Power and Energy Aware Computing

particularly important for Embedded Systems

A 5785 Whirlwind Tour

Al Davis

Agenda

- **CMOS digital centric view of the world**
  - nothing will displace CMOS in the next 10+ years
- **Power and energy are NOW the prime design constraints**
  - tutorial on basic issues
    - energy, power, CMOS, scaling properties, ….
  - caused fusion of embedded and high-performance design mindsets
- **What can be done**
  - past, present, and currently under research
    - process, circuit, architecture, and software communities
- **What should be done but isn't on the radar screen**

Computing's Catch 22

- **Unlimited expectation**
  - greed for speed
    - Moore's law scaling history is a major culprit
  - bounded only by human imagination
- **Limited physical resource budget**
  - cost in $, power density, battery energy, VLSI process, ….
  - bounded by several realities
    - physics
    - chemistry
    - economics
    - engineering state of the art

Design Constraint History

- **Correctness under escalating design complexity**
  - pre-mid-90's: a minor issue
  - now: a big problem which we'll ignore here
- **How big can chips be?**
  - area and transistor count have similar track records
    - pre Y2K: the primary physical constraint
    - now: not an issue
      - we have more transistors than we can use
      - largest dual core Itanium consists of 1.7×10^9 T's
- **Power and Energy**
  - post Y2K: the new prime constraint
  - hence the focus in this presentation

Another Constraint Viewpoint

- **The game hasn't changed**
  - achieve maximum performance with the available resources
  - performance definition is context dependent
    - usually it's about speed: latency or throughput
    - could be about other things like battery life
- **Basic trade-offs**
  - make one thing faster vs. doing multiple slower things in parallel
    - multiply in 3 GHz Pentium 4 is 10x larger than on a 400 MHz SA1100
  - find ways to be more efficient
    - process, circuit, architecture, software
  - adapt computational approach based on context and available resources
    - nobody really does this

Energy and Power

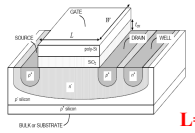
- **Energy – capacity for doing work**
  - essentially a force where 1 joule =
    - .737 ft-lb, .00095 BTU, .769 x 10^{-8} gal. of gas
  - how much can we compute with a joule
    - very dependent on process
      - scaling improves this dramatically
    - 22 picojoules/instruction on a 32-bit SA1100
    - 1 nanojoule/instruction on a 8-bit CoolRisc µcontroller
  - focus for battery powered devices
- **Power – rate of doing work or of using energy**
  - watt = joule/sec = volts * amperes
    - 1 kw-hr = 36 mega-joules
  - focus for the plugged in world
    - power density important for cooling
    - total power important for packaging and work efficiency

Cooling

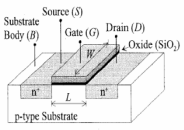
- **Basics**
  - convective (air cooling w/ heatsink) costs 1x
    - 100W cm2 possible with fancy bonding
  - 2 phase (boiling) costs 1.5 x
  - 2 phase turbulent costs 2-3x
  - spray cooling costs 4x

MOS Transistors

- **Digital view: a voltage controlled switch**
  - come in two varieties in CMOS

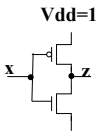


L=λ



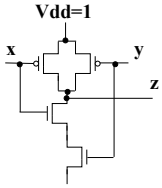
problem: scaling in horizontal dimension = scaling in vertical for the gate oxide

CMOS Logic



inverter

x	z
0	1
1	0



nand

x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

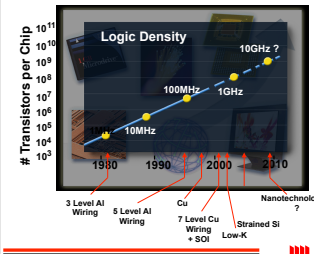
First Order Scaling Effects

- **Constant field model (constant voltage failed ages ago)**
  - gate delay α λ
  - DC power α $1/\lambda^2$
  - Dynamic power α $1/\lambda^2$
  - Power delay product α $1/\lambda^3$
  - Gate area α λ^2
  - Power density α 1 (doesn't consider leakage however)
  - Current density α $1/\lambda$ (assumes ideal wire scaling which is impossible)
  - wire resistance α $1/\lambda$ (the wire scaling issue)
  - local wire length α λ
  - global wire length doesn't change
    - die size is actually growing ➔ major power culprit

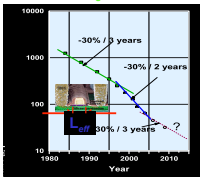
source: Weste and Eshraghian 2nd Ed.

