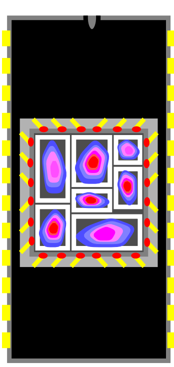
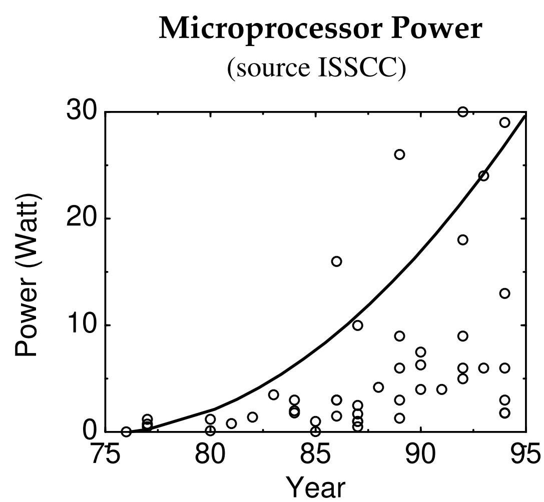
Low Energy & Power Design Issues

- Processor trends
- Circuit and Technology Issues
- Architectural optimizations
- Low power µP research project

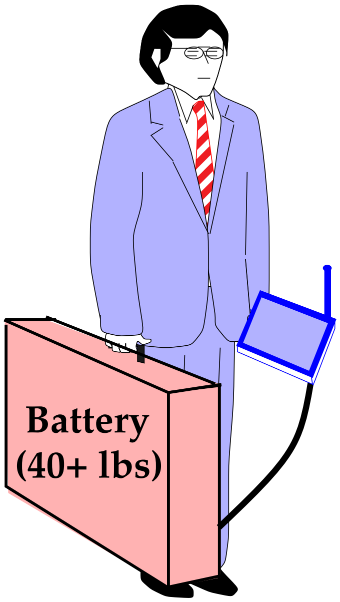
Low Power Design Problem





When supply voltage drops to 1Volt, then 100Watts = 100 Amps

Portable devices



Required Portable Functions

- Multimodal radio
- Protocols, ECC, ...
- Voice I/O compression & decompression
- Handwriting recognition
- Text/Graphics processing
- Video decompression
- Speech recognition
- Java interpreter

How to get 1 month of operation?

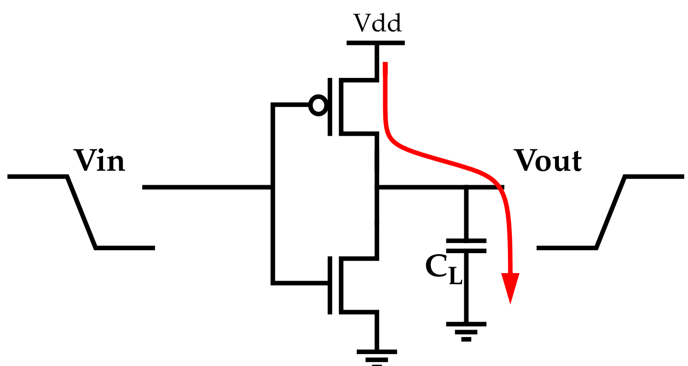
Two Kinds of Computation

- General purpose processing (what you have been studying so far)
  - Bursty - mostly idle with bursts of computation
  - Maximum possible throughput required during active periods
- Signal processing (for multimedia, wireless communications, etc.)
  - Stream based computation
  - No advantage in increasing processing rate above required for real-time requirements

Optimizing for Energy Consumption

- Conventional General Purpose processors (e.g. Pentiums)
  - Performance is everything ... somehow we'll get the power in and back out
  - 10-100 Watts, 100-1000 Mips = .01 Mips/mW
- Energy Optimized but General Purpose
  - Keep the generality, but reduce the energy as much as possible - e.g. StrongArm
  - .5 Watts, 160 Mips = .3 Mips/mW
- Energy Optimized and Dedicated
  - 100 Mops/mW

