

## Processor Energy Efficiency

SOCRATES'04

Joan Oliver

Escola Tècnica Superior d'Enginyeria

Universitat Autònoma de Barcelona

## Processor Design for Portable Systems

T. Burd and R. Broderson in the paper *Processor Design for Portable Systems* (Journal of VLSI Signal Processing – 1996) proposes a new method to evaluate power analysis techniques based on circuit-level or architectural-level.

- Imperative need to minimise the load on the battery while increasing speed of computation.
- Approaches:
  - Moving data processing to DSP specialised circuits with high orders of efficiency → Achievement of orders of magnitude in energy efficiency. → But not suitable in dedicated architectures.
  - Develop new energy-efficient design approaches for large amount of control processing.
- Elements in energy-efficient design:
  - Use best technologies: energy-efficiency improvements quadratically with technology.
  - Energy-efficient design: global optimisation, instead individual. Not retrofitting!
  - Use of "aggressive" energy-efficient techniques in processors.

SOCRATES'04 – Joan Oliver

## Scheduling

- Operation in a portable environment
- Model in CMOS circuit
- Energy efficiency model
  - Fixed throughput model
  - Maximum throughput mode
  - Burst throughput mode
- Design principles
  - High performance is energy efficient
  - Fast operation can decrease energy-efficiency
  - Clock frequency reduction is not energy efficient
  - Dynamic voltage scaling is energy efficient
- Energy-efficient VLSI design
  - At instruction set architecture level
  - In the microarchitecture
  - At the circuit design level
- Energy-efficient software
  - The operating system
  - Variable performance scheduling
  - Algorithms and compilers

SOCRATES'04 – Joan Oliver

## Operation in a portable environment-I

- Typically bursty: the useful computation is interleaved with periods of idle time.
- Throughput

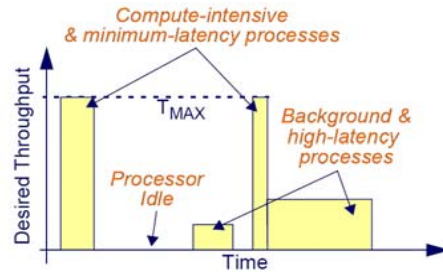
$$T = \frac{\text{Operations}}{\text{Second}}$$

- Operations:
  - fine grained: MIPS
  - coarse grained: SPECint92
- Sample usage pattern: usually throughput falls in...
  - Compute intensive and minimum-latency processes.
    - Examples: spell-check, scientific computation, ...
  - Background and high-latency processing.
    - Examples: screen update, low-bandwidth I/O control, data entry, ...
  - Processor idles: there are no processing

SOCRATES'04 – Joan Oliver

## Operation in a portable environment-II

### Processor usage model



#### ■ What to optimise?

- $T_{MAX}$  → Only employed in intensive-compute processes. Extra throughput not used in other processes!  
But peak throughput is the variable to be optimised, because the average consumption is only determined by the user!
- Amount of computation delivered for battery life → Minimisation of average energy/instruction, not the instantaneous energy consumption.

#### ■ Energy-efficient processor design target →

- Maximise peak deliverable throughput, and minimise the average energy consumed per operation

SOCRATES'04 – Joan Oliver

## Model in CMOS circuit

#### ■ Power in CMOS

$$P_{SW} = \frac{1}{2} f_{clk} V_{DD}^2 \sum_{\text{signal } y} \alpha(y) C(y) = C_{eff} f_{clk} V_{DD}^2$$

#### ■ Circuit delay

$$t_d = \frac{Q}{I_D} = \frac{C_L V_{DD}}{I_D} = \frac{C_L V_{DD}}{\mu C_{ox} \left(\frac{W}{L}\right) (V_{DD} - V_t)^2} = k' \frac{C_L V_{DD}}{(V_{DD} - V_t)^2}$$

#### ■ Throughput

$$T = \frac{\text{Operations}}{\text{Second}} = \frac{\text{OperationsPerClockCycle}}{\text{CriticalPathDelay}}$$

PDP: Power Delay Product: A common measure of the energy consumption, equivalent to power/f<sub>clk</sub>.

Energy/Operation can be considered as the PDP divided by the operations per clock cycle.

