University of Toronto

Power Management for VLSI Circuits

Farid N. Najm

University of Toronto

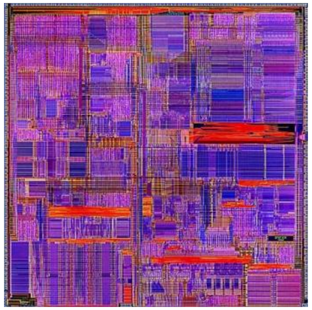
f.najm@utoronto.ca

Outline

- The Power Problem
  - Impact, issues, objectives, trends
- Technology Trends and Projections
  - Historical trends (since 1960)
  - Future projections (up to 2020)
- EDA for Power Management
  - Low-level power models
  - Power estimation and optimization
  - High-level power models
    - Bottom-up
    - Top-down
- Conclusion

The Power Problem

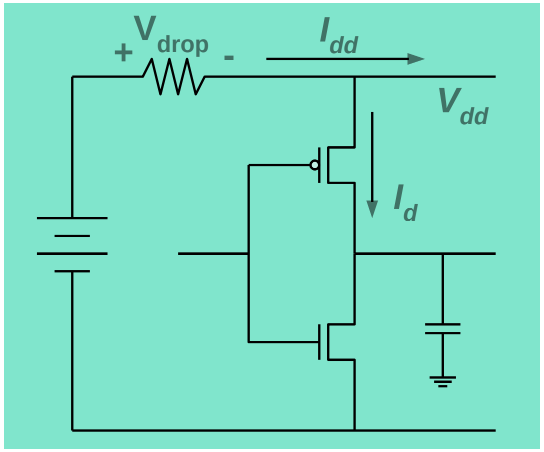
- High frequency and chip density lead to high power
  - Today's microprocessors consume 100-150 W
  - Future microprocessors may consume over 200 W
- Power has an impact on:
  - System performance (battery life)
  - Chip performance (circuit speed)
  - Packaging and cooling (cost)
  - Signal integrity: Inductive kick (Ldi/dt), IR drop, noise, etc.
  - Physical reliability: Electromigration, hot-carriers, etc.



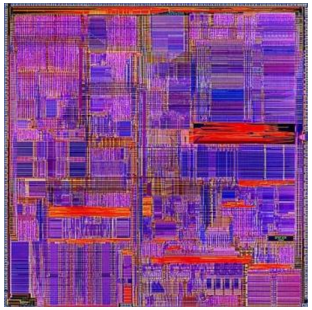
- Power is a problem in both portable & fixed equipment

Impact on Performance

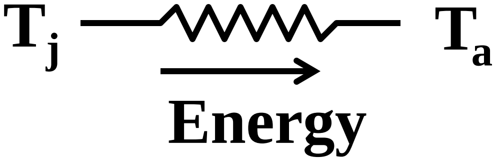
- Power dissipation affects chip speed in two ways:
  - Power supply (voltage) variations (IR drop, Ldi/dt drop)
  - Temperature variations
- Power supply reduction causes a circuit to slow down
  - A 5% reduction in V_{dd} may cause a 15% increase in gate delay
- Increased temperature has a complex effect on speed
  - Traditionally: slow down
  - Today: gates speed-up, wires slow down
    - Overall result depends on the "mix"
    - Speed-up is not necessarily a good thing, due to thermal runaway



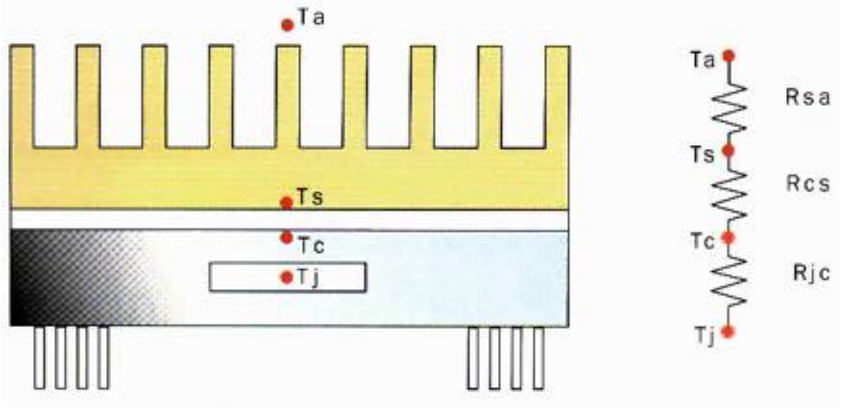
Cooling/Packaging Cost

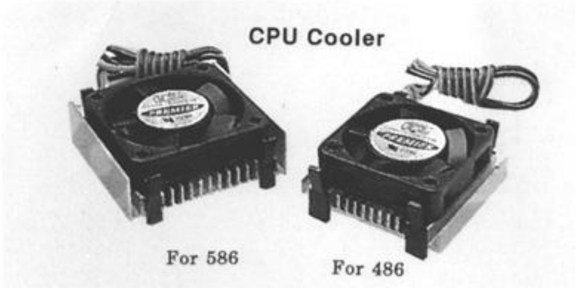


$$T_j - T_a = R_{th} \times P_{avg}$$



$$$

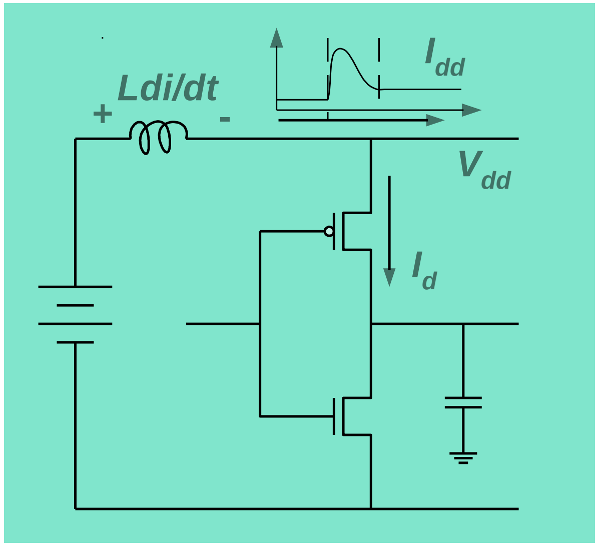




$$$

Impact on Signal Integrity

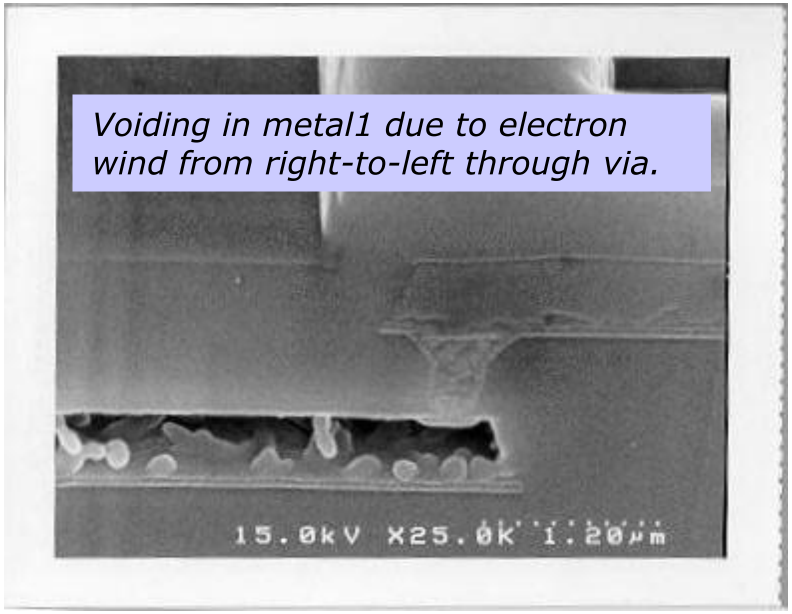
- Power supply current transients cause:
  - Inductive kick due to Ldi/dt
  - IR drop due to line resistance
- Leads to supply voltage glitches and signal integrity problems:
  - Glitches get coupled to sensitive analog or mixed signal nodes
  - Can cause dynamic circuits to loose charge
  - Can cause latches to change state



Impact on Chip Reliability

- ICs are subject to a variety of physical failure mechanisms:
  - Electromigration (EM)
  - Hot-carrier degradation (HC)
  - Other
- Reliability is worse under:
  - High switching activity
  - High temperature
- Result: chip MTF is reduced under high power conditions

Electromigration Damage



Voiding in metal1 due to electron wind from right-to-left through via.

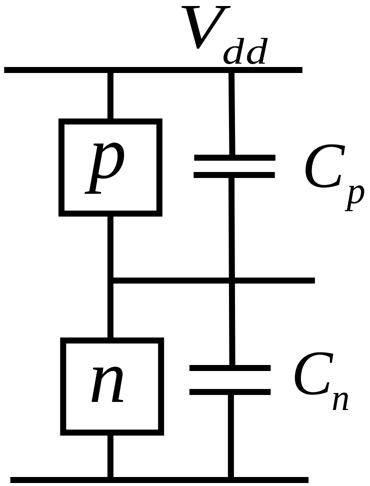
Power 101

$$C_{tot} = C_n + C_p$$

$$Q_{lh} = C_n V_{dd} \qquad Q_{hl} = C_p V_{dd}$$

$$Q_{lh+hl} = Q_{lh} + Q_{hl} = C_{tot} V_{dd}$$

$$E_{01} + E_{10} = Q_{lh} V_{dd} + Q_{hl} V_{dd} = C_{tot} V_{dd}^2$$



⇒ $$E_{avg} = \tfrac{1}{2} C_{tot} V_{dd}^2$$ **(average energy per logic transition)**

⇒ $$\boxed{P_{avg} = C_{tot} V_{dd}^2 f}$$ **(average power at frequency: *f*)**

- Average power depends on:
  - ***switching frequency***
  - ***supply voltage*** **(squared)**
  - ***transistor or gate count (C)***

Bottom Line

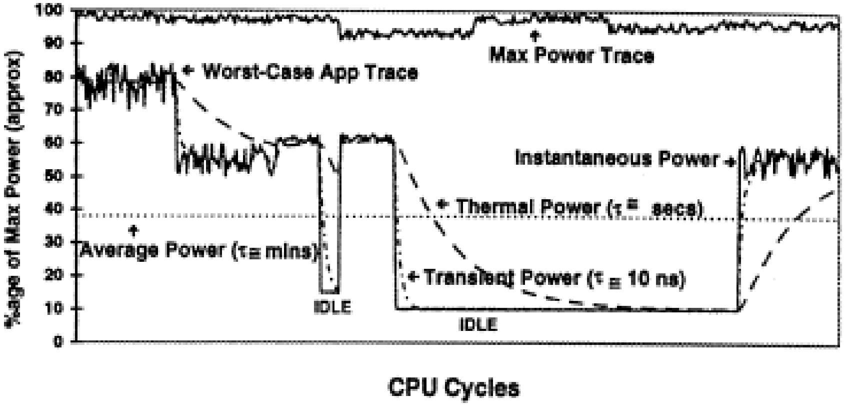
- Power has become a primary design concern

- As part of a *low-power design methodology*, tools are needed to accomplish several tasks:
  - Power Modeling and Characterization
    - Levels: MOSFET, gate, cell, macro, core, memory, IO
    - Objectives: static, average, RMS, peak power
  - Power Estimation (Analysis)
    - At all levels and for all objectives
  - Power Optimization (Synthesis)
    - At all levels and for all objectives

- High-level tools are highly desirable

Power Depends on Workload

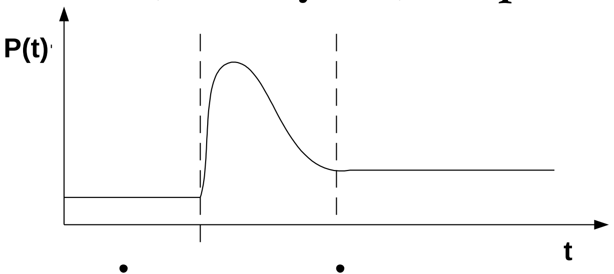
- Power, much more so than delay, depends on what stimulus is applied to the circuit, i.e., power is:
  - Pattern-dependent
  - Mode-dependent
  - Workload-dependent



- Example: Running MS-Word on a laptop requires much less power than running a 3D video game
- For power estimation, reasonable accuracy is possible only when the mode of operation is understood

Power Analysis Objectives

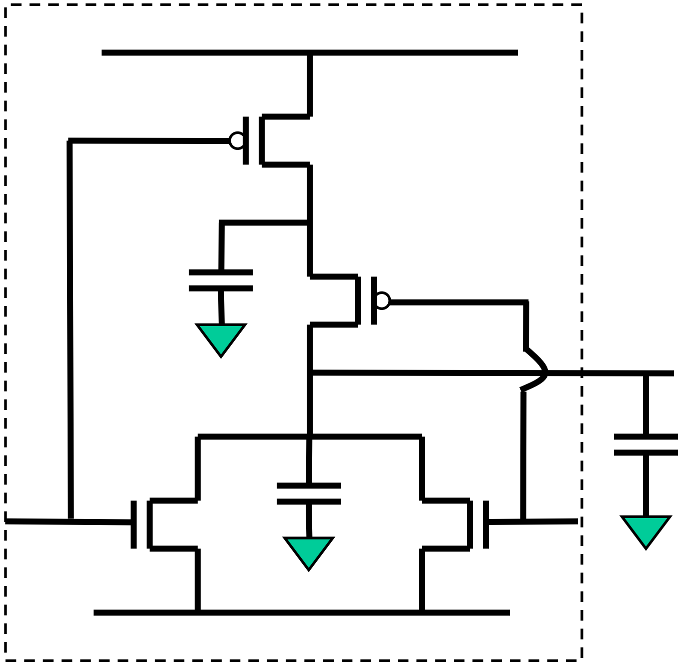
- Different power objectives:
  - Transient power (waveform over time) v.s. average power
    - Use transient power for IR drop, power bus sizing, signal integrity, physical reliability, etc.
      - ➔ Sometimes, only interested in *peak* power
      - ➔ May be interested only in highest current *slope*
    - Use average power for package selection, battery life, temperature analysis, etc.
  - Static power v.s. dynamic power



- It is possible to talk about the transient static power, and the average static power
  - Static power is pattern-dependent (it's not exactly DC)

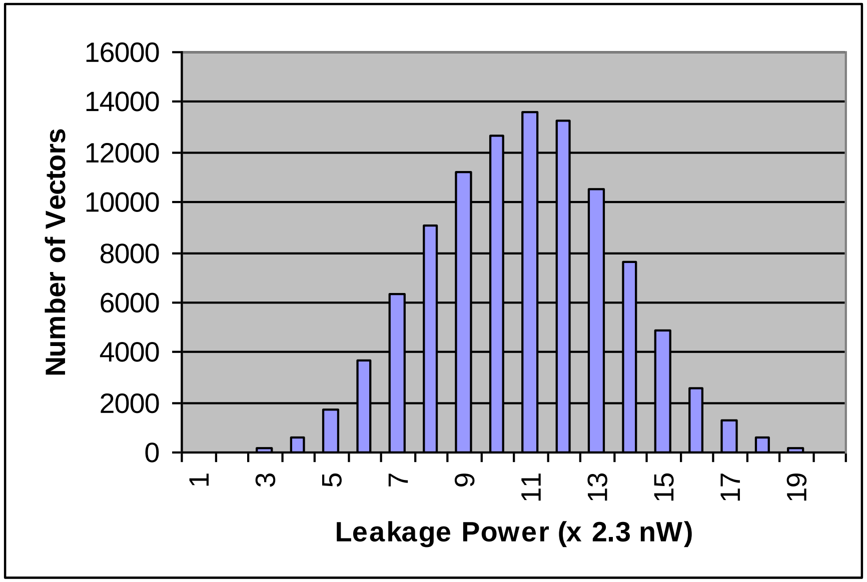
Dynamic Power

- Dynamic power is consumed when signals are switching, and is due to:
  - Short-circuit current, also called crowbar, or rush-through current
  - Current to charge internal nodes
  - Current required to charge output and loading capacitance
- Best understood as *dynamic energy per transition*



Static Power

- Static power consists of:
  - Off-current (leakage)
    - Sub-threshold current
    - Gate oxide leakage
    - pn-junction leakage
  - Standby current (sneak paths, pull-ups, trickle devices, sense-amplifiers)
- In CMOS, leakage can be large:
  - 1996: 100 MHz DSP core: 10%
  - 1998: 600 MHz Alpha uP: 2W/72W
  - 2002: 90nm FPGA: 10W
  - 2006: 65nm 2-core Itanium: 40%
  - Future: 50% ?



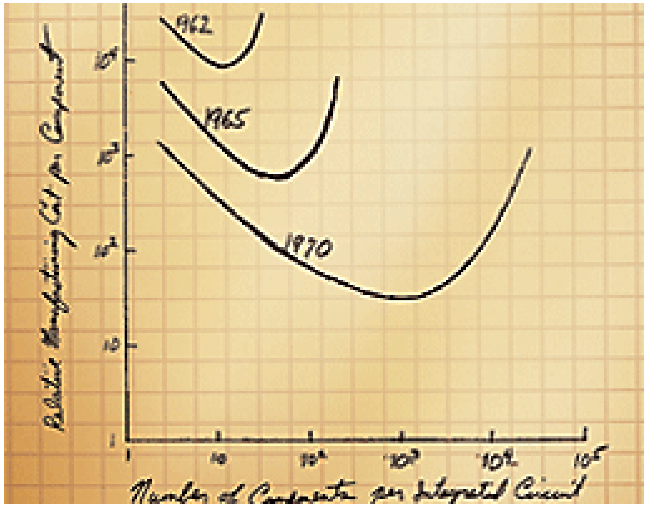
Leakage, being exponential in Vth, is the key reason why Vdd will <u>not</u> scale any more!

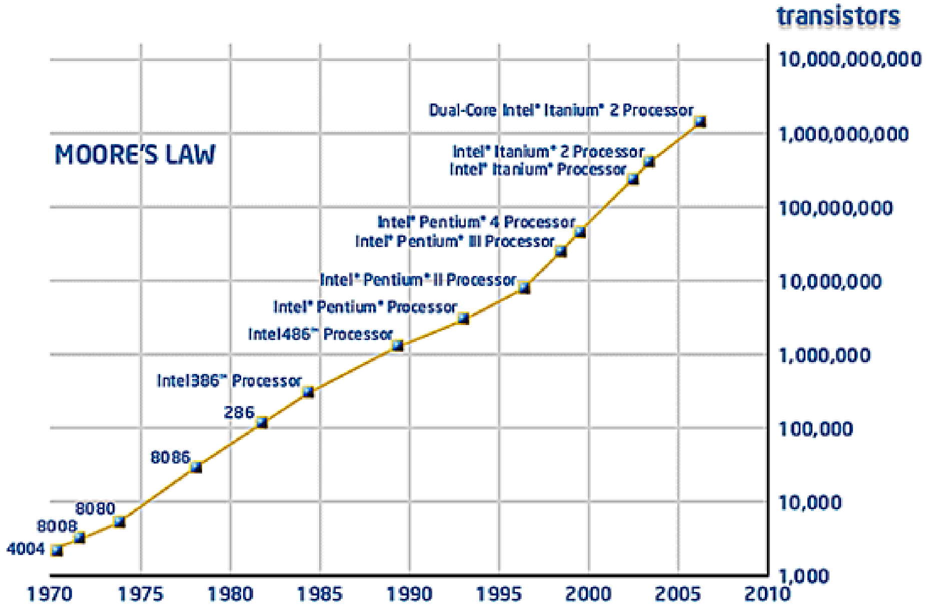
Outline

- The Power Problem
  - Impact, issues, objectives, trends
- Technology Trends and Projections
  - Historical trends (since 1960)
  - Future projections (up to 2020)
- EDA for Power Management
  - Low-level power models
  - Power estimation and optimization
  - High-level power models
    - Bottom-up
    - Top-down
- Conclusion

Moore’s Law

- Empirical observation by Gordon Moore, 1965: “*the number of transistors on a chip* [for minimum component cost] *doubles about every two years*”

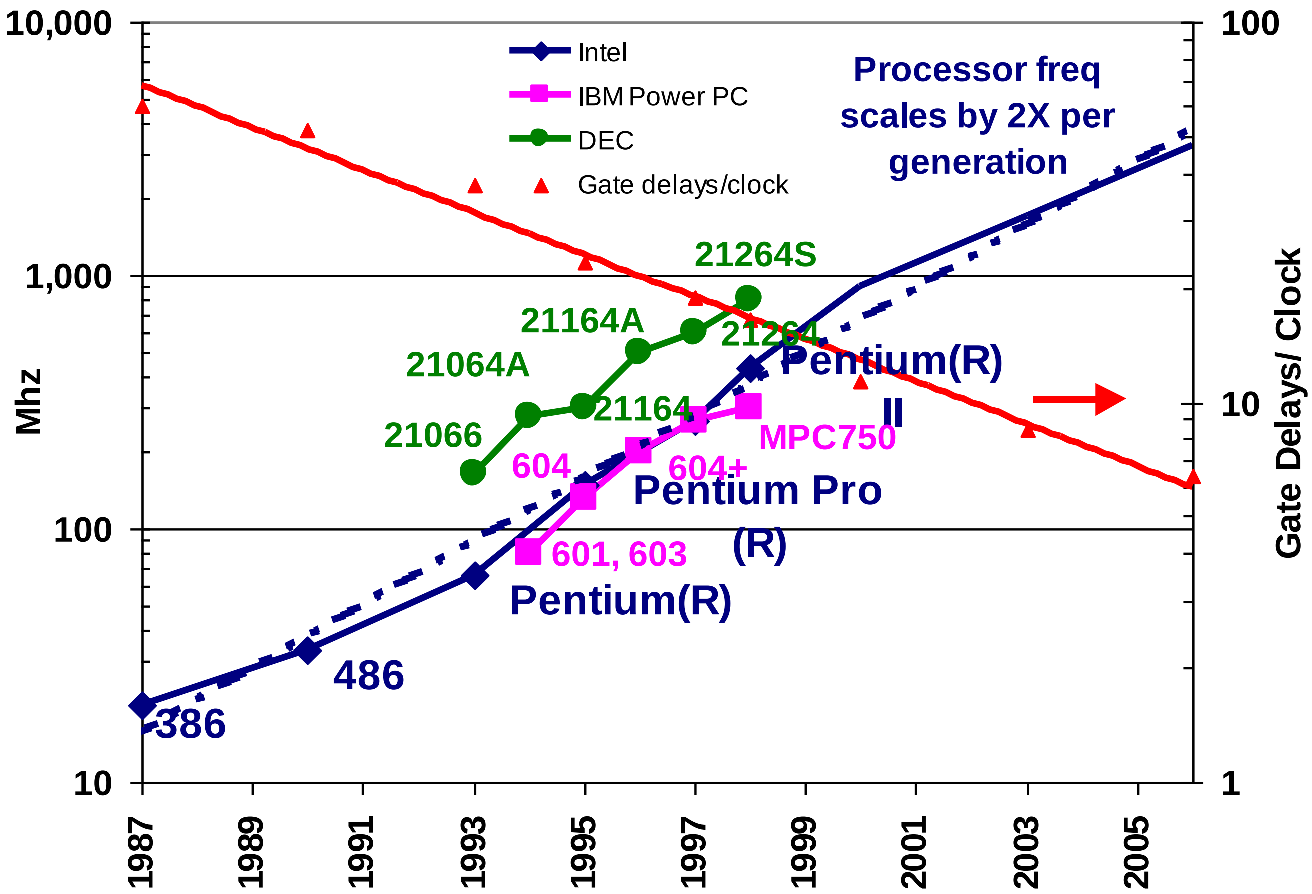




- This exponential trend has fueled economic expansion on a global scale
  - Very few things in nature are “exponential”
  - Things cannot grow exponentially forever!