Switching Energy



Energy/transition = $C_L * V_{dd}^2$

Power = Energy/transition * f = $C_L * V_{dd}^2 * f$

Low Power & Low Energy System Design

System	Design partitioning, Power Down
Algorithm	Complexity, Concurrency, Locality, Regularity, Data representation
Architecture	Voltage scaling, Parallelism, Instruction set, Signal correlations
Circuit/Logic	Transistor Sizing, Logic optimization, Activity Driven Power Down, Low-swing logic, Adiabatic switching
Technology	Threshold Reduction, Multi-thresholds

Energy Reduction in CPU's

- Standard power management helps
  - Sleep modes
  - Power down blocks
- Clock rate reduction doesn't help
  - Number of operations = N_{ops}
  - Energy/operation = CV^2
  - Total Energy = $N_{ops} * CV^2$

Energy is independent of clock rate!

- Reducing the clock rate only degrades throughput, but no savings in battery life - unless the voltage is changed

Node Transition Activity and Power

Switch a CMOS gate for N clock cycles

$$E_N = C_L \bullet V_{dd}^2 \bullet n(N)$$

E_N : the energy consumed for N clock cycles

$n(N)$: the number of 0->1 transition in N clock cycles

$$P_{avg} = \lim_{N \to \infty} \frac{E_N}{N} \bullet f_{clk} = \left(\lim_{N \to \infty} \frac{n(N)}{N}\right) \bullet C_L \bullet V_{dd}^2 \bullet f_{clk}$$

$$\alpha_{0 \to 1} = \lim_{N \to \infty} \frac{n(N)}{N}$$

$$\boxed{P_{avg} = \alpha_{0 \to 1} \bullet C_L \bullet V_{dd}^2 \bullet f_{clk}}$$

Factors Affecting Transition Activity, $\alpha_{0\text{->}1}$

- “Static” component (does not account for timing)
  - ➟ Type of Logic Function (NOR vs. XOR)
  - ➟ Type of Logic Style (Static vs. Dynamic)
  - ➟ Signal Statistics
  - ➟ Inter-signal Correlations
- “Dynamic” or timing dependent component
  - ➟ Circuit Topology
  - ➟ Signal Statistics and Correlations

Static 2 Input NOR Gate

Assume:

***prob*(A=1) = 1/2**

***prob*(B=1) = 1/2**

Then:

***prob*(Out=1) = 1/2**

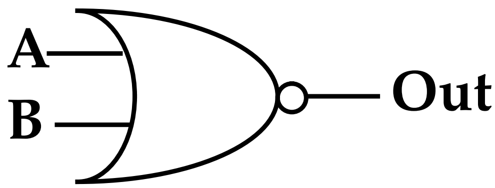
***prob*(0→1)**

= *prob*(Out=0).*prob*(Out=1)

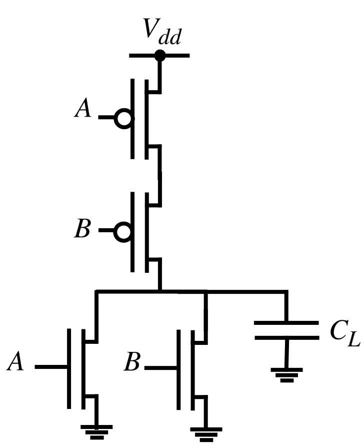
= 1/2 × 1/2 = 1/4

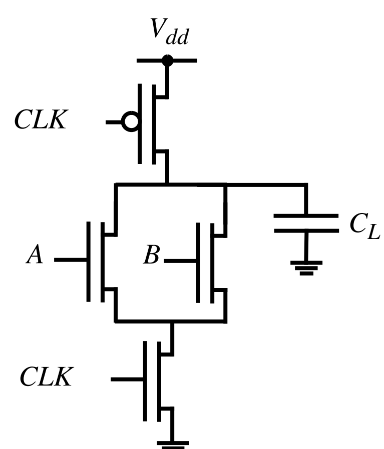
A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

