Minimizing Energy Consumption in Real-Time Systems

Enrico Bini

Scuola Superiore Sant'Anna

Pisa, Italy

e.bini@sssup.it

Acknowledgements

Daniel Mossé

Full Professor
University of Pittsburg,
Pittsburg (NY), U.S.A.

Hakan Aydin

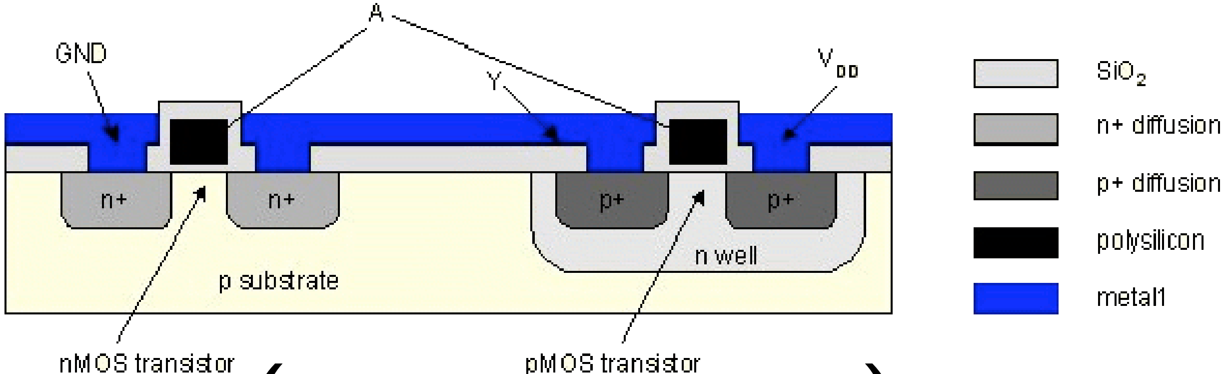
Associate Professor
George Mason University,
Fairfax (VA), U.S.A.

Mani Srivastava

Full Professor
University of California (UCLA)
Los Angeles (CA), U.S.A.

Contents

Energy Consumption in CMOS



$$p = N_{\mathrm{sw}}\left(C_L V_{DD}^2 + I_{\mathrm{sw}} V_{DD}\right) f + I_{\mathrm{leak}} V_{DD}$$

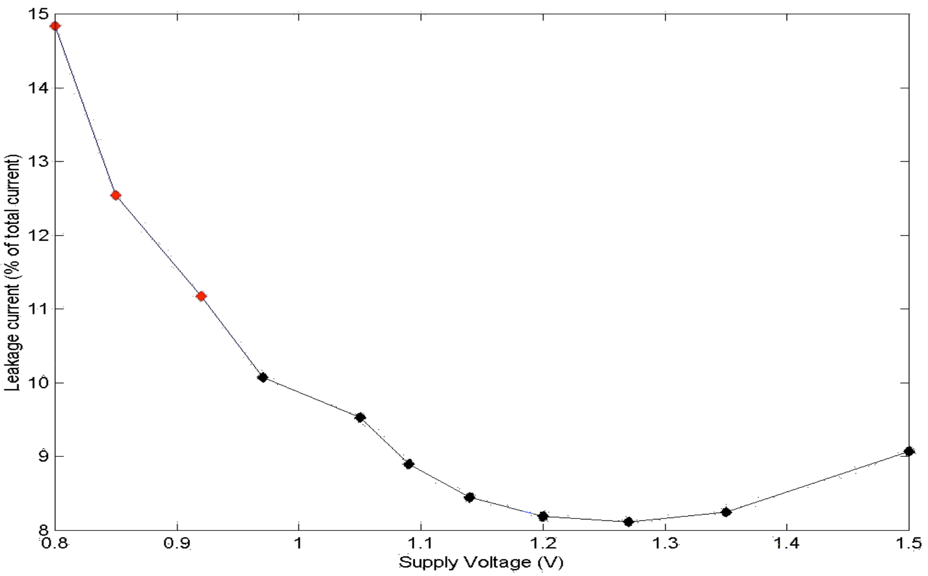
- N_{sw}, average # switches per cycle
- C_L, capacitance of single transistor
- V_{DD}, supply voltage
- f, clock frequency
- I_{sw}, short circuit current when logic level changes
- I_{leak}, leakage current

Static and Dynamic Power

$$p = \underbrace{N_{sw}\left(C_L V_{DD}^2 + I_{sw} V_{DD}\right) f}_{\text{dynamic}} + \underbrace{I_{leak} V_{DD}}_{\text{static}}$$

Dynamic power:
- consumed due to logic level switch
- account for ~ 90% in current technology

Static power:
- always consumed
- increases with: low voltage, high temp., small feature size (<90 nm)



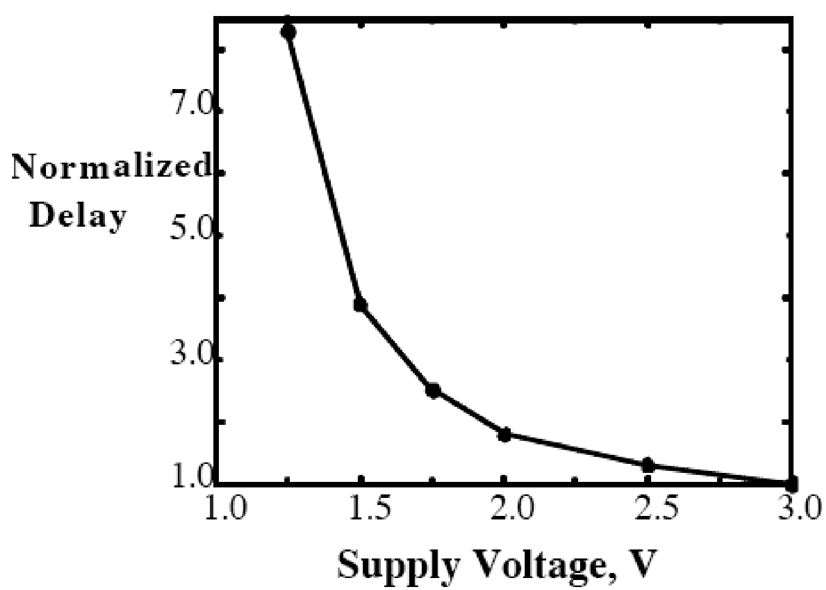
Reducing V_{DD}?

Why can't we simply lower V_{DD}?

Because the delay Δ of one CMOS increases!

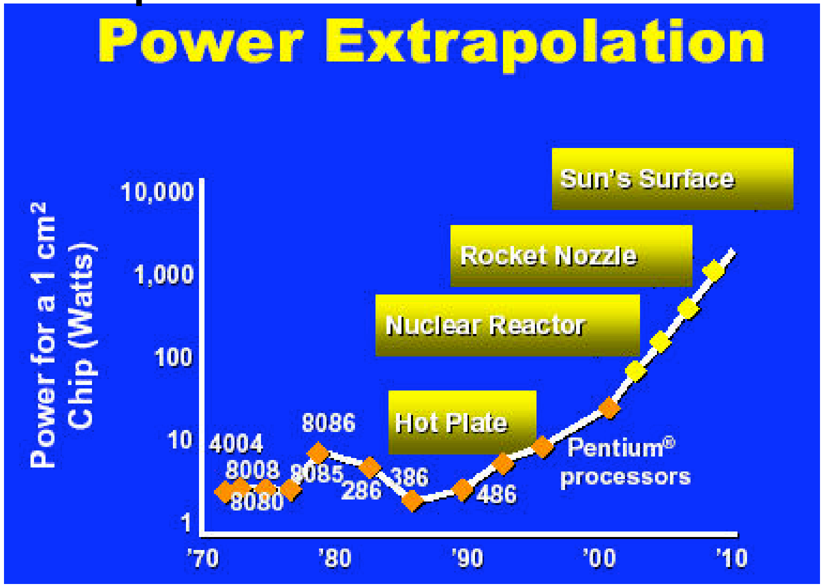
$$\Delta = \frac{C_L V_{DD}}{K\left(V_{DD} - V_T\right)^{\alpha}}$$

K, $\alpha \in [1,2]$, depend on the fabrication technology



Future trends

- Reducing V_{DD} requires reducing threshold voltage V_T as well
  - this increases leakage current I_{leak}
- Reducing the min. feature size (deep submicron technology)
  - increase the leakage current again, (static power, no longer negligible)
  - amazing power density (approaching nuclear reactor), advanced dissipation techniques required
- Robustness isseus
  - circuits are more sensitive to alpha particles, cosmic rays: increased probability of bit flipping. Robustness to soft errors.



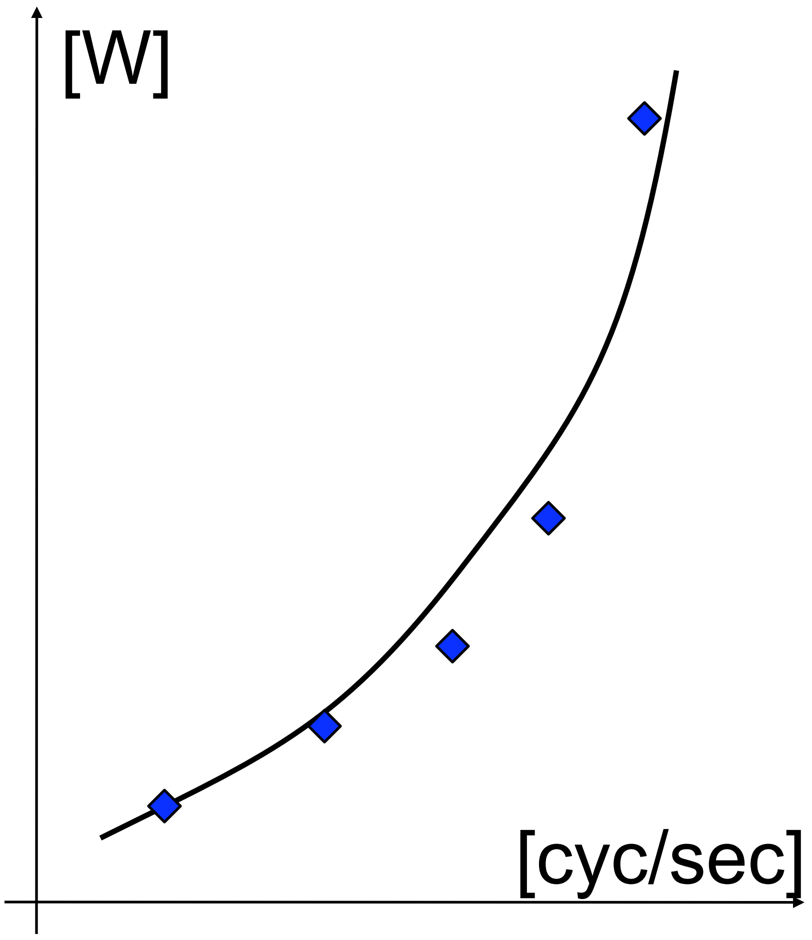
The CPU power/speed model

Continuous:

- power/speed function, polynomial
- convex relationship!

