Design of Clocked Synchronous Sequential Circuits

The design of a sequential circuit starts with the problem statement which specifies the desired relationship between the input and output sequences (scenario).

The process of designing a circuit to perform a given logical function is quite similar to the process of designing a computer program to perform a given task.

First, we should describe and appropriately model the real-world problem.

Then, we should design a circuit to solve the problem.

Designing a sequential circuit consists of the following steps:

1. We describe the problem (**functional requirements** of the circuit) verbally. We can use timing diagrams to avoid uncertainties.
2. We decide which design **model** (Mealy/Moore) would better represent the circuit.
3. We determine the **states** that will make up the finite state machine (FSM).
   a) We determine the state transitions based on the inputs and current states.
   b) We construct the state transition and output tables. We can use a state diagram if it makes the design easier.
   c) We reduce the number of states in the state table (if applicable). The purpose is to build a correctly functioning machine with the fewest possible number of states.
   d) This process is similar to the process of designing a computer program; that is why it requires an intuitional approach.

Steps of sequential circuit design (cont'd)

4. **Assigning codes to each state:** A binary code is assigned to each state. If there are n states, the number of variables (number of flip-flops) m is computed as follows:

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

   where $\lceil x \rceil$ denotes the ceiling function. For example, $\lceil 4.1 \rceil = 5$ and $\lceil 4.0 \rceil = 4$.
5. We construct the state transition and output table based on the values of the state variables.
6. We decide what type of **flip-flops** we will use.
7. Using the transition table for the selected flip-flop type, we determine the inputs of the flip-flops. We obtain the **function (F)** that drives the flip-flops.
8. From the output table, we obtain the **output function (G)**.
9. We design **combinational circuits for the functions** (F and G) and implement each with the minimum cost.

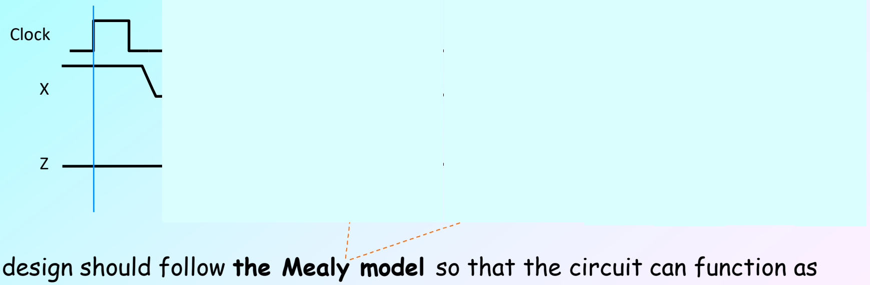
Synchronous Circuit Design Example:

Problem:

We will design a sequential circuit with a single input (X) and single output (Z).

After the input remains at "0" for two consecutive clock cycles, the output will be "1" as long as a "0" is read at the input.

We can use a timing diagram to show the state machine's expected behavior for a sequence of inputs.



The design should follow **the Mealy model** so that the circuit can function as shown in the above timing diagram.

This is because the output which corresponds to a given input appears immediately following the application of that input (before the active edge of the clock signal).

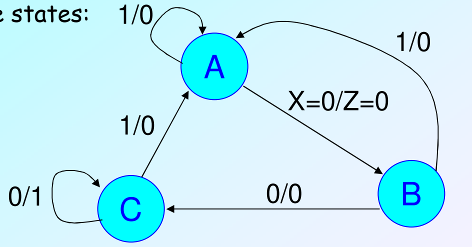
1. **We construct the state diagram** based on the problem statement (timing diagrams). This step requires an intuitional approach and experience.

   We can design the machine using three states:

   A: No zeros have been received

   B: First zero has been received

   C: Second zero has been received



2. **We construct the state, output table.**

S^+,Z

X S	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

C: 11

(Alternative coding is possible)

State variables:

Q_1 , Q_0

State transition, output table:

(in Karnaugh map format)

$Q_1^+Q_0^+,Z$

X Q_1Q_0	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 codings are possible. For example, A:00, B: 10, C:01.

In that case, the internal structure of the circuit will be different.