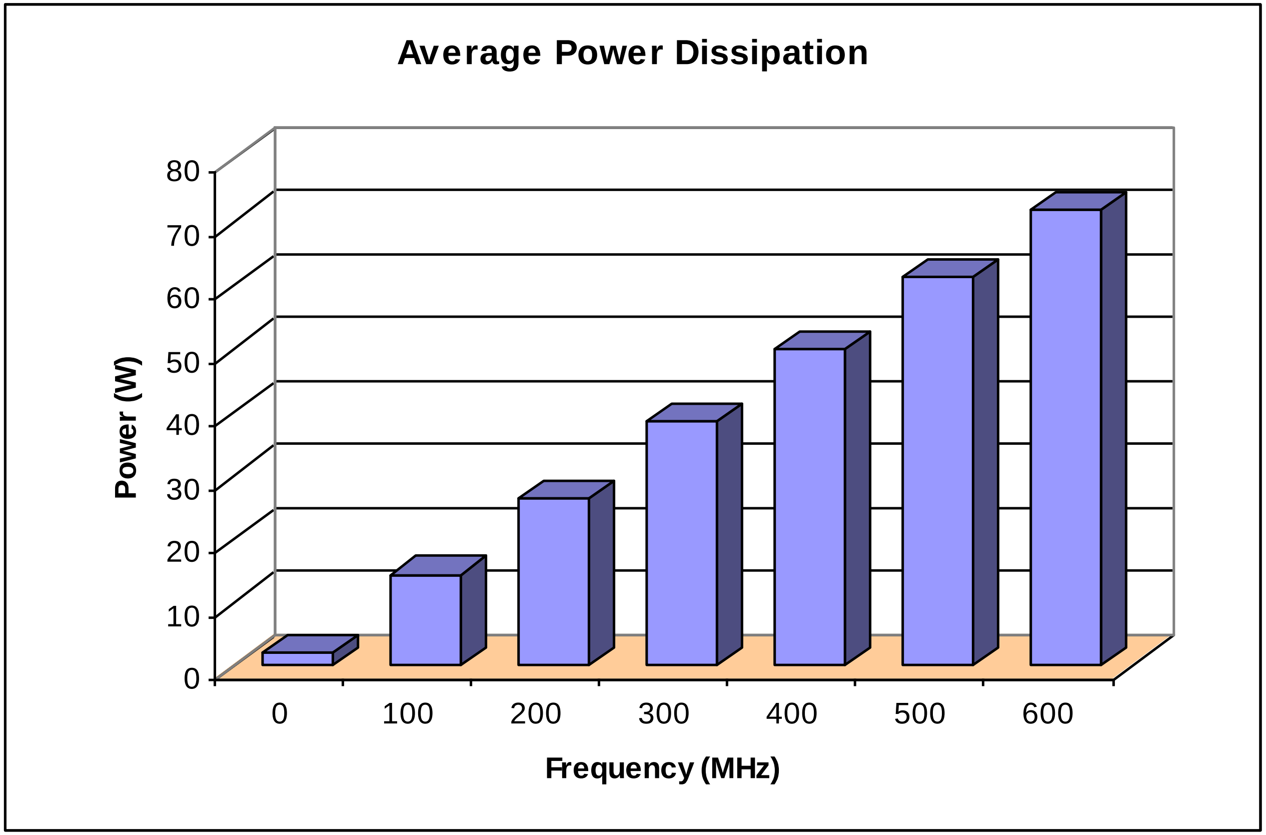
Source: http://www.intel.com/technology/mooreslaw/

Microprocessor Performance



Source: Intel

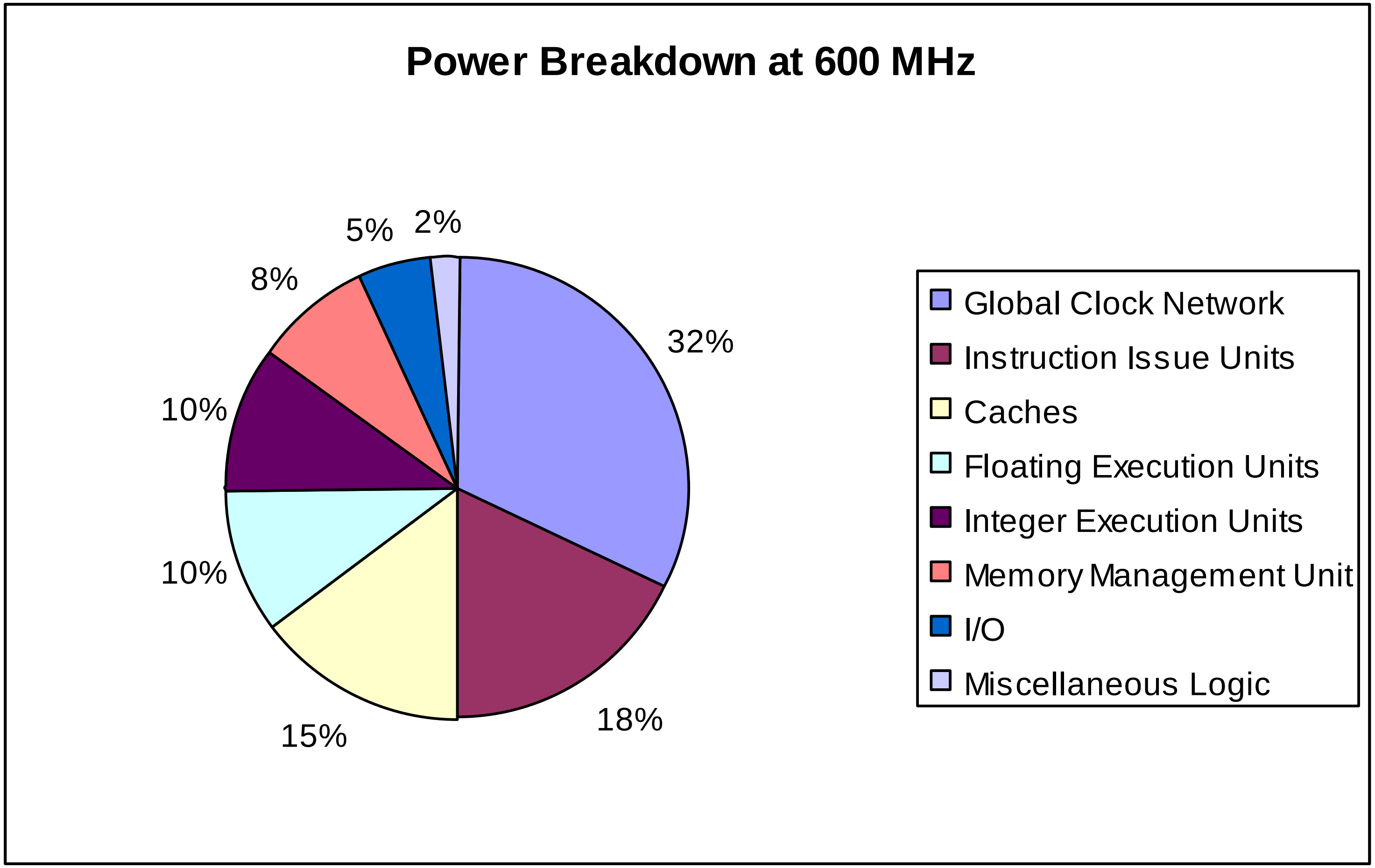
1998 DEC Alpha 21264



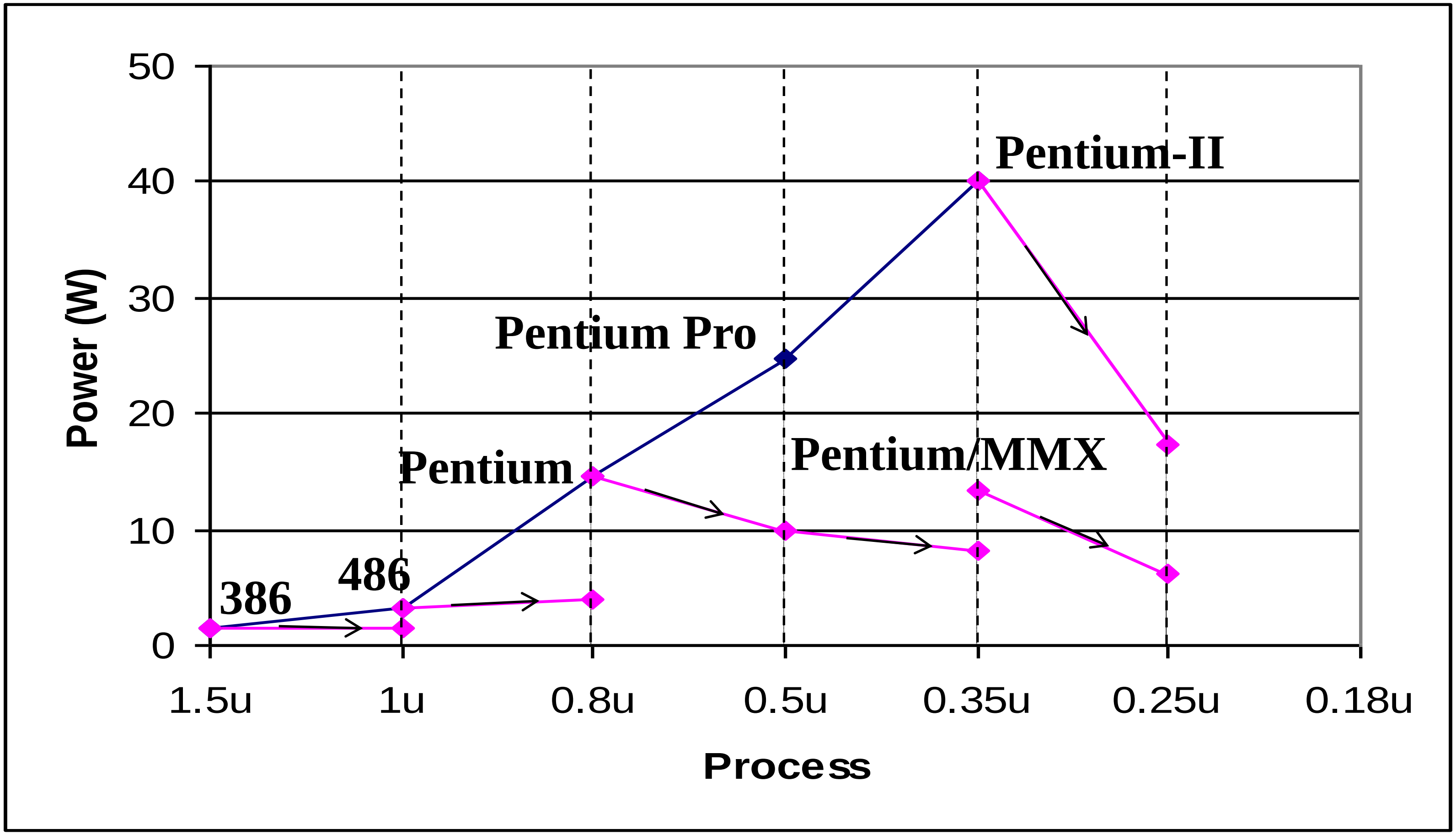
0.35μ, 2.2V
R_{th}=0.3 °C/W
2W at DC, mostly leakage
Max 95W at 600MHz

72 W, 33 A, at 600 MHz
15.2 Million Transistors, 3.14 cm^2

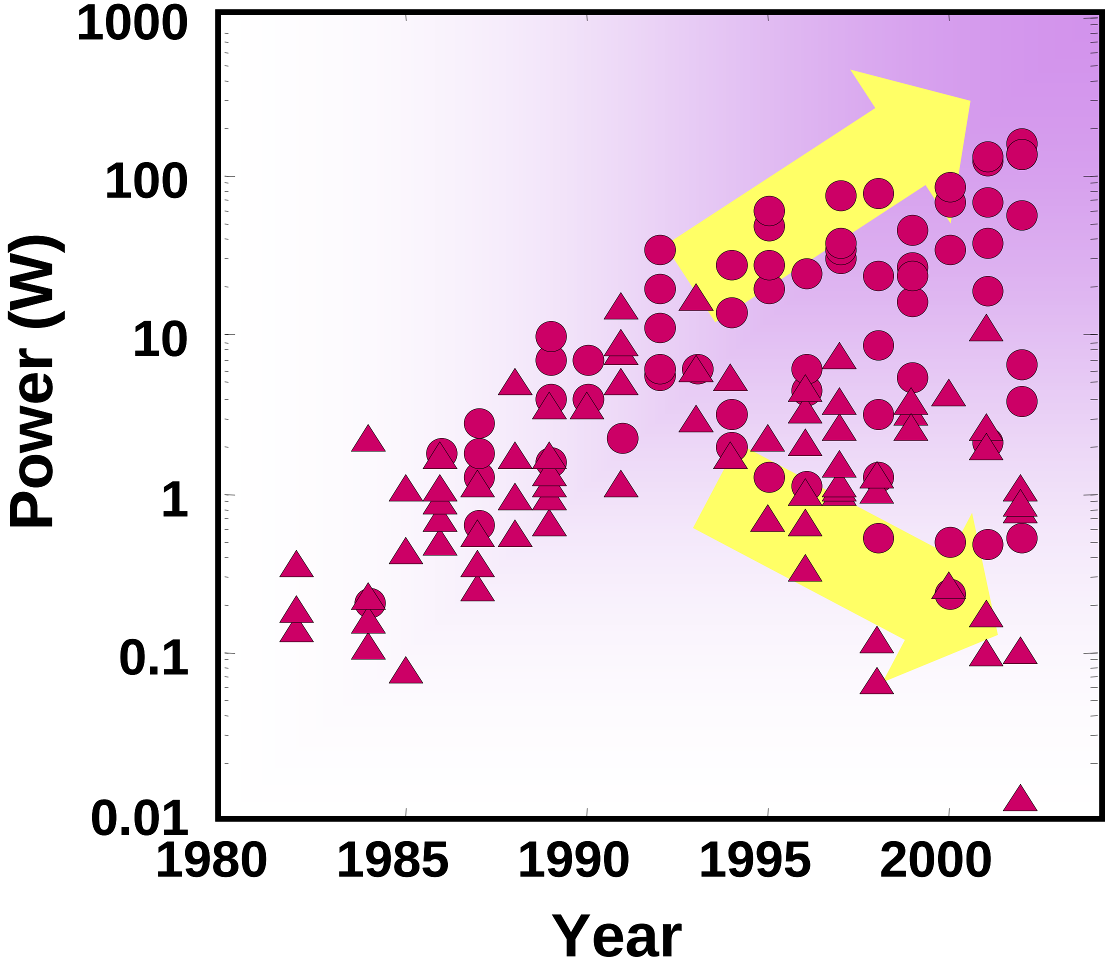
1998 DEC Alpha 21264



Intel uPs, Maximum Power

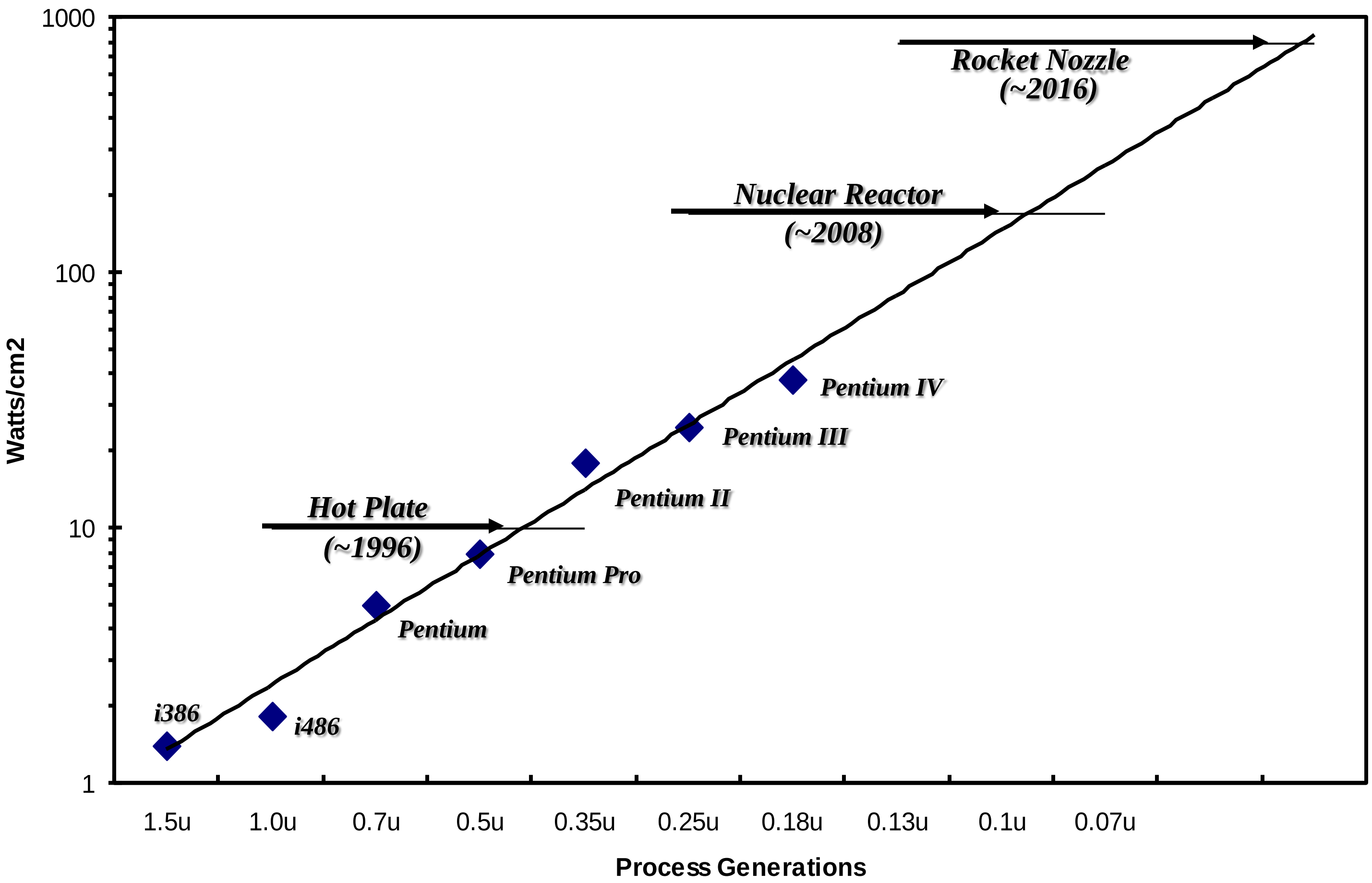


Processors Reported in ISSCC



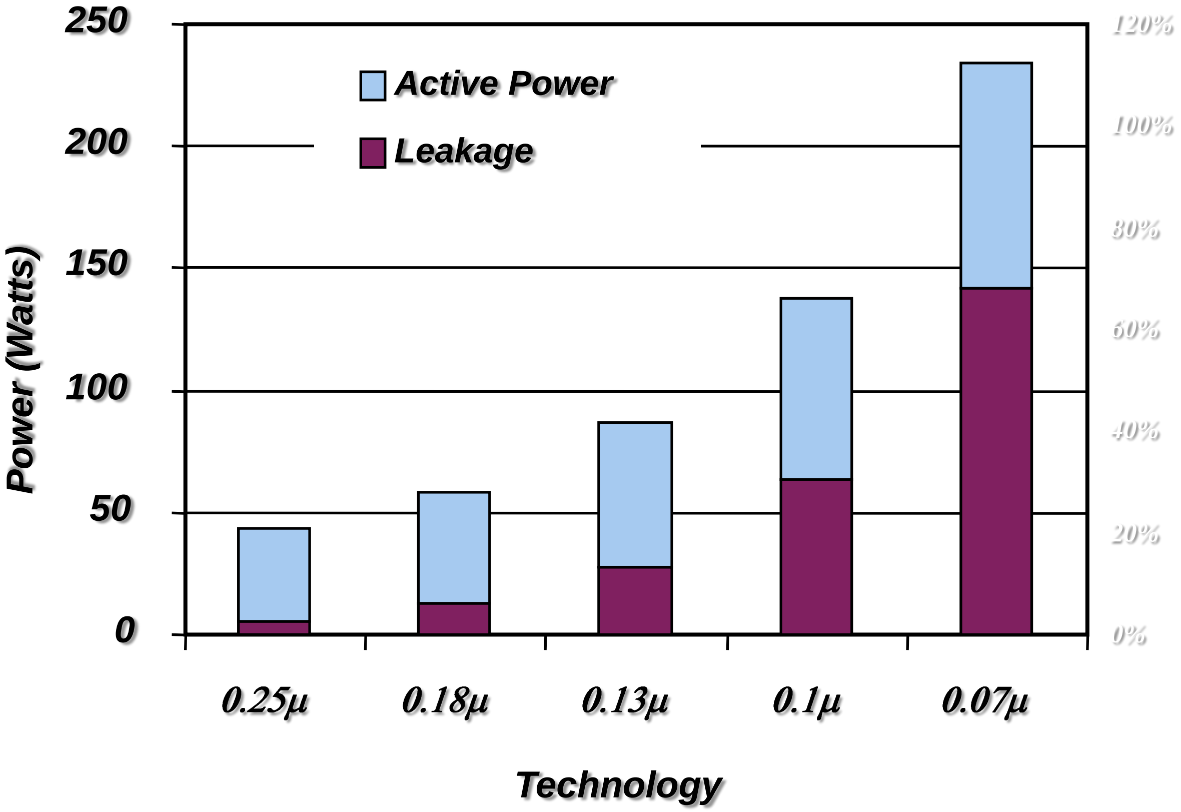
Source: T. Sakurai, ICCAD-02 Tutorial

Intel uPs, Power Density (1999)



