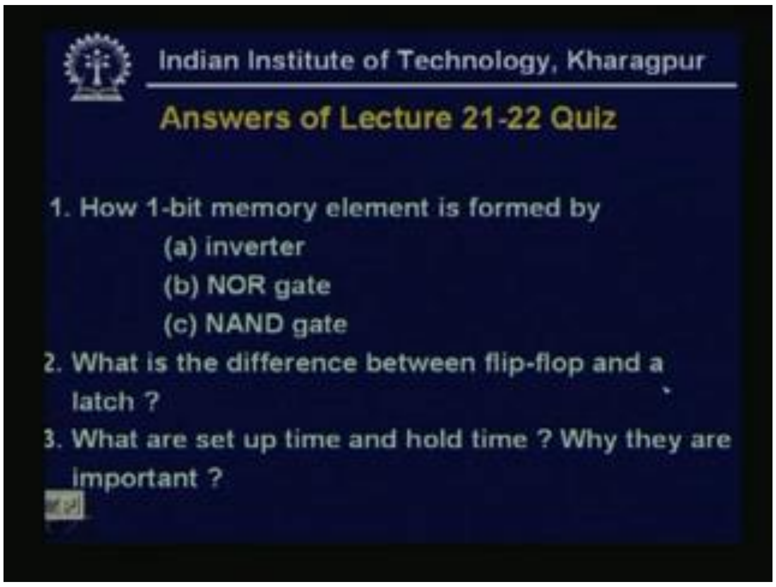
Digital Systems Design

Prof. D. Roychoudhury
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 23
Design of Registers and Counters

Last day we have learnt, how to design a different type of flip flops? Say the D flip flops, R S flip flop, the J K flip flop. And what is the problem? Or what is the, what are the advantages and disadvantages of different type of flip flops? Now today, we will see how the more complex equations of gates can be designed using the different type of flip flops. Now, this more complex sequential circuits, they contain mainly the registers and counters, shifters mainly the storage elements. So, first we see the design of registers and counters in this lecture.

(Refer Slide Time 01:41)



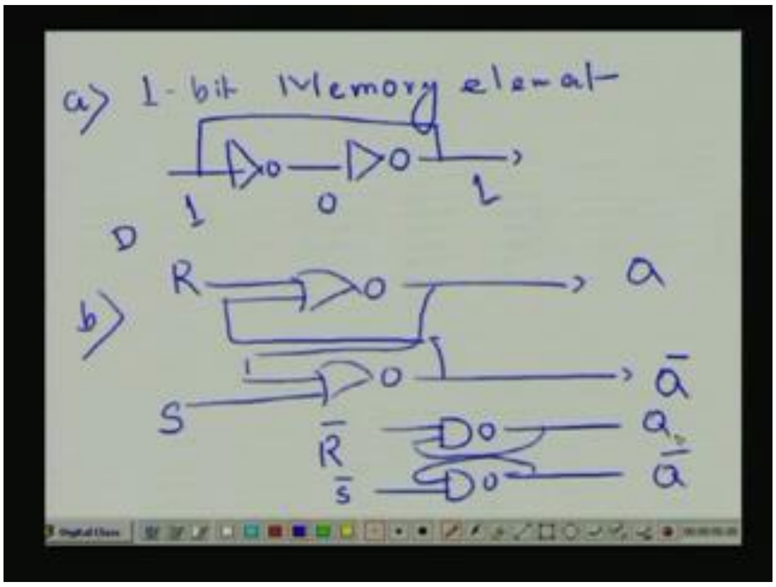
Indian Institute of Technology, Kharagpur

Answers of Lecture 21-22 Quiz

1. How 1-bit memory element is formed by
   (a) inverter
   (b) NOR gate
   (c) NAND gate
2. What is the difference between flip-flop and a latch ?
3. What are set up time and hold time ? Why they are important ?

Before that, we quickly see the answers of the quiz questions based on the lecture 21 and 22. Now, the first question was the, how 1 bit memory element is formed by inverter, NOR gate and NAND gate.

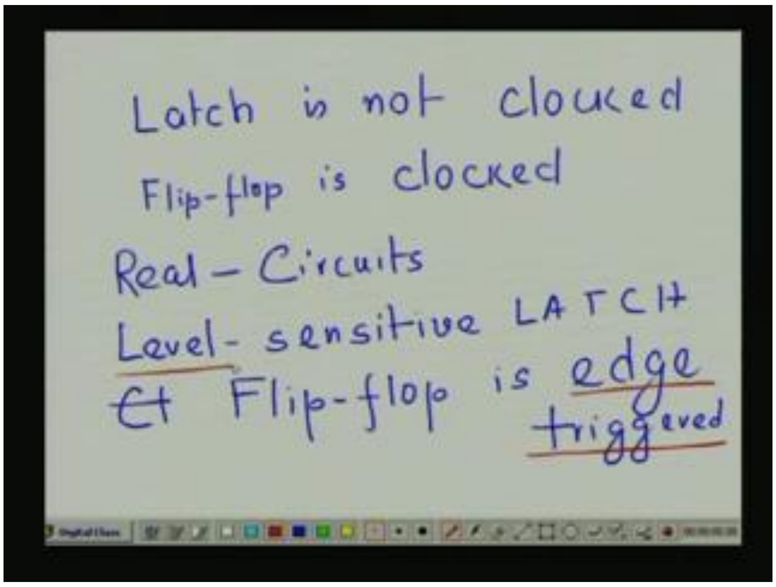
(Refer Slide Time 02:09)



We have read in the last lecture that, the 1 bit memory element, that is by using inverter 1 NAND gate and if it is cascaded with another inverter; that means, 1 NOT gate and another NOT gate. So, if 1 is given as the input, then this 0 is there and then again by using this 1 that means, if this is my data, again this becomes my data and this is feedback as the. So, this is the sample, by using the NOT gate, NAND using the NOR gate we have seen, that two cascaded NOR.

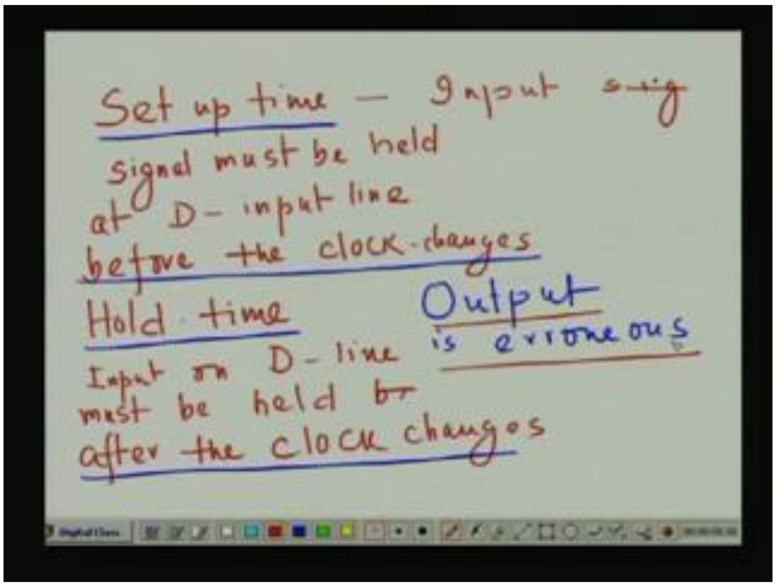
So, actually this input is coming from here, so this is the RS flip flop and we have seen the functional tables of these two, that Q and Q bar. Similarly, that using J K using the NAND gate also we have seen that, only the inputs can be inverted. That means R bar and S bar, then we replace the, if we replace the 2 NOR by NAND two input NAND, then it will be same. So, these are the different type of circuits we have seen now what is the difference between flip flop and the latch, the second question.

(Refer Slide Time: 05:06)


Latch is not clocked
Flip-flop is clocked
Real – Circuits
Level- sensitive LATCH
Et Flip-flop is edge triggered

So, again normally we have defined, the latch is not a clocked circuit, latch is not clocked, whereas the flip flop is clocked. But, in real life for real circuits, sometimes that level sensitive latch is called a latch and flip flop is an edge triggered if you consider that flip flop is edge triggered. So, mainly these are the basic difference between a latch and flip flop. Now, the third question was what are setup time and hold time, why they are important. Now, as we have already read, the setup time is the time during which the data on input D must be kept constant, so before the clock changes and the hold time.