Process Scaling

CMOS Device Performance Continues to Increase Due to:
- *Scaling Effects*
- *Non-Scaling Technology Enhancements*



Scaling Effects



Non-Scaling Technology Enhancements


Wiring Levels & Metallurgy (Cu)

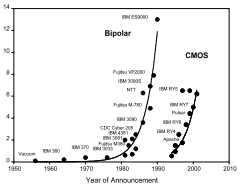

SOI

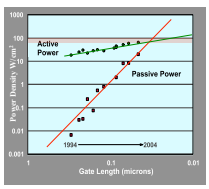

Strained Silicon


Low-K Dielectric

Power Wall

Challenge 1 Power Wall





Gate Stack

Gate dielectric approaching a fundamental limit (a few atomic layers)

- Power density near ceiling; power near limit for all but the largest systems
- Leakage limits scaling, no longer smaller = faster/less power, technology slows down
- **Consequence: Increased performance requires increasing efficiencies from design**
  - Lower power circuits (switch less, leak less, design for low supply voltage)
  - More efficient architectures (switch less for the same function)

CMOS Power Synopsis

- **Pactive = aCV²F (C associated with wires and Tgate)**
- **Leakage = α*Total-Active-Area**
  - α increases 10x every step given same materials
- **Ideally**
  - C and V scale as λ
    - in practice V scales slower
- **Ideally for the same chip**
  - Power scales
    - as λ^3 if run at the same frequency
    - as λ^2 if run at $1/\lambda$ faster
- **But reality isn't ideal**
  - leakage and #T's goes up exponentially
- **How do we deal with the current power wall?**

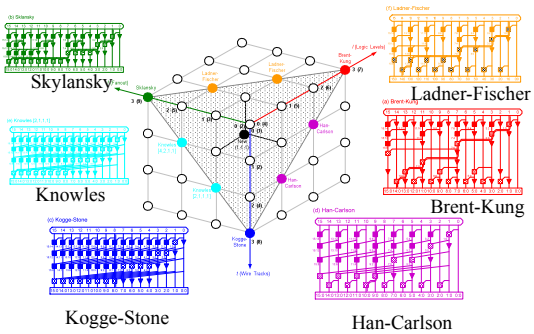
Process Solutions

- **New materials**
  - wires, vias, gate oxide, base material (Si, SiGe, 3-5)
  - all one trick ponies
    - unclear whether much can be expected here
- **Multiple transistor types**
  - fast leaky transistors for the critical path
  - slow less leaky transistors for the rest
  - complicates the process and increases device cost
- **New transistor structures**
  - MIGFET
  - FinFET

Circuit Design Solutions

- **Pareto analysis of circuit options – power delay product**
  - e.g. use fast power hungry circuits only where you need them
  - no longer go with the fastest design
- **Turn off things that aren't needed**
  - clock gating
  - standby and sleep modes
- **Dynamic frequency and voltage scaling**
- **Dynamic body bias**

Circuit Pareto: Fast Prefix Adders

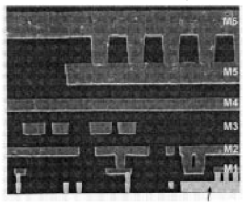


The Future of Wires

- **Common belief**
  - wires scale badly w.r.t. gate speed
- **A rosier picture is**
  - must assume 2 things
    - high-k dielectrics – Intel supposedly has this one in the bag
      - but the claim is widely disputed
    - higher aspect ration wires
      - Pat Bosshart's 2x4
      - problem: much harder to make as technology scales
- **Examine the data**
  - .25 micron process
  - M1 is lowest and finest pitch
  - bigger layers have less resistance since they are bigger

