Energy Efficient Architecture

Energy Efficiency

Initial motivation — mobile computing

- Large market potential identified in 1990s
- Market synergy with WiFi
  - Wireless networking makes laptop more attractive
  - Laptop makes WiFi hardware more attractive
- Limited by battery life-time
- Wireless transceiver in laptop limits battery life-time far more

Subsequent insight

- Electrical power limits computing power
  - Performance / rack = (performance / kW) × (kW / rack)
  - Improving (performance / kW) improves large system performance
- Electric bills for large server farms (ignoring cooling)
  - (400 W / PC server) × 1000 servers = 400 kW
  - 400 kW ⇒ 292,000 kW-Hr / month ~ ח"ש 146,000

Competition

- Mainframe centralization ⇒ greater energy efficiency

Power Consumption

Consumed power in digital circuits

Most current flows during 0 $\rightarrow$ 1 and 1 $\rightarrow$ 0 state transitions

$$\text{Power} = P = \frac{\text{energy}}{\text{transition}} \times \frac{\text{transitions}}{\text{second}} = \frac{\text{energy}}{\text{clock cycle}} \times F, \; F = \text{clock frequency}$$

$$= \frac{\text{energy}}{\text{task}} \times \frac{\text{tasks}}{\text{clock cycle}} \times \frac{1}{\tau} = \frac{\text{energy}}{\text{task}} \times \frac{1}{\text{clock cycle per task}} \times \frac{1}{\tau}$$

$$= \frac{\text{energy}}{\text{task}} \times \frac{1}{CPI \times IC} \times \frac{1}{\tau}$$

$$= \frac{\text{energy}}{\text{task}} \times \frac{1}{CPI \times IC \times \tau} = \frac{\text{energy}}{\text{task}} \times \frac{1}{T}, \qquad T = \text{run-time}$$

Reduce power consumption

Lower clock frequency F

Lower energy per transition

- Raise energy efficiency of transistor circuits
- Maintain performance with fewer circuit transitions per operation

Power and Performance

Energy consumption

$$\text{Power} = P = \frac{\text{energy}}{\text{task}} \times \frac{1}{T} \quad \Rightarrow \quad \frac{\text{energy}}{\text{task}} = P \times T$$

$T = \text{run-time}$

Performance

$$\text{Performance} = \frac{\text{tasks}}{\text{run-time}} = \frac{IC}{T} = \frac{IC}{CPI \times IC \times \tau} = \frac{F}{CPI} = IPC \times F$$

$IPC = \text{instructions per clock cycle}$

Energy and performance

$$\text{Performance} = \frac{IC}{T} \quad \Rightarrow \quad T = \frac{IC}{\text{Performance}}$$

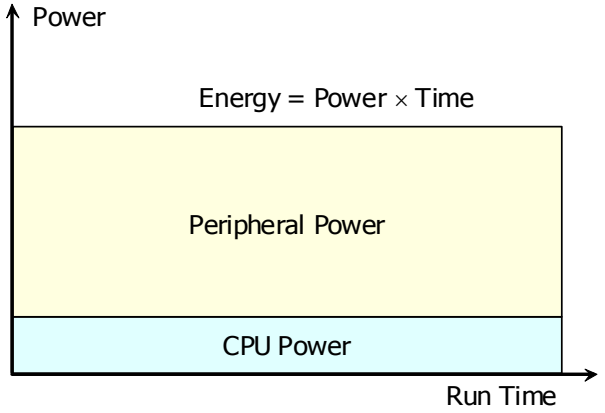
$$\frac{\text{energy}}{\text{task}} = P \times T = \frac{\text{power}}{\text{Performance}} \times IC$$

System Issues

CPU

Consumes 10% of power

Higher performance ⇒ higher CPU power



Peripherals

Consume 90% of power

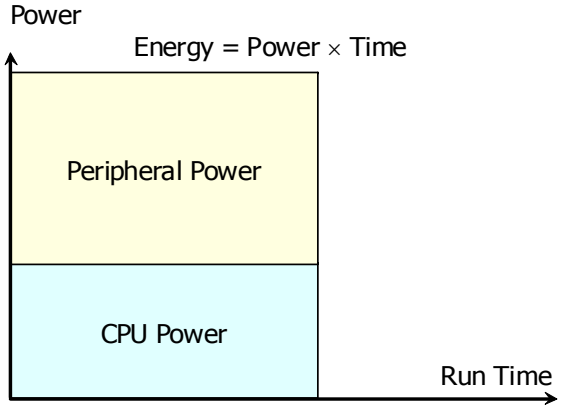
Power independent of CPU performance

LCD, hard disk, memory system, networking components, I/O system

Idle CPU ⇒ system decreases power to CPU and peripherals

Increase performance allowing energy per task to increase

- Increases power / performance ratio
- Increases total CPU energy consumption
- Decreases run-time
- Decreases total peripheral energy consumption



