Low Power Digital VLSI Design

Dr. Shubhajit Roy Chowdhury,

Centre for VLSI and Embedded Systems Technology,
IIIT Hyderabad, India

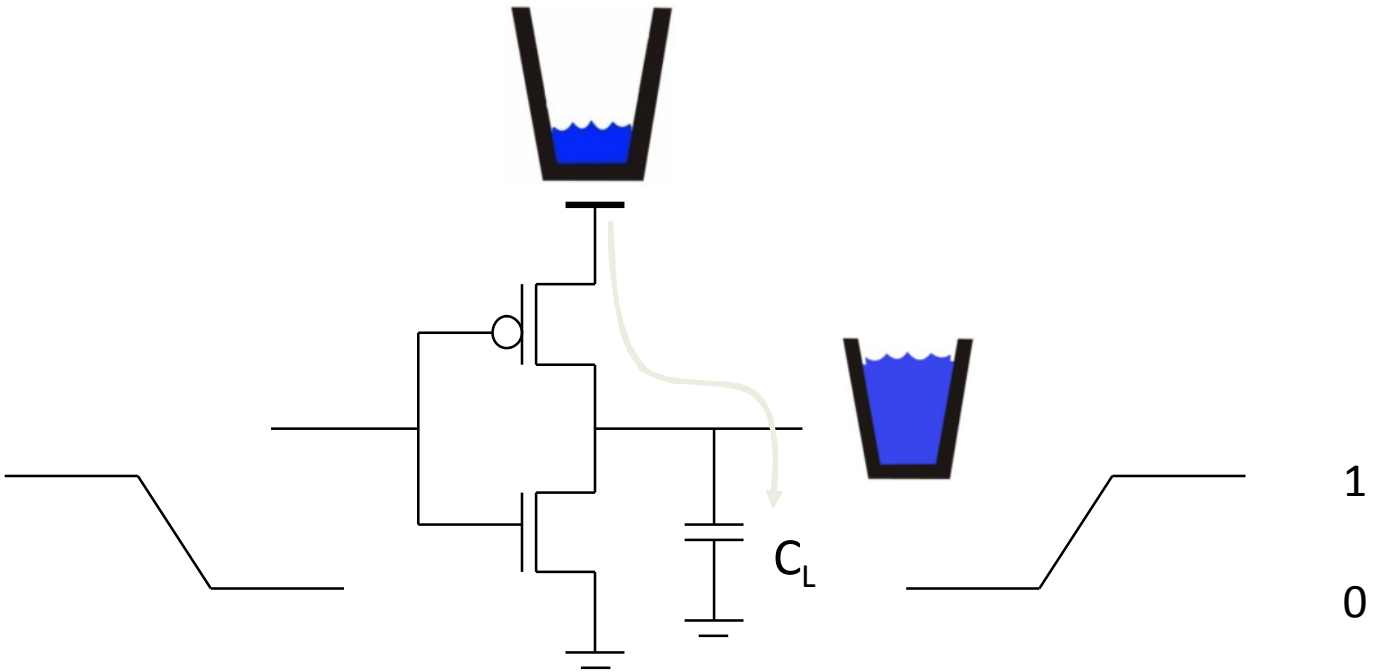
Email: src.vlsi@iiit.ac.in

Levels of consideration

- Power reduction on
  - Gate level
  - Architecture level
  - Algorithm level
  - System level