However, the functionality of the circuit will be the same.

3. We determine the transitions of state variables:

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

For this solution, we need two flip-flops, i.e., Q_1 and Q_2.

State transition table

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

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

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

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

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

To simplify notation, we assign symbolic names to transitions and reorganize the tables with these symbols.

symbol	QQ^+
0	00
α	01
β	10
1	11

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

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

Thus, we have determined what transition each state variable (flip-flop) will make for each input value and state.

4. We determine the input functions of the flip-flops:

We will use D flip-flops in this example.

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

In this step, we will investigate the values that must be applied to the inputs of the flip-flops to make the required transitions.

We will use the transition table of the flip-flop 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 value that must be applied to the input of a D flip-flop for a given transition.

Different types of flip-flops have different transition tables.

The transition table of the D flip-flop is simple. The value that must be applied to the input of the D flip-flop is equal to the next value of its state variable.

We derive the required inputs for the flip-flops using the transition tables.

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

$Q_1Q_1^+$ X / Q_1Q_0	0	1
00	0	0
01	α	0
11	1	β
10	ø	ø

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

$Q_0Q_0^+$ X / Q_1Q_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 :

D_1 X / Q_1Q_0	0	1
00	0	0
01	1	0
11	1	0
10	ø	ø

Input of D_0 :

D_0 X / Q_1Q_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 follow the order of **the Gray code**.

$D_1 = X'Q_0$ $D_0 = X'$ $\{D_1 , D_0\} = \mathbf{F}(\text{Input "X" , State "Q}_i\text{"})$

We have thus obtained the state transition function (**F**) that drives the inputs of the flip-flops to determine the next state (See slide 8.1).

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

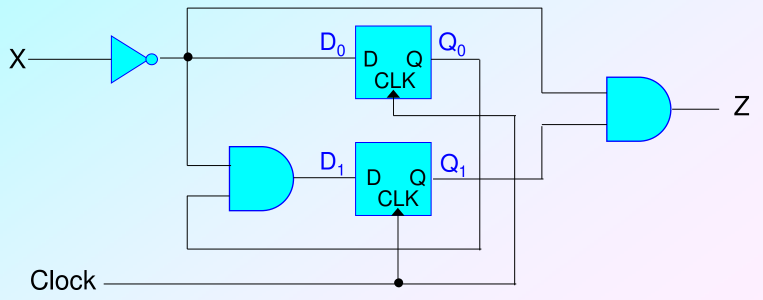
Z X / Q_1Q_0	0	1
00	0	0
01	0	0
11	1	0
10	ø	ø

When designing functions F and G, design methods for combinational circuits (prime implicants, prime implicant chart, minimization) that were 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.

$Z = X'Q_1$ $Z = \mathbf{G}(\text{Input "X" , State "Q}_i\text{"})$

6. We implement and draw the designed circuit using logic gates.



Example: Same circuit designed using J-K flip-flops

The first three steps are the same.

4. In this example, we will use positive edge-triggered J-K flip-flops.

J-K 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 J-K flip-flops instead of D flip-flops generally yields simpler logic functions for the next state.

However, since the functions in this example are already simple, the J-K flip-flop yields no further simplification.

We had determined the transitions of state variables from the state transition table in step 3.

$Q_1^+Q_0^+,Z$ / Q_1Q_0 \ X	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 ($Q_1 \rightarrow Q_1^+$):

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

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

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

We derive the required input values for the flip-flops using the transition tables.

Q_1 transitions:

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

Q_0 transitions:

$Q_0Q_0^+$ / Q_1Q_0 \ X	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

J_1 / Q_1Q_0 \ X	0	1
00	0	0
01	1	0
11	ø	ø
10	ø	ø

K_1 / Q_1Q_0 \ X	0	1
00	ø	ø
01	ø	ø
11	0	1
10	ø	ø

J_0 / Q_1Q_0 \ X	0	1
00	1	0
01	ø	ø
11	ø	ø
10	ø	ø

K_0 / Q_1Q_0 \ X	0	1
00	ø	ø
01	0	1
11	0	1
10	ø	ø

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

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

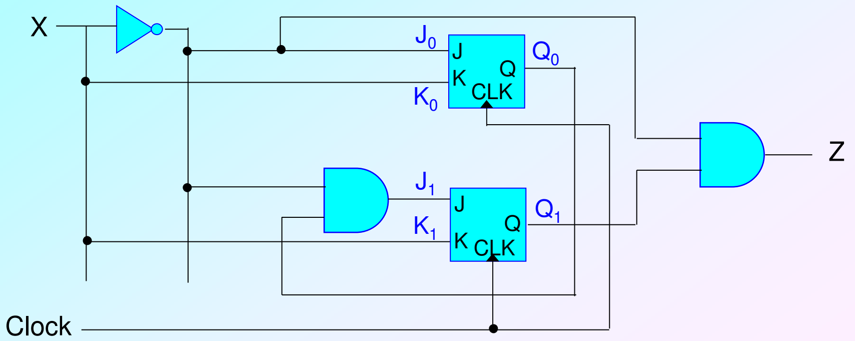
We have thus obtained the function (F) that drives the inputs of the flip-flops and determines the next state.

5. We determine the output function (G) using the output table.

Z X Q_1Q_0	0	1
00	0	0
01	0	0
11	1	0
10	ø	ø

$Z = X'Q_1$

6. We implement and draw the designed circuit using logic gates.



Transition tables for flip-flops:

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

Transition table for S-R 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 J-K 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

Designing a circuit using the Moore model has the same design stages that we have already seen.

It is important to note that

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

Problem:

We will design a synchronous sequential circuit with two inputs (X,Y) and a single output (Z).

If the number of 1s received at the input is a multiple of 4, the output of the circuit is 1. Otherwise, the output should be 0. If no 1s are received (the number of 1s is zero), the output should be 1.

Solution:

The circuit should perform the *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:

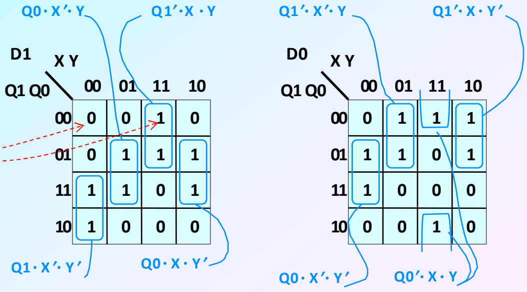
Meaning	S \ XY (S⁺)	00	01	11	10	Z
Modulo 0	S0	S0	S1	S2	S1	1
Modulo 1	S1	S1	S2	S3	S2	0
Modulo 2	S2	S2	S3	S0	S3	0
Modulo 3	S3	S3	S0	S1	S0	0

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: Q1 Q0 \ X Y	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 Q0 \ X Y	00	01	11	10
00	0	1	1	1
01	1	1	0	1
11	1	0	0	0
10	0	0	1	0

Since we are using the characteristic eq. of the D flip-flop ($Q^+=D$), $D_1=Q_1^+$, $D_0=Q_0^+$

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,

- The input of each D flip-flop is driven by a separate multiplexer.
- The state variables (flip-flop outputs) are connected to the selector (control) inputs of the multiplexers. Therefore, each multiplexer selects one of its data inputs according to the current state.
- The inputs of the multiplexer should have the necessary values that produce the next state of the machine.
- We obtain the values that will be applied to the inputs of the multiplexers from the rows of the state transition table.

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

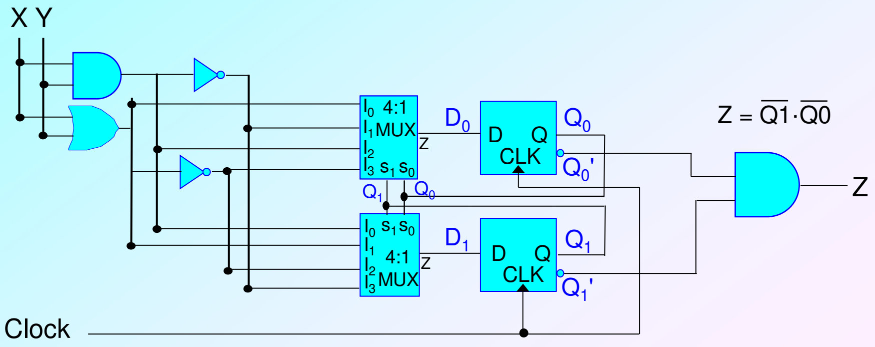
D input values of the flip-flop: (From the previous example)