Discrete:

- power/speed modes from datasheet
- integer optimization

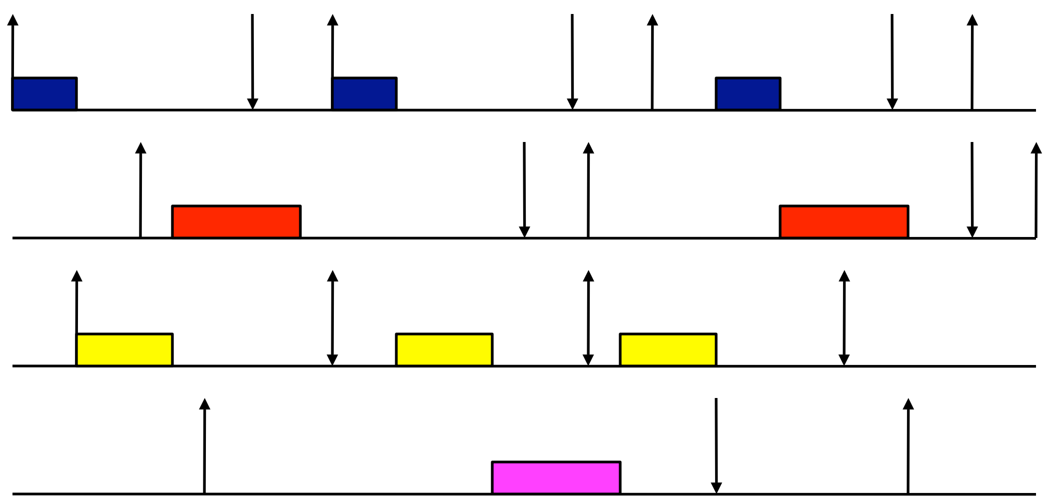


Contents

Simple real-time task model

- Worst-case execution cycles (WCEC) C_i
- Period (or min interarrival time) P_i
- Deadline D_i
- Real-time application composed by n tasks $\tau_1,\ldots,\tau_n$

Example of EDF schedule



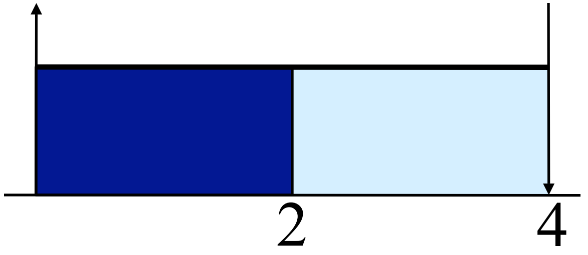
Real-Time + Power-aware

Suppose: $p \propto f^2$ $C_1 = 2, T_1 = D_1 = 4$

Real-Time + Power-aware

Suppose: $p \propto f^2$ $C_1 = 2, T_1 = D_1 = 4$

No technique

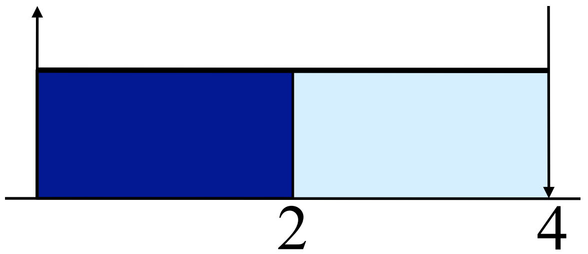


E

Real-Time + Power-aware

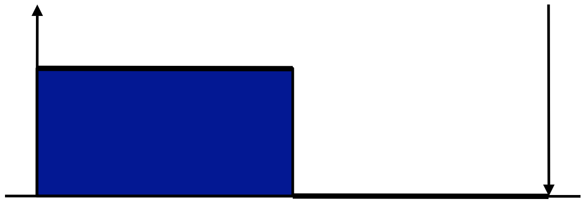
Suppose: $p \propto f^2$ $C_1 = 2, T_1 = D_1 = 4$

No technique



E

Shut down

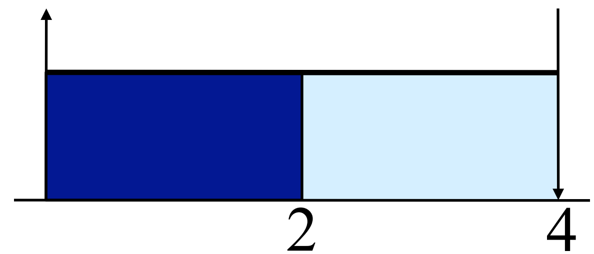


$E/2$

Real-Time + Power-aware

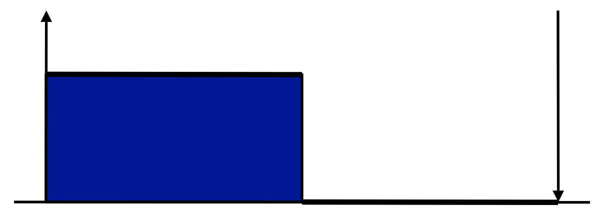
Suppose: $p \propto f^2$ $C_1 = 2, T_1 = D_1 = 4$

No technique



E

Shut down



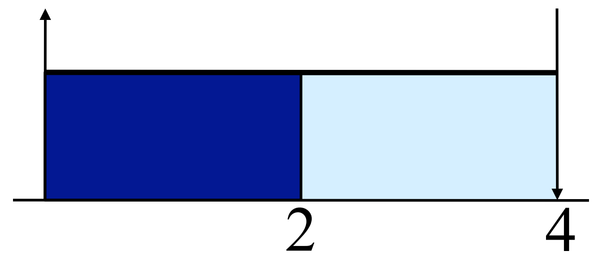
$E/2$

Slow down

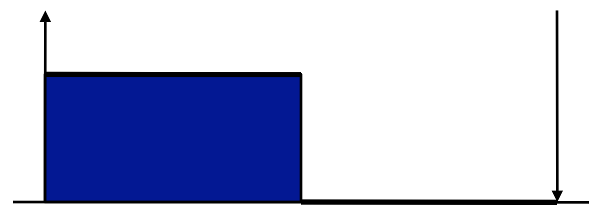
$E/4$

Real-Time + Power-aware

Suppose: $p \propto f^2$ $\quad C_1 = 2, T_1 = D_1 = 4$

No technique 

E

Shut down  $E/2$

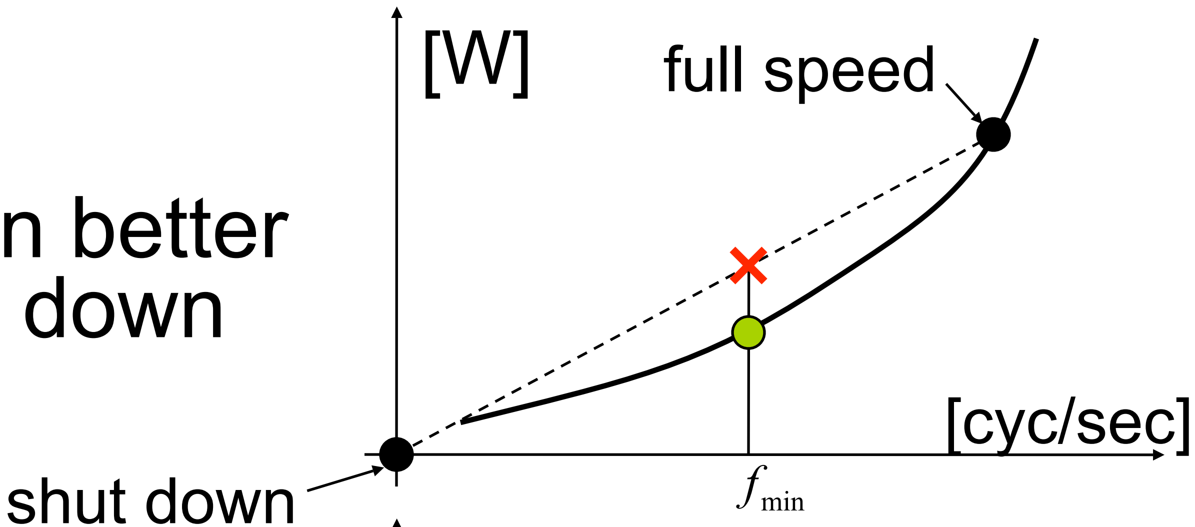
Slow down  $E/4$

Slow down, more convenient because f^2 is **convex!!**

Consequences of convexity

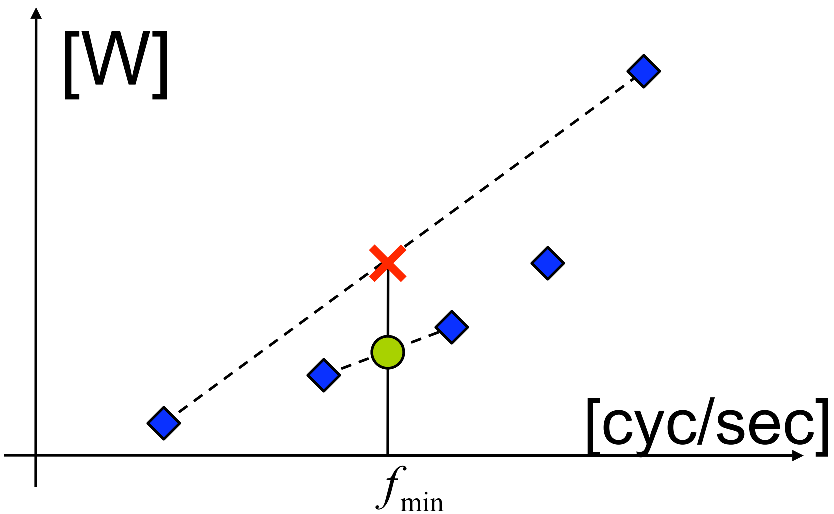
Continuous:

slow down better than shut down



Discrete:

two closest modes are best



Finding the min speed: EDF

Task set is feasible at speed f **if and only if**

$$\forall \text{deadline } d \quad \sum_{i=1}^{n} \max\left\{0, \left\lfloor \frac{d - D_i + P_i}{P_i} \right\rfloor\right\} \frac{C_i}{f} \leq d$$

Hence the minumum feasible speed $f_{\min}$ is

$$f_{\min} = \max_{d \text{ is deadline}} \frac{1}{d} \sum_{i=1}^{n} \max\left\{0, \left\lfloor \frac{d - D_i + P_i}{P_i} \right\rfloor\right\} C_i$$

If all $D_i = P_i$ then

$$f_{\min} = \sum_{i=1}^{n} \frac{C_i}{P_i}$$

Finding the min speed: FP

Task set is feasible <u>at speed f</u> **if and only if**:

$$\forall i \quad \exists t \in \text{schedP}_i \quad C_i + \sum_{j=1}^{i-1} \left\lceil \frac{t}{P_j} \right\rceil \frac{C_j}{f} \le t$$