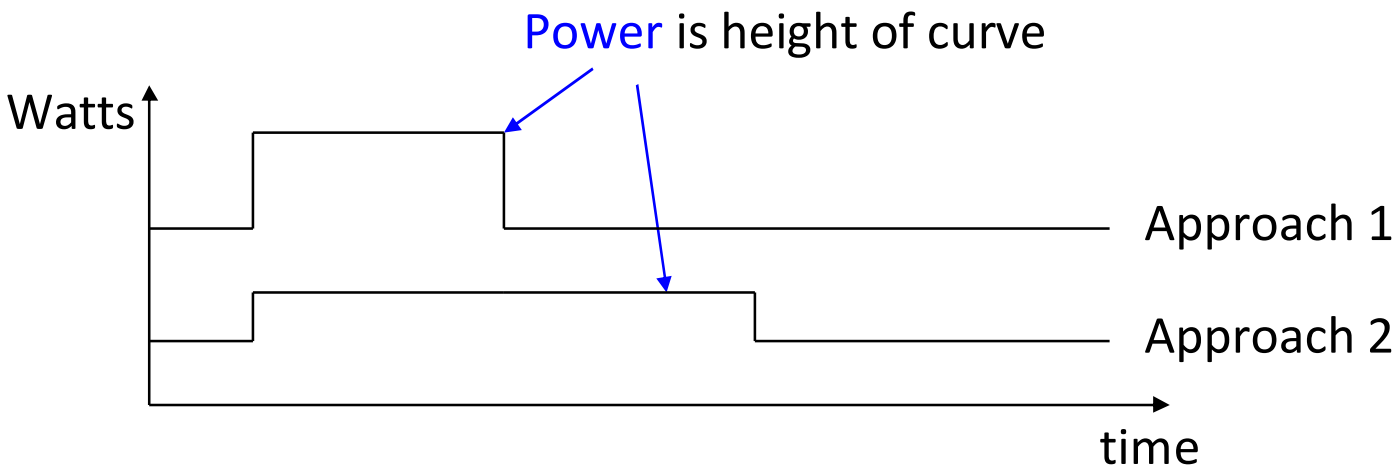
Power Consumption in CMOS

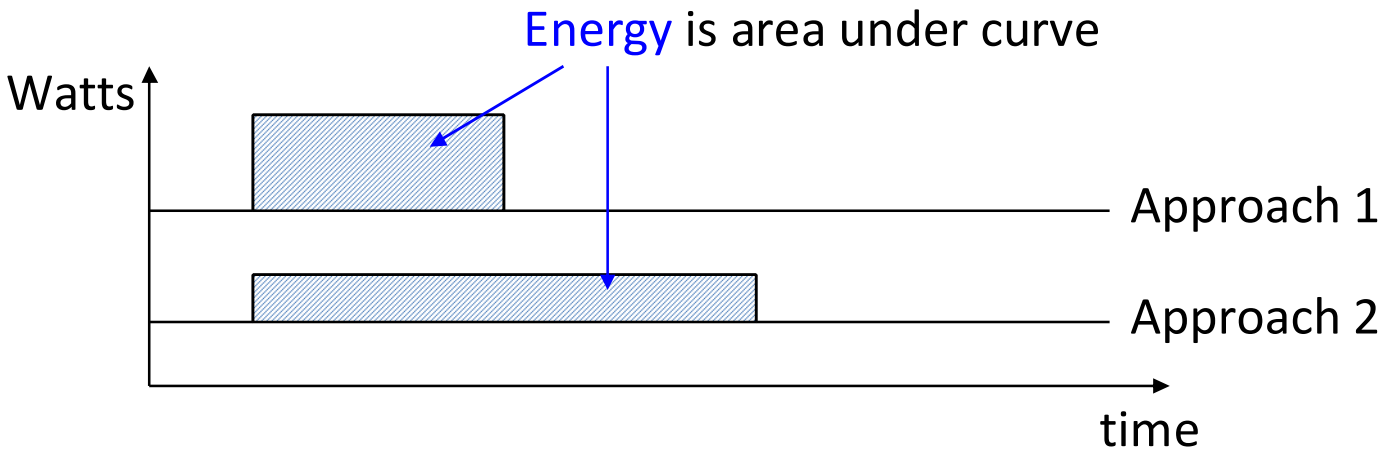
- Voltage (Volt, V) ⟷ Water pressure (bar)
- Current (Ampere, A) ⟷ Water quantity per second (liter/s)
- Energy ⟷ Amount of Water



Energy consumption is proportional to capacitive load!

Energy and Power





Energy = Power * time for calculation = Power * Delay

Power Equations in CMOS

$$P = \alpha f C_L V_{DD}^2 + V_{DD} I_{peak} (P_{0\rightarrow1} + P_{1\rightarrow0}) + V_{DD} I_{leak}$$

Dynamic power (≈ 40 - 70% today and decreasing relatively)

Short-circuit power (≈ 10 % today and decreasing absolutely)

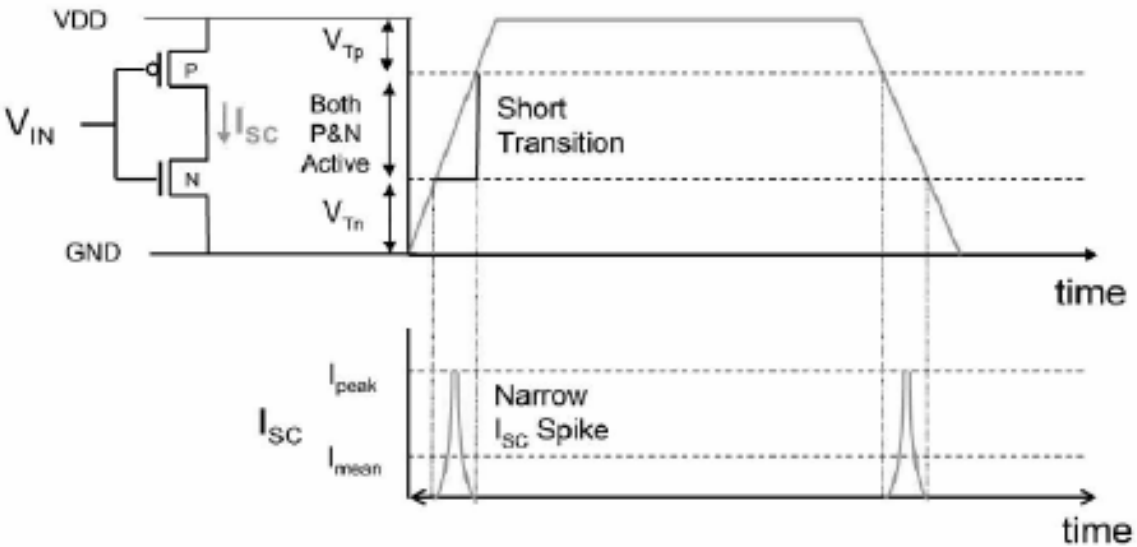
Leakage power (≈ 20 – 50 % today and increasing)

Dynamic Power Dissipation

- $P_{s/w} = \alpha * C_L * V_{dd}^2 * f_{clk}$
  - C_L physical capacitance, V_{dd} supply voltage, α switching activity, f_{clk} clock frequency.
  - $C_L(i) = \Sigma_j C_{IN}^j + C_{wire} + C_{par(i)}$
    - C_{IN} the gate input capacitance, C_{wire} the parasitic interconnect and C_{par} diffusion capacitances of each gate[I].
- Depends on:
  - Supply voltage
  - Physical Capacitance
  - Switching activity

Short Circuit Power Dissipation

- Caused by simultaneous conduction of n and p blocks.



