Operating System
Digital Assignment 2
Power Efficient Scheduling

Name: Om Ashish Mishra

Registration Number: 16BCE0789

Slot: A1+TA1

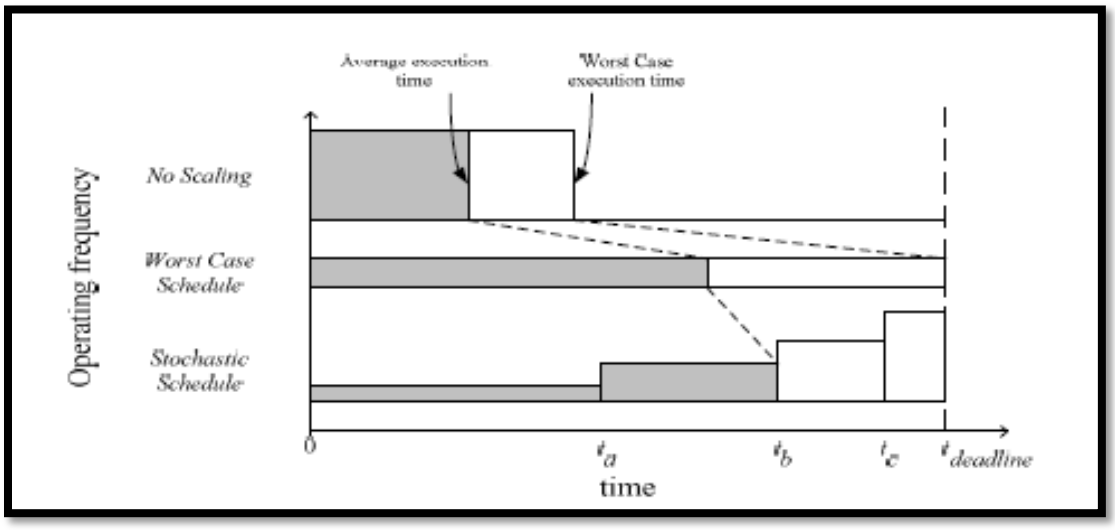
Introduction:

Multi-core systems are designed to bring the processor power consumption and heat density down to a manageable level without sacrificing performance. This claim is based on the dynamic power characterization of a CMOS device: $P = C * V^2 * f(1)$, where P is power, C is the device capacitance, V is the voltage, and f is the frequency. At the same time, we are entering a world of ubiquitous computing: there are microprocessors in watches, phones, televisions, simple kitchen appliances, cars, and power grids. Many of these systems need to operate within certain time bounds to ensure proper function. Embedded systems are often designed within a prescribed power envelope. This power constraint might be defined by thermal boundaries or by energy constraints, such as in a battery or solar cell-operated environment. Performance and power requirements must be carefully weighed against each other to ensure that the system functions as desired. With the current trend towards multiprocessors and the need to minimize power, there is a demand for a hard real-time multiprocessor scheduler that optimizes for minimal power usage.

Important Facts

Much work has dealt with creating power efficient, real-time schedulers, with one of two goals: minimize the power for the worst case scenario or for the typical scenario. The first type of scheduler assumes that the task will always require the maximum possible execution time possible. Under this assumption, the scheduler

scales back the operating frequency of the processor so that the task completes exactly at its deadline. This is an optimal solution for scheduling a given task set on a multiprocessor while providing hard real-time guarantees. The second type of scheduler, assumes that the task will execute in an average time and sets out to minimize the power. The scheduler sets the operating frequency such that the task on average will finish execution by a set time $t_a < t_{deadline}$. If the task does not finish executing by t_a then the scheduler increases the operating frequency such that the task can finish the task execution by $t_{deadline}$.
Diagram below depicts the differences in the speed scaling schemes.



Processors are based on CMOS technology where dynamic power is the bottleneck. Dynamic power (due to switching activity):-

From Physics

- $P \alpha V^2 * f$
- $V \alpha f$ V: voltage; P: power; E: Energy
- $E = P * Tcc$ $Tcc = CC/f$
- $E_i = K * cc_i * f^2$

Where Tcc= execution time; CC_i =clock cycles of task T_i.;f =frequency at which T_i is run.

Variable Voltage Processors

Modern processors operate at multiple frequency levels.

- Crusoe Processor: Transmeta Corporation
- PowerNow! Technology: AMD
- Intel XScale: Intel

Higher the frequency level higher the energy consumption

Crusoe processor

What is the power consumption of a Crusoe processor?

The extremely low power consumption delivered on multimedia applications is due to a new feature called LongRun power management. LongRun has the distinct ability to analyze the application workload dynamically and to adjust continuously the processor's speed (MHz) and voltage to provide the necessary performance. This new feature promises to extend the battery life of all applications, most specifically those requiring the constant attention of the processor. This is a dramatic departure from today's ultra-light PCs, which are incapable of delivering over one and a half or two hours of runtime for DVD movies.

Dynamic Voltage Scaling (DVS)

DVS scales the operating voltage of the processor along with the frequency. Since energy is proportional to f^2, DVS can potentially provide significant energy savings through frequency and voltage scaling.