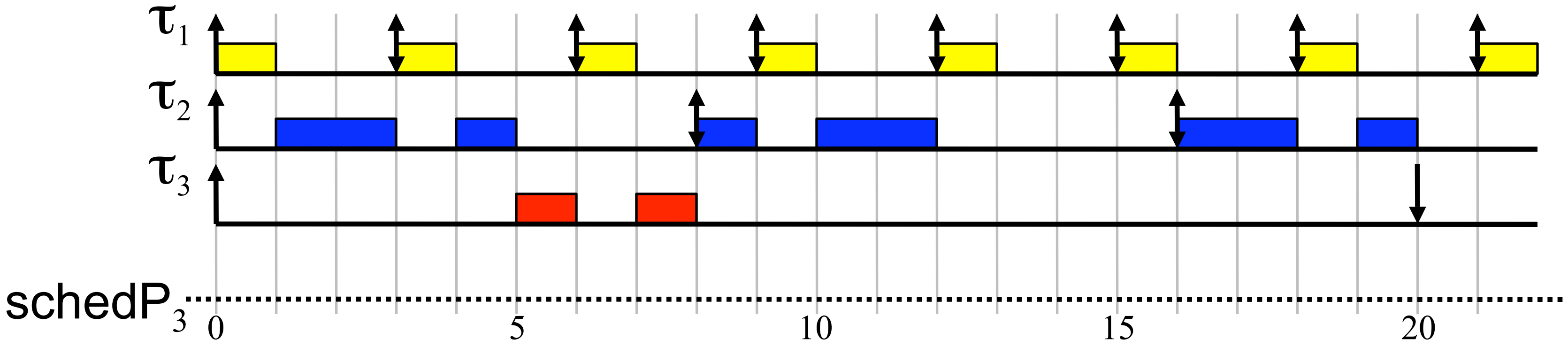
Set of *scheduling points* for task τ_i

Hence the minumum feasible speed f_{min} is:

$$f_{\min} = \max_i \min_{t \in \text{schedP}_i} \frac{C_i + \sum_{j=1}^{i-1} \lceil t/P_j \rceil C_j}{t}$$

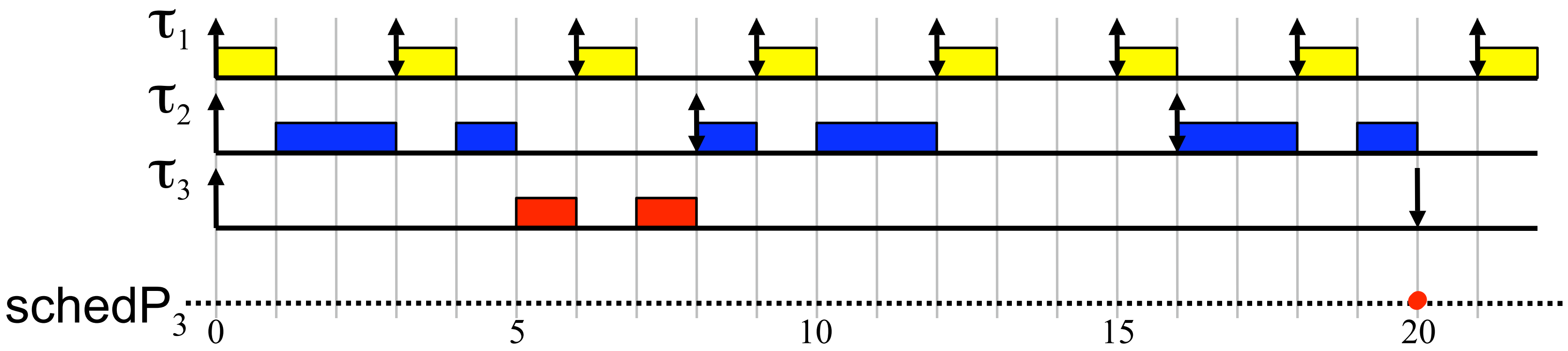
FP: the scheduling points

schedP$_i$: set of *"some"* activations before D_i



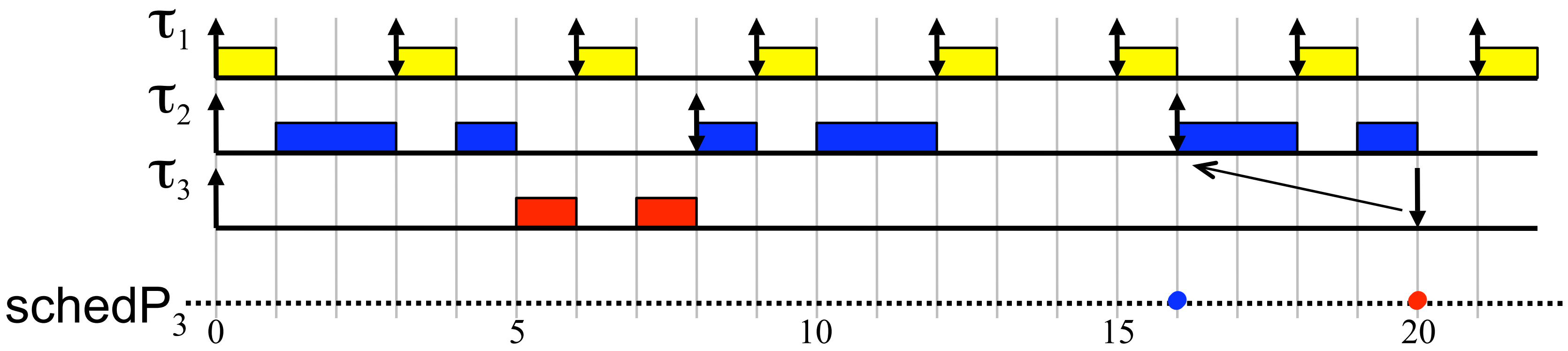
FP: the scheduling points

schedP$_i$: set of *"some"* activations before D_i



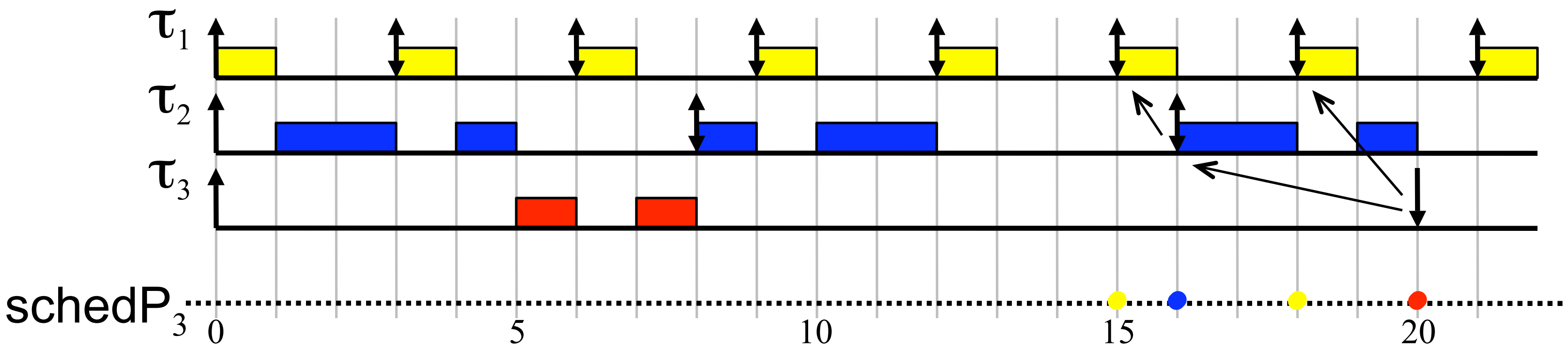
FP: the scheduling points

schedP$_i$: set of *"some"* activations before D_i



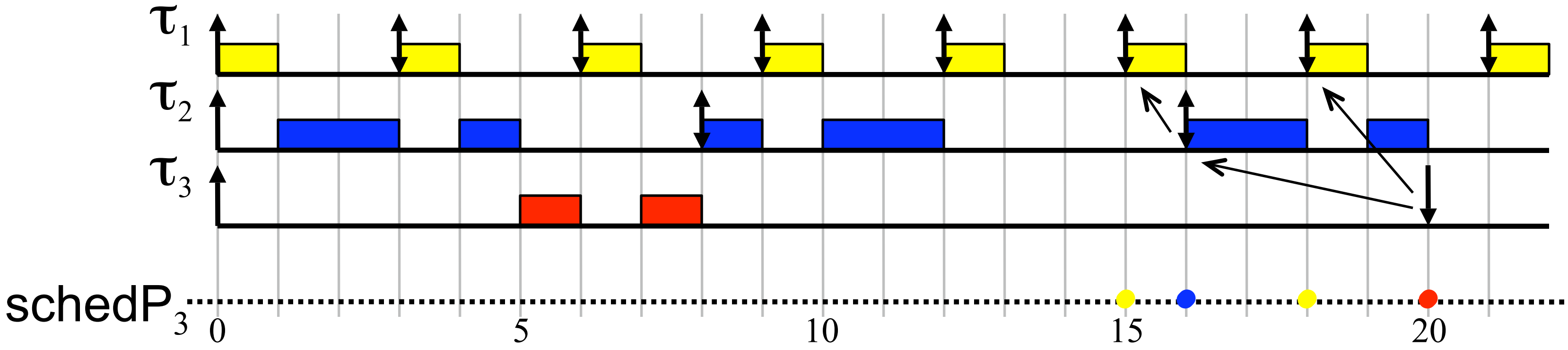
FP: the scheduling points

schedP$_i$: set of *"some"* activations before D_i



FP: the scheduling points

schedP$_i$: set of *"some"* activations before D_i



$$\text{schedP}_i = \mathcal{P}_{i-1}(D_i)$$

$$\begin{cases} \mathcal{P}_0(t) = \{t\} \\ \mathcal{P}_i(t) = \mathcal{P}_{i-1}\left(\left\lfloor \frac{t}{T_i} \right\rfloor T_i\right) \cup \mathcal{P}_{i-1}(t) \end{cases}$$

Sufficient conditions

Using sufficient conditions results in upper bounds of the min speed

In FP, from Liu and Layland bound

$$\sum_{i=1}^{n} \frac{C_i}{P_i} \le n\left(\sqrt[n]{2} - 1\right) \;\Rightarrow\; f_{\mathrm{LL}} = \frac{1}{n(\sqrt[n]{2} - 1)} \sum_{i=1}^{n} \frac{C_i}{P_i}$$

Clearly $f_{\mathrm{LL}} \ge f_{\min}$

Comp. times don't scale

We have assumed that

"The time required to execute a fixed amount of cycles is proportional to the CPU speed f"

However

"Operations involving I/O devices, memory have a fixed duration in time"

Hence the required cycles are $$C_i = c_i + m_i f$$

- c_i, cycles of the scalable part of the task [cyc]
- m_i, duration of the non-scalable part [sec]

and the task duration is $$\frac{c_i}{f} + m_i$$

Exercise

$C_1=1, P_1=4, C_2=1, P_2=6, D_i=P_i$

1. What is the min speed with EDF? What if $D_1=3$?
2. What is the min speed with FP?
3. What is f_{LL}?
4. What is the min speed using the Hyperbolic Bound Test?