Short Circuit Power Dissipation (Cont'd)

$$P_{sc} = I_{sc} \cdot V_{DD} = \frac{1}{12} \cdot k \cdot \tau \cdot (V_{DD} - 2V_T)^3 \cdot f$$

where $k = (k_n = k_p)$, the trans conductance of the transistor, $\tau = (t_{rise} = t_{fall})$, the input/output transition time, V_{DD} = supply voltage, f = clock frequency, and $V_T = (V_{Tn} = |V_{Tp}|)$, the threshold voltage of MOSFET.

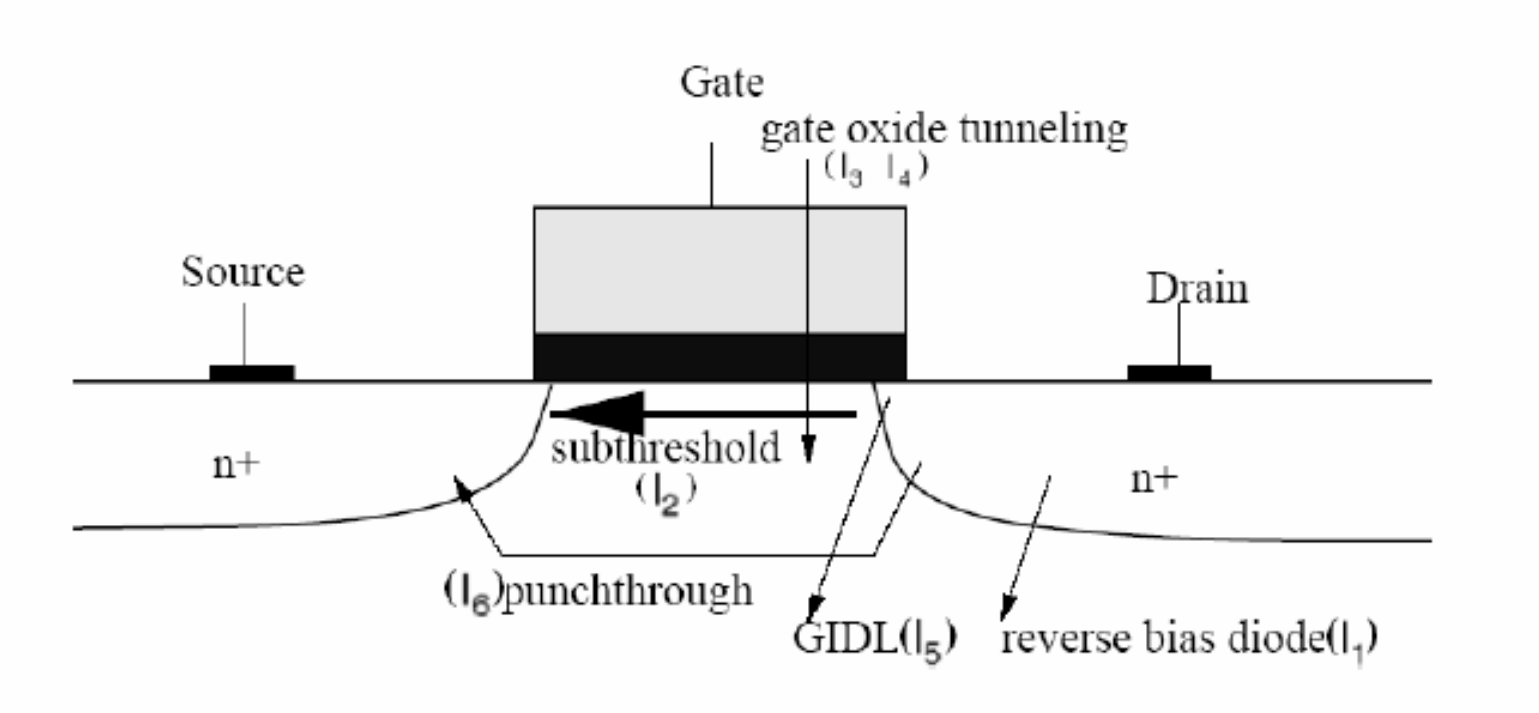
- Depends on :
  - The input ramp
  - Load
  - The transistor size of the gate
  - Supply voltage
  - Frequency
  - Threshold voltage.

Leakage Power Dissipation

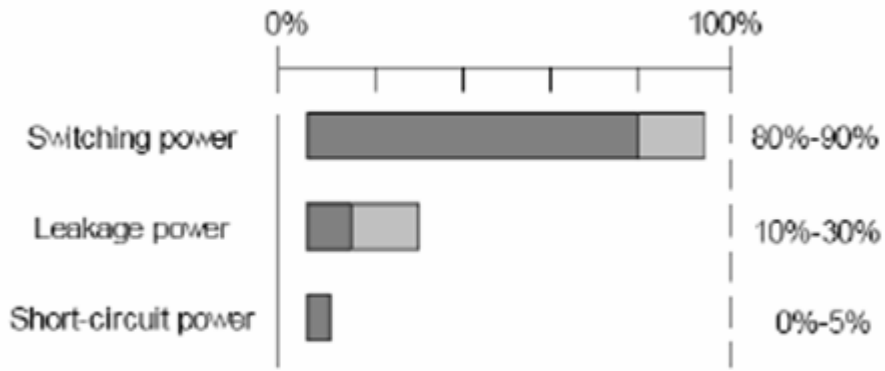
- Six short-channel leakage mechanisms are there:
  - I_1 Reverse-bias p-n junction leakage
  - I_2 Sub threshold leakage
  - I_3 Oxide tunneling current
  - I_4 Gate current due to hot-carrier injection
  - I_5 GIDL (Gate Induced Drain Leakage)
  - I_6 Channel punch through current