#### ■ Energy/operation

$$\frac{\text{Energy}}{\text{Operation}} = \frac{V_{DD}^2 C_{eff}}{\text{Operations/ClockCycle}}$$

SOCRATES'04 – Joan Oliver

## Energy efficiency-I

#### ■ In energy minimisation a metric must be considered for each throughput mode:

- Fixed throughput mode
- Maximum throughput mode
- Burst throughput mode

#### ■ Fixed throughput mode

- Represent processors with fixed number of operations per second. Any throughput excess will not be used.
- Systems like this work predominantly in DSP applications, with fixed throughput rate of I/O operations.
- Examples: speech, video, ...

$$\text{Metric}_{fix} = \frac{\text{Power}}{\text{Throughput}} = \frac{\text{Energy}}{\text{Operation}}$$

- A lower value means energy savings in the systems.
- A system twice energy efficient doubles the battery life time. It can be obtained, for example, reducing at half the energy per operation.

SOCRATES'04 – Joan Oliver

## Energy efficiency-II

#### ■ Maximum throughput mode

- In most multi-user systems the processor is continuously running and requires fast computation at the maximum throughput mode.
- Examples: mainframes, networked desktop computers, ...
- Energy metric of energy efficiency must balance the need for low energy/operation and high throughput.
- Energy to Throughput Ratio...

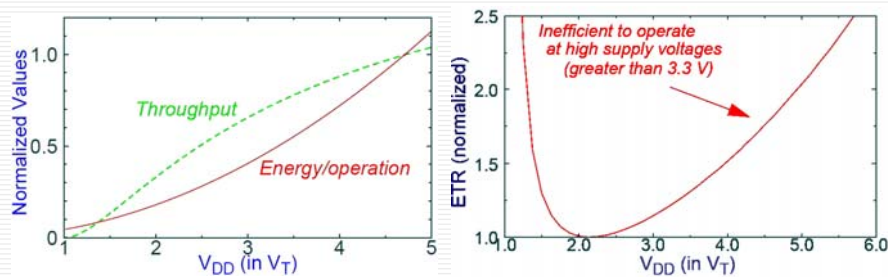
$$\text{Metric}_{MAX} = \text{ETR} = \frac{E_{MAX}}{T_{MAX}} = \frac{\text{Energy/Operation}}{\text{Throughput}} = \frac{\text{Power/Throughput}}{\text{Throughput}} = \frac{\text{Power}}{\text{Throughput}^2}$$

#### ■ So, lower ETR means better energy-efficient solution:

- lower energy/operation for a same throughput or
- more throughput for the same energy/operation
- In front of energy-delay metric, ETR can include the effects of parallelism when the delay is taken to be the critical path delay

SOCRATES'04 – Joan Oliver

## Energy efficiency-III



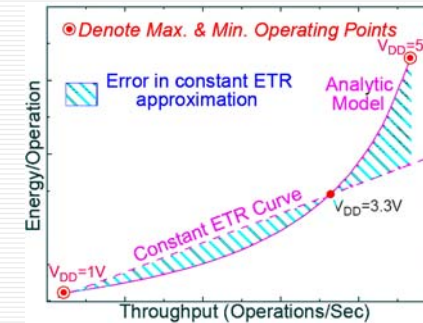
Energy/operation, Throughput vs.  $V_{DD}$

ETR as a function of  $V_{DD}$

- ETR varies with  $V_{DD}$ :  $V_{DD}$  can be adjusted by a factor of almost three ( $1.4V_t$  to  $4V_t$ ), with a variation on ETR of 50% over the minimum at  $2V_t$ .
  - For supply voltages over 3.3V there is a rapid degradation in energy efficiency
  - Energy efficiency also degrades for supply voltages approaching the device threshold

## Energy efficiency-IV

- Energy versus throughput (while varying power supply) plot allows to compare designs over a large range of operation.
- Plot is useful in designs optimised for a vastly different values of throughput.

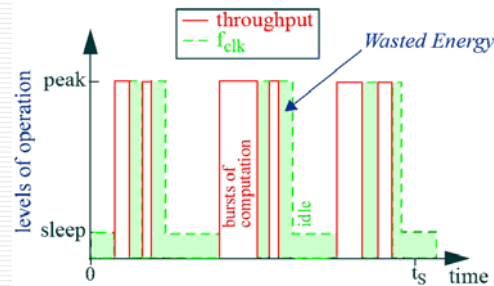


Energy vs. Throughput metric

## Energy efficiency-V

### ■ Burst throughput mode

- Most single-user systems (stand-alone desktop computers, PDAs, ...) spend a fraction of the time performing useful computation, while the rest of the time are idling between processes.
- But, when bursts of computation is demanded, the faster the throughput, the better.
- Then, energy-efficiency metric must balance the desire to minimise energy consumption (at both consumption and idling) and to maximise peak throughput when computing.



