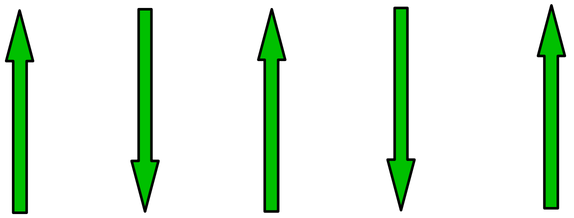
Tutorial Outline

8:30 - 8:45	Introduction and motivation
8:45 - 9:05	Sources of power in CMOS designs
9:05 - 9:30	Power analysis tools and techniques
9:30 - 10:30	Gate & functional unit design issues & techniques
10:30 - 10:50	BREAK
10:50 - 12:15	Architectural level issues and techniques
12:15 - 1:30	LUNCH
1:30 - 2:30	Low power memory system design
2:30 - 3:30	Software level issues and techniques
3:30 - 3:50	BREAK
3:50 - 4:30	Software level issues and techniques, con’t
4:30 - 4:45	Future challenges

Energy Analysis Techniques

- Simulation based techniques
  - » characterization - using lower level analysis results to construct higher level models
  - » very computational intensive; cycle accurate
- Probabilistic based techniques
  - » signals viewed as random zero-one processes with certain statistical characteristics
  - » computationally efficient; average values for sequences of cycles
  - » large errors are possible if statistical assumptions are incorrect

Analysis Abstraction Levels

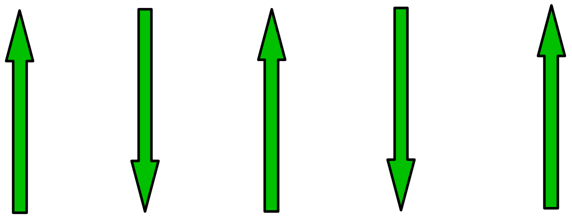


Abstraction Level	*Analysis Capacity*	*Analysis Accuracy*	*Analysis Speed*	*Analysis Resources*	*Energy Savings*
	Most	Worst	Fastest	Least	Most
Application					
Behavioral					
Architectural (RTL)					
Logic (Gate)					
Transistor (Circuit)					
	Least	Best	Slowest	Most	Least

Circuit Level Simulation

- Extract circuit netlist description from layout
  - » captures internal (diffusion) and external (wiring and gate fanout) capacitances
- Run an analog simulation
  - » characterization of device models (nfets, pfets)
  - » solution of large system of equations so very computationally intensive (< few thousand transistors)
  - » can accurately estimate (within a few %) dynamic and leakage power dissipation
- HSPICE (Avant!), spectre (Cadence), PowerMill (Synopsys)

Analysis Abstraction Levels



Abstraction Level	*Analysis Capacity*	*Analysis Accuracy*	*Analysis Speed*	*Analysis Resources*	*Energy Savings*
	Most	Worst	Fastest	Least	Most
Application					
Behavioral					
Architectural (RTL)					
Logic (Gate)					
Transistor (Circuit)					
	Least	Best	Slowest	Most	Least

Gate Level Simulation

- Perform logic simulation to obtain the switching events for each net (signal)
  - » logic description in structural VHDL or verilog
  - » zero-delay or unit-delay timing models
- Determine frequency of each net $f_y = t_y/(2T)$, where t_y is the number of logic switches of net *y* and T is the simulation time, to compute *dynamic* power

$$P_{dyn} = \Sigma C_y V_{DD}^2 f_y$$

- Pre-layout so must estimate C_y

Capacitance Estimation

- Device (diffusion and gate) capacitance
  - » depends on width/length of driving gate's source/drain diffusion and fanout gates
  - » part of characterization of cell based designs
- Wiring capacitance
  - » depends on placement and routing
  - » wire load - predict wire length of a net from the number of pins incident to the net
    - – mapping table can be constructed from historical data of existing designs

---

Gate Internal and Leakage Power

- Use gate characterization ($E(g,e)$) and logic simulation event count ($f(g,e)$) to calculate the gate's dynamic *internal* power (short circuit and charging/discharging of internal capacitors)