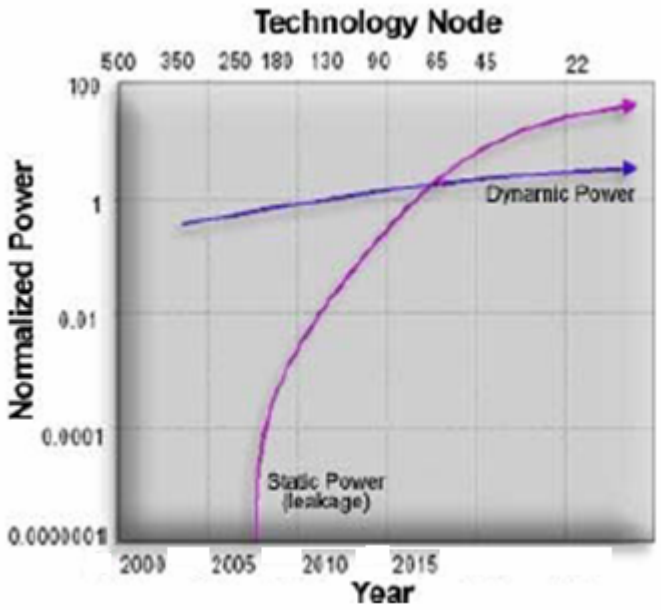
- I_1 and I_2 are the dominant leakage mechanisms

Leakage Power Dissipation (Cont'd)



Comparison of different Power Dissipations





Courtesy: Intel Corporation

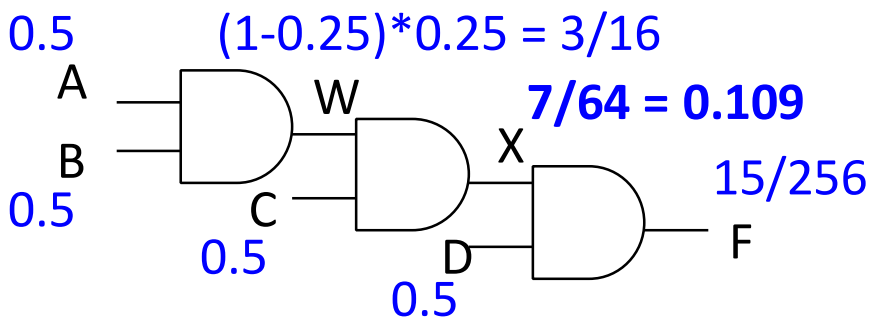
Levels of Optimization

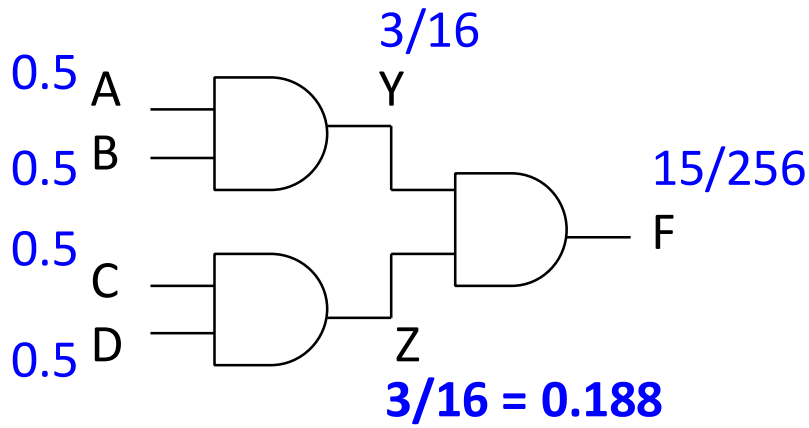
	Savings	Speed	Error
System	> 70 %	Seconds	> 50 %
Algorithm	40-70 %	Minute	25-50 %
Architecture	25-40 %	Minutes	15-30 %
Gate	15-25 %	Hour	10-20 %
Transistor	10-15 %	Hours	5-10 %

Logic Restructuring

- Logic restructuring: changing the topology of a logic network to reduce transitions

AND: $P_{0\to1} = P_0 * P_1 = (1 - P_A P_B) * P_A P_B$



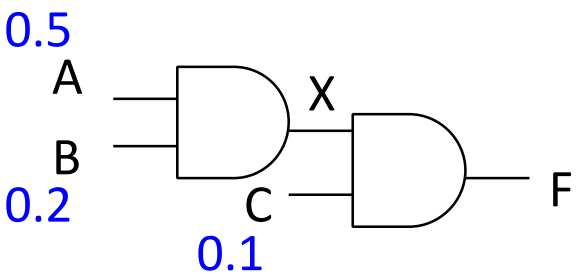


➔ Chain implementation has a lower overall switching activity than tree implementation for random inputs

