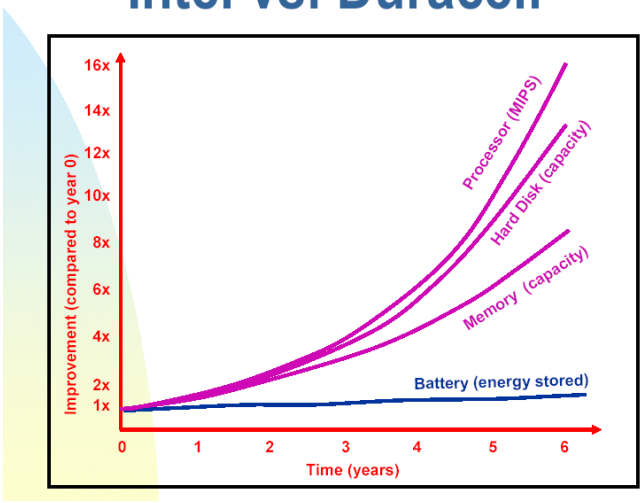
VLSI Design Fundamentals

Low-Power Design

Professor Yusuf Leblebici
Microelectronic Systems Laboratory (LSM)

yusuf.leblebici@epfl.ch

Why worry about power? Intel vs. Duracell



Current Battery Technology is Inadequate

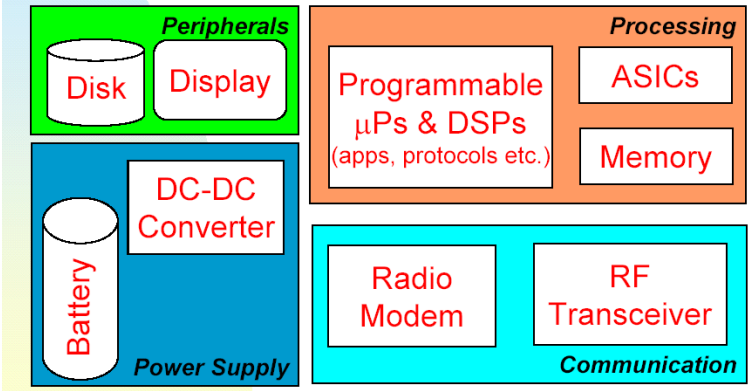
Battery	Rechargeable?	Wh/lb	Wh/litre
Alkaline MnO2	NO	65.8	347
Silver Oxide	NO	60	500
Li/MnO2	NO	105	550
Zinc Air	NO	140	1150
NiCd	YES	23	125
Li-Polymer	YES	65-90	300-415

- Example: 20-watt battery
  - NiCd weighs 0.5 kg, lasts 1 hr, and costs $20
  - Comparable Li-Ion lasts 3 hrs, but costs > 4x more

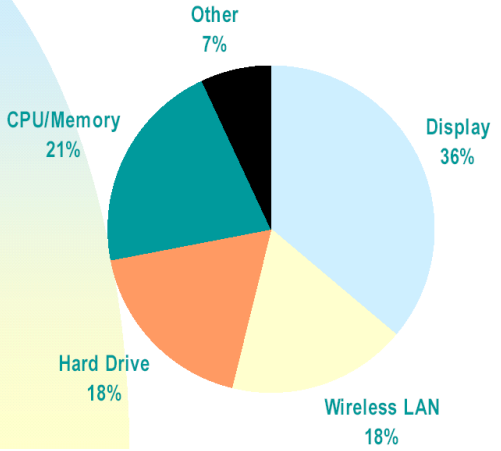
Barriers to Future Voltage Scaling

- Voltage scaling requires threshold voltage V_t to be scaled as well (15% per generation)
  - this increases sub-threshold leakage current
  - impact on power consumption and circuit robustness
- Leakage power
  - total leakage current goes up 7.5x per generation
  - leakage power power by 5x
  - soon will become a significant portion of total
    - active power remains constant for constant die size
  - leakage power, and therefore total power, can be substantially reduced by cooling
    - Essential to control die temperature
    - power density (W/cm^2): 0.6 micron chips surpassed a hot plate!

