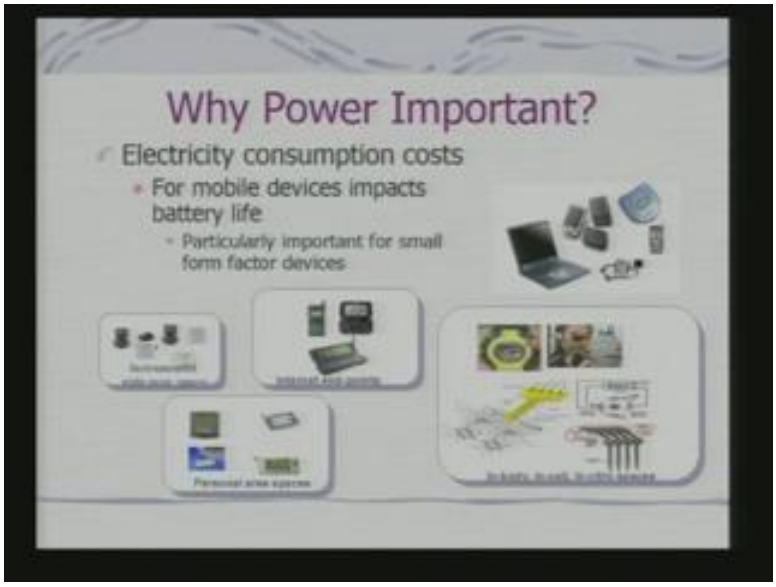
Embedded Systems
Dr. Santanu Chaudhury
Department of Electrical Engineering
Indian Institute of Technology, Delhi

Lecture - 18
Power Aware Architecture

Today, we shall discuss about the power consumption issues in an Embedded Systems. What are the components which had involved really in power consumption? What are the parameters, which can be dealt with to manage the power consumption is will be our topic of discussion in to this class.

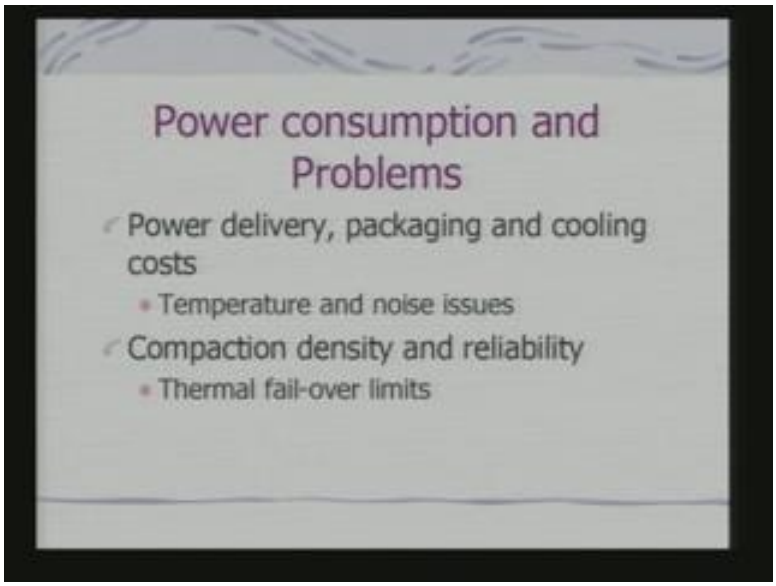
(Refer Slide Time: 01:26)



So, why power is important; obviously, the fundamental point is electricity consumption costs. And if you are talking about an battery powered embedded system. Then, that effects battery life are would like to have longer battery life. And obviously, that is a reason why I would like to optimize my power consumption.

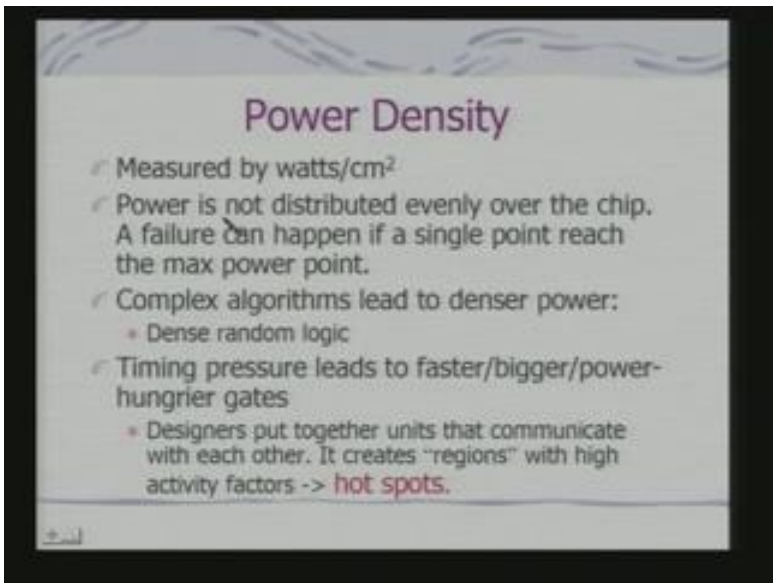
But, at the same time there are lots of other issues which are related to power consumption. Because, there are possibilities of failures, if the power distribution is not properly managed on the silicon, in which the hardware is implemented.

(Refer Slide Time: 02:19)



So, what are the power consumption and problem. So in fact, the power delivery packaging and the cooling costs and as a temperature increases, the effect of noise also becomes more dominate. So, it will be always desirable to minimize power consumption. These has implication in terms of compaction density and reliability. Because, there can be sparks which may be heated on the chip itself and that may lead to failures.

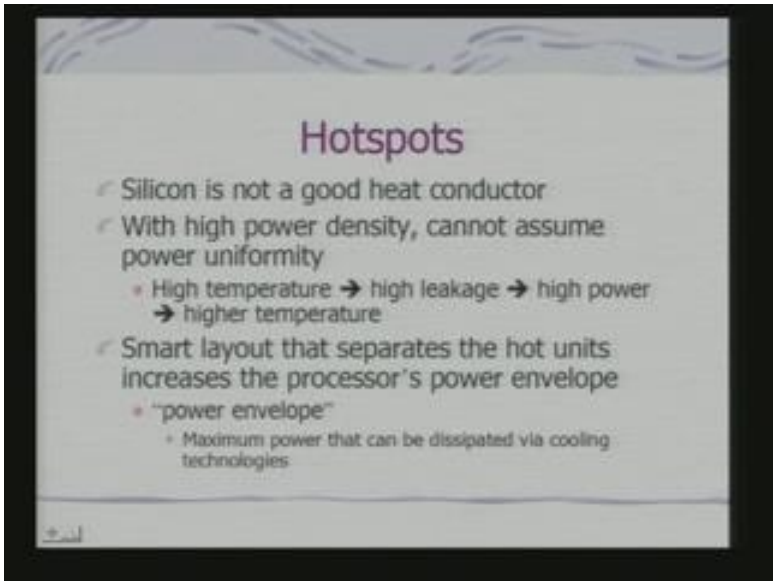
(Refer Slide Time: 02:54)



So, a particular point of consideration here is that of power density. That is how much heat is getting dissipated over the chip area. And this is typically measured in the unit

words per centimeter square. Power is not in general distributed evenly over the chip. So, a failure can happen, if really there is a spark which is highly heated up.

(Refer Slide Time: 03:28)



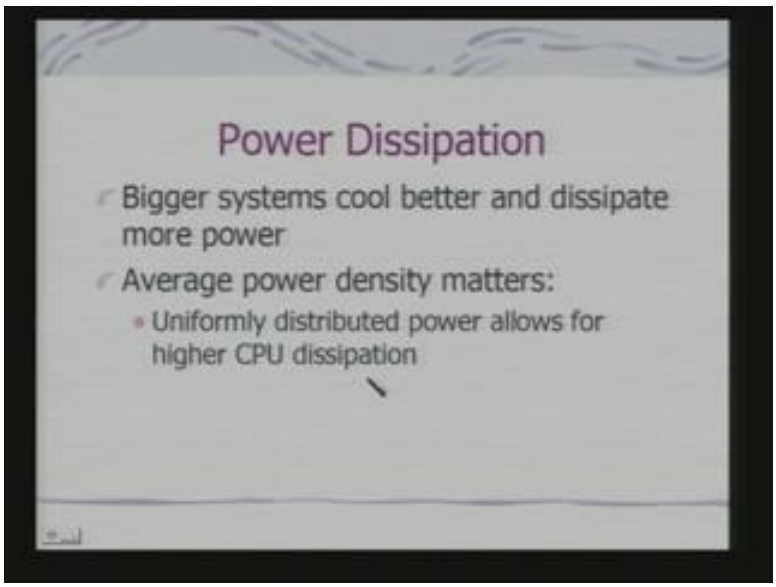
In fact, silicon is not a good heat conductor with high power density, we also cannot assume in a sense power uniformity. So, if you have high temperature that may lead to high leakage of the current and high power and that is leering to high temperature. So, it is becomes a kind of a look. Now, the hot units are those units in the silicon area, which actually or widely used which are being densely used.

And they are the power consumption would be more and there also there may be more dissipation. And the temperature at those points may actually increase and those actually a hotspots. So, one issue in terms of design in terms of real VLSI design, that you would like to look for a layout that separates the hot units, typically the hot spots.

And that can increase what is call processors, power envelope. What is power envelope? The maximum power that can be dissipated via cooling technologies if that is more distributed. So, the power envelope can actually increase in fact, this hotspot why this hotspot arises? Because, if you look at from another consideration, you would like those units, which are likely to take to each other frequently to be place close together for the purpose of efficiency.

The movement you place them close together. Obviously, what happens since they are being used intensively. The energy consumption over that area may be more. That is the temperature over those areas may can actually increase. So, a basic objective of a layout in a power aware layout would be to have a smart layout. So, that the hot units are distributed evenly over the chip area.

(Refer Slide Time: 05:42)

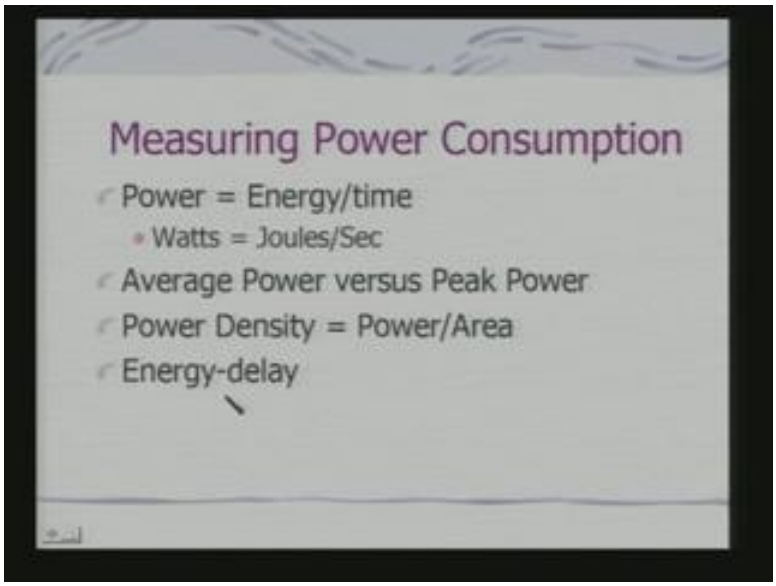


Now, bigger systems will cool better and dissipate more power. But, the moment I am trying to make the systems smaller, which is an objective in terms of our embedded systems or embedded processors that we are looking at. We would like may be us video camera to fit into a palm itself.

So, in that case I am packing in more circuitry on to a single SOC. Then; obviously, the cooling possibilities gets reduced. So, we have to optimize on the circuits as well as other parameters related to the circuits. So, that this power dissipation is much less and average power density is what is more significant.

So; obviously, uniformly distributed power allows for higher CPU dissipation. And that is again I pointed out earlier also, that is a motivation of the layout. So, this power dissipation is what is related to therefore, reliability of the system as well as that of battery life if it is operated via battery.