$$\prod_{i=1}^{n}\left(1+\frac{C_i}{P_i}\right)\leq 2$$

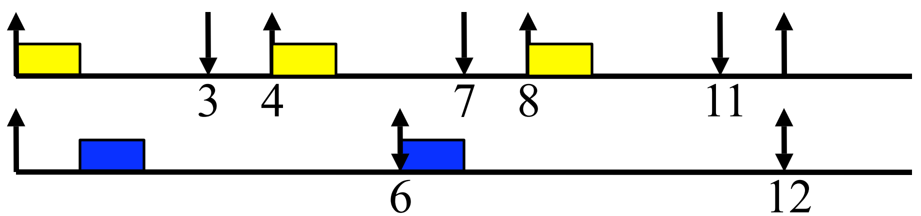
Solution

1. What is the min speed with EDF?

If $D_i = P_i$,

$$f_{\text{EDF}} = \frac{C_1}{P_1} + \frac{C_2}{P_2} = \frac{5}{12} \cong 0.41667$$

If $D_1 = 3$, then deadline are $\{3,6,7,11,12\}$



[redacted]

Solution

2. What is the min speed with FP?

$$\text{schedP}_1 = \mathbf{P}_0(D_1) = \{4\}$$

$$\text{schedP}_2 = \mathbf{P}_1(D_2) = \mathbf{P}_0\left(\left\lfloor \frac{D_2}{P_1} \right\rfloor P_1\right) \cup \mathbf{P}_0(D_2) = \{4,6\}$$

$$f_{\text{FP}} = \max\left\{\frac{1}{4}, \min\left\{\frac{2}{4}, \frac{3}{6}\right\}\right\} = 0.5$$

3. What is f_{LL}?

$$f_{\text{LL}} = \frac{1}{2(\sqrt{2}-1)}\frac{5}{12} = \frac{5\left(\sqrt{2}+1\right)}{24} \cong 0.50296$$

Solution

4. What is the min speed using the Hyperbolic Bound Test?

$$\left(1+\frac{C_1}{P_1 f}\right)\left(1+\frac{C_2}{P_2 f}\right)\leq 2$$

$$(P_1 f + C_1)(P_2 f + C_2) \leq 2P_1P_2 f^2$$

$$P_1P_2 f^2 - (C_1P_2 + C_2P_1) f - C_1C_2 \geq 0$$

$$f \geq \frac{C_1P_2 + C_2P + \sqrt{(C_1P_2 + C_2P_1)^2 + 4C_1C_2P_1P_2}}{2P_1P_2} = f_{\mathrm{HB}}$$

$$C_1 = C_2 = 1, P_1 = 4, P_2 = 6, D_i = P_i \quad \Rightarrow \quad f_{\mathrm{HB}} = 0.5$$

Contents

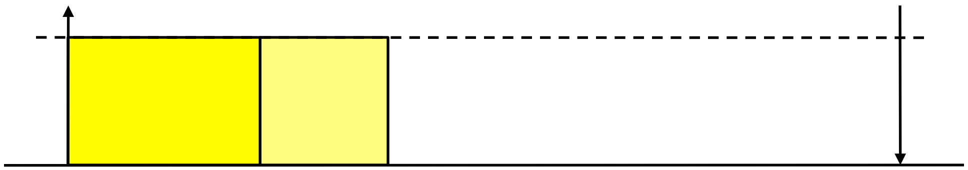
Early completion of tasks

min speed based on WCECs

Early completion of tasks

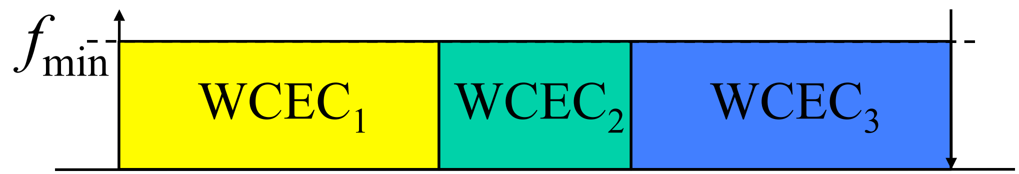
min speed based on WCECs

Worst-case conditions occur very rarely, when tasks execute run for less than WCEC



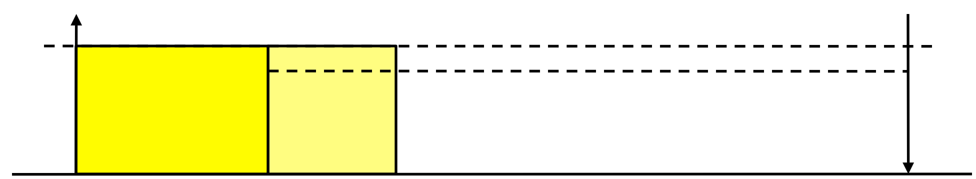
Early completion of tasks

min speed based on WCECs



Worst-case conditions occur very rarely, when tasks execute run for less than WCEC

Tasks reclaim unused cycles

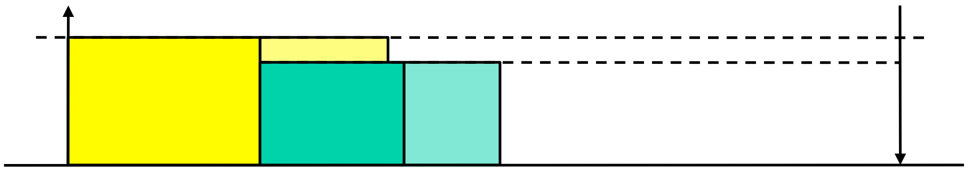


Early completion of tasks

min speed based on WCECs

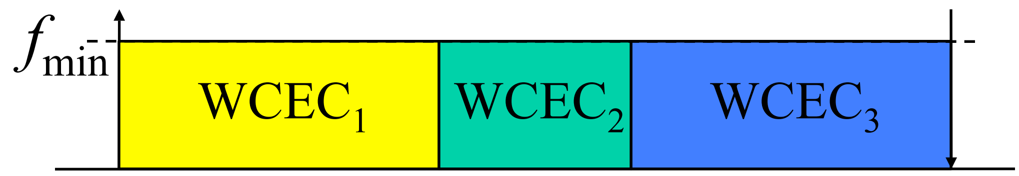
Worst-case conditions occur very rarely, when tasks execute run for less than WCEC

Tasks reclaim unused cycles



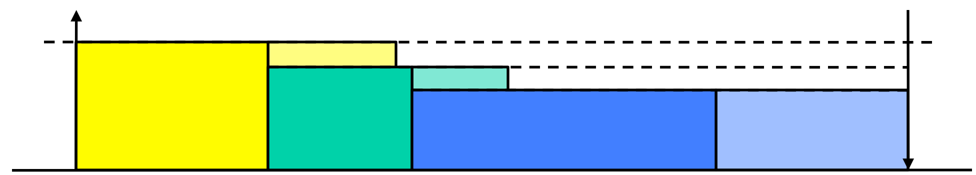
Early completion of tasks

min speed based on WCECs



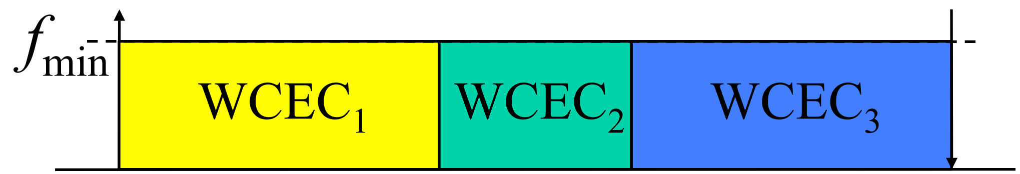
Worst-case conditions occur very rarely, when tasks execute run for less than WCEC

Tasks reclaim unused cycles



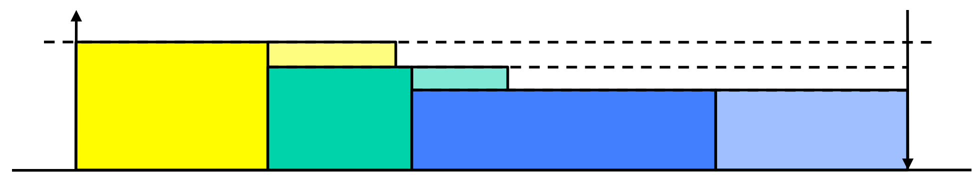
Early completion of tasks

min speed based on WCECs



Worst-case conditions occur very rarely, when tasks execute run for less than WCEC

Tasks reclaim unused cycles



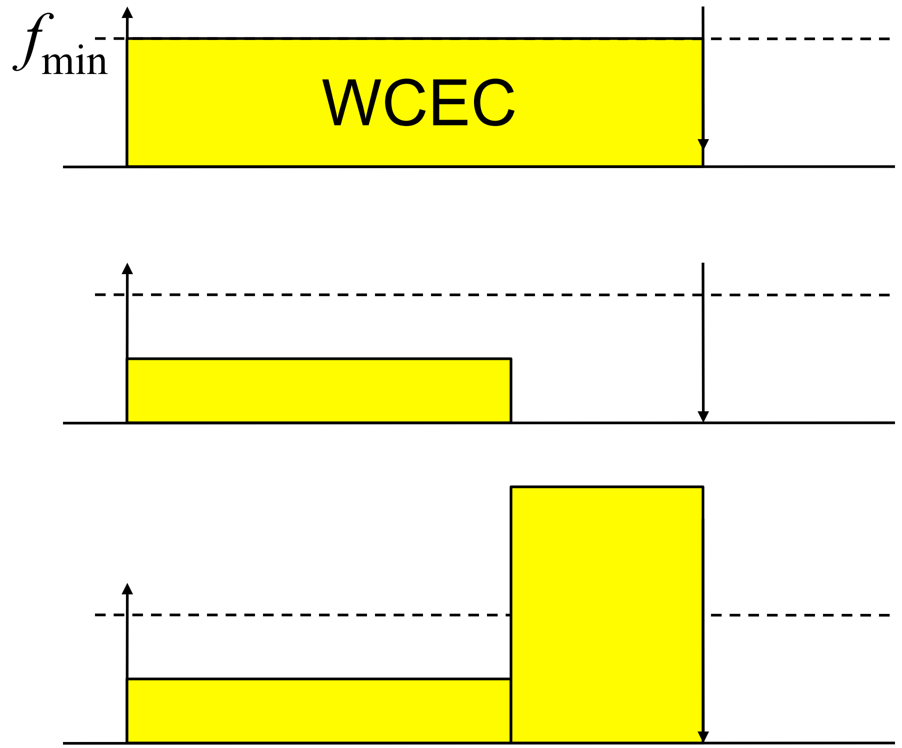
Special care must be taken to avoid bandwidth violation.

Aggressive schemes

min speed based on WCECs

assume speculatively much less cycles

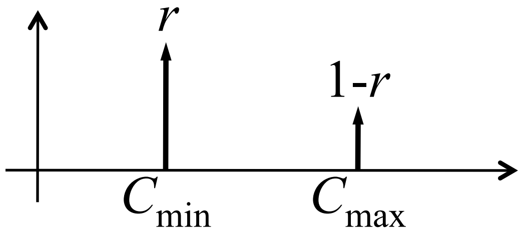
If the worst case occur increase speed



If probability(worst-case) very small, striking reduction of energy!!