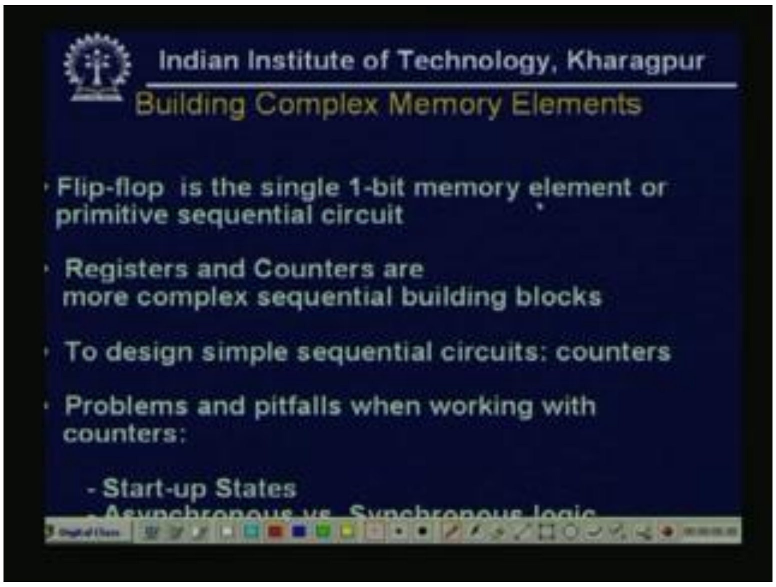
(Refer Slide Time: 07:55)


Set up time – Input sig
signal must be held
at D- input line
before the clock changes
Hold time
Output
Input on D- line
is erroneous
must be held br
after the clock changes

So, first if we think the setup time, the input signal must be held at D input line before the clock changes, so this is my setup time. So, mainly here, before the clock changes and if it is hold time, then already we have seen that it is the same thing. That input on D line must be held before the, this is before the, if it is before the clock changes, then it is setup time, if it is after the clock changes, then it is hold time after the clock changes. So this is after the clock changes.

So, why it is important because, if within this time; that means, within the setup time or within this hold time, the input changes, then our output will be erroneous. So, main thing is that, output is naught or output is erroneous, so mainly this is the problem. So, now, we start the design of complex memory elements.

(Refer Slide Time: 10:53)



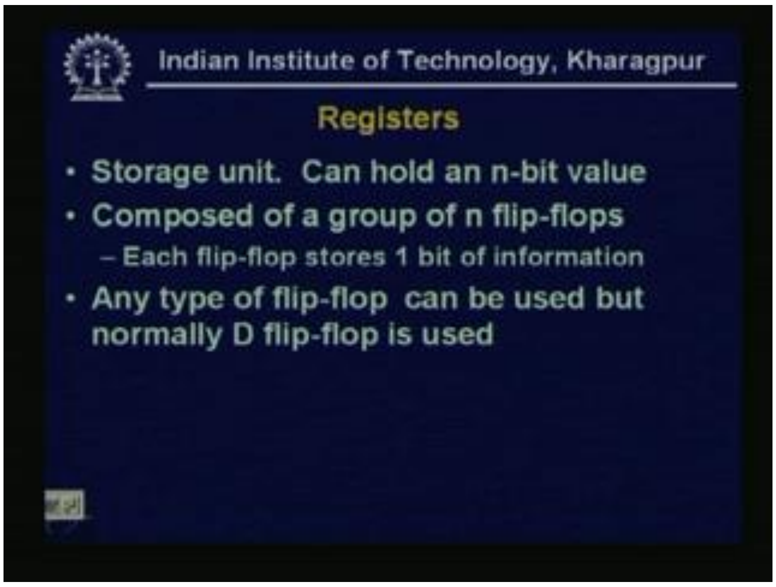
Indian Institute of Technology, Kharagpur

Building Complex Memory Elements

- Flip-flop is the single 1-bit memory element or primitive sequential circuit
- Registers and Counters are more complex sequential building blocks
- To design simple sequential circuits: counters
- Problems and pitfalls when working with counters:
  - Start-up States
  - Asynchronous vs. Synchronous logic

Now, as already we have seen the flip flop is the single 1 bit memory element or primitive sequential circuits and the different types of flip flops, we have seen. Now, the registers and counters are more complex sequential building blocks, now to design simple sequential circuits, first we see the counters. And mainly, what we will discuss in this lecture that problems and pitfalls, when working with counters and mainly the startup time or the asynchronous versus synchronous logic, now see the registers.

(Refer Slide Time: 11:36)



Indian Institute of Technology, Kharagpur

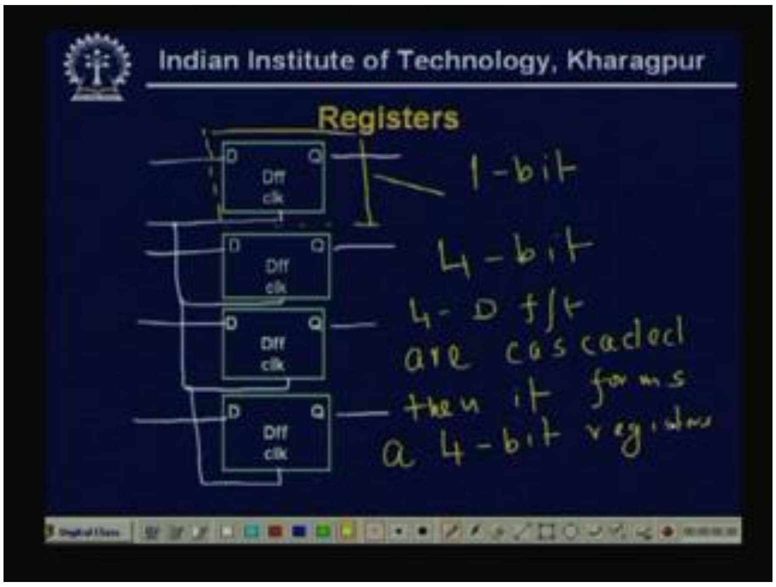
Registers

- Storage unit. Can hold an n-bit value
- Composed of a group of n flip-flops
  - Each flip-flop stores 1 bit of information
- Any type of flip-flop can be used but normally D flip-flop is used

So, this is a storage unit and can hold an n bit value, so first we will see that how or 1 bit memory element means, 1 bit flip flop. So, how that n bit storage element can be constructed from this 1 bit memory element. So, this is nothing, but it can be composed of a group of n flip flops and each flip-flops stores 1 bit of information. So, this together it will store an n bit information.

Now, any type of flip flop can be used, but normally D flip flop is used. So, first we will study the complex circuits with different type of flip flops. But, which one is better for implementation of real life circuits, that we will see later, now first is registers.

(Refer Slide Time: 12:37)

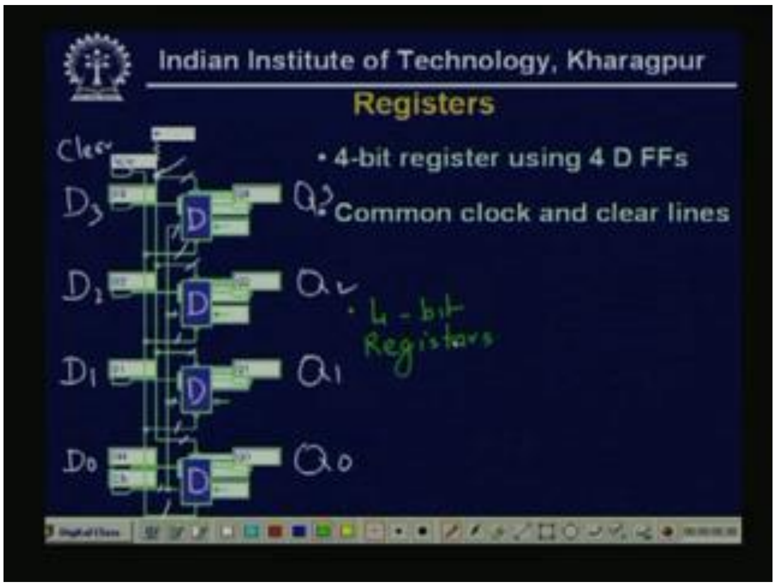

Indian Institute of Technology, Kharagpur

Registers

4- D f/f are cascaded then it forms a 4-bit register

So, see if we see this is a D flip flop we have seen, so already we have read the D flip flop, this is the input line, this is Q the output line and there is another input that is the clock. So, this is 1 bit, this is the 1 bit memory element, so if we consider this is 1 bit memory element. Now, similarly if we put another three that means, this is another D input Q output clock, see we are taking the same clock and, so now, see together. That means, if it is a, if this is a 4 bit, then it must store the 4 bit value or 4 bit information can be stored. Then, we are calling that this is a 4 bit register. So, if 4 D flip flops are cascaded, then it forms a 4 bit registers.

