Compiler Design

Prof. Y. N. Srikant

Department of Computer Science and Automation

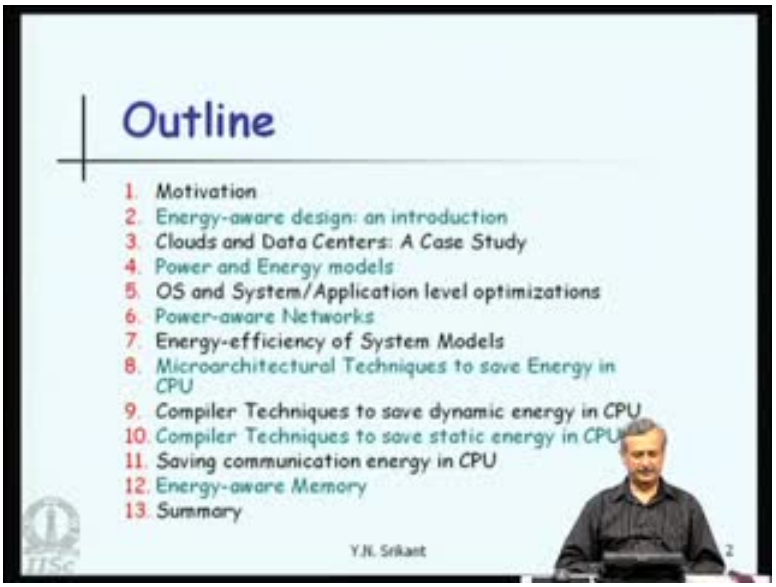
Indian Institute of Science, Bangalore

Module No. # 17

Lecture No. # 33

Energy-Aware Software Systems-Part 2

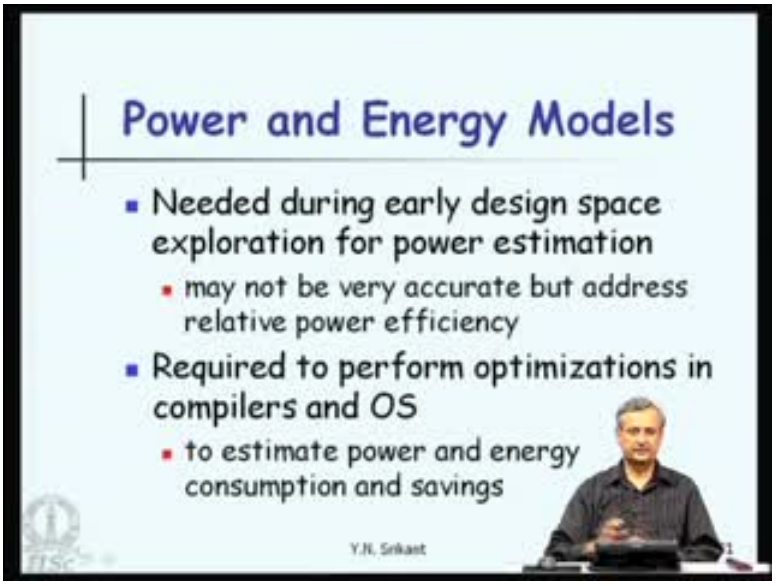
(Refer Slide Time: 00:22)



Welcome to the lecture on energy aware software systems. In the last lecture, we looked at the motivation for considering energy as a factor in design of software. We also went through an introduction on why this is important, what the consequences are etcetera.

And we considered a case study - that is the clouds and data centers case study. Today, we are going to consider power and energy model trend and take it from there onwards.

(Refer Slide Time: 01:01)



So, why are power and energy models needed? That is the first question. If you look at the design exploration for building let us say, hardware-software systems say embedded systems, very early in the exploration stage we require power estimation of the various alternatives that are available to us.

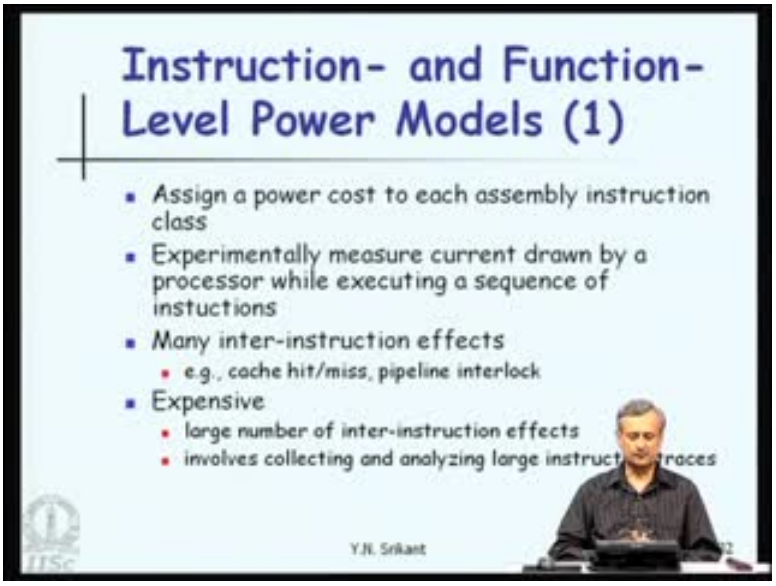
For example, whether a particular architecture and algorithm software combination requires less energy than another alternative - this is the question that we need to answer.

However, it is not necessary to have a very accurate estimate of the power, but it is enough to see the relative power efficiency of the various alternatives. So, the models that we are going to create will be slightly coarse, in some way and they will address only this relative power efficiency not the absolute power efficiency.

The power and energy models are also needed in compilers and the operating system, because these software actually control how the program is going to operate. Can we, for example, reduce the voltage and frequency of the processer so that the programs run slowly and consumes less power.

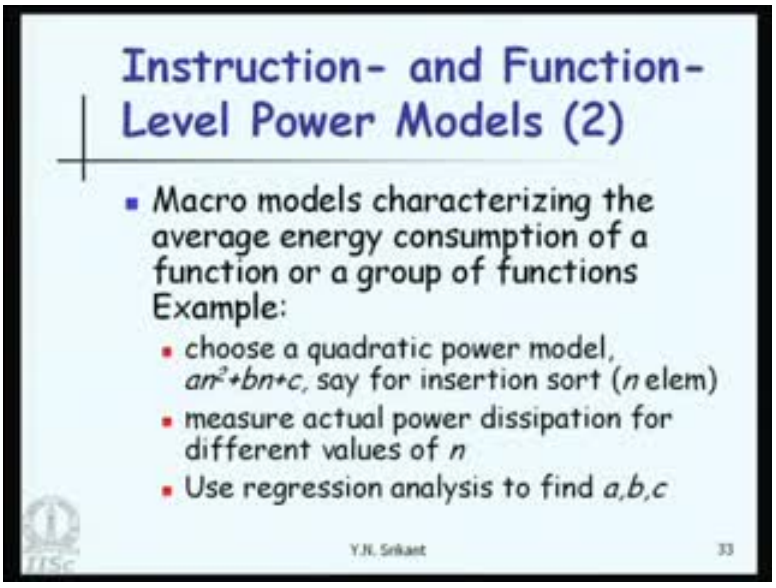
So, this is the kind of question that will be raised and to answer this question, an estimate of power and energy consumption of the program on a particular architecture will be necessary.

(Refer Slide Time: 02:58)



We considered the instruction and function level power model in the last lecture. So, an instruction level model assigns a cost to each of the instructions. This is a very expensive process because we need to look at many possible instruction mixes. Then measure the current that the processor uses and then finally, assign the power.

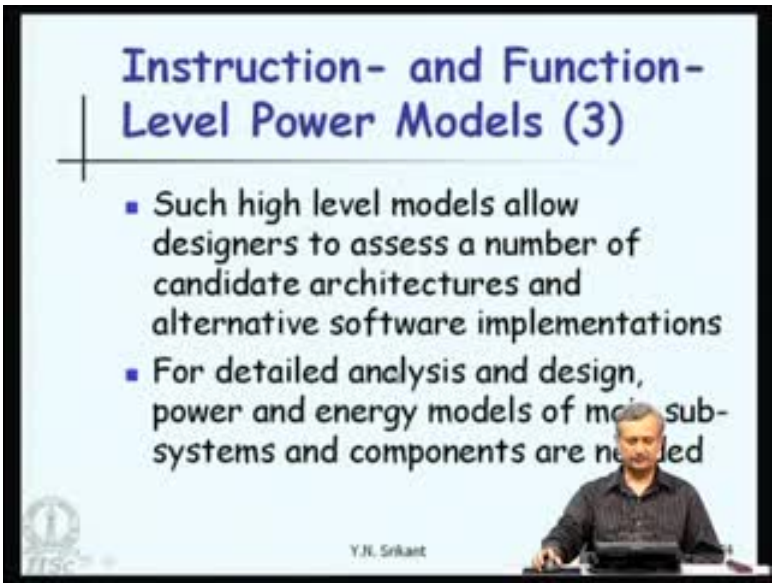
(Refer Slide Time: 03:27)



People also consider groups of functions and then see the average energy consumption of such groups. So, they fit you know when we know the algorithm, we probably know the complexity of that algorithm. Thereby, we can determine which type of a power model

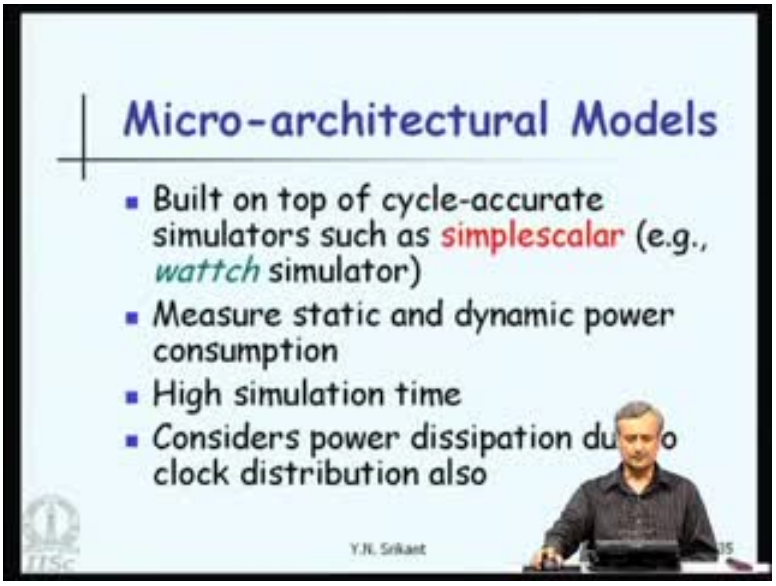
fits this algorithm say for an insertion sort it could be a n square plus b n plus c, etcetera. And then, you know, after couple of measurement we determine the values of a, b and c.