Exploiting probability

p.d.f.=2 Dirac's delta



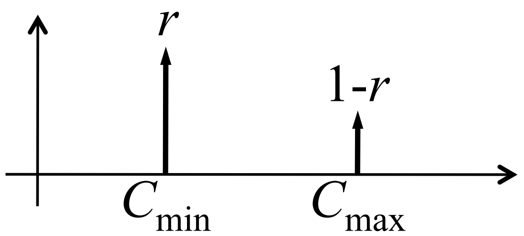
Adopting a 2-speeds scheme

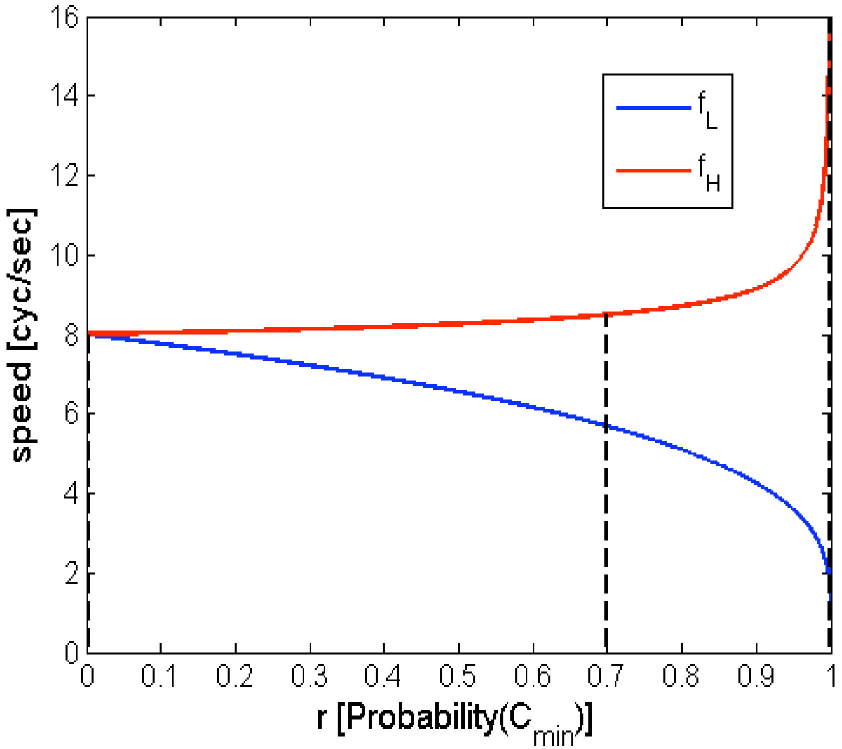
C_{min} cycles at f_L

C_{max}-C_{min} cycles at f_H

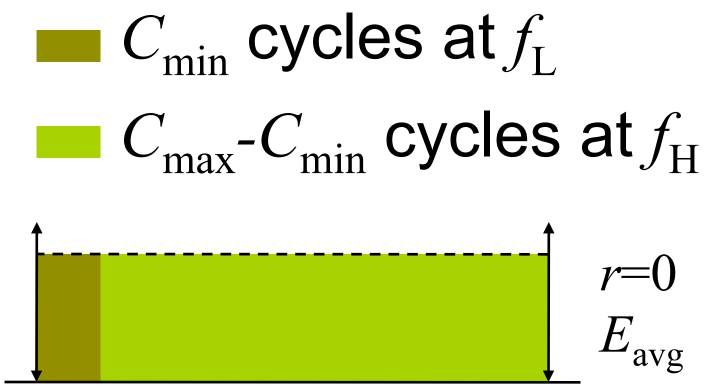
Exploiting probability

p.d.f.=2 Dirac's delta



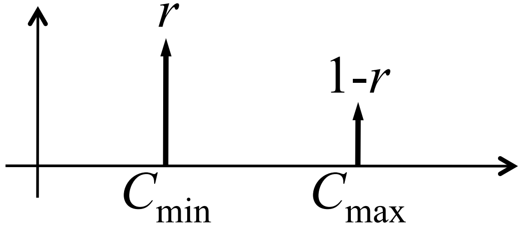


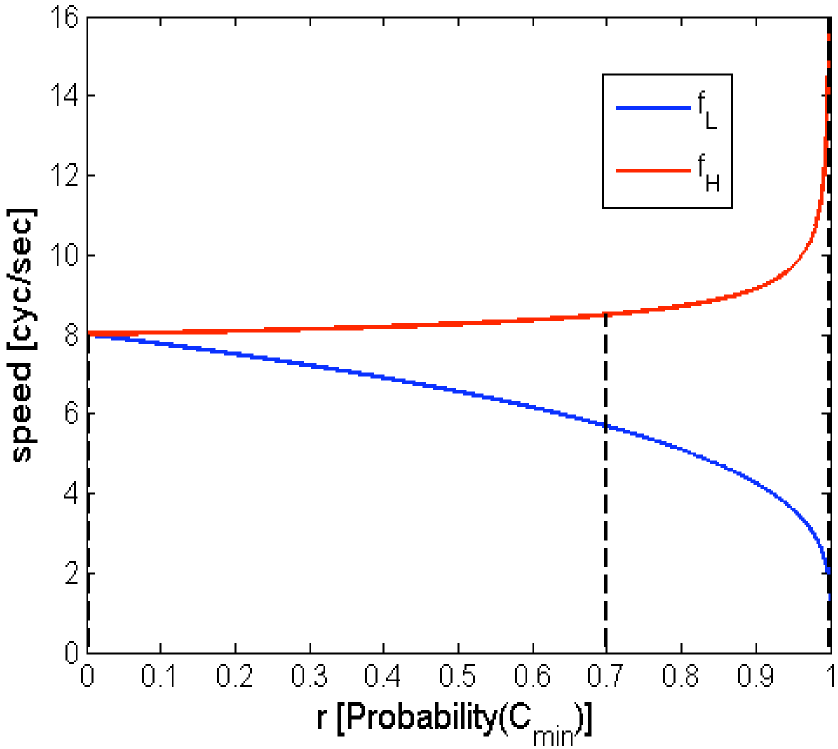
Adopting a 2-speeds scheme



Exploiting probability

p.d.f.=2 Dirac's delta

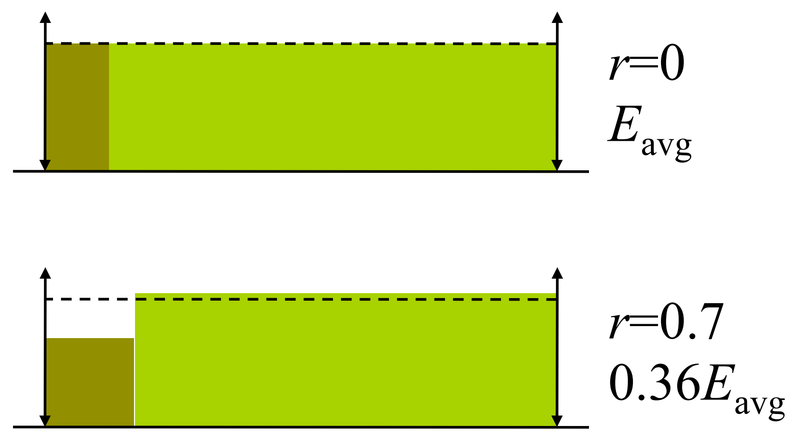




Adopting a 2-speeds scheme

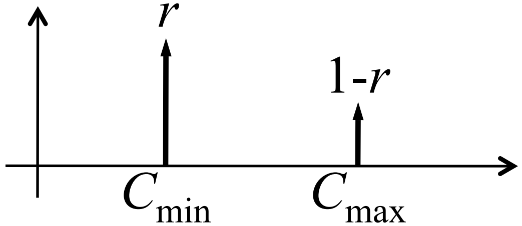
C_{min} cycles at f_L

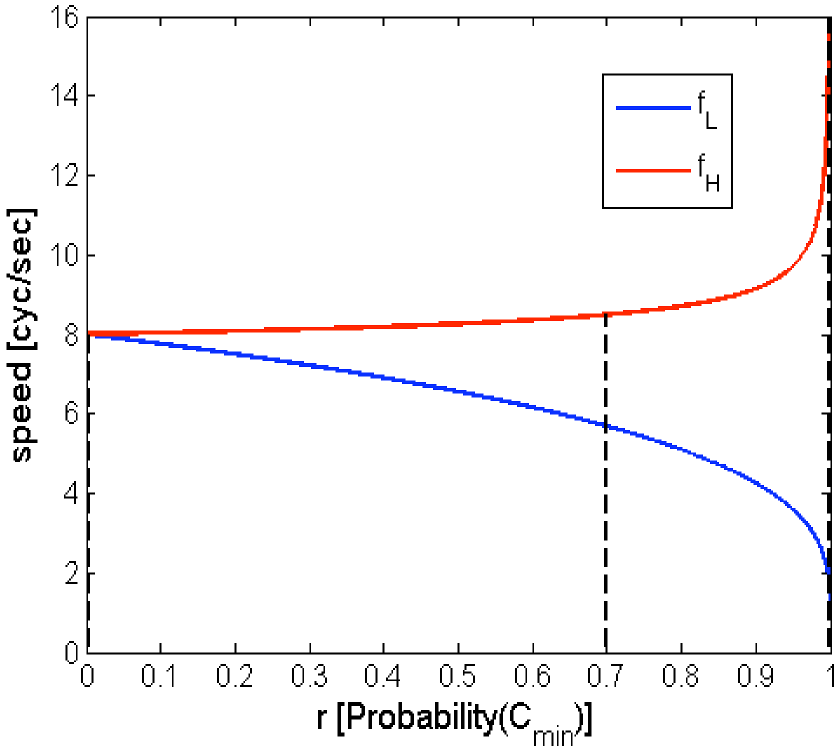
C_{max}-C_{min} cycles at f_H



Exploiting probability

p.d.f.=2 Dirac's delta

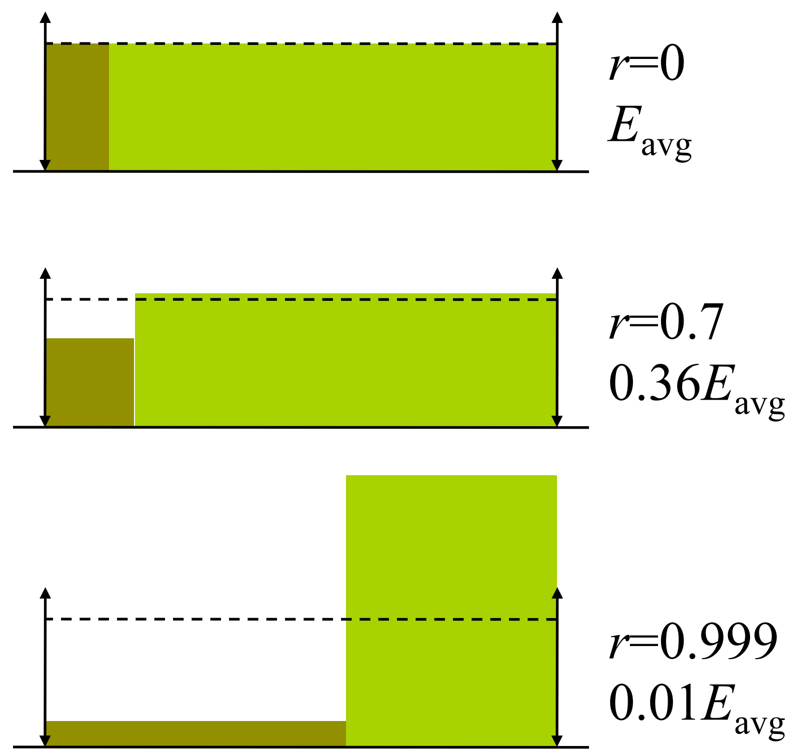




Adopting a 2-speeds scheme

C_{min} cycles at f_L

C_{max}-C_{min} cycles at f_H

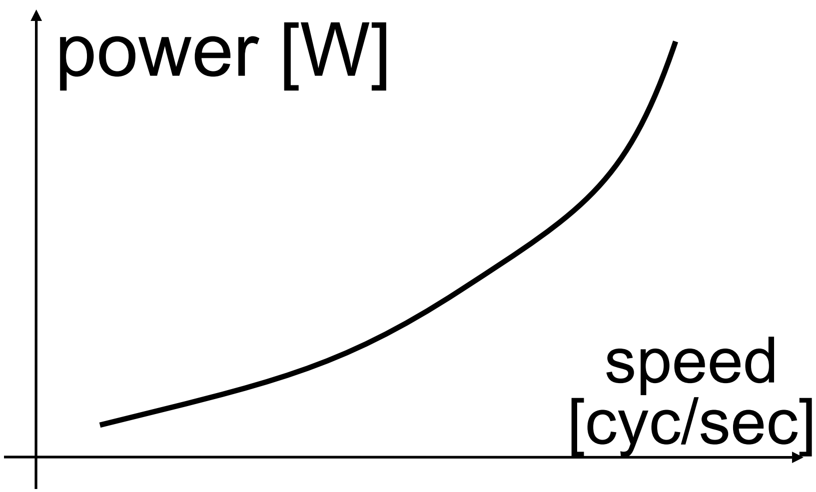