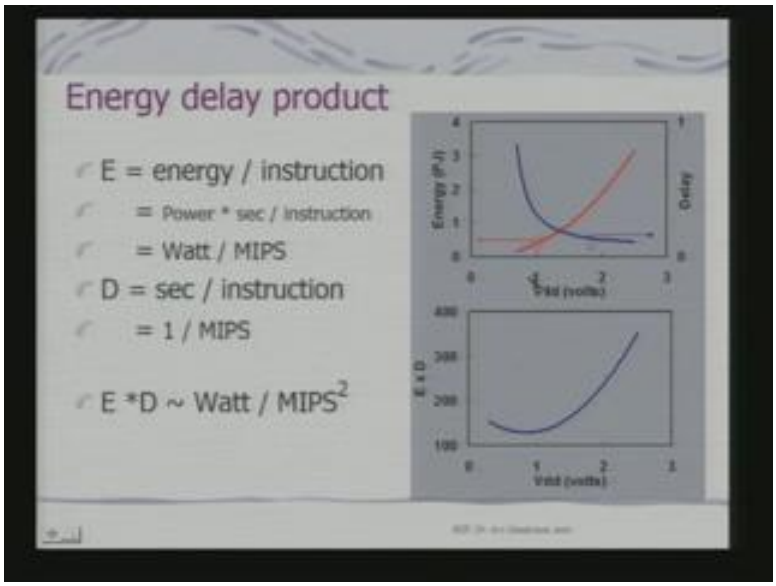
(Refer Slide Time: 07:02)



So, how do you measure this power consumption. The simplest thing is power all of us know, that energy by time if this is the dimensionality of the unit. So, this is how the simply the power is measured. The other parameters, this what is average power versus peak power, depending on the activity. The peak power may be much more, than that of the average power.

Then, you have got we have already looked at power density, which particularly is a measure to understand what would be the reliability in terms of the heated points on the chip area. So, that is power density ideally it should be uniformly distributed. The other factor is what is known as energy delay.

(Refer Slide Time: 07:49)

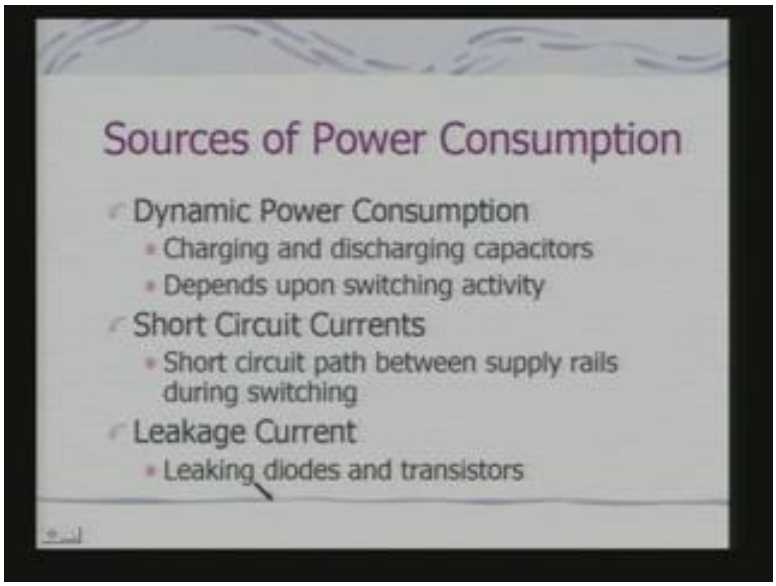


So, how is energy delay calculated? So, energy if you define energy E as energy per instruction. So, this should be Watt per MIPS, because I am having this is a rate at which the instructions a getting executed. And the delay is a time per instruction. So, if I take it, it would be 1 by MIPS. So, energy delay product the unit would be Watt per MIPS square.

Now, these are the typical curves, what you find here is... That, if I look at the energy VDD or VCC is a supply voltage. Now, if the supply voltage increases; obviously, your energy increases, in terms of you ((Refer Time: 08:43)) and if you look and the delay interestingly delay reduces. In fact, this is a phenomena and these we shall look at in the context of CMOS, what exactly the relations that has been suggested.

And because of these phenomena would like to look at it has a one quantity, what is called energy delay product. Now, why this energy delay product is important? Because, it tells you that if you are really reducing VDD. And if that, reduces energy consumption. Then you may need to pay in terms of efficiency, the time taken for execution of the core. So, if you want to look at these straight of together using one measurable unit, then you will use energy delay product.

(Refer Slide Time: 09:33)

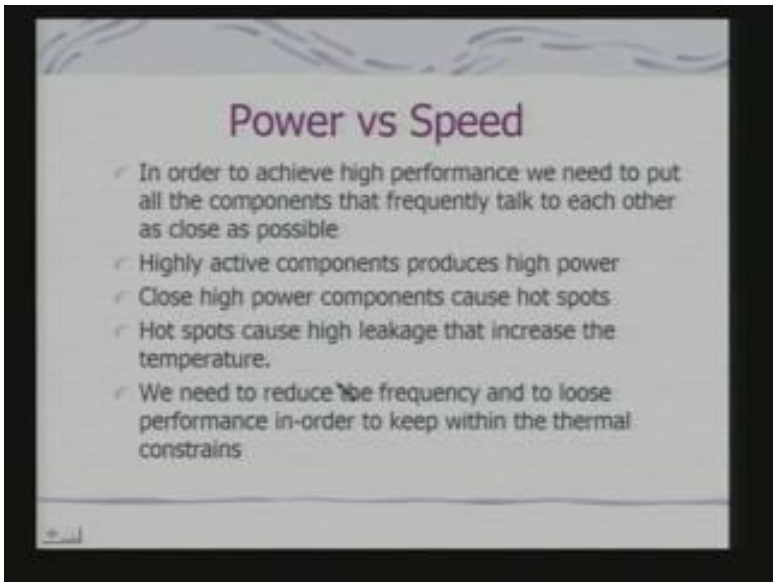


What are the sources of power consumption? See, dynamic power consumption is; obviously, charging and discharging of capacitors. Because, any bit transfer with regard to your mass and CMOS circuitry finally, boil down to charging and discharging of capacitors. And this is depends upon, what is call the basic switching activity.

In fact, this also related to the power consumption in memory. Because, you know the DRAMs are nothing but, capacitances. These also related to your interconnect power consumption. Because, on the line transmission line also there will be the capacitance is which needs to be charged up. Second resource is shot circuit currents, because short circuit path between supply rails during switching.

When really the transistor switches, then there is a shot circuit current between the supply rails. And there that would be peak current flowing and that would be also as so as power conjunction. The last point is leakage current, that is leaking diodes and transistors. That is even when it is not switching, there will be a current leakage through this diodes and transistors. So, in a circuit these are the basic components and the sources for consumption of power.

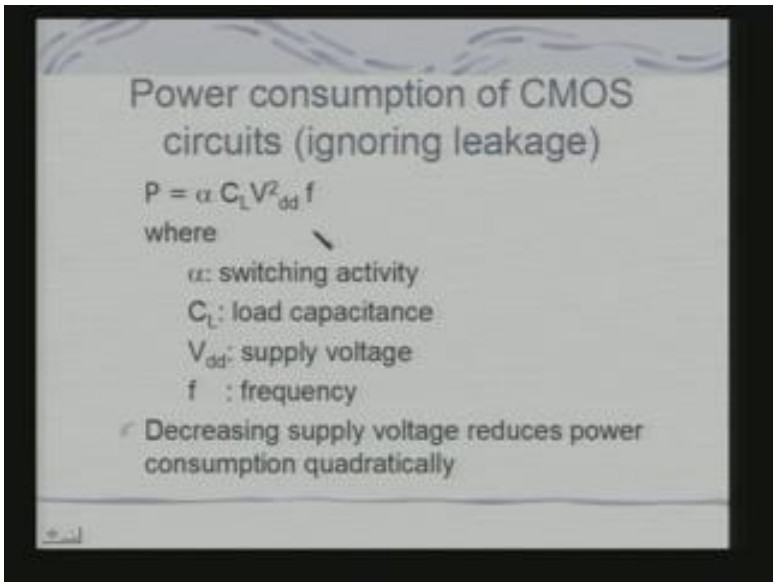
(Refer Slide Time: 10:58)



So, there is; obviously, a tread of in terms of Power vs. Speed. So, what we say in order to achieve high performance, we need to put all the components at frequently talk to each other as close as possible. Highly active components produces high power, close high power components would cause hot spots.

Hot spots cause high leakage that, increases again the temperature. So, we need to reduce may be the frequency to lose performance in order to keep within the thermal constrains. So, this is the basic relationship, basic conceptual relationship between components which are determining your efficiency as well as power consumption.

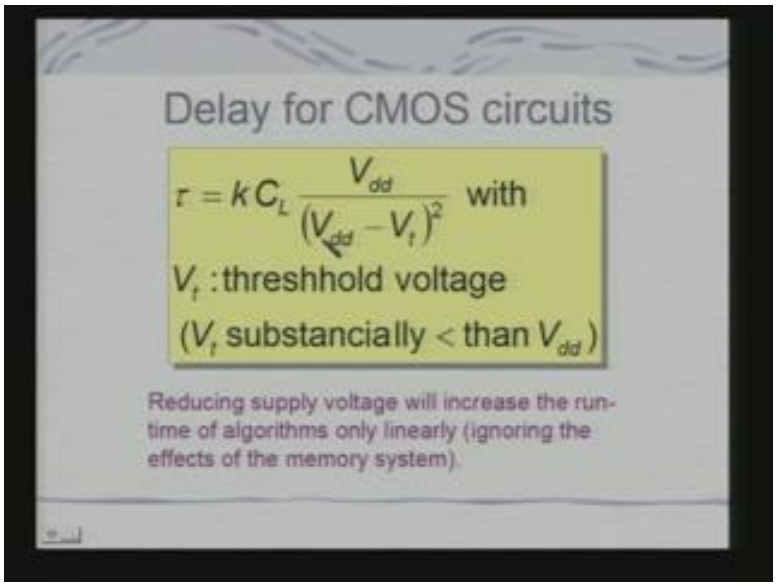
(Refer Slide Time: 11:46)



How the power consumption is calculated in a CMOS circuit. This is the expression alpha is a measure of switching activity, C L is the load capacitance because, primarily the switching is what consumes power. And what we are ignoring we are ignoring completely the leakage component of the power consumption.

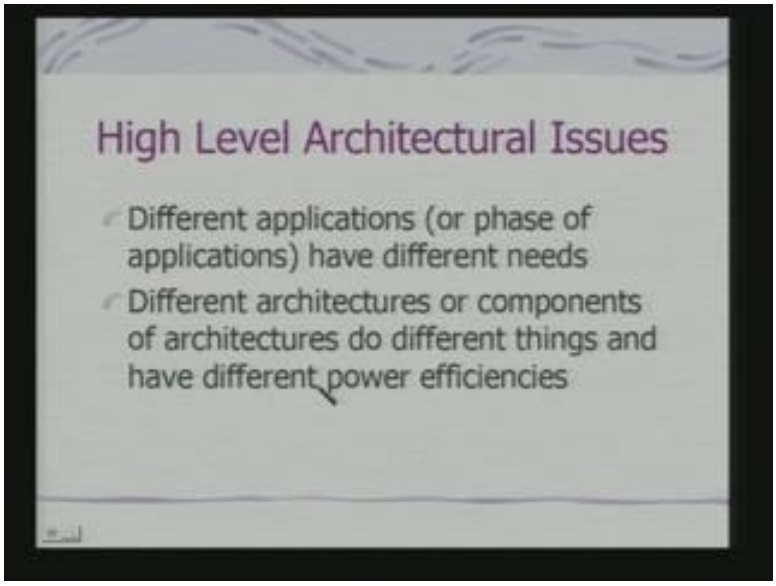
VDD is a supply voltage and F is a frequency. This is the relationship of power consumption in CMOS circuits. The, what is interesting to note is that decreasing supply voltage, reduces power consumption quadratically because, the relationship is squared. And you can understand why that becomes an issue for designing efficient and power aware architectures.

