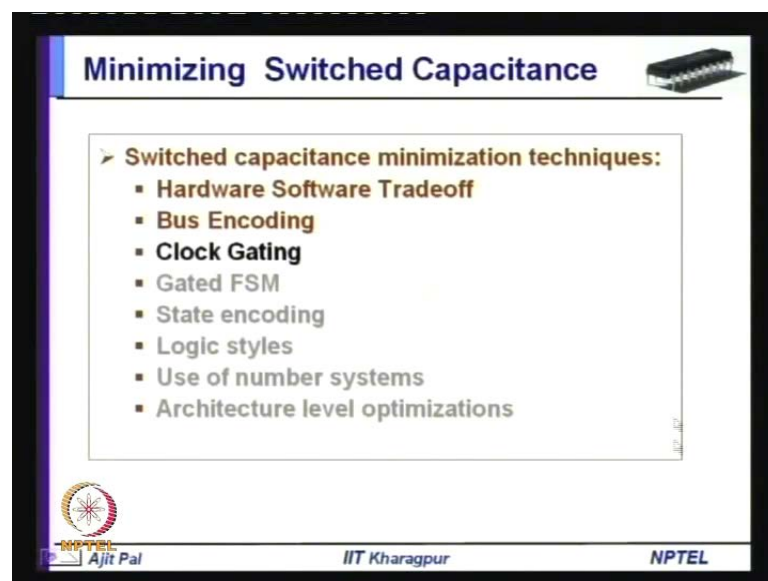


Low Power VLSI Circuits and Systems
Prof. Ajit Pal
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture No. # 29
Minimizing Switched Capacitance-III

(Refer Slide Time: 00:26)



The slide is titled "Minimizing Switched Capacitance" and features a small image of a microchip in the top right corner. The main content is a list of "Switched capacitance minimization techniques" enclosed in a box. The techniques listed are: Hardware Software Tradeoff, Bus Encoding, Clock Gating, Gated FSM, State encoding, Logic styles, Use of number systems, and Architecture level optimizations. The slide also includes logos for IIT Kharagpur, NPTEL, and the presenter's name, Ajit Pal, at the bottom.

Minimizing Switched Capacitance

➤ **Switched capacitance minimization techniques:**

- **Hardware Software Tradeoff**
- **Bus Encoding**
- **Clock Gating**
- Gated FSM
- State encoding
- Logic styles
- Use of number systems
- Architecture level optimizations

Ajit Pal
IIT Kharagpur
NPTEL

Hello and welcome to today's lecture on minimizing switch capacitance; this is the third on this topic. In the previous two lectures, we have discussed about hardware software tradeoff and bus encoding; in the last lecture, I have discussed about bus encoding, and we have seen, how the switch capacitance can be minimized by encoding the signal before sending over the bus.

(Refer Slide Time: 00:54)

Clock Gating

- A major component of processor power is the clock power (50% of the total power)
- One of the most successful and commonly used low power technique is clock gating
- It is based on the observation that a large number of transitions are unnecessary
- Such transitions can be suppressed without affecting functionality

The slide includes a pie chart illustrating the breakdown of power. A large pie chart on the left is divided into 'Others' and '50% clock related power'. An arrow points from the '50% clock related power' section to a smaller pie chart on the right. This smaller pie chart is divided into 'Clock loading' (10%), 'Local Clock drivers' (45%), and 'Global clock drivers' (13%). The 'Global clock drivers' section is further divided into 'Sequential cell internal clock' (32%).

NPTEL Ajit Pal IIT Kharagpur NPTEL

And today, I shall focus on another very important technique known as Clock Gating. And clock gating is has been found to be very popular and very useful technique. Now, why clock gating? It has been found that a major component of processor power is the clock power; and it has been found that 50 percent of the total power may be attributed to clock related circuits; that means, clock is generated, it is distributed throughout the chip and at different points of time, you will be you will see that we are providing drivers and those, because the switching of the clock, the dynamic power dissipation is quite large and you can see here fifty percent is due to clock related power.

So, others I mean others are due to other sources, but I mean process are in other things, but circuits related to power dissipate 50 percent and of course, here, is the breakup clock loading 10 percent 45 percent local clock drivers 13 percent global clock drivers and 32 percent sequential and internal clock.

So, you can see if, we can minimize on the power dissipation in clock related circuits that can be significant saving in power dissipation. So, this is, what we shall discuss today and it has been found that this is, the most successful and commonly used low power technique widely used.

I mean among the digital circuit designers this is possibly the most popular and widely used technique what is the basic idea? Basic idea is it is based on the observation that a large number of transitions are unnecessary; that means, when you are applying a clock it

is making transitions and those transitions are always not necessary particularly, we will find that some parts of the circuits are not in use or their function is not **is not** the outputs are not used in such situations. They can be shut off that is the basic idea; that means, such transitions unnecessary transitions can be suppressed without affecting the functionality. This is the key idea of incorporating clock gating; that means, you have to stop clock to different parts of the circuit without affecting the functionality of the circuit and how it can be done we shall discuss today.

(Refer Slide Time: 03:27)

Clock Gating

- Usage of a functional element is highly dependent on the application in hand
- There are opportunities to shut off circuits that are not in use
- Dynamically preventing the clock from propagating to some parts of the circuit under certain conditions

Active mode (clock running) Idle mode (clock fixed) Active Idle

dynamic Leakage

NPTEL Ajit Pal IIT Kharagpur NPTEL

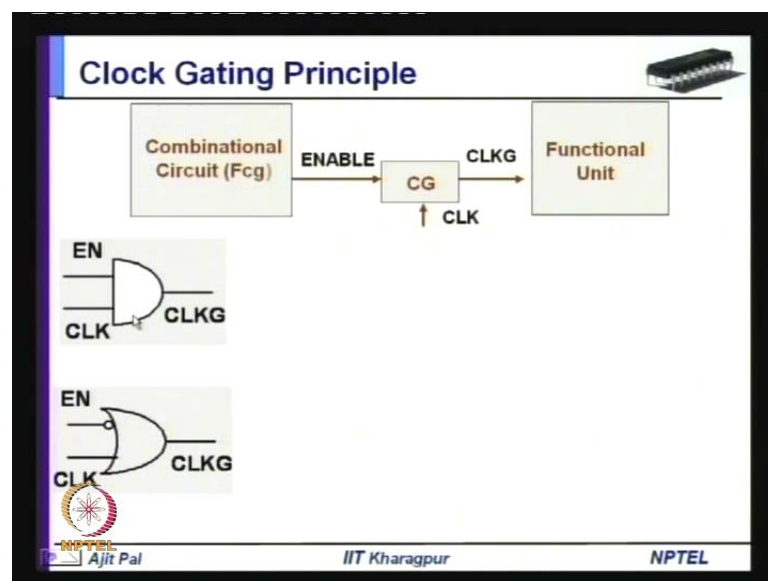
Now, one another very important observation is you see the uses of a functional element is highly dependent on the application in mind; that means, a particular functionality or function, which has been implemented by hardware, when it will be used; how it will be used that, dependent on the application. So, same function can be used differently by different applications.

So, best it is actually it cannot be really decided independent of the application. We have to take into consideration the application in hand, and which we will decide, which function is required and when it is required. So, there opportunities to shut off circuits, that are not in use. That is, the basic idea and dynamically, preventing the clock from propagating to some parts of the circuit, under certain conditions is essentially clock gating.

So, you can see here, you have a circuit which is in operation and when particular part of the circuit is in active mode, you can see clock is applied and two different types of power dissipation is taking place one is dynamic that is, because of charging and discharging of capacitors as, clock is applied with another component is link component. So, you see if the activity of a particular block can be stopped by shutting off the clock. I mean enabling disabling the clock then, you can see; only leakage power dissipation will take place, but the dynamic power dissipation will not take place.

So, you can see; this is how, you can plot the activities over time. This is x axis; this is a long time. So, in this path it is not clock gated. In this part in the ideal mode, it is clock gated and again, when they say you need the circuit to be used or then, it is in active mode and again, you can put it in the idle mode by fixing the clock. So, this is how, the power saving can be done; dynamic power saving can be done by using clock gating.