Overall strategy

Fast CPU clock ⇒ maximize CPU idle time

Minimize total consumed energy

Classes of Architectural Changes

Changes to processor architecture affect

Power consumption

Performance = IC / run-time

Energy = Power × run-time

		Run-Time	
		Decrease ↓	**Increase ↑**
Consumed Power	**Decrease ↓**	Energy per Task ↓ Total Consumed Energy ↓ Improvement in all categories	Total Consumed Energy ↓ or ↑ Worthwhile for small increase in run-time
	Increase ↑	Total Consumed Energy ↓ or ↑ Possibly worthwhile	Total Consumed Energy ↑ Degradation in all categories

Choosing Architectural Changes

Default processor operating point (zero point)

Consumes power P_0

Achieves performance S_0

Defines default state vector (P_0, S_0)

Change in processor operating point

$$(P_0, S_0) \rightarrow (P_0 + \Delta P, S_0 + \Delta S) = (P_0, S_0) + (\Delta P, \Delta S)$$

Energy efficiency criteria

$\Delta P < 0$ and $\Delta S > 0 \Rightarrow$ accept as improvement

$\Delta P < 0$ and $\Delta S < 0 \Rightarrow$ accept as improvement if same ΔP cannot be achieved with a smaller $|\Delta S|$

$\Delta P > 0$ and $\Delta S > 0 \Rightarrow$ accept as improvement for small ΔP and large ΔS

$\Delta P > 0$ and $\Delta S < 0 \Rightarrow$ reject

Physical Parameters in Circuit Performance

1. **Consumed power at zero point (default working Voltage)**

$$P_0 = \alpha \times C_0 \times V_0^2 \times F_0$$

α = constant

C_0 = effective capacitance

V_0 = default CPU voltage

F_0 = default clock frequency

2. **Effective contribution of clock frequency**

$$F_0 = K_f \times V_0$$

K_f = constant

3. **Performance for given clock frequency**

$$S_0 = IPC_0 \times F_0$$

$$IPC_0 = \frac{1}{CPI_0}$$

4. **Total energy consumed by task**

$$E_0 = P_0 \times \text{run-time}$$

Voltage Scaling Relations

Small change in Voltage $V_0 \rightarrow V_0 + \Delta V \Rightarrow$ changes $(\Delta P, \Delta S)$

Combine (1) and (2) $\Rightarrow P_0 = \left(\alpha C_0 V_0^{\,2}\right)\left(K_f V_0\right) = \alpha C_0 K_f V_0^{\,3}$

For small Voltage change ΔV

$$\Delta P \approx \frac{dP_0}{dV_0} \times \Delta V = \alpha C_0 K_f \times 3V_0^{\,2} \times \Delta V$$

Relative change in power

$$\frac{\Delta P}{P_0} = \frac{3\alpha C_0 K_f V_0^{\,2} \Delta V}{\alpha C_0 K_f V_0^{\,3}} = \frac{3\Delta V}{V_0}$$

Combine (2) and (3) $\Rightarrow \quad S_0 = \left(IPC_0\right)\left(K_f V_0\right) = IPC_0 K_f V_0$

Relative change in performance

$$\frac{\Delta S}{S_0} = \frac{IPC_0 K_f \Delta V}{IPC_0 K_f V} = \frac{\Delta V}{V_0}$$

Energy Scaling Relations

Relative change in energy

$$E_0 = P_0 \times \text{run-time} = \frac{P_0}{S_0} = \frac{\alpha C_0 V_0^{\,2}}{IPC_0} \Rightarrow \frac{\Delta E}{E_0} = \frac{2\Delta V}{V_0}$$

General change in operating point

$$\frac{\Delta E}{E_0} = \frac{1}{E_0}\left[\frac{\partial E_0}{\partial P_0}\Delta P + \frac{\partial E_0}{\partial S_0}\Delta S\right] = \frac{S_0}{P_0}\left[\frac{1}{S_0}\Delta P + \frac{-P_0}{S_0^{\,2}}\Delta S\right] = \frac{\Delta P}{P_0} - \frac{\Delta S}{S_0}$$

Energy break - even point

$$\frac{\Delta E}{E_0} = 0 \Rightarrow \frac{\Delta P}{P_0} = \frac{\Delta S}{S_0}$$

Voltage scaling stronger than break - even

$$\frac{\Delta E}{E_0} = \left[\frac{\Delta P}{P_0} - \frac{\Delta S}{S_0}\right]_{\substack{\text{Voltage}\\ \text{scaling}}} = 3\frac{\Delta V}{V_0} - \frac{\Delta V}{V_0} = 2\frac{\Delta V}{V_0}$$

Voltage Scaling Criterion

Voltage Scaling Relations

$$\frac{\Delta S}{S_0} = \frac{\Delta V}{V_0} \qquad \frac{\Delta P}{P_0} = 3\frac{\Delta V}{V_0} \qquad \frac{\Delta E}{E_0} = 2\frac{\Delta V}{V_0}$$

1% drop in CPU voltage $\Rightarrow \frac{\Delta V}{V_0} = -0.01 \Rightarrow$

3% drop in power consumed
1% drop in performance
2% drop in total energy consumed