(Refer Slide Time: 12:44)



At the same time, if you look into delay for CMOS circuits. That is a tow, tow is given by these parameters, where V t is threshold voltage. And V t is substantially less than V d d. Now obviously, what it shows you is that, reducing supply voltage will increase the runtime of algorithms only linearly. Because, your tow is linearly dependent on your actually V d d really. And if it is very small it becomes inverse relationship. Because, this is V d d effectively it is becoming V d d square.

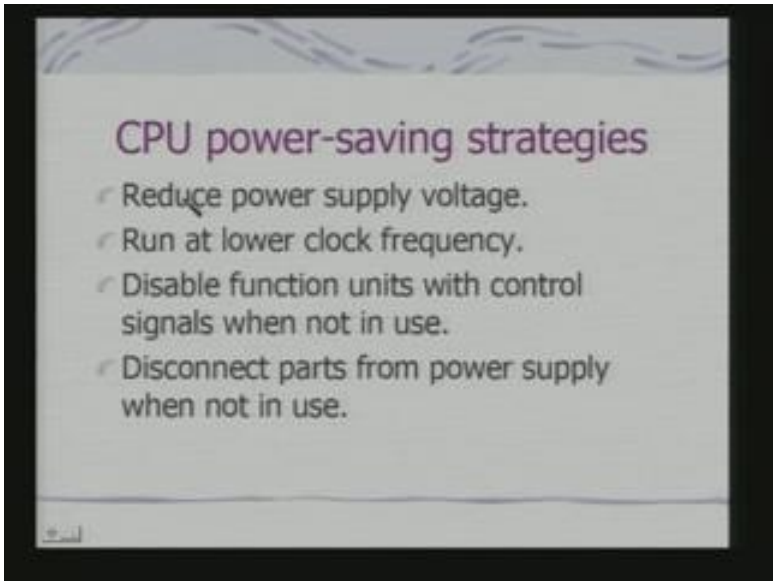
(Refer Slide Time: 13:24)



Now, high level architectural issues with these background. Now, we need to look at high level architectural issues. Because, different applications are face of applications can have different needs. And different architecture components of architectures do different things and have different power efficiencies.

Now obviously, we need to explode these fact to make my power consumption optimal. And the other point that we have seen, that relationship of power consumption with voltage and efficiency with voltage. So, depending on computation needs can we play with voltage and the frequency. So, all these issues therefore, have to be looked at in an integrated fashion for the purpose of having power efficient architecture for Embedded Systems. So, how what the CPU power saving strategies are implemented.

(Refer Slide Time: 14:26)

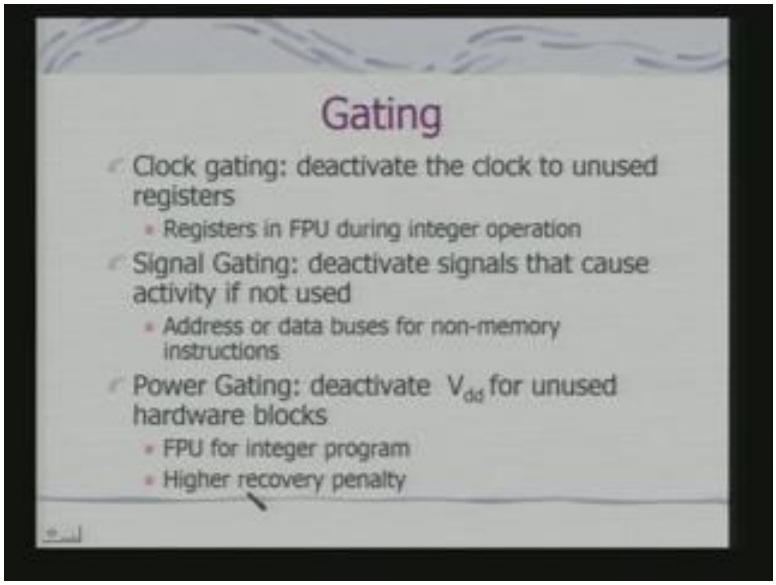


The first thing is, you reduce power supply voltage. The movement are reduce the power supply voltage power would go down I have already seen from the CMOS relationship. Run at lower clock frequency, that is also another way of reducing the power. The relationship is already established that, the other thing is disable function units with control signals when not in use.

Because, the high level architecture issues that not all units will be used for a computation. And disconnect parts from power supply when not in use. In fact, disable is one aspect and disconnect is completely putting into sleep parts of the architecture. So,

these becomes more from the high level issues. And these more from the circuit level perspectives.

(Refer Slide Time: 15:18)

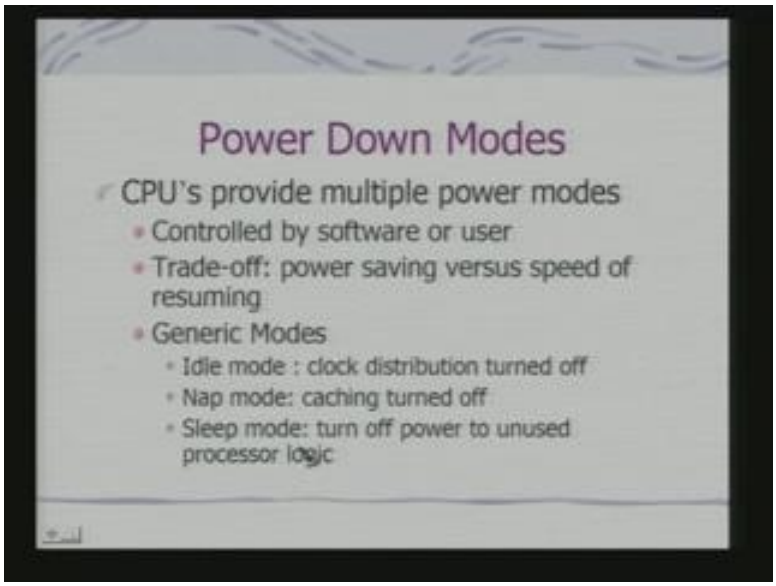


Therefore, what we have a mechanisms called gating for controlling power. First is clock gating, deactivate the clock to unused registers. If the registers are not being used deactivate the clock. So, the power consumption disappears. So, if I have a floating point unit, then registers in floating point unit can be deactivated, when you are doing really integer operations.

Next thing is signal gating, deactivate signals that cause activity if not used. So, address or data buses for non memory instructions. So, when it is not being used I can deactivate an effectively I can save ((Refer Time: 16:04)). Then, power gating is actually deactivating of V d d for unused hardware blocks. So, FPU for integer program and in these case what I have done, this clock gating I had stop the clock.

But, the VDD supply is still on. So, there would be leakage current, if I stop the V d d to those registers, then leakage current is also minimized. But, obviously, when I am doing a power gating there could be higher recovery penalty. What does that mean? More time required to recover to active state.

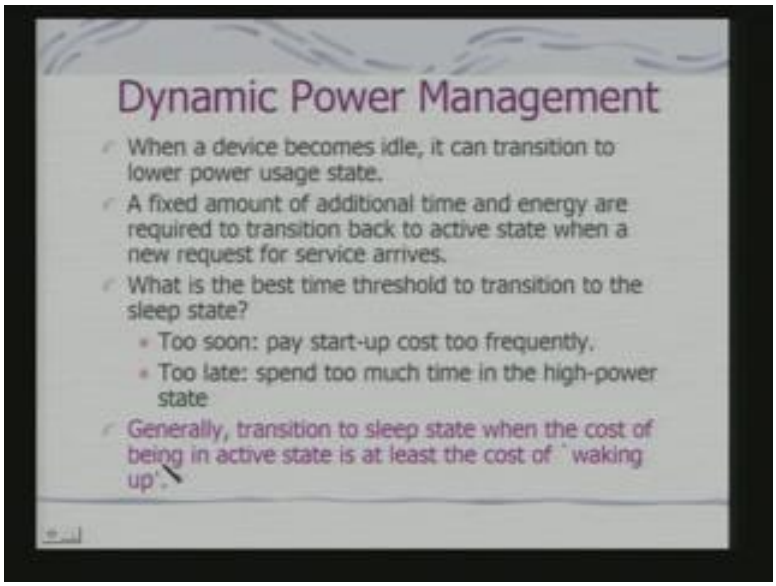
(Refer Slide Time: 16:50)



So, CPUs by making use or a combination of these gating modes provide multiple power modes. And these modes a typically controlled by software or user. The tradeoff is power saving versus speed of resuming. Because, you can save the power, but you have to pay for in terms of the time taken to resume active work.

The generic modes that, you find in many of these CPUs are what are called idle mode nap mode, sleep mode. Idle mode, whether clock distribution is turned off. So, this is basically a clock gating nap mode typically causing turned off, your cache memory is put to off. And typically this is relation to your on-chip cache mostly, sleep mode is turn off to unused processor logic. So, it is more of a power gate. In causing turned off if I have to do the causing turned it off I have to do signal gate.

(Refer Slide Time: 18:01)



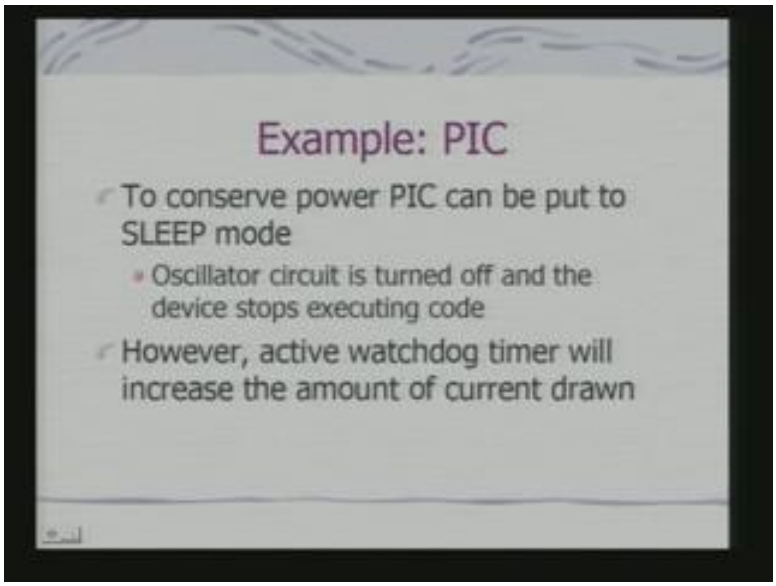
So, dynamic power management, because this power modes have to be controlled by the software the user. So, what is dynamic power management therefore, implies. When a device becomes idle, it can transition to lower power usage state. A fixed amount of additional time and energy or required to transition back to active state, when a new request for service arrives.

What is the base time threshold to transition to sleep state? Because, this is the key question because, you can put the processor to different modes or you can put different hardware components of the processor or in SOC to different modes. And that is the decision of the controlling software. That what point of time on the basis of what objectives this decision be taken.

So, it says what is the best time threshold to transition to sleep state too soon, you pay start-up cost too frequently, you may save more power. But, you save, you are not saving in terms of time you are paying in terms of time too late. May be spend too much time in high power state when you are not really using it.

So, what we say it is a kind of a thumb rule, the generally transition to sleep state, when cost of being in active state it is at least the cost of waking up. So, that balances out the two things. If it is not show, then there is no justification for doing a mode switch. So, your controlling software has to make these estimates before taking these decisions. So, let us take some example of processor, where these modes are available.

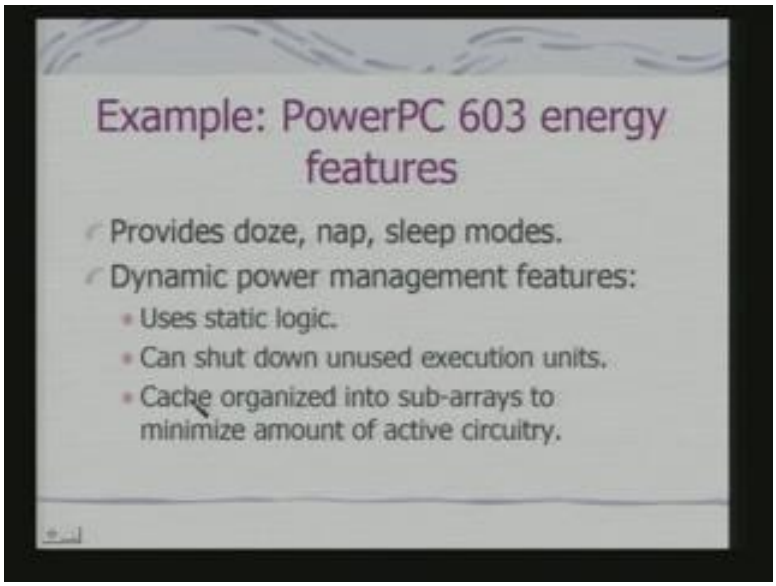
(Refer Slide Time: 19:47)



First example is PIC we have studied PIC. So, PIC has got to conserve power a sleep mode. In sleep mode oscillator circuit is turned off and the device actually stops executing code, you cannot really execute code on the device. However, active watchdog timer will increase the amount of current drawn. If the watch dog timer is on and it is monitoring, it will continue to draw current.