Typical Wire Pitch

FIGURE 1. A 0.25μm technology with typical pitches



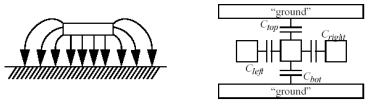
Typical metal pitches

Layer	Width and spacing
M5-6	8 λ and 8 λ
M2-4	4 λ and 4 λ
M1	3 λ and 2 λ

Picture source: N. Rohrer, ISSCC 1998

A View of Wire C's

FIGURE 2. Isolated and realistic on-chip capacitance models



source: Horowitz "Future of Wires"

R & C

- **R**
  - Al
    - 3.3 mOhm/cm
    - vias are tungsten and are highly resistive M1-M2 = 5 ohm
  - Cu
    - 2.2 mOhm/cm
    - vias are copper ➔ BIG win
    - much better electromigration properties than Al
- **C**
  - tall thin wires
    - side to side capacitance is a bigger piece of the action
    - side C's also greatest culprit for noise injection
    - delay will be data dependent ➔ need to play worst case games
      - middle goes same way ➔ C=0
      - middle goes opposite way of both sides = C

Wires by Layer

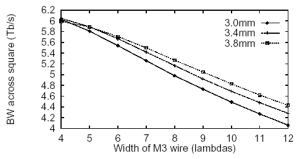
0.25µm tech	M1	M2-M4	M5-M6
Width, µm	0.375	0.5	1.0
Spacing, µm	0.25	0.5	1.0
Height, µm (1.8 aspect ratio)	0.675	0.9	1.8
Resistance (Al), Ω/mm	130	73	18
Resistance (Cu), Ω/mm	108	57	13
Capacitance, fF/mm (ε_r=3.9)	296	230	230
% of Cap is Xcap	78%	69%	69%
Wire delay (Al), FO4/mm^2	0.21	0.09	0.02
Wire delay (Cu), FO4/mm^2	0.18	0.07	0.01

Speed of Wires

- **Note F04 basis for everything**
  - Horowitz and Harris metric
- **Model**
  - wait 3 propagation delays before sending the next signal
  - allows output to transition past the 90% mark
  - conservative model but unlikely you can beat this by 2x
- **Equation**

$$BW_{area} = \frac{1}{3FO4 + 1.2R_{wire}C_{wire}} \cdot \frac{areawidth}{wirepitch}$$

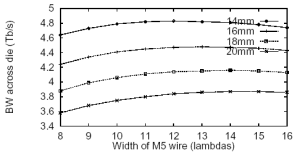
Wire Bandwidth vs. Size



source: Horowitz "Future of Wires"

Wire speed α RC – R is per square – big wires are faster

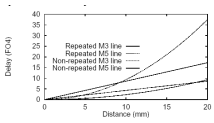
Die Bandwidth



source: Horowitz "Future of Wires"

Repeaters

- **Wire delay**
  - quadratic with length
- **Repeaters**
  - adds gate load to each segment wire
  - but breaks the quadratic delay dependence

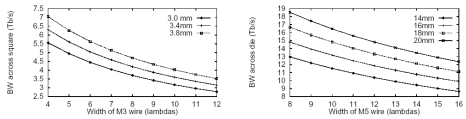


Repeater Complications

- **Floor planning**
  - remember – only useful for long wires
  - use of vias is likely
    - causes route blockage
- **Device gain**
  - → large devices
  - for a wide bus → greatly increases bus pitch

Repeated Wire Bandwidth

FIGURE 5. Bandwidth of repeated M3 and M5 unloaded lines



source: Horowitz "Future of Wires"

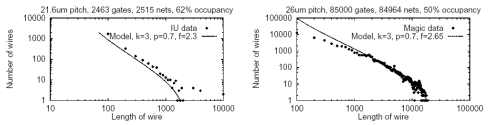
Power and Noise

- **Power**
  - P=CVΔVf
    - V = supply
    - ΔV = the swing
    - C is the line capacitance
    - f is the effective switching frequency
- **Injected noise from the coupling (side) capacitance**
  - Vpeak = ΔV*Ccoup/Ctot
    - too pessimistic since it assumes the attacker is strong and the victim is weak (undriven)
  - more realistic is based on time constants of attacker and victim

$$V_{peak} = \Delta V \cdot \frac{C_{coup}}{C_{tot}} \cdot \frac{1}{1 + \tau_{att}/\tau_{vic}}$$