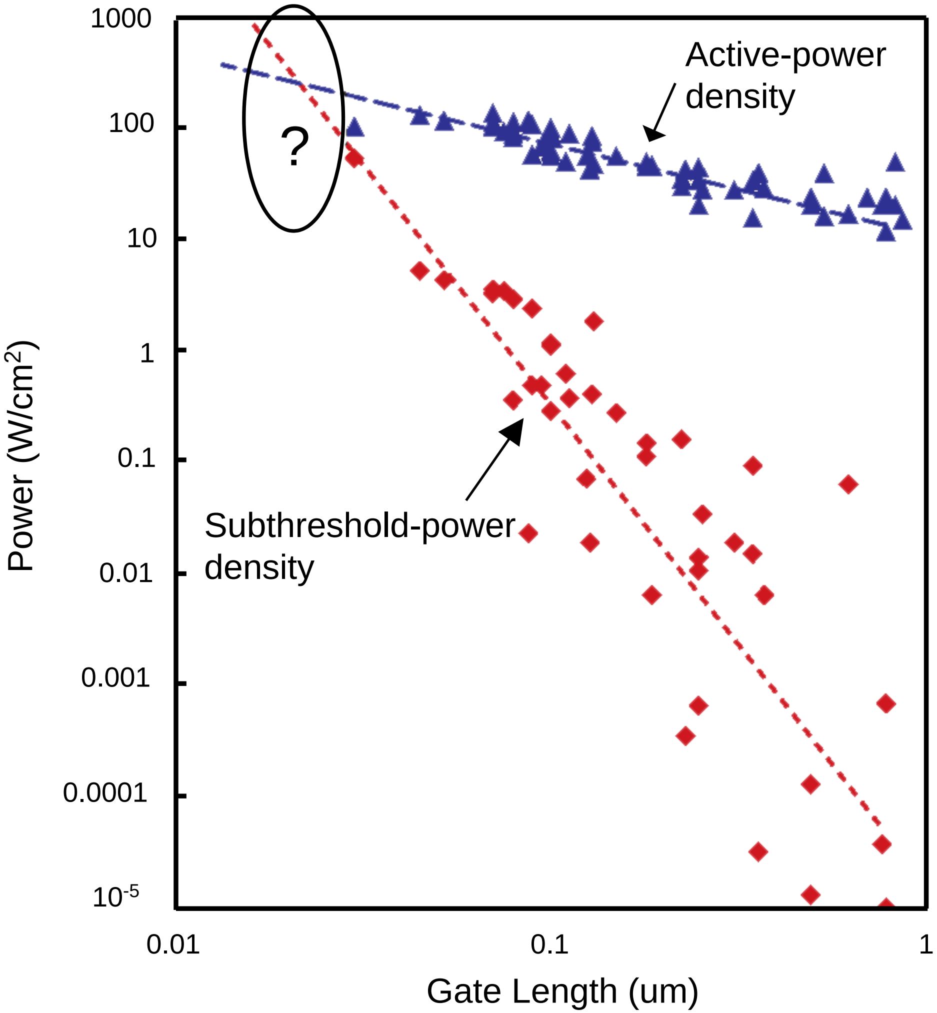
Source: F. J. Pollack, "New microarchitecture challenges in the coming generations of CMOS process technologies," 32nd Annual ACM/IEEE International Symposium on Microarchitecture (Micro32), Haifa, Israel, p. 2, 1999. (Plenary Presentation)

Intel Leakage Trend (2000)



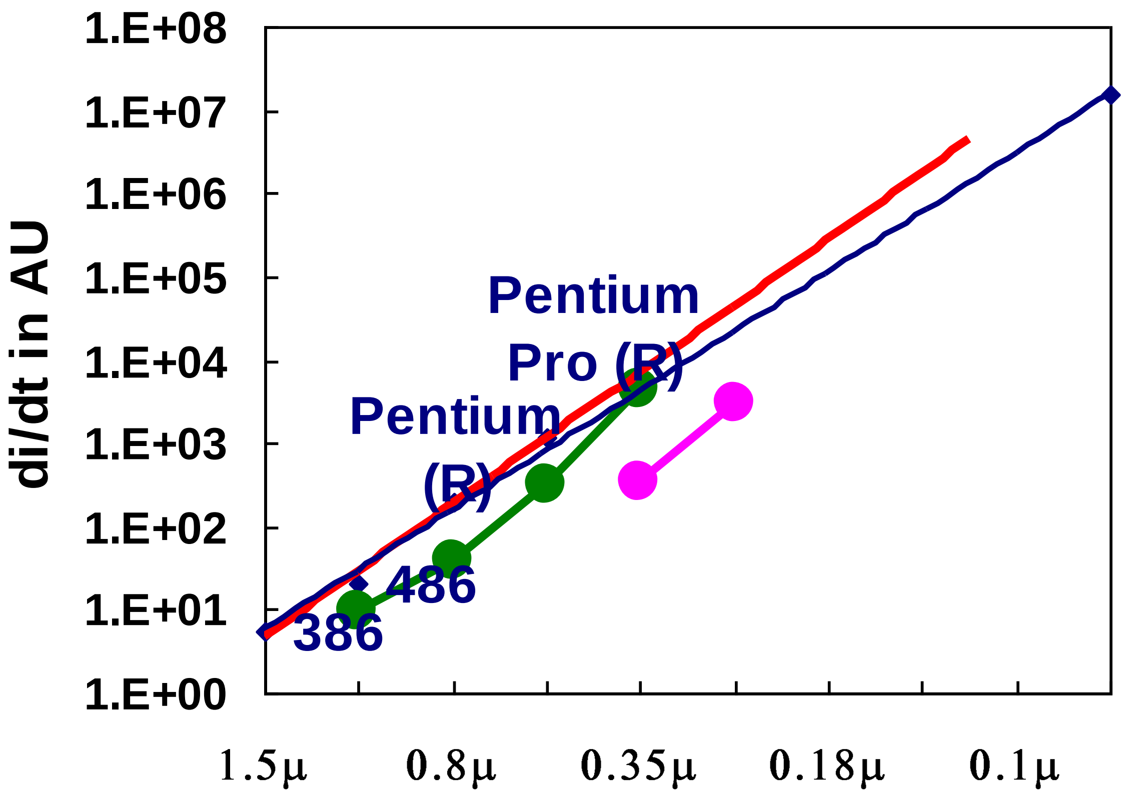
Source (Intel paper): Kam, Rawat, Kirkpatrick, Roy, Spirakis, Sherwani andPeterson, "EDA challenges facing future microprocessor design", IEEE Transactions on Computer-Aided Design, Vol. 19, No. 12, December 2000.

IBM Leakage Trend (2002)



E.J. Nowak. Maintaining the benefits of CMOS scaling when scaling bogs down. Volume 46, Numbers 2/3, Page 169 (2002)

Intel di/dt Trend (2000)

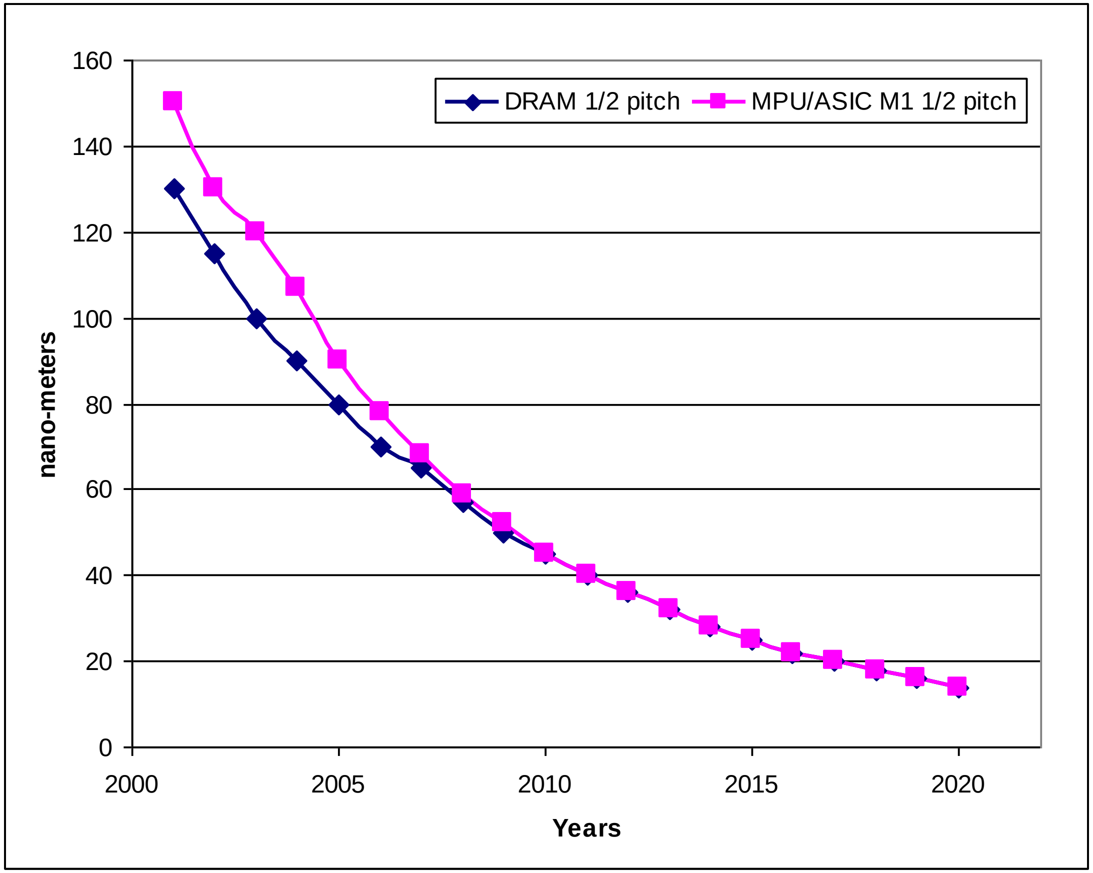


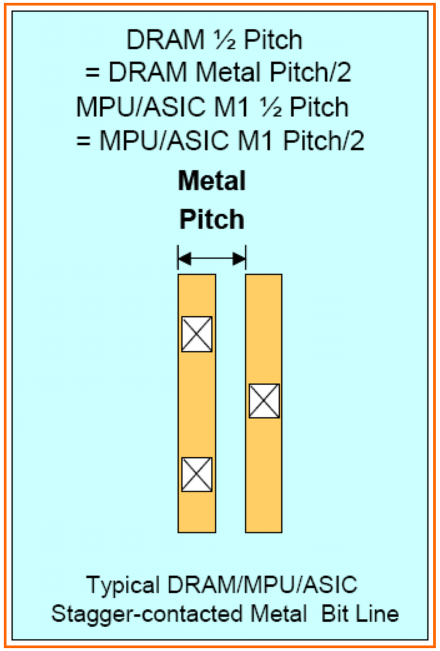
Source (Intel paper): Kam, Rawat, Kirkpatrick, Roy, Spirakis, Sherwani andPeterson, "EDA challenges facing future microprocessor design", IEEE Transactions on Computer-Aided Design, Vol. 19, No. 12, December 2000.

Outline

- The Power Problem
  - Impact, issues, objectives, trends
- Technology Trends and Projections
  - Historical trends (since 1960)
  - Future projections (up to 2020)
- EDA for Power Management
  - Low-level power models
  - Power estimation and optimization
  - High-level power models
    - Bottom-up
    - Top-down
- Conclusion

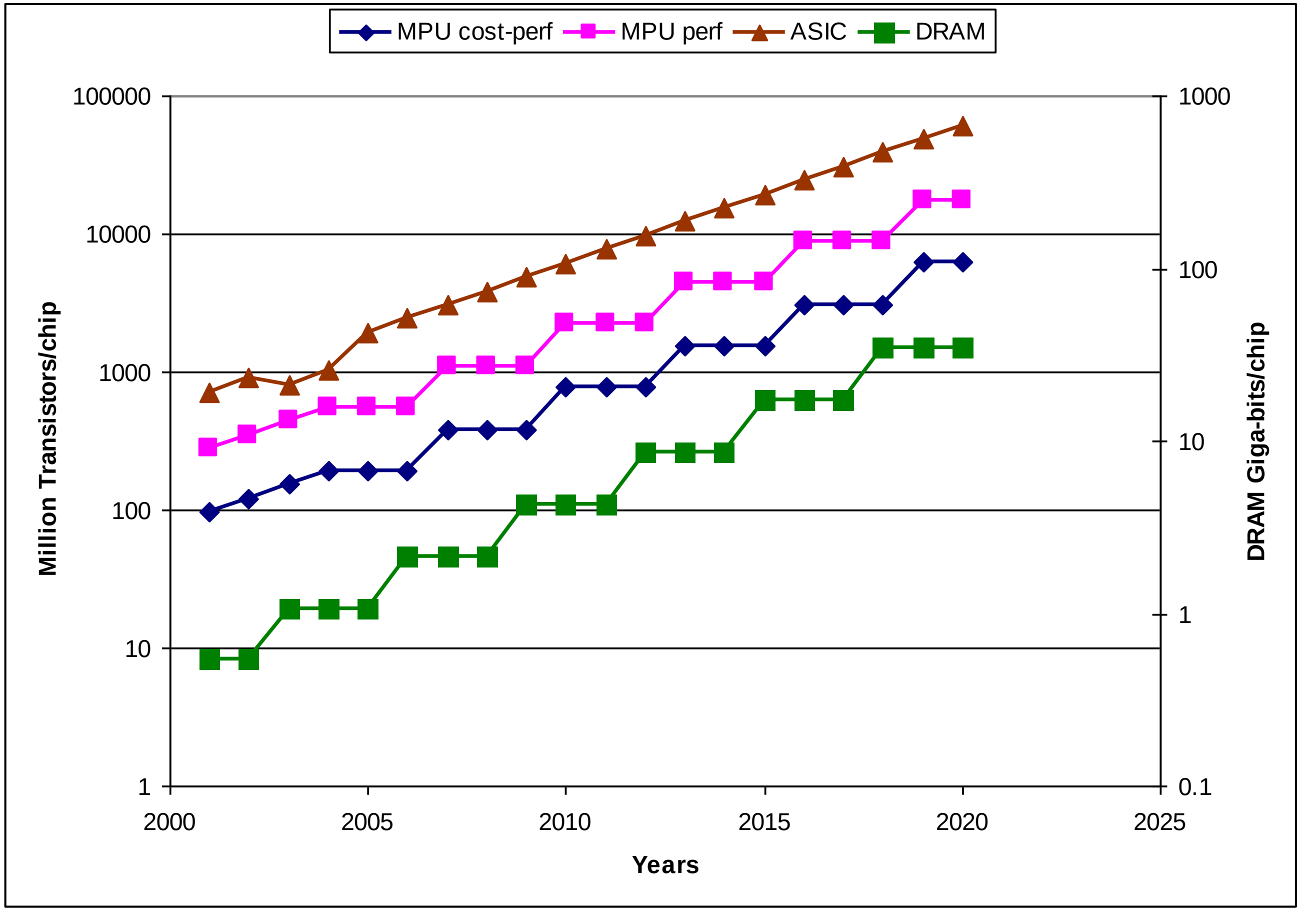
Feature Size





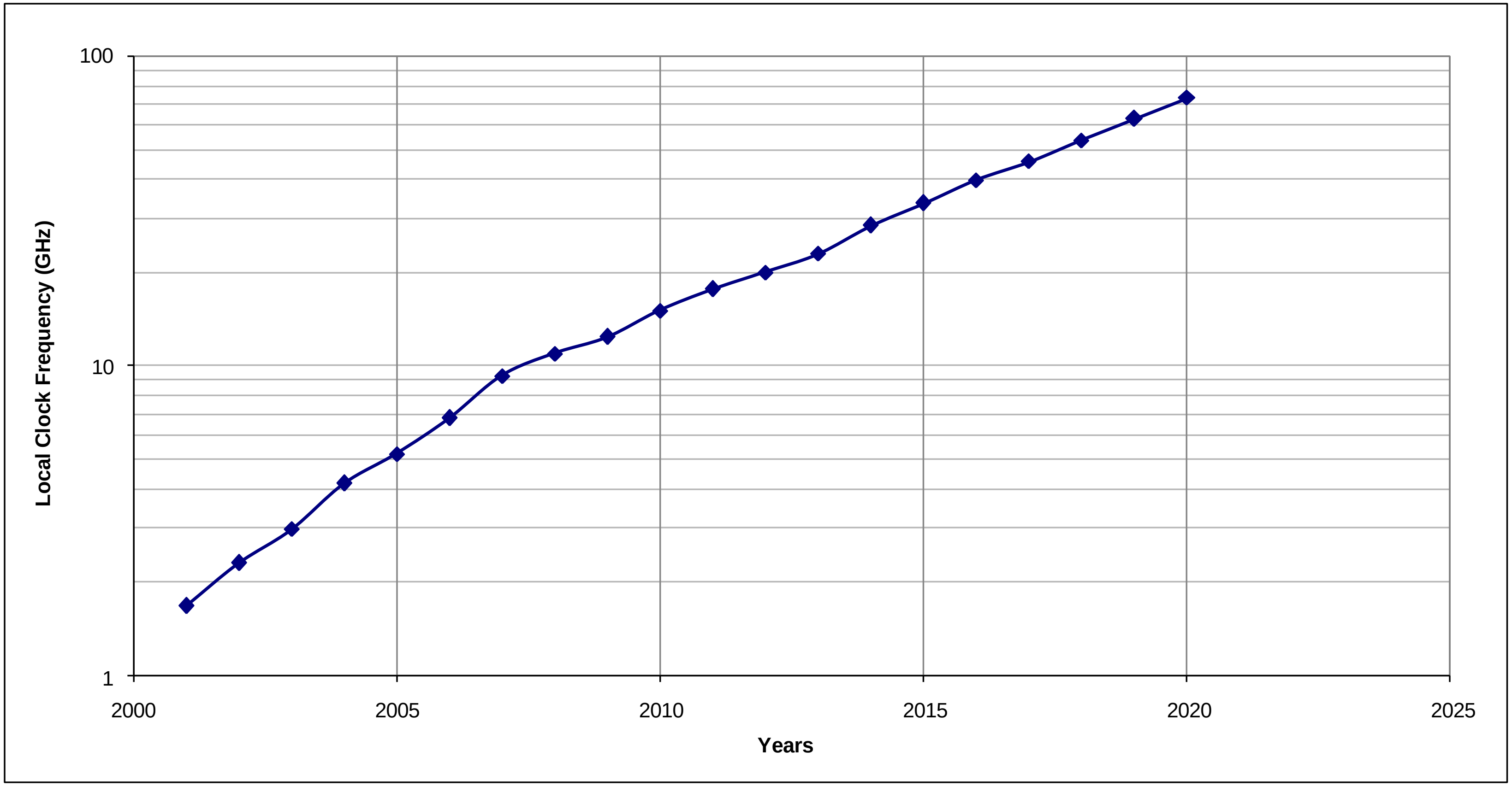
Source: ITRS-05 & 06, http://public.itrs.net.