Criterion for architectural enhancement

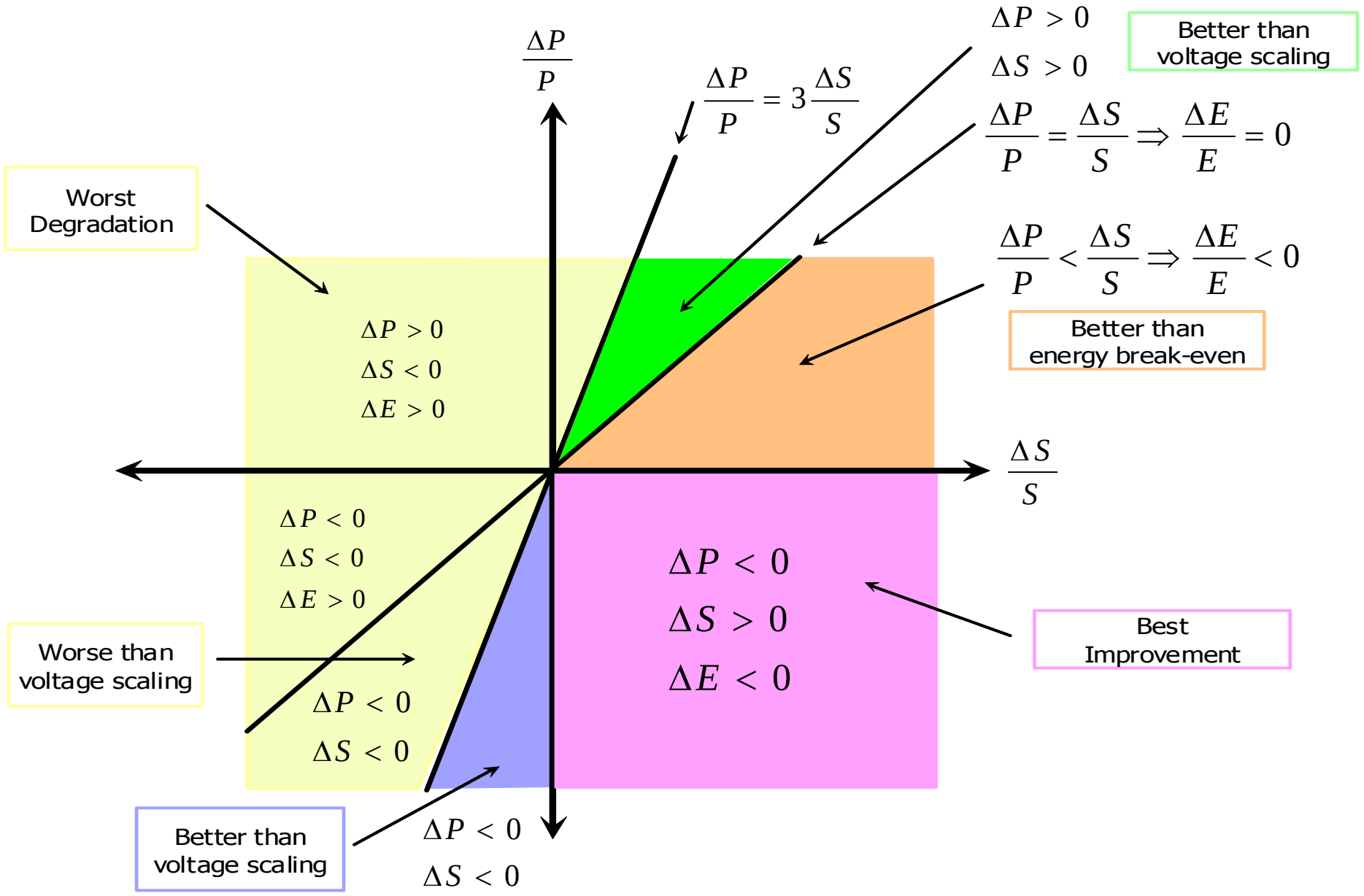
Enhancement worse than voltage scaling

- Lowers power by 3% ($\Delta P / P_0 = -0.03$)
- Lowers performance by more than 1% ($|\Delta S / S_0| > 0.01$)
- Preferable to lower CPU voltage

Enhancement better than voltage scaling

- Lowers power by 3% ($\Delta P / P_0 = -0.03$)
- Lowers performance by less than 1% ($|\Delta S / S_0| < 0.01$)
- More cost-effective than lowering CPU voltage

Architectural Enhancement Criteria



Intel Pentium M

Intended for laptop computers

- Processor component in Centrino system
- M = mobile

Design requirements set by Intel

- Satisfy IA-32 Instruction Set Architecture (ISA)
  - IA-32 instruction set for integer and floating point numbers
  - MMX and SSE/SSE2 vector extensions
- Implement Net-burst micro-architecture
  - Enhancements of P6 architecture used on Pentium II and III
  - Perform internal operations as on Pentium 4
- Save power
  - Maximize battery life in laptop
- Provide "close-to-desktop" performance
  - Performance comparable to Pentium 4 of same generation

Strategy

- Introduce architectural changes that meet Voltage scaling criterion

Enhancements to Lower Power

Improve branch prediction

- Decrease number of instructions per task
  - Fewer mispredicted speculative loads
- Decrease run time
  - Higher performance and lower power