Empirical Wire Lengths

FIGURE 6. M-Machine and Magic/Flash wire load models



source: Horowitz "Future of Wires"

Wires and the ITRS Roadmap

L_{drawn}	0.25µm	0.18µm	0.13µm	0.10µm	0.07µm	0.05µm
Mid-layer width (4λ) in mµ	0.50	0.36	0.26	0.20	0.14	0.10
Global layer width (8λ) in µm	1.0	0.72	0.52	0.40	0.28	0.20
Chip edge length, mm	17.3	19	20.7	22.8	24.9	27.4
FO4 delay, pS	90	65	48	36	25	18
Frequency at 16 FO4s, GHz	0.7	1	1.3	1.7	2.5	3.5

Roadmap R's and C's

TABLE 1. Conservative scaling

L_{drawn}	0.25µm	0.18µm	0.13µm	0.10µm	0.07µm	0.05µm
Mid-layer R, Ω/µm	0.07	0.11	0.19	0.32	0.65	1.29
Global layer R, Ω/µm	0.018	0.026	0.044	0.074	0.15	0.30
Mid/Global C, fF/µm	0.23	0.22	0.22	0.20	0.19	0.17

TABLE 2. SIA roadmap scaling

L_{drawn}	0.25µm	0.18µm	0.13µm	0.10µm	0.07µm	0.05µm
Mid-layer R, Ω/µm	0.07	0.11	0.18	0.26	0.39	0.70
Global layer R, Ω/µm	0.018	0.025	0.042	0.061	0.091	0.16
Mid/Global C, fF/µm	0.23	0.19	0.18	0.16	0.16	0.16

source: Horowitz "Future of Wires"

Delays and Scaling

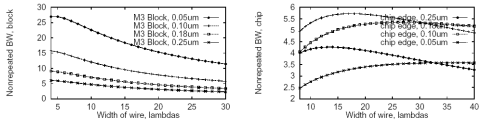
TABLE 3. Delays under scaling

L_{drawn}	0.25µm	0.18µm	0.13µm	0.10µm	0.07µm	0.05µm
50K block semi-perim length, mm	3580	2500	1750	1315	990	740
Conservative 50K block delay, FO4	1.2	1.2	1.4	1.5	2.3	3.4
SIA 50K block delay, FO4	0.94	1.02	1.06	1.03	1.21	1.76
Chip edge, cm	17.3	19	20.7	22.8	24.9	27.4
Conservative chip edge delay, FO4	7.0	15.3	43.7	107	345	1073
SIA chip edge delay, FO4	5.1	13	34	72	180	560

source: Horowitz "Future of Wires"

Medium (50K gates on M3) vs Global BW Scaling

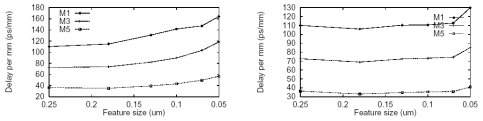
FIGURE 7. Unrepeated bandwidth: Mid-level wires on a block and global wires on a chip



source: Horowitz "Future of Wires"

Repeated Wire Delay

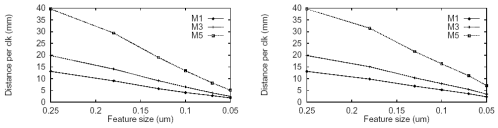
FIGURE 8. Repeated wire delay (conservative and SIA scaling)



source: Horowitz "Future of Wires"

Distance traveled per clock

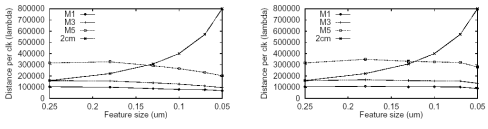
FIGURE 9. Distance travelled per clock cycle



source: Horowitz "Future of Wires"

The Punch Line

FIGURE 10. Relative distance travelled per clock cycle

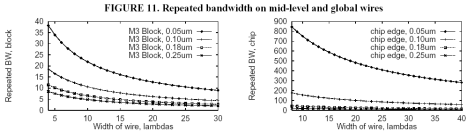


the amount of λ's you can get to is relatively constant

source: Horowitz "Future of Wires"

It's the big wire distances that kill you!!