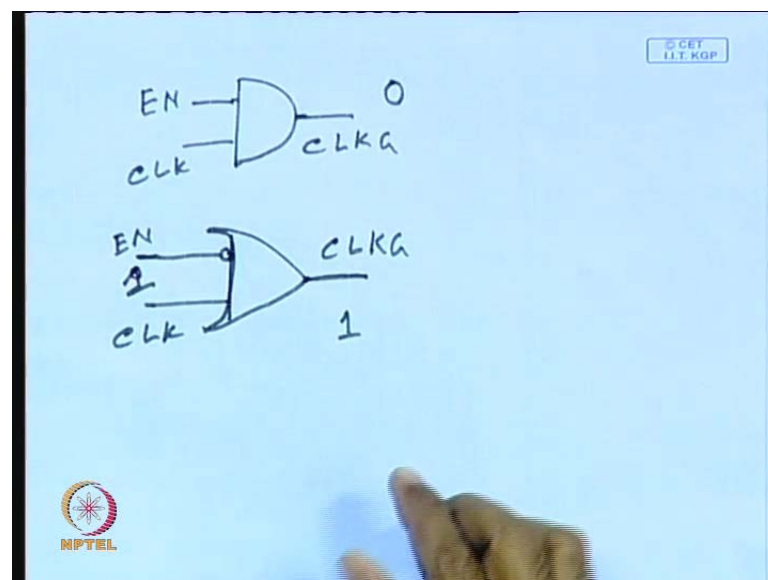
(Refer Slide Time: 05:44)



Basic idle is shown here, it can be a Functional Unit it can be a register, it can be A L U it can be any component or it can be receiver it can be anything. So, that functional element receives a clock normally, it comes from the directly from the clock circuit, but what you can do? you can now, add one additional circuitry that is, F c g that F c g actually identifies when clock need to be applied to this Functional Unit; that means, there are situations where clock need, but need not be applied to this Functional Unit.

So, this come the job of this Combinational Circuit is to identify, when an enable signal has to be applied to a clock gating circuit, CG is a clock gating circuit. And this clock gating circuit will allow the clock whenever; it is enabled to reach the Functional Unit; that means, when the enable signal is one then this clock will reach the functional unit. And so, you can see here, the clock that is being applied to the functional unit is gated clock. So, clock G is the gated clock, that is applied and instead of the clock. So, you have added two additional components, one is the Combinational Circuit that identifies, when a particular Functional Unit has to remain in active mode, and another is clock gating circuit which disables or enables the clock to reach the functional unit.

(Refer Slide Time: 07:27)



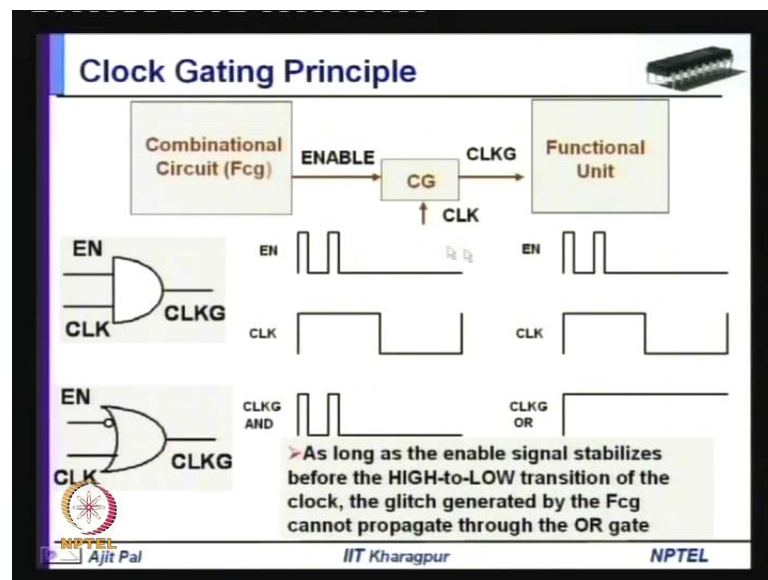
Now, how it can be realized one simple circuit can be AND gate. So, you can see if you use an AND gate the operation is very simple. So, when the enabled signal is 1 then, this clock is allowed to go to the output, so CLKG. So, gated clock you get the output at the clock; that means, when the enable signal is 0 then, output is that is, in case of AND gate output will be 0. So, clock does not reach the gated clock output, when the enable signal is 0 another way of doing it is to use an OR gate.

So, in case of an or gate enable signal is applied to the low active and here, is your clock and it is one; that means, when this is one when enable signal is one; that means, enable signal is one means then output will be; that means, output enable signal is one; that means, it will be it is low active.

So, when this is 0 only then depending on the clock it will change, but when the enable signal is one then enable signal is one means this will be 0.

So, enable signal is one that; that means, according to the clock it will change. So, 1 means this continuation of this will be 0, but whenever; the enable signal is 0. Then it will be always 1; that means, when the enable signal is 0 then instead of allowing the clock to reach here, it will always remain 1, on the other hand here when enable signal is 0 then output is 0.

(Refer Slide Time: 09:36)



So, you can see the functionality is different here, the output is you can say clamp to 1 when it is not enable and when it is not enable here, it is clamped to 0. So, clock is not reaching here, here, also it is clock gated output. So, the two circuits function in a different way, but perform the same operation it does not allow the clock to reach the output, but if we consider this circuit it has some limitation. What is the limitation? Limitation is that for example, there is some glitch in the enable signal. So, if there is a glitch in the enable signal and when the clock is one, you will see the glitch will reach the gated output.

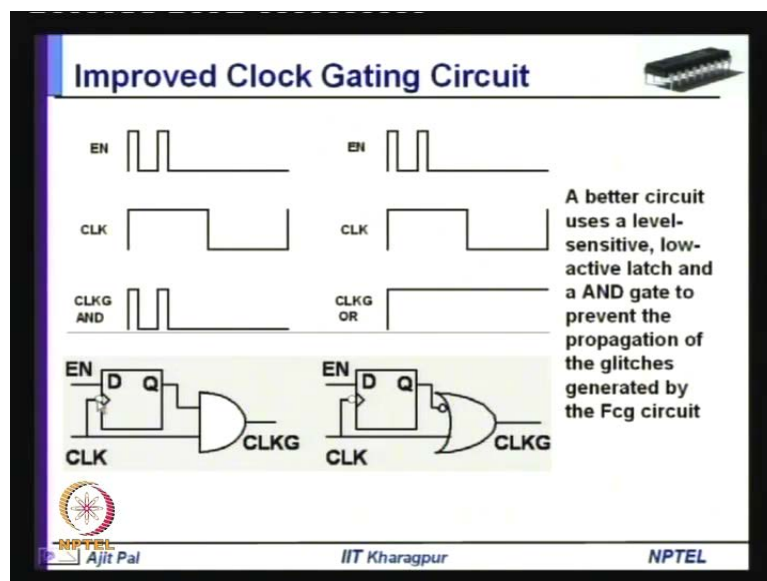
So, this Combinational Circuit may be a multilevel Combinational Circuit and, because of the delay in the gate, gates it may generate some glitch. We have already discussed how glitches can be generated in a multi-level circuit, because of the finite delay of the gates. So, whenever there is a delay generated by the enable signal then, you can see this

glitch will reach the output of the I mean clock gating circuit, and as a result what will happen this Functional Unit may malfunction or there will be this may lead to malfunctioning, because you know it may be latching some data with the help of this clock.

So, it will be latching some unnecessary data or if, it is a Combinational Circuit then, it will lead to glitching power dissipation. So, in either way it is not acceptable to ask. So, this type of glitching has to be prevented fortunately, this type of glitching is prevented in this OR circuit. How is prevented whenever ; the enable signal you can see enable is 1 then it is going it is going I mean this enable signal is going to 0 to 1.

So, when in such a case as we know when the when it is 0 only then it is activated; that means, this 1 will not reach then output, because the output is clamped already to 1 whenever; the clock is 1 output is 1. And so, this glitch will not reach the output only requirement here is that, this glitching has to be stopped. I mean has to be over within this part within as when the as long as the enable signal stabilizes, before the high to low transition of the clock the glitch generated by the Functional Unit that, Fcg cannot propagate through the OR gate; that means, normally, it is considered that all the glitching activities, because the delay within this part will be over during this duration; that means, half period of the clock.