Decrease number of micro-ops per instruction

- Micro-op fusion — combine micro-ops
  - Less micro-op independence
  - Lower performance at lower total energy

Decrease number of transistor switches per micro-op

- Lower performance at lower total energy

Decrease energy per transistor switch

- Higher quality transistors
- Higher performance and lower power

Branch Prediction Enhancements

Loop detector

- Branch instruction executes N+1 times
- Either
  - Taken N times and not-taken once
- Or
  - Not-taken N times and taken once
- Detector allocates counter to accurately predict subsequent loops

Indirect branch predictor

- Branch target is data-dependent
- Records global target history
- Lowers misprediction of branch target

Jump detector

- Branch instruction misses in branch predictor (first encounter)
- Predicted not taken if jumps forward
- Predicted taken if jumps backward

Pentium-M misprediction rate

- 20% lower than Pentium 4

Micro-Op Fusion

P6 / Netburst architecture

- IA-32 instructions decoded to micro-ops
- Out-of-order micro-op execution produces high throughput

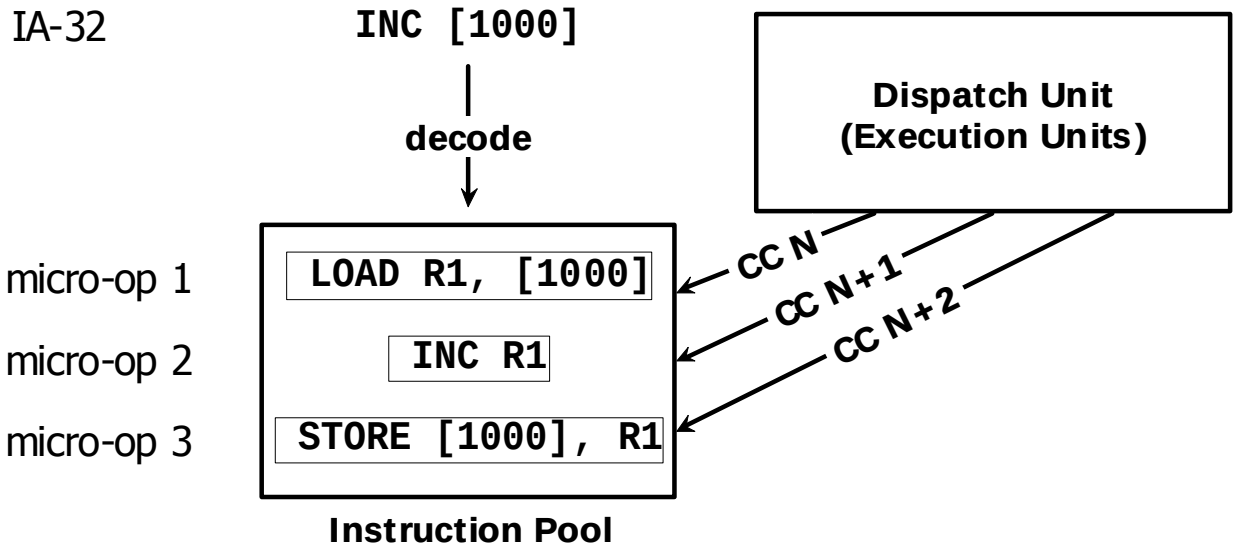
Cost of execution efficiency

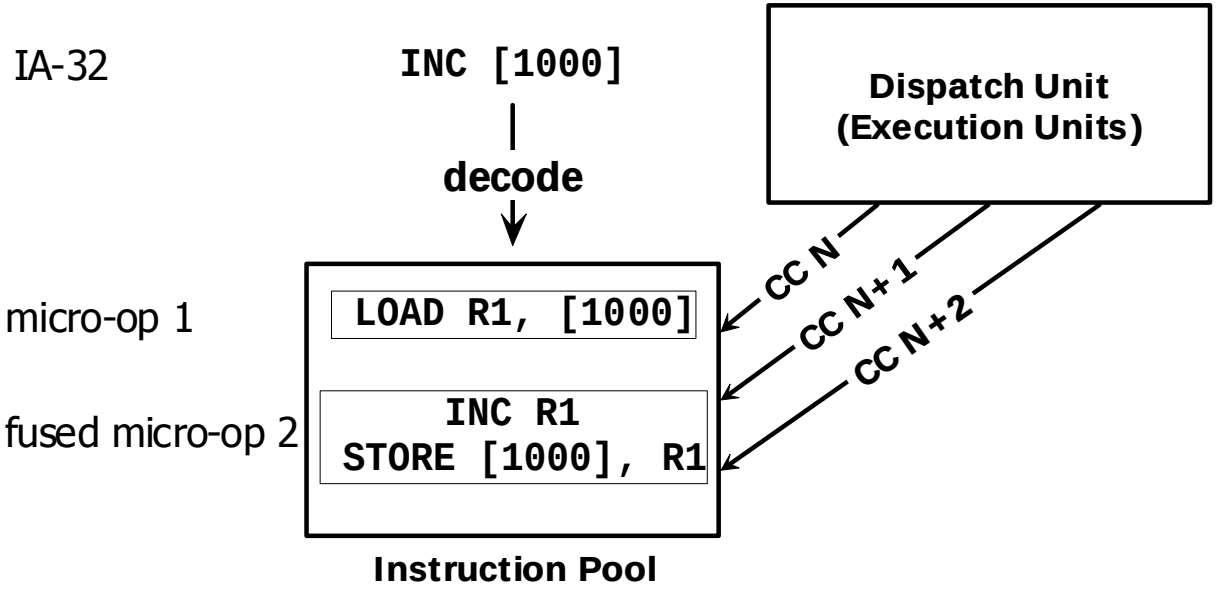
- Resource requirements depend on number of micro-ops
- Decoders, instruction pool registers, registration station
- Independent processing of micro-ops consumes more power