FIGURE 11. Repeated bandwidth on mid-level and global wires

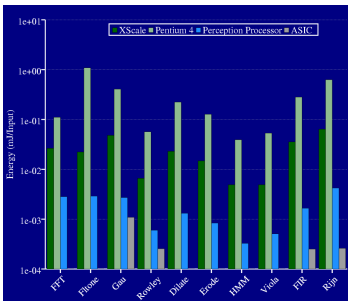


source: Horowitz "Future of Wires"

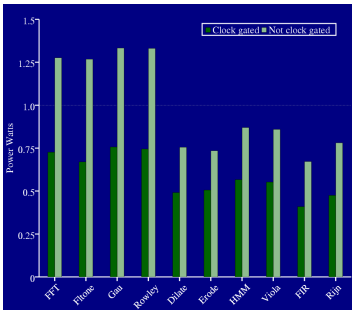
Architecture Implications

- **Clocks**
  - aren't really the culprit except for power\
    - power is a huge deal however
    - clock gating helps a lot but adds both skew and jitter to the margins
  - even though they are long wires
- **Long wires are evil**
  - it takes 2 clocks to get across a Pentium IV
    - register in the middle since a wire is now part of the pipeline
    - → forwarding now becomes inter- vs. intra-block
      - → clustered execution units
- **Communication management becomes critical**
  - floor planning happens early in the design cycle
  - new architectures may help things
    - MIT's RAW, Stanford's Smart Memory (evolved Imagine)
    - Utah's fine-grain perception clusters

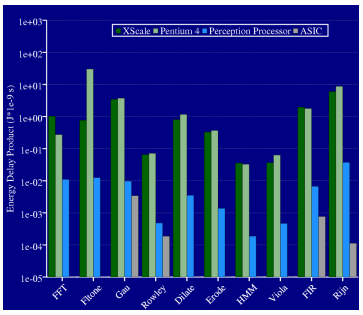
Results: Energy



Results: Clock Gating Synergy



Results: Energy Delay Product



Specialization

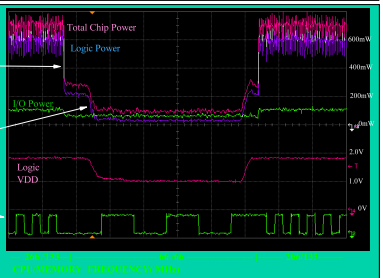
- **Key to energy efficiency**
  - GP CPU – lots of overhead/work
  - DSP's – less overhead/work 10x better energy-delay product
  - ASICS – not GP but highly specialized 10,000x better energy delay product
    - iPhone: ARM GP core surrounded by 50-60 ASIC IP blocks
      - video & audio codec's
      - rake, turbo, … for RF communication
      - plus others – check out Anandtech.com for details
  - DSA's – 100 – 1000x better energy delay and more flexible than ASICS

Active Power Management : Run Mode Dynamic Voltage and Frequency Scaling

Dynamic Frequency Scaling
266Mhz CPU to 66MHz CPU

Dynamic Voltage Scaling
1.8V -> 1.0V at upto 1V/100us

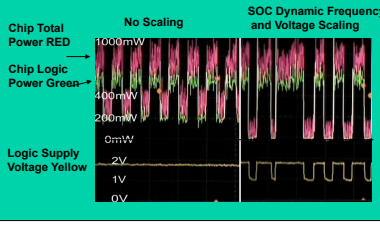
Uninterrupted Operation
Linux 2.3.17 Running
Dhrystone 2.1 code
400 loops per cycle .



Power consumption for the CPU and logic was reduced by 13X dynamically under the control of the Linux kernel
(NO PLL Relock and NO stopping of the application)

DVS & DFS for PPC405LP

Benefits in active power reduction can be seen in important and demanding applications like MPEG 4 video decoding



