊕Q0

D_0:

Q_1Q_0 \ X	0	1
00	1	1
01	0	0
11	0	0
10	1	1

D0 = Q0'

Output:

Z0 = Q0
Z1 = Q1

Logical gates (AND, OR, XOR etc.) or multiplexers can be used to implement counters.

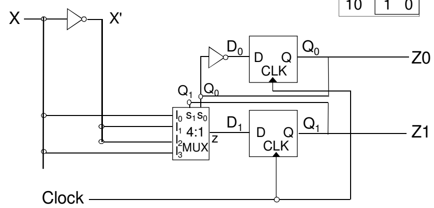
Below, since D0 input has a very simple expression (D0 = Q0') a logical gate (single inverter) is preferred instead of a multiplexer.

A multiplexer is used to drive input D1.

Remember: State variables (Q_1Q_0) will be connected to the selection inputs of the multiplexer.

D_1

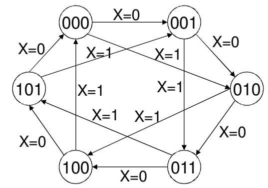
Q_1Q_0 \ X	0	1	to multiplexer:
00	0	1	X
01	1	0	X'
11	0	1	X
10	1	0	X'



Example:

Design a counter that counts in 0-1-2-3-4-5 sequence and has a single control input (X).

If X=0 count up by one, if X=1 count up by 2.



State table:

$Q_2^+Q_1^+Q_0^+$

$Q_2Q_1Q_0$ \ X	0	1
000	001	010
001	010	011
010	011	100
011	100	101
100	101	000
101	000	001
110	ØØØ	ØØØ
111	ØØØ	ØØØ

Q_0 is moved to the columns.

State table is organized as a Karnaugh diagram:

$Q_2^+Q_1^+Q_0^+$

Q_2Q_1 \ Q_0X	00	01	11	10
00	001	010	011	010
01	011	100	101	100
11	ØØØ	ØØØ	ØØØ	ØØØ
10	101	000	001	000

In this example, T flip-flops will be used.

Remember:

Transition table for T flip-flop:

symbol	QQ^+	T
0	00	0
α	01	1
β	10	1
1	11	0

$Q_2^+Q_1^+Q_0^+$

Q_2Q_1 \ Q_0X	00	01	11	10
00	001	010	011	010
01	011	100	101	100
11	ØØØ	ØØØ	ØØØ	ØØØ
10	101	000	001	000

By considering transitions ($Q_2 \rightarrow Q_2^+$, $Q_1 \rightarrow Q_1^+$, $Q_0 \rightarrow Q_0^+$) we determine T_2 , T_1 , T_0 .

T_2

Q_2Q_1 \ Q_0X	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	Ø	Ø	Ø	Ø
10	0	1	1	1

T_1

Q_2Q_1 \ Q_0X	00	01	11	10
00	0	1	1	1
01	0	1	1	1
11	Ø	Ø	Ø	Ø
10	0	0	0	0

T_0

Q_2Q_1 \ Q_0X	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	Ø	Ø	Ø	Ø
10	1	0	0	1

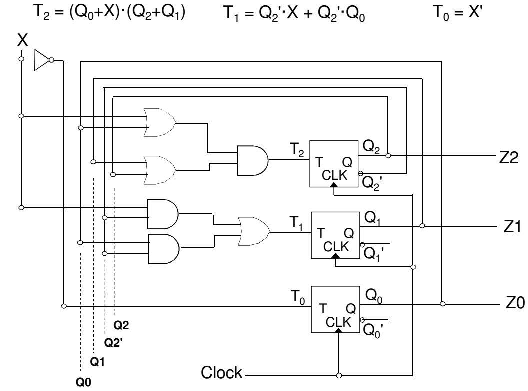
$T_2' = Q_0' \cdot X' + Q_2' \cdot Q_1'$

$T_2 = (Q_0+X) \cdot (Q_2+Q_1)$

$T_1 = Q_2' \cdot X + Q_2' \cdot Q_0$

$T_0 = X'$

$T_2 = (Q_0+X) \cdot (Q_2+Q_1)$ $T_1 = Q_2' \cdot X + Q_2' \cdot Q_0$ $T_0 = X'$



Implementation of synchronous circuits using PAL

As combinational circuits, it is also possible to use PAL units to implement synchronous circuits.

For this purpose, PAL units that include flip-flops are used.

At the right, a **16R8** PAL circuit is shown.