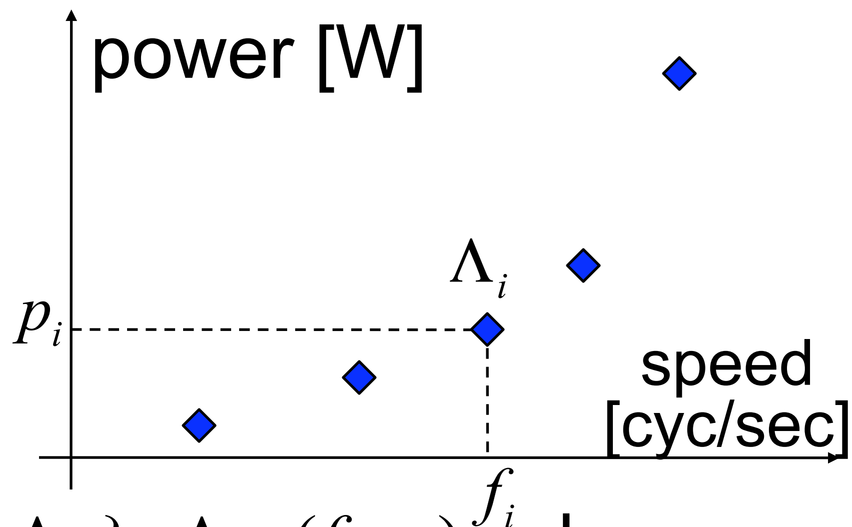
Contents

1. Model of energy consumption
2. Fixed speed schemes
3. Varying speed schemes
4. **Accounting for discrete modes and overheads**

Discrete modes

Available processors only allow a finite and predetermined number of operating modes





Operating modes $\{\Lambda_1,\ldots,\Lambda_m\}$. $\Lambda_k=(f_k,p_k)$, where:

- f_k, the speed;
- p_k, the power consumed.

Rounding to the upper level

1. compute the desired continuous level f_{opt}
2. select the mode Λ_k such that

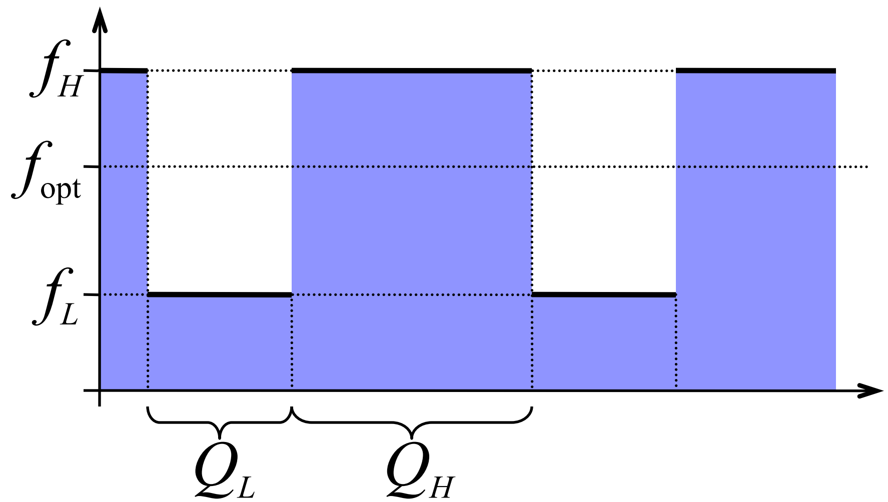
$$\min\{p_k : f_k \geq f_{\text{opt}}\}$$

Pros: it applies to any continuous speed algorithm

Cons: a quantization error is incured

Approximating the ideal speed

Two available speed approximate the ideal one

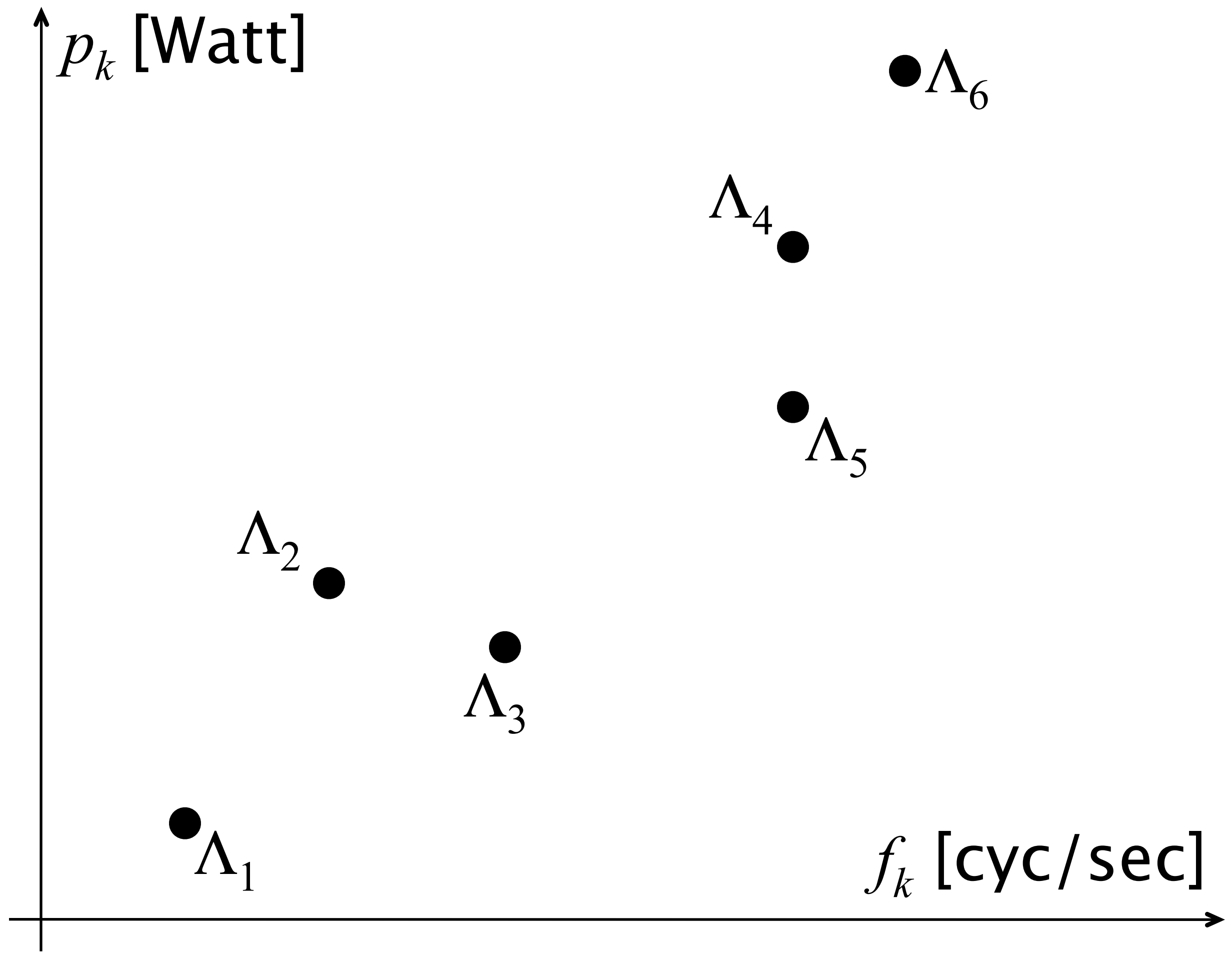


Speed schedule does not depend on task schedule: more robust practice!!