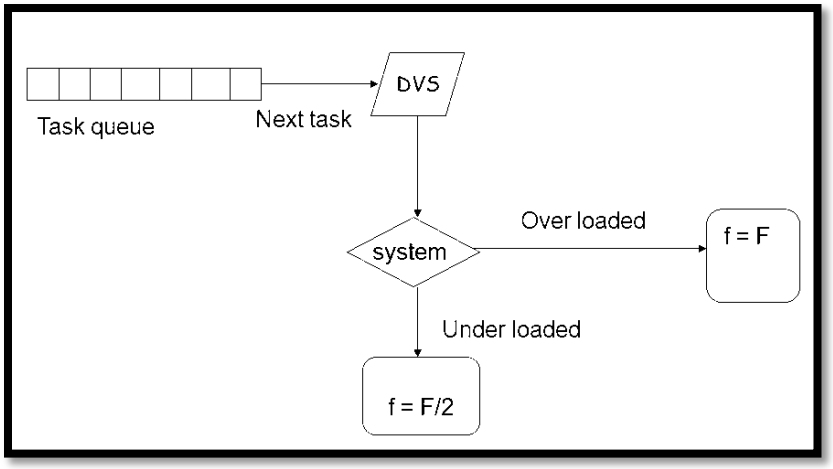
Case study (smart phones)

Device	Chip maker	Processor name	Frequency	# cores
iPhone 5	Apple	A6	1.02GHz	2
Galaxy S III	Samsung	Exynos 4412	1.44GHz	4

Motorola RAZR	Ti	OMAP 4430	1.2GHz	2
HTC one S	Qualcomm	MSM8260A	1.2GHz	2
Asus transformer	NVIDIA	Tegra 3	1.3GHz	4
New iPad	Apple	A5X	1 GHz	2

- Practical multi-core processors
- Contemporary multi-core processors have more than 2 cores at about 1 GHz.

Simple DVS-Scheme



Consider a task with a computation time 20 units.

Energy of T_i without DVS:

$E1 = K * 20 * F^{2.}$

Energy of T_i with DVS:

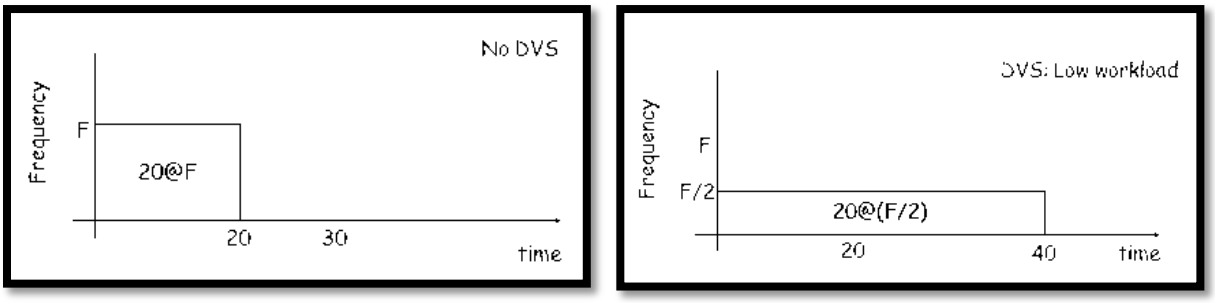
$E2 = K * 20 * (F/2)^{2.}$

Clearly, E2 = (E1)/4

Therefore, if we reduce the frequency we save energy but, we spend more time in performing the same computation.

Simple DVS scheme handling RT-task

Applying the simple DVS scheme
T1 runs at maximum frequency (F) and meets the deadline with no power savings and T1 runs at half the maximum frequency (F/2) and completes at time = 40 thereby missing its deadline.



Therefore power consumption is important for (i) Minimizing energy consumption and (ii) Meeting the deadlines.

Real Time - DVS schemes

The RT-DVS algorithms can be broadly classified based on the granularity at which voltage scheduling is performed as follows :-

Inter-task DVS scheme: Voltage scheduling is done on a task by task basis.

Intra-task DVS scheme: Voltage scheduling is done within a task boundary.

Inter-task EDF

- Static voltage scaling EDF
- Cycle conserving RT-DVS

Static voltage scaling: Example

Task set: T1 = (1, 4) and T2 = (2, 8)

U = 1/4 + 2/8 = 0.5 (< 1) @ F_{max}

What is the "k" at which the task set is still schedulable @ (F_{max} / k):

Let K = x

U = (1*x)/4 + (2*x)/8 = x*(0.5) = 1

X = 2, that is k = 2

Therefore, we can operate at f = F_{max} / 2 and still meet the deadlines

What if $C_i < WC_i$?

Then we have to reduce the time evenly and proportionally in order to get the needed deadlines to get fill.

Cycle conserving EDF: Example

Task set: T1 = (3, 6) and T2 = (6, 12)

U = 3/6 + 6/12 = 1 @ F_{max}

What is the "k" at which the task set is still schedulable @ (F_{max} / k):

Let K = x

U = (3*x)/6 + (6*x)/12 = x*(1.0) = 1

X = 1, that is k = 1

Therefore, we should operate at f = F_{max} in order to meet all the deadlines and this case we have to check the first deadlines taking place time and accordingly we have to adjust the value of energy and therefore we can derive the required k.

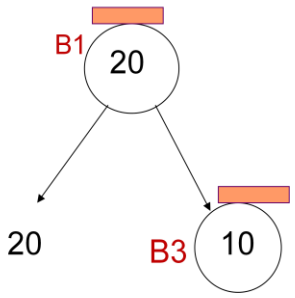
Intra Task Energy Management

A novel intra-task voltage-scheduling algorithm controls the supply voltage within an individual task boundary. By fully exploiting slack time, it achieves a high-energy reduction ratio. Using this algorithm, a software tool automatically converts an application into a low energy version. Intra-task DVS: adjusts the voltage and clock speed within a task. Identifies the slack time generated within a task due to workload variation. Application code is preprocessed to enable the run-time clock/voltage adjustment.