Where does the Power Go?



Power Consumption for a Computer with Wireless NIC



Power Consumption in Post-PC Devices

- Pocket computers, PDAs, wireless pads, wireless sensors, pagers, cell phones
- Energy and power usage of these devices is markedly different from laptop and notebook computers
  - much wider dynamic range of power demand
  - share of memory, communication and signal processing subsystems become more important
    - disk storage and displays disappear or become simpler
- Design of power-aware higher layer applications and protocols need to be re-evaluated

Metrics for Power

- Power
  - sets battery life in hours
  - problem: power $\propto$ frequency (slow the system!)
- Energy per operation
  - fixes obvious problem with the power metric
  - but can cheat by doing stuff that will slow the chip
    - Energy/op = Power * Delay/op
- Metric should capture both energy and performance: e.g. Energy/Op * Delay/Op
  - Energy*Delay = Power*$(\text{Delay/Op})^2$

Power

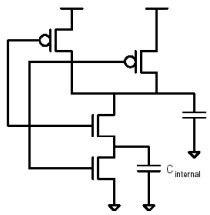
- As complexity and performance has increased, so has the power dissipated in CMOS chips. Today, some CMOS chips dissipate over 50W. In addition to this growing power dissipation in the high-end chips, there is a large interest in portable electronics, where low power operation is very important. Thus power has become an important design parameter to optimize.
- In CMOS circuits most of the power is dissipated in charging and discharging the load capacitance when a gate switches. This so-called dynamic power is consumed when charge is pulled from the Vdd supply to charge up the capacitor, and then this charge is returned to the Gnd supply when the capacitor is discharged. Since energy was needed to add this charge to the power supply, this action looses energy.

Power in CMOS gates

$$P_{total} = \underbrace{\alpha \cdot C \cdot V^2 \cdot f_{clk}}_{\text{Dynamic}} + \underbrace{Vdd \cdot I_{short-circuit}}_{\text{Short-circuit}} + \underbrace{Vdd \cdot I_{leakage}}_{\text{Leakage}}$$

- Power is governed by the 3 part equation above
  - Dynamic CV^2F (switching) power
    - Currently the largest part, but can make it smaller by lowering Vdd and Vth
  - Leakage Power
    - Subthreshold conduction – gets larger when Vth is lowered
    - Reverse leakage of diodes
  - Short-circuit (crowbar) current
    - Both pull-up and pull-down devices are partially conducting for a small, but finite amount of time
    - Small (<10%) in a well designed circuit, acts like dynamic power (happens only on a transition) so we won't worry about it here

A Closer Look at Average Dynamic Power

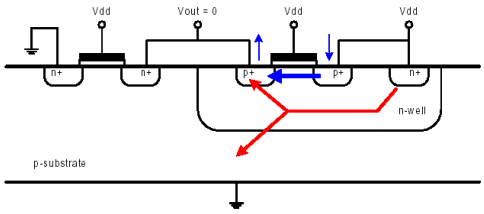


$$P_{dynamic} = \frac{1}{2}\alpha\ C_{switched} \cdot Vdd \cdot V_{swing} \cdot f_{CLK}$$

$$C_{drain} + \sum C_{wire} + \sum C_{gate}$$

- Capacitance (C)
  - Gate and parasitic source/drain capacitances
  - Wires or interconnects
- V^2 = Vdd * V_{swing}
  - Power supply voltage (Vdd) and output swing (V_{swing})
- f_{CLK} - frequency of operation
- Activity factor (α – the number of times the node switches/cycle)
  - not all nodes switch every cycle – data dependent
  - Internal nodes can switch without changing the output