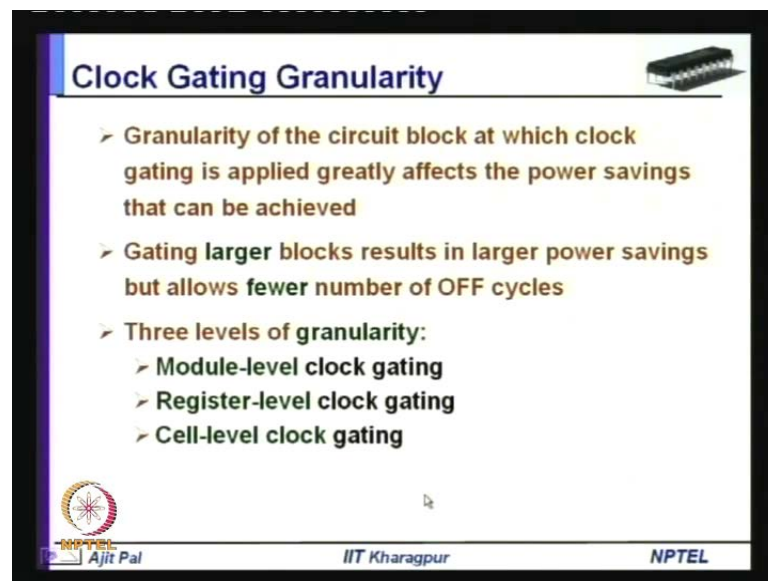
(Refer Slide Time: 12:30)



So, as long as that is satisfied block will not glitch will not reach the output this is a slightly better circuit than, the and gating; however, more robust use of this is with the help of additional latch. So, a better circuit uses a level sensitive low active latch and a AND gate to prevent the propagation of the glitches generated by the Fcg circuit. So, here you see in this particular case now, instead of directly connecting the enable signal to this input you are applying through a latch. So, this is a level sensitive low active latch.




So, since, it is low active latch you see unless you know the enable signal is not enabled before this part. So, enable signal does not reach here. So, as a consequence these glitches will not reach the output here and as a consequence the gated clock signal will be glitch free and same is true for this or circuit here also you can add a latch. So, this either this AND gate based clock gating circuit or OR gate based clock gating circuit is commonly, used for gating the clock in different parts of the circuit. So, this is about the clock gating circuit.

(Refer Slide Time: 13:49)



Clock Gating Granularity

- Granularity of the circuit block at which clock gating is applied greatly affects the power savings that can be achieved
- Gating larger blocks results in larger power savings but allows fewer number of OFF cycles
- Three levels of granularity:
 - Module-level clock gating
 - Register-level clock gating
 - Cell-level clock gating

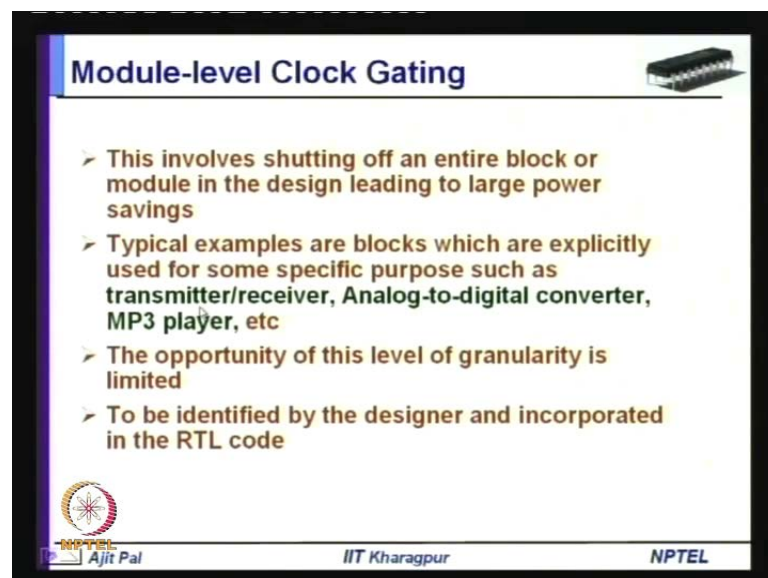
 Ajit Pal  IIT Kharagpur  NPTEL

Now, let us, focus on one very important aspect that is your Clock Gating Granularity. So, Granularity of the circuit block at which clock gating is applied greatly affects the power saving that can be achieved. So, you know and this granularity means what is the size of the block that is being clock gated and there are three possible alternative first is Module Level clock gating, second is Register level clock gating and third is Cell level

clock gating and Gating larger blocks results in larger power savings, but allows fewer number of OFF cycles; that means,




You know whenever you choose a large block say C P U and C P U need not be cannot be always made off say turned off, but if it is a small circuit say a register or maybe say an analog to digital converter which can be turned off more frequently, than a C P U. And so, gating large block result means larger power savings, but allows fewer number of off cycles; that means, the number the opportunity for applying clock gating is less, when the size of the block is larger.

(Refer Slide Time: 15:15)



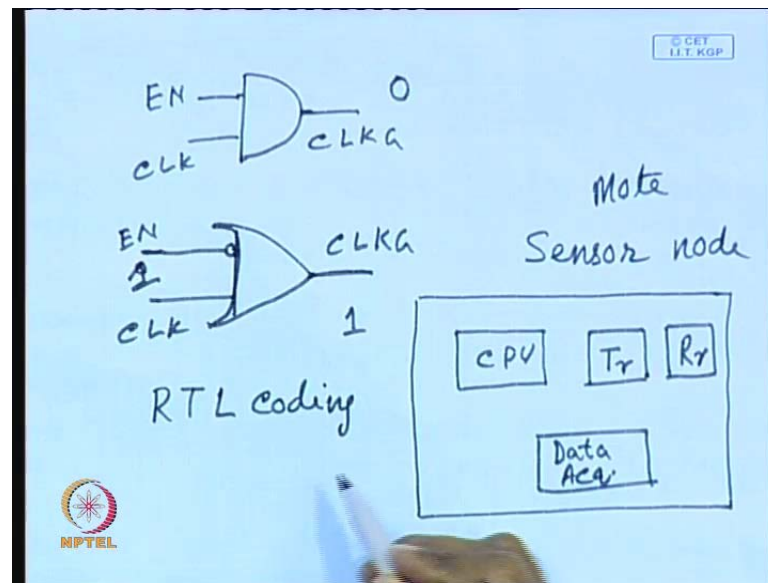
Module-level Clock Gating

- This involves shutting off an entire block or module in the design leading to large power savings
- Typical examples are blocks which are explicitly used for some specific purpose such as transmitter/receiver, Analog-to-digital converter, MP3 player, etc
- The opportunity of this level of granularity is limited
- To be identified by the designer and incorporated in the RTL code

 Ajit Pal  IIT Kharagpur  NPTEL

So, first we shall focus on Module Level Clock Gating this involves setting of an entire block or module in the design leading to large power saving typical examples, are blocks which are explicitly used in some specific purpose such as transmitter receiver analog to digital convertor MP3 player, etc.

(Refer Slide Time: 15:38)



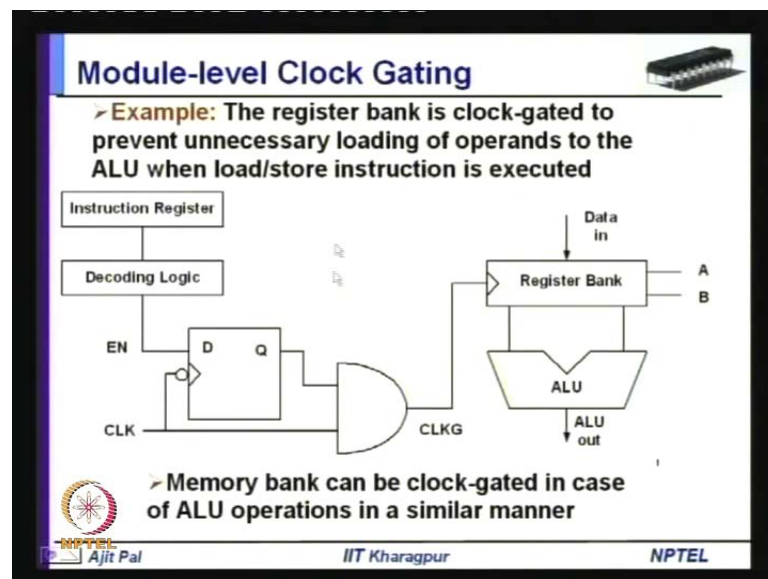
We can consider one very, I mean commonly used example, I mean which is becoming increasingly popular for example, a sensor node. So, this is also known as mote sensor network is becoming increasingly popular and if, you look at any sensor node you will find it has got these building blocks, there is one processor you may call it CPU there will be a rf circuit consisting of transmitter and receiver. You may call it transceiver and also you will require some Data acquisition circuit DAC and shown say data acquisition circuit. So, at least these four components are required in a sensor node. So, normally the sensor node operates in this way it acquires some physical data may be temperature pressure humidity or any other parameter or humidity of sand like that and that is done with the help of this data acquisition system.