Intra-task RT-DVS

Intra-task DVS algorithms typically work with the control flow graph (CFG) of the real-time programs. Each node in the CFG denotes a basic block of computation. The edges in the CFG indicate the control dependency between the blocks.

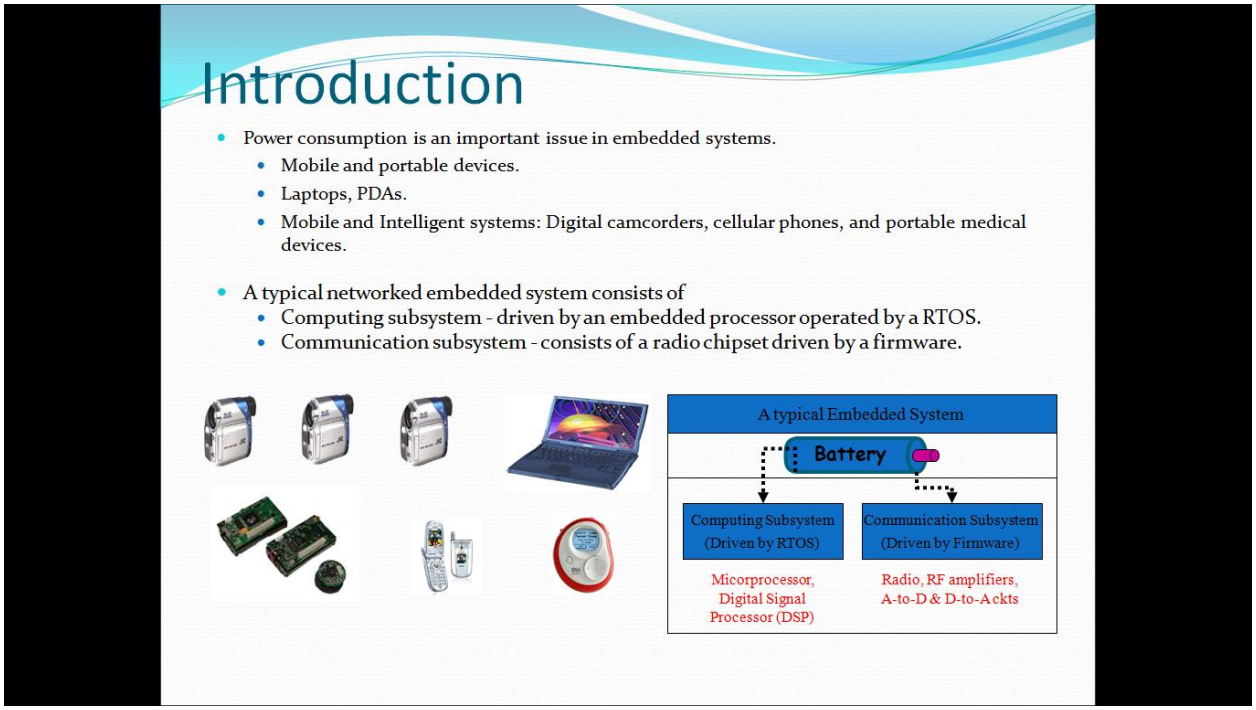
Objective is to assign proper clock frequency to each of the basic blocks so as to minimize the total energy consumption while meeting the task deadline.



Summary

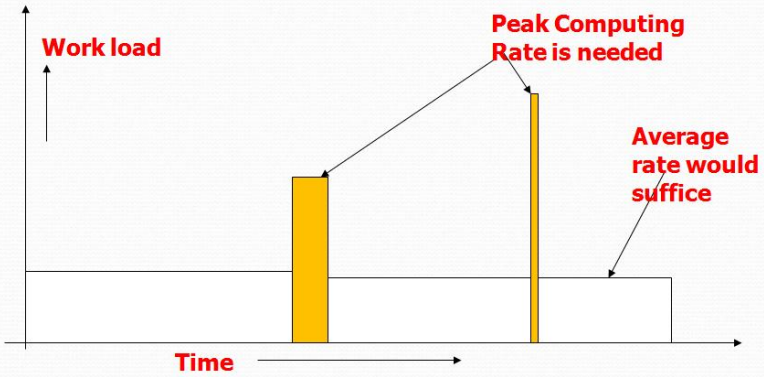
- DVS schemes can significantly reduce energy in embedded systems.
- Thus the power scheduling is important to handle and efficient handing is required.

The PowerPoint Presentation



Some Important Facts

- High performance is needed only for a small fraction of time, while for the rest of time, a low-performance, a low-power processor would suffice.



Continued..

- Processors are based on CMOS technology where dynamic power is the bottleneck

Dynamic power (due to switching activity)

- $P \alpha V^2 . f$
- $V \alpha f$ V: voltage; P: power; E: Energy

- $E = P * Tcc$ $Tcc = CC/f$
- $E_i = K .cc_i . f^2$

Where Tcc : execution time;

CC_i : # clock cycles of task T_i

f : frequency at which T_i is run.

Variable Voltage Processors

- Modern processors operate at multiple frequency levels.
  - Crusoe Processor: Transmeta Corporation
  - PowerNow! Technology: AMD
  - Intel XScale: Intel
- Higher the frequency level higher the energy consumption

Crusoe processor

- **What is the power consumption of a Crusoe processor?**

  The extremely low power consumption delivered on multimedia applications is due to a new feature called LongRun power management. LongRun has the distinct ability to analyze the application workload dynamically and to adjust continuously the processor's speed (MHz) and voltage to provide the necessary performance. This new feature promises to extend the battery life of all applications, most specifically those requiring the constant attention of the processor. This is a dramatic departure from today's ultra-light PCs, which are incapable of delivering over one and a half or two hours of runtime for DVD movies.