Pentium-M

- Decodes certain IA-32 instructions to fewer micro-ops
- Fused micro-ops
  - Contain multiple inherently sequential P6 micro-ops
  - Stored as single fused micro-op with independent sections
  - Dispatch unit acts on one independent section of micro-op
  - Executed in dispatch unit once for each section
- Reduces number of micro-ops by about 10%

Example of Fused Micro-Op





Dedicated Stack Engine

IA-32 stack operations operate on hardware stack pointer

PUSH source	POP destination
ESP ← ESP – 4	destination ← SS:ESP
SS:ESP ← source	ESP ← ESP + 4

P6/Netburst implements ESP updates as micro-ops

POP EAX decoded as

EAX ← SS:ESP

ESP ← ESP + 4

ESP update micro-op executed in execution unit

Pentium-M updates ESP in dedicated hardware

Each **PUSH/POP** decoded as transfer and update

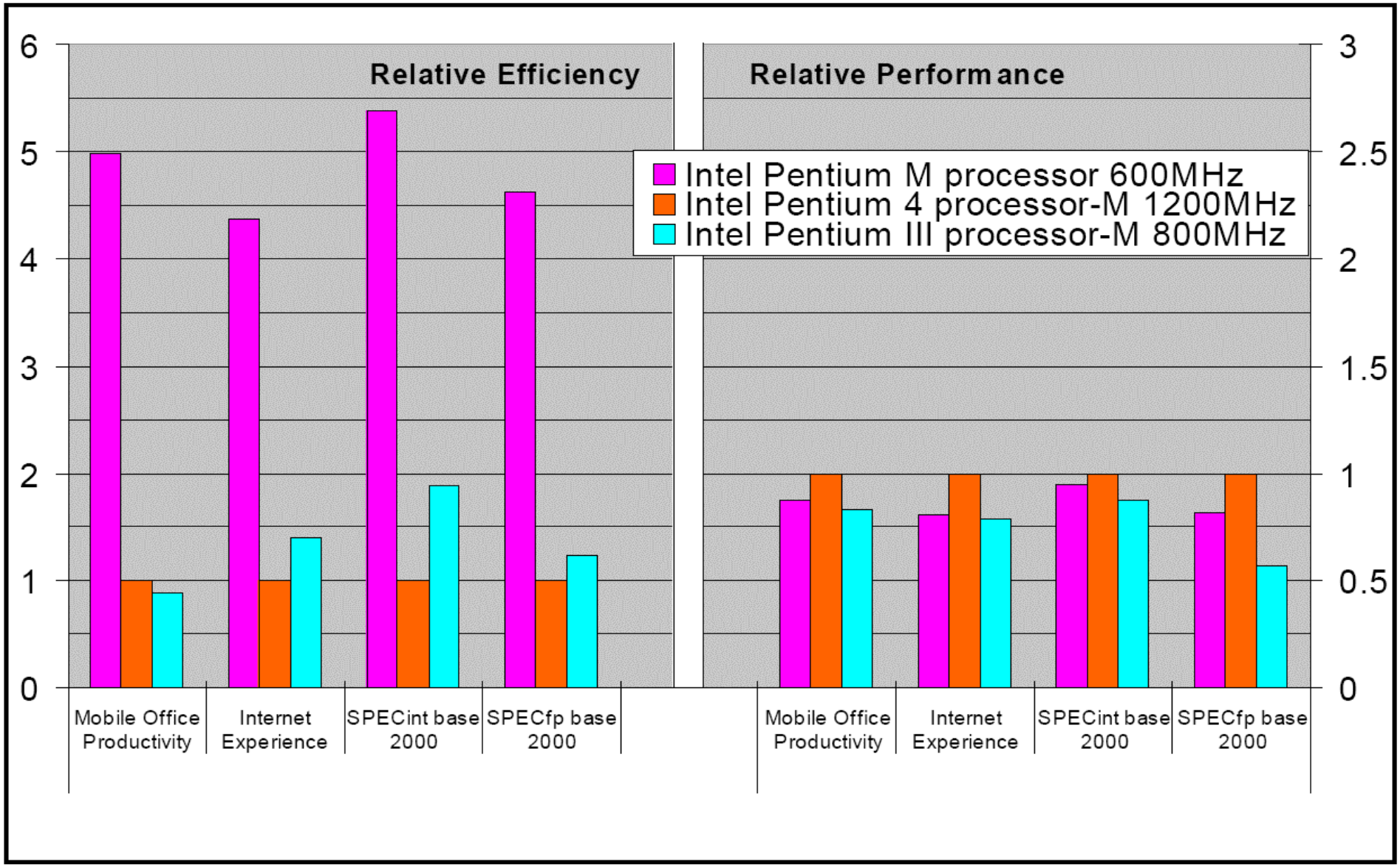
$$ESP_{IA\text{-}32} = ESP_{decode\text{-}time} + ESP_{delta}$$