$$\boxed{\alpha_{0\text{->}1} = 1/4}$$



Type of Logic Style: Static vs. Dynamic





Power is dissipated when Out=0

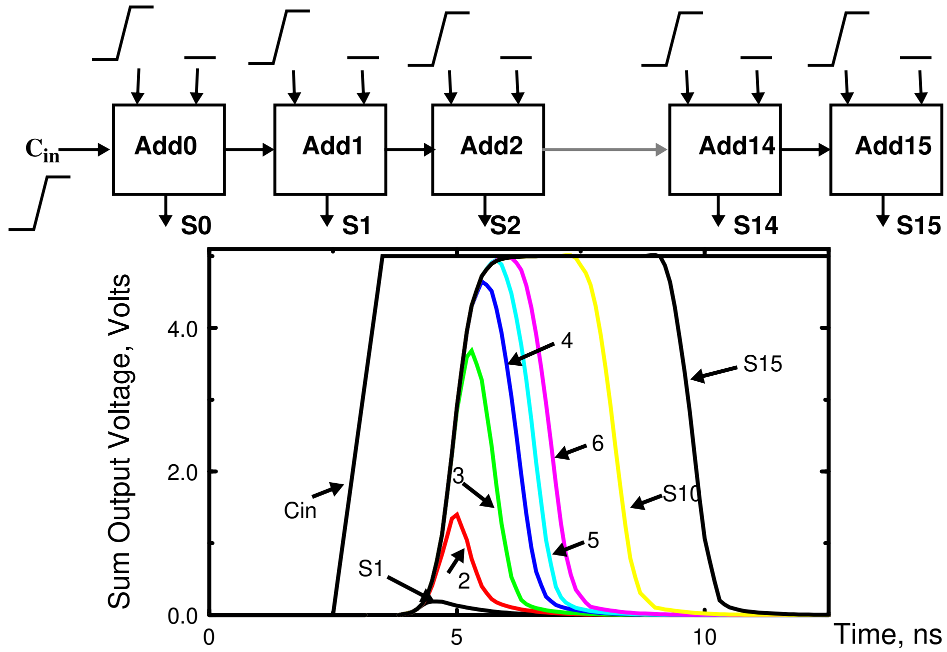
STATIC NOR

$$\boxed{\alpha_{0\text{->}1} = 3/16}$$

DYNAMIC NOR

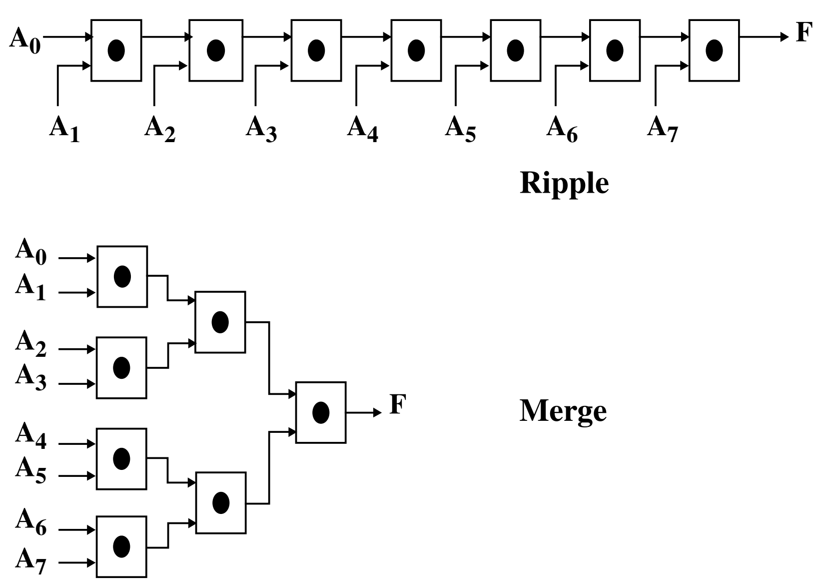
$$\boxed{\alpha_{0 \to 1} = \frac{N_0}{2^N} = \frac{3}{4}}$$