Functionality



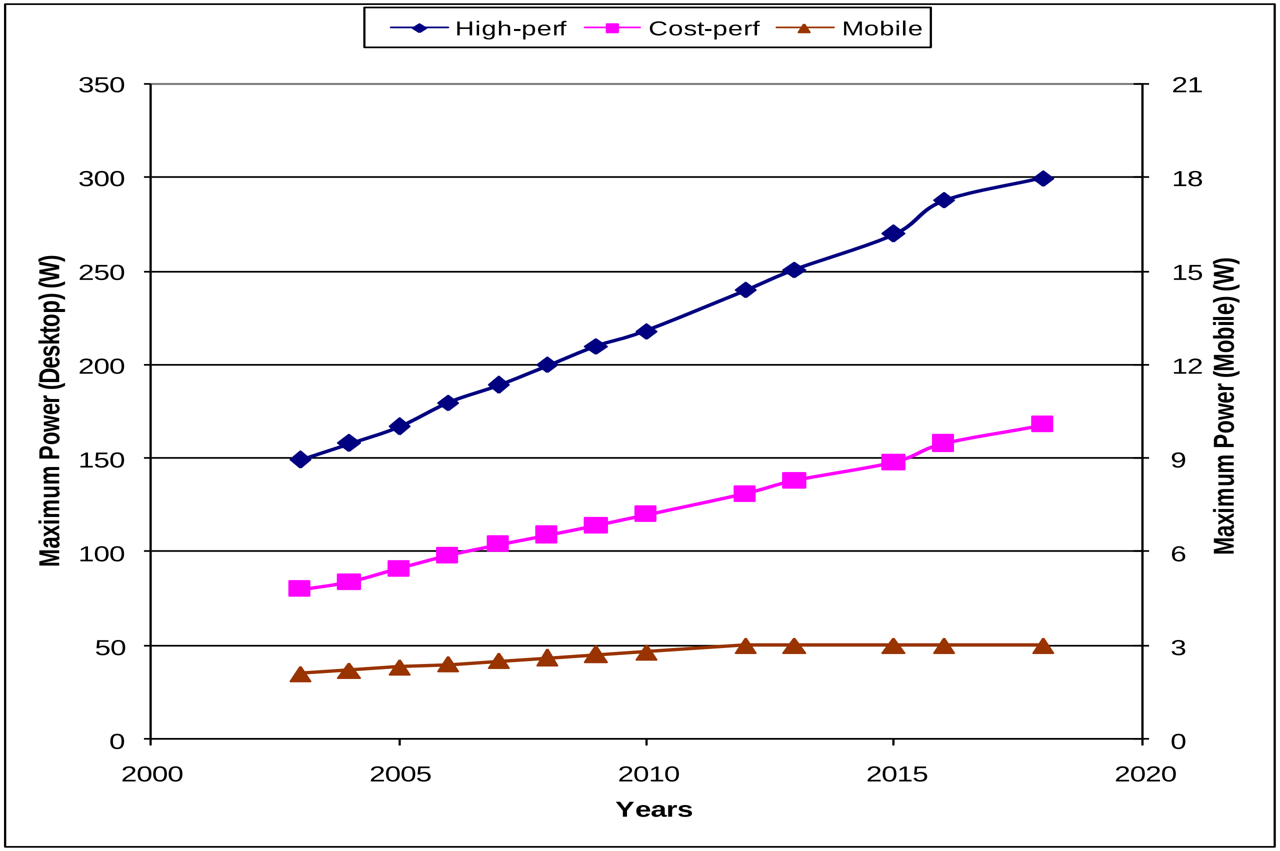
Source: ITRS-05 & 06, http://public.itrs.net.

Speed



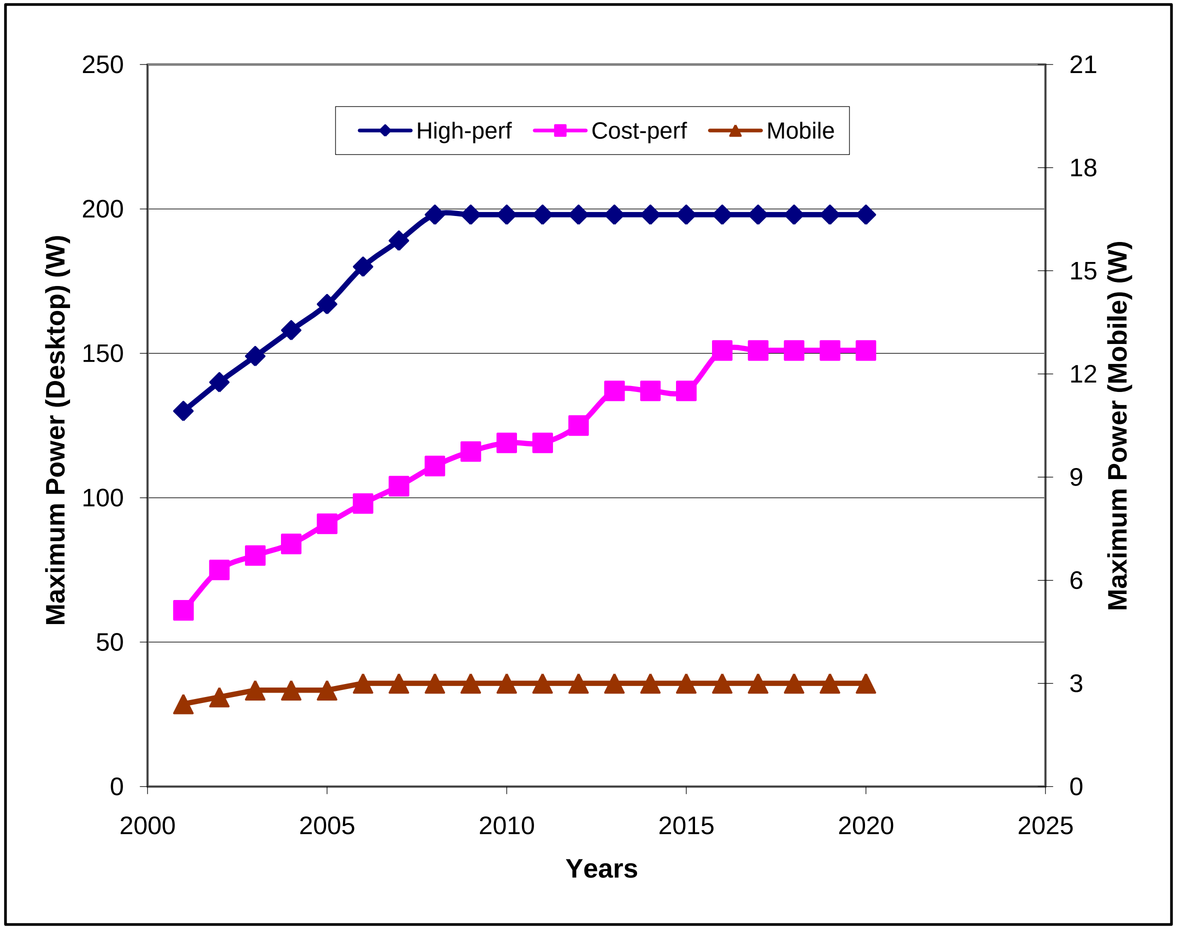
Source: ITRS-05 & 06, http://public.itrs.net.

Power Dissipation (ITRS-03)



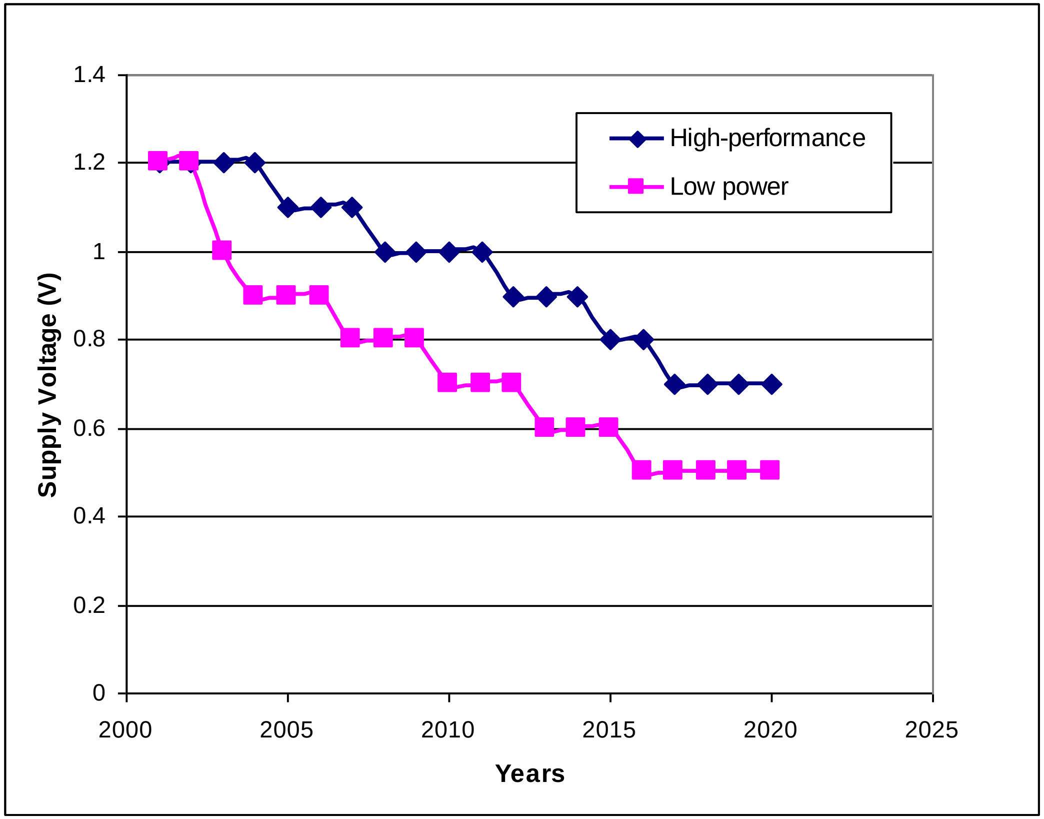
Source: ITRS-03, http://public.itrs.net.

Power Dissipation (ITRS-06)



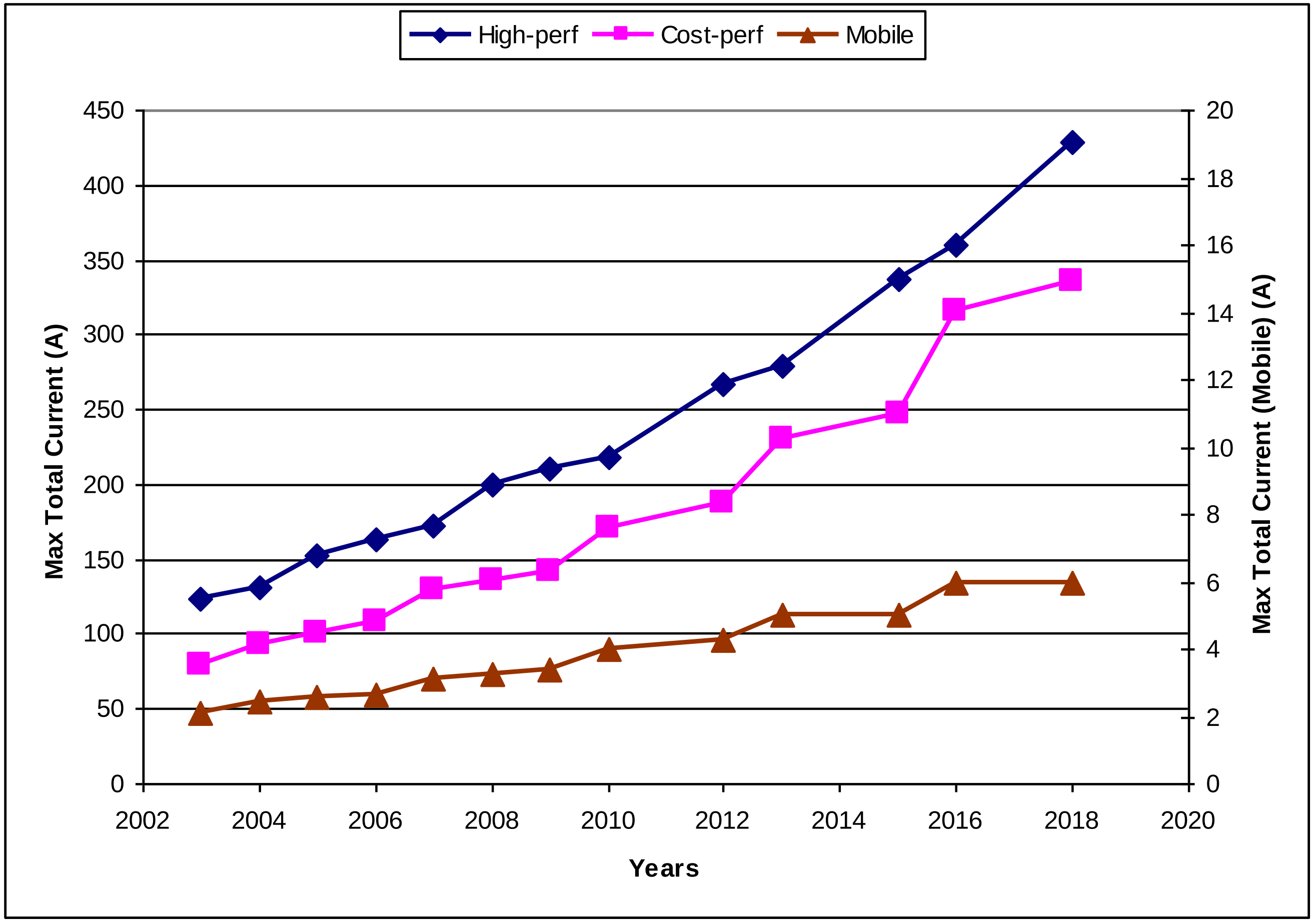
Source: ITRS-06, http://public.itrs.net.

Supply Voltage

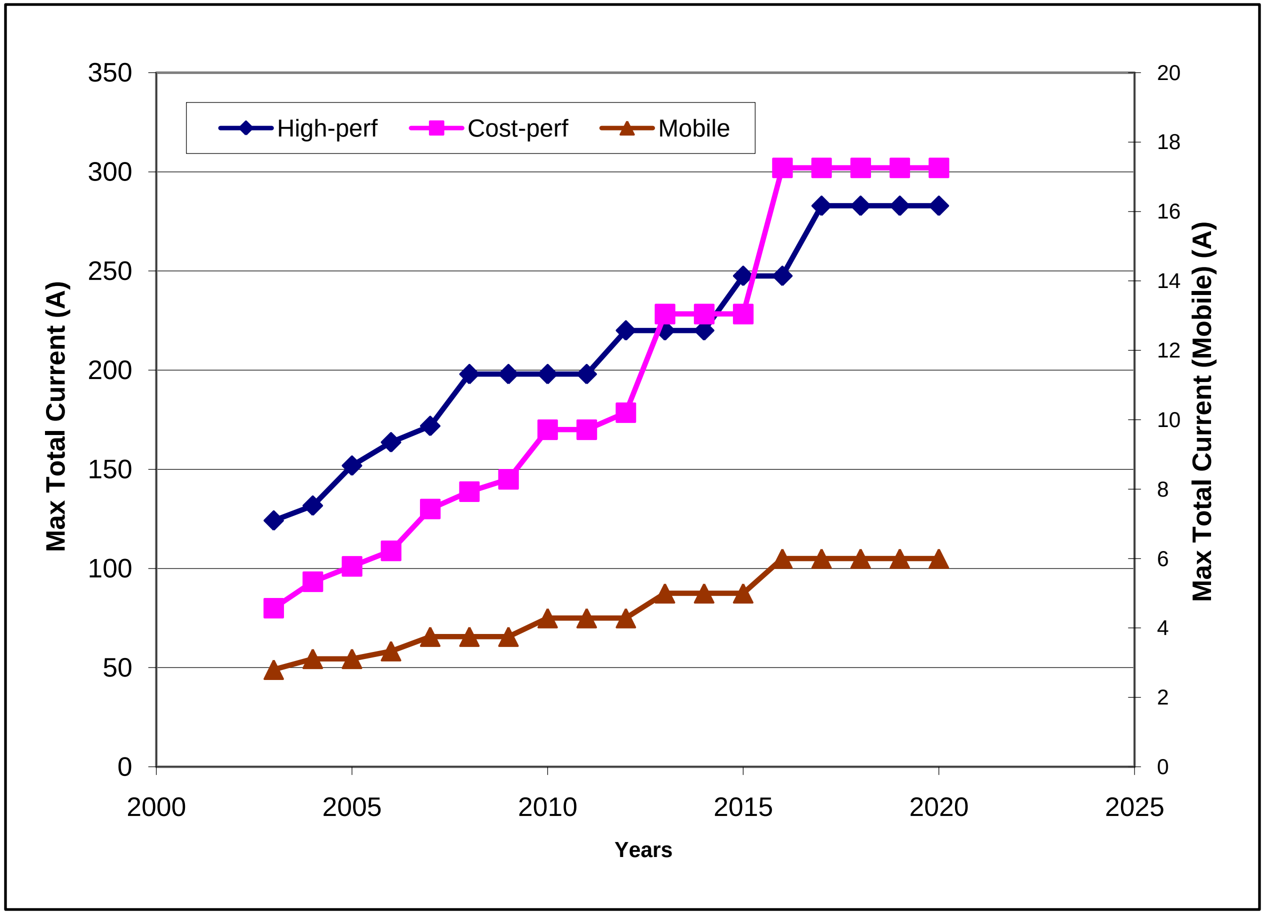


Source: ITRS-05 & 06, http://public.itrs.net.

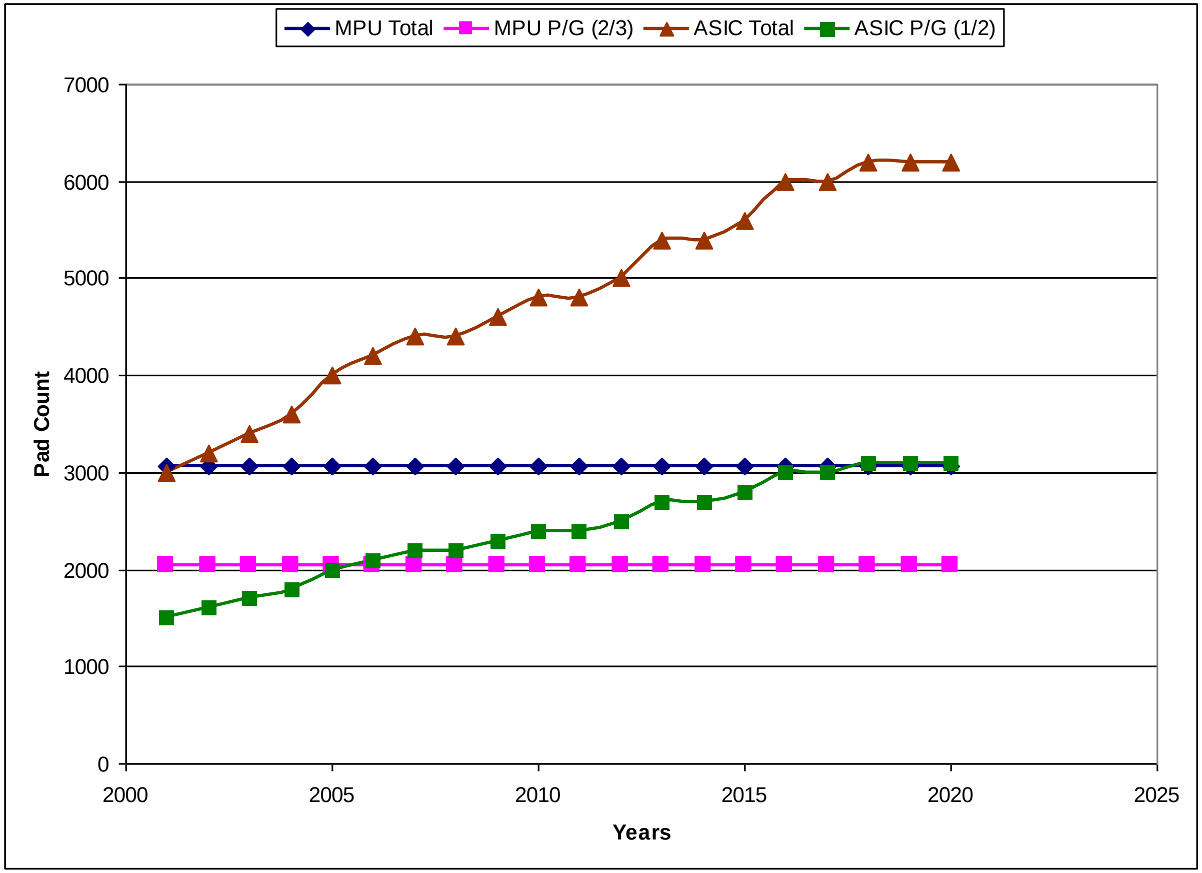
Max Supply Current (ITRS-03)



Max Supply Current (ITRS-06)



Chip Pads



Source: ITRS-05 & 06, http://public.itrs.net.