ESP updates performed on ESP_{delta} until retirement of $ESP_{IA\text{-}32}$

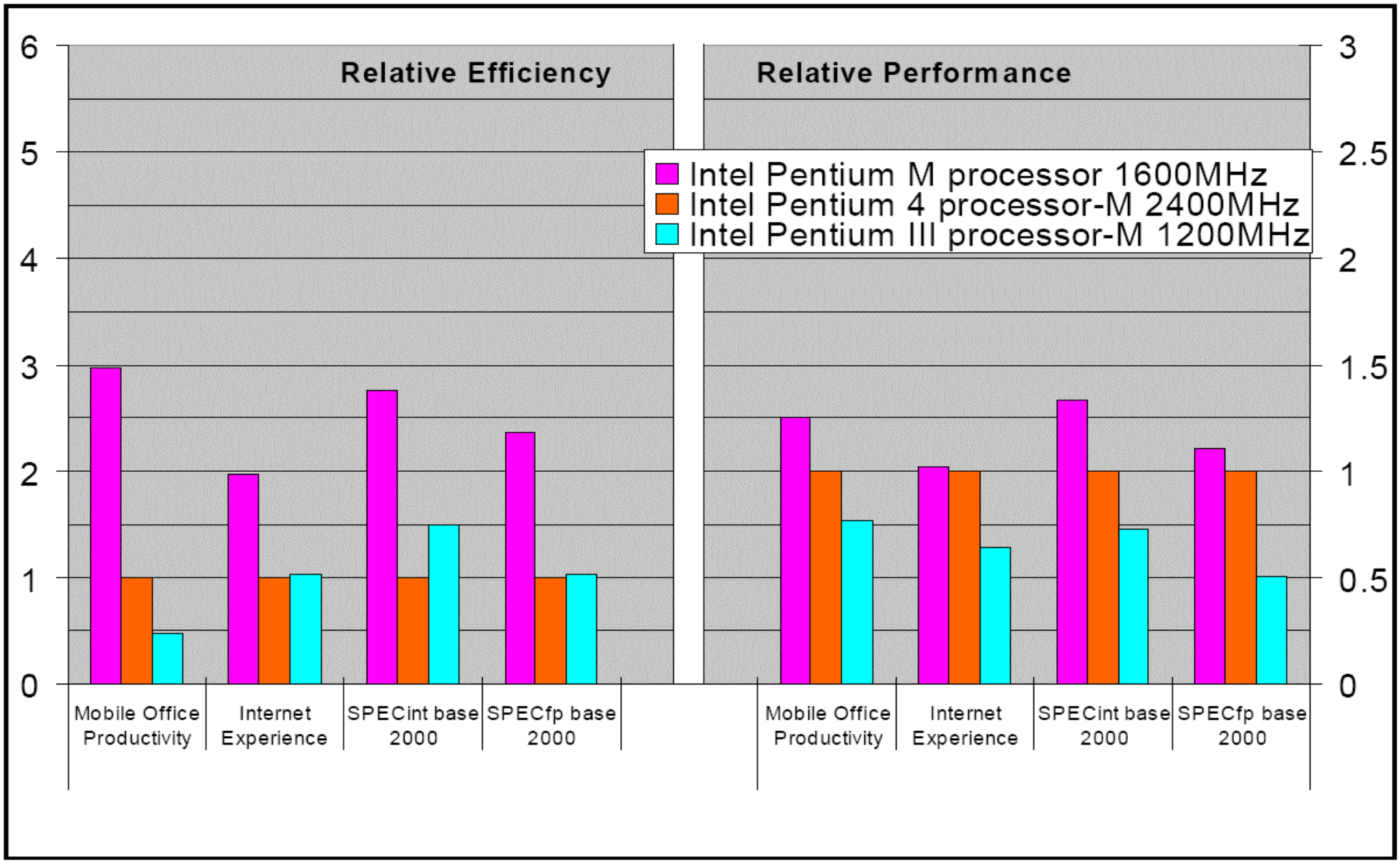
Dedicated counter keeps track of each ESP_{delta} and status

$ESP_{IA\text{-}32}$ retirement updates $ESP_{decode\text{-}time}$ and zeros ESP_{delta}