So, when this data acquisition system is working at that time CPU along with Data Acquisition Circuit has to remain on. So, these two remain I mean it should remain on, but transmitter and receiver can be turned off and similarly after the Data Acquisition is done the only some pre processing or some processing is necessary that can be accomplished only by this CPU and at that point of time the data acquisition circuit transmitter and receiver all the three can be shut off by clock gating and when you require to send the data to another part of the network of a sensor network then, you will **you will** turn on the transmitter and with the help of which you will send the data to other nodes or a base station similarly, when you are receiving data from other nodes or from base station you turn on the receiver.

So, you can see here, you have the opportunity to selectively turn on and off different components different building blocks within a sensor node or node and this is **this is** where you can use this Module level Clock Gating. So, the opportunity of this level of granulate is limited, but all that is limited it can be used in many applications. So, whenever; you are using Module level Clock Gating then, I mean when a particular module or sub module can be turned off or turned on that has be decided by the designer; that means, the person who is doing RTL coding.

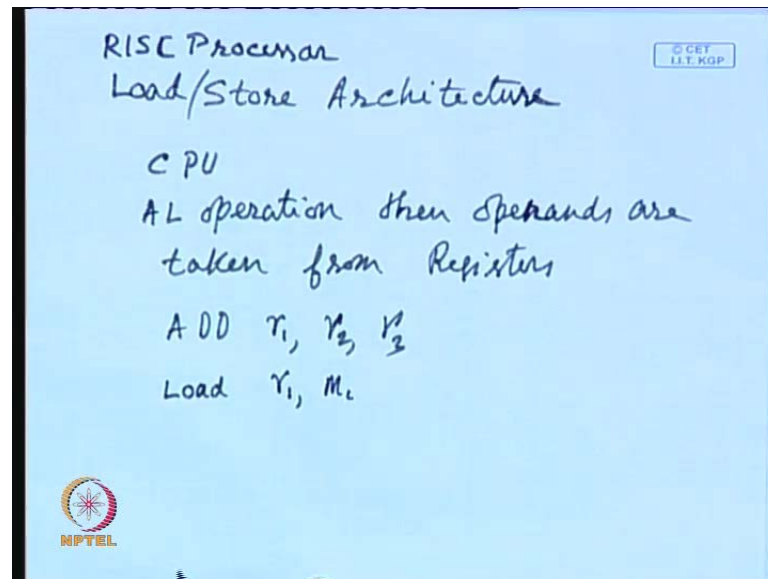
He has to identify when a particular sub system can be shut off or that clock of that particular circuit can be turned off and accordingly RTL coding has to be done to turn on turn off the different parts of the circuit. I mean clock gating circuit can be turned on and off. So, Clock Gating is incorporated in RTL level in such a case and it is **it is** done with the help of I mean human it is manual it is done with the help of a by designer.

(Refer Slide Time: 19:13)



And here, is an example of a Module level Clock Gating here, the register bank is clock gated to prevent unnecessary loading of operands to the ALU when load store instruction is executed. All of you may be familiar with Risc processor **the Risc processor processors**.

(Refer Slide Time: 19:39)



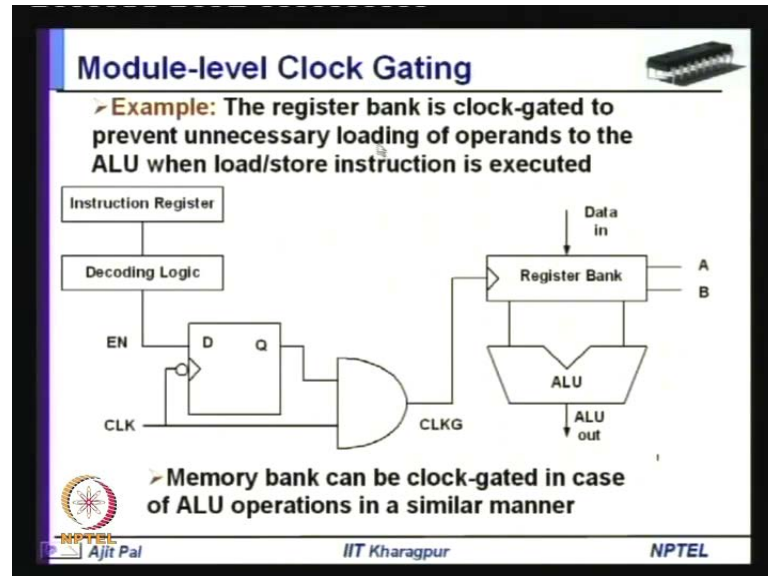
And that interesting feature and that interesting feature is known as they have Risc Processor using instruction set computer. They have one interesting features that is you they have Load Store Architecture. What does it mean? It means that whenever performs any C P U operation; that means, performing any A L U Arithmetic and Logic Operation then operands are taken operation **then operands are taken are taken** always from registers; that means, for example, it can be a ADD r 1, r 2, r 3; that means, content of part two is added with content of r 3 result stored in r 1.

That means, you cannot perform add operation with the content of register with that of a memory location. So, that means, all the arithmetic and logical operations are performed with the Registers. So, you may say where from we shall get the data these Registers you have to use specific Load instructions to Load the value of data in r1 and from a particular memory location m2. So, this is a explicit Load instruction. You have to use to load data into the Register and then you can perform these Arithmetic and Logical Operations.

So, similarly you can store the data in m_i location from a register say r_i . So, you have to use explicit Load Store instruction to Load register or Store data in a memory location. When you are when you are performing Arithmetical Logical Operations you can see it does not involve **does not involve** access of memory; that means, Access of memory is

not required when you are performing arithmetic and logical operation this can be exploited in clock gating and when that is being done here.

(Refer Slide Time: 22:16)

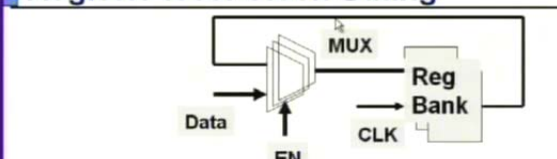


Here you can see an instruction has been fetched. So, the Register bank is clock gated to prevent unnecessary loading of operands to the ALU when load store instruction is executed; that means, after the instruction is fetched decoder Decoding Logic can identify whether it is a load or store instruction. And if it is a load and store instruction then it can disable this circuit; that means, enable is 0; that means, clock is not allowed to be applied this Register Bank. So, unnecessary loading of these operands to this ALU. I mean will not take place as a consequence you know this part lot of power saving can be done in this part of the circuit. So, you see here, Clock Gating is done at module level.

Similarly, you know when you are performing say ALU operations then memory bank can be clock gated. So, as you know you already know that whenever you are performing arithmetical and logical operation memory will not be accessed. So, memory modules can be clock gated when you are performing arithmetic operations. So, in this way at Module level Clock Gating can be done and sometimes the ALU along with register bank can be Clock Gated, sometimes the Memory Bank can be clock gated depending on the instructions that you are executing type of instruction that you are executing.