Outline

- The Power Problem
  - Impact, issues, objectives, trends
- Technology Trends and Projections
  - Historical trends (since 1960)
  - Future projections (up to 2020)
- EDA for Power Management
  - Low-level power models
  - Power estimation and optimization
  - High-level power models
    - Bottom-up
    - Top-down
- Conclusion

The Language of Power

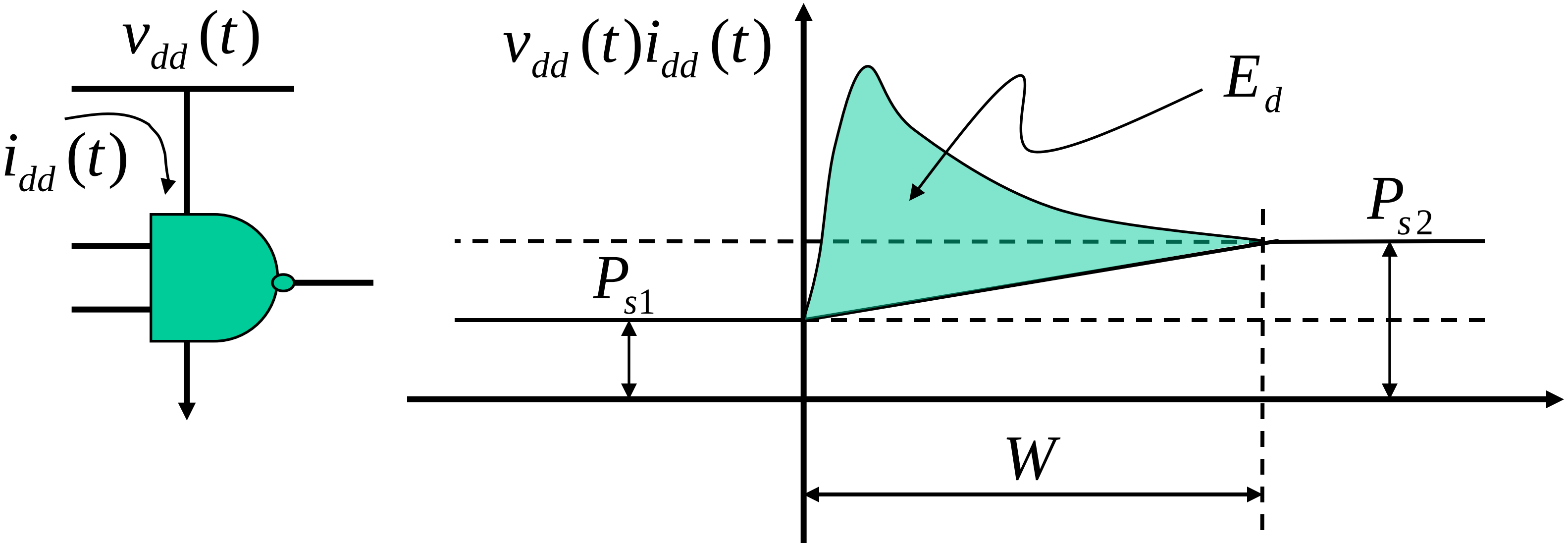
- Power modeling, estimation, and optimization methods have been in development since the late 80s
- A specialized terminology has developed
  - signal probability, static probability, transition probability, entropy, state probability, conditional probability
  - switching activity, transition density, transition activity, toggle rate, activity factor
  - spatial correlation, temporal correlation, spatio-temporal correlation, pair-wise correlation, correlation factors
  - independence, conditional independence, lag-one model
  - bottom-up, top-down models, cycle-accurate model, compaction
  - etc ...

Basics

- In order to introduce basic power concepts, it is helpful to consider the case of a single CMOS logic gate
- The static power, P_s, for a given input vector, is the constant power dissipated by the gate in steady state when that vector is applied
  - Due to "off-current" (leakage)
- The dynamic energy, E_d, for a given input (ordered) pair of vectors is the energy dissipated in the gate due to that vector transition.
  - Includes short-circuit, internal charging, and external charging currents
  - Energy = $QV = CV^2$ is measured in Joules, 1 J = 1 Watt-second

Avoid Double-Counting

- During a transition, we must avoid mixing up the static and dynamic components; here's one way of doing this:



$$E_{tot} = E_d + \left(\frac{P_{s1} + P_{s2}}{2}\right) \times W$$

Power Vector Set

- Power dissipation is well defined in connection with a given input vector sequence, called a power vector set
- Let T_{pvs} be the time duration of this vector set, then:
  - Transient power waveform is given by: $v_{dd}(t)i_{dd}(t)$
  - Average static power is given by: $$\frac{1}{T_{pvs}}\int_0^{T_{pvs}} P_s(t)dt$$
  - Average dynamic power is given by: $$\frac{1}{T_{pvs}}\sum_i E_d(i)$$

Signal Statistics

- Define: Indicator Function $I_x(t)$ of a signal $x(t)$:
  - $I_x(t)$ is 1 when $x(t)$ is at logic high
  - $I_x(t)$ is 0 when $x(t)$ is at logic low
  - $I_x(t)$ changes value instantaneously halfway through the transition window duration W
- Define: Signal Probability of a signal $x(t)$ as the fraction of time during which $x(t)$ is high; formally:

$$P(x) = P(x;0,T_{pvs}) = \frac{1}{T_{pvs}} \int_0^{T_{pvs}} I_x(t)dt$$

- The signal probability is a non-negative dimensionless real number between 0 and 1.

Switching Activity

- Define: Switching Activity of a signal *x(t)* as the average number of logic transitions per unit time; formally:

$$R(x) = R(x;0,T_{pvs}) = \frac{N_x(0,T_{pvs})}{T_{pvs}}$$

where $N_x(0, t)$ is the number of (complete) logic transitions of the signal between *0* and *t*.

- Other names for the switching activity:
  - switching rate, toggle rate, activity factor, transition activity, and transition density
- The switching activity is a non-negative real number with units of *transitions per second*

Outline

- The Power Problem
  - Impact, issues, objectives, trends
- Technology Trends and Projections
  - Historical trends (since 1960)
  - Future projections (up to 2020)
- EDA for Power Management
  - Low-level power models
  - Power estimation and optimization
  - High-level power models
    - Bottom-up
    - Top-down
- Conclusion

Single Gate Power

- Consider a gate with output node *x*, and let the static power when *x* is low (high) be P_{s0} (P_{s1}), then the <u>Average Static Power</u> is:

$$P_{avg,s} = \frac{1}{T_{pvs}} \left(P_{s1} \int_0^{T_{pvs}} I_x(t)dt + P_{s0} \int_0^{T_{pvs}} (1 - I_x(t))dt \right)$$

$$= P_{s1}P(x) + P_{s0} \big(- P(x) \big)$$

- Let E_d be the dynamic energy per transition, then the <u>Average Dynamic Power</u> is:

$$P_{avg,d} = \frac{E_d N_x(0, T_{pvs})}{T_{pvs}} = E_d R(x)$$

Different Rise/Fall Energies

- If the dynamic energy on a low to high transition is $E_{d,01}$ and is different from the dynamic energy on a high to low transition $E_{d,10}$, then:
  - If the initial state of the node is low, the expression for average dynamic power becomes:

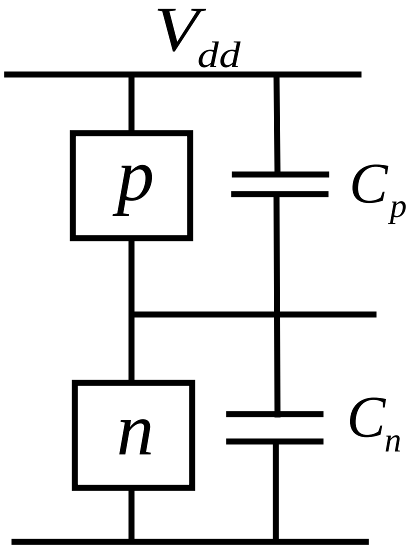
$$P_{avg,d} = \frac{E_{d,01}\left\lceil N_x(0,T_{pvs})/2 \right\rceil + E_{d,10}\left\lfloor N_x(0,T_{pvs})/2 \right\rfloor}{T_{pvs}}$$

$$\approx \left(\frac{E_{d,01} + E_{d,10}}{2}\right) R(x)$$

  - It is enough to talk about the average of $E_{d,01}$ and $E_{d,10}$ as the average dynamic energy per transition, E_d

Power and Capacitance

- Let us ignore the internal power of a gate and focus on the energy required to switch the output (drain and interconnect) capacitance:



$$C_{tot} = C_n + C_p$$

$$Q_{lh} = C_n V_{dd} \qquad Q_{hl} = C_p V_{dd}$$

$$Q_{lh+hl} = Q_{lh} + Q_{hl} = C_{tot} V_{dd}$$

$$E_{d,01} + E_{d,10} = Q_{lh} V_{dd} + Q_{hl} V_{dd} = C_{tot} V_{dd}^2$$

$$E_d = \tfrac{1}{2} C_{tot} V_{dd}^2$$

$$\Longrightarrow \boxed{P_{avg,d} = \tfrac{1}{2} C_{tot} V_{dd}^2 R(x)}$$

Clocked Circuits

- If the circuit is *clocked*, then it becomes convenient and natural to normalize by the clock period

- Let T be the clock period, and let $T_{pvs} = NT$, and let $t_0=0$, $t_1=T$, $t_2=2T$, …, $t_N=NT=T_{pvs}$, then:

$$P(x) = P(x;0,T_{pvs}) = \frac{1}{N}\sum_{i=1}^{N} P(x;t_{i-1},t_i)$$

- Thus, the signal probability is equal to ***the average fraction of a cycle in which the signal is high***.

Clocked Circuits

- **Switching activity is best normalized by the clock frequency, leading to a *revised definition* :**

$$D(x) = D(x;0,\tau) = \frac{1}{N}\sum_{i=1}^{N} N_x(t_{i-1},t_i)$$

  - **Thus, the (normalized) switching activity is *the average number of transitions per cycle*, which is a dimensionless non-negative real number**

- **The two definitions are related by:**

$$D(x) = T \times R(x)$$

  - **And the average dynamic power is now given by:**

$$\boxed{P_{avg,d} = E_d R(x) = E_d D(x) f}$$

The Zero-Delay Case

- **If all gates have zero-delay, then:**

$$P(x) = P(x;0,T_{pvs}) = \frac{1}{N}\sum_{i=1}^{N} I_x(i)$$

  - **where *$I_x(i)$* is the indicator function evaluated at any time during (but not at the start of) cycle *i***
  - **So, the signal probability becomes: *the average fraction of cycles in which the signal has a final value of logic high*.**

- **In this case, we also have:**

$$D(x) = P_t(x)$$

  - **where *$P_t(x)$* is the transition probability of *x(t)*, defined as: *the average fraction of clock cycles in which the final value of the signal is different from its initial value***

Outline

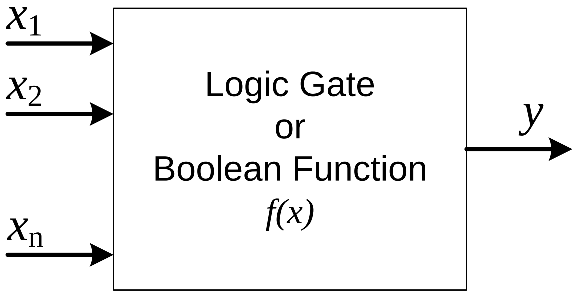
- The Power Problem
  - Impact, issues, objectives, trends
- Technology Trends and Projections
  - Historical trends (since 1960)
  - Future projections (up to 2020)
- EDA for Power Management
  - Low-level power models
  - Power estimation and optimization
  - High-level power models
    - Bottom-up
    - Top-down
- Conclusion

Power Estimation

- At a low-level, power estimation requires computation of the node switching activity *R(x)* or *D(x)*
  - A major difficulty is: pattern-dependence
- The obvious solution approach is to use simulation:
  - Simulate a power-vector set to measure the toggle count
  - Combine with capacitance information to compute the power
- Disadvantages: simulation is slow and expensive
- Many attempts have been made to develop "static" techniques to compute the power without simulation:
  - Analytical (probabilistic) methods
  - Monte Carlo (statistical) methods

Analytical Techniques

- Propagate switching activity directly on the netlist
- Key result: transition density propagation:



$$\boxed{D(y) = \sum_{i=1}^{n} P\left(\frac{\partial y}{\partial x_i}\right) D(x_i)}$$

- The partial derivative is the *Boolean difference*:

$$\boxed{\frac{\partial}{\partial x_i} f(x) = f(x)\big|_{x_i=0} \oplus f(x)\big|_{x_i=1}}$$

- Probability computation is possible using a BDD

Analytical Techniques

- A key assumption: inputs are uncorrelated
  - In practice, signals are often correlated, in complex ways
  - It is impractical to keep track of signal correlation
- As a result, overall, these methods are inaccurate
  - Total large circuit accuracy within 10%
  - Local error can be much larger
- Limited use:
  - Inside an optimization loop
  - Incremental analysis

Monte Carlo Techniques

- Randomly select and simulate (only) short vector sequences, from a large power-vector set
- Use Monte Carlo theory to converge on the mean, which is the average power
- Although more accurate, these methods have not displaced the simple simulation based approach
  - Dealing with unknown circuit state is a complication
  - The power vector set can itself be viewed as a collection of "samples" and can be chosen small enough to simulate

- *The method of choice today remains simulation, and that remains unsatisfactory*

Low-Power Synthesis

- Incorporate power as an objective during synthesis

- Practical experience shows that doing this during logic synthesis is not very effective
  - This refers to standard logic synthesis, employing Boolean optimizations followed by technology mapping
  - Once a circuit has been optimized for area, there is only 5-10% further improvement to be had by further power optimization

- ***It is generally agreed that optimizations at a higher level would have a higher impact***

Design Techniques for Power

- Despite nearly 20 years of research, there are no good truly EDA-driven solutions for power-aware design
  - Power Estimation is still done by simulation
  - Low Power Logic Synthesis is ineffective
- A number of “design techniques” have emerged:
  - For dynamic power control:
    - Clock gating , dynamic frequency and/or voltage control, low-power libraries
    - Duplication and parallelism … multicore processor architectures
  - For leakage power control:
    - Multiple supply voltages and/or voltage islands, multiple V_{th} transistors, sleep transistors and/or body bias control
- While they benefit from EDA, the above techniques are not EDA-centric

Duplication & Parallelism

- Single core:

$$\boxed{P_{avg} = \tfrac{1}{2} C_{tot} V_{dd}^{2} D_{avg} f}$$

- Reduce the supply voltage (slows down the circuit):

$$\boxed{P_{avg} = \tfrac{1}{2} C_{tot} \left(\frac{V_{dd}}{2} \right)^{2} D_{avg} \left(\frac{f}{2} \right)}$$

- Run two identical copies in parallel at the lower (half) frequency:

$$\boxed{P_{avg} = \tfrac{1}{4} \;\; \tfrac{1}{2} C_{tot} V_{dd}^{2} D_{avg} f}$$

Outline

- The Power Problem
  - Impact, issues, objectives, trends
- Technology Trends and Projections
  - Historical trends (since 1960)
  - Future projections (up to 2020)
- EDA for Power Management
  - Low-level power models
  - Power estimation and optimization
  - High-level power models
    - Bottom-up
    - Top-down
- Conclusion

High-Level Power Models

- Power estimation is required early in the design process in order to:
  - Avoid costly redesign, in case the power is too high
  - Enable design exploration in a design-reuse environment
- This requires power estimation at a high level of abstraction
- The key technology required is power models that can be used at various levels of abstraction

Two Types of Power Models

- Two types of models have been pursued:
  - Bottom-up (for hard macros)
    - Given complete low-level implementation, build a compact high level model for power in terms of vector stimulus
  - Top-down (for soft macros)
    - Given a high-level functional description (and no low-level implementation), along with a delay specification, a gate library, and I/O switching activity, predict the required power
- In practice, a high level design exploration environment may need to mix both types of power models

Outline

- The Power Problem
  - Impact, issues, objectives, trends
- Technology Trends and Projections
  - Historical trends (since 1960)
  - Future projections (up to 2020)
- EDA for Power Management
  - Low-level power models
  - Power estimation and optimization
  - High-level power models
    - Bottom-up
    - Top-down
- Conclusion

Bottom-Up Power Modeling

- Logic block internal details *are available*

- Generate a high level *power macromodel* :
  - Ward et al., Proc. IEEE, 1984
  - Powell and Chau, TCAS-91
  - Landman and Rabaey, TVLSI-95
  - Mehra and Rabaey, IWLPD-94
  - Mehta, Owens, and Irwin, DAC-96
  - Raghunathan, Dey, and Jha, ICCAD-96
  - Gupta and Najm, DAC-97
  - Qiu, Wu, Pedram, and Ding, ISLPED-97
  - etc. ...

Landman’s Method

- Block power model is in terms of:
  - Block I/O statistics
  - Block structural attributes
- Expert analysis is needed for a new type of function, in order to determine:
  - The type of model required
  - The type of characterization needed

Example

- For an SRAM with *W* words, & bit-width *N*:
  - The *average capacitance switched* per input bit, for a certain transition ID, is modeled as:

$$C = C_0 + C_1 W + C_2 N + C_3 WN$$

  - The C_i coefficients are obtained by a process of characterization and fitting
- From this, the power is simply $P_{avg} = CV^2 f$

Other Methods

- **Mehra's method (with Rabaey):**
  - Power model is $P_{avg}= N_a C_a V^2 f$, where:
    - N_a is the average number of accesses to the resource, per cycle
    - C_a is the average capacitance switched per access
  - Even though C_a depends on input statistics, they compute it up-front using UWN (Uniform White Noise)
  - System behavioral flowchart used to compute N_a

- **Raghunathan's method (with Dey and Jha):**
  - Estimate activity and power in presence of glitching activity
    - They use both analytical and look-up tables
    - Power macromodel is built in terms of 9 or more variables

Our Approach

- Power model : $P_{avg}=D_{avg}C_{tot}V^2f$, where D_{avg} is a function of input statistics
- What input statistics are sufficient to capture D_{avg} over a wide range of I/O data?
  - In terms of every input vector pair?
  - In terms of probability and activity at every input node?
- Can we express P_{avg} in terms of aggregate input/output statistics?