Wasted energy due to idle cycles

## Energy efficiency-VI

- In ideal processor: clock tracks computation periods to shut off when entering into idling
  - ETR will measure only wasted energy in computation periods.
  - (In reality many energy-saving systems only reduces clock activity, or support simple clock reduction/deactivation modes)
- Energy consumption (shaded areas are idling cycles) can be find as:

$$E_{MAX} = \frac{\text{TotalEnergyConsumingComputing}}{\text{TotalOperations}}, E_{IDLE} = \frac{\text{TotalEnergyConsumingIdle}}{\text{TotalOperations}}$$

assuming a processor shut down clock during idling time.

### □ Considering

$t_s$  = A large sample time period over which total operations and total energy are calculated.

$T_{MAX}$  = Peak throughput of the bursts of computation.

$T_{AVE}$  = Total Operations/ $t_s$  = Time-average throughput

$$\text{Metric}_{BURST} = M_{ETR} = \frac{E_{MAX} + E_{IDLE}}{T_{MAX}}$$

## Energy efficiency-VII

- $E_{MAX}$  is the ratio of power computed to peak throughput  $T_{MAX}$ . Thus, it is hardware dependent and, so, measurable at full processor use.
- $E_{IDLE}$  can be calculated as a function of the idle power dissipation:

$$E_{IDLE} = \frac{[IdlePowerDissipation][TimeIdling]}{[AverageThroughput][SampleTime]}$$

- Defining Power Down Efficiency as

$$\beta = \frac{PowerDissipation\ while\ Idle}{PowerDissipation\ while\ Computing} = \frac{P_{IDLE}}{P_{MAX}}$$

- Then

$$E_{IDLE} = \frac{[\beta E_{MAX} T_{MAX} [1 - T_{AVE}/T_{MAX}] t_s]}{[T_{AVE}][t_s]}$$

$$METR = ETR \left[ 1 + \beta \left( \frac{T_{MAX}}{T_{AVE}} - 1 \right) \right], \text{ with } T_{MAX} \geq T_{AVE}$$

SOCRATES'04 – Joan Oliver

## Energy efficiency-VIII

- METR is a good metric of energy efficiency for all values of  $T_{AVE}$ ,  $T_{MAX}$  and  $\beta$ .

- For idle energy consumption is negligible

$$\beta \ll T_{AVE}/T_{MAX} \rightarrow T_{AVE} = T_{MAX}$$

- For idle energy consumption dominating.

energy consumption should increase by either reducing the idle energy/operation while maintaining constant throughput, or by increasing the throughput while keeping idle energy/operation constant

$$\beta \gg \frac{T_{AVE}}{T_{MAX}} \rightarrow \frac{E_{IDLE}}{E_{MAX}} \cong \frac{P_{IDLE}/T_{AVE}}{P_{MAX}/T_{MAX}} = \beta \frac{T_{MAX}}{T_{AVE}}$$

- If  $\beta$  remains constant for varying throughput (and  $E_{MAX}$  remains constant), then  $E_{IDLE}$  scales with throughput as shown in equation.

SOCRATES'04 – Joan Oliver

## Design principles I

- High performance is energy efficient

- R4700, ARM710 processors: similar 0.6 $\mu$ m technologies
- Typical metric for measuring energy efficiency is SPECint92/Watt
- it seems that the ARM710 is *5 times as energy efficient* as R4700. But its performance is only the 15% of the R4700, and the ETR metric indicates that the R4700 is actually more energy efficient than the ARM710

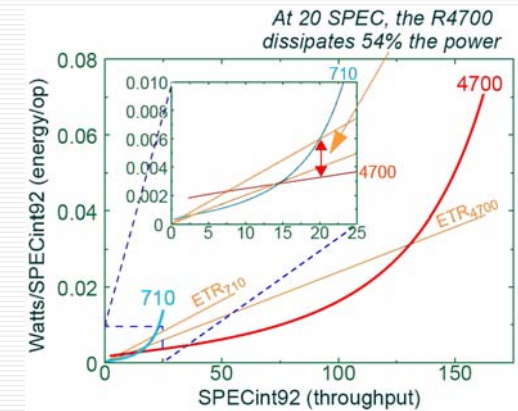
Processor	SPECint92 ( $T_{MAX}$ )	Power (Watts)	Supply voltage, $V_{DD}$ (volts)	SPECint92/Watt ( $1/E_{MAX}$ )	ETR ( $10^{-3}$ )
R4700	130	4.0	3.3	33	0.24
ARM710	20	0.12	3.3	167	0.30