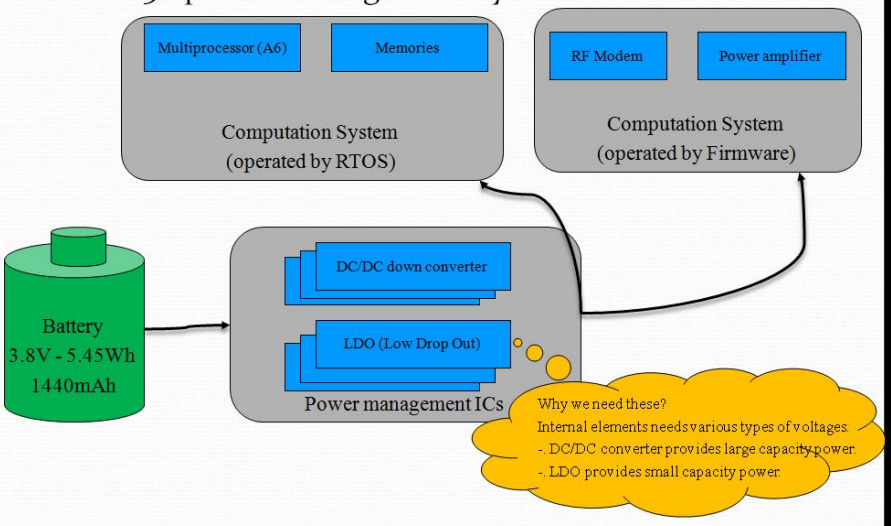
Dynamic Voltage Scaling (DVS)

- DVS scales the operating voltage of the processor along with the frequency.

- Since energy is proportional to f^2 , DVS can potentially provide significant energy savings through frequency and voltage scaling.

Case study (iPhone 5)

- iPhone 5's power management system



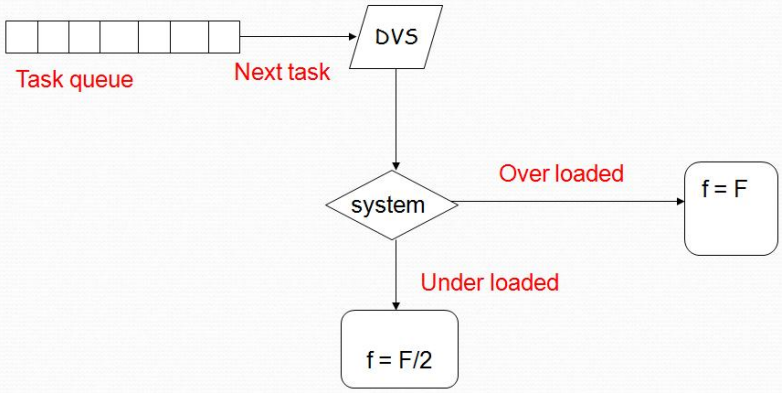
Case study (smart phones)

- Practical multi-core processors

Device	Chip maker	Processor name	Frequency	# cores
iPhone 5	Apple	A6	1.02GHz	2
Galaxy S III	Samsung	Exynos 4412	1.44GHz	4
Motorola RAZR	Ti	OMAP 4430	1.2GHz	2
HTC one S	Qualcomm	MSM8260A	1.2GHz	2
Asus transformer	NVIDIA	Tegra 3	1.3GHz	4
New iPad	Apple	A5X	1 GHz	2

- Contemporary multi-core processors have more than 2 cores at about 1 GHz.

Simple DVS-Scheme



DVS-example

- Consider a task with a computation time 20 units.
- Energy of T_i without DVS:
  - $E1 = K * 20 * F^{2}$

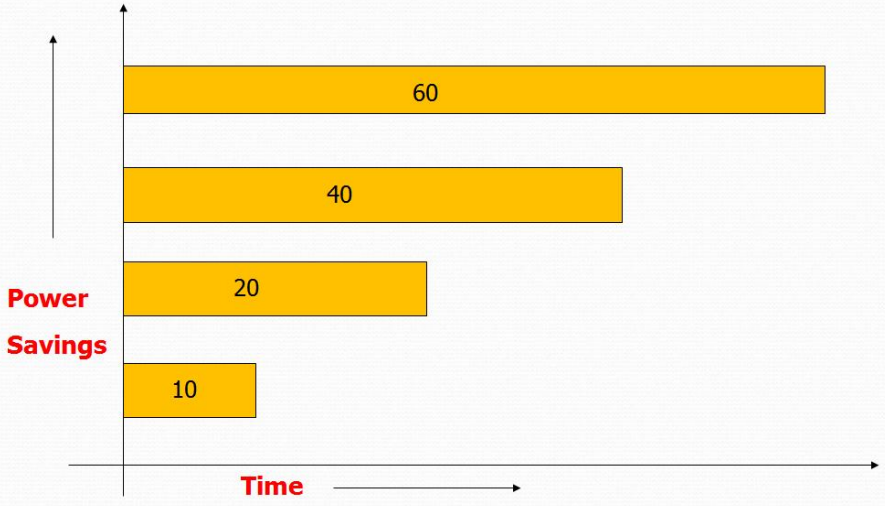
Time taken = t1 (say)

- Energy of T_i with DVS:
  - $E2 = K * 20 * (F/2)^{2}$
- Clearly, $E2 = (E1)/4$

Time taken = t2 = 2 * t1

Therefore, if we reduce the frequency we save energy but, we spend more time in performing the same computation