(Refer Slide Time: 04:05)



Such high level models allow what we aim for; that is, allow the designers to assess a number of candidate architectures and alternative software implementations. However, if we want little more fine grained analysis - a detailed analysis in design, it is not enough to know the power consumption at the higher level. It is also necessary to know, what type of power and energy consumption happens at the ALU level, at the register level, at the cache level, at the memory level, at the bus level, etcetera.

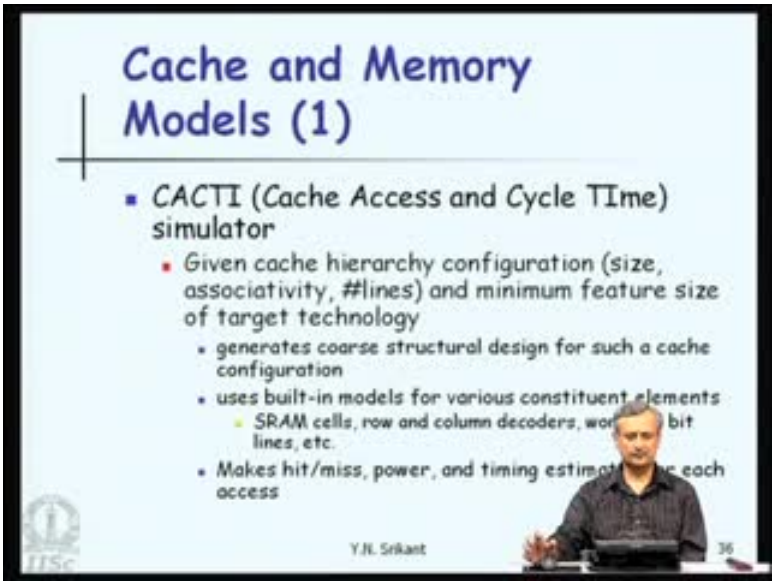
(Refer Slide Time: 04:48)



So, we definitely require power and energy models of main subsystems in components. Micro-architectural models are lower level energy and power models. These are usually, built on top of cycle-accurate simulators such as simplescalar. So, simplescalar is a very flexible simulator, it can simulate…. there are variants of simplescalar, which simulate x 86 type of architectures, alpha type of architectures, arm type of architectures and so on. And people have also added the energy computation layer in the simulator.

And for example, the wattch simulator tells us how much energy a program takes on a particular architecture; but the problem with this type of a simulation is it requires extremely high simulation time. It measure both static and dynamic power of course. But the time requirement is a problem. It considers power dissipation due to clock distribution as well because it is a lower level model. It is quite accurate. So, people use such models even though they require high simulation time.

(Refer Slide Time: 06:05)



Cache and Memory Models are the next that we consider. So, we need to simulate you know cache access cache energy has to be measured and so on. CACTI Cache Access and Cycle Time simulator is one such simulator. CACTI is a very famous simulator and very large number of people use this. So, given cache hierarchy configuration, say size of the cache, associativity of the cache, number of lines in the cache, etcetera and also the minimum feature size. What exactly is the technology that we are going to use to implement the cache mentioned?

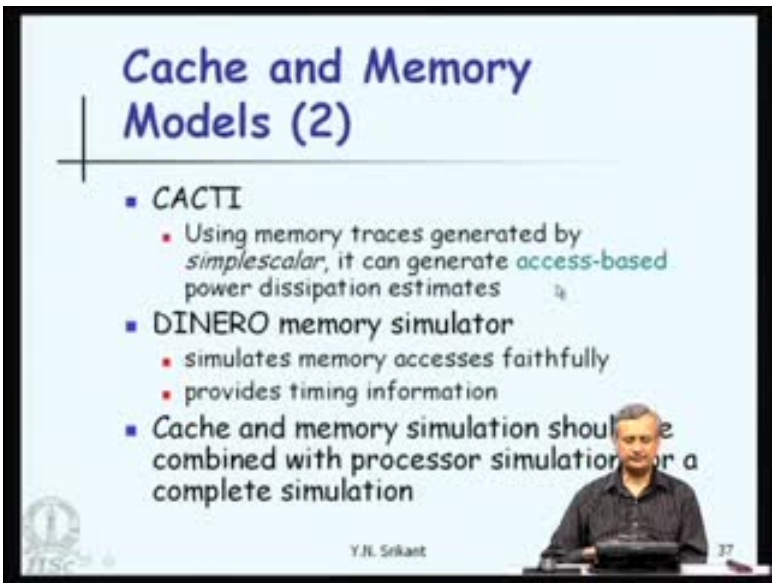
Is it a 90 nanometer technology, 45 nanometer technology, etcetera? Once we mention all these, the simulator generates coarse structural design for such a cache configuration. Why should it really do this? So, this really is the approximate chip layout for the cache, it indicates how much area is needed by such a cache so on and so forth.

Unless such detailed structural design is made, it is not possible to compute the power and energy consumptions of the cache. That is the reason, why such a coarse structural design is made by the tool.

It uses built in models for various constituent elements. In other words, there is a library already available for each type of technology that is for S RAM cells, row and column decoders; word and bit lines, etcetera; registers and so on.

So, with all this various buffer registers, it knows exactly how to put these two things together to achieve the cache configuration that the user has mentioned. Thereby, it synthesizes a structural design. It makes estimates of hit/ miss, power; timing requirements for each access.

(Refer Slide Time: 08:31)



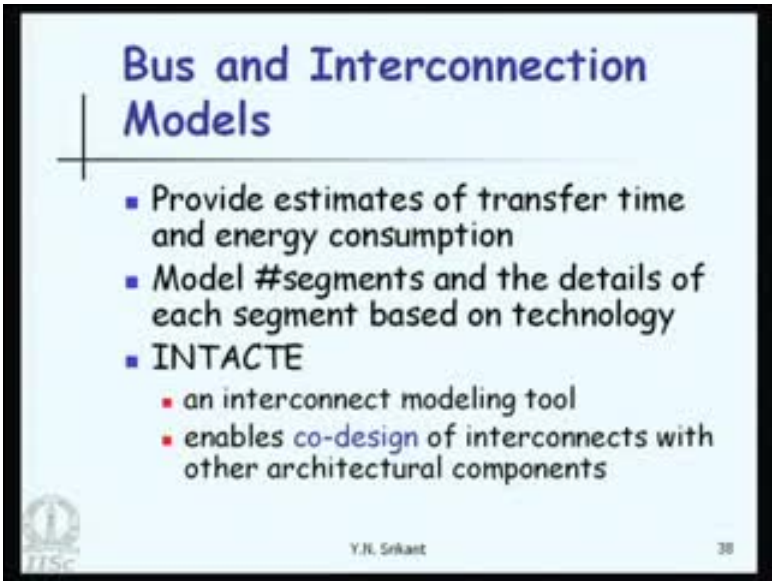
Using the memory traces generated by simplescalar you know these are fed to the cache CACTI stimulator. CACTI can generate access based power dissipation estimates for each access. It can tell you whether it is a hit or a miss and what is the amount of power or energy consumed by that particular access.

This is very useful; there is a simulator for the main memory called DINERO. CACTI is for the cache; DINERO is for the main memory. It may simulate memory accesses very faithfully and it provides timing information also on the memory accesses.

So, cache and memory simulation should be combined with processor simulation for a complete simulation of the program on the processor. If we have a processor simulator a cache simulator such as CACTI; memory simulator such as DINERO, then the first time the processor generators address, traces which addresses are going to be accessed by the program. These are first fed to CACTI to find out whether they hit or miss and then you get the estimate of the power etcetera.

If it is a hit then the contents are automatically used and if it is a miss then we need to feed the address to the DINERO memory simulator and then it gives you power energy requirement, timing information, etcetera, for that memory access as well. So this is how a complete simulation is run with these simulators.

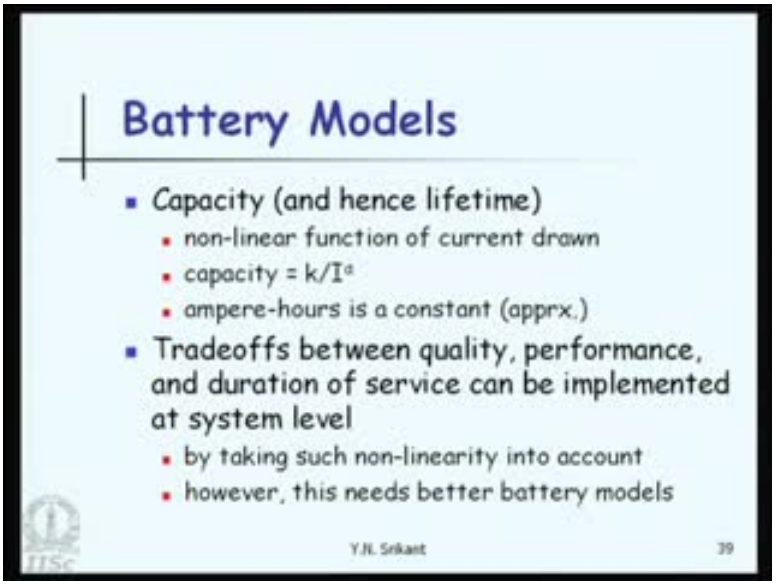
(Refer Slide Time: 10:12)



The next type of models we are going to consider are bus and interconnection models. So, as you know there are many buses and you know interconnections in a computer, in a chip CPU chip, these models provide a, estimate of transfer time and energy consumption on the bus or the interconnection. They model number of segments; the details of each segment based on the technology that is 45 nanometer, 90 nanometer, etcetera.