Leakage Power



$$I_{subthreshold} = I_O \cdot e^{\frac{q(V_{gs}-V_T)}{\alpha kT}}$$

$$I_{reverse} = A \cdot J_S\left(e^{\frac{qVbias}{kT}} - 1\right)$$

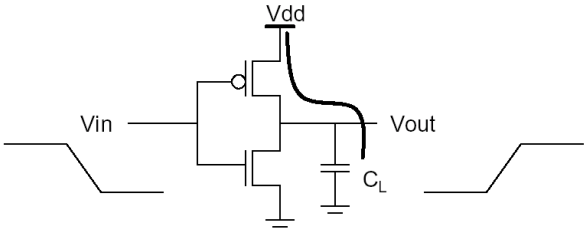
- Mostly depends on processing parameters and operating conditions (i.e. temperature and voltage)
- Reverse leakage current of reverse biased pn-junctions
- Subthreshold conduction in pMOS although *OFF*
- Getting to be larger percentage due to aggressive scaling
  - Leakage in memory arrays is becoming more of a concern

$$P_{leakage} = Vdd \cdot \left(I_{reverse} + I_{subthreshold}\right)$$

Low-Power Options

- Reduce α and C
  - Reduce switching activity by optimizing algorithms, architecture, logic topology, etc.
  - Glitch reduction
  - Clock gating
    - Clocks consume >30% of power
- Reduce Vdd and/or clock frequency
  - Both will reduce performance
- Reduce V_{swing}
  - Low swing busses and logic
  - But then you need to figure out how to use small signals
- Good Engineering!

Dynamic Power Consumption



$$\text{Energy/transition} = C_L * V_{DD}^2 * P_{0\rightarrow1}$$

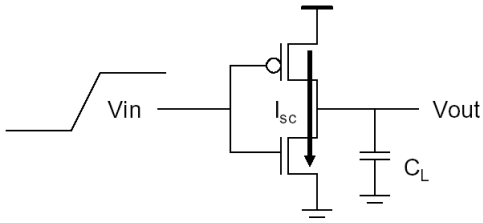
$$P_{dyn} = \text{Energy/transition} * f = C_L * V_{DD}^2 * P_{0\rightarrow1} * f$$

($f_{0\rightarrow1}$)

$$P_{dyn} = C_{EFF} * V_{DD}^2 * f \quad \text{where } C_{EFF} = P_{0\rightarrow1} C_L$$

Not a function of transistor sizes!
Data dependent - a function of switching activity!

Short Circuit Power Consumption



Finite slope of the input signal causes a direct current path between V_{DD} and GND for a short period of time during switching when both the NMOS and PMOS transistors are conducting.

Review: Energy & Power Equations

$$E = C_L V_{DD}^2 P_{0\to1} + t_{sc} V_{DD} I_{peak} P_{0\to1} + V_{DD} I_{leakage}$$

$$f_{0\to1} = P_{0\to1} * f_{clock}$$

$$P = C_L V_{DD}^2 f_{0\to1} + t_{sc} V_{DD} I_{peak} f_{0\to1} + V_{DD} I_{leakage}$$

Dynamic power (~90% today and decreasing relatively)	Short-circuit power (~8% today and decreasing absolutely)	Leakage power (~2% today and increasing)

Reducing the Power Dissipation

The power dissipation can be minimized by reducing:

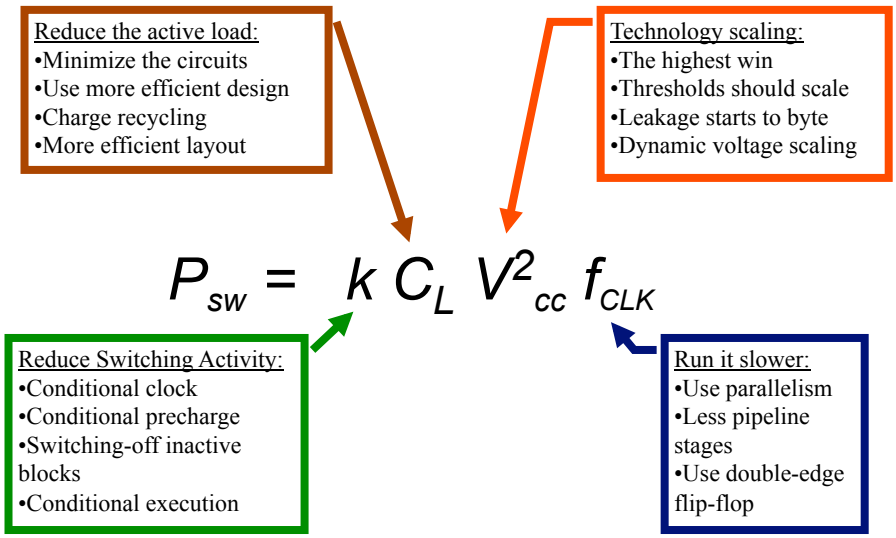
supply voltage

load capacitance

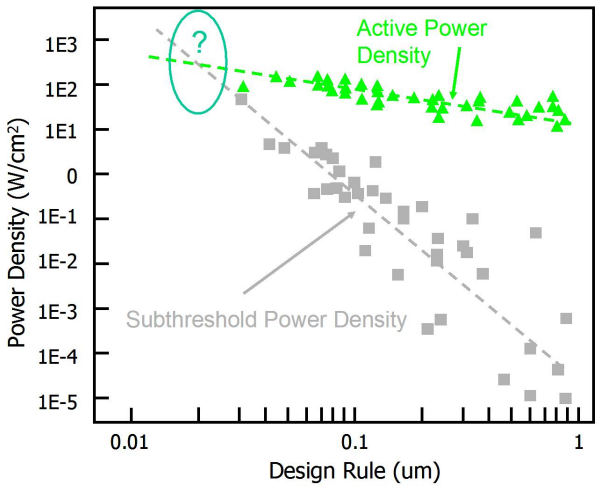
switching activity

Reducing the supply voltage brings quadratic improvement.

Reducing the load capacitance contributes to the improvement of both power dissipation and circuit speed.



Disturbing Predictions for Power



Integrated System Design

19

TSMC Processes Leakage and V_T

	CL018 G	CL018 LP	CL018 ULP	CL018 HS	CL015 HS	CL013 HS
V_{dd}	1.8 V	1.8 V	1.8 V	2 V	1.5 V	1.2 V
T_{ox}(effective)	42 Å	42 Å	42 Å	42 Å	29 Å	24 Å
L_{gate}	0.16 µm	0.16 µm	0.18 µm	0.13 µm	0.11 µm	0.08 µm
I_{DSat} (n/p) (µA/µm)	600/260	500/180	320/130	780/360	860/370	920/400
I_{off} (leakage) (ρA/µm)	20	1.60	0.15	300	1,800	13,000
V_{Tn}	0.42 V	0.63 V	0.73 V	0.40 V	0.29 V	0.25 V
FET Perf. (GHz)	30	22	14	43	52	80

Integrated System Design

20

Controlling V_{DD} and V_{TH} for low power

Low power → Low V_{DD} → Low speed → Low V_{TH} → High leakage → V_{DD}-V_{TH} control

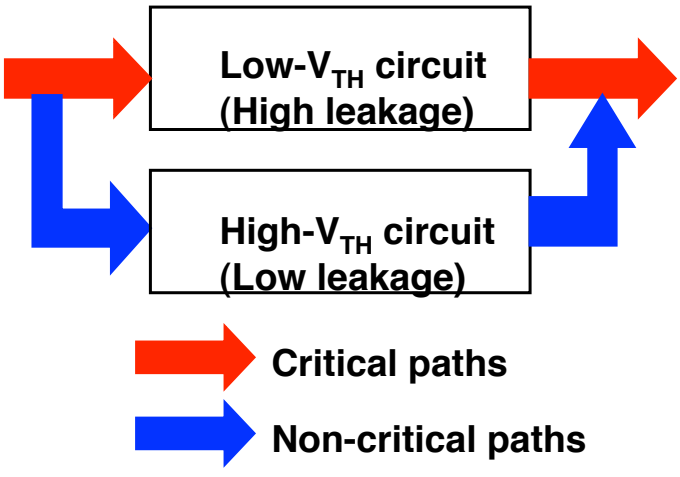
	Active	Stand-by
Multiple V_{TH}	Dual-V_{TH}	MTCMOS
Variable V_{TH}	V_{TH} hopping	VTCMOS
Multiple V_{DD}	Dual-V_{DD}	Boosted gate MOS
Variable V_{DD}	V_{DD} hopping	