Maximum Battery Save Mode



Always-On Full Speed Mode



Intel Core Duo

Follow-up to Pentium-M

Contains two enhanced Pentium-M cores

Extends Intel Speedstep Technology

Design requirements set by Intel

Similar or better performance to Pentium M on single thread code

Improved performance for multithreaded and multi-process code

Average power consumption of dual core same as Pentium M

Meets thermal requirements of target systems

Independent thermal control per core

Overall package-level power state control

Per-core power state control in package

Gochman, S. et. al., "Introduction to Intel Core Duo Processor Architecture." Intel Technology Journal.
Mendelson, A. et. al., "CMP Implementation in Systems Based on the Intel Core Duo Processor" Intel Technology Journal
Naveh, A. et. al., "Power and Thermal Management in the Intel Core Duo Processor" Intel Technology Journal

ACPI

Advanced Configuration and Power Interface

- Industry standard for OS interface to processor configuration
- Developed by HP, Intel, Microsoft, Phoenix, Toshiba

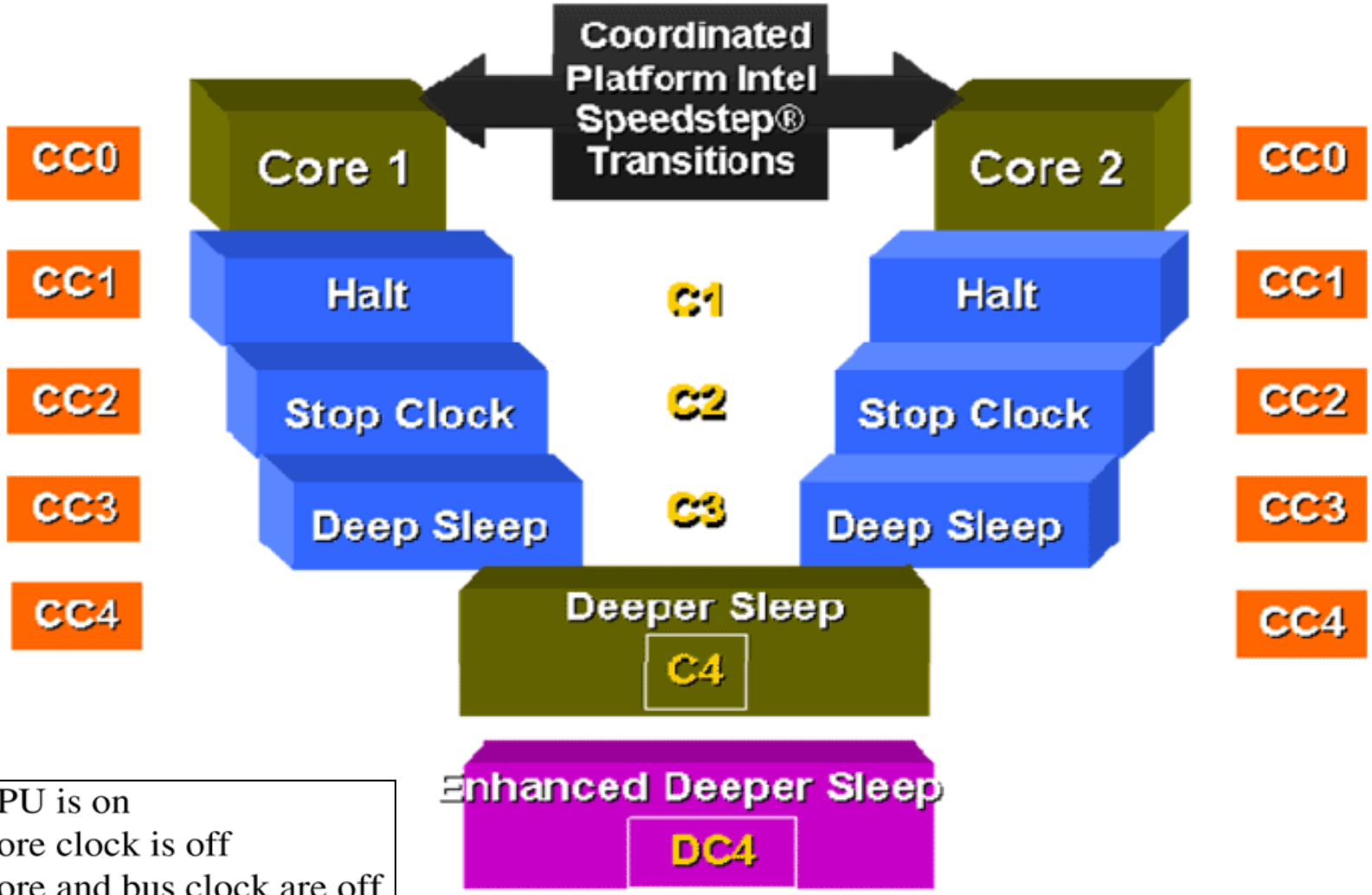
Defines operating states

- Global (system) states
  - G0 (working), G1 (sleeping), G2 (software off), G3 (mechanical off)
- Device states
  - D0 (on), D1/D2 (device dependent) , D3 (off)
- CPU power states
  - C0 (active), C1 (halt), C2 (stop clock), C3(sleep), C4 (deep sleep), ...