(Refer Slide Time: 15:04)


Indian Institute of Technology, Kharagpur

Registers

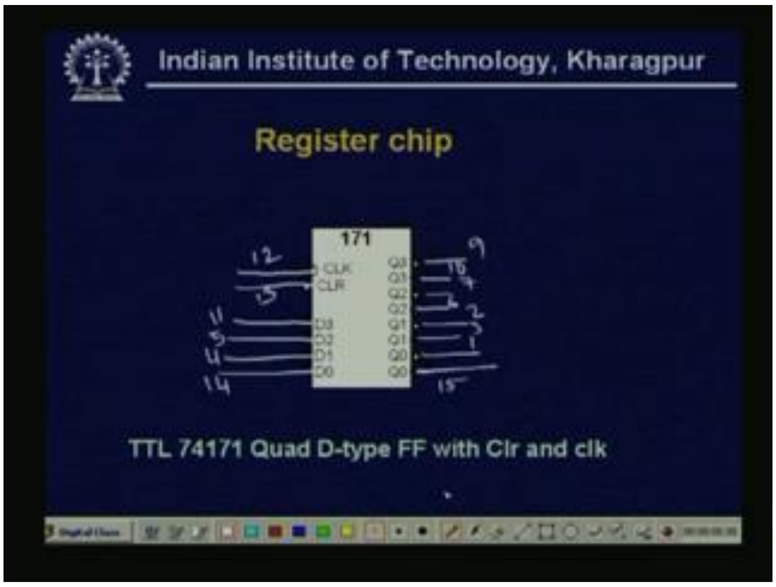
- 4-bit register using 4 D FFs
- Common clock and clear lines
- 4-bit Registers

So, now we see that just now what we have seen, a common clock, now we add one clear line also; that means, as it is a memory it is a. So, 1 4 bit storage again, if we want to reuse this thing, then we will clear this and a new value can be stored in this register. But this is only a 4 bit value can be stored, so just now, what we have seen, that say this is a, this is my one D flip flop, this is one D flip flop, this is one D flip flop and this is, so 4 D flip flop.

Now, clock is, this is the clock line, so one common clock line is there and here it has a fed to 4 D flip-flops. Another clear line, see this is the clear line, so again it is a common clear line and it is fed to the 4 D flip-flops. Now for each D flip flop there is, see if this is D 0, D 1, D 2, D 3, then this is my D input. So, one D 0 similarly this is D 1, D 2 and D 3, four inputs are there.

This is the clock line, clock line is here, so the two inputs 1 clock and 1 data and this is the clear lines, this is the power line. And four outputs are, as this is Q 0, Q 1, Q 2 and Q 3, so it forms a 4 bit registers. Now, this type of chips are available, so this is one example, that the chip what is available.

(Refer Slide Time: 17:52)

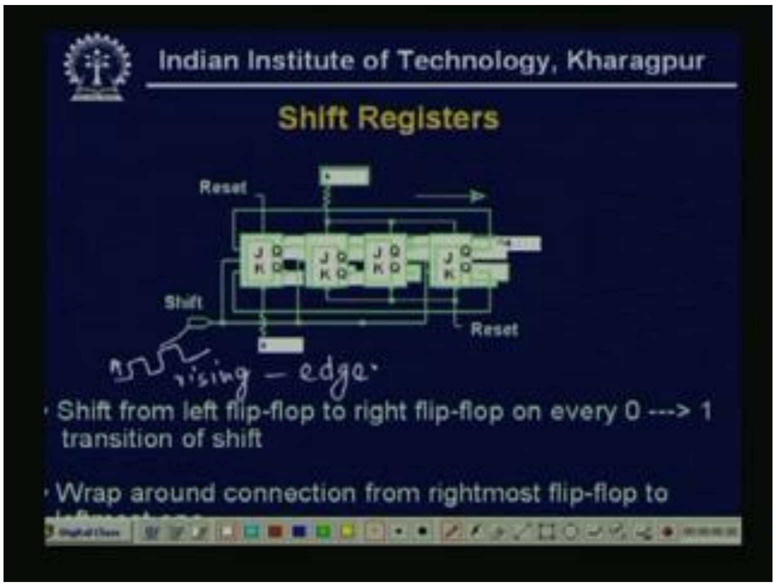

Indian Institute of Technology, Kharagpur
Register chip
171
12
CLK
CLR
13
11
5
4
14
D3
D2
D1
D0
Q3
Q3
Q2
Q2
Q1
Q1
Q0
Q0
9
10
7
6
2
3
1
15
TTL 74171 Quad D-type FF with Clr and clk

See this is a 16-pin chip, this is a TTL 74171 quad D type flip flop that means, here that 4 D type flip-flops are available. And there is two common lines the, here this is a clock and this is clear, normally they are pin 12 and pin 13. And the four inputs are D 0, D 1, D

2 and D 3 and the lines pin numbers are 11, 5, 4 and 14. And the outputs that these are Q 0 is pin number is 15 and this is Q 0 bar, so this is 1.

Similarly, this is Q 1 3, Q 2 bar 2, Q 2 6, Q 2 bar 7, Q 3, 10 and Q 3 bar 9, so this is one real life of 4 bit D flip flops, that quad D type flip flops. One thing is that here, this chip can be used for 1-bit can be used for 2-bit, 3-bit or 4 bit. Then, accordingly we will select the Q 0, Q 1, Q 2 or Q 3 or together, now how we construct a shift register.

(Refer Slide Time 19:56)



Indian Institute of Technology, Kharagpur

Shift Registers

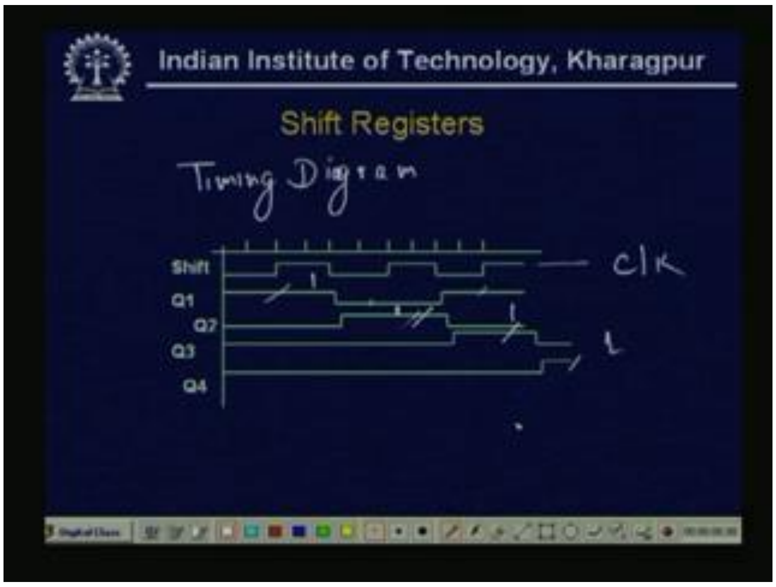
- Shift from left flip-flop to right flip-flop on every 0 ---> 1 transition of shift
- Wrap around connection from rightmost flip-flop to

See, now we consider a shift register with J K flip flops. So, there are 4 flip flops we have taken, this is the first one, that say Q 0, so we take or this one is or the first one 1, 2, 3, 4. And, here the power lines, the common power lines are there this is a reset line, because it is a J K, so reset lines. Now, this is the shift line, shift and see this is a, shift from left flip flop to right flip flop on every 0 to 1 that means, low to high transition of the shift clock.

So, when this shift clock say when here this shift line there will be a clock and when it will be in the rising edge; that means, 0 to 1 this is the edge triggered. So, this is a, as this is a flip flop. So, this will be an edge triggered and this is a rising edge triggered, so that time it will be shifted, every flip flop will shift 1 bit to its right. That means, it is a left to right shifter.