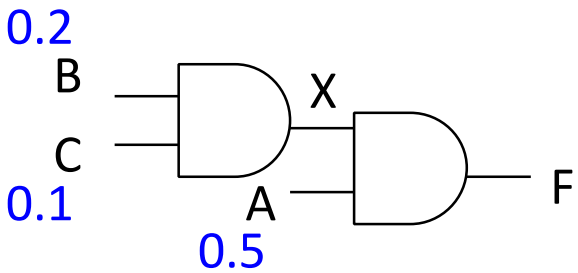
- BUT: Ignores glitching effects

Input Ordering

(1-0.5x0.2)*(0.5x0.2)=**0.09**



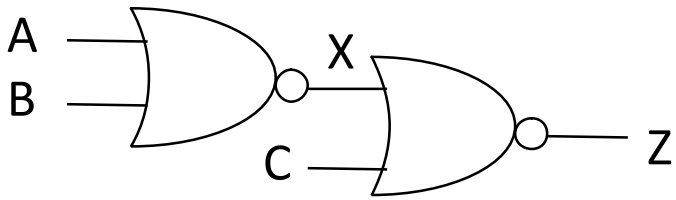
(1-0.2x0.1)*(0.2x0.1)=**0.0196**

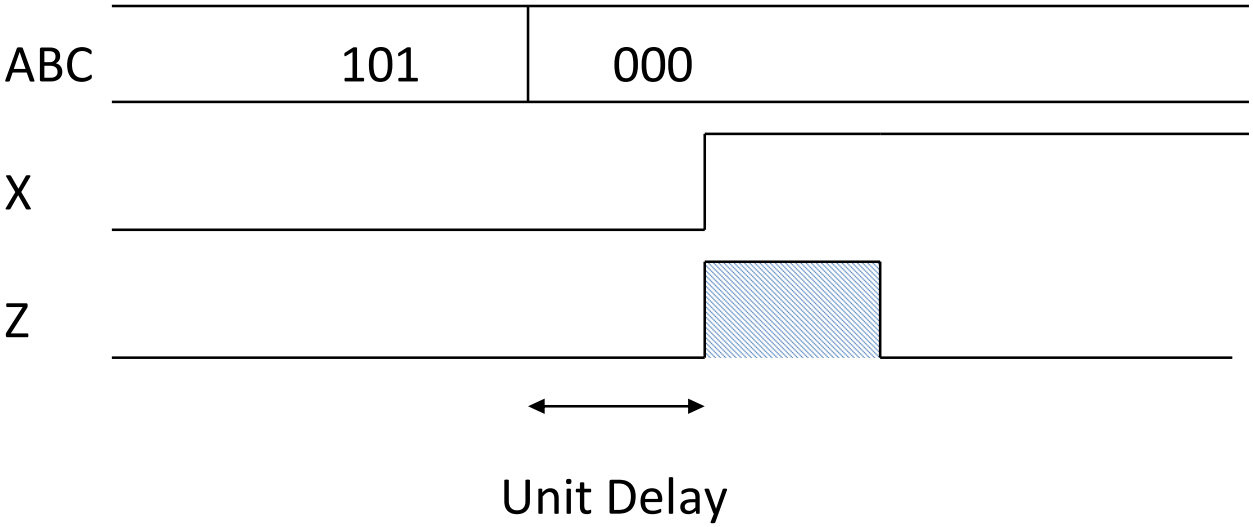


AND: $P_{0\rightarrow1} = (1 - P_A P_B) * P_A P_B$

Beneficial: postponing introduction of signals with a high transition rate (signals with signal probability close to 0.5)