And in fact, you can have timers which may be driven by external clocks, which can be used for operations even when the processor is in sleep mode. So in fact, this is a very simple architecture, where you have identified only sleep mode to conserve power for the processor.

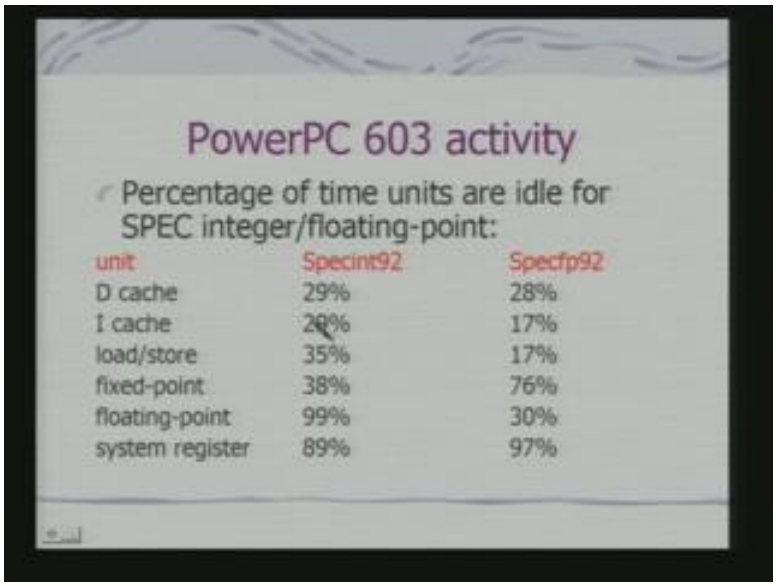
(Refer Slide Time: 20:39)



Power PC you have not studied really. But, that is another example of a processor, which is targeted for embedded applications. Why we are looking at it? Because, it has got a number of these modes, doze, nap, sleep. These are different kinds of modes with different functionalities. And what are the features, it uses static logic and not really dynamic logic.

Can shutdown unused execution units, that provision is provided explicitly. And the cache is organized into sub arrays to minimize amount of active circuitry, when a cache access taking place. In fact, these point we shall discuss when we look at the memory architectures for power optimization.

(Refer Slide Time: 21:25)

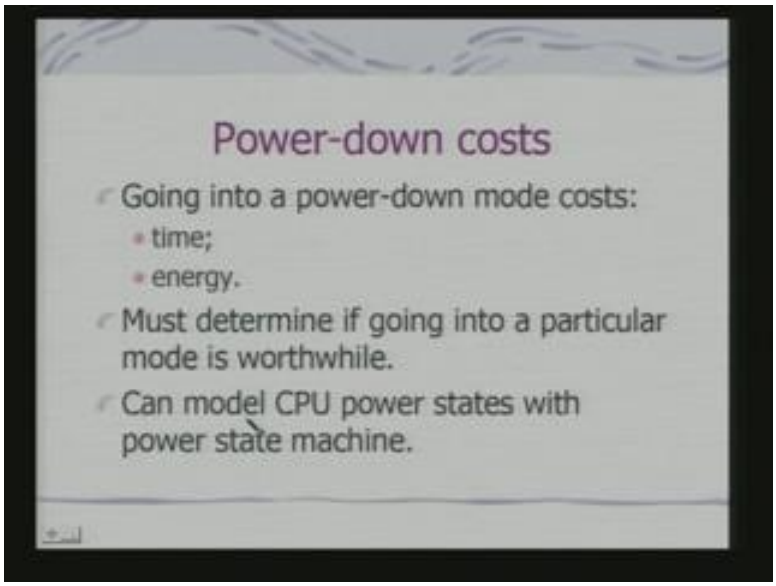


The interesting thing is, these statistics for power PC. This is we are looking in the context of power PC. But, this is true in general across other processes as well. This it says the percentage of time units are idle for this SPEC integer of floating point. This is a some standards, a standards for evaluating the performance of the processor, it is a bench mark programs.

So, this spec int is for integer computation and spec f p for floating point computations. Now, you see that how the percentage of time these units are ((Refer Time: 22:07)). It is the most clearing example is this one, when you are doing actually a integer computation 99 percent of time your floating point ALU and registers will beyond used. And if you put the power completely off, if you do the V d d of to this part, then you leakage current is definitely minimized.

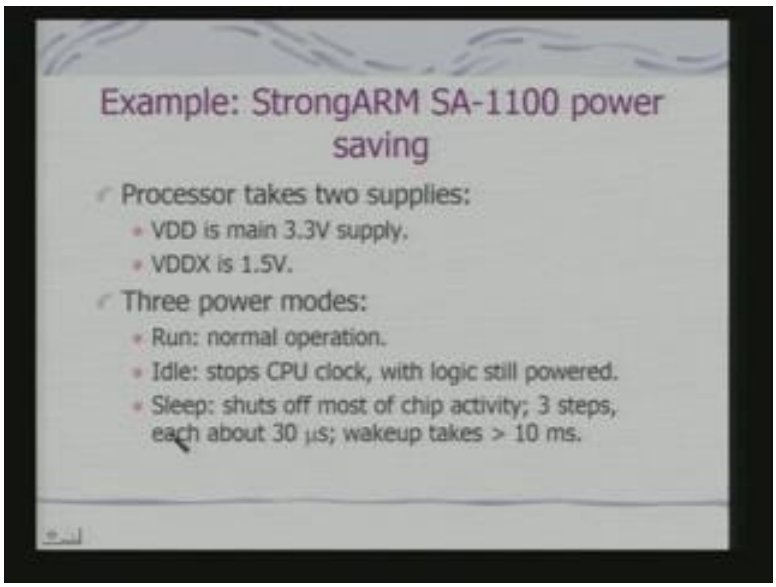
Now, you have also you will see that these cache, data cache and this is instruction cache, which we have already seen in the context of Hardware architectures. So, they are also not always used. So, about 30 percent of the time, in case of these kind of bench mark is unused. So, here also the issue comes up how you can effectively organize this cache so, that you can minimize this kind of energy consumption at this point.

(Refer Slide Time: 23:09)



Next issue comes up is, obviously, I had already talked about the power down costs going in to a power down mode costs. Obviously, time as well as energy and I told you; obviously, that it must determine if going into a particular mode is worthwhile. And what is interesting is that to study this phenomena, we can model CPU power states, with what is called a power state machine.

(Refer Slide Time: 23:40)

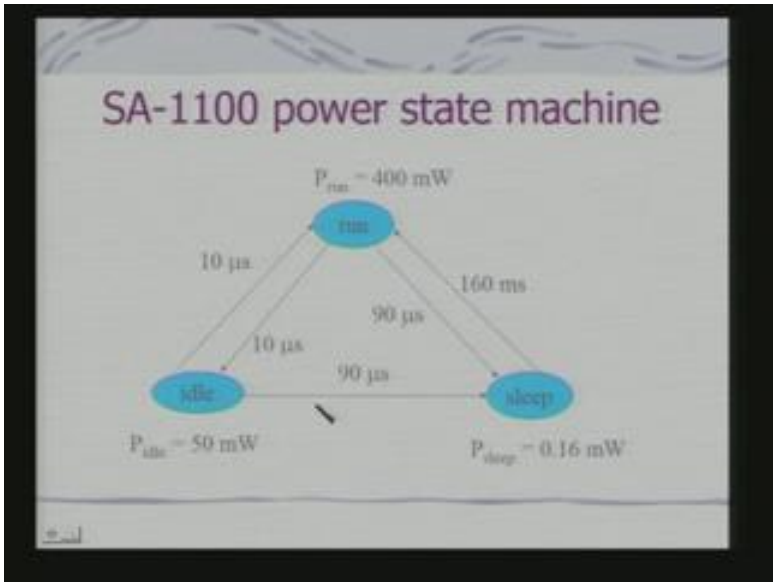


So, we are looking at now and ARM variant call strong ARM. And with respect to the strong ARM we shall have look at the power state machine. This processor takes two

supplies VDD is a 3.3 volt supply and VDDX is 1.5 volts. It has got three power modes run, which is normal operation. Idle here it is stops CPU clock with logic still powered on this is clock gating.

They knew of got sleep, shuts of most of chip activity takes 3 steps, each about 30 microsecond and wakeup states greater than 10 millisecond. So, these are the three modes for these strong ARM. Now, these are the three states, if I talk about power state machine of the processor you also have these as states.

(Refer Slide Time: 24:39)



So, this is the power state machine of this strong ARM, now this is run. So, when you run in the normal conditions the typical consumption is about 400 milli volt. And in idle state it is about 50 milli volt and in sleep mode it is 0.16 milli volt. Therefore, if you can intelligently switch between the states, what you really get you can really gain in terms of power consumption.

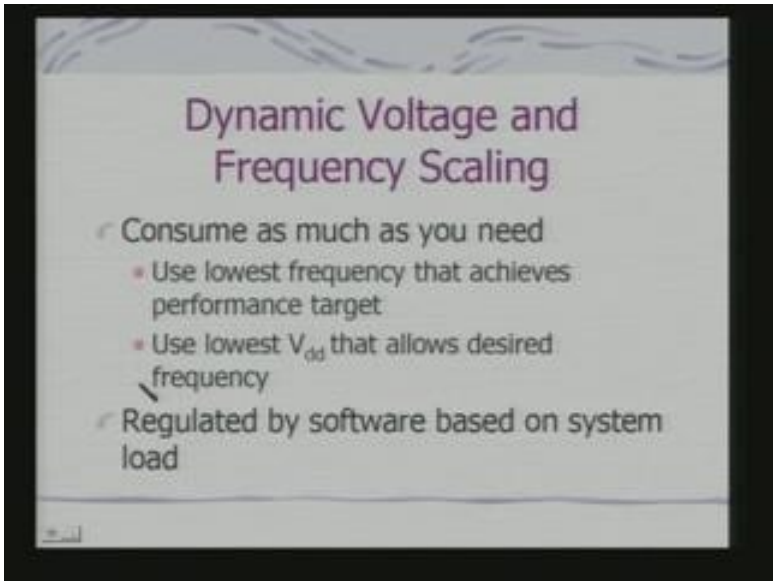
So, what is the other thing that is being shown here. Typically, it is a time that takes for transitions. The transitions both ways from run to idle or idle to run, it is of the order of 10 microsecond. From idle to sleep and run to sleep it is order of 90 microsecond. And from sleep you can only go back to the run with consumption of about 160 milliseconds.

So, these delay substantial. So, whether you would put a processor to sleep or not, it has to be a stage precession. That, if there is really you do not expect much activity in near

feature, then only you should put the processor to the sleep. These power state machine we have seen with in relation to the strong ARM. But, these kind of state machines can be drawn for any processor, which supports multiple power models.

In fact, these state machine is actually to be used by the controlling software, which in many cases would be the operating system to decide about the different states. In which the processor should operate based on may be the load as well as the usage pattern. And there are other strategies also by which even without involving OS. This power switch in of different state switching into different states power states can be done.