Aggregate Statistics

- Use a fixed model template, on only 4 variables
  - Can be a LUT or equation-based using RLS
  - Model size is fixed, independent of circuit size
- Variables are I/O switching activity statistics
  - Average input pin probability, P_{in}
  - Average input pin switching activity, D_{in}
  - Average pair-wise input pin joint probability, SC_{in}
  - Average output zero-delay pin switching activity, D_{out}
- Characterization process is automatic, using simulation
  - No user intervention is required
- Model evaluation is fast, based on the results of high-level simulation

Spatial Correlation

- Spatial correlation is defined as :

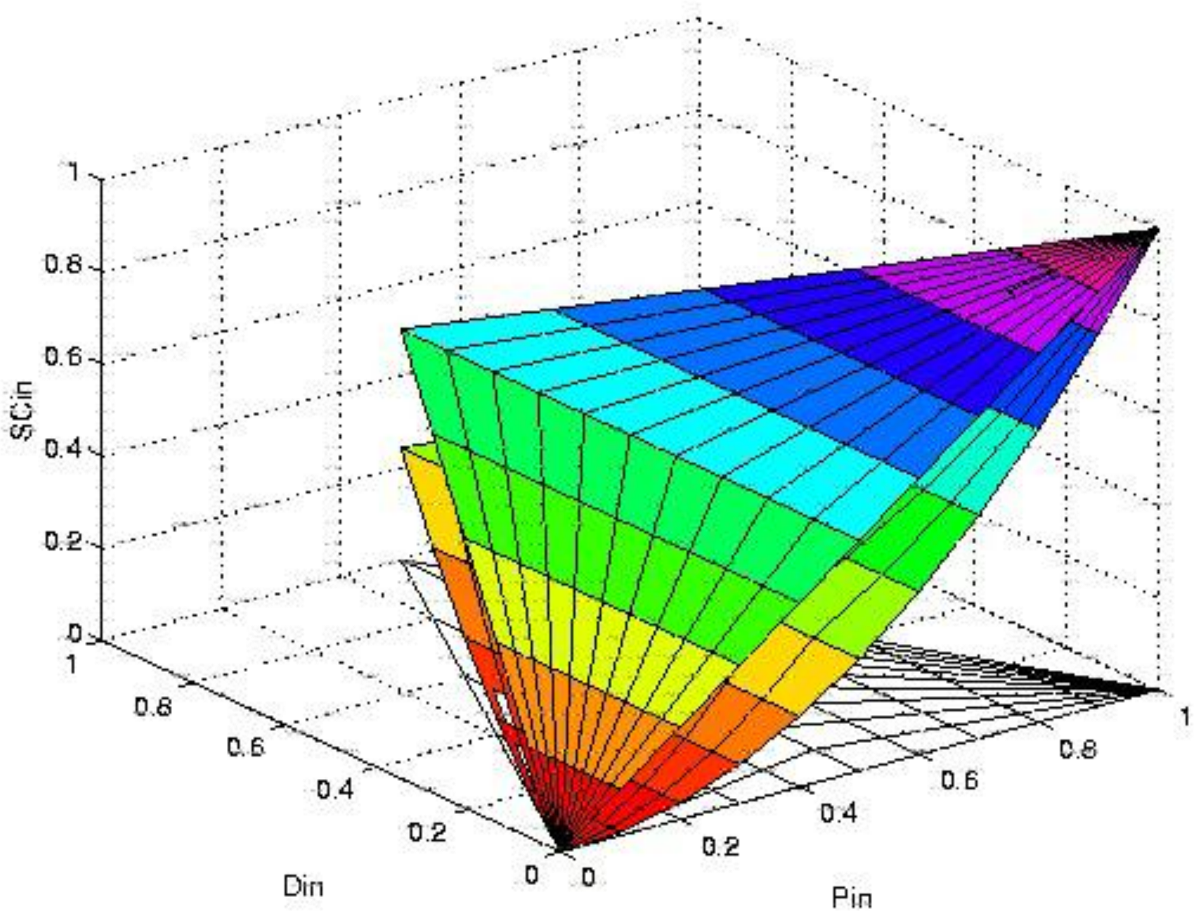
$$SC_{ij} = P\{x_i x_j = 1\}$$

- The 4-dimensional look-up table becomes :

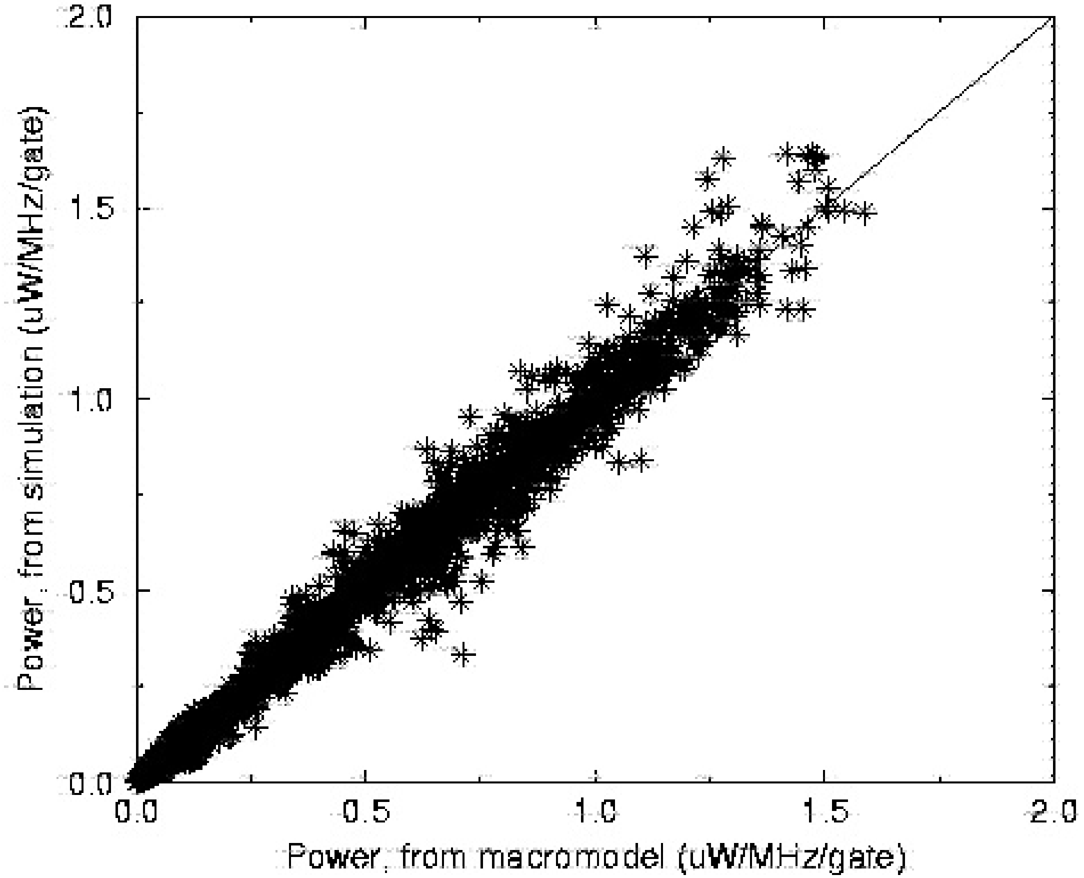
$$P_{avg} = f(P_{in}, D_{in}, SC_{in}, D_{out})$$

- Average error is 2-5%, worst case is -16% to +22%

Feasibility Region

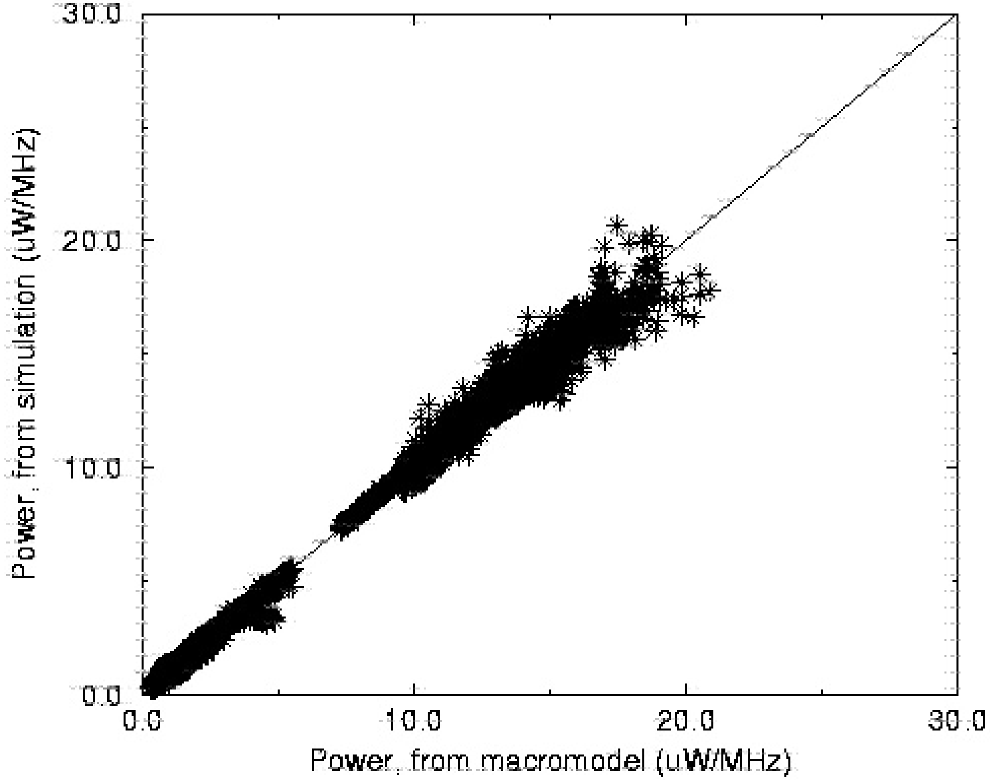


Combinational Ckts - RLS



Sequential Circuits

- State line statistics are implicit in the input vector statistics, assuming correlation dies down over time



Cost of Characterization

Circuit	#I	#O	#G	Model	Time	#RLS
c499	41	32	202	C	16.31m	258
c880	60	26	383	Q	29.10m	162
c1355	41	32	546	C	22.2m	230
c1908	33	25	880	Q	36.14m	120
c432	36	7	160	Q	22.52m	163
c5315	178	123	2307	Q	3.74hrs	286
c2670	233	140	1193	C	2.24hrs	207
c3540	50	22	1669	Q	5.08hrs	228
c7552	207	108	3512	Q	19.75hrs	601
c6288	32	32	2406	C	9.23hrs	286

Cost of Characterization

Circuit	#I	#O	#G	#FF	Model	Time	#RLS
s420	18	1	218	16	Q	56.96m	164
s641	35	24	379	19	Q	1.33hrs	179
s1196	14	14	529	18	Q	1.69hrs	2716
s510	19	7	211	6	Q	1.45hrs	147
s953	16	23	395	29	Q	5.5hrs	227
s298	3	6	119	14	Q	38.46m	288
s1238	14	14	508	18	Q	25.6m	456
s444	3	6	181	21	Q	22.2m	137
s820	18	19	289	5	Q	1.47hrs	141
s838	34	1	446	32	Q	2.86hrs	179
s5378	35	49	2779	179	Q	5.5hrs	89
s9234	36	39	5597	211	Q	14.64hrs	127

Outline

- The Power Problem
  - Impact, issues, objectives, trends
- Technology Trends and Projections
  - Historical trends (since 1960)
  - Future projections (up to 2020)
- EDA for Power Management
  - Low-level power models
  - Power estimation and optimization
  - High-level power models
    - Bottom-up
    - Top-down
- Conclusion

Background

- Estimation from a functional view is highly desirable
- Early history: Two methods based on *rough synthesis*:
  - Muller-Glasser et al., ICCAD-91
  - Svensson et al., IWLPD-94
- Function is synthesized as Boolean network, using reference gates (eg. NAND2)
- The gate count is multiplied by the average activity per gate, estimated under UWN.

- Subsequent work has explored other metrics, which would not rely on rough synthesis

Estimation without Synthesis

- Block internal details are *not* available
- One approximation seems inevitable:

$$P_{avg} \propto \sum_{i=1}^{N} C_i D(x_i) \cong D_{avg} \sum_{i=1}^{N} C_i = D_{avg} C_{tot}$$

- This approximation is very good in practice, leading to:

$$\boxed{P_{avg} = \tfrac{1}{2} C_{tot} V_{dd}^2 D_{avg} f}$$

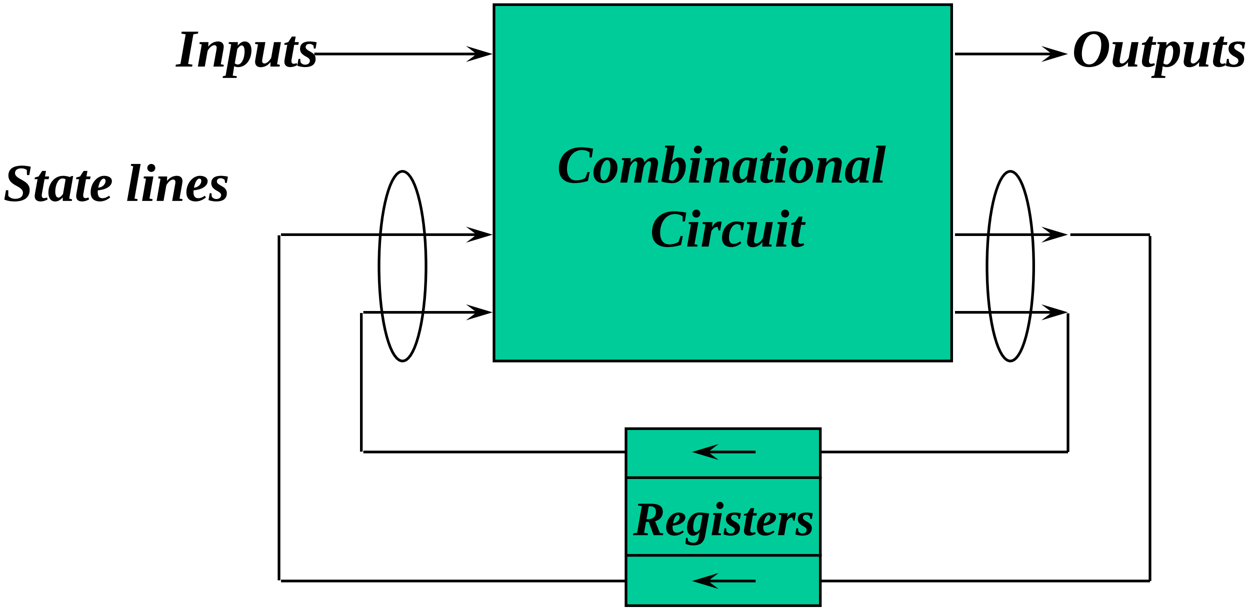
- Both D_{avg} and C_{tot} must be estimated from knowledge of the input/output behavior only!

Our Approach

- Power depends on the product of total capacitance and average switching activity
- Activity model:
  - Average internal activity in terms of average I/O activities
  - Several refinements
- Capacitance model:
  - Computed as the product of:
    - Predicted average capacitance per gate
    - Predicted gate count
- The resulting model is acceptable, but not final

Circuit Model

- Assume the circuit is described in terms of:
  - Registers or flip-flops, and
  - Boolean equations for the combinational logic blocks



Analysis Flow

- **Given a (structural) RTL view,**

  1. Run simulation to find the latch switching activity and power

  2. Predict average activity D_{avg} inside the combinational block

  3. Predict the total capacitance C_{tot} of the combinational block

  4. Compute average power as:

$$\boxed{P_{avg} = \tfrac{1}{2} C_{tot} V_{dd}^{2} D_{avg} f}$$

Average Activity Prediction

- Study *information transfer* across the block:
  - Najm, ISLPD-95
  - Marculescu et al., ISLPD-95
- Empirically, the scaled total switching activity per level varies *quadratically* with circuit depth
- This gives a model for the average switching activity that depends only on I/O activity
  - This model gives zero-delay activity prediction only
- We have also developed some refinements based on pair-wise signal correlations

Average Activity Model

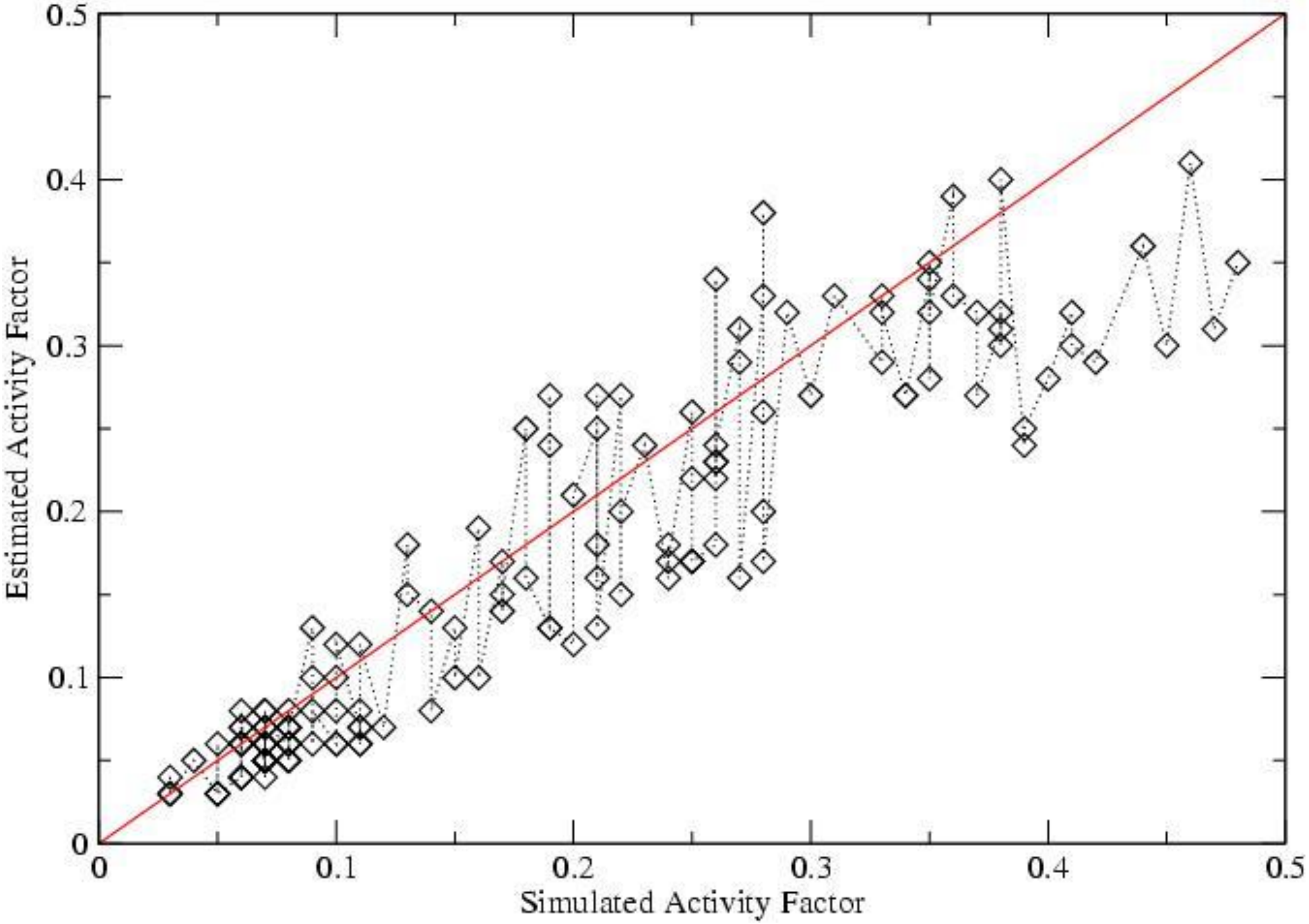
- Assuming the quadratic behavior, this leads to:

$$\boxed{D_{avg} \cong \frac{{}^{2}\!/_{3}}{m+n} \; D_i + 2D_o}$$

where:

- D_i is the sum of the input switching activities
- D_o is the sum of the output switching activities
- n is the number of input nodes
- m is the number of output nodes

Zero-Delay Activity Results



Capacitance Prediction

- Use $C_{tot} = AC_{avg}$, where:
  - A is an estimate of the gate count of an "optimal" (in some sense) implementation of the function
  - C_{avg} is the average gate + wire capacitance per gate

- We have estimated A in two ways:
  - A theoretically appealing model that benefits from the structure of the Boolean space of the function
  - A more practical empirical model based on Boolean networks and tuning with a given synthesis flow
- The practical model works better

Theoretical Approach

- Area model:
  - Define <u>complexity</u> of a Boolean function
  - Consider Randomly Generated Boolean Functions (RGBF)
  - Characterize RGBFs synthesized in the target library for dependence of area on function complexity
  - Typical logic functions exhibit similar dependence
  - For model evaluation, estimate complexity then use the RBGF model to get the area
- Limitations:
  - Not suitable for arithmetic blocks; good for control
  - Awkward concept of an output multiplexer and area recovery
  - Awkward handling of delay specification under synthesis
  - Model is expensive to evaluate

Area and Entropy

- Early history: Entropy has been used to predicted area of Randomly Generated Boolean Functions (RGBF)
- For *very small* input count "*n*", the average RGBF was found to have area:

$$A \propto 2^n H$$

- This model is empirical
- As well, for *very large* "*n*", it has been shown theoretically that the area complexity is exponential

Previous Model is Unrealistic

- When applied to moderate size functions that are typical of VLSI, this model breaks down
  - Predicts an area of ~ 400 million gates, for a 32-input function which can be built with 84 gates
    - The area function fitting was done for RGBFs with 25 inputs
- Typical VLSI functions are very different from RGBFs - they are not "average"