(Refer Slide Time: 23:48)

Register-level Clock Gating



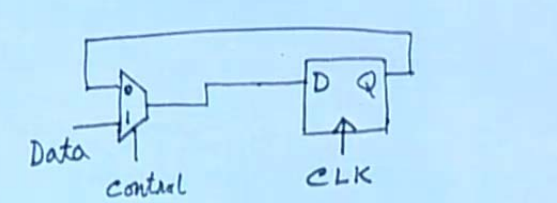
The diagram shows a 'Data' input entering a 'MUX' (Multiplexer). The output of the MUX goes to a 'Reg Bank' (Register Bank). The 'Reg Bank' has a 'CLK' (Clock) input. An 'EN' (Enable) signal is shown entering the MUX from below. A feedback loop is shown from the output of the Reg Bank back to the input of the MUX.

- The clock to a single register or a set of registers is gated
- A synchronous load-enabled register bank typically implemented using clocked D flip-flops and a recirculating multiplexer
- The register gets the clock in all the cycles even when no new data is loaded
- Gating condition for registers may be obtained through symbolic analysis of the combinational circuit that feeds it

NPTEL
Ajit Pal
IIT Kharagpur
NPTEL

So, this is an example of Module level Clock Gating now coming to the Register level Clock Gating here, normally how let us, consider a register and how the **how the** data is loaded in a register.

(Refer Slide Time: 24:10)



The hand-drawn diagram shows a 'Data' input entering a 'control' block. The output of the control block goes to the 'D' input of a 'D Q' flip-flop. The 'CLK' (Clock) input of the flip-flop is connected to the output of the control block. A feedback loop is shown from the 'Q' output of the flip-flop back to the 'Data' input.

© CSE
IIT KGP

NPTEL

Let us, assume this is a register bank let us, consider a single register before we consider a register bank. So, this is a register and to this register you can load data normally, it is a it is implemented with the help of a d flip flop and d flip flop has got a Clock input and then sometimes new data will be coming from outside which has to be loaded. So, that

data is coming from outside and you can Load it with the help of a multiplexer. So, this will go to the multiplexer, but when you are not loading the Data into this multiplexer then the old data which is already stored there is re circulate.

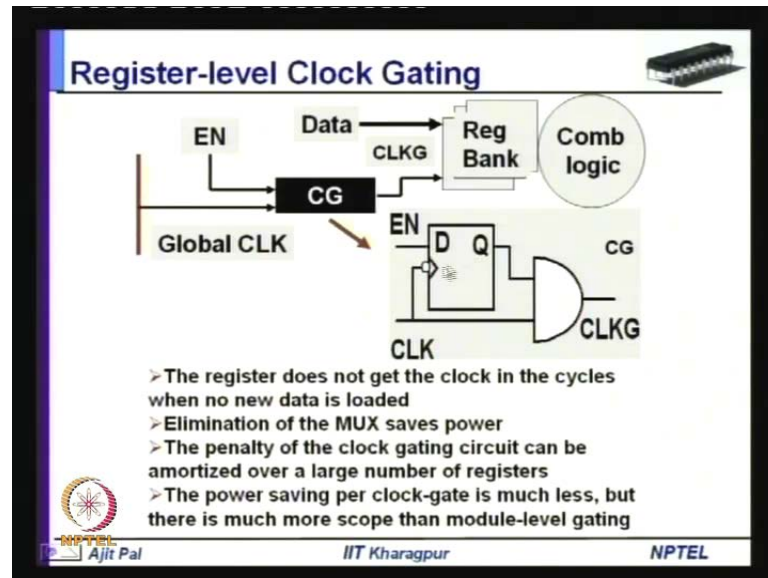
So, clock is always applied there each time clock is applied and a control signal decides which will be applied to this D input; that means, when new data has to be loaded this path say you may call it 0 path you may call it 1 path when this path is enabled; that means, control signal is 1 and this signal will this data will go to this register.

So, only, but in all other situation when new data is not loaded into the register you know old data is loaded, because clock is already applied there old data is re circulate with the help of this multiplexers. So, whenever you have got a register bank; obviously, this multiplexer is duplicated the flip flops are duplicated and so on. But the basic operation is operation remains the same.

So, in this case you can see when you are not loading new data into this register you are unnecessary dissipating power by applying the clock and re circulating the old data and; obviously, this can be clock gated. So, the clock to a single register or a set of registers can be gated because this type of unnecessary loss of power can be prevented.

So, a synchronous load enabled register bank typically is implemented using clock d flip flop which I have already explained and a re circulating multiplexer as I have shown and explained and the register gets the clock in all the cycles even when no new data is loaded. So, getting condition for registers may be obtained through symbolic analysis of the combination circuit that feeds it; that means, you can perform symbolic analysis of the circuit which is feeding data here, and find out the situation when this data needs to be loaded into this resistance bank and you can insert clock gating the way it is shown here.

(Refer Slide Time: 26:57)



So, here as you can see your Global Clock is not applied to the Register Bank directly, but a gated clock is applied and an enable signal is generated with the help of a additional hardware that FCG which I have mentioned earlier and that will generate the enable signal and that enable signal can be found out by symbolic analysis of the combinational circuit that feeds it you can find out the situation. When new data needs to be loaded and only then a clock is applied to this Register Bank and as we can see here, you can that clock gating circuit. I have already explained in detail that type of circuit you can use for getting the clock.

And the in this case you can see, when you are not loading data new data then unnecessarily you are dissipating power not only in the register bank, but the combinational logic that the register bank is feeding, because at the output of the register bank changes the they will transitions on the within the Combinational Logic and power dissipation will take place. So, all these so; that means, you are preventing transitions on this part; that means, that clock output of the gated clock to the Register Bank on this path there is some finite capacitance here, you are preventing unnecessary transitions within this Register Bank and also in the Combinational Circuit.

No not only that you are in this case the register bank does not get clock in the cycles when no new data is loaded. So, on very few occasions it will receive the clock and elimination of the MUX saves power. So, earlier we have seen it is implemented with the

help of recirculation multiplexer and which is now, replaced with the help of a clock gating hardware and particularly a single clock gating hardware may be shared by a large number of Registers and as a consequence instead of multiple Multiplexers only one Clock Gating Circuit will be required and that will save power as well save power not only power area later on we shall see some case study result.

And the penalty of the clock gating circuit can be amortized over a large number of registers as I have mentioned it need not be a single register and it can be large number of register three four ten depending on. They can be fed by the same clock and the power saving per clock gate is much less, but there is much more scope then module level gating; obviously, this type of clock gating can be done more frequently, then, what it can be done in the module level clock gating.

(Refer Slide Time: 30:00)

Challenges in Massively Gated Clocks

- Provides many more opportunities and it lends itself to automated insertion and can result in massively clock gated designs
- **Massively clock gated design issues:**
 - Clock latency
 - Effect of clock skew
 - Clock-tree synthesis
 - Clock gating leads to additional complexity to synthesis and analysis tools making testing and verification difficult

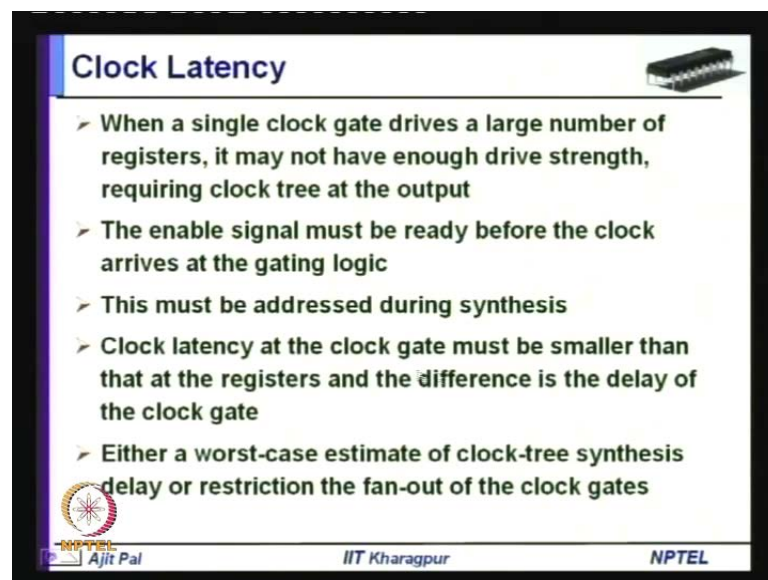
NPTEL Ajit Pal IIT Kharagpur NPTEL

Now today there is opportunity to implement clock gating in a massive way and automated tools and techniques are available by which this can be done. So, it has provides many more opportunities and it lends itself to automated insertion of clock gating hardware and can result in massively clock gated designs. So, it is a reality in the present day v l s i circuit design; however, whenever you got for massively clock gated design there are many issues and challenges you have to overcome and some of the challenges are mentioned here, number one is clock latency number two is effect of clock skew number three is Clock tree synthesis.