OS controls CPU power state

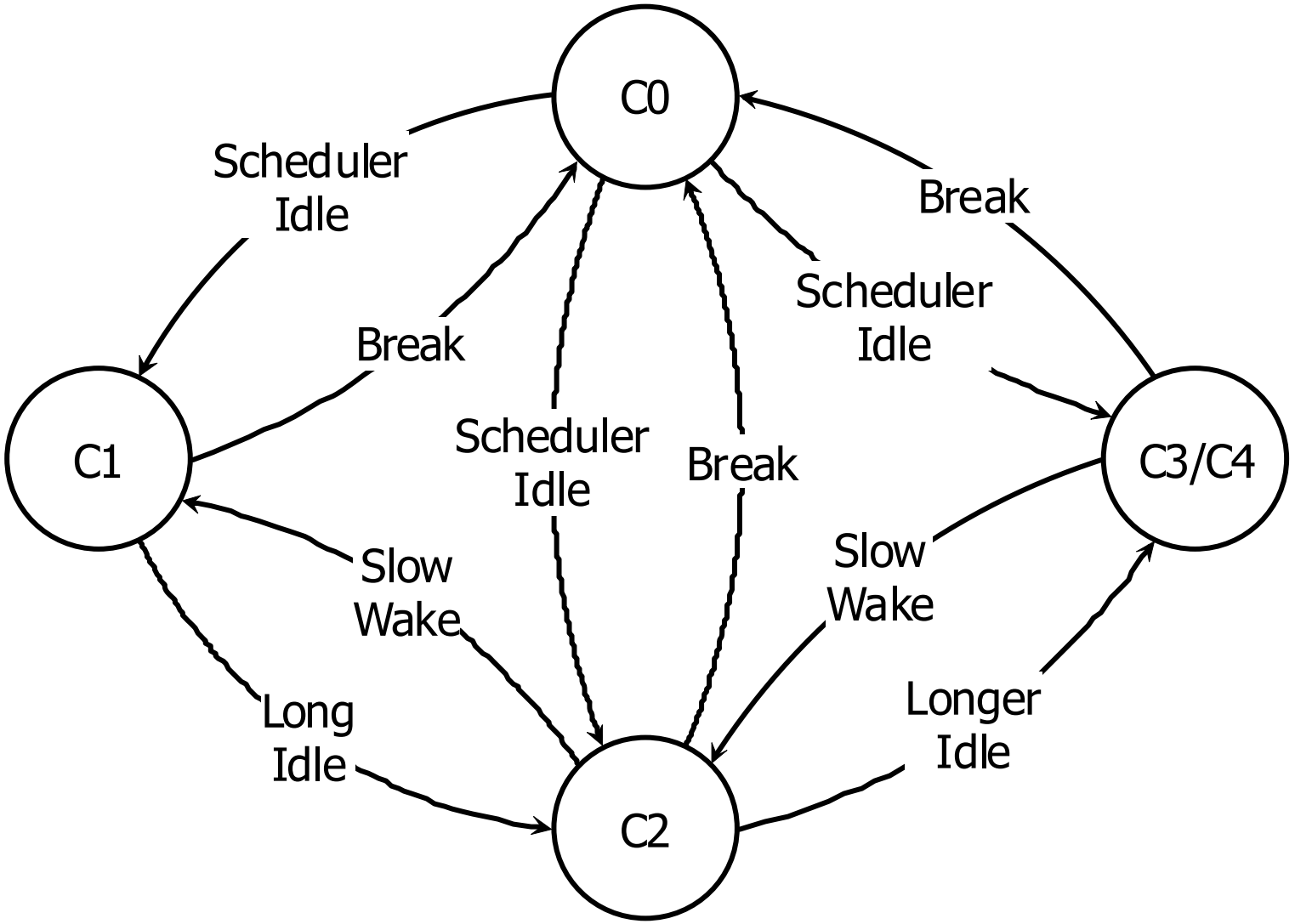
- Scheduler detects empty task queue
- Transitions processor into idle state with BIOS instruction sequence
- Processor remains idle until break event
  - Interrupt
  - Indicator of new task to be executed
- OS returns processor to C0 state

Core Duo Sleep States



C0 – Active	CPU is on
C1 – Auto Halt	Core clock is off
C2 – Stop clock	Core and bus clock are off
C3 – Deep sleep	Clock generator is OFF
C4 – Deeper sleep	Reduced VCC
DC4 –Deeper C4	Further reduced VCC

Possible ACPI Idle Management



Anticipated Sleep State Transitions

OS given range of sleep options

Can move idle processor from C0 directly into any sleep stage

C1, C2, ...

Can move idle processor from C0 to sleep states in stages

C1 → C2 → C3 → C4 → ...

Core Duo sleep states

C1	**Instruction execution and core clocks halted**
C2	**Processor cannot access external bus** **Front side bus and chipset placed in lower power state**
C3	**Processor disables internal phase locked loops** **L1 cache flushed and powered off**
C4	**Processor decreases internal voltage** **No processing circuits can operate**
DC4	**L2 cache flushed and powered off**