Glitching



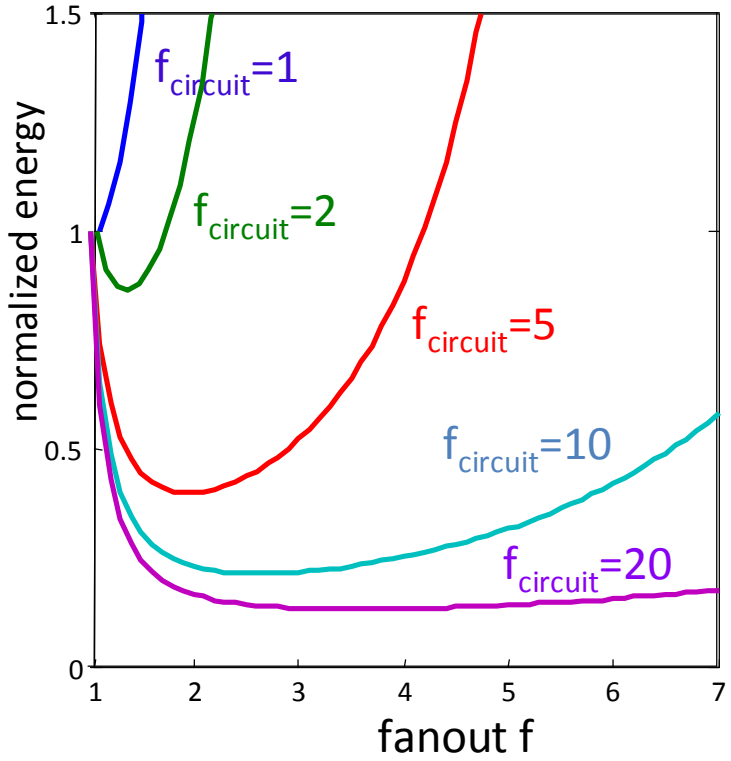


Dynamic Power and Device Size

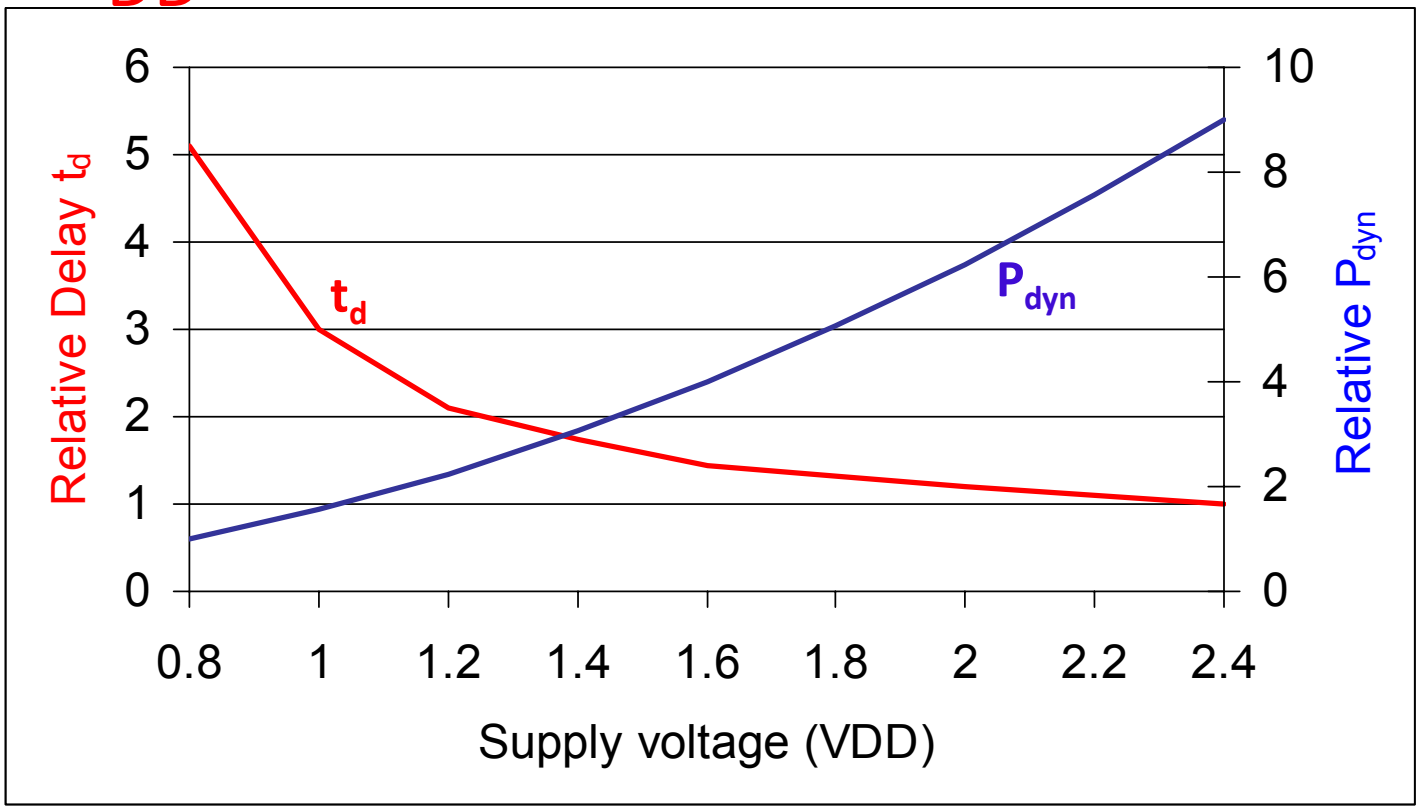
- Device Sizing (= changing gate width)
  - ➔ Affects input capacitance C_{in}
  - ➔ Affects load capacitance C_{load}
  - ➔ Affects dynamic power consumption P_{dyn}

  e.g., for C_{load}=20, C_{in}=1

  ➔ $f_{circuit}$ = 20
- For Low Power: avoid oversizing (f too big) beyond the optimal



V_{DD} versus Delay and Power



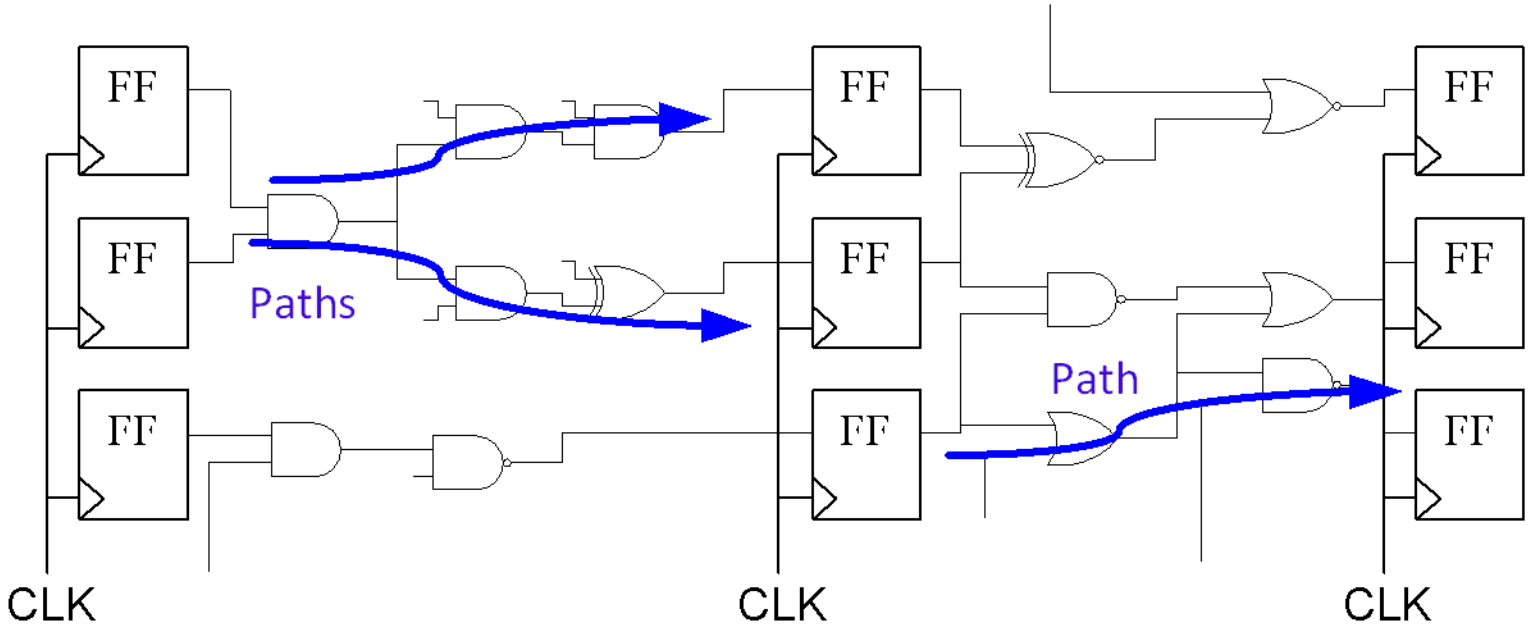
- Delay (t_d) and dynamic power consumption (P_{dyn}) are functions of V_{DD}