Which I shall discuss in little more detail and another important issue is clock gating leads to additional complexity to synthesis and you know you have to perform what is known as statistic timing analysis. So, as you add clock gating hardware delay of this circuit increases and static timing analysis becomes more complex and difficult and not only static timing analysis.

But testing you know you have to incorporate you have to keep provision for testing built in self-test bit circuit has to incorporated and that becomes more complex and verifications become more difficult. So, these are the things which will be faced whenever we go for massively clock gated design, but in spite of these problems challenges and they can be used and let us, discuss some of the challenges.

(Refer Slide Time: 31:43)



Clock Latency

- When a single clock gate drives a large number of registers, it may not have enough drive strength, requiring clock tree at the output
- The enable signal must be ready before the clock arrives at the gating logic
- This must be addressed during synthesis
- Clock latency at the clock gate must be smaller than that at the registers and the difference is the delay of the clock gate
- Either a worst-case estimate of clock-tree synthesis delay or restriction the fan-out of the clock gates

Ajit Pal IIT Kharagpur NPTEL

Number one is clock latency. So, when a single clock gate you know normally in a synchronous sequential circuit. We assume that a single clock is driving all the circuits and we also assume that all the circuits are receiving the clock at the same instant of time which is not a reality in the practical circuits particularly when clock gating is used. So, when a single clock gate drives a large number of registers it may not have enough drive strength. So, you will **you will** be require **a require** a clock tree design you have to design a clock tree not a single clock which you will drive then the circuit. So, you have to use drivers and a clock tree has to be design. So, clock tree design it has become very important in present day digital circuits.

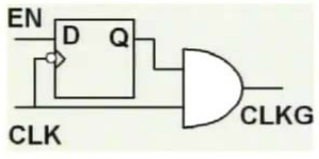
So, the enabled signal must be ready. So, whenever you are doing that clock latency problem will you have to take into consideration, because we have seen the enable signal must be ready before the clock arrives at the gating logic and this must be addressed during synthesis and clock latency at the clock gate must be smaller than, that of the registers and the difference is the delay of the clock gate. So, this type of stringent and timing requirements has to be satisfied.

So, either a worst case estimate of clock synthesis you have to do you have to assume a worst case delay or you have to enforce restriction on the fan out of the clock gates. So, you will find that whenever; a clock gating circuit is used numbers of registers that you will be driving has to be restricted or you have to use drivers later on we shall discuss about the clock gating circuits.

(Refer Slide Time: 33:40)

Effect of clock skew

- Clock skew between the latch and the AND gate may lead to glitches at the gated-clock output



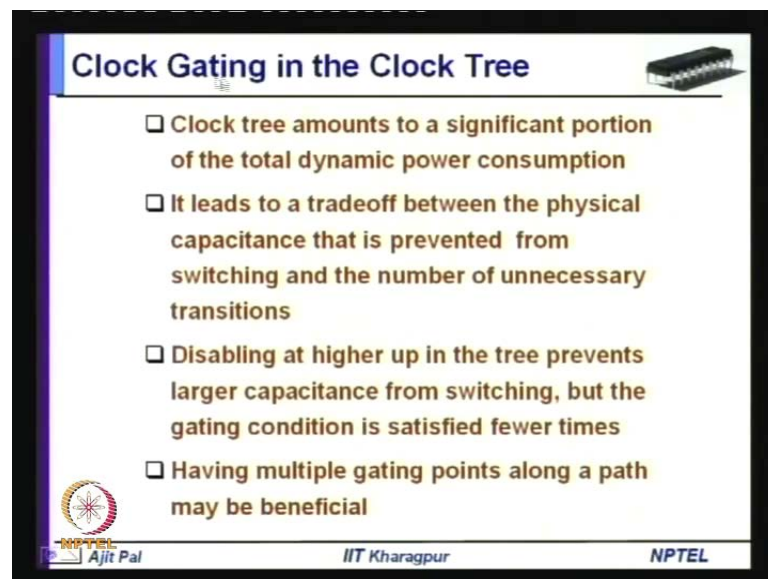
- It is essential to ensure that glitches are not introduced, which may lead to circuit malfunction due to spurious loading of registers
- The clock-skew between the latch and the AND gate should be less than the clock-to-output delay of the latch
- Relative placement of the latch and the AND gate imposes stringent constraints on the clock-tree synthesis tool

NPTEL Ajit Pal IIT Kharagpur NPTEL

Then another important issue is effect of clock skew; that means different parts of the circuits will receive clock at different instant of time. So, clock skew between the latch and AND gate may lead to glitches at the gated clock output. Which I have already explained in detail and it is essential to insure that glitches are not introduced which may lead to circuit malfunction due to spurious loading of the registers which I have already explained when, I mean when I explained the clock gating circuit by using only the AND gate. So, what is the way out the clock skew between the latch and the AND gates should be less than the clock to output delay of the latch.

So, this condition has to be satisfied and this will be achieved by relative placement of the latch and the AND gate imposes I mean stringent constraints on the clock-tree synthesis tool. So, the clock tree synthesis tool will become quite complex because it has to take into consideration the relative placement; that means, placement and routing is closely you have to take into consideration the various factors at the time of placement and routing. So, relative placement of latch and AND gate imposes a stringent constraint on the clock-tree synthesis tool.

(Refer Slide Time: 35:12)



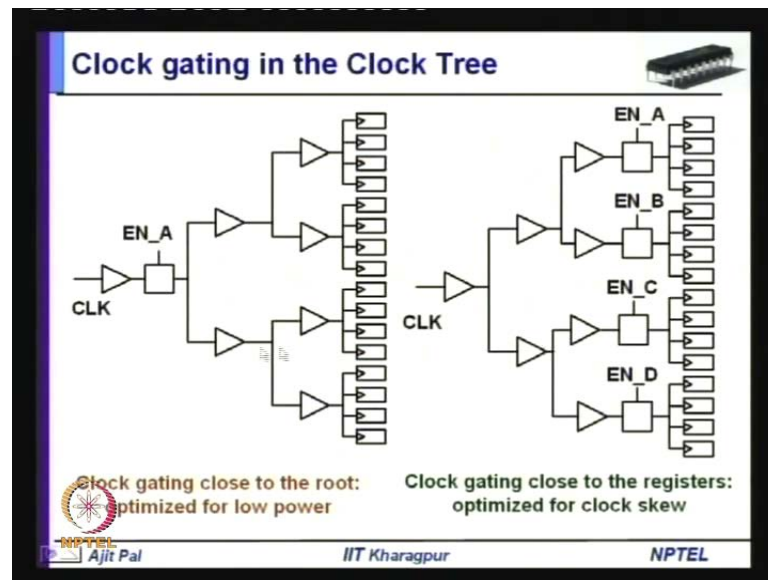
Clock Gating in the Clock Tree

- ❑ Clock tree amounts to a significant portion of the total dynamic power consumption
- ❑ It leads to a tradeoff between the physical capacitance that is prevented from switching and the number of unnecessary transitions
- ❑ Disabling at higher up in the tree prevents larger capacitance from switching, but the gating condition is satisfied fewer times
- ❑ Having multiple gating points along a path may be beneficial

NPTEL Ajit Pal IIT Kharagpur NPTEL

Then, the third problem that is Clock Gating in the Clock Tree. I have already mentioned about the clock tree synthesis and clock tree amounts to a significant portion of the total dynamic power consumption which I have already told. It leads to tradeoff between the physical capacitance that is prevented from switching and the number of unnecessary transitions. So, here, the tradeoff is where you will put the clock gating circuit nearer to the root nearer to the leaf and that will decide the number of clock gating circuits. So, disabling at higher up in the tree prevents large capacitance from switching, but the gating condition is satisfied fewer times.