There is a tool called INTACTE which has been built by our team, it models interconnects and it enables co-design of interconnects with other architectural components also. We will see few more details of this a little later.

(Refer Slide Time: 11:03)



Then, what are battery models and why are they needed? So the battery models actually, model the capacity and life time of the battery. These are really non-linear functions of the current drawn capacity is actually, inversely proportional to current k by i to the power alpha, where alpha is some constant and amperes is approximately ampere hours is the product of ampere in hours is approximately a constant. So, the capacity in some sense is a constant it is a question of whether you are drawing more current for less time or less current for more time.

The Tradeoff between quality performance and duration of service can be implemented at the system level, using such models, so they also take into consideration non-linearity however, we know we need better memory this battery models in order to do these thing better. How are these tradeoffs what are these tradeoffs? for example, if we say we do not need such a high quality image then the algorithm can possibly skip some of the pixels and it can do coarse image processing, so then the performance will be much better because it going to be faster, but the quality and image will be slightly lower. However, since we are going to run the program for lesser duration the battery life will be enhanced.

(Refer Slide Time: 12:44)

Now before we consider, how to compute the power dissipation in a CPU we need to know what exactly is the power consumption at the lowest level that is a CMOS device a mosfet and so on and so forth.

So there are several types of power dissipation in a device, at the lowest level. One is the dynamic power dissipation the other is the static power dissipation and finally, the short circuit power dissipation. What is dynamic power dissipation? When a circuit performs the functions it was designed for then obviously, there is power consumed. This is so it is really doing some useful work say a ludel's audition (13:37) and in that one of the CMOS devices switches from 1 to 0 or 0 to 1 as a part of the adder. This is useful work.

This is dynamic power consumption dissipation and this is the dominant factor as of today. At the 90 nanometer level dynamic power is really the dominant factor, the other factors coming to play as we cross the 70 or 65 nanometer level.

And the dynamic power dissipation depends on circuit size, circuit complexity, speed and rate and switching activity. So, the models that we are going to build for dynamic power dissipation should consider these factors as well.

(Refer Slide Time: 14:35)

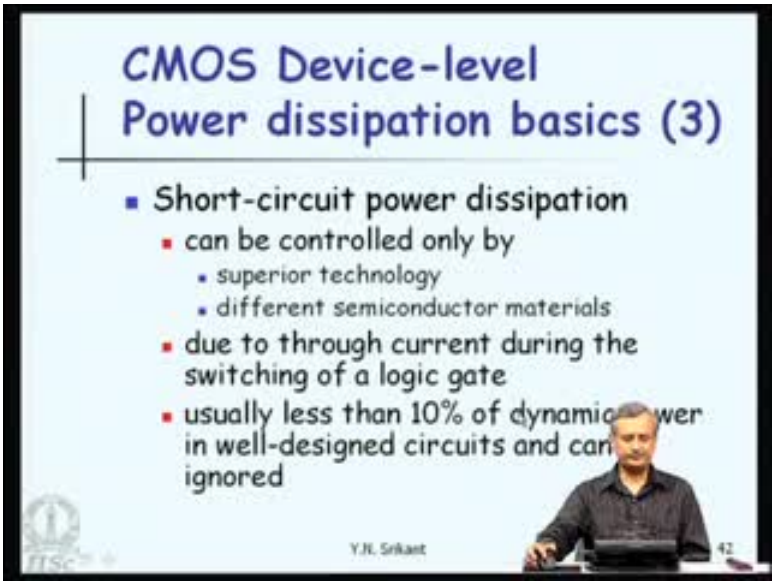
The second type of power dissipation that can happen in a CMOS level device, CMOS device is static power dissipation, so what is static power dissipation? Well, you know it is needed to preserve the logic state of circuits between switching activity, so in other words there is a lull (14:52) between two switching activities in a device, but then if the device has gone to state 1 it has to remain in state 1, until the next switching activity takes place.

And if it was in state 0 it needs to remain state 0 until the next switching activity takes place. So, to preserve this logic state of circuits between switching activities some power has to be dissipated and this is static power dissipation.

So, in some sense this is not useful activity, but it is necessary to make the circuit work properly. This static power dissipation is caused by the sub-threshold leakage mechanism in the device, it increases dramatically with shrinking device sizes this is the alarming problem.

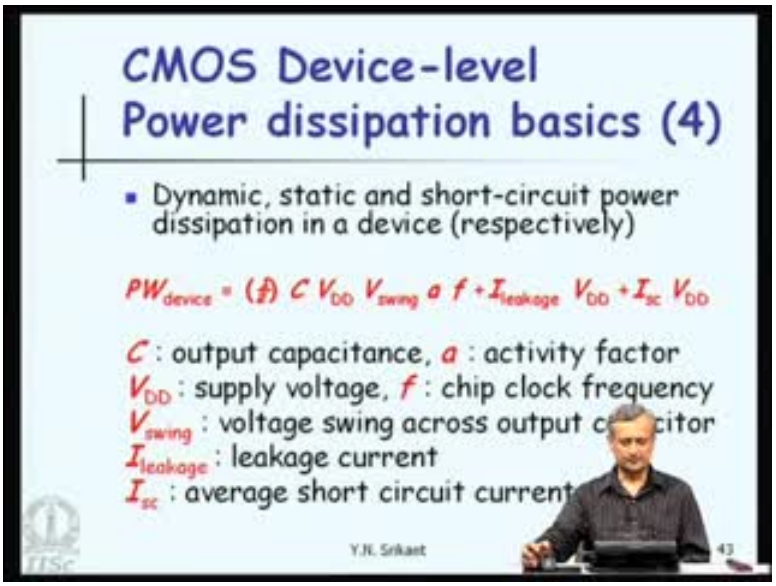
If we make the device CMOS device smaller, then the static power dissipation goes out and this is very significant for technologies below 70 nanometer and that is what I mention for above 70 nanometer dynamic power dissipation is the major factor.

(Refer Slide Time: 16:07)



The third type of power dissipation is the short circuit power dissipation and it can be controlled only by superior technology and different semiconductor materials rather than silicon so like gallium arsenide and so on. It is due to the current during the switching off through the due to through current during the switching of a logic gate.

(Refer Slide Time: 16:50)



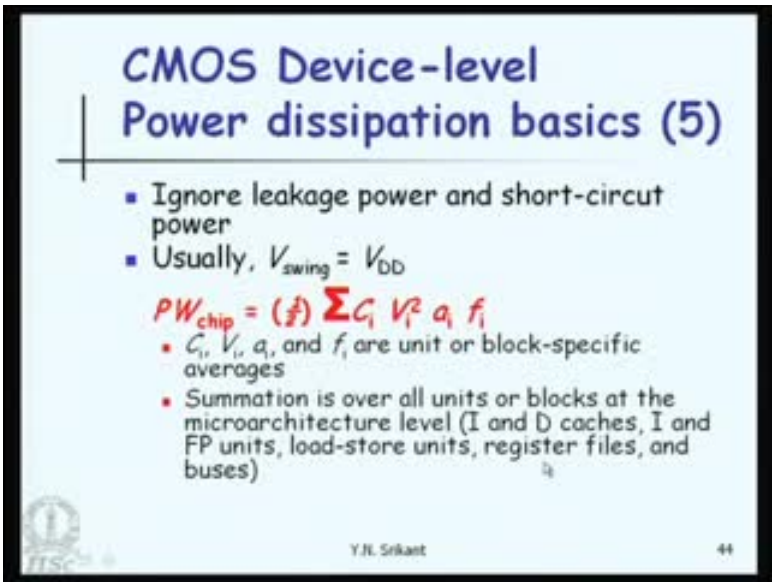
There is not much we can do to control this. It is fortunately less than 10 percent of the dynamic power in well-designed circuits and it can be ignored. How do we combine

these 3 factors and how do we model them to provide the total power consumption in the device.

So, power consumption in a device P W device has 3 parts. The first part which is f, you know half C V DD V swing a f is the dynamic power dissipation, I leakage into V DD is the static power dissipation and I s c into V DD is the short circuit power dissipation.

Here, C is the output capacitance of the device, a is the activity factor. In certain duration what is the percentage of time that you know the device switches. That is the activity factor V DD is the supply voltage, f is the chip clock frequency, V swing is the voltage swing across the output capacitor, I leakage is the leakage current and I s c is the average short circuit current. We have already discussed few of these.

(Refer Slide Time: 18:03)

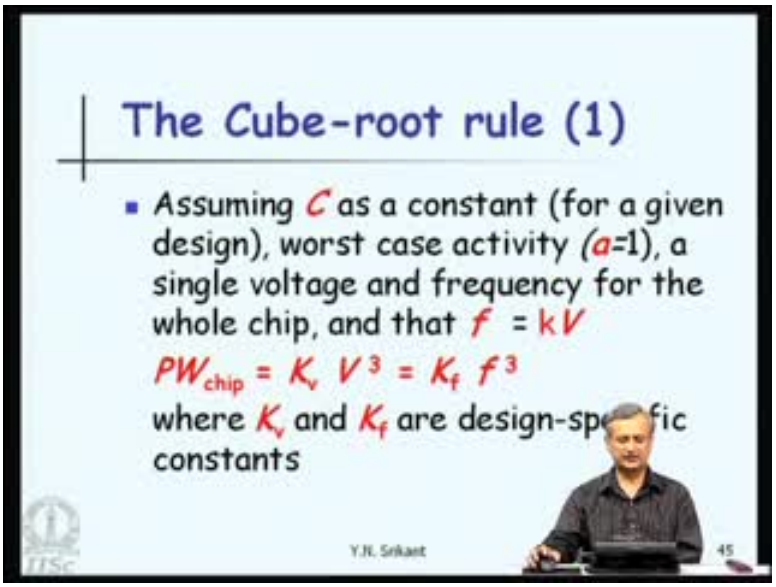


Now, let us see how to simplify this type of any equation. Suppose we ignore leakage power and short-circuit power. For technologies above 70 nanometer these 2 are actually very small so, we can ignore them. If we then only dynamic power dissipation is the most important and usually, the swing voltage is V DD, so if we take only the dynamic power dissipation then P W chip will be half sigma C i V i square a i f i.