And another is the wrap around connection from rightmost flip-flops to leftmost one, see here this is a, this is the wrap around fashion. So see, this is the last output or the, if you call the 4 flip flops, then the last output of the fourth one is taken as the input of the first one. So, this is a wrap around fashion or sometimes we call the periodic boundary. So, the connection has been taken like that, now if we consider the timing diagram.

(Refer Slide Time 22:48)



Indian Institute of Technology, Kharagpur

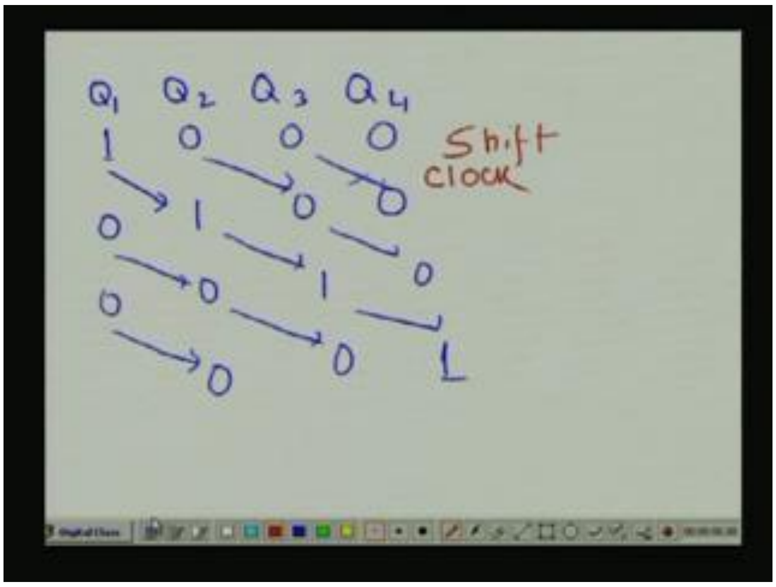
Shift Registers

Timing Diagram

Then, the timing diagram of, see this is the shift clock, so this is my clock, then the four outputs Q 1, Q 2, Q 3. See that Q 1… this is at the high to low because, Q 1 becomes 1 and this becomes 0 when it is a. That means, again when it is a high to low it will be getting 1. Now, again Q 2, Q 2 will be taking, Q 2 is that just the reverse of the Q 1.

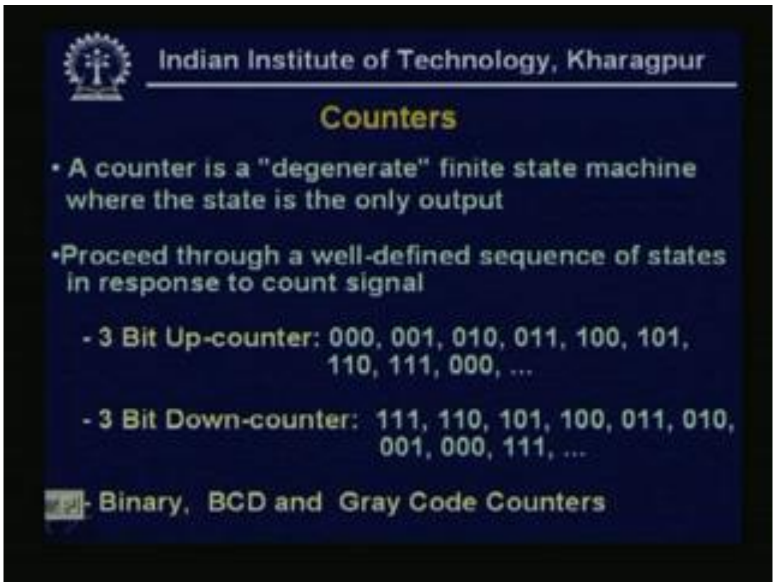
If we take Q 1 as the clock that means, the timing of Q 1 and Q 2 are the reverse. Then Q 3 again Q 3 will be, that first Q 1 as if Q 1… if this is my 1 value then Q 1 is shifted here and then Q 2 is shifted, this Q shifted here and Q 4 when, this is it is shifted here. So, if we see the shifting, this will be like that.

(Refer Slide Time 25:14)



See the Q 1 or Q 1, Q 2, Q 3, Q 4, see initially if it is say 1, 1 was here 0, then 1 shift, it will be here, this is the 0. Then for another shift, this 0 will come here, 1 will be here, 0 will be here, again this will be 0, for another shift it will be here, this will be here and this 1 will be here. So, in this way; that means, and this shift means this is a shift clock, so this will be a every time, this will be a shift clock, so this will be the timing diagram.

(Refer Slide Time: 26:46)



Indian Institute of Technology, Kharagpur

Counters

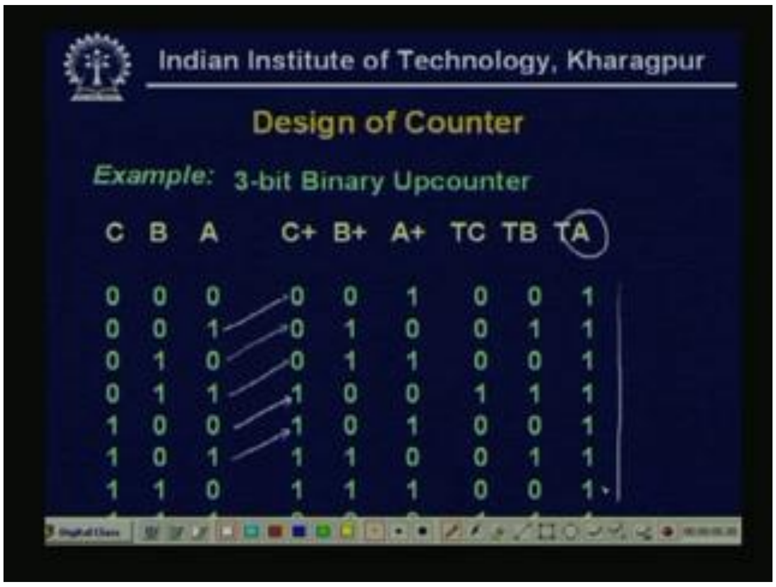
- A counter is a "degenerate" finite state machine where the state is the only output
- Proceed through a well-defined sequence of states in response to count signal
  - 3 Bit Up-counter: 000, 001, 010, 011, 100, 101, 110, 111, 000, ...
  - 3 Bit Down-counter: 111, 110, 101, 100, 011, 010, 001, 000, 111, ...
  - Binary, BCD and Gray Code Counters

Now, if you see the counters, now a counter we can define in this way the counter is a degenerate finite state machine, where the state is the only output. So, there is no other primary output from this machine, so the counter is defined like that. Now, it proceeds through a well defined sequence of states, in responses to count signal. So, it is defined that as if it counts and a predefined sequences or we can tell the well defined sequences are there and the, with the count signal we it will give that sequences.

So, a fixed number of counts the or a fixed number of sequences it will generate and that will be the direct measure of the count and that is why the name is the counter. So, normally we will see that 3 bit up bit counter or down counter, it can be n bit, n bit counter. And what do you mean by up bit means, it is a increasing sequences, so if we consider 0 1 2 binary 0 0 0 0 0 0 1 like that, then it will be a 3-bit counter.

Similarly, down counter means, it starts from 7 and then 7, 6, 5, 4, 3, 2, 1, 0 and again it will repeat the initial states. So, it will be a again, the 3-bit down counter or other counters, the binary counters, the B C D, gray code counters. So, these are the different type of counters, are normally constructed by using the flip-flops, different type of flip flops, so if we see the one design of 1 3 bit up counter.