"Dynamic" or Glitching Activity in CMOS

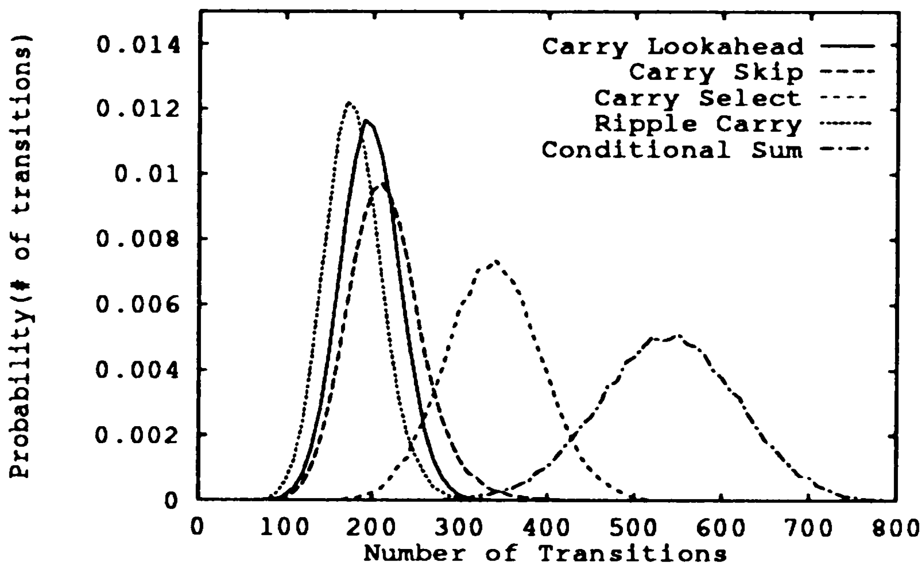


$\alpha_{0\text{->}1}$ **can be > 1 due to glitching!**

Glitch Reduction Using Balanced Paths



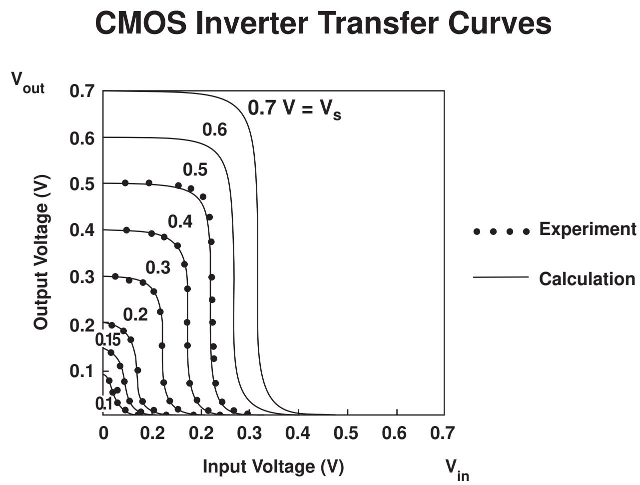
Comparison of Adder Topologies



Switching activity and capacitance minimization

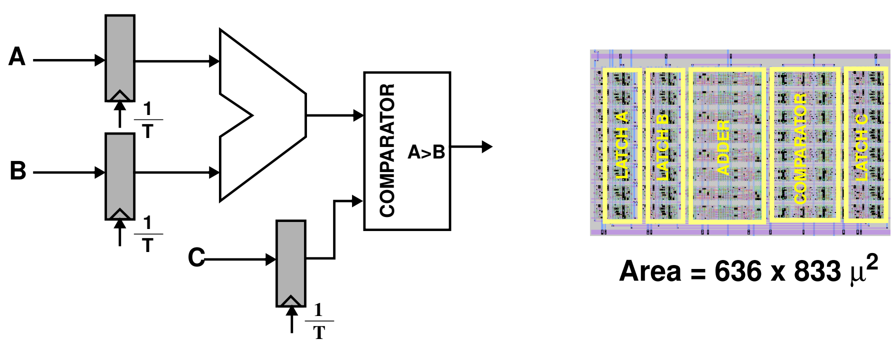
- Gated clocks. (*disable all modules not in use each cycle*)
- Block enables. (*enable only those modules using a bus*)
- Instruction Buffer. (*0th level cache*)
- Add stop and sleep instructions to the instruction set.
- Minimum size busses
- Minimize I/O - on-chip memory