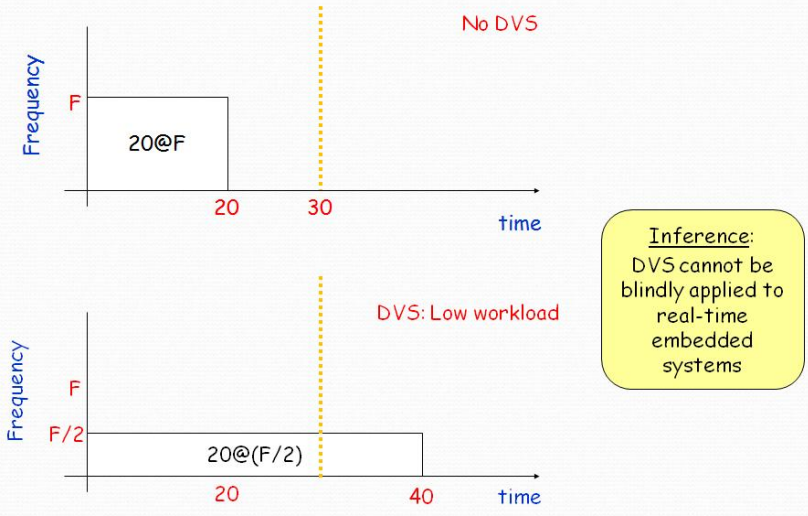
Power-Time Tradeoffs



Simple DVS scheme handling RT-task

- Consider a real-time task T1 = (20, 30)
- Applying the simple DVS scheme
  - T1 runs at maximum frequency (F) and meets the deadline with no power savings
  - T1 runs at half the maximum frequency (F/2) and completes at time = 40 thereby missing its deadline

Simple DVS scheme handling RT-task



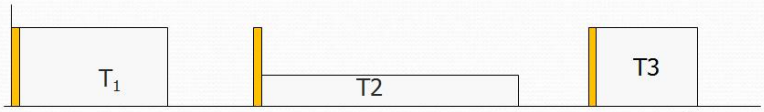
Inference:
DVS cannot be blindly applied to real-time embedded systems

Energy aware scheduling in RT Systems

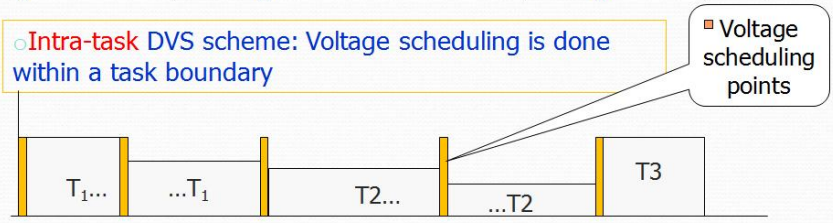
- Objectives
  - Minimizing energy consumption
  - Meeting the deadlines

Real Time - DVS schemes

- The RT-DVS algorithms can be broadly classified based on the granularity at which voltage scheduling is performed as follows
  - Inter-task DVS scheme: Voltage scheduling is done on a task by task basis.



  - Intra-task DVS scheme: Voltage scheduling is done within a task boundary



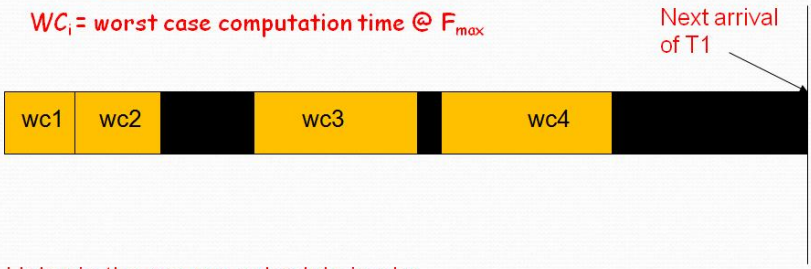
Inter-task EDF

- Static voltage scaling EDF
- Cycle conserving RT-DVS

Static Voltage Scaling EDF: Motivation

Pre-run schedule with holes

WC_i = worst case computation time @ F_{max}



Holes in the pre-run schedule imply:

EDF Test:

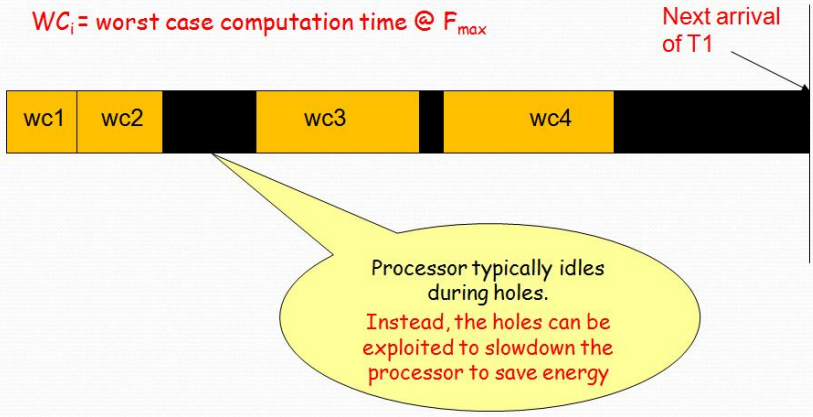
$$\sum(wc_i/p_i) < 1 \quad \text{at frequency} = F_{max}$$

In other words, whenever $\sum(wc_i/p_i) < 1$ there are holes in the EDF schedule