SOCRATES'04 – Joan Oliver

## Design principles II

- Energy/operation versus throughput plot

- Despite the low VDD (1.5V<sub>T</sub>) of the R4700, at 20 SPECint92 dissipates 65mW, about 1/2 the ARM710's power. The R4700 can deliver 30 SPECint92 at 120mW ( $V_{DD}=1.7V_T$ ), a 150% ARM710's throughput
- If the lower bound in operating supply voltage is greater than 1.7V<sub>T</sub>, then ARM710 is more energy-efficient
- Typically, a processor is rated to operate at standard supply voltages (3.3V or 5.0V). But the processor can operate at a non-standard supply voltage by using a high-efficiency, low-voltage DC-DC converter



Energy/operation versus throughput plot

SOCRATES'04 – Joan Oliver

## Design principles III

## □ Fast operation can decrease energy-efficiency

□ At fast response time, rather than reducing the voltage, the processor can be left at the nominal supply voltage, and shut down when it is needed

□ For example, assuming a target application with  $T_{AVE}=20$  SPEC, with a  $\beta = 0.2$  factor for both processors and  $V_{DD}=3.3V$ . Then

- $METR_{ARM710} = ETR_{ARM710} = 3.0 \cdot 10^{-4}$ . It remains the same because it never idles
- The R4700 spends 85%  $(1-T_{AV}/T_{MAX})$  of the idle time, with  $METR_{R4700} = 5.0 \cdot 10^{-4}$

□ So, the ARM is more efficient!

□ But, with a  $\beta = 0.02$ , then  $METR_{R4700} = 2.66 \cdot 10^{-4}$ , and again is more energy-efficient. For this example,  $\beta_{crossover} = 0.045$

## Design principles IV

## □ Clock frequency reduction is not energy efficient

□ Not always a reduction on clock frequency is energy-efficient. If compute energy consumption dominates, consumption is quite the opposite!

□ Since compute energy consumption is independent of  $f_{clk}$  and throughput scales proportionally with  $f_{clk}$ ,

- when compute energy consumption dominates ( $E_{MAX} \gg E_{IDLE}$ ): decreasing  $f_{clk}$  increases ETR. That is, energy-efficiency decreases. For example, halving  $f_{clk}$  is equivalent to doubling the computation time, while maintaining constant computation per battery life!
- when idle energy consumption dominates ( $E_{MAX} \ll E_{IDLE}$ ): Clock reduction may trade-off throughput and energy/operation when PowerDown efficiency  $\beta$  remains independent of throughput. Then  $E_{IDLE}$  scales with throughput. That is, halving  $f_{clk}$  will double the computation time, but will also double the amount of computation per battery life. But, if  $\beta$  is inversely proportional to throughput, the reduction of  $f_{clk}$  does not affect the total energy consumption, and the energy efficiency drops.

## Impact of clock frequency reduction on energy efficiency

Operating Conditions:	Compute Energy Consumption Dominates	Idle Energy Consumption Dominates	
		$\beta$ independent of throughput	$\beta$ inversely proportional to throughput
Throughput	decreases	decreases	decreases
Energy	unchanged	decreases	unchanged
Energy Efficiency (METR)	decreases	unchanged	decreases

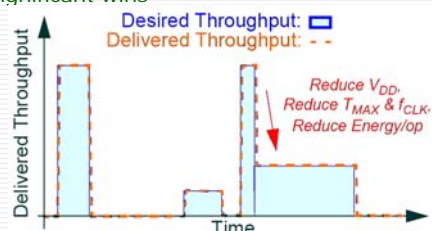
## Design principles V

## □ Dynamic voltage scaling is energy efficient

□ If  $V_{DD}$  tracks (dynamically during processor operation)  $f_{clk}$  (critical path delay inversely equal to clock frequency), energy efficiency could be maintained while varying  $f_{clk}$ . → Dynamic voltage scaling during processor operation.

□ If  $E_{IDLE}$  is present and dominates the total energy consumption, simultaneous  $V_{DD}$ ,  $f_{clk}$  reduction during periods of idle yield a more energy-efficient solution.

□ Even when idle energy consumption is negligible, dynamic voltage scaling can still provide significant wins



Dynamic voltage scaling

## Design principles VI

□ For applications that require maximum deliverable throughput only a small fraction of time, dynamic voltage scaling has a significant win. Next table compares the behaviour of the R4700 processor in several operating conditions. For each category of throughput the total number of operations completed remains the same. For simplicity, the example assumes that idle energy consumption is always negligible.