Minimum Supply Voltage



$V_{dd\text{-}min} \geq$ 50-100mV above V_t

Architecture Trade-offs - Reference Datapath

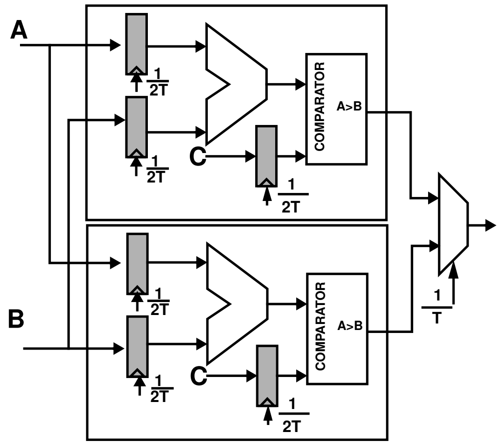


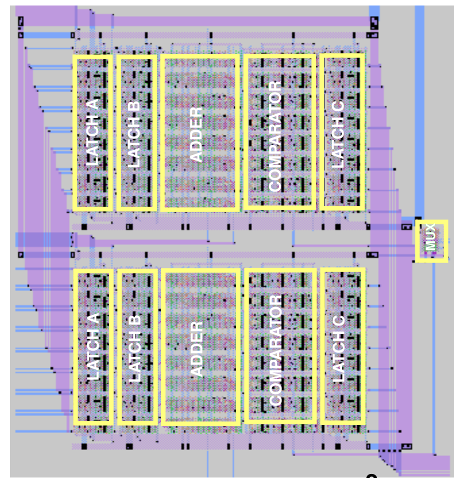
Area = 636 x 833 μ^2

- Critical path delay $\Rightarrow T_{adder} + T_{comparator}$ (= 25ns)
  $\Rightarrow f_{ref}$ = 40Mhz
- Total capacitance being switched = C_{ref}
- $V_{dd} = V_{ref}$ = 5V
- Power for reference datapath = $P_{ref} = C_{ref} V_{ref}^2 f_{ref}$

from [Chandrakasan92] (*IEEE JSSC*)

Parallel Datapath

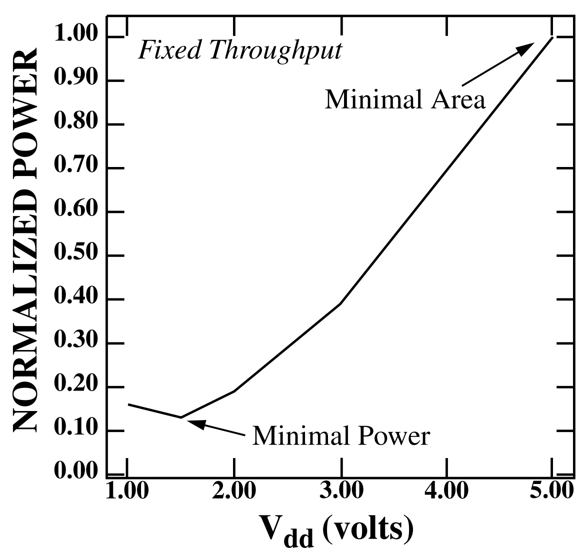




Area = 1476 x 1219 μ^2

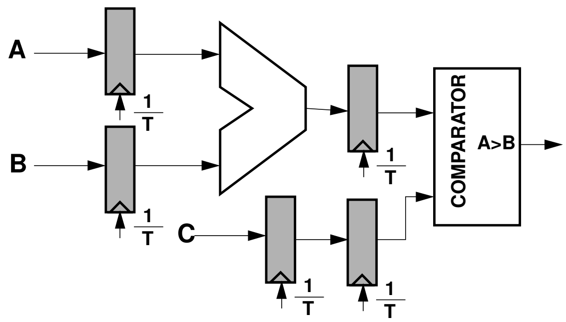
- The clock rate can be reduced by half with the same throughput $\Rightarrow f_{par} = f_{ref} / 2$
- $V_{par} = V_{ref} / 1.7$, $C_{par} = 2.15C_{ref}$
- $P_{par} = (2.15C_{ref})\,(V_{ref}/1.7)^2\,(f_{ref}/2) \approx 0.36\,P_{ref}$

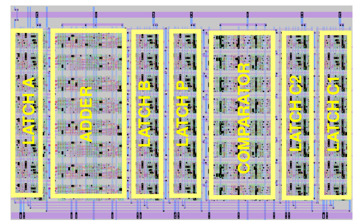
The More Parallel the Better??