So in other words, C i V i a i and f i are unit or block-specific averages and we are summing over all the units or blocks at the microarchitecture level. For example,

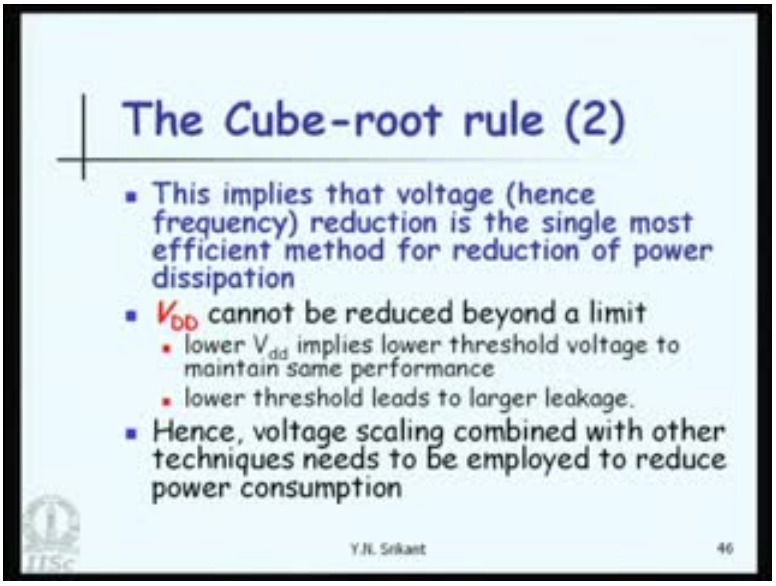
instruction data caches, integer and floating point units, load-store units, registers and buses.

(Refer Slide Time: 19:28)



The CMOS devices exist in all of them so, we just take the average rather the summation over all these so each one of them will spend some power. For each unit we are assuming that C i V i are different. We can probably simplify this even further and say C is a constant for a given design, worst case activity so that is a equal to 1, always active. That is it is never switched off and single voltage and frequency for the whole chip and let us assume that frequency is proportional to voltage.

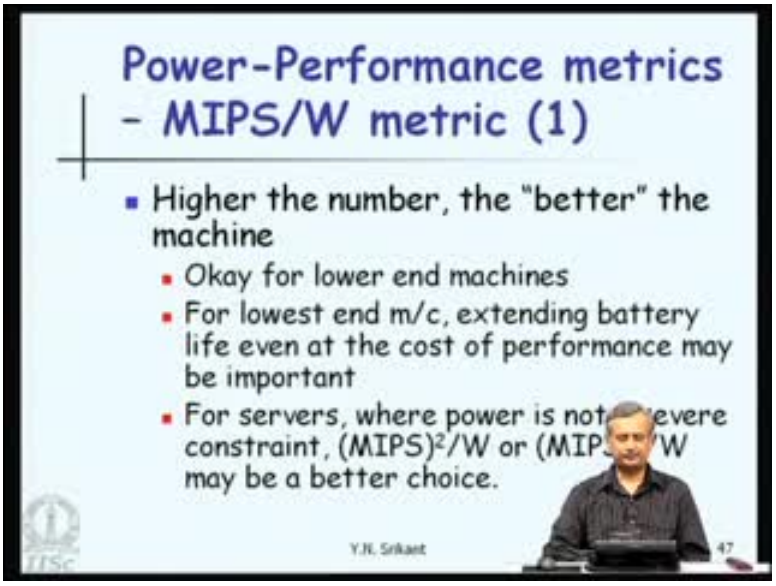
(Refer Slide Time: 20:13)



As voltage increases, frequency increases and as voltage decreases, frequency also reduces. Now, the power dissipation in a chip can be approximated as: k v into v cube or k f into f cube. That is k v and k f are the design specific constants, so the implication of this is quite heavy. This implies that voltage and hence frequency reduction is the single most efficient method for reduction of power dissipation why? Simple the power dissipation varies as v cube.

If we change this voltage even a little bit power consumption will change quite a bit. But our problem is we cannot always use the voltage reduction as a method of controlling power consumption because the source voltage V DD cannot be reduced beyond a limit. So, lower V DD implies lower threshold voltage; we need to lower the threshold voltage to maintain the same performance and lower threshold leads to larger leakage, so our equation will be wrong. We cannot ignore the static power or leakage power anymore at very low voltage levels. Therefore, voltages scaling combined with other techniques are also needed to be employed to reduce power consumption in processors.

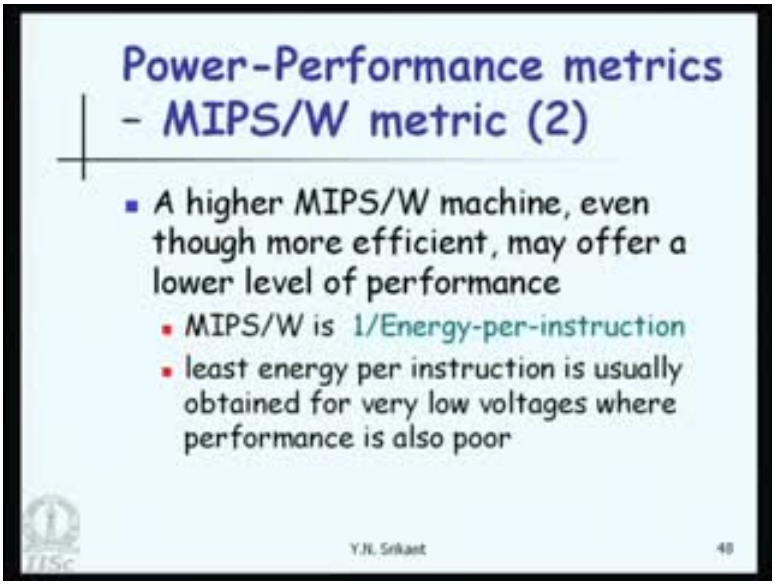
(Refer Slide Time: 21:33)



What are the usually used power performance metrics? So for example, people use MIPS per watt metric that is million instructions per second per watt. Higher the number the better the machine that is the understanding, but this type of a metric is for lower level p cs and machines for the lowest end machine, extending battery life even at the cost of performance may be very important.

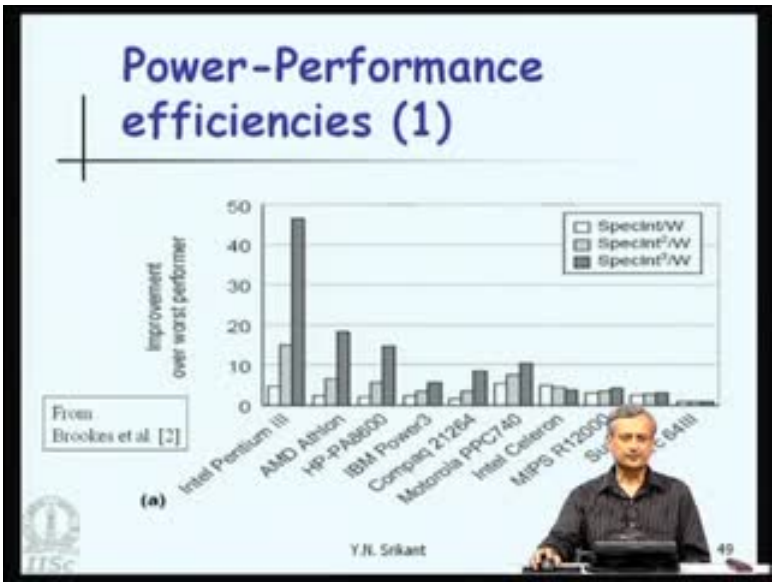
But for servers, where power is not a very severe constraint, MIPS per watt is not a very good performance metric, MIPS square per watt or even MIPS cube per watt may be a better choice.

(Refer Slide Time: 22:35)



A higher a MIPS per watt machine even though more efficient, that is it executes more instructions per watt. It may actually offer a lower level of performance why? MIPS per watt is nothing but 1 by Energy-per-instruction as you can see if you push that MIPS to below W then you get 1 by Energy-per-instruction and that means least Energy-per-instruction is what is necessary but it is obtained for very low voltages where performance is also very poor.

(Refer Slide Time: 23:21)

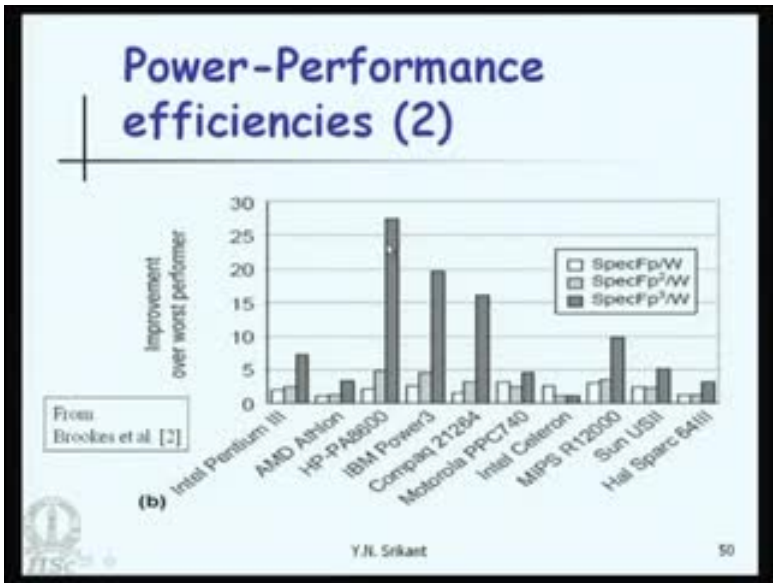


So, MIPS per watt is not a great way of measuring the performance, for all types of machines but it is for lower end machine. To show you why these differences are how these differences occur, let us take a number of these processors, this is a slightly old slide because we still have Pentium 3 here, but the relative performance is what we are looking it. The first bar is for you know SpecInt by per watt, the second bar is for SpecInt square per watt and the third bar is SpecInt cube per watt for various processors on the average.