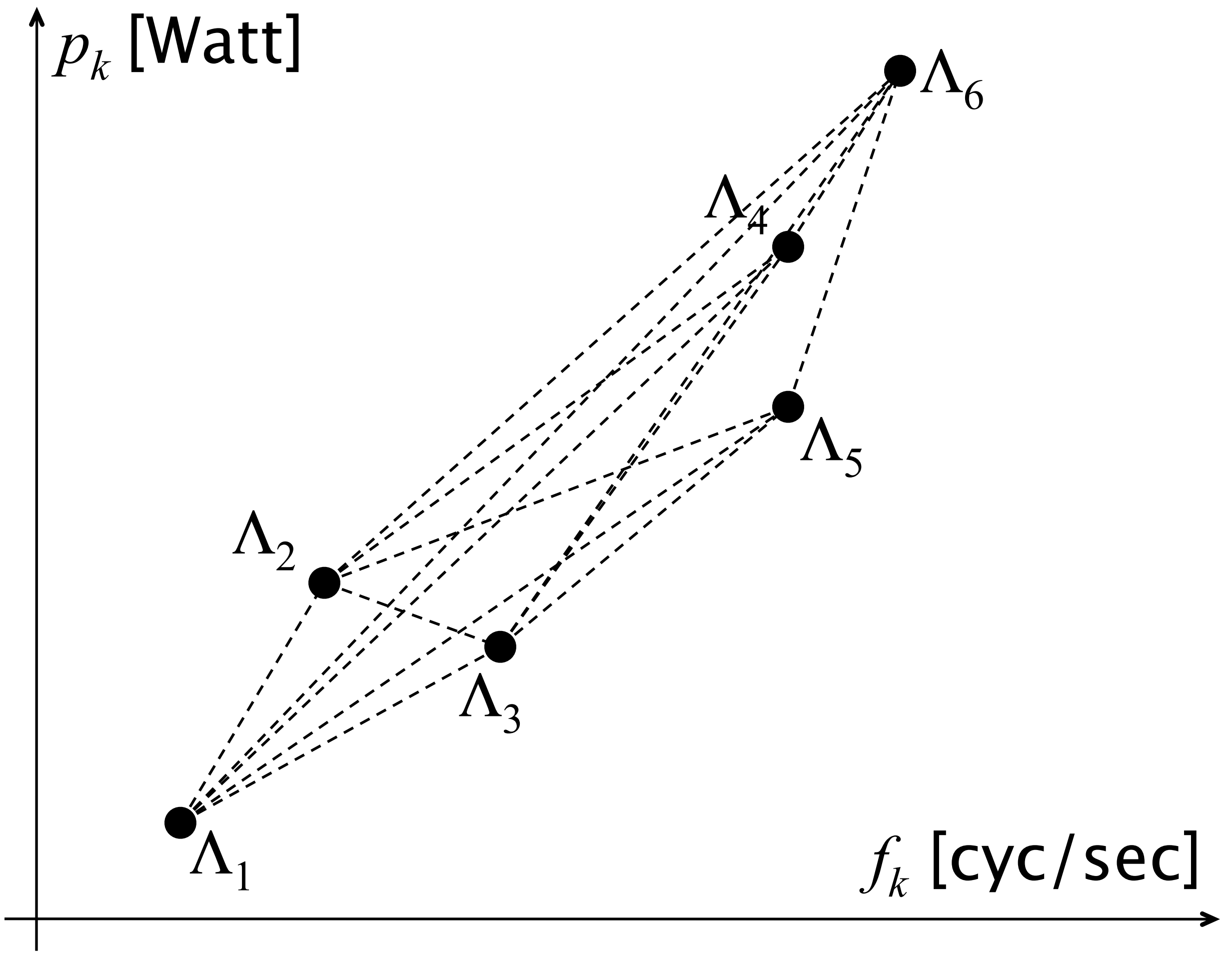
Two open problems:

1. to find the mode pair (Λ_L, Λ_H);
2. to find Q_L and Q_H

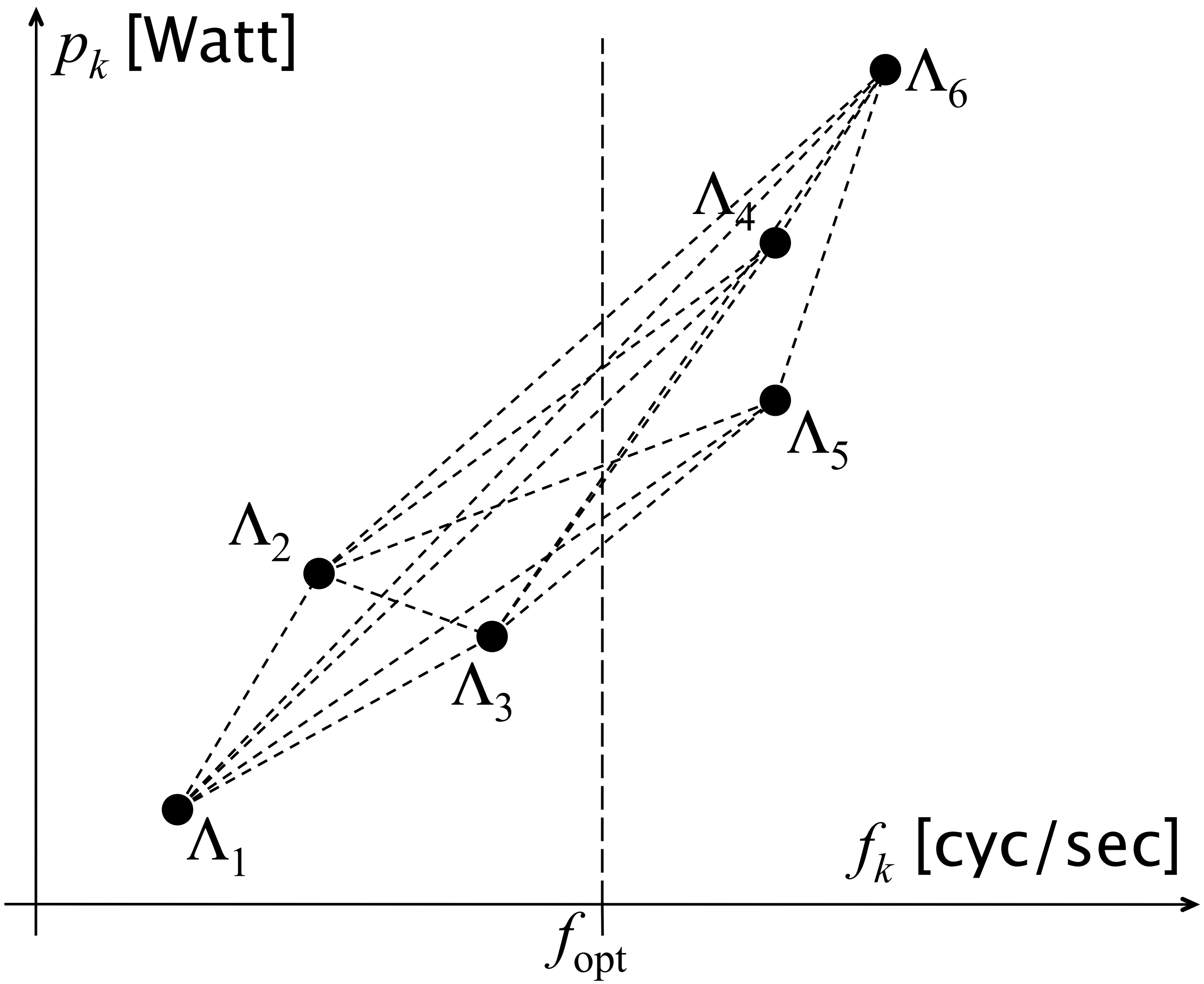
Selecting the pair (Λ_L, Λ_H)



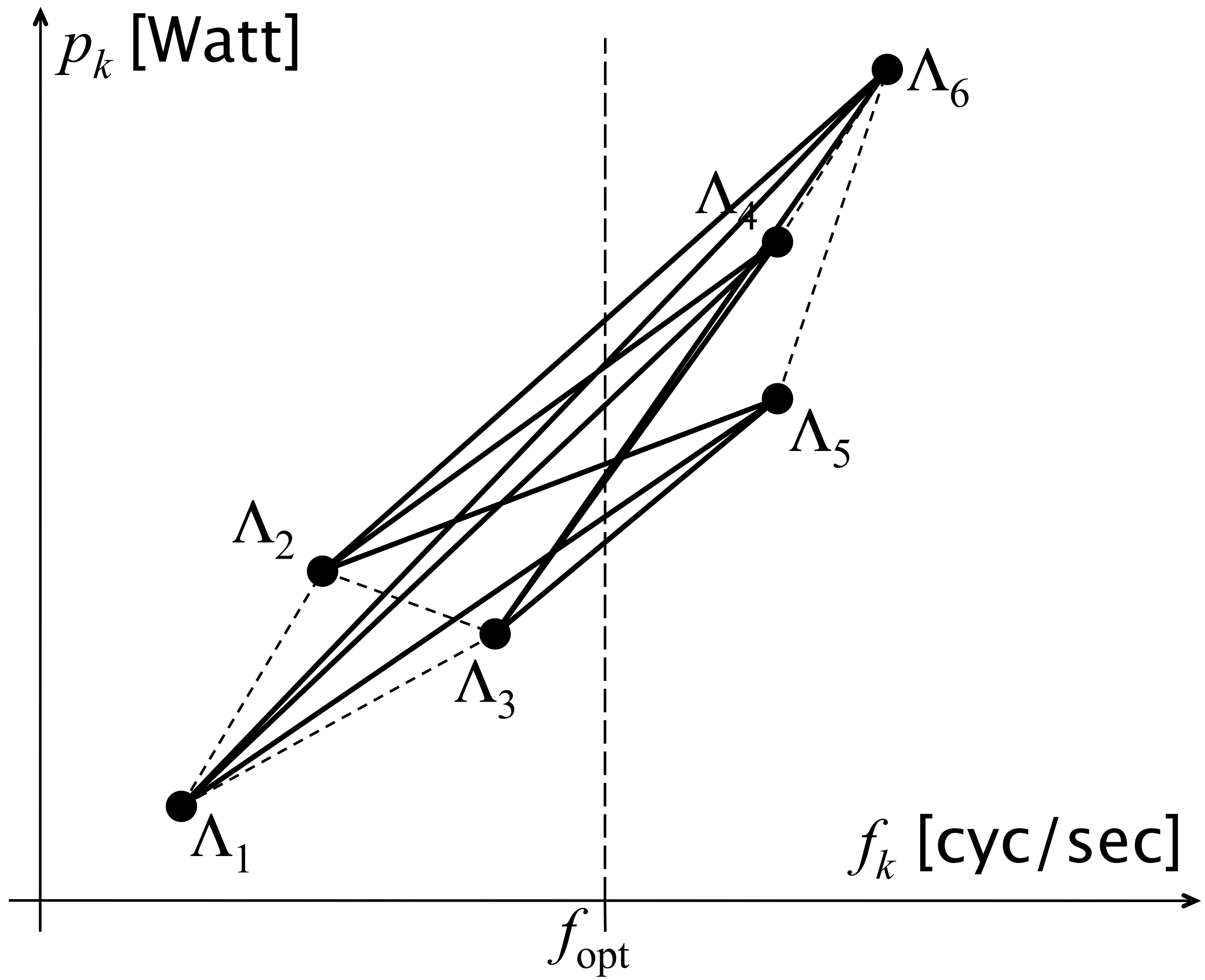
Selecting the pair (Λ_L, Λ_H)