Capacitance overhead starts to dominate at “high” levels of parallelism and results in an optimum voltage

Pipelined Datapath





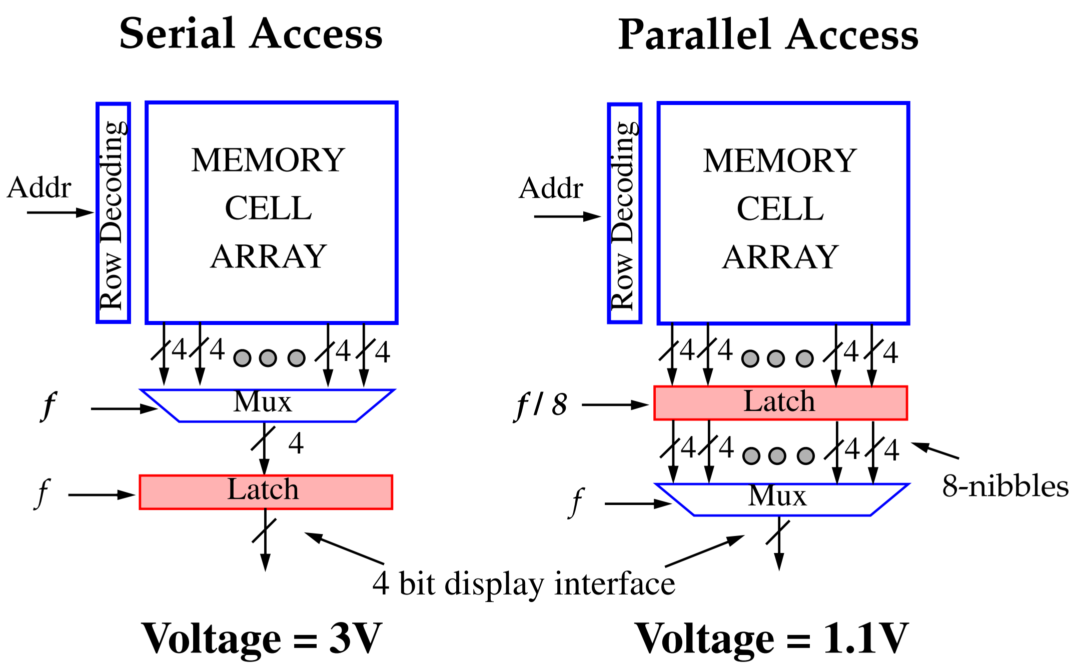
Area = 640 x 1081 μ^2

- **Critical path delay is less $\Rightarrow$ max [T_{adder}, $T_{comparator}$]**
- **Keeping clock rate constant: $f_{pipe} = f_{ref}$**
  Voltage can be dropped $\Rightarrow$ $V_{pipe} = V_{ref} / 1.7$
- **Capacitance slightly higher: $C_{pipe} = 1.15C_{ref}$**
- **$P_{pipe} = (1.15C_{ref})\,(V_{ref}/1.7)^2 f_{ref} \approx 0.39\, P_{ref}$**