Software-hardware cooperation

Technology-circuit cooperation

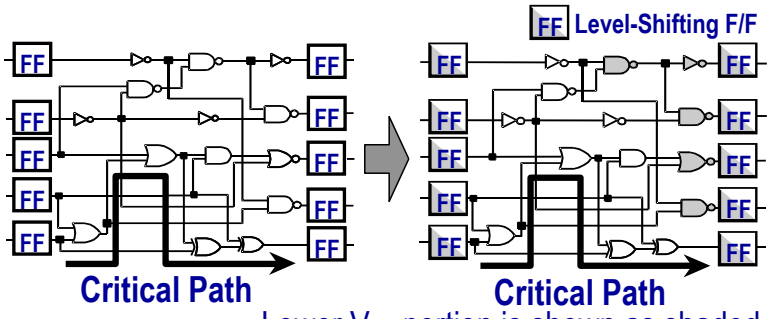
*) MTCMOS: Multi-Threshold CMOS
*) VTCMOS: Variable Threshold CMOS
• Multiple : spatial assignment
• Variable : temporal assignment

Dual-V_{TH} concept



Clustered Voltage Scaling for Multiple V_{DD}'s

Conventional Design **CVS Structure**



Lower V_{DD} portion is shown as shaded

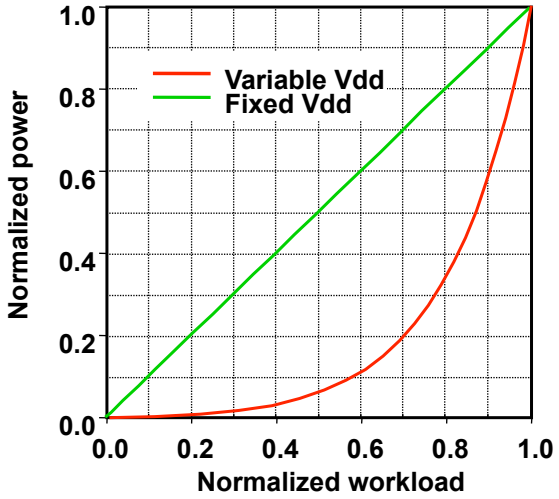
Once V_L is applied to a logic gate, V_L is applied to subsequent logic gates until F/F's to eliminate DC current paths. F/F's restore V_H.

M.Takahashi et al., "A 60mW MPEG4 Video Codec Using Clustered Voltage Scaling with Variable Supply-Voltage Scheme," ISSCC, pp.36-37, Feb.1998.

V_{DD} should be as low as possible

Energy consumption is proportional to the square of V_{DD}.

V_{DD} should be lowered to the minimum level which ensures the real-time operation.



TransMeta Example

LongRun Technology Demonstration

MHz	Voltage	% Full Power
700	1.65	100%
400	1.4	41%
333	1.2	25%

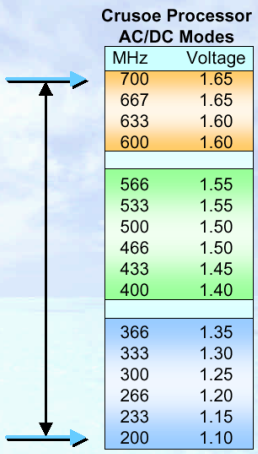
Power = C x V^2 x F = 400MHz/700MHz * $1.4V^2/1.65V^2$ = 41%

- Crusoe processor starts off at 700MHz
- DVD movie requires between 333 and 400MHz
- Power is reduced to 25 or 41% of full power
- The result is extended DVD playtime

TransMeta Example

LongRun Technology in Operation

- Crusoe processor starts off at 700MHz
- Code Morphing software detects user activity
- The software dynamically adjusts MHz and voltage to the most efficient power level