If you look at the SpecInt per watt, the performance of most of the chips is similar not too much difference. But once you look at the SpecInt square and SpecInt cube for example, the Intel Pentium 3 has a huge SpecInt cube per watt. Whereas, the others which had higher SpecInt per watt like Intel Celeron, have much lower SpecInt cube per watt.

And so this goes to show and that SpecInt per watt is not really the only metric that should be used. So, you also observe that SpecInt per watt in the H P- P A 8 600 and SpecInt cube for watt in this 2 chips.

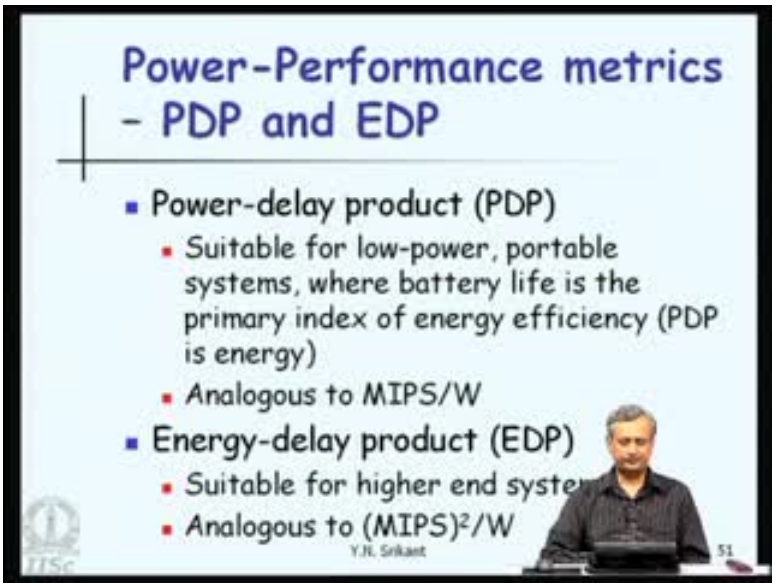
(Refer Slide Time: 24:52)



We are going to look at the floating point performance and you can see that there is a dramatic difference. Pentium 3 has this small Spec Fp cube per watt whereas, this PA 8 600 has a huge a Spec Fp cube per watt. In other words, we need to actually when we want to look at the performance and power tradeoff in a server, we need to consider all

this very carefully, look at the workloads and then decide at what level which metric has to used and at what level the voltage etcetera has to be maintain.

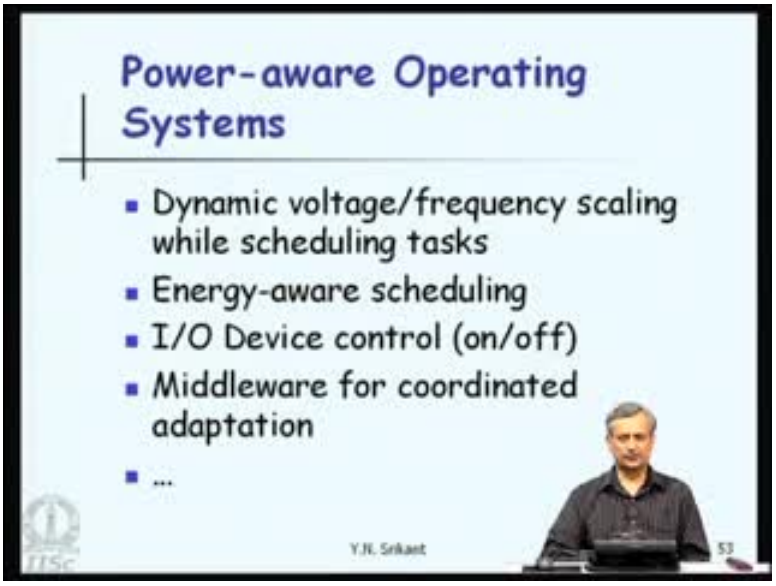
(Refer Slide Time: 25:31)



Then Power-delay product is suitable for low-power portable systems, where battery life is the primary index of energy efficiency, so PDP is nothing but energy and it is analogous to MIPS per watt and Energy-delay product EDP is nothing, but square per watt and it is useful for higher end systems.

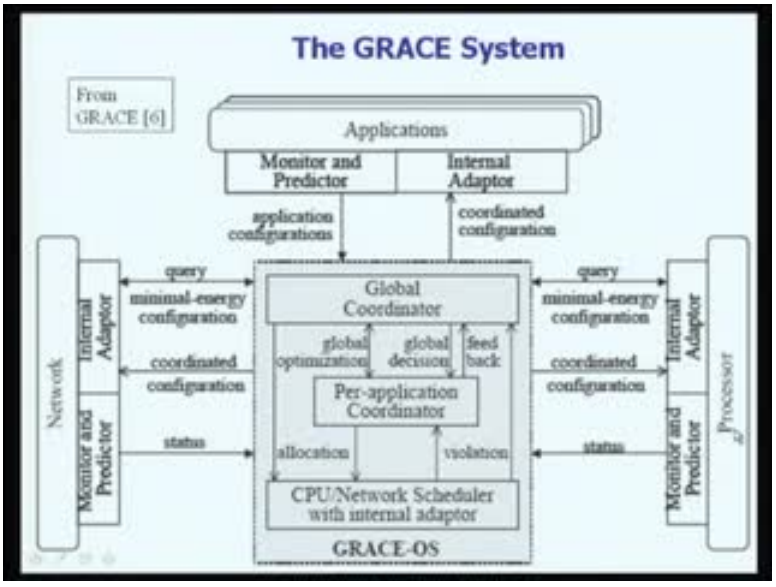
(Refer Slide Time: 25:54)

(Refer Slide Time: 26:03)



So, let us now look at some operating systems and system application level optimizations of course, with respect to energy. Operating systems can do dynamic voltage and frequency scaling while scheduling the tasks, so in other words, if there are many tasks and let us say the energy consumption of the various stars is kind of known, power energy consumption is known and we also know how much time each of these tasks takes. We know we can probably say more time is available for this task to run, so why not run it a bit slowly by reducing the frequency and voltage, so such a scheduling decision may be possible at the operating system level.

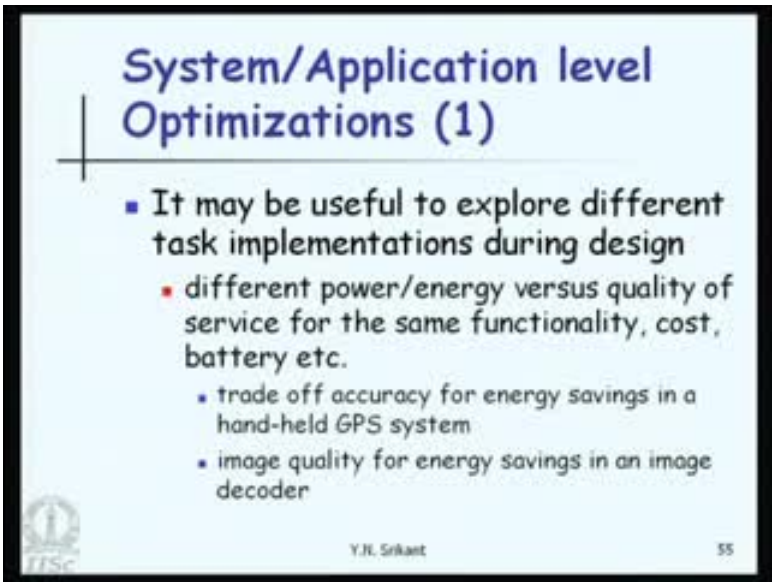
(Refer Slide Time: 27:06)



Then of course, it can do energy aware scheduling as well. I O device control is possible by the operating system and there possibly could be middleware for coordinated adaptation. So for example, the GRACE system, which was builted the university of Illinois Urbana Champaign, has a you know within the operating system a Global coordinator and a Per-application coordinator, for each application there are monitors and predictors internal adaptors, etcetera.

For example application, then network and we have the processor each of these is stated in a similar way, information from the device which as Processor or Network card etcetera is given to the operating system and then depending on its decisions it informs the Network or the Processor to behave in an appropriate way.

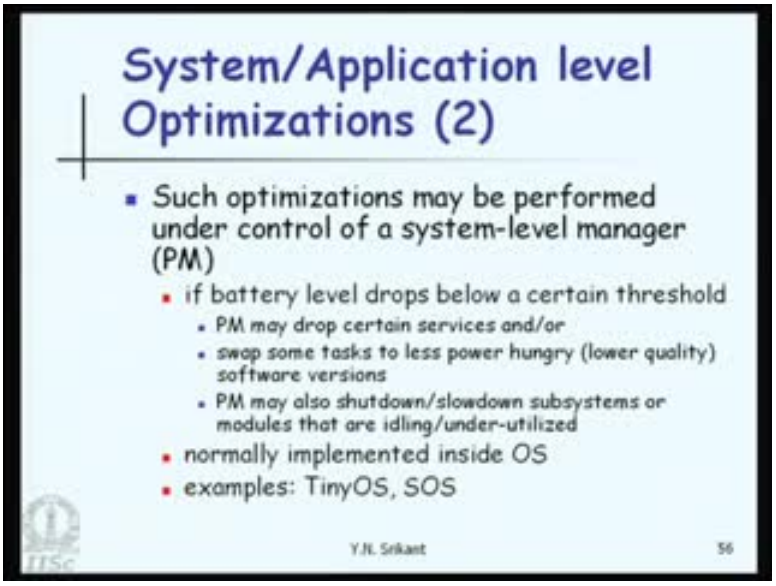
(Refer Slide Time: 28:06)



This is a very complex system and it has been shown to operate the system at a reasonably efficient level. What are system or application level optimizations possible? What are they and when are they possible? It may be useful to explore different task implementations during design so in such a case, we want to know how much power is used by a particular implementation on a particular device, so different power / energy versus quality of service for the same functionality, cost battery, etcetera are possible.