(Refer Slide Time: 35:59)



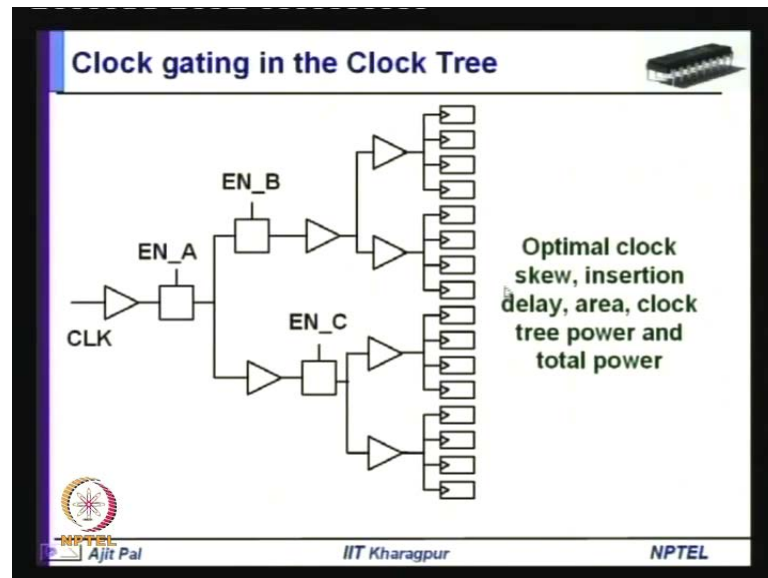
So, as it is explained with the help of this diagram you can see here, is a clock tree you can see here, clock is applied and there are drivers inserted at different points of the circuit and these are the registers which the clock is driving you can see they have some restriction on the number of registers that can be driven by a single driver. So, fan out is restricted to four in this particular case and you can see a number of drivers have been added.

And now question is arises where you will applying the clock gating circuits. So, here a clock gating circuit can be used. So, in this case clock gating circuit can be used here only when all these registers satisfies the condition; that means, the gating of all these registers can be done by a single enable signal only then, this can be done and; obviously, this will save power, because you are using only one clock gating circuit; however, this will lead to clocks skew problem and to overcome the clock skew problem you can place the clock gating circuit closer to the registers; that means, just before the registers you are putting clock gating circuit and directly you are driving the registers. So, you can see it is they are present more or less at the leaf nodes of the Clock Tree.

And the in this case you will require more number of clock gating hardware. So, clock gating is close to registers and this is optimized for clock skew. So, question is will we use this clock gating close to the root or clock gating close to the leaf; that means, close to the registers. So, essentially it is a tradeoff you have it will vary from circuit to circuit

and you have optimize for a particular application in circuit as it is illustrated with the help of this example.

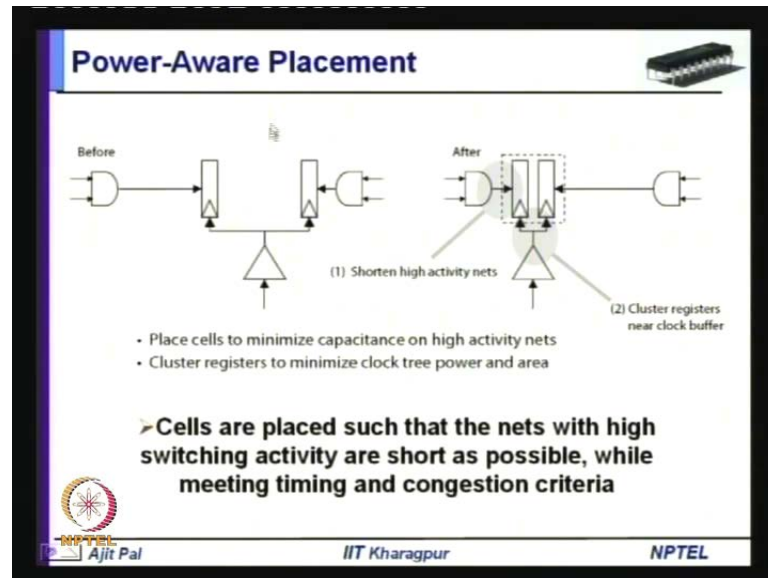
(Refer Slide Time: 38:04)



So, this in this particular case it has been optimized optimal clock skew in such an delay area clock tree power and total power. So, all these parameters have been optimized you can see in each path you have got two clock gating hardware number of clock gating hardware number of buffers all are same, but they have been judiciously, placed some of them close to the root and here, you know clock gating this the clock signal will reach here when both of them are enabled.

Similarly, clock signal will reach here when both of them are enable A and C enable A and enable C. So, these conditions you have to find out and design the enable A the hardware which is generate these enable signals and then you have to place. The clock gating hardware insert buffers and do the placement and routing to optimize clock skew in search and delay area clock tree power and total power.

(Refer Slide Time: 39:24)



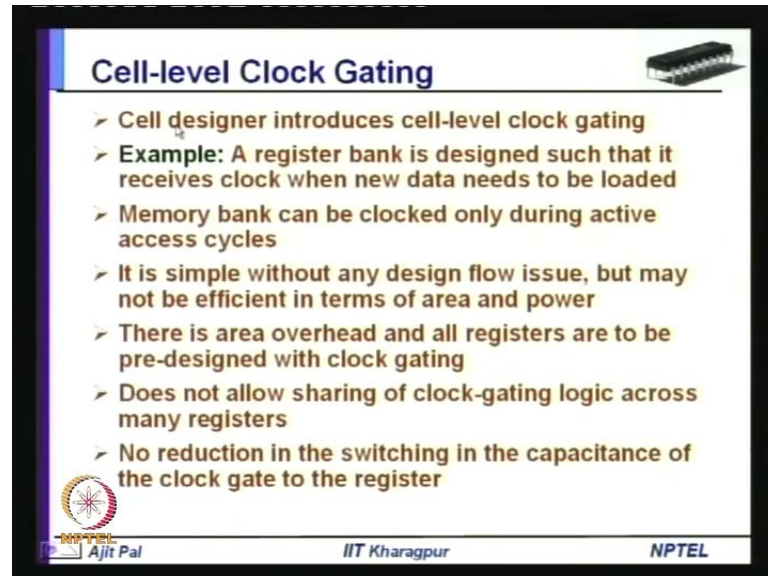
So, this is this shows how the clock gating effects. The clock tree and how the various parameters can be optimized. Now, another very interesting aspect is the placement, I have already told about placement of relative placement of the clock gating hardware and the buffers. So, here, you can see this is a un optimized circuit here, this is a clock gating hardware and there is a signal net and that is input is going there and the clock is coming here. So, this is data is coming here, and signal is coming here.

Now, if, this part of this net signal net is having lot of switching activity then, it is necessary to reduce the length of this particular net. Similarly, you know here is a clock buffer and this buffer is driving several registers it is essential to cluster them together and place them close to each other such that the buffer the capacitance that is driven by the buffer is minimized. So, in this particular case you are reducing the length of this particular net to based on the switching activity and here, you are doing the clustering to reduce the capacitance such that the because you are driving with the help of a single clock drive as a single clock drivers. So, place cells to minimize capacitance on high activity nets. So, here, the capacitance is minimized on high activity nets and cluster registers to minimize clock tree power and area.

So, here, the area and power will be minimized. So, the cells are placed such that the nets with high level switching activity are placed are short as possible while the while meeting the timing and congestion criteria so; obviously, it has to satisfy the timing and

congestion condition criteria; when you are designing this type of clock gate. I mean when you are, performing this placement and routing.

(Refer Slide Time: 41:37)



Cell-level Clock Gating

- Cell designer introduces cell-level clock gating
- **Example:** A register bank is designed such that it receives clock when new data needs to be loaded
- Memory bank can be clocked only during active access cycles
- It is simple without any design flow issue, but may not be efficient in terms of area and power
- There is area overhead and all registers are to be pre-designed with clock gating
- Does not allow sharing of clock-gating logic across many registers
- No reduction in the switching in the capacitance of the clock gate to the register

NPTEL Ajit Pal IIT Kharagpur NPTEL

Coming to another very important type of clock gating that is Cell-level Clock Gating. I have already told about, a two types of clock gating **that is your you know** global clock gating ,that is, block level clock gating and then ,we have discussed register level clock gating now, it is cell level clock gating. What is that key feature of this cell level clock gating? The key feature is that in this case, the cell designer introduces cell level clock gating. So, earlier you know, idea either the designer is explicitly incorporating clock gating hardware, while doing RTL coding or automatically clock gating circuits are introduced.