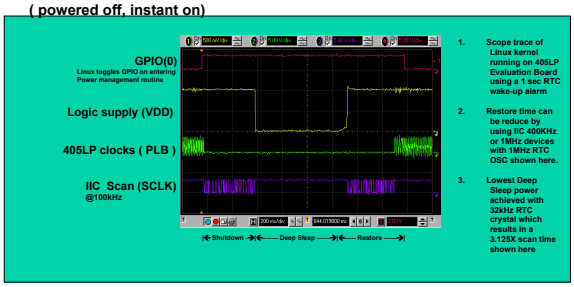
405LP Running MPEG4 Player At 30 FPS

SOC operating point is dynamically scaled between frames

25 to 30 % Power Reduction in average Power

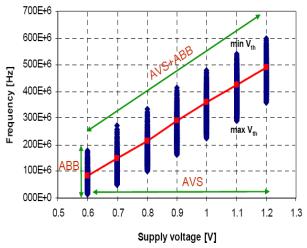
PPC405LP Deep Sleep

(powered off, instant on)



1. Scope trace of Linux kernel running on 405LP Evaluation Board using a 1 sec RTC wake-up alarm
2. Restore time can be reduce by using IIC 400KHz or 1MHz devices with 1MHz RTC OSC shown here.
3. Lowest Deep Sleep power achieved with 32kHz RTC crystal which results in a 3.125X scan time shown here

Adaptive Body Biasing



Architecture Tactics Gone Bad

- **Pipelining**
  - performance mechanism of the 80's
- **Super Pipelining**
  - to ramp frequency as aggressively as you can get away with
    - 50% of Alpha's power went into the clock tree
  - marketing darling of the last 15 years or so
- **Superscalar & dynamic out of order execution**
  - performance mechanism of the 90's
- **Speculation**
  - performance mechanism of the (20)00's
- **Need for long wires**
  - long wires scale badly with respect to transistors
  - high power drain OR huge performance bottleneck

Architecture Solutions

- **Multi-Core (the main merchant game of today)**
  - multiple simpler processors on a chip
    - both heterogeneous and homogeneous options exist
  - gotchas
    - what to do with memory
    - finally the programming community is forced to change
      - thread level parallelism is there
      - how to use it is a SMOP
- **Specialization (the embedded game past and present)**
  - custom pipelines for key kernels
    - often cast as an ASIC coprocessor ➔ inflexibility problem
  - avoids significant overhead of a RISC instruction
- **Hybrid options**
  - specialize and turn things off when they're not needed

Software Solutions

- **Sadly there is little to report to date**
  - OS, compilers, & applications haven't changed much
  - exception is the embedded space & research
    - compiling for energy/power efficiency is a hot topic
    - OS scheduling under power/energy constraints is being studied
    - some applications have been studied in a power/energy aware fashion
      - e.g. the AES competition
      - Rijndahl chosen for variety of platform implementations
- **Current big scare**
  - how does the application community react to take advantage of multicore devices
    - jury is still out
  - point success
    - IBM Cell processor adopted for Sony's Playstation 3

Key Research Results

- **Giving the programmer control over physical resources is a win**
  - RAW, APE, ACT, Imagine, ….
- **Non Von Neumann programming models enable new levels of efficiency**
  - Streamit, Brook, dataflow, functional (ML, Haskell)
- **Synthesizing only what you need is a win**
  - Tensilica
  - ASIC world in general
- **Problem**
  - all have limits in where they apply
- **Opportunity**
  - architectures which have many specialized components
  - turn off the stuff you don't need – question is where is the control

Untouched Territory

- **Integrated approach to energy and power aware computing**
  - circuits which report their consumption rates
    - and potentially adapt when signaled to do so by the OS or application
  - architectures with multiple resource target options
  - OS schedules based on application demands and circuit information
    - also makes consumption data visible at the application level
  - compilers that use energy or power in their cost functions
  - applications which are capable of adapting to available energy/power
    - note: some media application research considers this but the decision is not based on circuit information
- **Break the von Neumann model**
  - multiple tiles but what should the tiles look like? memory? ….
- **Probabalistic circuits**
  - to date we've had the luxury of living in a deterministic world
  - this will change in approximately 10 years
  - how do we design reliable systems from unreliable components?

Conclusion

- **It's a brave new world out there**
- **Challenge is at all levels**
  - process, circuit, architecture, OS, compiler, application
  - clear that process benefit is slowing
- **Problem**
  - each camp is taking an evolutionary approach
    - no point in making a machine that nobody can program
    - researchers have been good at this
  - unlikely that all camps can march in a consistent direction
- **Need new abstractions**
  - provides application area independence
  - inspires new ways to think about things