For example, you could say tradeoff of accuracy for energy savings in a hand-held GPS system. In other words, the computation can be coarse so it will give you some what approximate position information but at the same time it requires less energy.

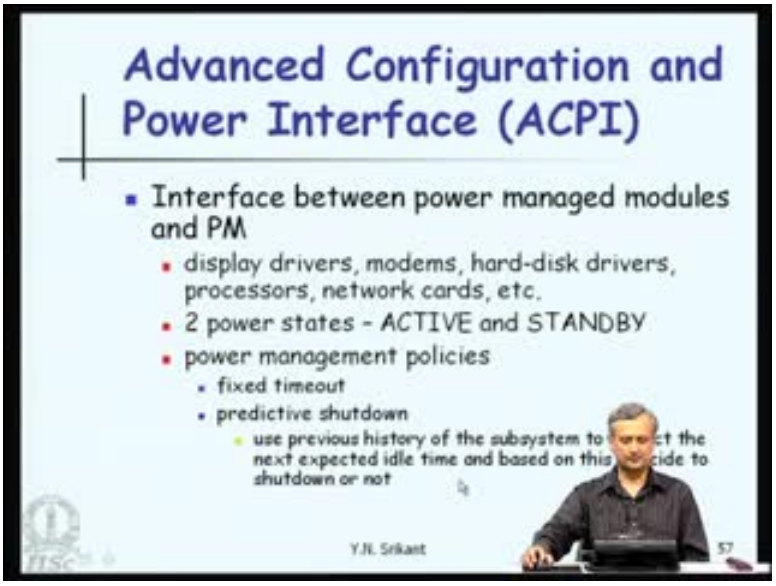
(Refer Slide Time: 29:09)



In an image decoder, image quality may suffer but energy may be saved so these are the tradeoffs that are possible at the system application level. Such optimizations may be performed under control of a system-level manager, so if a battery level drops below a certain threshold. For example - the power manager may drop certain services and possibly swap some task for to less hungry software versions. The power manager may also shut down or slowdown subsystems or modules that are idling or under-utilized.

So, these are all possible to and the battery life may be enhanced by such steps. These are all normally implemented inside the operating system for example, the tiny operating system Tiny OS, which is used in sensor networks or the SOS which is also used in sensor networks.

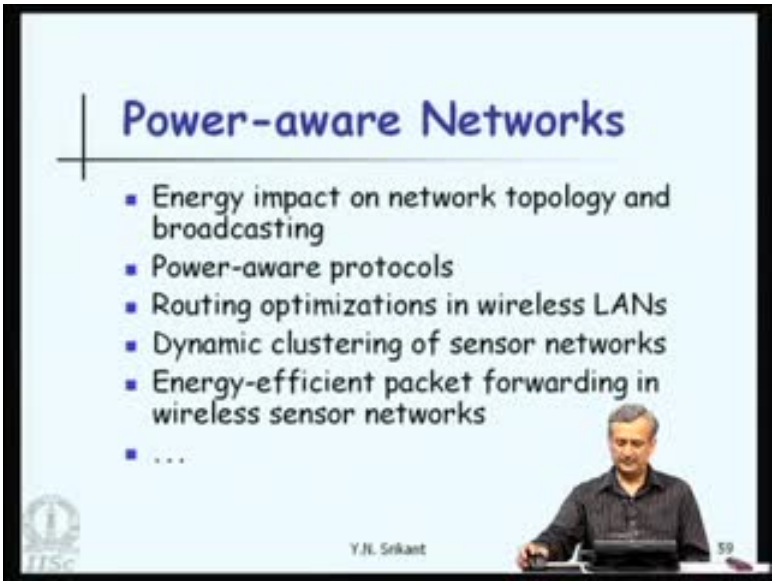
(Refer Slide Time: 30:03)



There is also an Advanced Configuration and Power Interface you know ACPI, which is available as standard. So, this is the interface between power managed modules and the power manager. For example, display drivers, modems, hard-disk drivers, processors, network cards, etcetera are all controlled through this interface. There are usually 2 power states in ACPI. ACTIVE and STANDBY and power management policies could be fixed time out. In other words, every after every few seconds fixed by the operating system or the user there is a switch from active to standby, a standby to active, etcetera. But this may not be correct in all cases and useful in all cases so predictive shutdown may be more useful. Use the previous history of the sub system to predict the next expected idle time.

And based on this decide the shutdown or no not allow. Whether to shut down the device or not to shut down the device, so this is the much better way and this helps in making the system more-friendly to the user.

(Refer Slide Time: 31:27)



So, let us see how power is possibly saved in computer networks? Say the energy impact on network topology and broadcasting. This is important, so what kind of topology is or energy efficient and useful for broadcasting, what are power aware protocols, what are the routing optimizations in wireless LANs and you know you it is possible to actually cluster sensor networks, dynamically to save power and energy-efficient packet forwarding in wireless sensor networks is also possible. So, let us look at a few of these - to understand how they work.

(Refer Slide Time: 32:12)

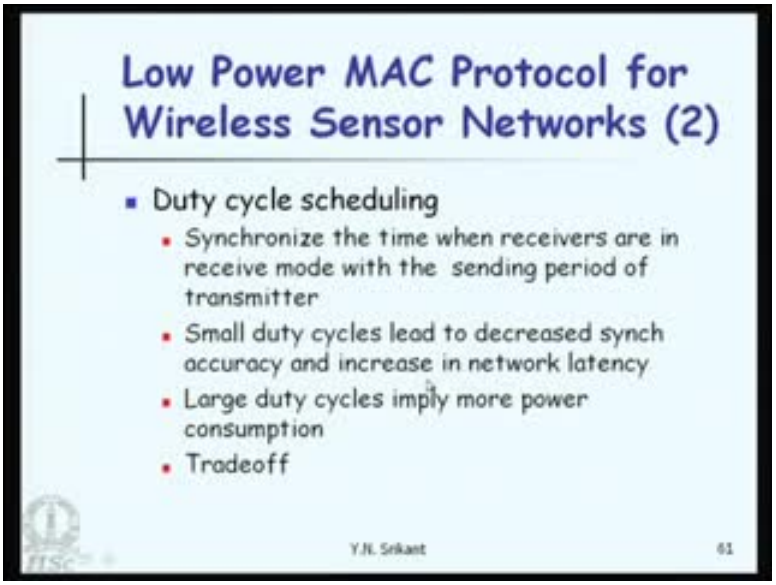
Lower level Low Power Mac Protocol so, this is a lower level protocol for wireless sensors or networks.

Sensors and actuators are integrated into the environment and are powered by cheap batteries, so this is how sensor networks operate. And it is not possible to charge or change batteries frequently because the sensors are all there in the field it may not even the possible sometimes to reach the sensor and then change the battery, so we may actually discard the entire sensor if the battery goes out.

Inside a sensor mode the RF transceiver is probably the biggest power consumer. So, receiving and transmitting on wireless is a very power hungry task. Power consumption for idle listening is almost the same as that of transmitting. In other words, keeping the transmitter in idle mode does not help at all. Whether you are transmitting or you are keeping quiet the power consumption is about to about the same.

And if the radio frequency transceiver is in receiver transmit mode, for only 1 percent of the time, so in other words the rest of the time it is shutdown, overall system performance can be reduced by about 50 times. So, if you keep the RF transceiver on all the time you save no power because idle or otherwise power consumption is the same. But if you switch off the RF transceiver and keep it on only for 1 percent of the time that is the duty cycle is only 1 percent, then the system power consumption for the transceiver will be reduced by 50 times, not 50 percent 50 times.

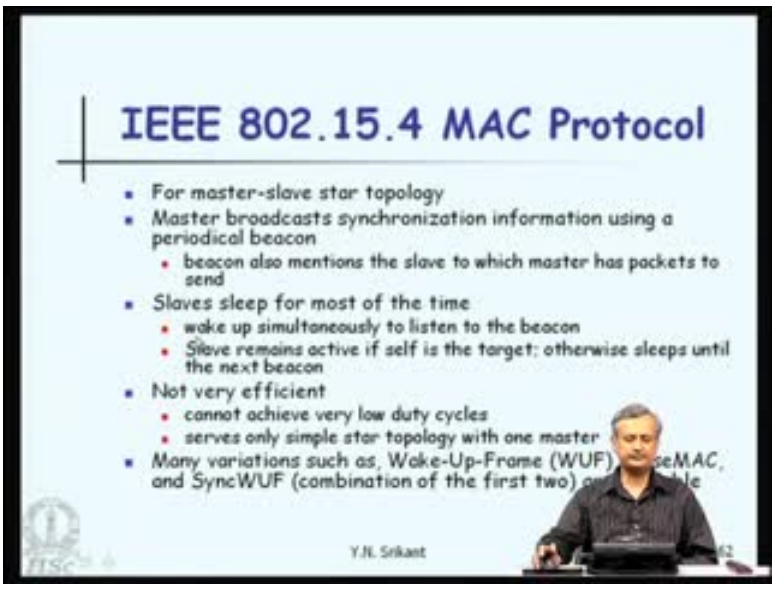
(Refer Slide Time: 34:23)



Duty cycle scheduling is a very important task in a low power mac protocol. How does this work? It synchronizes the time when transceivers are in receive mode with the sending period of transmitter. So, it is on for a very short duration, the receiver is on for a very short duration and at exactly the same time the transmitter is also on, that is the assumption. If this happens, the transmitter transmits and the receiver receives and then both of them sleep again.

Very small duty cycles, lead to decreased synchronization activity, so if we keep the transmitter and receiver on for very short durations, then they may miss each other. There may be some clock synchronization problems and because of this when the receiver is on the transmitter may be off and vice versa. And if the duty cycle is slightly larger than this problem does not arise but then more power consumption is the result.