D1

Q1 Q0 \ X Y	00	01	11	10	to multiplexer:
00	0	0	1	0	$I_0 = X \cdot Y$
01	0	1	1	1	$I_1 = X + Y$
11	1	1	0	1	$I_3 = \overline{(X \cdot Y)}$
10	1	0	0	0	$I_2 = \overline{(X + Y)}$

D0

Q1 Q0 \ X Y	00	01	11	10	to multiplexer :
00	0	1	1	1	$I_0 = X + Y$
01	1	1	0	1	$I_1 = \overline{(X \cdot Y)}$
11	1	0	0	0	$I_3 = \overline{(X + Y)}$
10	0	0	1	0	$I_2 = X \cdot Y$



Counter Design

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

The Moore model is appropriate for counter design.

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

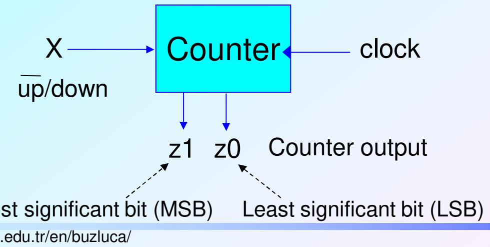
We can obtain outputs directly from the state variables (Output = State, O=S). This means that the design must use the Moore Model.

Example:

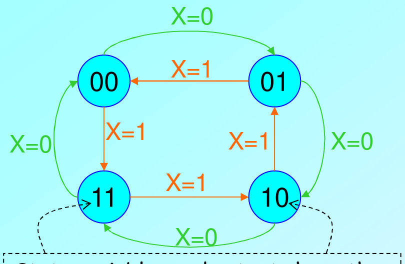
Design the counter shown below that has a single control input (X).

The counter should count at each rising edge of the clock signal in the natural order, 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.


X
$\overline{\text{up}}$/down
Counter
clock
z1
z0
Counter output
Most significant bit (MSB)
Least significant bit (LSB)

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 ordered according to the Gray code so that state table can also be used as a Karnaugh map.

Designing the counter using D flip-flops:

Recall:
$Q^+ = D$

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

Q_1Q_0 \ X (D_1)	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

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

D0 = Q0'

Output:

Z0 = Q0
Z1 = Q1

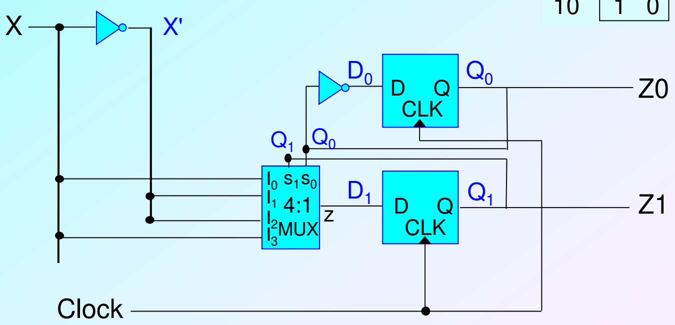
We can use logic gates (AND, OR, XOR etc.) or multiplexers to implement counters.

Below, since the D0 input has a very simple expression (D0 = Q0') , we prefer a logic gate (single inverter) to a multiplexer.

A multiplexer is used to drive input D1.

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

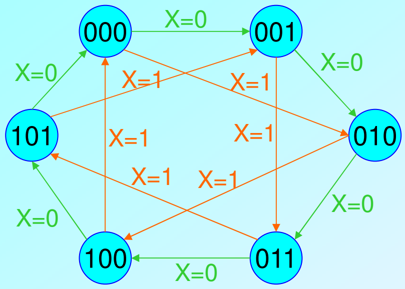
D_1 — Q_1Q_0 \ X	0	1	to multiplexer:
00	0	1	I_0 = X
01	1	0	I_1 = X'
11	0	1	I_3 = X
10	1	0	I_2 = X'



Example:

Design a counter that counts in the sequence 0-1-2-3-4-5 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

We move Q_0 to the columns.