Our Model

- In addition to entropy, we use the structure of the Boolean ON-set/OFF-set of the function
- Based on Average Cube Complexity *C(f)* :
  - Weighted average number of literals in a prime implicant
  - Obtained by random sampling and logic simulation
- We have found that (approximately):

$$\boxed{A \propto 2^{C(f)} H}$$

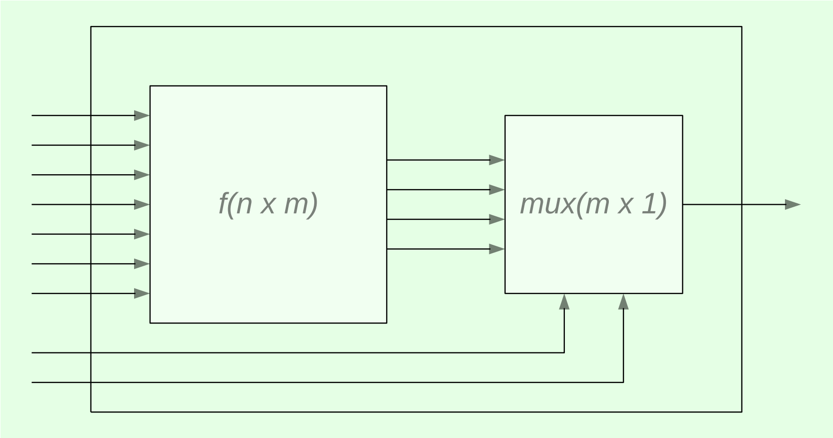
Limitations of Proposed Model

- It is expensive to find the average cube complexity

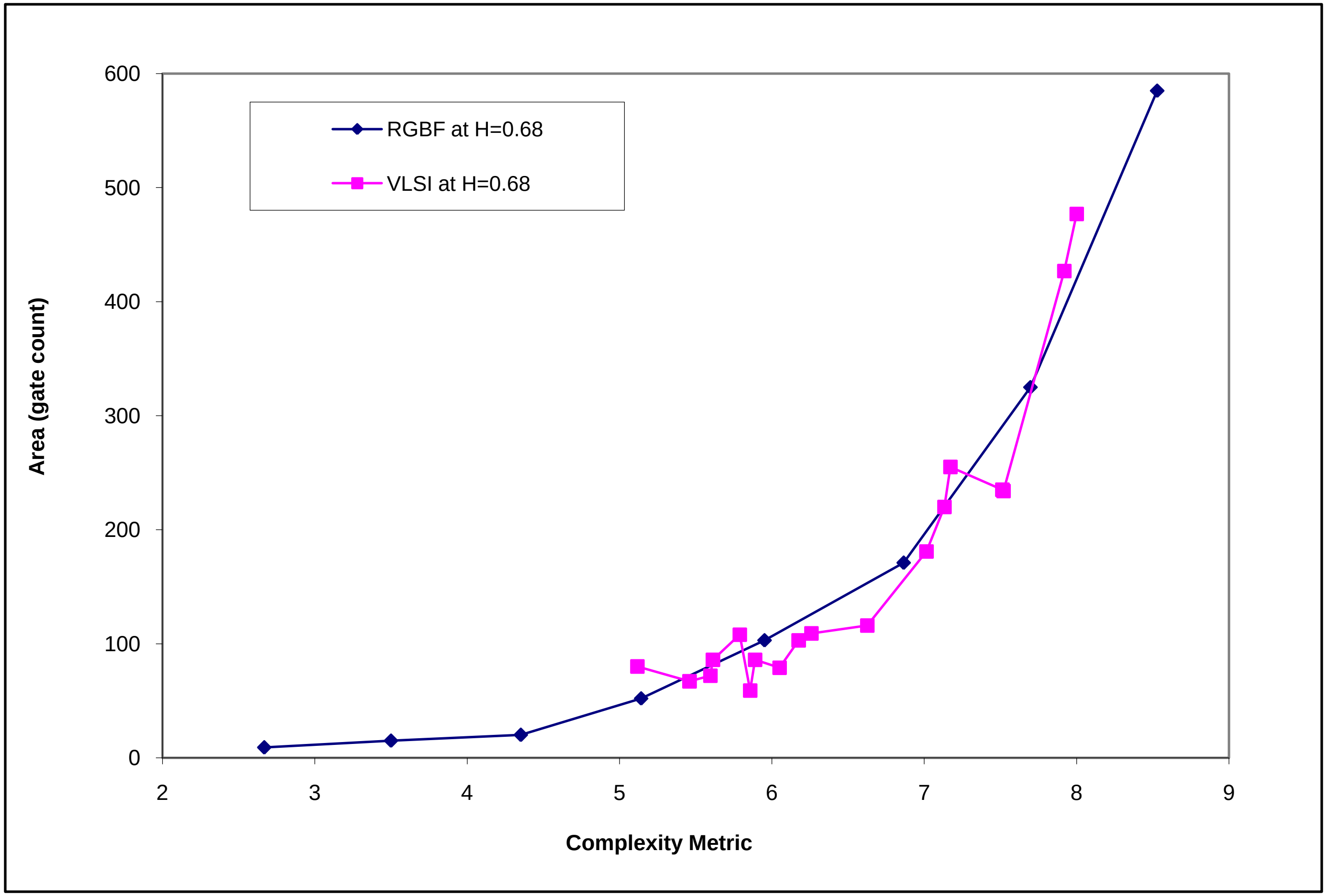
- This model breaks down for functions that contain <u>arrays of exclusive-or gates</u>
  - This includes adders and multipliers
  - These circuits are perhaps best modeled via a bottom-up power macromodel
  - This model is perhaps best suited for control-intensive applications
  - The following results exclude XOR circuits

Area Prediction Technique

- General procedure (for multi-output functions):
  - Transform to a single output function
  - Compute single-output function complexity
  - Compute single-output function area
  - Recover multi-output function area

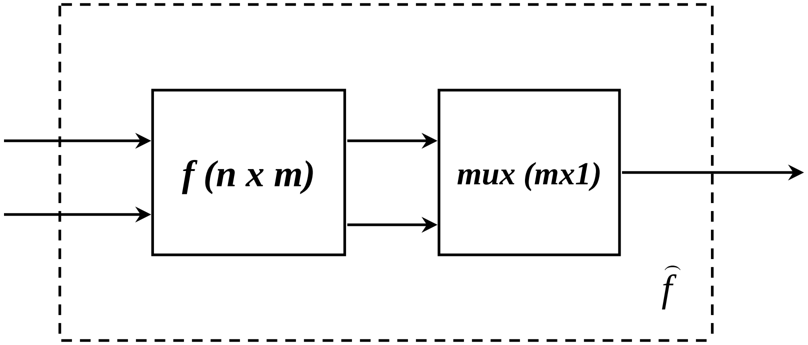


Single Output Area Model

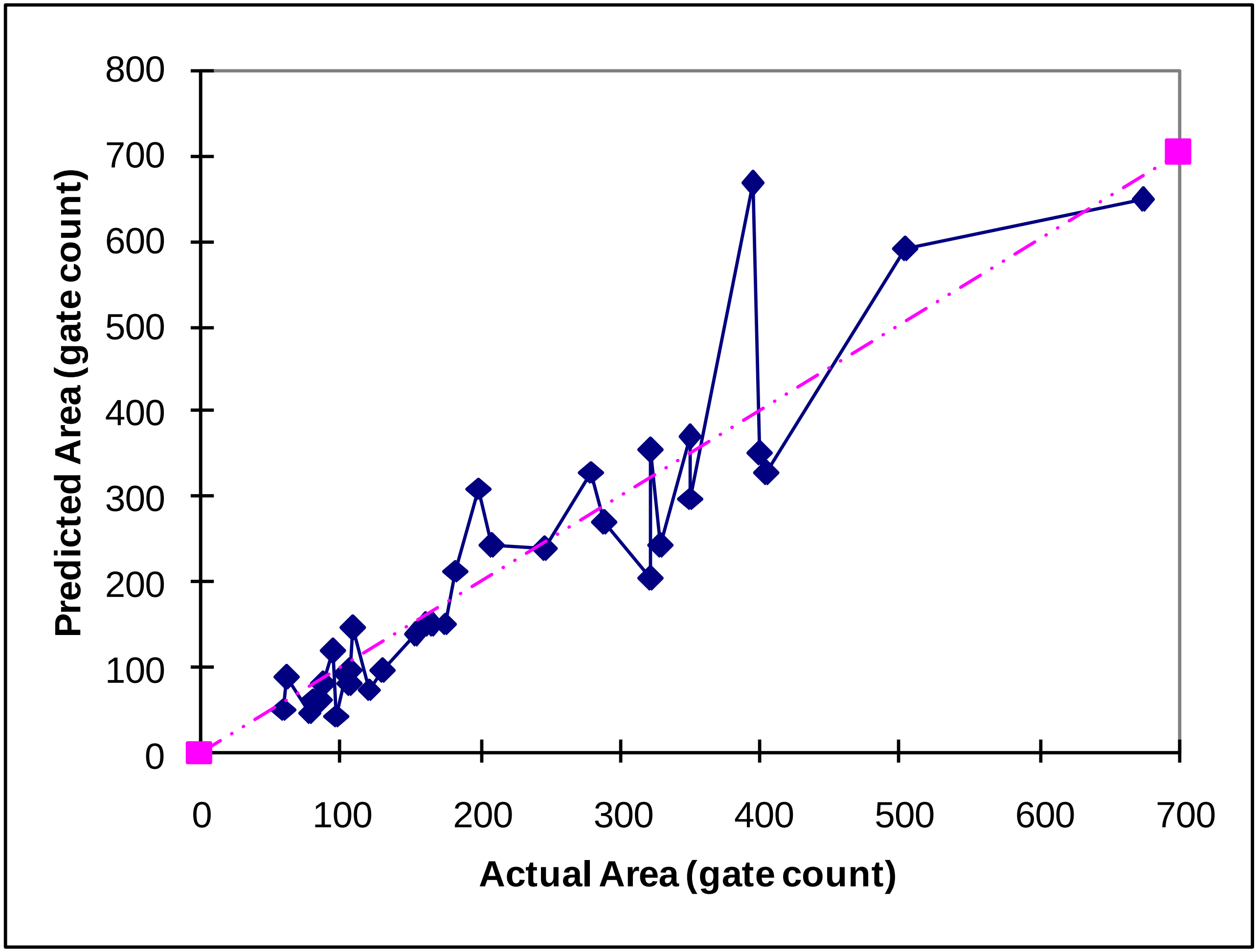


Area Recovery

- $A(f) = A(\hat{f}) - \alpha A_{mux}$
  - α lies between 0 and 1
- The fraction, α , depends on the average number of control inputs in the prime implicants of $\hat{f}$
- Based on this observation, an empirical model for estimating α was derived



Results


800
700
600
500
400
300
200
100
0
Predicted Area (gate count)
0
100
200
300
400
500
600
700
Actual Area (gate count)

A More Practical Approach

- Problem:
  - Previous method inaccurate and computationally expensive
  - Need fast and relatively accurate alternatives

- Possible solution:
  - Use a Boolean Network (BN) representation of RTL
  - Pros: very easy to build and makes use of established graph theoretical methods
  - Cons: not a canonical representation

Boolean Networks

- Various measures can be defined to capture the complexity of the BN
  - Node count, fan-in, fan-out, cut sets…
- We need a complexity measure that
  - Is invariant for different BNs of the same Boolean function
  - Captures the gate count requirements of the actual circuit
- Define area complexity measure as

$$\boxed{C(B) = n \cdot f_{in} \cdot f_{out}}$$

where:

- n : number of nodes in the BN
- f_{in} : average fan-in of the nodes in the graph
- f_{out} : average fan-out of the nodes in the graph

Complexity Measure

- Can be written as

$$C(B) = E_{out} \cdot f_{in}$$

where:

- f_{in}: average fan-in of the BN
  - Rough measure of gate-count (silicon cost) of the node
  - Represents the amount of computation done in the nodes
- $E_{out} = n\, f_{out}$: total number of all outgoing edges
  - A measure of connectedness of the graph
  - Represents the overall communication in the graph

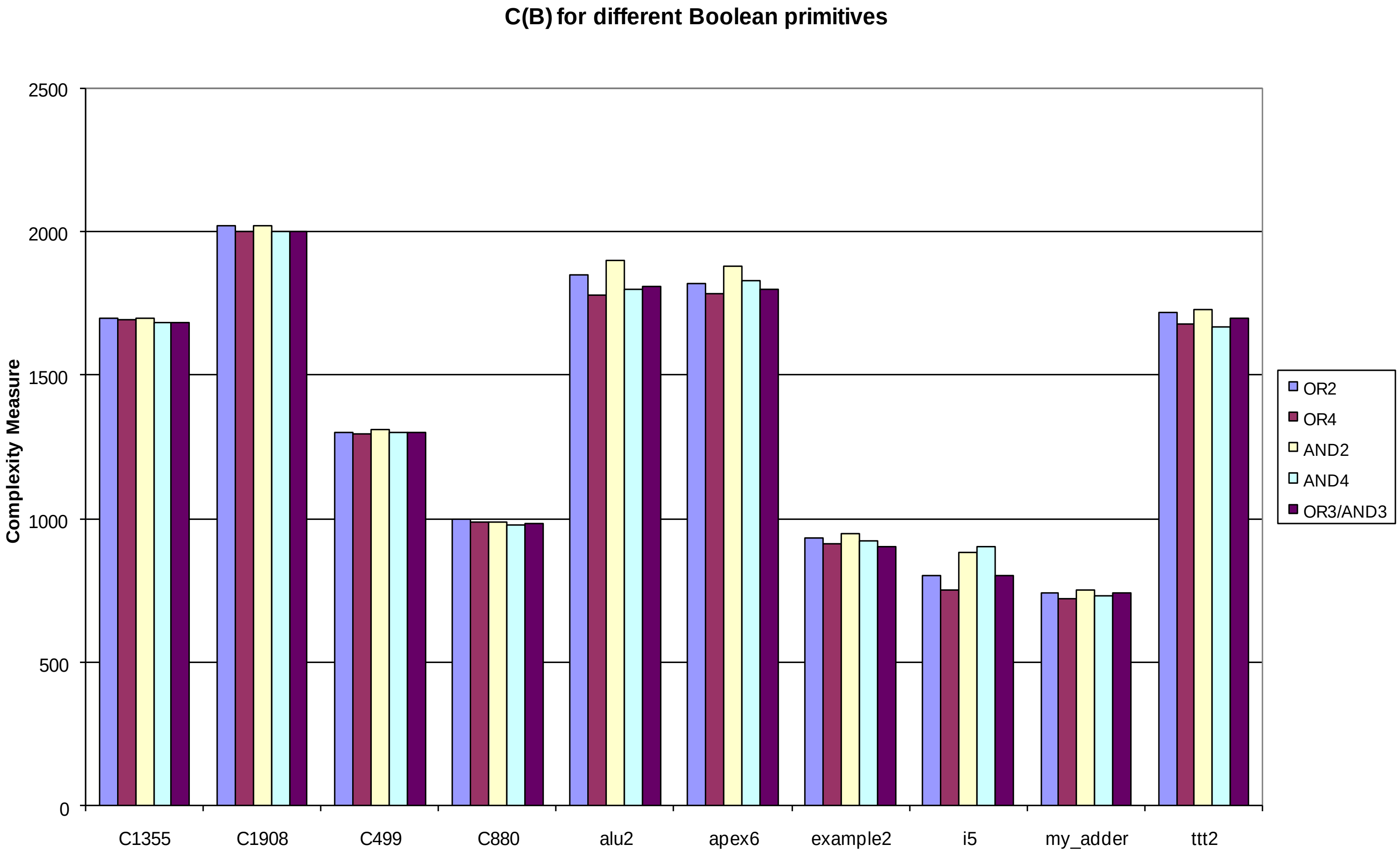
Complexity Measure

- As f_{in} increases
  - More computation is performed in the nodes
  - Less communication is needed
  - Typically, E_{out} decreases

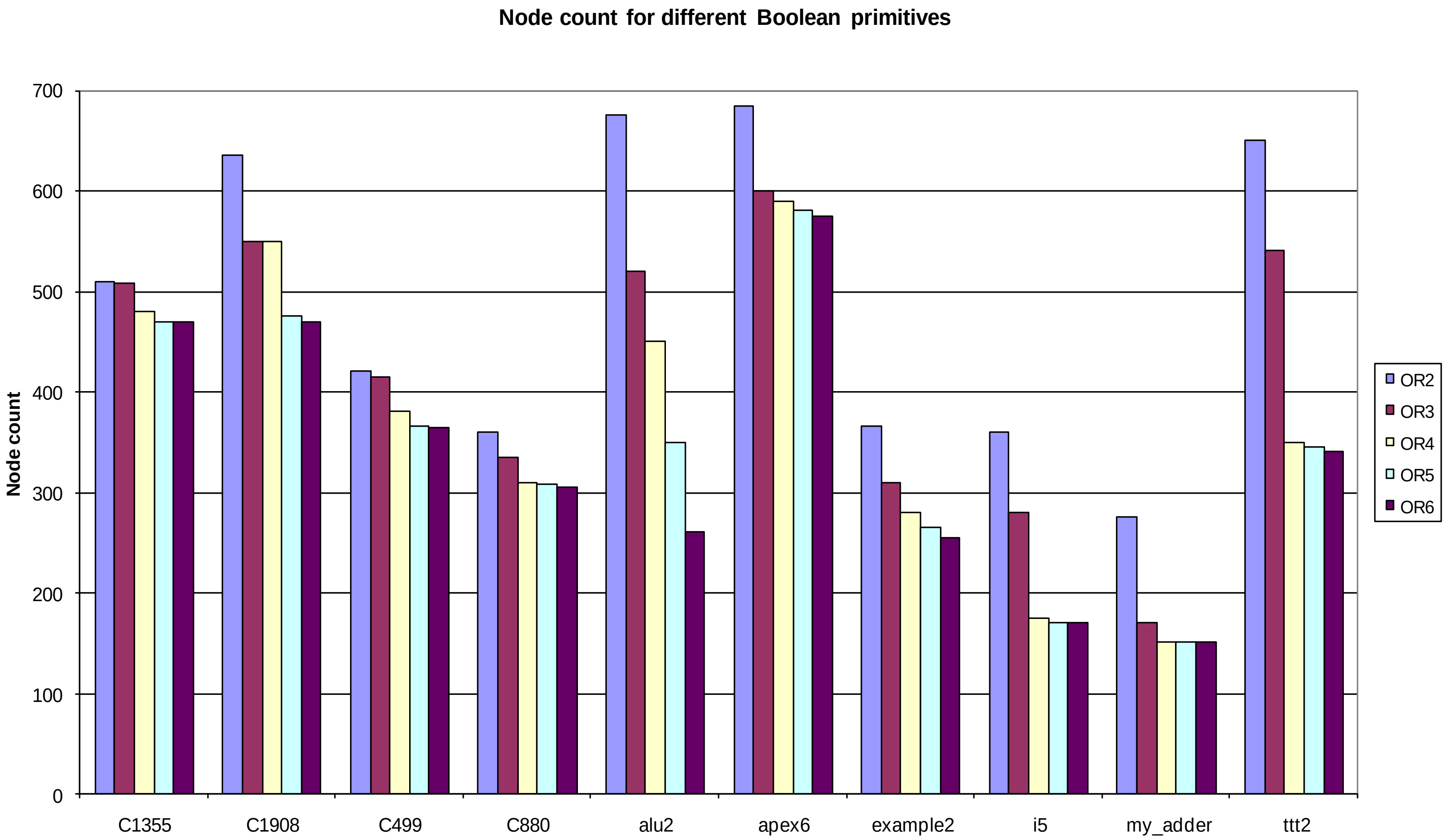
- As f_{in} decreases
  - Less computation is performed in the nodes
  - More communication (intermediate signals) are needed
  - Typically, E_{out} increases