(Refer Slide Time: 35:34)



So, let us look at the, I triple E 8 naught 2.15.4 mac protocol. This is actually for master-slave star topology only. The master broadcast synchronization information using a periodical beacon. So, once in a certain number of cycles the master broadcast synchronization information, beacon also mentions the slave to which the master has packets to send.

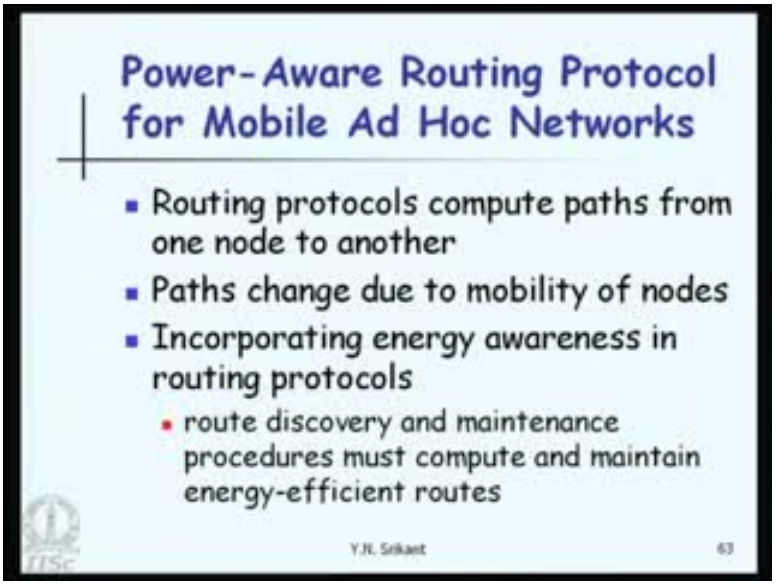
So, this is synchronization information and then the information itself, comes a little later. Slaves Sleep for most of the time they wake up simultaneously, at a fixed time to

listen to the beacon. If the beacon goes on for a reasonable amount of time, then all the slaves will be guaranteed to listen to it.

This is the basic principle. Slaves remain active, if self is the target that is the information is going to be received by itself otherwise, the slave goes back to sleep until the next beacon arrives. This is not very efficient, because it cannot achieve very low duty cycles, as I said, the beacon has to actually broadcast the synchronization information for a reasonable period, so that even without you know with bad clock synchronization, the slaves receive the beacon and then decide what to do. Very low duty cycle is not possible and it serves only simple star topology with one master only the master sends information and the slaves receive it and it is not possible for slaves to exchange information among themselves.

Many variations of this are possible Wake-Up-Frame, WiseMac, SyncWUF etcetera. Where even the beacons are spread over time short beacons many times, etcetera and these have been shown to work better.

(Refer Slide Time: 37:53)

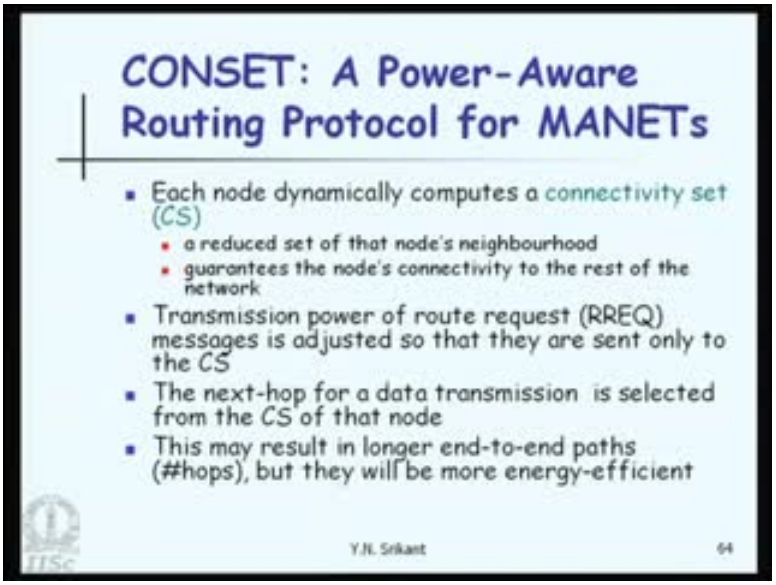


So, what about the Routing protocol? We saw how power can be saved in a Mac protocol, but about the Routing protocol.

Routing protocols compute paths from one node to another otherwise, it would not know how to route information and paths change due to mobility of nodes. So, incorporating

energy awareness in Routing protocols means, route discovery and maintenance procedures must compute and maintain energy-efficient routes so it is not enough to just look at the distance, we must now look at the energy consumption on the routes and maintain energy efficiency route information.

(Refer Slide Time: 38:34)



How does it do it let us there is a particular algorithm called CONSET so, let us look at it. Each node dynamically computes a connectivity set that is the CS. What is CS?

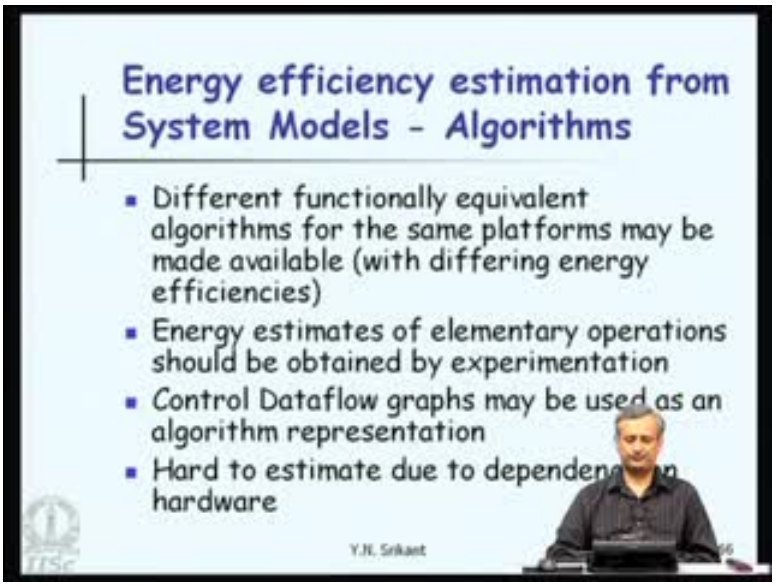
It is a reduced set of that particular nodes neighbourhood, so there are many nodes in the neighbourhood of a particular node, so the nodes that guarantee the particular nodes connectivity to the rest of the network are the once, which are included in the connectivity set. If you send information if our nodes send information to one of the nodes in the CS, it is guarantee that the information will reach all the nodes, it is possible to make the information reach any node in the network.

This transmission of transmission power of the route request message RREQ is adjusted so that they are sent only to the CS. See, if we have to send if the particular node has to send or broadcast information to all the nodes in the network then it may require a large transmission power because some of the nodes are very far and some of them are very near.

Whereas if we take only the neighbourhood which is very close to a particular node, then we need to make sure that the information from the transmitter from the broadcaster reaches only those which are in the neighbourhood that is the connectivity set. We do not have to really broadcast so that it reaches every node in the network. The next-hop for a data transmission is selected from the CS of the particular node.

So, we transmit to the CS set of our node then the node in the CS set will compute its own CS and then choose the next-hop and so on and so forth. This actually may result in a few more hops than the shortest path that is available but all these are going to be energy efficient paths. So, we may be spending much less energy when we go through this CS method rather than the shortest path method.

(Refer Slide Time: 41:03)



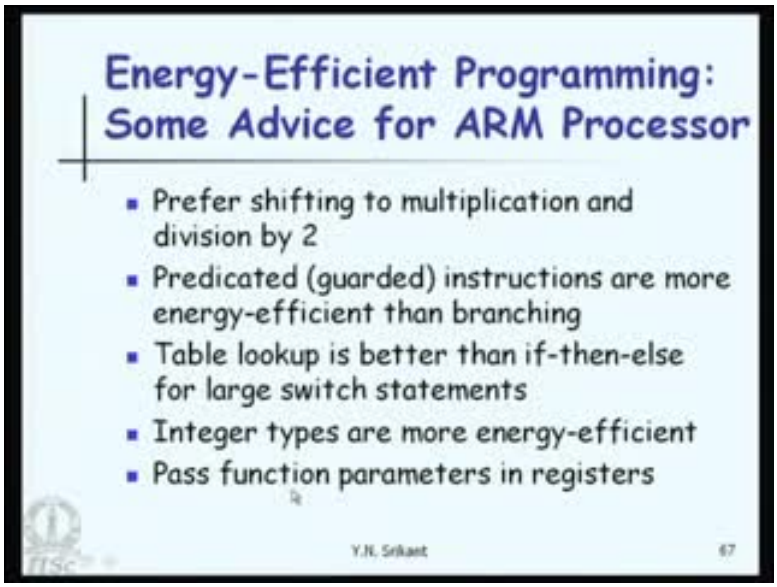
How is energy efficiency estimation carried out from system models? So, let us consider this task. The first system model that we are going to consider are the algorithms so, the question is very simple given an algorithm, can we say this algorithm requires so much energy consumption, that is the question. We do not have an energy complexity very similar to time complexity, the only way we can determine the energy consumption is actually by running or considering the energy consumption on various physical platforms for various algorithms, which are available to us. So, different functionally equivalent algorithms for the same platform may be made available. So, if we have different

platforms then we must have functionally equivalent algorithms for all these platforms each platform considered separately.

So, each of these platforms and different algorithms may actually have different energy efficiencies. Energy estimates of elementary operations should be obtained by experimentation, so in the high level language or the algorithmic language every operation such as plus, star, minus, etcetera are the branch comparison must be given a particular amount of energy consumption and this can be done by experimentation on the platform or a simulator.

Control data flow graphs may be used as an algorithm representation so we know what data flow graphs are - we draw them and then use that for estimation of energy consumption, assuming that we know the energy consumption of elementary operations. These are very hard to estimate due to the dependence on hardware. So, depending on how accurate our energy estimation of elementary operations is and how accurate the data flow graph is the estimates of the energy consumption of an algorithm for a particular platform will vary.