## Benefits of voltage scaling

Throughput:	Time spent operating in:			$T_{MAX}$ (SPECint92)	$E_{AVE}$ (W/SPECint92)	ETR (10 <sup>-9</sup> )	Normalized Battery Life
	Fast Mode	Slow mode	Idle Mode				
Always full-speed	10%	0%	90%	130	0.031	237	1 hr.
Sometimes full-speed	1%	90%	9%	130	0.006	45.0	5.3 hrs.
Rarely full-speed	0.1%	99%	0.9%	130	0.003	25.8	9.2 hrs.



## Energy-efficient design

- ETR is a valid energy-efficient metric including when idle energy consumption of the processor is negligible.
- Design techniques drawn from the literature that impact on ETR
- Energy-efficient VLSI design
  - At instruction set architecture level
    - 16-bit instruction word and register processors
    - Optimisation number of registers
    - Supported operation types and addressing modes
  - At the microarchitecture level
    - Concurrency, i.e. parallelism, pipelining.
    - Optimisation in caches.
    - Processor controls help energy-efficiency reduction
  - At the circuit design level
    - Design rules
- Energy-efficient software
  - In the operating system
  - In variable performance scheduling
  - Algorithms and compilers

## Energy-efficient VLSI design I

- At instruction set architecture level
  - For low-energy processors 16-bit instruction word and register processors are energy-efficient in front of 32-bit instruction registers:
    - Static code density reduction in a 30-35%, while increase in dynamic run length by 15-20%
    - Instruction-fetch energy-cost reduction of a 50% because memory size halves
    - With external busses, since instruction fetch is about one third of the processor's energy, total energy consumption is reduced in 15-20%, but also performance is reduced in 15-20%, which approximately equals energy efficiency
  - Optimisation number of registers
    - Cache rarely is energy-efficient in front of moderately sized (32) register-files
    - But in a register window, with very large (+100) register file, consumed energy increases dramatically, increasing total processor energy consumption in a 10-20%. Only energy-efficient if this increase can be compensated in an equivalent performance
    - There appears to be an optimum on the number of registers since the energy efficiency is near equal for 16 to 32 register file
  - Supported operation types and addressing modes
    - Complex ISAs (CISC's) have higher code density, which reduces the energy consumed fetching instructions and the total number of instructions executed
    - Simple ISAs (RISCs) typically have simpler data and control paths, which reduces the energy consumed per instructions, but there are more instructions

## Energy-efficient VLSI design II

- At the microarchitecture level
  - Energy-efficiency increase techniques in custom DSP ICs based in concurrency, i.e. parallelism, pipelining, ...: energy-efficiency improvement of approx. N on an N-way parallel/pipelining architecture
    - Moderate pipelining (4-5 stages) in RISC processors operating near one-cycle per instruction improves energy-efficiency in a factor of two or more
    - Limited energy-efficient improvements in superpipelined structures and superscalar architectures speedup increase due to the limited instruction-level parallelism of codes. Poor ETR improvement in many cases
    - VLIW's best exploits instruction-level parallelism of the hardware using specific compilers, with a speedup factor between 2 and 6, and energy efficiency improvements between 33% to 300%
  - Caches consume about a third of the processor's energy consumption. On-chip caches reduces energy/access and increments throughput
    - Proposed techniques that reduces the access to the instruction exploiting spatial locality increases processor efficiencies between 5-25%
  - Processors control can help energy-efficiency reduction
    - Disabling of pipelined stages not used in certain cycles. With a small overhead cost in superscalar architectures, energy-efficiency improvements could be of 15-25%
    - Use of clock gating. NOP instructions must be avoided.

## Energy-efficient VLSI design III

- At the circuit design level
  - ETR metric can be used to determine which proposed low-power techniques are energy-efficient.
  - There are a variety of energy-efficient design techniques at the circuit design level that can be introduced in order to help to globally improve processor's energy-efficiency

## Energy-efficient software

---

- Power down modes and halt instructions + energy-efficiency minded operating system
- Energy-efficient software optimisation
  - In the operating system
    - OS invokes halt instructions to disable the processor, and power on and off peripheral hardware components
    - Energy consumption savings up to 50%
  - In variable performance scheduling
    - Dynamic reduced supply voltage and clock frequency to just to meet the required throughput for each proces with lower performance demands.
    - Predictive scheduling for changing CPU performance over evaluation of activity CPU intervals
    - Key optimisation parameter in code is cycle count, and not instruction count
  - Algorithms and compilers
    - Algorithms traditionally tuned for high performance. But algorithm implementations with fewer operations increases throughput and consumes less energy, improving ETR until quatratic factors
    - The same improvement can be obtained using optimised compilers