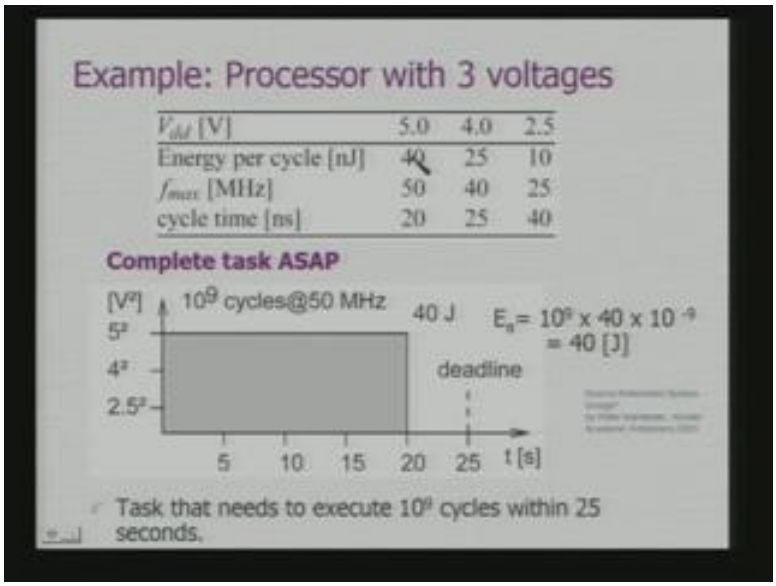
(Refer Slide Time: 26:57)



Next we shall look at this concept of what is called dynamic voltage and frequency scaling. The basic concept is consume as much as you need so, use lowest frequency that achieves the performance. What does that mean? If I am using the lowest clock frequency; obviously, the time taken for execution of an instruction increases.

But, if you really do not want a performance better than a particular time period or a threshold, you would not like to use higher frequency. Because, higher frequency can mean higher power consumption. Also the same thing is related to the lowest V d d that allows desired frequency. And this is can be also regulated by software based on system load. So, this is known as dynamic voltage and frequency scaling.

(Refer Slide Time: 27:55)



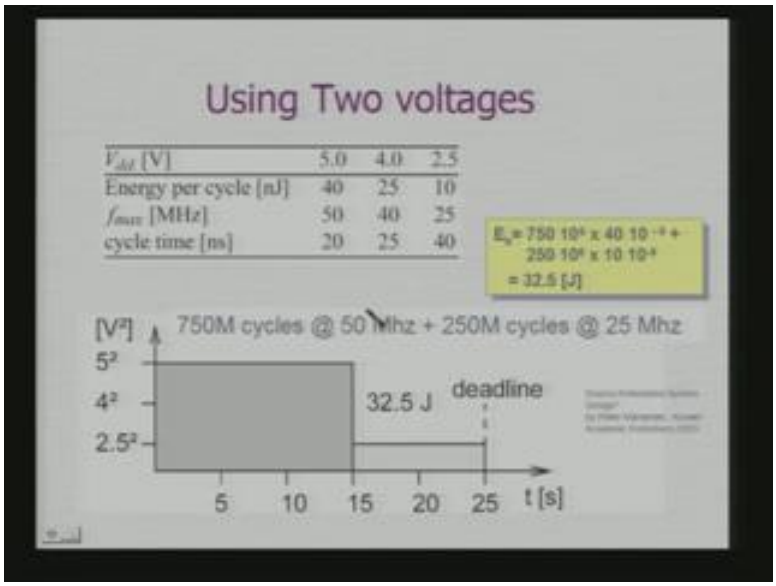
We shall look at example to understand this phenomena. Let us consider we are talking about a hypothetical processor, which supports three voltage levels 5, 4 and 2.5. And the energy consumption part cycle, which is being given by in nano joule is 40, 25 and 10 pay correspond to the different power supplies. And the corresponding frequency at which it operates are also different for different supply voltages.

You operate at 50 megahertz, when it is 5 volts. If the supply voltage is reduced, then you operated a lower frequency. And as you reduce the cycle time; obviously, increases. Let consider a task which may have started at this point, which is 0. And has got a deadline of say 25 seconds. This is the deadline for the task. Now, if this is the deadline of the task, what as at mean? The processor should operate in a way, such that the task is finished before you cross the deadline.

Now, this is one policy at the philosophy could be complete task as soon as possible. So, if I am using that policy of completing the task, as soon as possible. I shall be operating at highest voltage and highest frequency. So, it takes say this much amount of time to complete the task. So, it says the task takes 10 to the power 9 cycles at 50 Megahertz.

So, if it takes at 50 Megahertz, what is the energy consumption it is 40 joule. And this is calculated by this formula. So, I am finding that I can finish of the task in 20 seconds iIf I am really using the processor at the highest voltage and highest frequency.

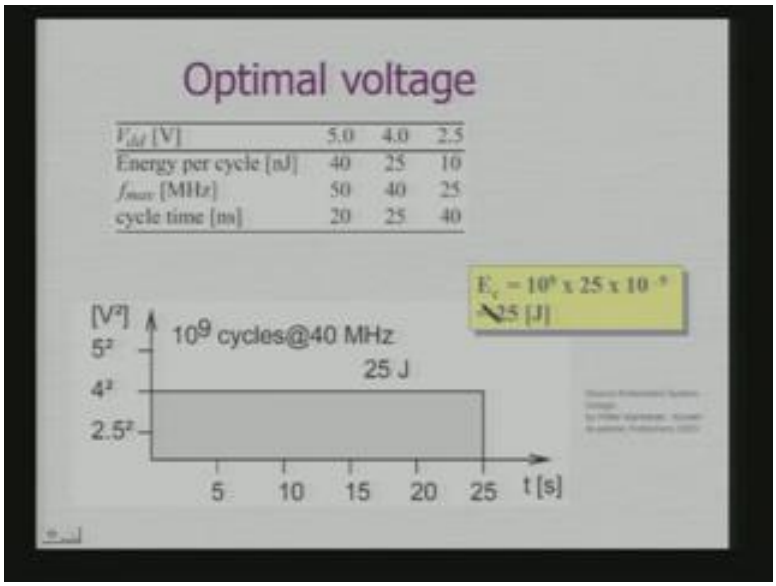
(Refer Slide Time: 30:13)



Let us say that I would like to do this job using two voltages or not three, then what happens. So, the whole thing was that whole operation requires what 750 plus 250 cycles I can add it up. Because, the whole thing I had operated earlier at 50 Megahertz now I am operating this as at 50 Megahertz and at 25 Megahertz. 25 MHz means, I can now have a supply voltage, which is what less.

And correspondingly I see that at the supply voltage my energy per cycle is also much less not 40. So, if I add this to up I get 32.5 joule, which is less than definitely 40 joule. So, what I am doing I am dynamically switching the voltage, as well as a frequency.

(Refer Slide Time: 31:17)

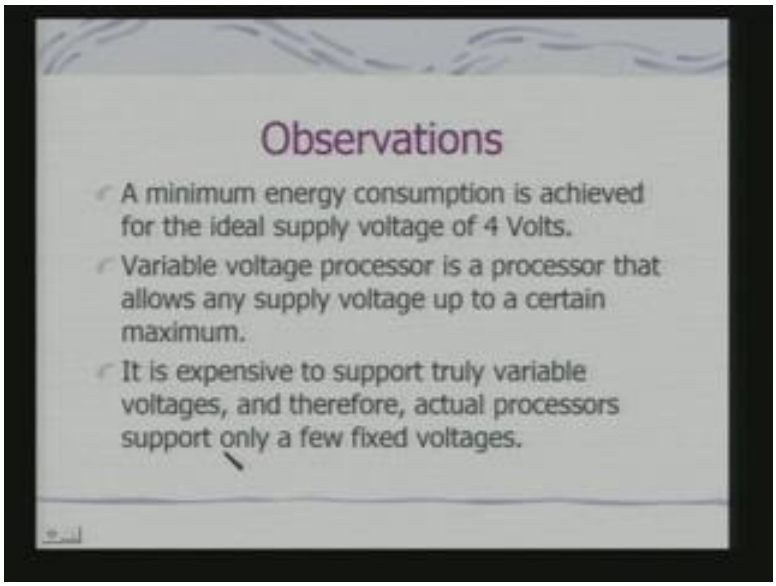


And then, if you say that this is optimal voltage. So, what I am knowing is? I am doing the entire operation, since I have got a deadline. If I have the deadline to operated that. So, I would like to operate at this 4 volts, if I operate completely at 4 volts, then I can do the job in 25 seconds and consuming effectively just 25 joule.

So, this becomes optimal. So, what it turns out to be if you look into it, that to the OS these scheduling becomes an optimization problem. In fact, if you think in terms of a standard optimization techniques, say integer programming. All these things are applicable here for finding out an optimal schedule of voltage and frequency for doing a task.

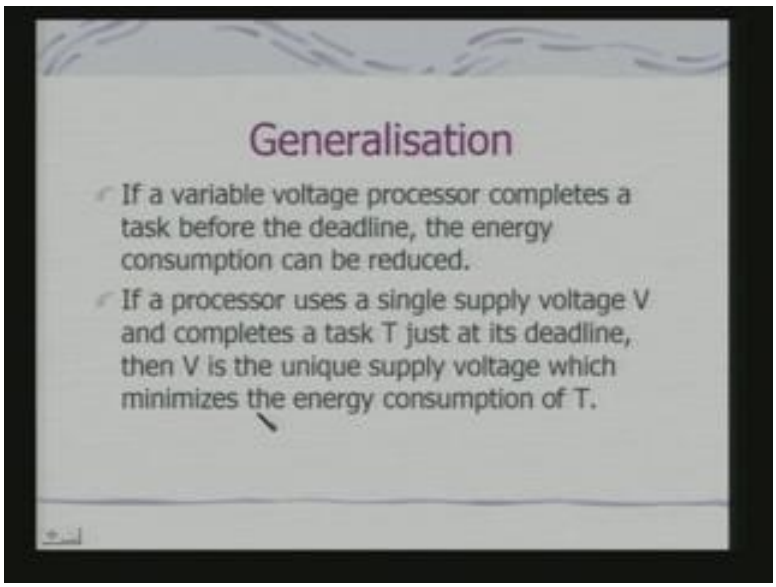
So, what is important therefore, to note is that this energy management, you need to have support in the hardware. But, also your software has to be tune to exploide this supports to optimally use energy.

(Refer Slide Time: 32:37)



So, observations therefore, are the minimum energy consumptions is achieve for the idle supply voltage of 4 volts. A variable voltage processor is a processor, that allows any supply voltage up to a certain maximum. If you can change it contain noisily. And it is expensive to support truly variable voltages. Therefore, actual processor support only a few fixed voltages and you have to switch between this fixed voltages.

(Refer Slide Time: 33:15)



So, some generalization can be drawn from these is that, if a variable well voltage processor completes a task before the deadline. The energy consumption can be

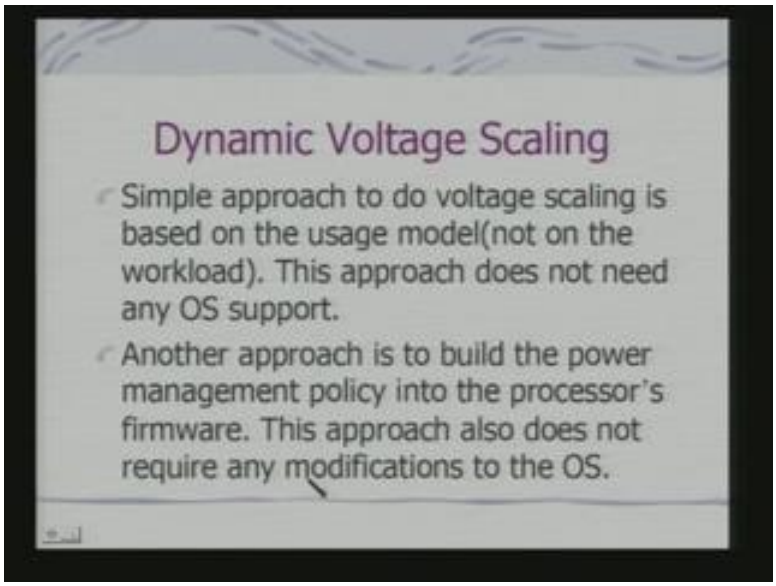
definitely reduced. So, you are not optimally using the supply voltage and frequency to minimize the energy consumption.