(Refer Slide Time: 28:38)



Indian Institute of Technology, Kharagpur

Design of Counter

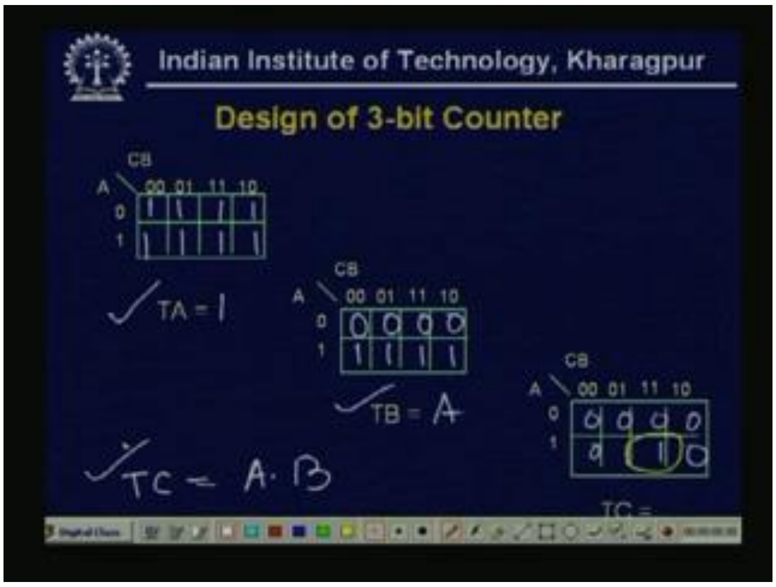
Example: 3-bit Binary Upcounter

C	B	A	C+	B+	A+	TC	TB	TA
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1

So, 3-bit up counter, so say one machine we have already, one machine with the flip-flops, then these are my present, this C B A as if these are my present state, these are my present states. And this C plus B plus A plus, these are my next state, these are my next state. And we will be seeing that this T C T B T A as if, if we consider the 3 flip-flops, then these are the outputs of the flip-flops.

Now, as it is a 3-bit up counter, so this 0 0 0 2 1 1 1 that we are considering; that means, 0 1 0 1 2, 3 up to 7, now when it is 0 0 0 the, I want that next state should be 0 0 1. See, here I want this should be my next state, again the next state should be the, this one, in this way it will work. So, we have to construct a machine, so that the if the present state is C B A. Then the next state will be C plus B plus A plus. That means, the increment or 1 in that way. So, what we will be doing this is nothing, but a truth table and as there are 3 inputs, so we will, we can form 3 Karnaugh maps.

(Refer Slide Time: 30:32)



Indian Institute of Technology, Kharagpur

Design of 3-bit Counter

A \ CB	00	01	11	10
0	1	1	1	1
1	1	1	1	1

✓ $TA = 1$

A \ CB	00	01	11	10
0	0	0	0	0
1	1	1	1	1

✓ $TB = A$

A \ CB	00	01	11	10
0	0	0	0	0
1	0	1	1	0

✓ $TC = A \cdot B$

$TC =$

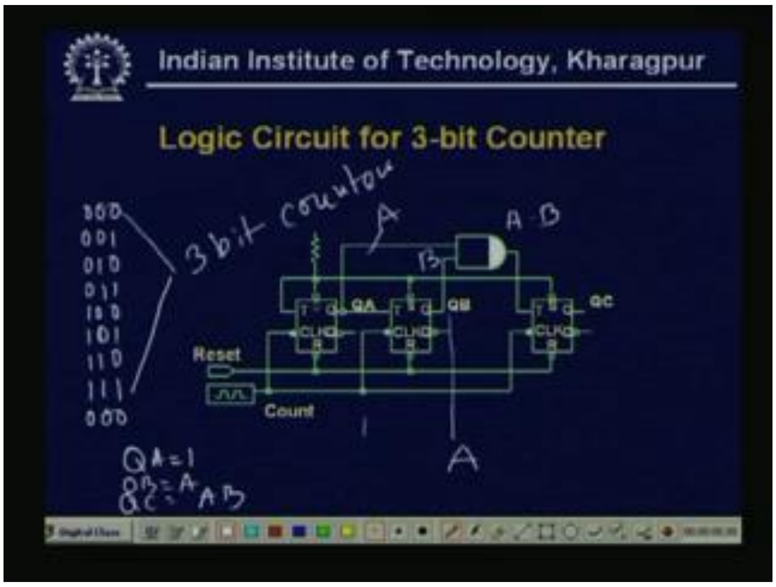
Now, see if we consider the T A that means, see if we consider the, these output the Karnaugh map for this, Karnaugh map for these output T A, then see every time whatever be the inputs whatever be the present state, that this value outputs are T A value is 1. So, what will be the thing, if we put the value, then this Karnaugh map this value will be all 1.

Now, if we put the, for the see for the T B, see whenever A is 0, T B value is 0, see A is 0 1 0 1 0 1 0 1 and see here T B value is 0 1 0 1 0 1 0 1. That means, whenever A is 0, T B is 0. And here, C B here that will be the all possible C B values we are taking. Now if we consider T C, then actually the T C value is, when this is 0 1 or if you consider that 0 0 0 or it will be 2 ones. When A is 1, both the cases A is 1, these are the cases that A is 1 and then C is 1. Here another A is 1, T C is 1 and that time C B value is 0 1 and here C B value is 1 1.

So, now if we consider, then already we have seen, the T A that, this will be all 1; that means, whatever be the presents states these are all ones. Now, here we have seen whenever A is 1, C B is 1, but whenever A is 0, C B value is 0. Here we have, we have seen that when A is 1, but C B is 0 1 and 1 1, that time only it is other 1 otherwise it is 0.

So, what we get that T A is 1, T B is A and T C is A dot B because, these are my, these are the 2 values, so 0 1 1 1 means AA and this is B, so this is T C is A dot B. So, what from the Karnaugh map, what we get that T A equal to 1, we get T A equal 1, T B is equal to A and T C is A B, so now if we see the design.

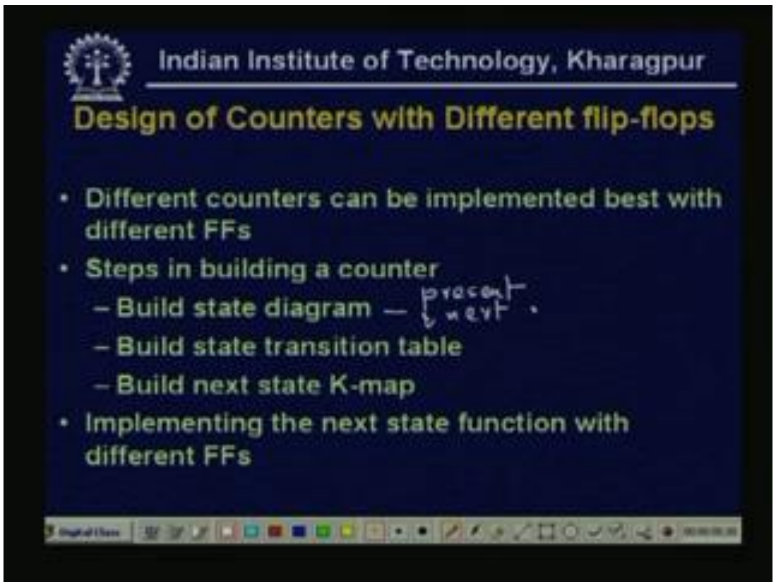
(Refer Slide Time: 34:36)



If we see the design, then see, that first we have taken 3 D flip-flops, here we have constructed the logic circuit by D flip-flop. Then see, as usual the reset lines and the clock lines are there, then Q A value, here the Q A value is always 1. So, this is my, this is my T A value or the first one. See the Q A is A means the previous input and Q C, see this is my, this is my A and this is my B, so this is my A dot B and that is my Q C.

So, that Q A is 1, Q B is A and Q C is A B, so this will give a, if we now if we just allow to run this circuit or apply the clock, then at each clock pulses, it will generate the 0 0 0. Then next clock pulse, it will give 0 0 1, it will give 0 1 0, 0 1 1, 1 0 0, 1 0 1, 1 1 0, 1 1 1 and again 0 0 0. So that means, it will count only the, this 3 bit, this is a 3-bit counter, this is a 3 bit counter. That means, all 0 to 1 1 1; that means this 8-bit it will measure. So that is, why it is a 3-bit counter using D flip flops.