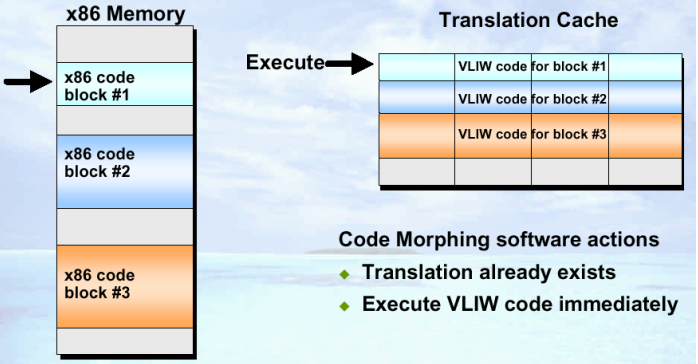


Crusoe Processor AC/DC Modes

MHz	Voltage
700	1.65
667	1.65
633	1.60
600	1.60
566	1.55
533	1.55
500	1.50
466	1.50
433	1.45
400	1.40
366	1.35
333	1.30
300	1.25
266	1.20
233	1.15
200	1.10

TransMeta Example

Dynamic Software Execution (2nd Pass)



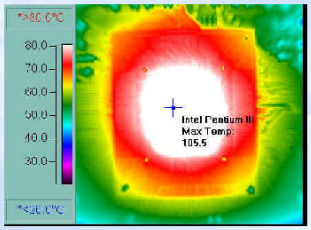
Code Morphing software actions

- Translation already exists
- Execute VLIW code immediately

*Taken from Doug Laird's presentation, January 19 th 2000

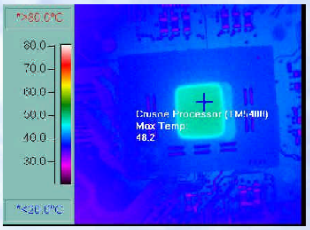
TransMeta Example

Processor Thermal Comparison



Pentium III Playing DVD

105.5º C
221.9º F



Crusoe Processor Playing DVD

48.2º C
118.8º F

Why is lowering VDD not enough ?

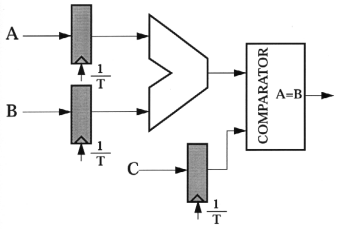
- Total P can be minimized by lower V
  - lower V are a natural result of smaller feature sizes
- But… transistor speeds decrease dramatically as V is reduced to close to "threshold voltage"
  - performance goals may not be met
  - $t_d = CV / k(V-V_t)^{\alpha}$ where α is between 1-2
- Why not lower this "threshold voltage"?
  - makes noise margin and I_{leak} worse!
- Need to do smarter voltage scaling!

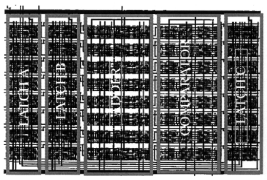
Reducing the Supply Voltage: Architectural Approach

- Operate at reduced voltage at lower speed
- Use architecture optimization to compensate for slower operation
  - e.g. concurrency, pipelining via compiler techniques
- Architecture bottlenecks limit voltage reduction
  - degradation of speed-up
  - interconnect overheads
- Similar idea for memory: slower and parallel

Trade-off AREA for lower POWER

Example: Reference Datapath

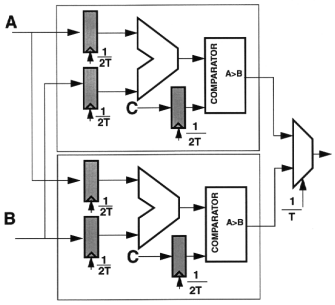


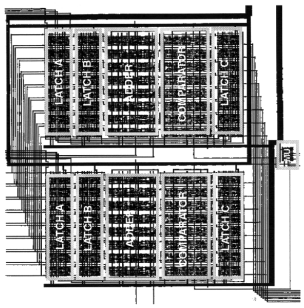


Area = 636 x 833 u^2

- Critical path delay: $T_{adder} + T_{comparator} = 25$ ns
- Frequency: $f_{ref} = 40$ MHz
- Total switched capacitance = C_{ref}
- $V_{dd} = V_{ref} = 5V$
- Power for reference datapath = $P_{ref} = C_{ref}V_{ref}^2f_{ref}$