So, it has some impact on the flow of the design; that means, you have to perform design placement routing. I mean RTL level design, gate level design, placement and routing. So it wills, it has impact on the flow, but whenever; you are doing cell level design. Then, the design is performed by the cell designer. So, for example, a register bank is designed such that, it receives clock when new data needs to be loaded. So, register bank is designed in such a way, the functionality is built in such that, it will receive clock; it will the clock will go there only, when new data has to be loaded. Similarly, a memory circuit memory block will receive clock only when that memory part is accessed; that

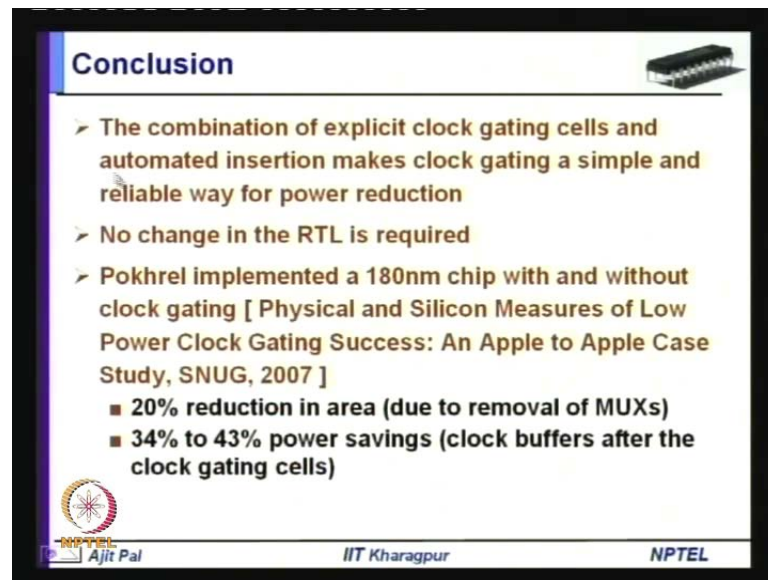
means, memory bank can be clocked only during active access cycles. So that means, within the memory bank the clock gating circuit is integrated.

So, it is simple without any design flows to as I have already told, it has no impact on the design flow, neither the designer has to bother about, it nor automatic tool will need to introduce clock gating hardware, because it is already built in as part of the hardware, but may not be sufficient in terms of area. And power for example, whenever you are doing at register level, each register has to be designed with built in clock gating hardware.

So, there is the question of sharing a single clock gating hardware, for large number of registers. So, that sharing will not take place in this case, as a consequence there is area overhead and all registers are to be designed with clock gating. And it does not allow sharing of clock gating. As I have mentioned clock gating, logic across many registers and no reduction in switching in the capacitance of the clock gate, to registers though no reduction in switching; that means, earlier we have seen clock gate to the register you know. So, this clock gate to the register there is some path. In this path there is reduction in switching active in this particular case.

But whenever; it is built in within this register itself then, the clock will through offers the entire path with high activity, and within this register only gating is done. So, this part of the circuit will be having higher activity, and wherever you are doing this cell level clock gating.

(Refer Slide Time: 45:34)



Conclusion

- The combination of explicit clock gating cells and automated insertion makes clock gating a simple and reliable way for power reduction
- No change in the RTL is required
- Pokhrel implemented a 180nm chip with and without clock gating [Physical and Silicon Measures of Low Power Clock Gating Success: An Apple to Apple Case Study, SNUG, 2007]
 - 20% reduction in area (due to removal of MUXs)
 - 34% to 43% power savings (clock buffers after the clock gating cells)

NPTEL Ajit Pal IIT Kharagpur NPTEL

So, these are the various types of cell clock gating circuits that we have discussed now, coming to conclusion the combination of explicit clock gating cells and automated insertion makes clock gating. A simple and reliable way for power reduction; that means, now a day.

What is being done the designer can incorporate explicit clock gating. When it is done at module level and automated insertion can be done by the tool making clock gating. A simple and reliable way for power reduction earlier when tools, and techniques were not available in I mean using clock gating was a very difficult task, because I have already mentioned about, various challenges and as a consequence, it was not very reliable, but with the availability of suitable tool, which we will do necessary analysis later on I shall also discuss about, the cad tool available for this kind of thing. And you will see that now, it has become a very reliable way for power reduction and there is no change in RTL; that means, RTL coding need not be changed, because you are not really changing any functionality. Since, you are not changing any functionality that, RTL coding part remains unaffected.

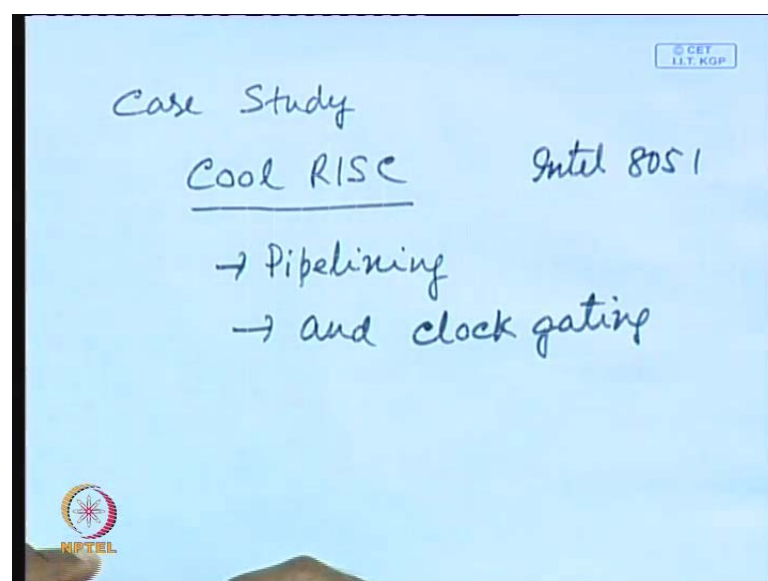
Some case studies have been done; one very important case study has been done; by ochre and his group they implemented an 180 nanometer chip with, and without clock dating. So, say one for the functionality of the chip is not mentioned, but by using 180 nanometer technology they, implemented two chips one without clock gating and 1 with

clock gating and they compared there, performance that is, why they have given a very interesting name physical, and silicon measures of low power clock gating success and apple to apple case study so; that means, they have compared the I mean, the circuit with clock gating and without clock gating and that was reported in the conference SNUG.

In the year 2007 and you can see here, at 20 percent reduction in area was achieved whenever clock gating is used; this was primarily due to the removal amount of multiplexers. We have seen if you do not use clock gating hardware, the registers are feeding inputs from a are circulating multiplexers. So, each register will be having a multiplexer, but whenever it is being replaced by a clock gating hardware then, a single clock gating circuitry will be placed many multiplexers. So, this area saving has is primarily due to this removal of these multiplexers.

Then 34 percent to 43 percent of power savings have been achieved primarily, because clock buffers are after the clock gating cells; that means, normally you will find. I have already shown, let me go back to this diagram, you find that after the clock gating cells you have get lot of power buffers, clock buffers. So, as a consequence these buffers will be having lot of I mean much less activity. So, the power dissipation at the output of these buffers will be much less and as a consequence there, was significant reduction in the power dissipation 34 percent to 43 percent power reduction.

(Refer Slide Time: 49:18)



In fact, another important I mean case study was done the chip the name of the chip is cool rise. So, a one simple RISC micro controller was implemented incorporating pipelining along with clock gating.

So, by using pipeline and clock gating together, this full cool rise chip was implemented. So, far as functionality was concerned it is similar, **very similar** to the popular, you know that Intel 8051 you may be familiar with the processor Intel 8051, which is a eight bit micro controller. Which is very popular and functionality, wise it is very close, but it has been implemented in cooperating pipelining and clock gating and it has been found that, the power dissipation of this particular micro controller is very small and it is now, available as a ip, which can be used in many imbedded applications.

So, with this let us, conclude today is lecture, I have introduced a very important technique known as clock gating, and it is widely used and as we have seen 30 percent to 40 percent, dynamic power reduction can be done by using this and in the next lecture. We shall continue our discussion on other techniques of switch capacitance minimization. Thank you.