Multiple V_{DD}

- Main ideas:
  - Use of different supply voltages within the same design
  - High VDD for critical parts (high performance needed)
  - Low VDD for non-critical parts (only low performance demands)
- At design phase:
  - Determine critical path(s) (see upper next slide)
  - High V_{DD} for gates on those paths
  - Lower V_{DD} on the other gates (in non-critical paths)
  - For low VDD: prefer gates that drive large capacitances (yields the largest energy benefits)
- Usually two different V_{DD} (but more are possible)

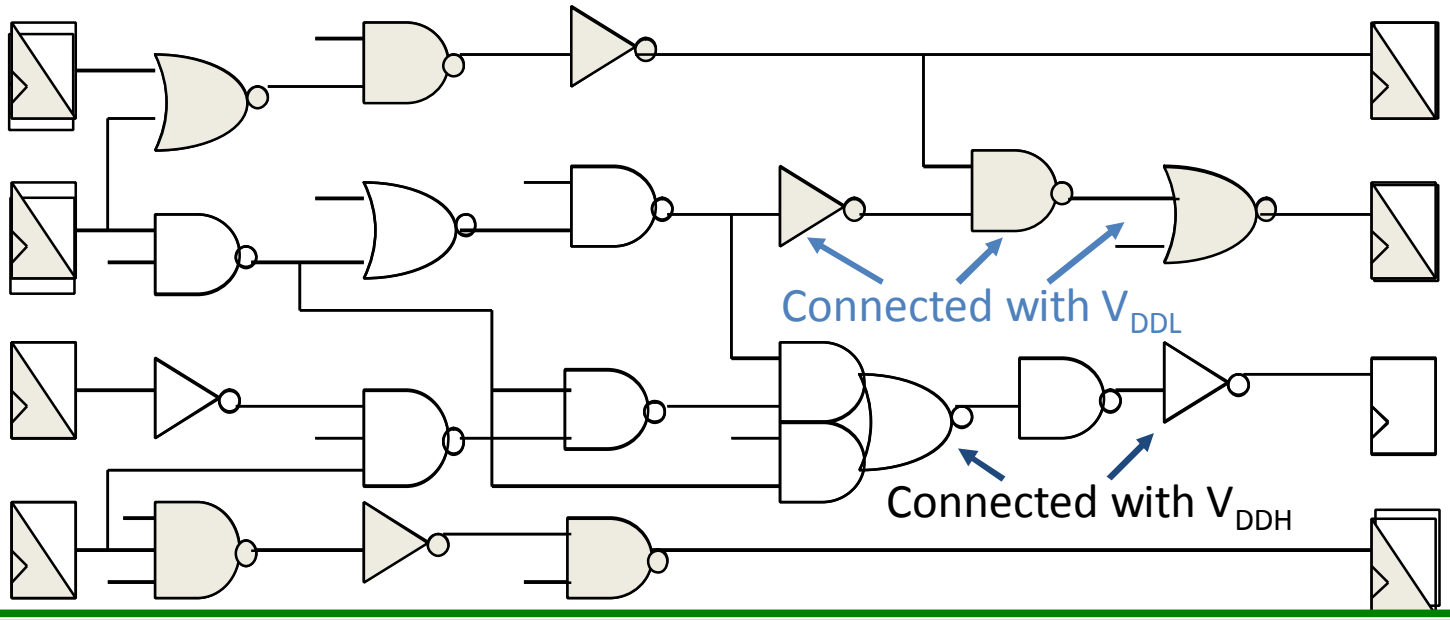
Data Paths

- Data propagate through different data paths between registers (flipflops - FF)
- Paths mostly differ in propagation delay times
- Frequency of clock signal (CLK) depends on path with longest delay ➔ critical path



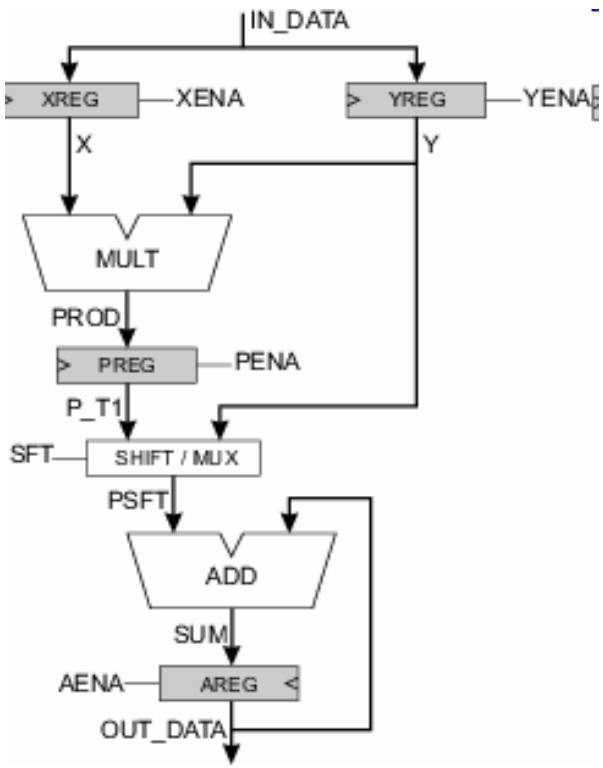
Multiple V_{DD} in Data Paths

- Minimum energy consumption when all logic paths are critical (same delay)
- Possible Algorithm: clustered voltage-scaling
  - Each path starts with V_{DDH} and switches to V_{DDL} (grey gates) when slack is available
  - Level conversion in flipflops at end of paths