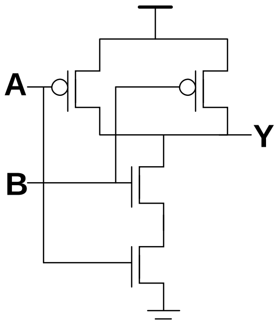
$$P_{int} = \sum\sum E(g,e)\, f(g,e)$$

- During simulation record the fraction of time $T(g,s)/T$ that each gate g stays in a particular state s to calculate *leakage* power

$$P_{leak} = \sum\sum P(g,s)\, T(g,s)/T$$

Characterization Tables Example

- Internal energy events



A	B	Y	E(g,e) (pJ)
1	r	f	1.67
1	f	r	1.39
r	1	f	1.94
f	1	r	1.72

- Leakage power

A	B	Y	P(g,s) (pW)
0	0	1	5.05
0	1	1	13.1
1	0	1	5.10
1	1	0	28.5

Gate Level Simulation Considerations

- Simulation vectors need to be chosen carefully (application dependent)
- Internal power really depends on operating voltage, temperature, process,… → multi-dimensional characterization
- Accuracy within 5-10% of SPICE
- Signal glitches may not be modeled precisely (glitches depend on delays in the circuit)

Probabilistic Gate Level Power Analysis

- For each internal net *y* determine the signal probability of the net wrt to the given signal probabilities of the primary inputs
- From the signal probabilities determine the transition density D(y) of each internal net *y*
- Compute the total power

$$P = \Sigma\, 0.5\, C_y\, V_{DD}^2\, D(y)$$

- Pre-layout so must estimate C_y

Determining Signal Probabilities

- Signal probability definition

$$P_1 = t_1/(t_0 + t_1) \quad \text{and} \quad P_0 = 1 - P_1$$

- Propagate the given statistical quantities from the primary inputs to the internal signal nets and outputs of the circuit
- Propagate quantities using probabilistic signal propagation model

Signal Propagation Model

- Apply Shannon's decomposition to the *n*-input Boolean function $y = f(x_1, \dots x_n)$
  $$y = x_i\, f_{xi} + !x_i\, f_{!xi}$$
  where f_{xi} ($f_{!xi}$) is the new Boolean function obtained by setting $x_i = 1$ ($x_i = 0$) in $f(x_1, \dots x_n)$

$$\boxed{P(y) = P(x_i f_{xi}) + P(!x_i f_{!xi}) = P(x_i)P(f_{xi}) + P(!x_i)P(f_{!xi})}$$

- Apply recursively (note: $P(!x_i) = 1 - P(x_i)$)

Signal Probability Example

- Find output probability of $y = ab + c$

$$\begin{aligned}P(y) &= P(a)P(b+c) + P(!a)P(c)\\ &= P(a)[P(b) + P(!b)P(c)] + P(!a)P(c)\\ &= P(a)[P(b) + (1 - P(b))P(c)] + (1 - P(a))P(c)\\ &= P(a)P(b) + P(a)P(c) - P(a)P(b)P(c)\\ &\qquad + P(c) - P(a)P(c)\\ &= P(a)P(b) + P(c) - P(a)P(b)P(c)\end{aligned}$$

- Given $P(a)=0.2$, $P(b)=0.3$, $P(c)=0.4$
  then $P(y) = 0.2 \times 0.3 + 0.4 - 0.2 \times 0.3 \times 0.4 = 0.436$

Determining Transition Density

- For a transition (1-to-0 or 0-to-1) to have occurred $f_{x_i} \oplus f_{!x_i} = 1$ the Boolean difference of y wrt x_i denoted dy/dx_i
- $P(dy/dx_i)$ is the probability that dy/dx_i evaluates to 1 and $D(x_i)$ is the transition density of x_i
- Then the total transition density of the net y is

$$\boxed{D(y) = \Sigma\, P(dy/dx_i)\, D(x_i)}$$

Transition Density Example

For $y = ab + c$

$$dy/da = (b+c) \oplus c = b!c$$
$$dy/db = (a+c) \oplus c = a!c$$
$$dy/dc = 1 \oplus ab = !(ab)$$

so

$$P(dy/da) = P(b)\,[1 - P(c)] = 0.18$$
$$P(dy/db) = P(a)\,[1 - P(c)] = 0.12$$
$$P(dy/dc) = 1 - P(a)P(b) = 0.94$$