The other thing is that you processor uses single supply voltage V and completes a task T just at it is deadline, which was the previous case. Then, V is the unique supply voltage which minimizes, the energy consumption of T because, you cannot have anything other than that. So in fact, there has been obvious queries with regard to this parameters.

So, what we have showing here is that, the two aspects one is we talked about the CPU having different modes and consuming different amounts of power. CPUs can also provide for different voltages to operate. And corresponding to voltages it would have different frequencies to operate.

And it is the job of the operating system or the control software to look at the deadline of the task. And then, accordingly choose appropriate voltage and the frequency so, that energy consumption is minimum.

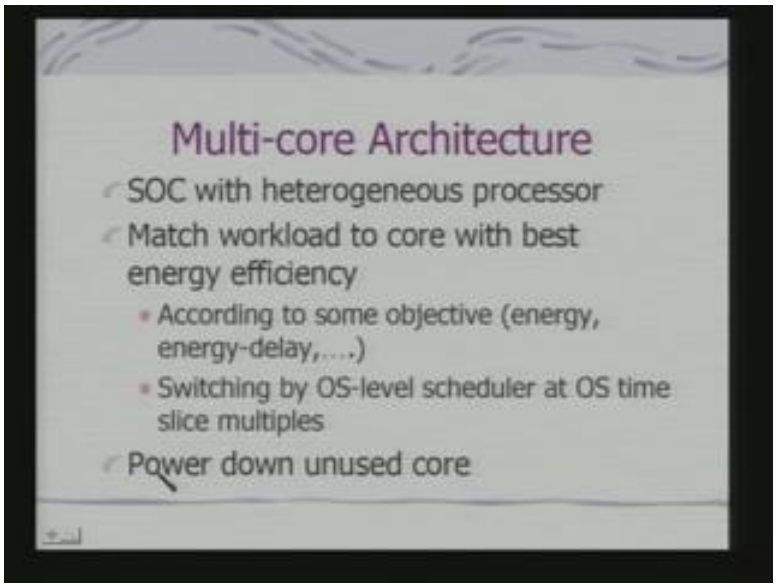
(Refer Slide Time: 34:38)



So, how to do this voltage scaling, the simple approach to do voltage scaling is based on usage model. And not may not be the current workload and these approach does not need any OS support. This is the simplest way of doing it, another approach is to build the power management policy into the processor's firmware itself.

These approach also does not require any modification to the OS. So, depending on a kind of usages that is taking place, you switch to different voltages as part of the firmware specification.

(Refer Slide Time: 35:21)

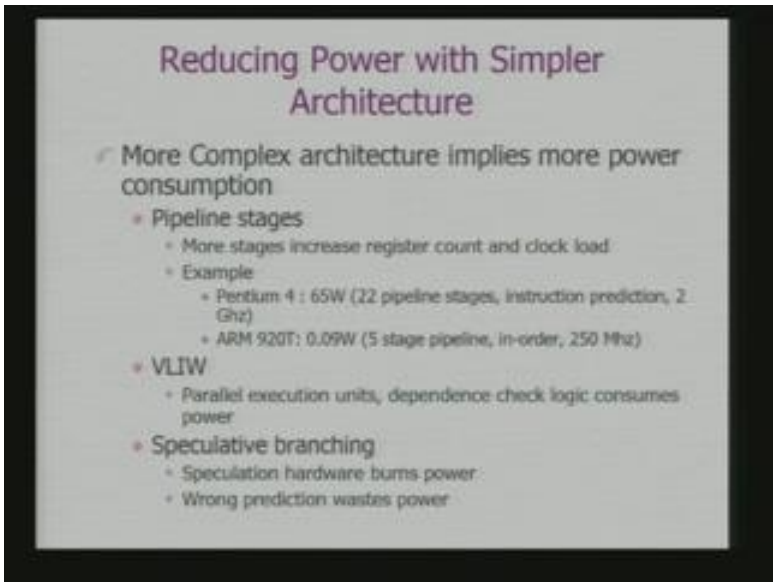


The other thing is that today in your SOCs, we have already found there are multi core architectures. And that is SOCs having heterogeneous processors. And these processors are targeted for doing different kinds of work. So, one of the basic issue involve is match workload to the best processor. And best processor from which point of you, from the point of you of energy consumption.

So, it can be according to some objective of energy consumption or even or the energy delay product, which looks at both the parameters together. And this switching here is done by the OS level scheduler at OS time slice multiples. Because, in such a case you expect an operating system to run and we the operating system can switch from one processor to another processor.

And obvious solution and the most cross solution is, if you are not using something. So, power down unused core. So, if you are consider a O map processor and if you a just using basically your OS routines. So, you should power down the C55 core and use only the ARM. Another important issue is that, you if you have a simpler architecture you have less power consumption. Obviously, more complex architecture means more logic, more switching and more power consumption.

(Refer Slide Time: 37:03)



So, more complex architecture implies more power consumption, just look at this pipeline stages. So, more stages increase register count and the clock load. Since, the register count and the clock load increases, the power consumption is bound to increase. In fact, Pentium 4 uses 22 pipeline stages. And if you compare ARM9 which you have studied uses 5 stage pipeline.

And this figures that we are seeing here one is 65 watts and another is .09 watt. This is not only because of the pipeline, because of the other factors as well I shall come to it. But, pipeline is one of the major factors which reduces the power consumption. So, what it is pointing to, pointing to is that the processors or the processor cores which are targeted for the embedded applications, will be architecturally must simpler compare to that of your processors targeted for general purpose, main frame computer.

Because, they are you have the option to use more sophisticated cooling technology dissipate the power and have higher power envelopes. Another example is VLIW, you have got parallel execution units and depends logic consumes power. So, this VLIW although this is used in the context of DSPs.

But, there also this power consumption becomes an issue. But, here the advantage is that if you can do the other point is very straight forward. What is the advantage, advantage is if you can execute multiple instruction in parallel, then the total time taken will be less.

So, total energy consumption can be less although you may consume more power per instruction.

Next thing is speculative branching, today most of this Pentium as others they use speculative branching and hardware. So, what is really speculative branching. That means, you try to speculate what would be the next instruction to be executed, you predict an accordingly take actions. So, this speculation that is prediction hardware burns power.

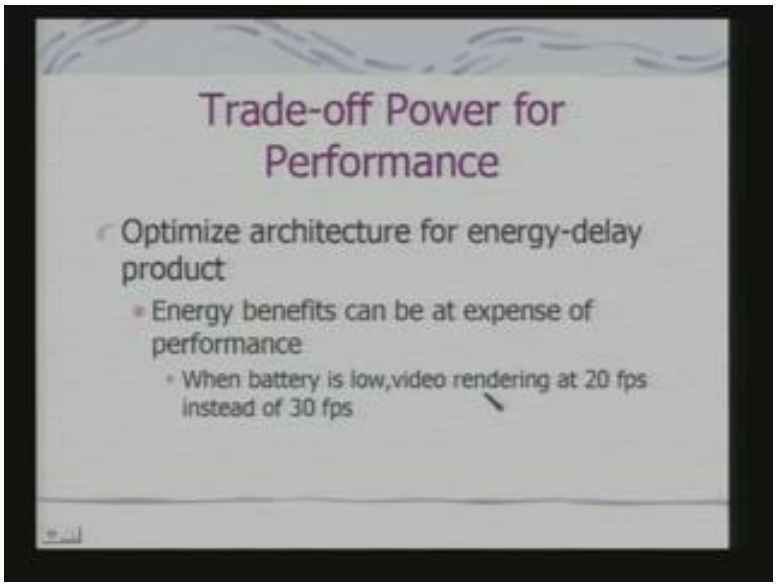
And if there are long prediction, you actually waste power because of wrong prediction. In fact, these hardware is predict complex hardware. In fact, if you go back to the Pentium example, what we find is it uses instruction prediction. It predicts which instruction to be executed in future. An accordingly on that basis is try to brief age and keep the pipeline full.

That prediction process consume substantial power. And also it operates at a much higher frequency and that is why the power consumption if the order of 65 watts. You find an ARM in that case does not use prediction, although it uses pipelining. But, it does not use instruction prediction, it is a in order execution.

So, that prediction logic is not there at all. In fact, you won not find prediction logic in any of these processors, which are actually targeted for embedded systems. And one of the basic reasons for these is this power consumptions. Also the other reason is use of additional chip area. Now, interesting feature is if you go back and remember that ARM had what conditional instructions.

And conditional instructions, where use to make your pipeline flashing overhead much less. And that also eliminates the need for kind of predictive execution to make the optimal utilization of the pipeline. The other thing is we have already looked at voltage scaling, one point is for the purpose of meeting the deadline.

(Refer Slide Time: 41:06)



The other thing is that, you can also optimize a architecture for energy delay product. You have already seen that, that the energy benefits can be at the expense of performance. All you what we said, that when we are talking about the OS, if you have the deadline, why should schedule voltage and frequencies such that your meeting the deadlines.

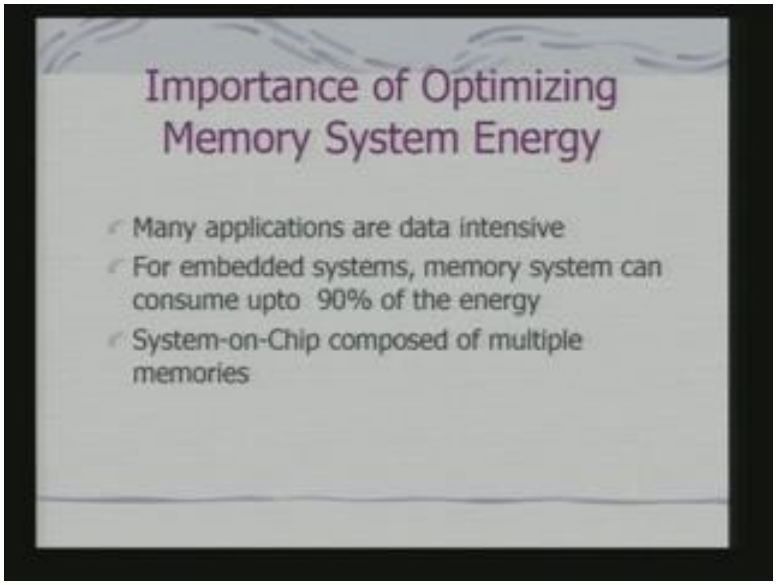
But, many of the deadlines for many of these embedded systems a soft deadlines. Let us say a CD player or say DVD player, they soft deadlines. An if their operating on the battery in if the battery, if I have the battery condition, monitoring circuitry in the embedded system. Then, what can be done? The video rendering instead of a 20 frames per second I can reduce it would 30 frames per second.

So, really what will happen is your battery life is increased. But, you are really a not suffering in terms of performance, when it is a soft deadline. Try to understand the basic issue, the when it is a deadline unit to meet. Then, you; obviously, would try to schedule the voltages and the frequencies, such that you meet the deadline and still have power efficiency.

But, when it is a case of a soft deadline and you really do not have battery power. How can you along it the lifetime of the battery. So, you down perform the operation, that is another strategy which is also used for dealing with this kind of problems. Now, we shall

look at memory and cache issues related to the power. So, far we had looked at primarily the processor.

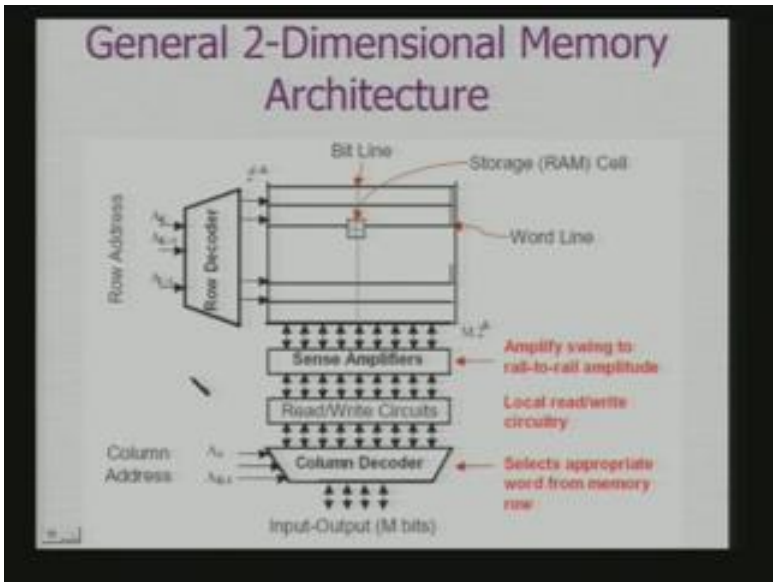
(Refer Slide Time: 42:54)



So, what is the importance of optimizing memory system energy. Many applications are data intensive. That means, your data is in the memory and need to access memory for writing, reading the data. And for embedded systems some estimate says that, the memory system can consume up to 90 percent of the energy and system on chip are compose of multiple memories we have already seen.