(Refer Slide Time 36:48)


Indian Institute of Technology, Kharagpur

Design of Counters with Different flip-flops

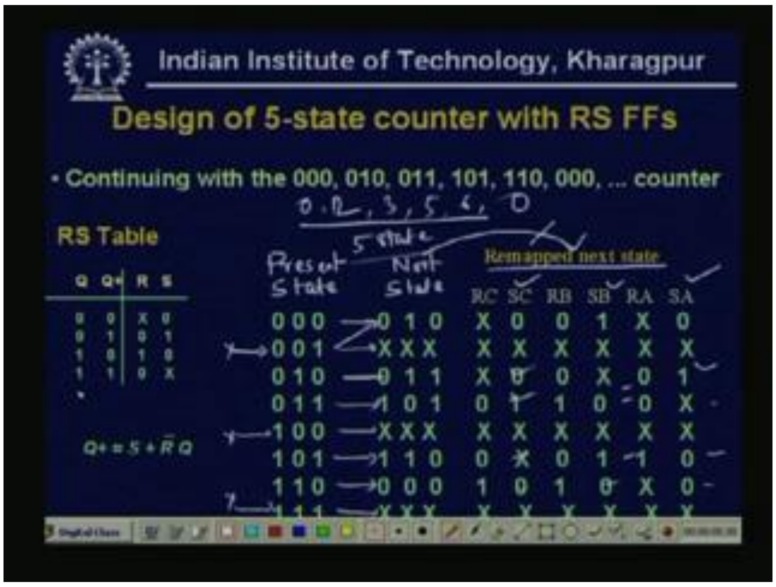
- Different counters can be implemented best with different FFs
- Steps in building a counter
  - Build state diagram — present, next.
  - Build state transition table
  - Build next state K-map
- Implementing the next state function with different FFs

Now, if we consider the design of counters with different type of flip flops, so different counters can be implemented, best with different flip-flops. Now, it is not the case, that always T flip flop is chosen or the D flip flop is chosen. So, depending on the behavior of the counters, we have to choose the flip flop. Now, the steps of designing a counter, is that first we will build state diagram.

That means what will be the present state and what will be the build state diagram, depending on the present state and the next state present and next state. Then we will build the state transition table, then build next state k map because, if a state transition table is given; that means, this is my truth table, that if this is a present state this will be my next state. And, next state is the only output we have defined the counter in that way, so that will be my output.

So, input and output is given means that is similar to my truth table, so this state transition table is nothing, but my truth table. Then, I will map this truth table by an unknown Karnaugh method and then implement the next state function with different flip flops. So, again we see a design of five state counter, with R S flip-flop.

(Refer Slide Time 38:17)



Indian Institute of Technology, Kharagpur

Design of 5-state counter with RS FFs

- Continuing with the 000, 010, 011, 101, 110, 000, ... counter

0, 2, 3, 5, 6, 0

5 state

RS Table

Q	Q+	R	S
0	0	X	0
0	1	0	1
1	0	1	0
1	1	0	X

$Q+ = S + \bar{R}Q$

		Remapped next state					
Present State	Next State	RC	SC	RB	SB	RA	SA
000	010	X	0	0	1	X	0
001	XXX	X	X	X	X	X	X
010	011	X	0	0	X	0	1
011	101	0	1	1	0	0	X
100	XXX	X	X	X	X	X	X
101	110	0	X	0	1	1	0
110	000	1	0	1	0	X	0
111	XXX	X	X	X	X	X	X

So, here it is a 5 state only and we have taken random 5 states, see in this example, we have taken 0 1 0 2 0 1 0 3 then 5 6 7 5 6 this again 0 because, these are my 5 state. So, this is a 5 state counter, one thing is that, as it is a 5 state, so in any case we need 3 bit. So, this is another thing that how many flip flops we need, so log if it is a, if it is a n state counter then log n flip flops are needed.

So, as it is a 5 state flip flop so; that means, ceiling of log 5; that means, this is a 3 bits are needed for that, so now here, we have seen that or if we draw the present state and next state. So, this is my present state or we can tell that input also and these are my next state, see this is a, when it is 0 0 0, I want to, I want the next state should be 2. If it is 1 see, we have not consider the 1 as the state. So, I do not want any output for this, that is why it is undefined type of thing, any type it can be.

Now, if it is a 2, I want 3, so for 0 1 0 the state will be, for 0 0 0 it will be 2, for 0 1 0 it will be 3, for 3 it will be 5, 4 is again undefined, for 5 it should be 6, for 6 I want the next state should be 0, because this is the 5 state, I do not want all one. So, see here this 0 0 1, 1 0 0 and 1 1 1, means my decimal 1 4 and 7, this states are not defined in this machine. So now, it is a remapped next state because, these are my present state, these are my next state.

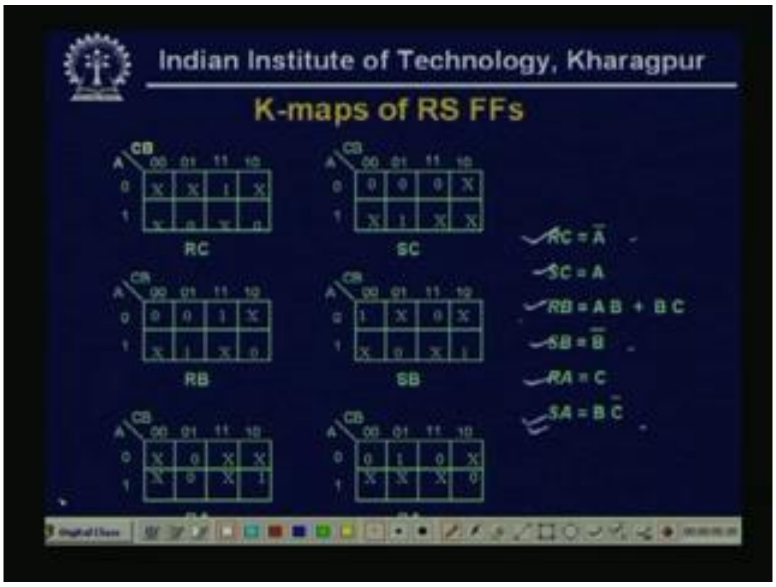
So now, again that 3 flip-flops and we here in this design, we consider the three R S flip-flops, so C B A if we consider, we consider that this is a 0 1 0 so; that means, this is S C

0 1 and 0. Similarly for 0 1 1, I want, I want that 0 1 1, so this should be, this should be anything, this can be 1, 0 1 1. Similarly, 1 0 1, I want 1 0 1 or it can be 1 1 0 for and say for these, these are not defined, this can be 1, this can be 1, this is 0 and so, this is 0 0 0.

So, see what we have done, that we have some first, that given the number of states or the given states, we have considered these five states. What should be the next state? That state transition table we have drawn. From there, depending on the flip-flops we want to use, we have remapped the next state, so this is one very important thing. So, this is as if the original next state and depending on the flip-flop, we used the remapped next state is the, this thing.

That means, what will be the output of the flip-flops, this is, this table governs the outputs of that thing. Now, we know the RS table and that is Q plus is S plus R bar Q or if we consider the function table of the RS table. We know that, this will be the if Q is 0, R is not defined, S is 0, Q is 0, then this is 0 1 1, then R is 1, Q is 1, then it will be 0.

(Refer Slide Time 43:48)


Indian Institute of Technology, Kharagpur

K-maps of RS FFs

A \ CB	00	01	11	10
0	X	X	1	X
1	X	0	X	0

RC

A \ CB	00	01	11	10
0	0	0	0	X
1	X	1	X	X

SC

A \ CB	00	01	11	10
0	0	0	1	X
1	X	1	X	0

RB

A \ CB	00	01	11	10
0	1	X	0	X
1	X	0	X	1

SB

A \ CB	00	01	11	10
0	X	0	X	X
1	X	0	X	1

A \ CB	00	01	11	10
0	0	1	0	X
1	X	X	X	0

$RC = \bar{A}$

$SC = A$

$RB = A B + B C$

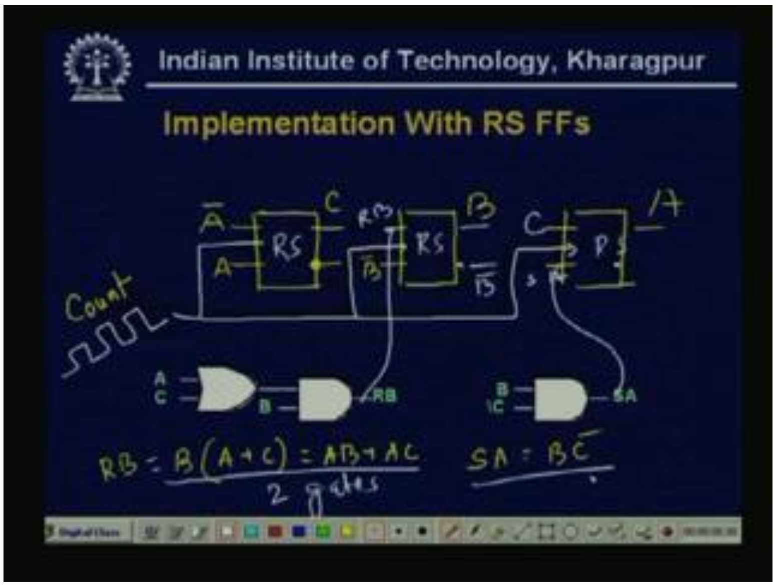
$SB = \bar{B}$

$RA = C$

$SA = B \bar{C}$

So, depending on that, now we will get the, as for the three flip flops and now there are two outputs R and S. So, for the C flip-flops, what will be the R C S C, for the C B flip flop, what will be the R B S B and for A, what will be the R A S A. So, the based on the, the remapped next state, that Karnaugh maps have been filled up. Now, if that, we do that Karnaugh map minimization, we will be getting that this type of logics. We will be getting the RC equal to A bar, SC is A, because this must be the complimented, than R B is A B plus B C, S B is B bar, R A is C and S A is B C bar.