- Our observation: $\boxed{f_{in} \cdot E_{out} \approx const}$

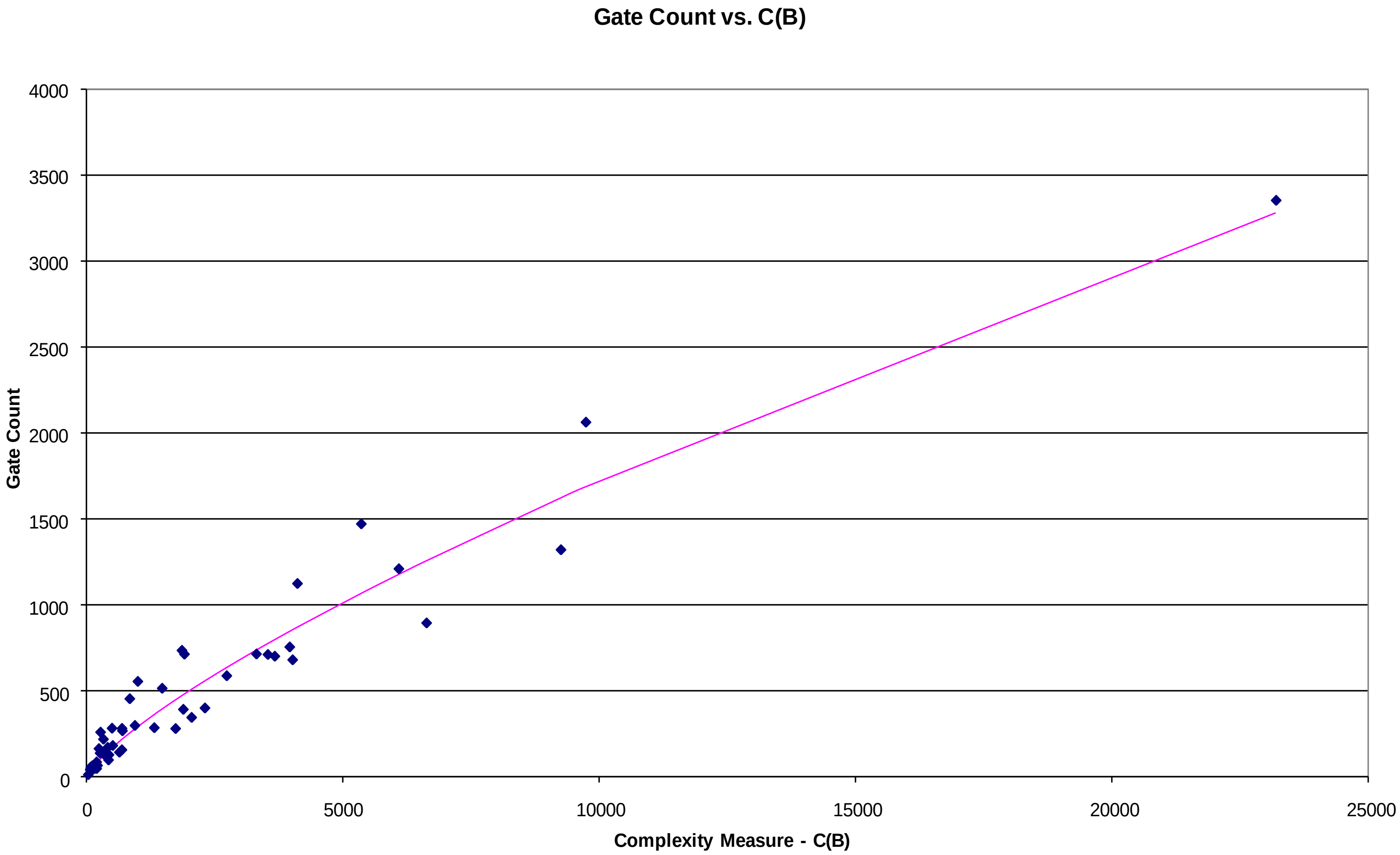
Complexity Measure



Complexity Measure



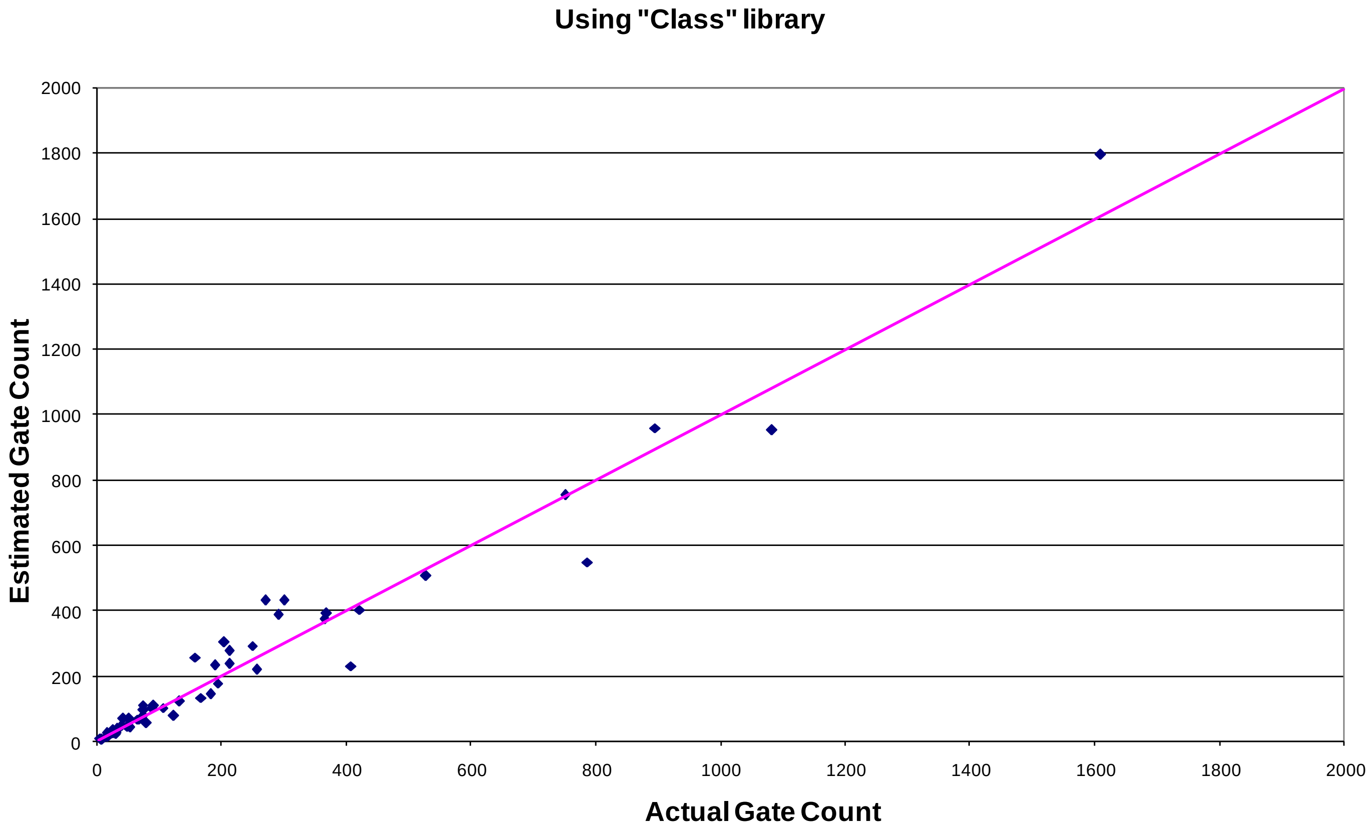
Complexity Measure



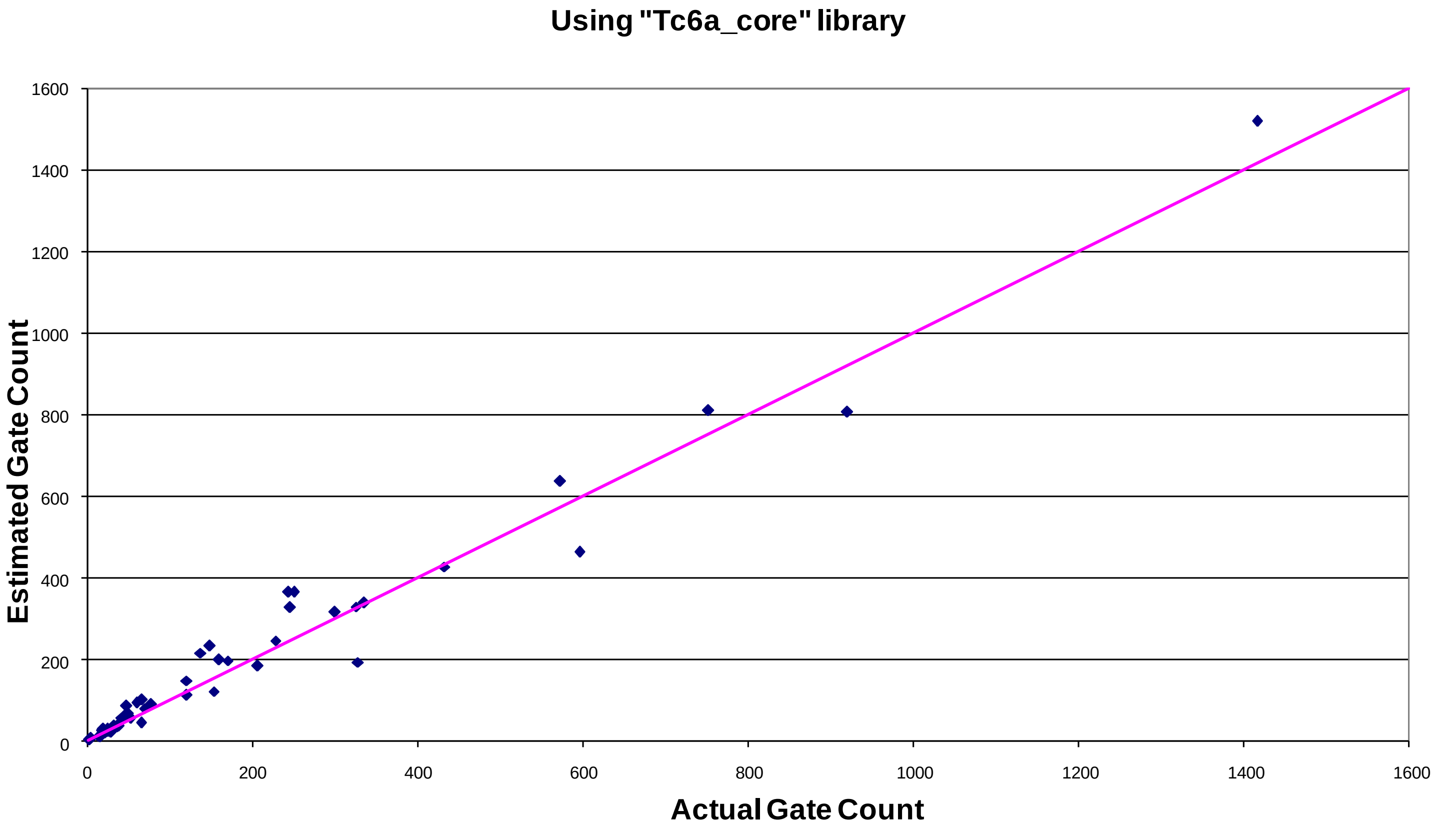
Gate Count Estimation Flow

- Characterize the synthesis/mapping tool and the target library (up-front one-time cost)
  - Choose a number of previously synthesized designs
  - Build the BNs corresponding to each
  - Compute the area complexity measure C(B)
  - Establish a model for the relationship between C(B) and the gate-count (regression analysis, or look-up tables)
- Gate count estimation
  - Build the BN for a new candidate design
  - Compute its area complexity measure
  - Using the above model, find the gate count

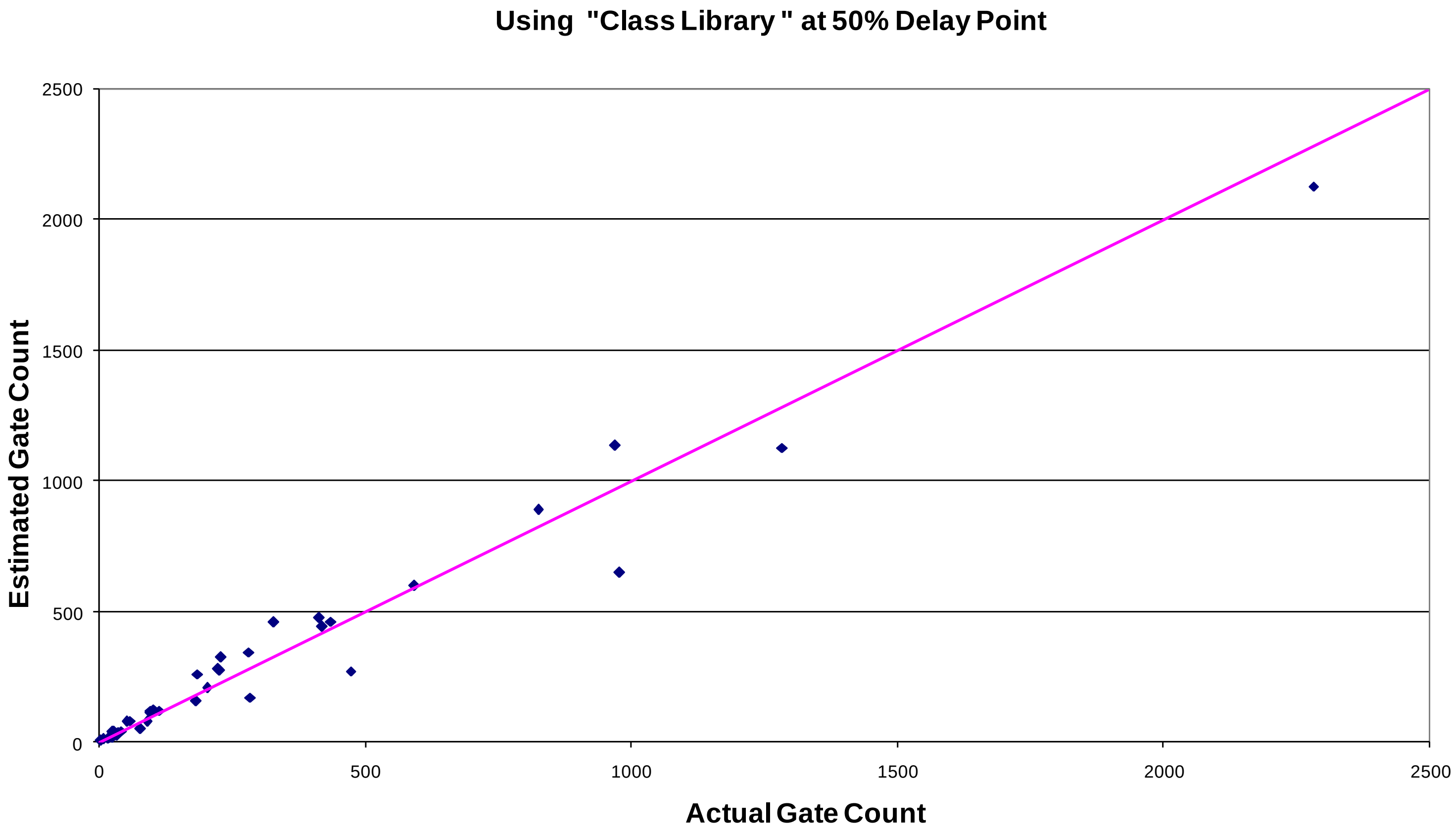
Results – Gate Count Estimation



Results – Gate Count Estimation



Results – Gate Count Estimation

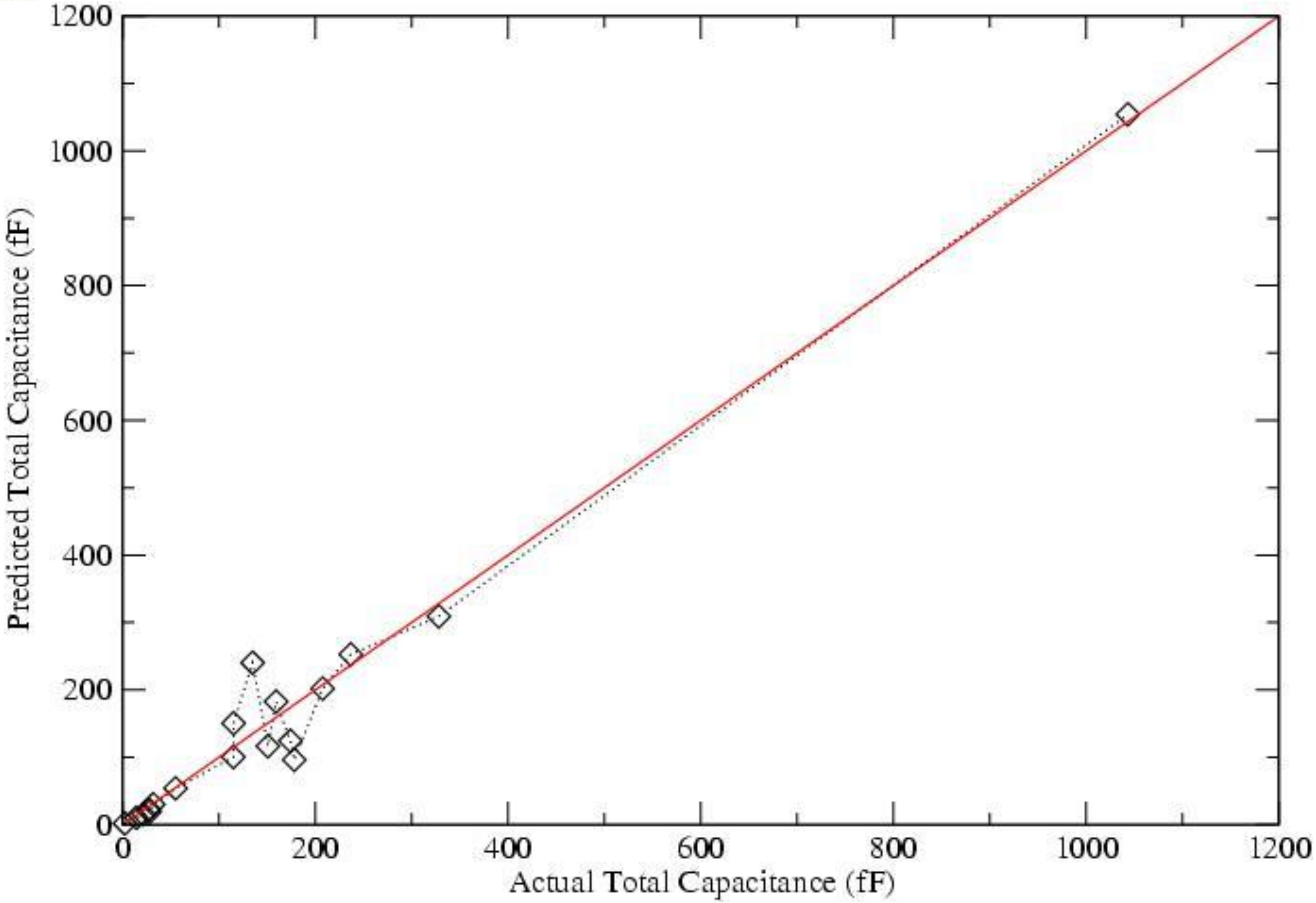


Capacitance Estimation Flow

- Pre-computing C_{avg} (average capacitance per gate) for the target library (characterization)
  - Obtain total capacitance for a number of synthesized designs in the target library
  - Divide by gate count to obtain c_{avg} for each circuit
  - Average these numbers to get C_{avg} for the library

- Estimation of C_{tot} (total capacitance) for a given circuit (estimation)
  - Estimate the gate count using the *C(B)* model
  - Obtain C_{tot} by multiplying the gate count by C_{avg}

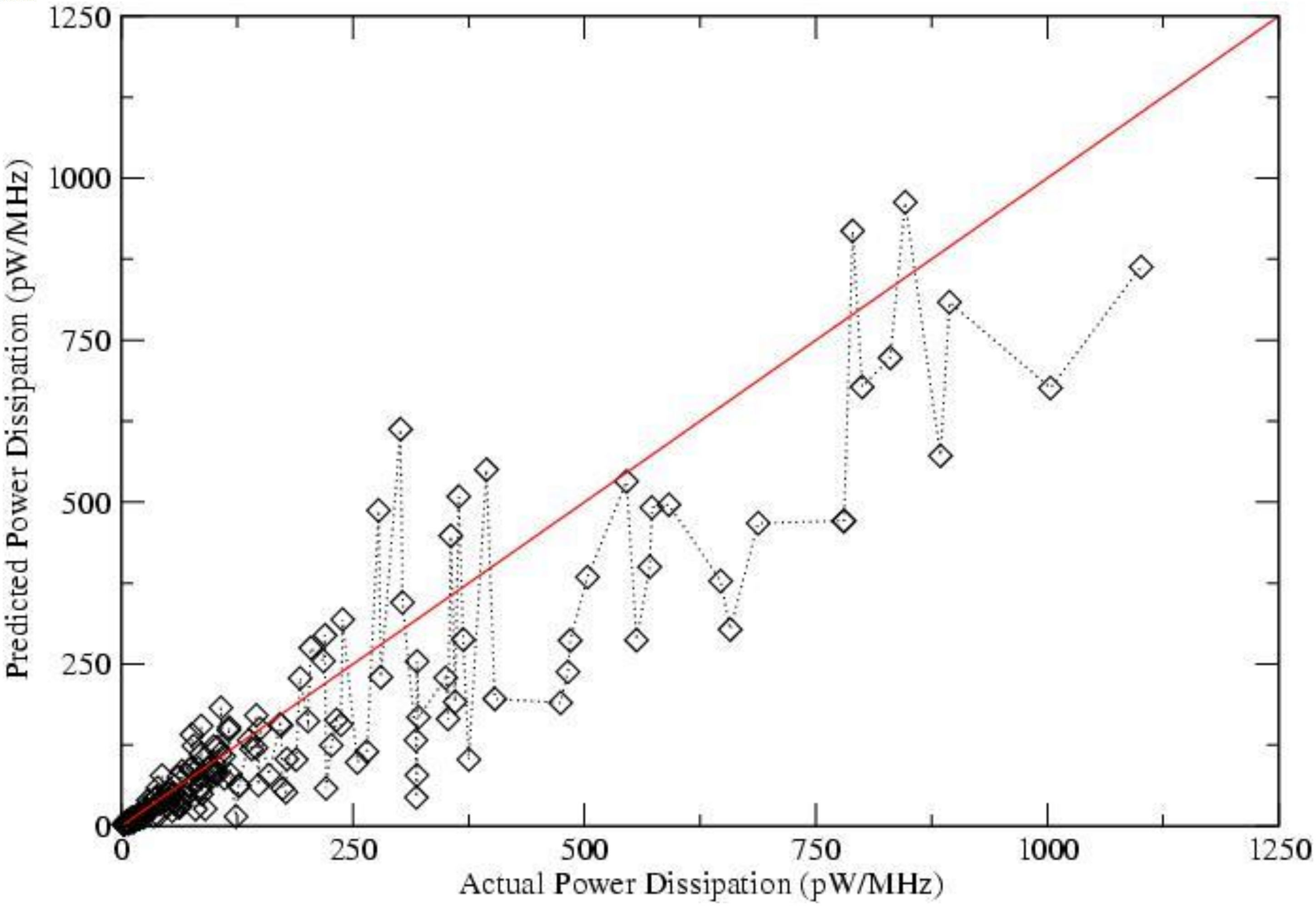
Results – Capacitance Estimation



Overall Power Estimation Flow

- Read in:
  - The Boolean equations describing the design
  - The input statistics and delay constraints
- Build a BN representation of the function
- Estimate the gate count and the total capacitance
- Run a zero-delay logic simulator to get the output switching activity, D_o, given the input activity, D_i
- Estimate the average (internal) switching activity
- Estimate the power

Results (zero-delay power)



Outline

- The Power Problem
  - Impact, issues, objectives, trends
- Technology Trends and Projections
  - Historical trends (since 1960)
  - Future projections (up to 2020)
- EDA for Power Management
  - Low-level power models
  - Power estimation and optimization
  - High-level power models
    - Bottom-up
    - Top-down
- Conclusion

Conclusion

- Power is a primary concern for IC design
  - Dynamic power limits how fast we can clock an IC
  - Leakage power limits how much we can reduce the supply
  - Power is why the industry moved to CMOS 20 years ago
  - Power is why the industry is moving to multi-core processors
- Several design techniques have been adopted to mitigate the power problem
- EDA contributions at the netlist/logic/RTL have had a limited impact
- It is felt that high-level early EDA intervention would have a higher impact on the power of the final IC