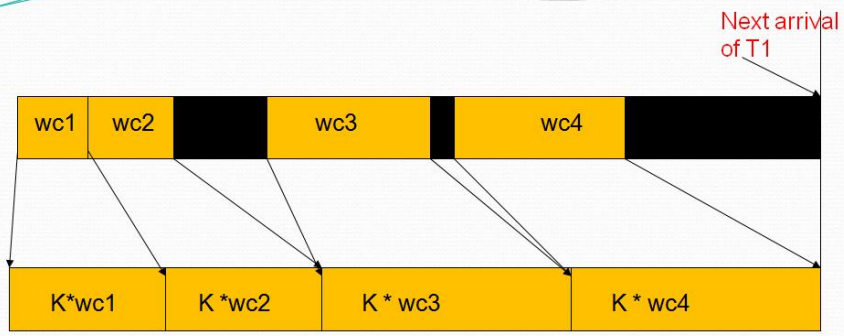
Static Voltage Scaling EDF: exploiting holes

Pre-run schedule with holes

WC_i = worst case computation time @ F_{max}



Static Voltage Scaling EDF



EDF Test:

$$\sum(wc_i/p_i) < 1 \quad \text{at maximum frequency} = F_{max}$$

Static-VS EDF Test:

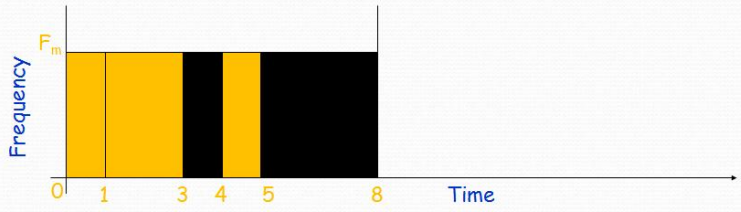
$$K^* [\sum(wc_i/p_i)] = 1 \qquad \text{at frequency} = F_{max}/K$$

Static voltage scaling: Example

- Task set: T1 = (1, 4) and T2 = (2, 8)
- U = 1/4 + 2/8 = 0.5 (< 1) @ F_{max}
- What is the "k" at which the task set is still schedulable @ (F_{max} / k):
  - Let K = x
  - U = (1*x)/4 + (2*x)/8 = x*(0.5) = 1
  - X = 2, that is k = 2
  - Therefore, we can operate at f = F_{max} / 2 and still meet the deadlines

Task set: T1 = (1, 4) and T2 = (2, 8)
U = 1/4 + 2/8 = 0.5 (< 1) @ Fmax

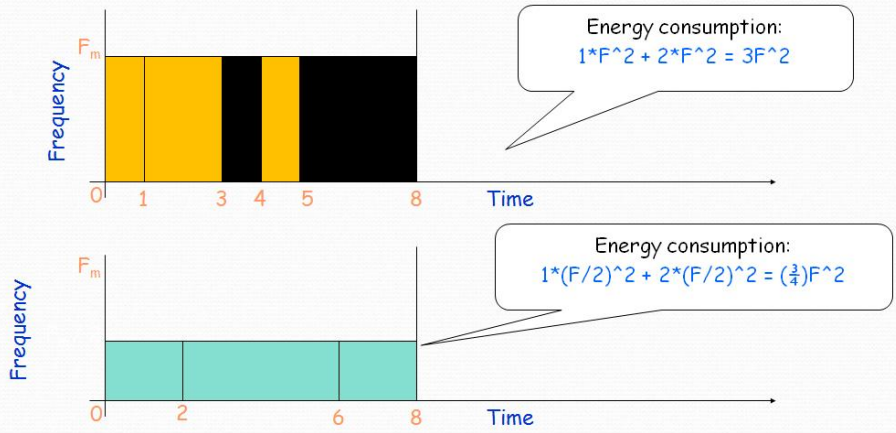
Static voltage scaling: Example



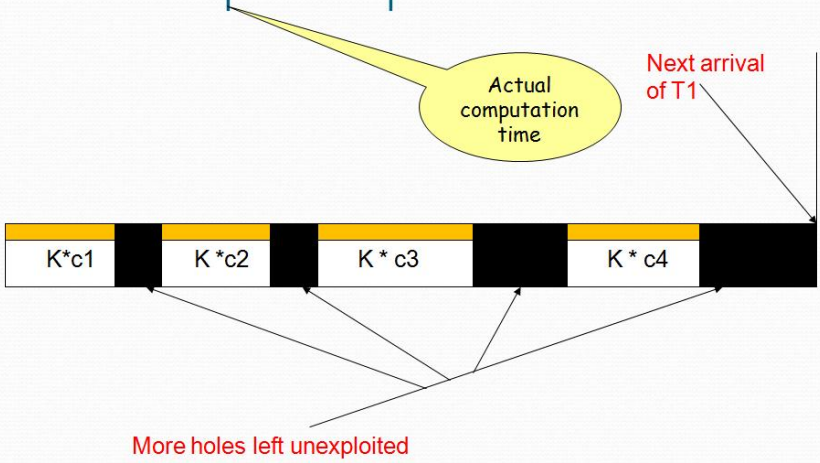
Finding the right frequency scaling parameter (say, k)
U = (1*k)/4 + (2*k)/8 = 0.5*k = 1 @ (F_{max}/k)
This gives, k = 2. Therefore, operating frequency = F_{max}/2

Modified Task set @ (Fmax/2): T1 = (2, 4) and T2 = (4, 8)
U = 2/4 + 4/8 = 1 @ (Fmax/2)