Given $D(a)=1$, $D(b)=2$, $D(c)=3$

$$D(y) = (0.18x1)+(0.12x2)+(0.94x3)=3.24$$

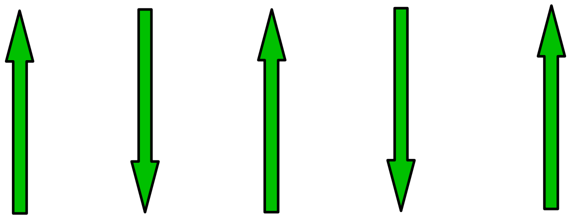
Probabilistic Gate Level Analysis Considerations

- Computationally efficient
  - » must compute signal probabilities and transition densities for each net to evaluate

$$P = \Sigma\, 0.5\, C_y\, V_{DD}^2\, D(y)$$

- Assumes given correct signal probabilities for primary inputs (and if wrong, large errors are possible)
- Gives average power dissipation values

Analysis Abstraction Levels



Abstraction Level	*Analysis Capacity*	*Analysis Accuracy*	*Analysis Speed*	*Analysis Resources*	*Energy Savings*
	Most	Worst	Fastest	Least	Most
Application					
Behavioral					
Architectural (RTL)					
Logic (Gate)					
Transistor (Circuit)					
	Least	Best	Slowest	Most	Least

Architectural Level Simulation

- Perform RTL simulation to obtain the input activity for each major functional unit
  - » architectural description in behavioral VHDL or verilog or C, C++
- Energy characterization of functional units
  - » analytical energy models
    - – caches, DRAMs
  - » transition sensitive energy models
    - – system buses
    - – ALUs, register file, pipeline registers

Analytical Energy Model Example

- On-chip cache

Energy = Ebus + Ecell + Epad + Emain

…

Ecell = β*(wl_length)*(bl_length+4.8)*(Nhit + 2*Nmiss)

wl_length = m*(T + 8L + St)

bl_length = C/(m*L)

Nhit = number of hits; Nmiss = number of misses; C = cache size; L = cache line size in bytes; m = set associativity; T = tag size in bits; St = # of status bits per line; β = 1.44e-14 (technology parameter)

Transition Sensitive Energy Model

- Must first design and layout a functional unit and then simulate it to capture switch capacitances
  - » bit independent – bus lines, pipeline registers
    - – one bit switching does not affect other bit slices' operations
  - » bit dependent – ALU, MAC, decoders
    - – one bit switching does affect other bit slices' operations
- Once constructed, the models can be reused in simulations of other architectures built with the same technology

---

Switch Capacitance Table

Previous Input Vector	**Current Input Vector**	**Switch Capacitance**
0 . . . 0 0	0 . . . 0 0	$\text{cap}_{0\rightarrow0}$
0 . . . 0 0	0 . . . 0 1	$\text{cap}_{0\rightarrow1}$
1 . . . 1 1	1 . . . 1 1	$\text{cap}_{2^n-1\rightarrow2^n-1}$