We organize the state table as a Karnaugh map:

$Q_2^+Q_1^+Q_0^+$ — 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, we will use T flip-flops.

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 examining transitions ($Q_2 \rightarrow Q_2^+$, $Q_1 \rightarrow Q_1^+$, $Q_0 \rightarrow Q_0^+$), we determine T_2 , T_1 , and 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

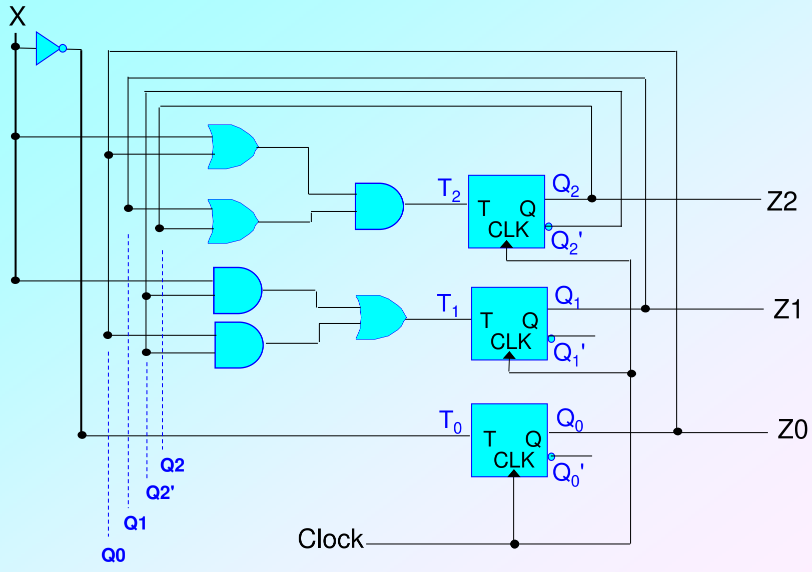
$T_2' = Q_0' \cdot X' + Q_2' \cdot Q_1'$

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

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

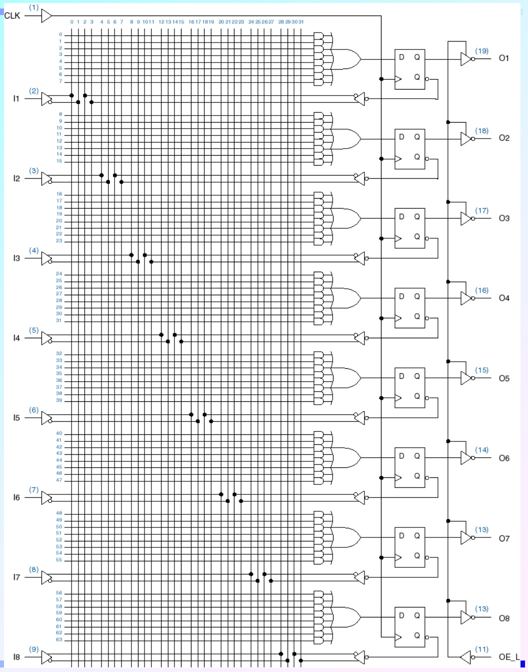
$T_0 = X'$

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



Implementation of synchronous circuits using PLD

- Earlier, we looked at implementing combinational circuits using programmable logic devices (PLDs).
- It is also possible to use PLDs to implement synchronous circuits.
- For this purpose, we use PLD units that include flip-flops.
- At the right, a **16R8** PAL circuit is shown.
- Currently, synchronous circuits are commonly implemented using CPLDs and FPGAs.


CLK
(1)
I1
(2)
I2
(3)
I3
(4)
I4
(5)
I5
(6)
I6
(7)
I7
(8)
I8
(9)
D
Q
Q
(19)
O1
(18)
O2
(17)
O3
(16)
O4
(15)
O5
(14)
O6
(13)
O7
(13)
O8
(11)
OE_L