Architecture Summary for a Simple

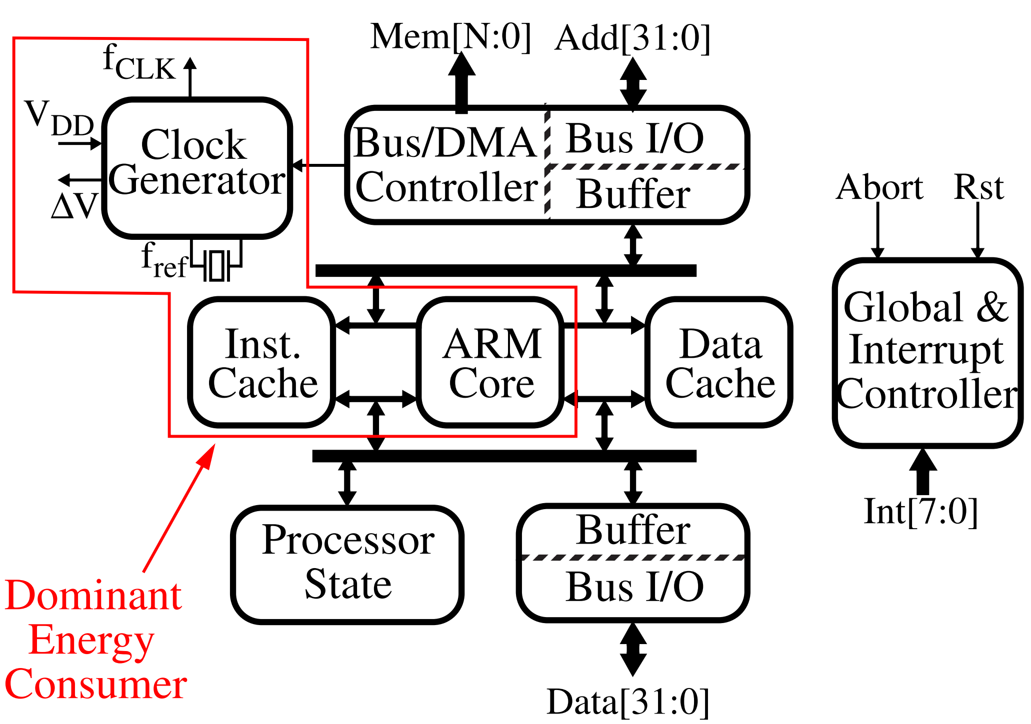
Architecture type	Voltage	Area	Power
Simple datapath (no pipelining or parallelism)	5V	1	**1**
Pipelined datapath	2.9V	1.3	0.39
Parallel datapath	2.9V	3.4	0.36
Pipeline-Parallel	2.0V	3.7	**0.2**

Memory Architecture



Slide 23

Proposed CPU Architecture: LP-ARM



Slide 24

LP-ARM: Energy Estimation

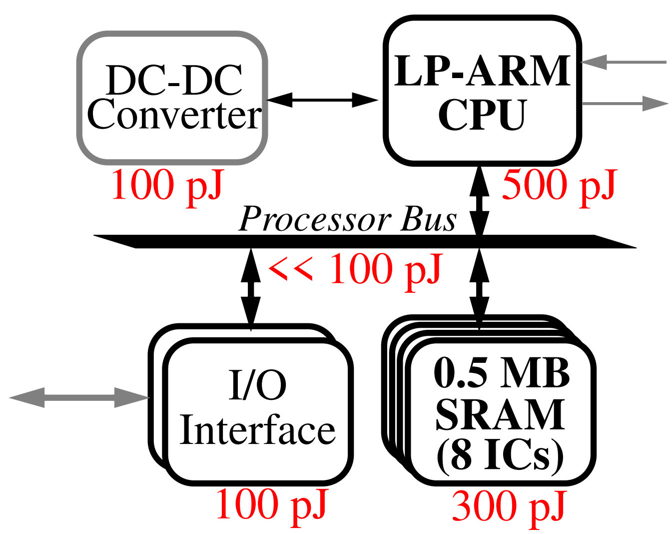
Instruction Cache (8kB):
Low-Power SRAM: 2 kByte Block = 78 pJ [Burstein]
Complete Instruction Cache Design: ~150 pJ

Clock Generation
External 50pF line = 70 pJ
Total Clock Generation: ~100 pJ

ARM Core
Register File + ALU + Shifter > 50% Total [ARM,Burd]
Register File: 30 pJ, ALU: 24 pJ, Shifter: 16 pJ
Total Core: ~140 pJ

Research Goal

10 MIPS, 1 nJ/inst. ⇔ 80 MIPS, 9 nJ/inst.
(10 mW) *(720 mW)*



Improve energy efficiency by an order of magnitude