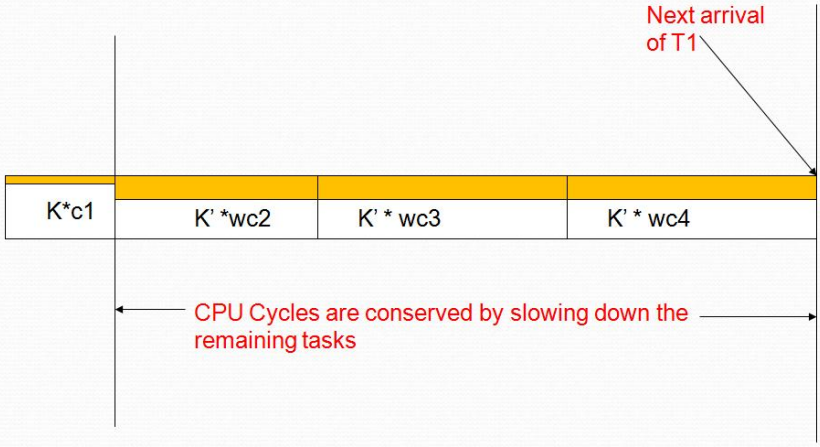
Static voltage scaling: Example



What if $C_i < WC_i$?



What if $C_i < WC_i$? (contd..)

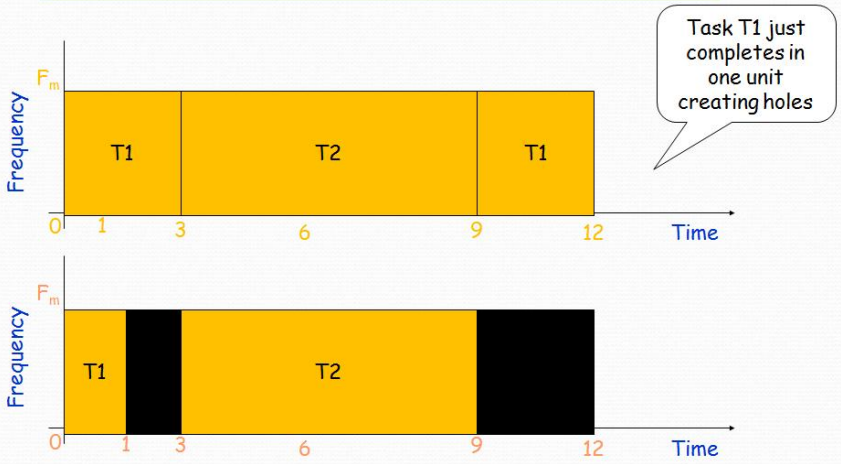


Cycle conserving EDF: Example

- Task set: T1 = (3, 6) and T2 = (6, 12)
- U = 3/6 + 6/12 = 1 @ F_{max}
- What is the "k" at which the task set is still schedulable @ (F_{max} / k):
  - Let K = x
  - U = (3*x)/6 + (6*x)/12 = x*(1.0) = 1
  - X = 1, that is k = 1
  - Therefore, we should operate at f = F_{max} in order to meet all the deadlines

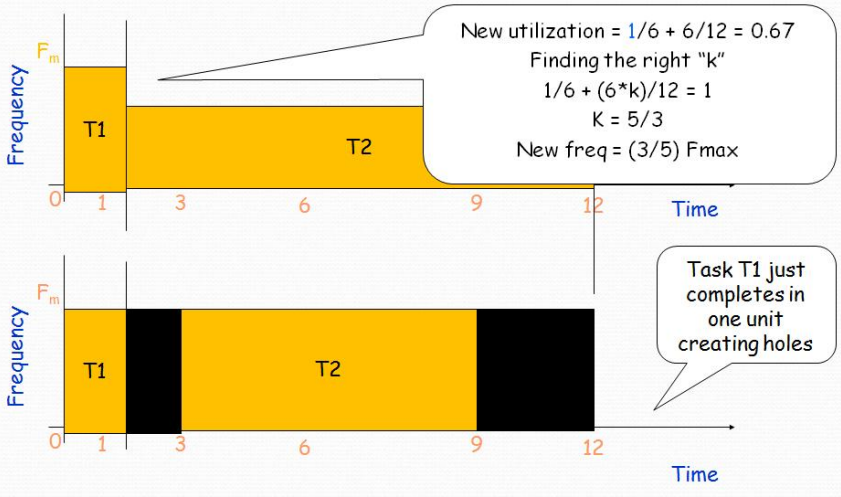
Cycle conserving EDF: Example

Task set @ (Fmax): T1 = (3,6) and T2 = (6,12)
U = 3/6 + 6/12 = 1 @ (Fmax)



Cycle conserving EDF: Example

Task set @ (Fmax): T1 = (3,6) and T2 = (6,12)
U = 3/6 + 6/12 = 1 @ (Fmax)



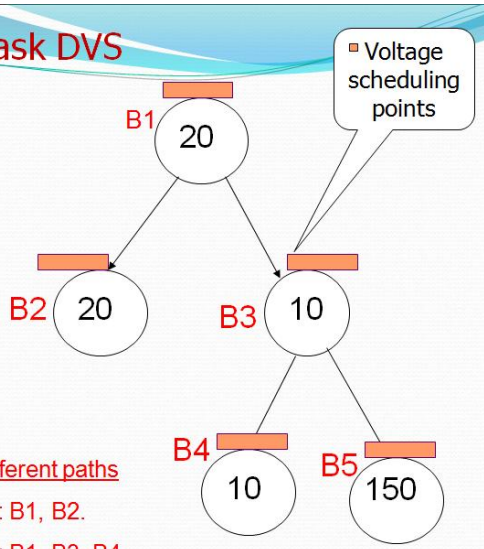
Intra Task Energy Management

- Intra-task DVS: adjusts the voltage and clock speed within a task.
- Identifies the slack time generated within a task due to workload variation.
- Application code is preprocessed to enable the run-time clock/voltage adjustment.