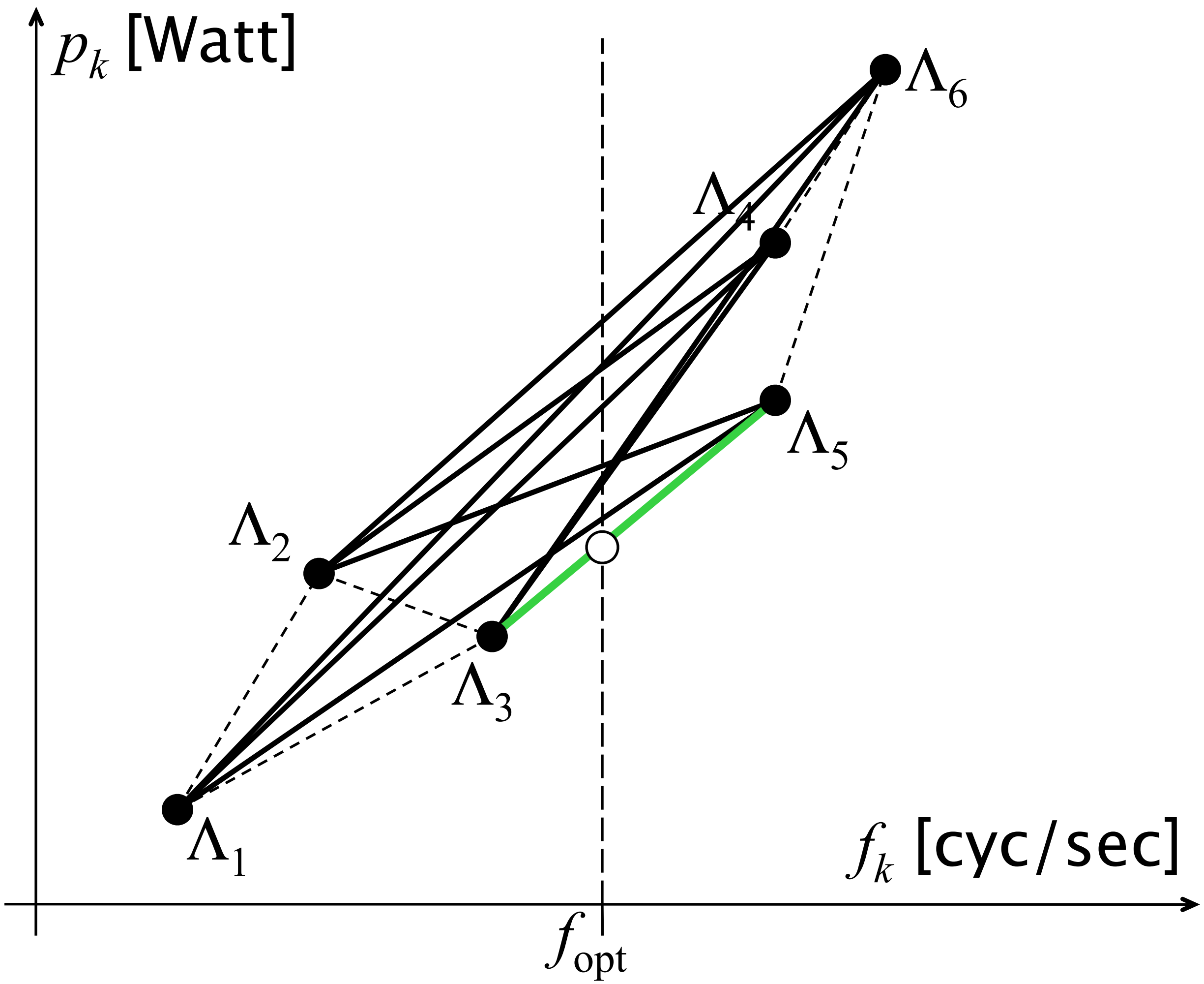
Selecting the pair (Λ_L, Λ_H)



Selecting the pair (Λ_L, Λ_H)



Selecting the pair (Λ_L, Λ_H)



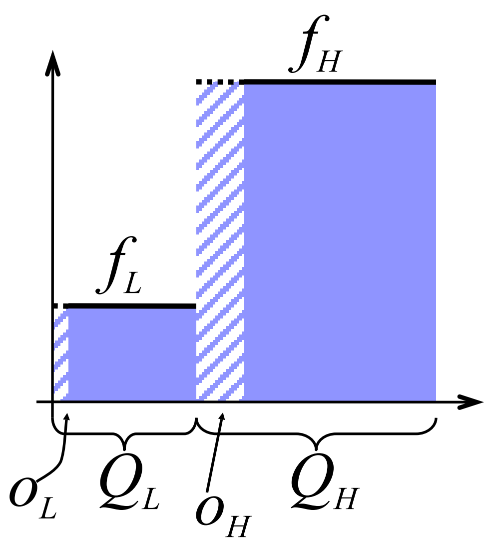
$\Lambda_L=\Lambda_3$ and $\Lambda_H=\Lambda_5$ is the minimal energy choice!!

Switching overhead

Mode switch requires energy and time overhead.

Let be $\Lambda_k = (f_k, p_k, o_k, e_k)$, where:

- o_k, time overhead [sec] to enter mode Λ_k;
- e_k, energy overhead [Joule] to enter Λ_k.



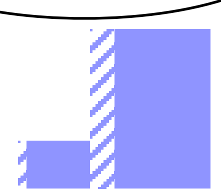
$$P = Q_L + Q_H$$

period of the scheme

$$\lambda_L = Q_L / P$$

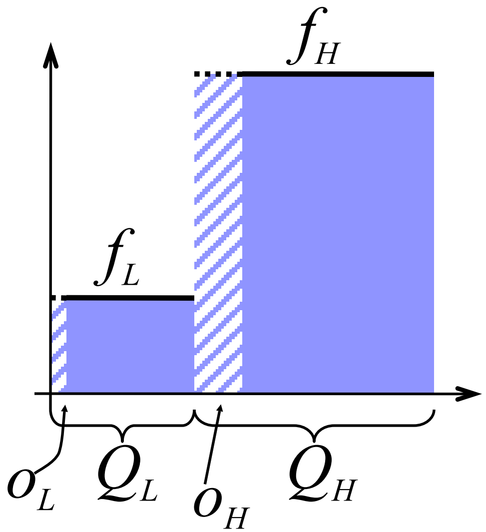
fraction in Λ_L mode

convex combination



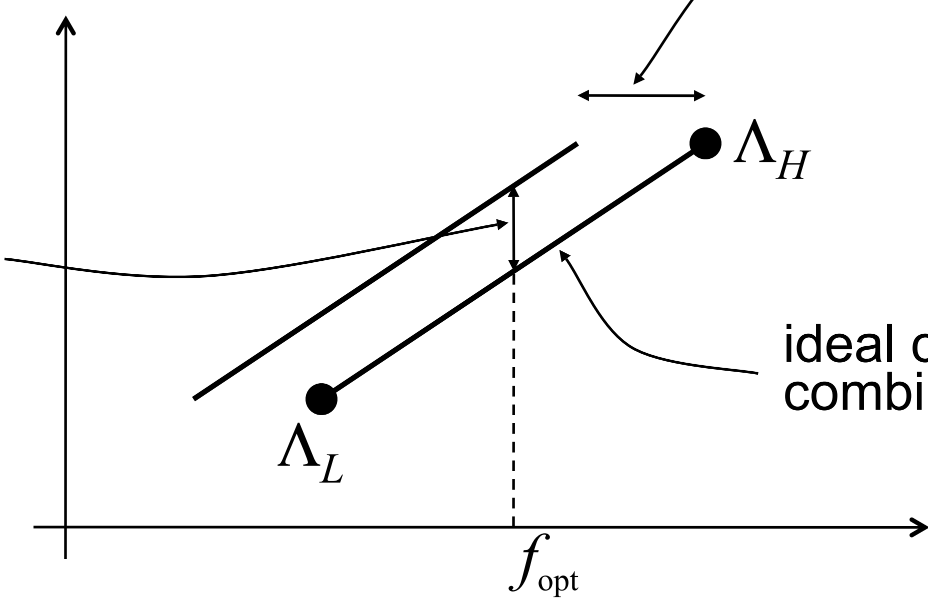
$$f_{\text{opt}} = \frac{\quad}{P} = \lambda_L f_L + (1 - \lambda_L) f_H$$

Accounting for time overhead



translation
of the convex
combination

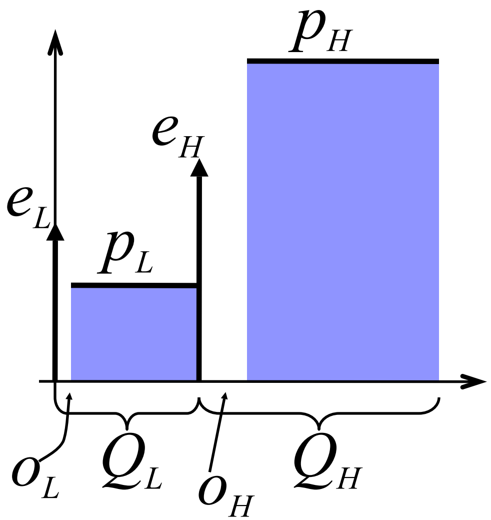
$$f_{\text{eff}} = f_{\text{opt}} - \frac{\quad}{P} = f_{\text{opt}} - \frac{o_L f_L + o_H f_H}{P}$$



power increase for
same speed f_{opt}

ideal convex
combination (P®∞)

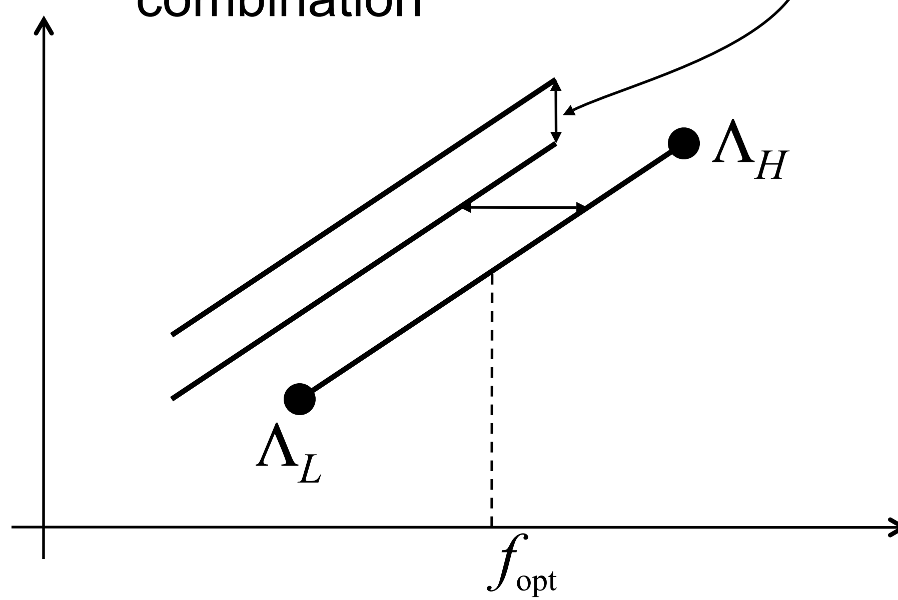
Accounting for energy overhead



$$p_{\text{eff}} = \lambda_L p_L + (1-\lambda_L) p_H + \frac{e_L + e_H - o_L p_L - o_H p_H}{P}$$

ideal convex combination

power increase due to energy overhead



Selecting the values (Q_L, Q_H)

1. The energy consumption is minimized when P®∞
2. There are arbitrary long interval Q_L running at speed f_L
3. Some task will miss a deadline, otherwise $f_{\mathrm{opt}} \leq f_L$ (by construction $f_{\mathrm{opt}} > f_L$)
4. (Q_L, Q_H) must be selected as large as possible (min power), such that no deadline is missed
5. We can use network-calculus-like technique

Conclusion

1. Model of energy consumption
   CMOS, future trends
2. Fixed speed schemes
   fixed Priorities, EDF
3. Varying speed schemes
   reclaiming unused cycles, exploiting probability
4. Accounting for discrete modes and overheads
   time, energy overhead