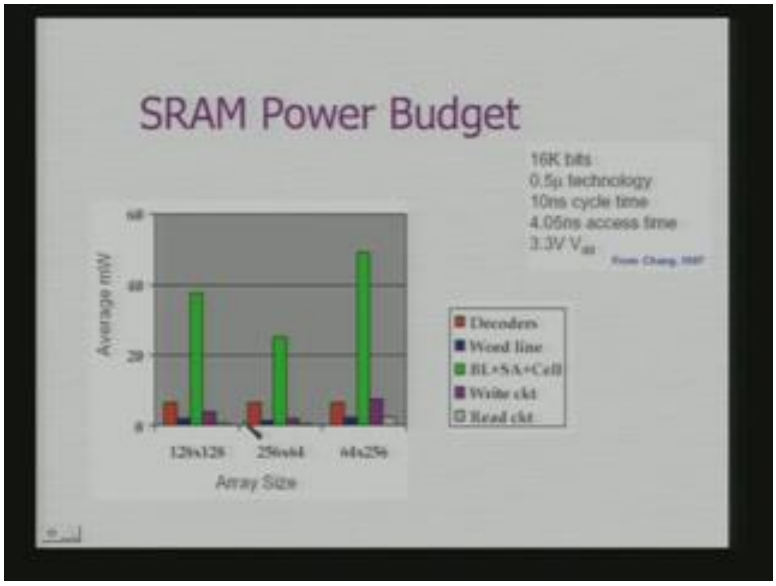
So, these consumption in the memory is also another source of power train, it is not only the CPU. So, far what we have discussed is primarily strategies related to the CPU but, memories also another source of power consumption.

(Refer Slide Time: 43:36)



So, if you look into this, this a general to dimensional memory architecture. So, you have these bit lines, this a storage RAM self. So, you have a row decoder, you decode the road, you sends then you may you have a read write circuits. Then, you have a column address and then, you get input, output that selects appropriate watt from memory row. This a basic architecture why I am showing this basic architecture is with relation to this architecture you will fine how the power consumption is done.

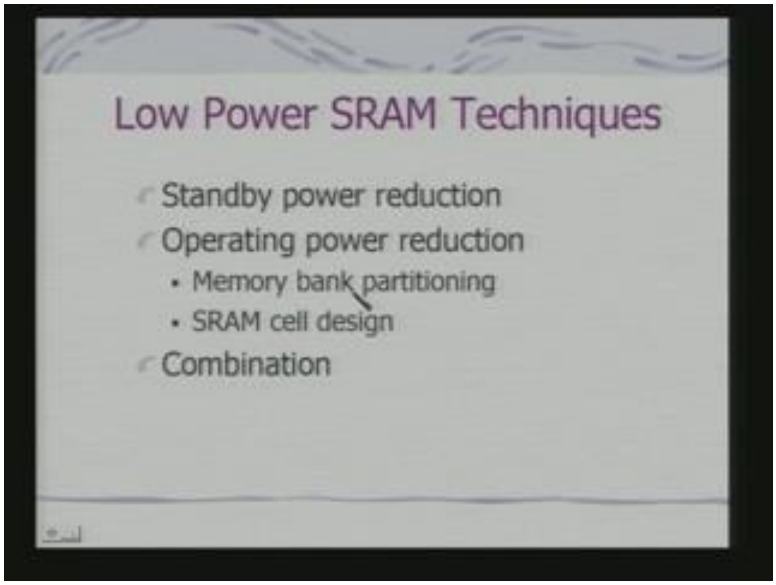
(Refer Slide Time: 44:04)



So, this is called SRAM this the how the power is used. Now, what has been done is here it shows the consumption with relation to different array sizes. And these different color indicate that power being consumed in different circuitry. So, it is in decoders, word line this is bit line, then sense simplifier and the cell actual cell itself, write circuitry, read circuitry.

So obviously, the maximum power consumption is with respect to this cell itself. And your write operation is also in many cases or consuming. But, your read is in a way much less.

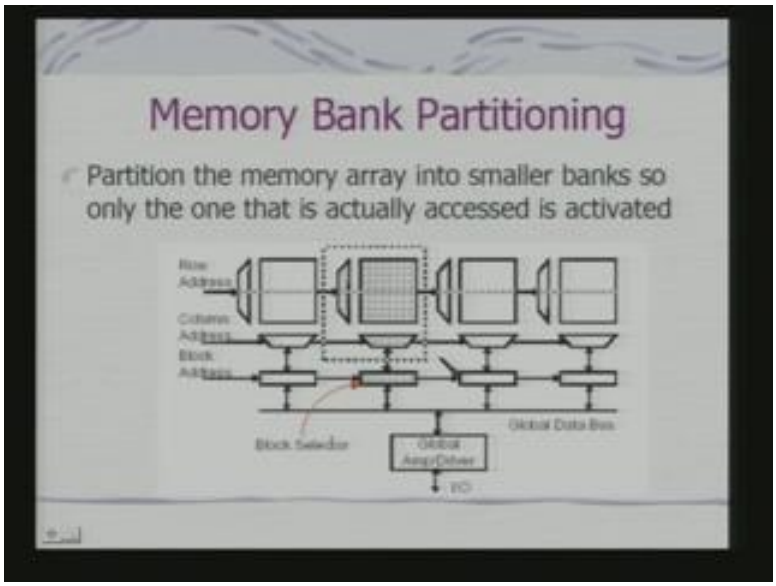
(Refer Slide Time: 44:56)



So, how do you lower this power consumption let us in static RAM. Because, there are different kinds of RAMs and with respect to that we have to talk about the strategies. The fundamental point is standby power reduction. And then, the operating power reduction. Operating power reduction is memory bank partitioning.

In fact, this point we had earlier discussed as well. And modifications in the design of the SRAM cell and may be combination of the two. So, in terms of the basic architectural parameter, this memory bank partitioning gives as an predict optimize strategy.

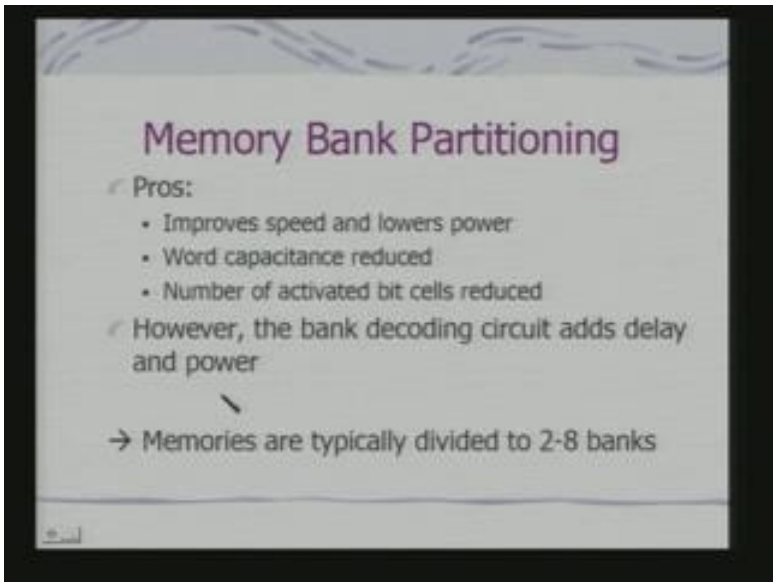
(Refer Slide Time: 45:39)



So, what it say is that, if you go back to the basic architecture. You partition the memory into smaller banks. So, that only the one that is actually accessed is activated. So, the consumption takes place only with respect to the bank. So, then the corresponding cell that is being made use of and the other circuitry here is leading to much less power consumption because, your lines are not driving all these cells at the same point in time.

So, charging and discharging of capacitors on the lines. And that really is eliminated the moment I go into the smaller banks. So, typically the SRAM is expected to be organized in terms of this kind of smaller banks to optimize power consumption.

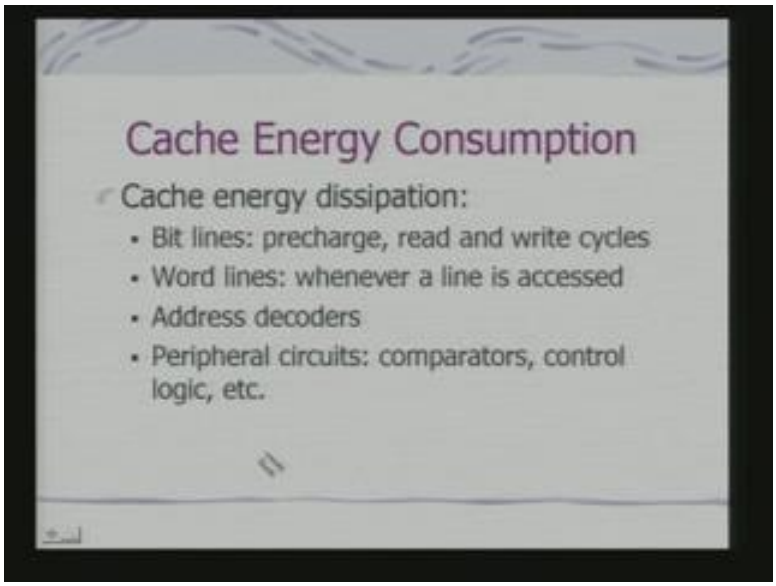
(Refer Slide Time: 46:42)



So, what are the advantages therefore, improve speed. Because, lower bank I can access it faster, word capacitance reduced. Because, you accessing less number words number of activated bit cells is reduced, all these leads to what less power consumption, power efficiency.

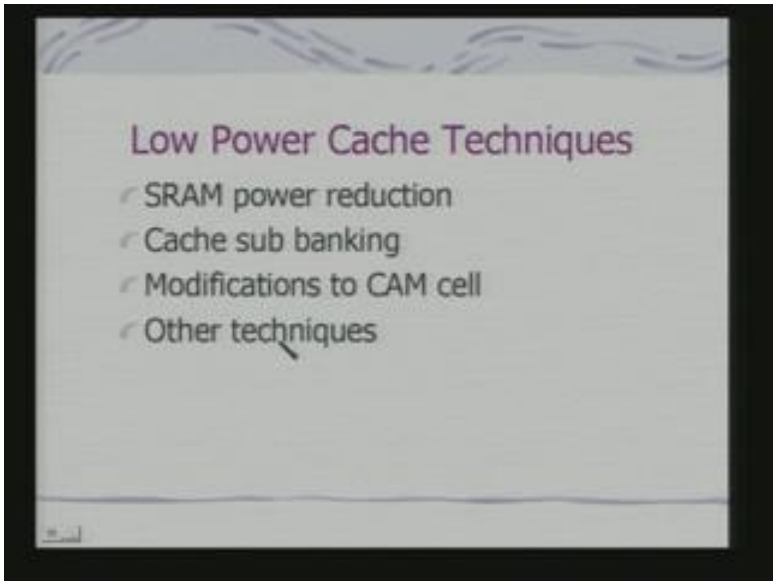
However, the bank decoding circuit adds delay and power. Because, I have got more decoding circuitry in place. But, these advantages are typically much more compare to that of disadvantages. And so you typical... So, you find that memories are divided to 2 to 8 banks in applications.