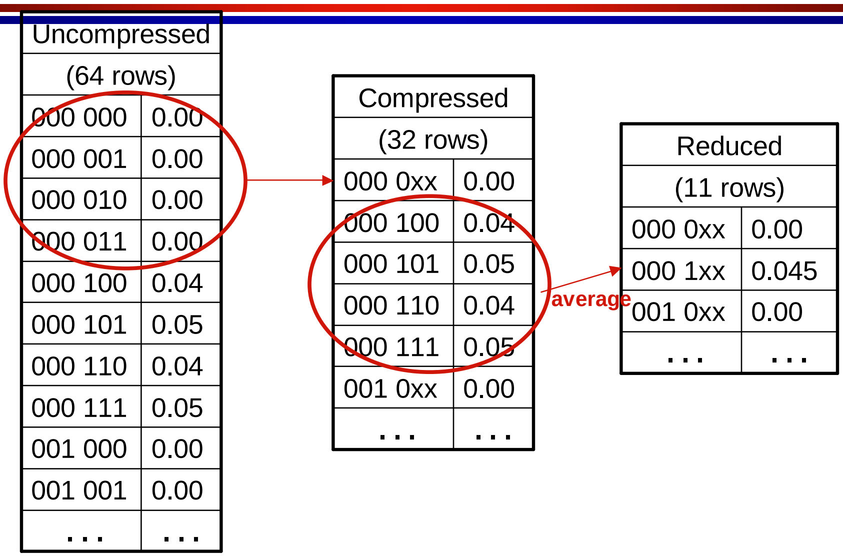
Table Compression

- Problem
  - » Results in large uncompressed table (e.g., 16-bit adder → 2^{32} rows)
  - » Excessive simulation (e.g., 2^{32} !)
- Solution
  - » Clustering Algorithm

    Reference: Huzefa Mehta, etal *“Module Energy Characterization using Clustering”,* DAC’96
  - » For 16-bit adder, to keep 12% average error → 1000 simulation points, 97 rows

---

2:1 Multiplexer Table

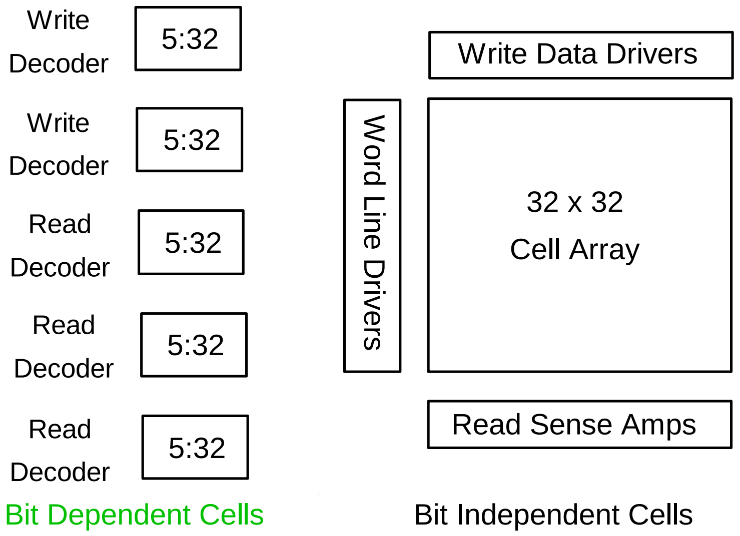


Uncompressed (64 rows)	
000 000	0.00
000 001	0.00
000 010	0.00
000 011	0.00
000 100	0.04
000 101	0.05
000 110	0.04
000 111	0.05
001 000	0.00
001 001	0.00
. . .	. . .

Compressed (32 rows)	
000 0xx	0.00
000 100	0.04
000 101	0.05
000 110	0.04
000 111	0.05
001 0xx	0.00
. . .	. . .

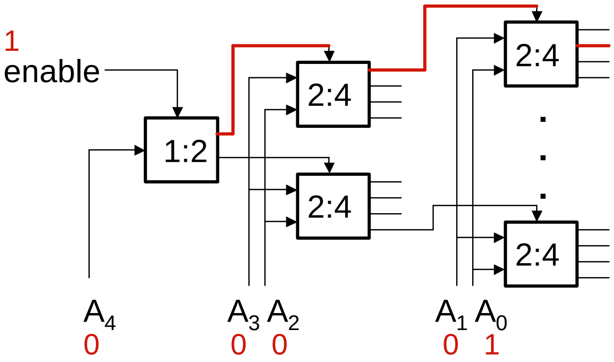
Reduced (11 rows)	
000 0xx	0.00
000 1xx	0.045
001 0xx	0.00
. . .	. . .