(Refer Slide Time: 43:49)

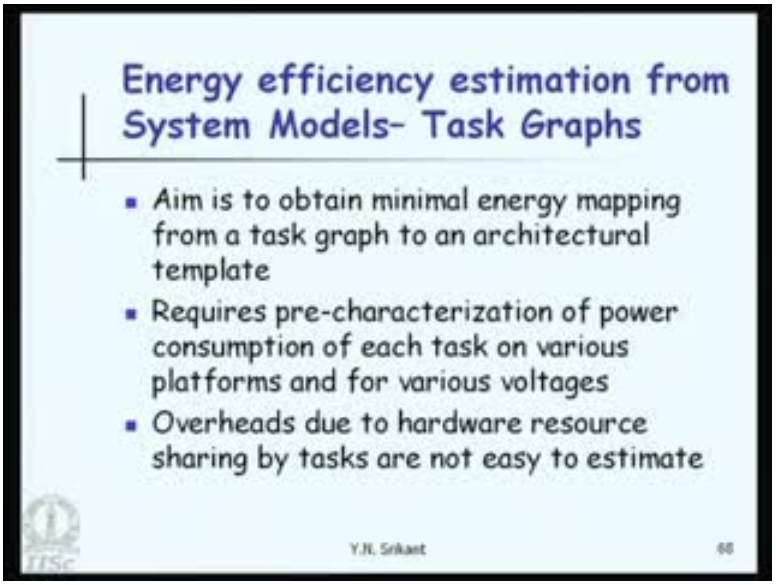


So, for Arm Processors typically let us say what type of advice does one give if energy efficiency is the consideration?

Prefer shifting to multiplication and division by 2. This is very obvious this saves time and also saves energy. Predicated instructions are more energy efficient than branching so if we have predication hardware then use it.

Table look up is better than if-then-else for large switch statements. So, this again time wise also this better. Integer types are more energy-efficient the than floating point types and passing function parameters in registers is better than passing it in stack. These are simple tips for programming efficiently with respect to energy on Arm processor.

(Refer Slide Time: 44:45)

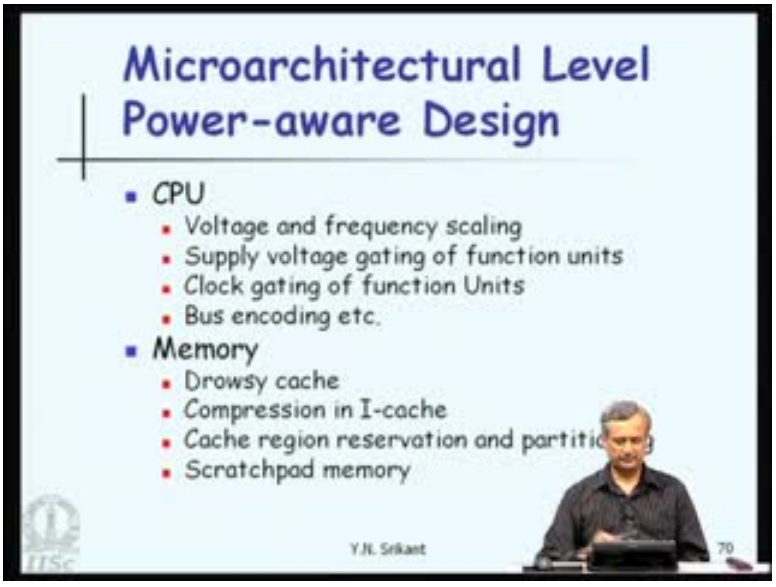


So, what about the next model let us say the Task Graph? There are actually many tasks each of these tasks has to be mapped to an architectural template, so that is our aim. Aim is to obtain minimal energy mapping from a task graph to an architectural template. So, we need pre-characterization of power consumption each task on various platforms and for various voltages so this is a hard job.

Overheads due to hardware resource sharing, etcetera by hardware resource sharing by task is not easy to estimate so, if many tasks run on the system and they share resources so by characterizing each one of the task separately will not give us the exact estimate so this will be somewhat incorrect.

(Refer Slide Time: 45:39)

(Refer Slide Time: 45:50)



Now, let us move on to Microarchitectural Techniques to save energy at the lowest level in a CPU, how is energy saved?

So, at the CPU level, we have voltage and frequency scaling possible. So, that is the CPU is to be made to run at a lower frequency and lower voltage then obviously, we have seen from the CMOS device model that the power consumption will vary, energy consumption will obviously vary. Supply voltage gating of function units is possible.

So, as I mentioned the static power consumption will not go away unless you switch off the unit, so for example, if arithmetic login unit ALU or a cache line is not being used. It is better to stop the power supply to that particular function unit, so that the static power dissipation in that unit is going to be 0.

Supply voltage gating of function units is another technique at the Microarchitectural level to control energy consumption. Normally, supply voltage getting function units is not done by programs it is done by the architecture itself. So, what happens is the electronics just before an ALU, keeps track of a brief history of what the function unit status was. If the function unit status was idle for let us say 1 or 2 cycles then, automatically the electronics associated with the gating electronics associated with the ALU features of the power are the power supply to the function unit and when there is a request to use the function and it brings back the power supply and makes the function unit active again.

This happens automatically based on the architecture rather than the program. Bus encoding is possible so the pattern on the buses, so it can be is a sequence of 0s and 1s, so whenever there is a 1 the bus switches to high state and whenever there is a 0 it reaches to 0 states. So, if we control the number of such switches we actually save energy, so that may be possible inside a CPU.

This control may be possible inside a CPU. What about Memory? There are what are known as Drowsy caches so the as you know cache has many lines, so if a cache line is not accessed for a several cycles, then possibly we can bring the cache line rather cut the cache line power supply bring it to a sleep state or standby state, so that the power consumption of that particular line is reduced.

What can go wrong here? It is possible that the cache line loses its data. In such a case, it is a destructive scheme, so we may have to load that is cache again after it comes back to active state. The other thing is it goes to standby state it retains its data. If it retains the data when it comes back to active state, we can still access the data.

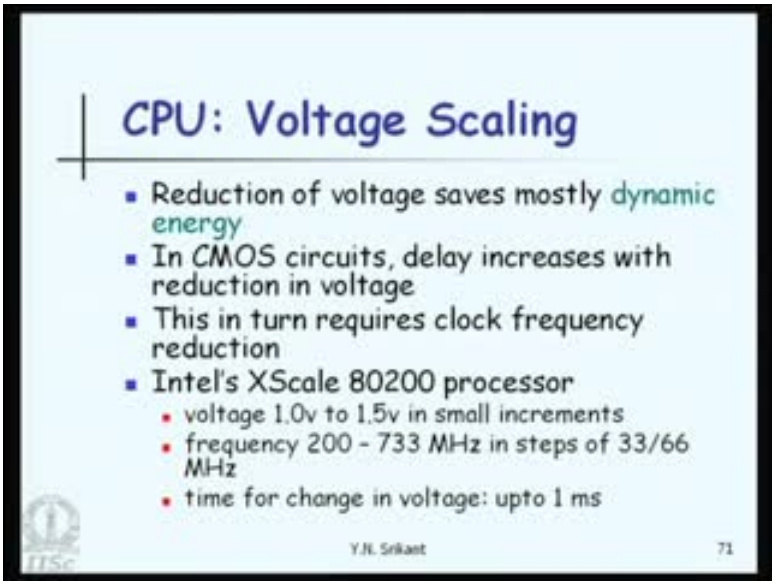
It is possible to compress information in an instruction cache. If we compress information in the instruction cache, the instruction will all be kind of a in a compressed encoded state, so we may have we will have to decompress the instruction before it is executed by the processor and obviously, compression happens you know in the

compiler and decompression happens when the instruction is going to be run on the processor. So, decompression, power consumption for decompression should not be very high.

Cache region reservation and partitioning is another technique so, we are going to look at this in some detail a little later. It is possible to reserve parts of cache for as separately for different parts of the program, thereby or different variables and thereby control it in a much better fashion.

Scratchpad memory is an alternative to cache and this actually saves energy quite a bit, so and we are going to consider scratchpads in some detail little later.

(Refer Slide Time: 51:00)



Let us look at CPU voltage scaling a little more in detail. So, when we reduce the voltage in a processor, we are going to save only the dynamic energy. As I mentioned before static energy or the leakage energy cannot be saved by reducing the voltage. It can be saved only by cutting off the power supply to the unit.

Dynamic energy can be saved by reduction of voltage and in CMOS circuits the delay increases with reduced voltage. In other words, if we reduce the voltage of the CPU, then the clock frequency automatically must be reduced otherwise, the device cannot function. Obviously, voltage reduction implies clock frequency reduction and therefore the program will run for a much longer duration.

The problem is are we saving energy in the process, the program used to run for let us say 10 seconds now, we reduce the voltage and it runs for 20 seconds. So, energy is nothing, but power into time. Even though power consumption has gone down the time requirement has gone up so are we saving energy?

The compiler or the operating system actually has to make this judgment, it has to determine whether it is worth running the program at a lower voltage or it is not worth running the program at a lower voltage and this type of a decision requires energy models for the program and the hardware that is why, we looked at these models a while ago.

Are such voltage changing features available in processors, yes for example, in the Intel X Scale 80200 there are instructions to change the voltage you know from 1.0 to 1.5 in small increments. So, the program can change the voltage of the chip. Of course, the frequency automatically scales from between 200 and 733 Megahertz in steps of 33 or 66 Megahertz, as we change the voltage and the penalty is the time for change in voltage which is very high it can be up to 1 millisecond.

So, when we actually do voltage scaling one has to keep in mind that not only does the program slow down, the time for changing the voltage from 1 voltage to another voltage. Whether you are going in the positive direction or reducing the voltage in the negative direction, the change of voltage requires up to 1 millisecond and during this time the processor really cannot do anything.

This adds to the processors the total program time, so it is necessary that either the operating system or the compiler take all this into consideration, before it determines makes a decision that the voltage of the processors has to either reduced or increased. We will stop here during this lecture and continue our discussion in the next lecture. Thank you.