(Refer Slide Time: 47:22)



Next thing is cache, cache energy dissipation could be alone bit lines, they have to be precharge for read and write cycles, when they are being executed. Word lines whenever line is accessed, then you got to have an address decoders. Then, you got to have peripheral circuits which, actually do the comparators to know really, the data is there in the corresponding locations, using in tag information and the control logic. So, all these a source of power consumption in cache.

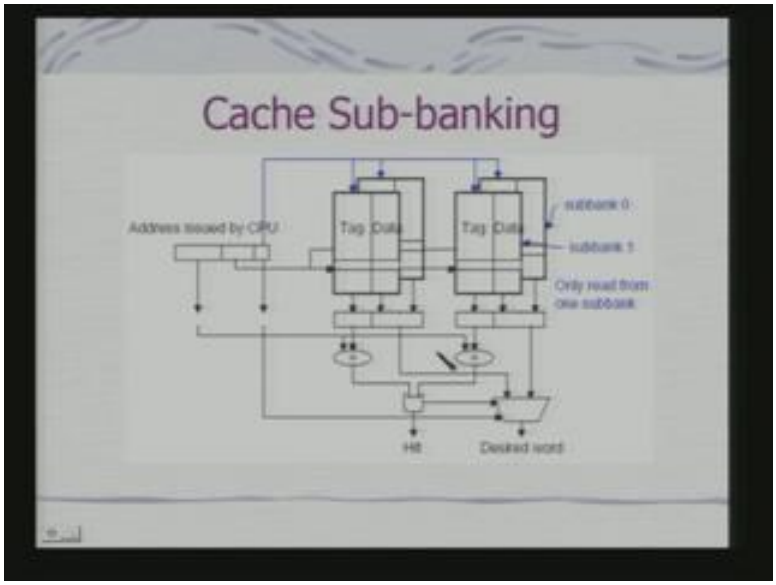
(Refer Slide Time: 47:55)



So, how do you go for low power cache techniques. Because, caches implemented primarily using SRAM. So, whatever SRAM techniques I have talked about, they in many cases will translate to here as well. And you can use cache sub banking modifications to CAM cell. So, CAM is content accessible memory, which is in many cases used to implement translation look aside buffers.

So, we can modified the design to CAM cell to reduce the energy. Because, that is frequently accessed. There are various other techniques, which we shall not discuss in this class. That is also used to optimize the power consumption on cache.

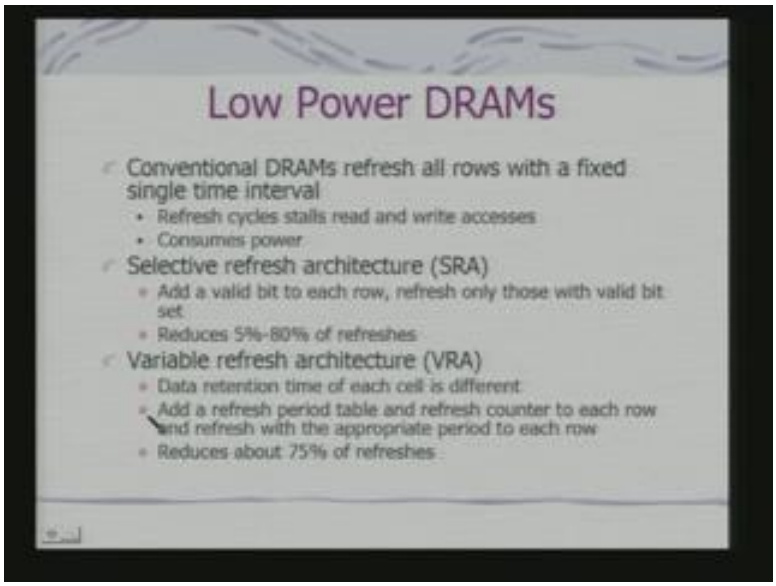
(Refer Slide Time: 48:39)



The cache sub banking is the simplest way to deal with the problem. In terms of the architectural, modification. So, what you have got, your cache is again now organized in terms of sub banks. So, this is your sub bank 1 and this is your sub bank 0. And at a given point in time you are doing only one rate from one sub bank and getting the data.

So, what your finding out the advantages that we had talked and discussed in the context of sub banking of SRAM's can be used in the context of cache as well. Then, what we have is DRAM's.

(Refer Slide Time: 49:27)



Because, a DRAM's are primarily used for implementation of main memory blocks so, conventional DRAM's refresh all rows with fixed single time interval. That is the typical DRAM refresh cycles. So, I say that given a DRAM, it has execute refresh cycles at regular intervals. Because, the data is stored in the capacitances, the charge is stored in capacitances.

So, the charge will de queue of... So, I need to refresh, so that I can retain the charge. So, typically refresh cycles stalls and read and write accesses. And each refresh cycle would consume power. Now, you cannot really do anything with read and write accesses. Read and write accesses will be required for any kind of program execution and particularly data intensive a program execution.

But, the question is can I do any something with the refresh cycles by which I can minimize the power consumption. So, there has been some suggestions and some implementations to take care of this fact. So, one is what is called selective refresh architecture. Add a valid bit to each row and refresh only those with valid bit set.

So, what does that valid bit means? Valid bit means that bit really has a data, the other things it may have a do not care. So, if it is a do not care, then it is no point refreshing those rows. Classically when the refresh takes place, this point is not taken into account. You refresh all the rows, when the refresh cycle really comes up. So, in these case you refresh only those rows for which the valid bit is set.

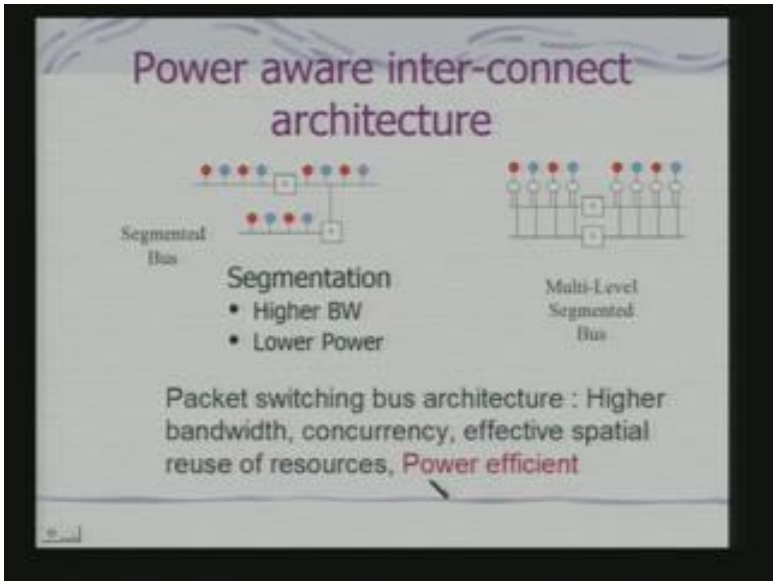
In fact, it has been found it can be reduced between 50 percent to 80 percent refreshes, it depends on how intensively the memories actually being used. The other thing is variable refresh architecture. So, data retention time of each cell in these case is can be different. Because, in terms of the technological being used, the data retention time for each cell is different.

Add a refresh period table and refresh counted to each row. And refresh with appropriate period to each row. So that means, your refreshing the rows, but each row gets refreshed with different intervals. Because, the data retention time of each cell becomes different. And the other issue is that why it can also become different it depends on at what point of time you have actually return the data on to that row.

Because, you refresh cycles are coming at a regular interval. So, if you have just return the data onto that row, you would might not need refresh at that point in time. So, what you are talking about is a refresh cycles with respect to each row. And it reduces about 75 percent of refreshes.

Obviously, this increases the hardware on-chip hardware. But, what is the advantage, advantage is your major power consumption is an refresh cycle. And which is actually in a sense unnecessary consumption, although for retaining of the data, but not related to the computation. So, try to minimize the energy consumption there.

(Refer Slide Time: 53:13)



Next issue is next most important component is your interconnect the bus. And bus is also another source of power consumption. Because, signals have to flow in many cases power supply has to be provided. And if things are connected via wire in a bus, there would be power consumptions in the interconnect. So, the issue becomes that of dealing with power consumption in the interconnect.

In fact, you can have we are discussed this already segmented bus and multilevel segmented bus. So, you have the bridges and with that you can have things connected and here the component can be connected in either of it either of the buses. So, you have a multilevel segmented bus with bridge operating in between.

Now obviously, this multilevel segmented bus, increase the higher bandwidth. Because, the whole idea is that, the high speed transfer takes place on a smaller bus, where it is required. And the low speed devices are connected via bridge. So, effective transfer rate is more.

Now, these also saves power, it uses lower power. Why? Because, if I got to have a single level bus, which has to operate at a higher clock frequency. That means, there would be more consumption of power. But, many of the devices would cannot operate at that frequency, it will introduce wait cycles. But, during that period, the consumption of the power in a sense is a wastage.

So, if I have buses operating at two different frequencies, two different speeds, suited to the devices. Then, the power consumption becomes different it is not same for the entire bus on to which the devices are connected. The other thing, if you we impact in the last class also we discussed, the new trends in the bus architecture is that of having packet switching bus architecture.

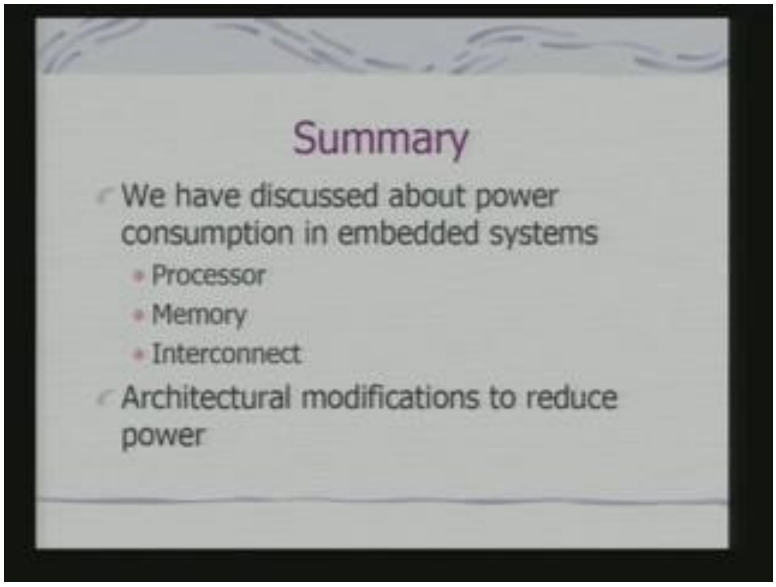
So, instead of having direct circuit switching, which is typically which happens on a bus you have packet switching. So, you packets are getting routed from one point to another point. So, it obviously, offers higher bandwidth concurrency. Because, there can be multiple packets being transfer simultaneously.

Effective spatial reuse of resources, why? Because, if there is a single transfer in terms of a circuit switching. A part of the spatial resources, because the connection between two

other elements, which are not participating in the transfer does not get utilize. But, when it is a packet switching through the router, you can utilize that part of the path as well.

These also becomes power efficient, why power efficient? Because, you are actually making use the bus much more optimally. If the bus resources are getting much more optimally used, your bus traffic is managed better. The moment you manage your bus traffic better, what you have? You have energy efficiency. So, packet switching bus architecture in another way optimizes energy consumption in interconnect.

(Refer Slide Time: 56:36)



This completes are discussion on this power wire architecture issues, we had looked at processor, memory as well as that of interconnect. And what we have really looked at is, basically the hardware issues, hardware design issues, hardware architectural issues. But, a very important component is software. How do you write your program, such that this features can be efficiently exploited OS cannot exploit this features, if you are software is not compatible to exploit this features.

In fact, program optimization one aspect of program optimization for embedded system is that of writing program. So, that your power consumption is minimize, is not just that of efficiency in execution. So, this finishes are discussions on power. In the next class we shall look at features for features which going to architecture and which enable as to do efficient device. Any question?

Student: ((Refer Time: 57:44))

Typically, the question your asking is whether you have discrete values of VDD. If you have continuous value, you have the best efficiency. But, it becomes extremely difficult to implement. That is why you get processors with discrete VDD value. And that is what is to be used for optimal energy budget.