Register File Example



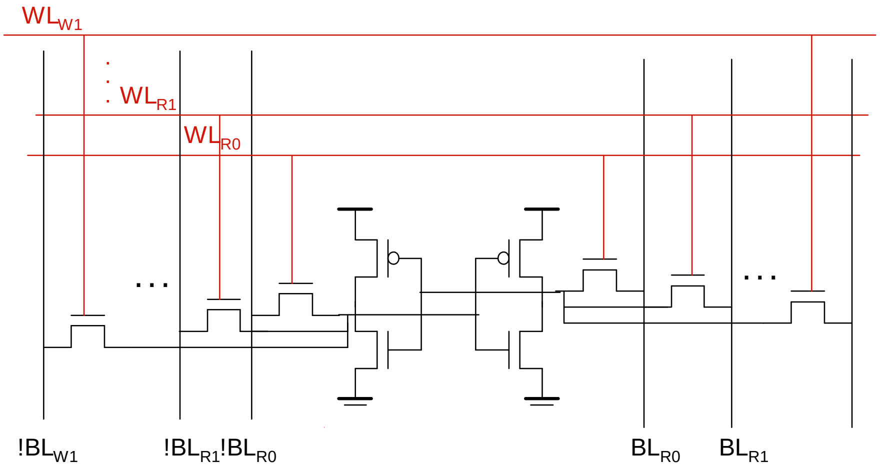
Decoder Characterization

- A 5:32 decoder → 2^{10} row table!
- Build 5:32 decoder out of smaller decoders