(Refer Slide Time 45:10)


Indian Institute of Technology, Kharagpur
Implementation With RS FFs
Count
RS
RS
RB
SA
RB = B(A+C) = AB+AC
2 gates
SA = BC̄

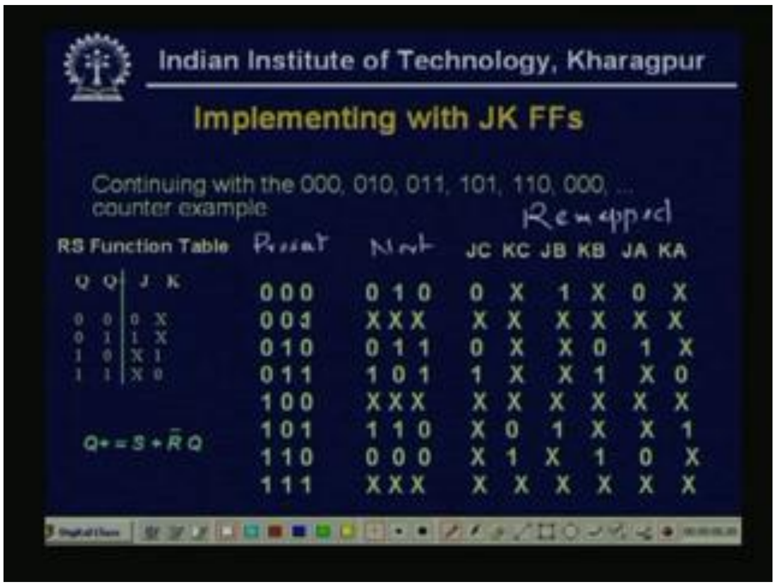
So, now if we start drawing, so first we need three R S flip flops, so this is one, actually this is my C this is Q bar, so this should be A bar, this should be my A, this is the B, B bar, C and this is A and C. The clock line is these are the, so this is the clock or we can tell, this is my, this is my count. Because, at each clock it will generate one state; that means, it will 1 one count will be going on. So, this is my, this output is my R B, this is my B bar and this is my A B C.

Now, what we have seen that, this is the B and this is A and Q Q bar. So, first thing is, that R B is, what we have seen that R B is A plus C. And that means, B into A plus C; that means, A B plus A C and this is my R B. Now, S A is, S A is B C bar. So, this will be my, this will be my S A, so this is say as if, this is my SA, this is my R B, so as if, this is my RB. So, this type of and this is my C, this is my R B, this is my B and this is my B bar, so we will be getting this circuit.

So, now if this is that these are all R S flip flop that means, the 2 inputs are there R S and 2 outputs Q, Q bar and then we will be extra logic that we need that are that R this B into A B plus S C that means, this 2 gates for R B. These are here, there are two gates and here there is for S A there is there is 1 AND gate that is needed. So, see here that we need three extra gates three logic gates in addition with the three flip flops, three RS flip flops we need three gates and that number of literals.

So, if we consider see here the literals then actually see that A bar and then B bar C bar that means, for R C S C R B S B R A and S A and for just to get R B and to S A we have need three gates. Now, the same logic if we if we want to implement or the same counter with J K flip flop.

(Refer Slide Time 51:42)



Indian Institute of Technology, Kharagpur

Implementing with JK FFs

Continuing with the 000, 010, 011, 101, 110, 000, ... counter example

RS Function Table

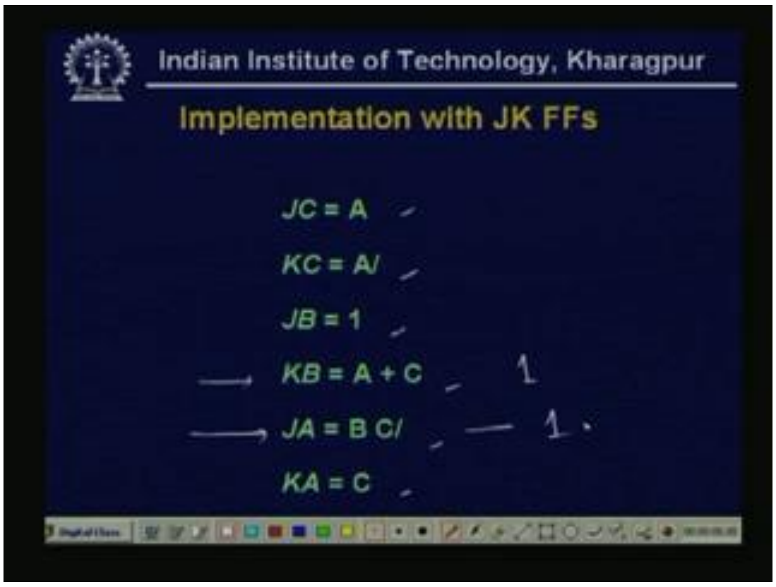
Q	Q+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

$Q+ = S + \bar{R} Q$

Present	Next	JC	KC	JB	KB	JA	KA
		Remapped					
000	010	0	X	1	X	0	X
001	XXX	X	X	X	X	X	X
010	011	0	X	X	0	1	X
011	101	1	X	X	1	X	0
100	XXX	X	X	X	X	X	X
101	110	X	0	1	X	X	1
110	000	X	1	X	1	0	X
111	XXX	X	X	X	X	X	X

The same counter then the circuit will be say again we have taken that 0 0 2 3 5 6 7 then again 0 that means, again five state counter, so we take the 5 state counter. Then see the again this we can put this we can put as the present state and similarly this is my next state and this is my remapped next state. So, similar way because these are the same as that of R S only this remapped next state that will be different for J K flip flops and now if we do using this flip flops.

(Refer Slide Time 53:09)



Indian Institute of Technology, Kharagpur

Implementation with JK FFs

$JC = A$

$KC = A/$

$JB = 1$

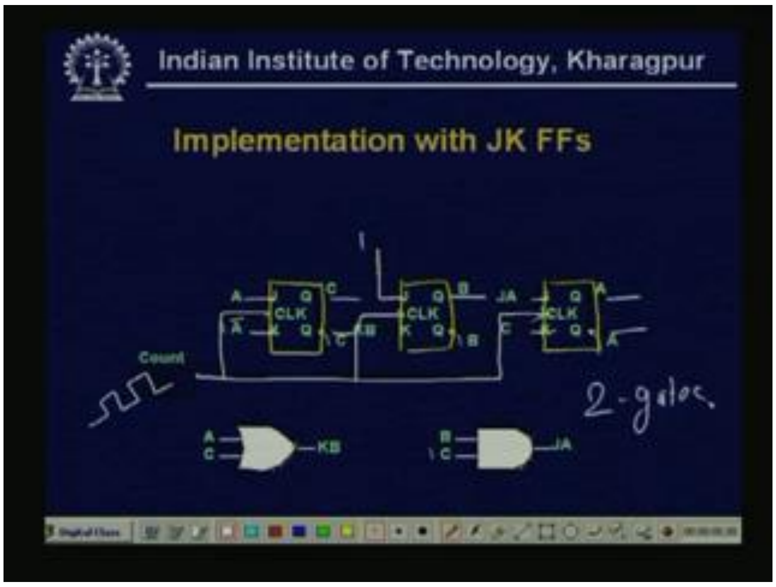
$\longrightarrow KB = A + C$ 1

$\longrightarrow JA = B\ C/$ — 1.