Intra-task DVS

Intra-task RT-DVS

- ✓Intra-task DVS algorithms typically work with the control flow graph (CFG) of the real-time programs.
- ✓Each node in the CFG denotes a basic block of computation.
- ✓ The edges in the CFG indicate the control dependency between the blocks.
- ✓Objective is to assign proper clock frequency to each of the basic blocks so as to minimize the total energy consumption while meeting the task deadline



Different paths

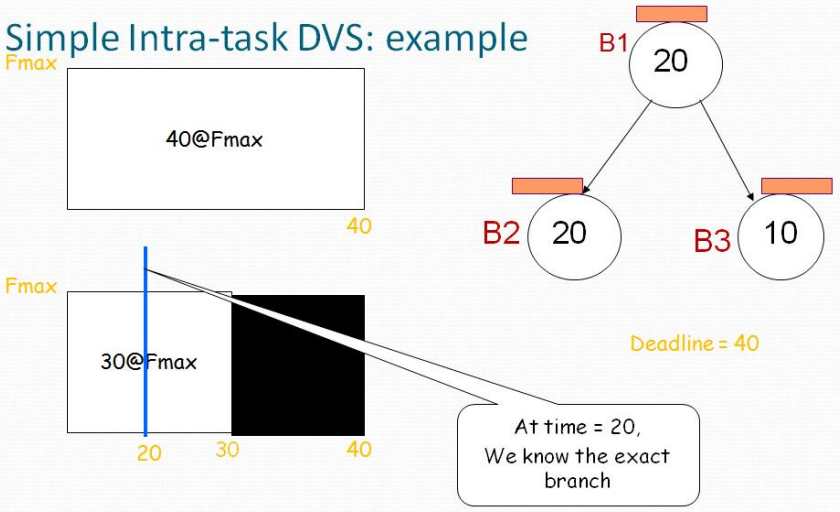
P1: B1, B2.

P2: B1, B3, B4.

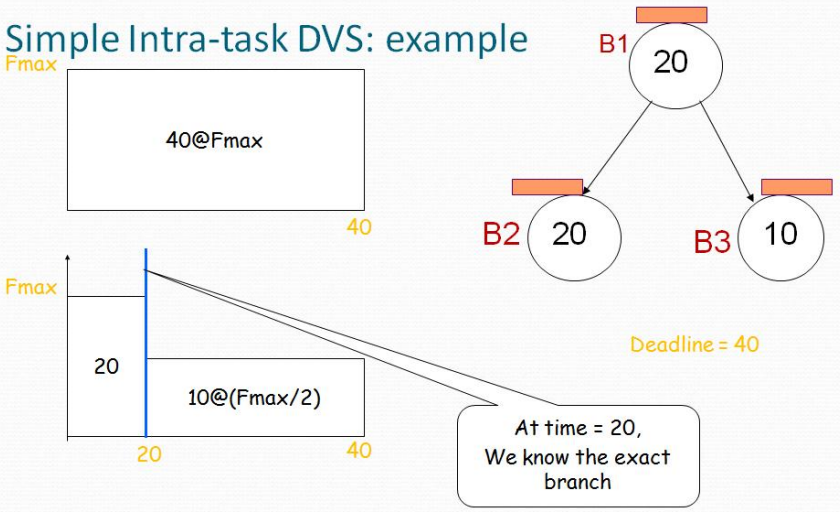
P3: B1, B3, B5.

Deadline = 200

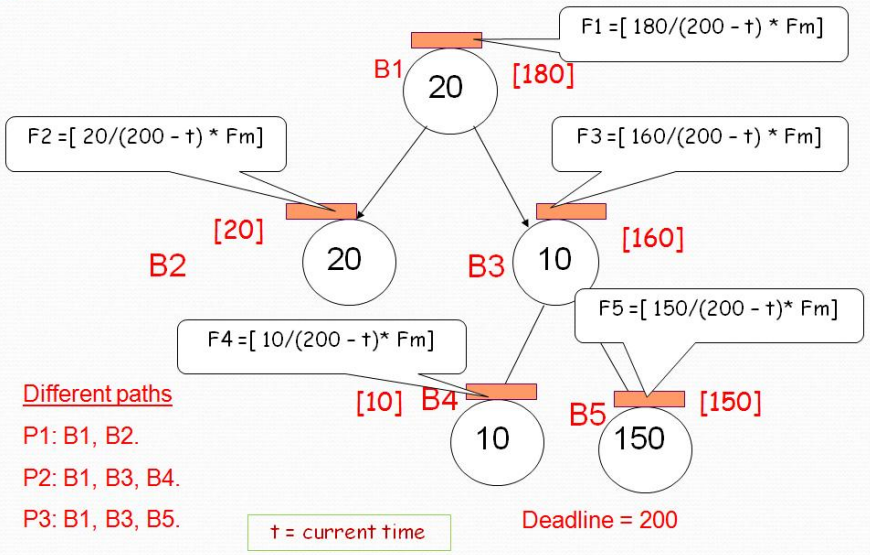
Simple Intra-task DVS: example



Simple Intra-task DVS: example



Worst Case Execution Time (WCET) based scheme



Summary

- DVS schemes can significantly reduce energy in embedded systems.
- Thus the power scheduling is important to handle and efficient handing is required.

----------XXXX----------