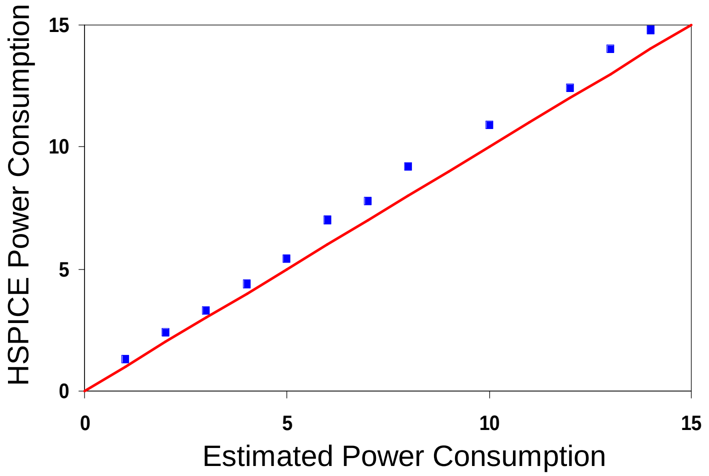


- A 2:4 decoder (with enable) → 2^6 row table

6-T SRAM Cell

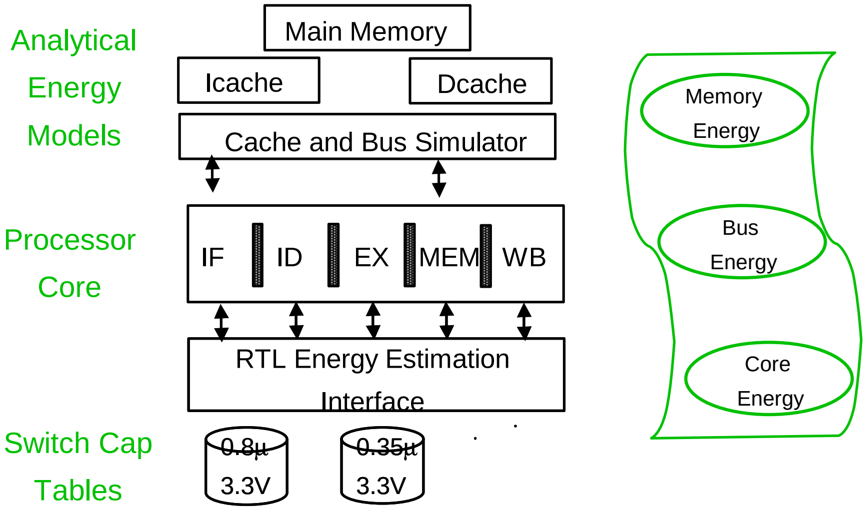


Validation of Register File Energy Model



From Ye,DAC,2000

SimplePower Framework



Analytical Energy Models

Main Memory

Icache

Dcache

Cache and Bus Simulator

Processor Core

IF ID EX MEM WB

RTL Energy Estimation Interface

Switch Cap Tables

0.8µ 3.3V

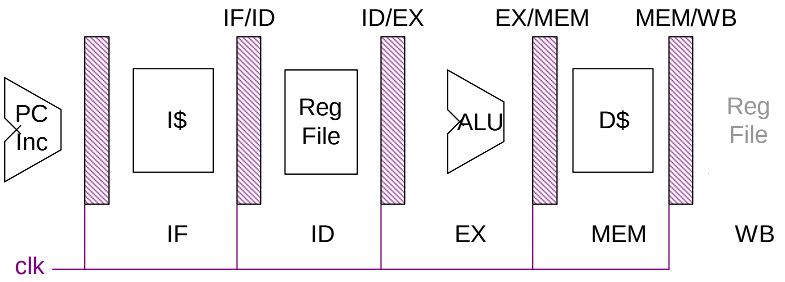
0.35µ 3.3V

Memory Energy

Bus Energy

Core Energy

Pipelined Processor Core



IF/ID ID/EX EX/MEM MEM/WB

PC Inc

I$

Reg File

ALU

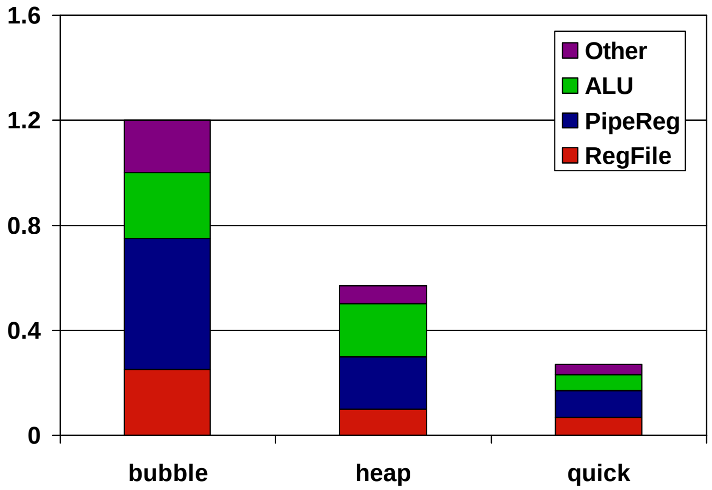
D$

Reg File

IF ID EX MEM WB

clk

Datapath Energy Consumption



From Ye,DAC,2000

Architectural Level Analysis Considerations

- Very computationally efficient
  - » requires predefined analytical and transition-sensitive energy characterization models
  - » requires design only to RTL (with some idea as to the kind of functional units planned)
  - » coarse grain - use of gated clocks implicit
- Simulation based so can be used to support architectural, compiler, operating system, and application level experimentation
- WattWatcher (Sente), DesignPower and PowerCompiler (Synopsys), prototype academic tools (Wattch - Princeton, SimplePower - PSU)

Key References

Avant!, www.avanticorp.com

Brooks, etal., Wattch: A framework for architectural-level power analysis and optimization, *ISCA*, 2000.

Cadence, www.cadence.com

Chen, etal., Validation of an architectural level power analysis technique, *DAC*, 1998.

Kamble, Ghose, Analytical energy dissipation models for low power caches, *ISLPED*, 1997.

Li, Henkel, A framework for estimating and minimizing energy dissipation of embedded hw/sw systems, *DAC*, 1998.

Mehta, etal., Energy characterization based on clustering, *DAC*, 1996.

Najm, A survey of power estimation techniques in VLSI circuits, *IEEE Trans. on VLSI Systems*, 2(4):446-455, 1994.

Sente, www.senteinc.com

Synopsys, www.synopsys.com/products/power/powerfamily_ds.html

Vijaykrishnan, etal., Energy-driven integrated hardware-software optimizations using SimplePower, *ISCA*, 2000.

Yeap, *Practical Low Power Digital VLSI Design*, KAP, 1998.