$KA = C$

That do the design then we will be getting the functions from the k map at that J C equal to A then K C is A bars A bar J B is 1 K B is A plus C J A is B C bar and K A is C. So, see here I need one gate here two gate K B, one OR gate here. We need actually two gate one AND and 1 INVERTER or if we take the C bar as the literal then we take only 1 AND gate here. So, here we need only 2 gates whereas, in RS flip flop the design with the RS flip flops we have needed 3 gates 3 extra logic gates, so if we consider the thing.

(Refer Slide Time 54:07)

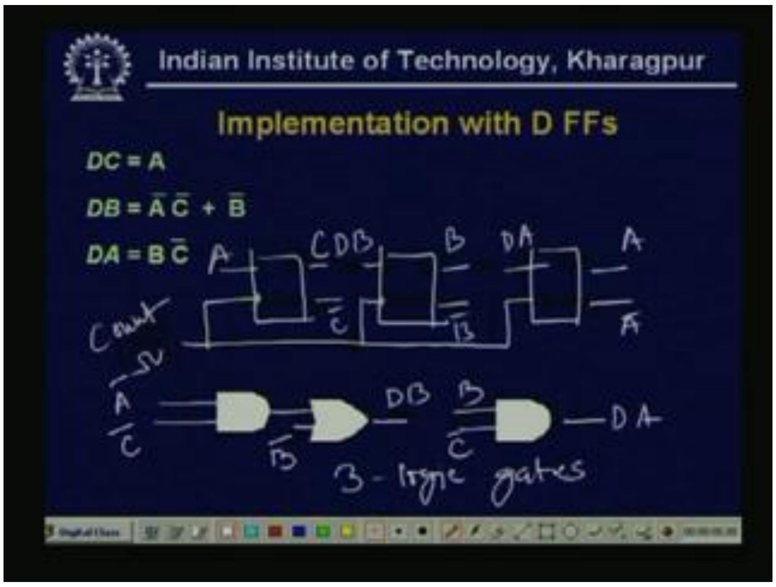


Indian Institute of Technology, Kharagpur

Implementation with JK FFs

See here K B is A C K B is A C and J A is B C bar, so that means, if we again this is my clock. So, this is my clock and it is shared to three flip flops then this is my A and this is A bar this is my A this is A bar J A C B this is 1 C A, A bar C bar. So, in this case we need 2 extra gates 2 extra gates with JK flip flop, now if we do the implementation with D flip flops.

(Refer Slide Time 55:29)



Indian Institute of Technology, Kharagpur

Implementation with D FFs

$DC = A$

$DB = \bar{A}\bar{C} + \bar{B}$

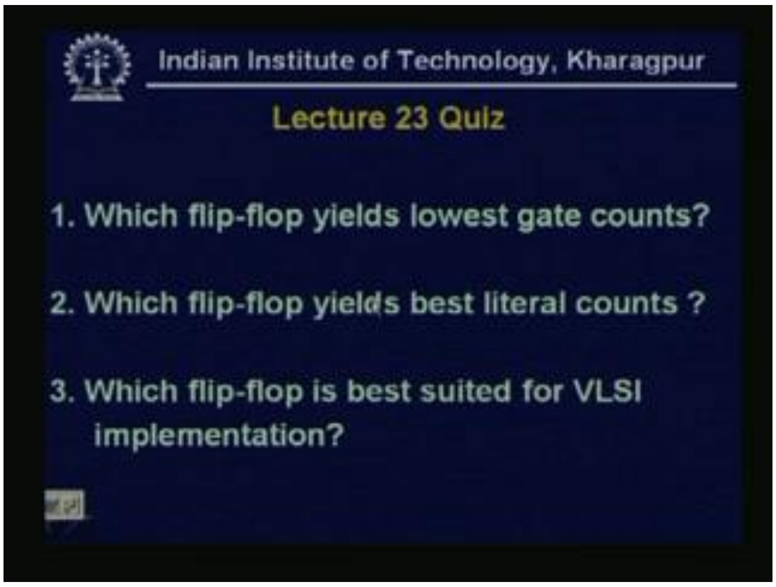
$DA = B\bar{C}$

Simple D flip flop then it will be that logics are D C equal to A D B equal to A bar C bar plus B bar and DA is B bar C bar. So, if we first we compute DB, so this is my DB. So, this will be B bar C B bar and then A bar C bar, so another is this is A bar and this is C bar. So, as if this is my D B. What is my D A? My D A will be B C bar, so this is B and this is this is C bar and if again if I consider 3 D Flip flops 3 D Flip flops then this input is A this input is DB this is my C this is my C bar.

This is my B this is B bar this is my DA and this clock because 1 single input this is my clock, clock goes to every flip flop. So, actually this is my count with every count it will generate the circuit as A and this is my A bar. So, see here also three logic gates are needed three logic gates are needed, but this is D flip flop is always very much suited for VLSI implementation. Because it has only one input and this is the simplest flip flop as last time we have read.

So, here also we have seen that 3 logic gates that is needed for R S flip flop here also three extra logic gates are needed. But the literals are minimum literals are there literal count is less. So, we have seen a 1 1 counter design 1 5 state counter design with 3 different type of flip flops mainly the R S flip flop JK flip flop and D flip flop. Now we see the today's quiz.

(Refer Slide Time 57:42)



Indian Institute of Technology, Kharagpur

Lecture 23 Quiz

1. Which flip-flop yields lowest gate counts?

2. Which flip-flop yields best literal counts ?

3. Which flip-flop is best suited for VLSI implementation?

That first one is which flip flop yields lowest gate counts and which flip flop yields best literal counts which flip flop is best suited for VLSI implementation.

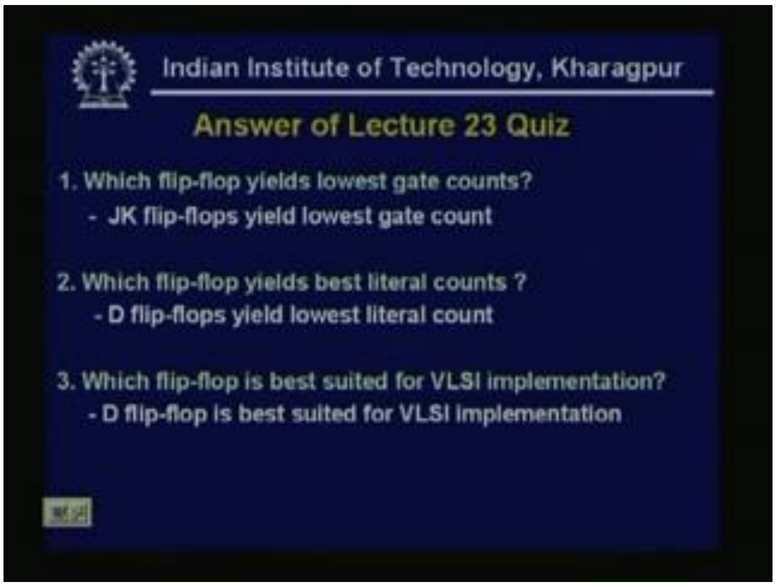
Thank you.

Digital System Design
Prof. D.Roychoudhury
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 24
Finite State Machine Design

Last two classes we have read how to design the synchronous sequential circuit. Now today, we will read the Finite State Machine Design again. That is a sequential machine and synchronous sequential machine.

(Refer Slide Time: 58:55)



Indian Institute of Technology, Kharagpur

Answer of Lecture 23 Quiz

1. Which flip-flop yields lowest gate counts?
   - JK flip-flops yield lowest gate count

2. Which flip-flop yields best literal counts ?
   - D flip-flops yield lowest literal count

3. Which flip-flop is best suited for VLSI implementation?
   - D flip-flop is best suited for VLSI implementation

Before that, we quickly see the answers of the lecture 23 quiz. Now the question is which flip flop yields lowest gate counts and as already we have J K flip flops, D flip flops, R S flip flops and we have seen than J K flip flops yield lowest gate count. Now, which flip flop yields best literal counts, the most popular D flip flop yield literal count and which flip flop is best suited for VLSI implementation. Again D flip flop is best suited for VLSI implementation.