Parallel Datapath

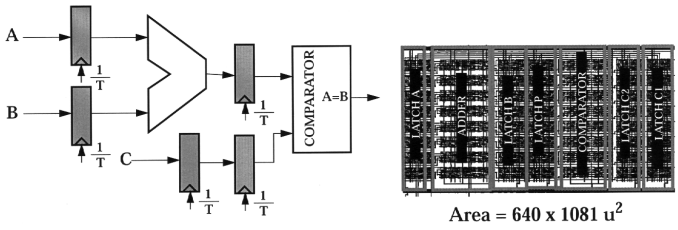




Area = 1476 x 1219 μ^2

- The clock rate can be reduced by x2 with the same throughput: $f_{par} = f_{ref}/2 = 20$ MHz
- Total switched capacitance = $C_{par} = 2.15C_{ref}$
- $V_{par} = V_{ref}/1.7$
- $P_{par} = (2.15C_{ref})(V_{ref}/1.7)^2(f_{ref}/2) = 0.36P_{ref}$

Pipelined Datapath



- $f_{pipe} = f_{ref}$
  $C_{pipe} = 1.1C_{ref}$
  $V_{pipe} = V_{ref}/1.7$
- Voltage can be dropped while maintaining the original throughput
- $P_{pipe} = C_{pipe}V_{pipe}^2 f_{pipe} = (1.1C_{ref})(V_{ref}/1.7)^2 f_{ref} = 0.37P_{ref}$

Datapath Architecture-Power Trade-off Summary

Datapath Architecture	Voltage	Area	Power
Original	5V	1	1
Pipelined	2.9V	1.3	0.37
Parallel	2.9V	3.4	0.34
Pipeline-Parallel	2.0V	3.7	0.18

Power Dissipation Depends on Switching Activity !

Power = Energy/transition * transition rate

$$= C_L * V_{dd}^2 * f_{0\rightarrow 1}$$

$$= C_L * V_{dd}^2 * P_{0\rightarrow 1} * f$$

$$= C_{EFF} * V_{dd}^2 * f$$

Power Dissipation is Data Dependent
Function of *Switching Activity*

C_{EFF} = **Effective Capacitance** = $C_L * P_{0\rightarrow 1}$

Example: Static 2 Input NOR Gate

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

Truth Table of 2 input NOR gate

Assume:
P(A=1) = 1/2
P(B=1) = 1/2

Then:

P(Out=1) = 1/4
P(0→1)
= P(Out=0).P(Out=1)
= 3/4 × 1/4 = 3/16

$$C_{EFF} = 3/16 * C_L$$

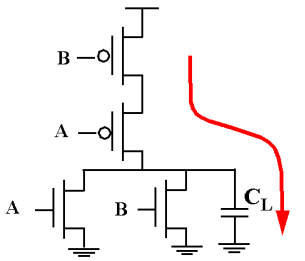
Transition Probabilities for Basic Gates

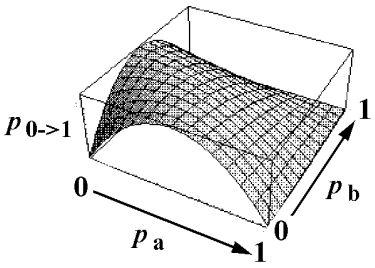
	$P_{0\rightarrow1}$
AND	$(1-P_AP_B)P_AP_B$
OR	$(1-P_A)(1-P_B)(1-(1-P_A)(1-P_B))$
EXOR	$(1-(P_A+P_B-2P_AP_B))(P_A+P_B-2P_AP_B)$

Switching Activity for Static CMOS

$$P_{0\rightarrow1} = P_0.P_1$$

Transition Probabilities for Basic Gates





$$p_1 = (1-p_a)\,(1-p_b)$$

$$p_{0->1} = p_0\,p_1 = (1-(1-p_a)\,(1-p_b))\,(1-p_a)\,(1-p_b)$$

- $\alpha_{0->1}$ **is a strong function of signal statistics**