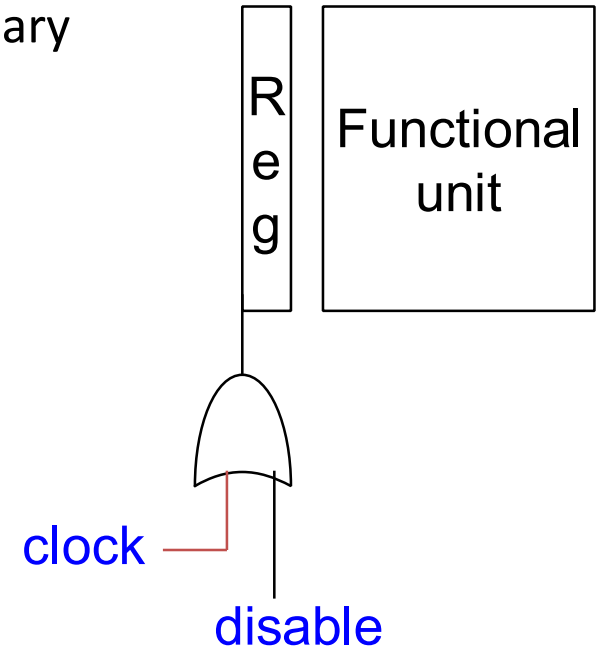
Design Layer: Architecture Level

- Also known as Register transfer level (RTL)
- Base elements:
  - Register structures
  - Arithmetic logic units (ALU)
  - Memory elements
- Only behavior is described (no inner structure)

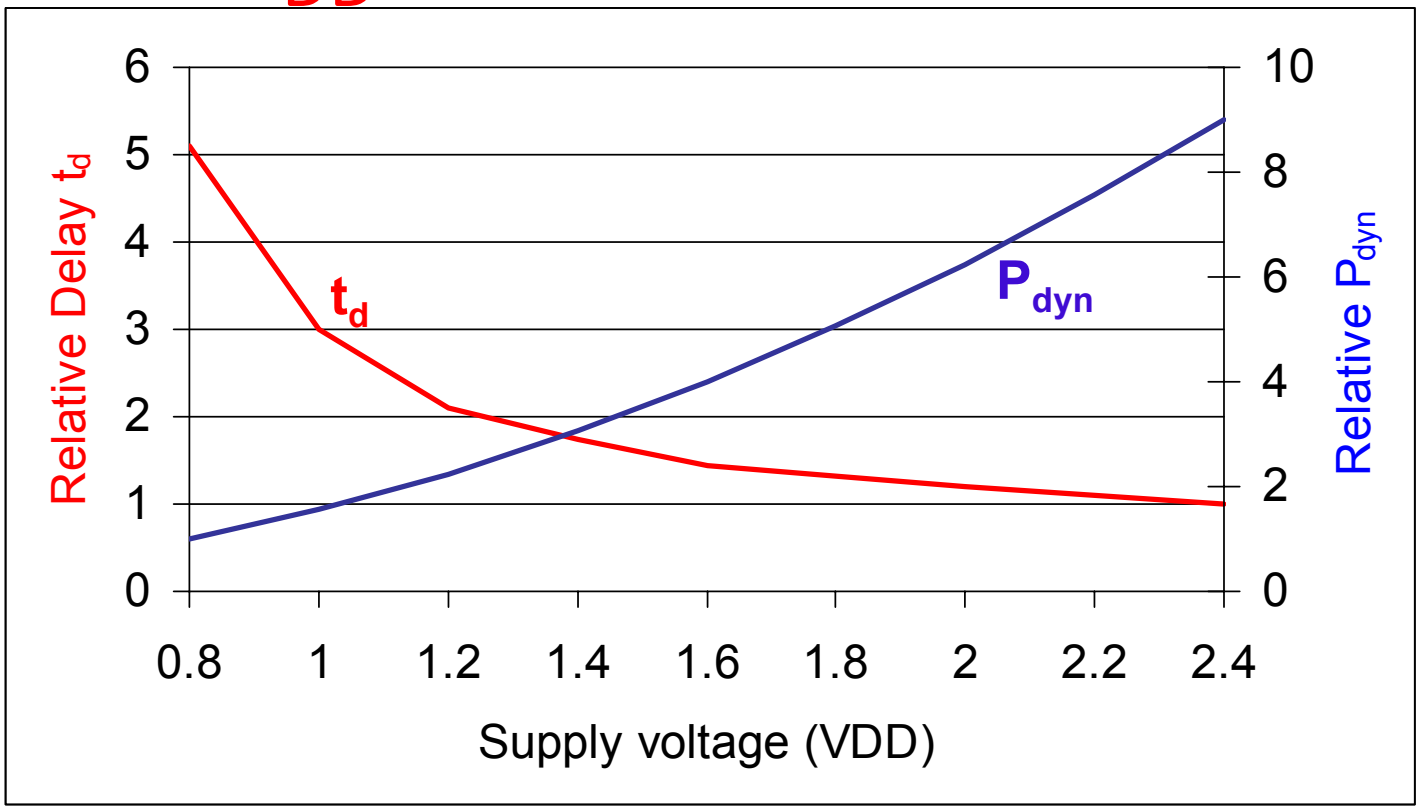


Clock Gating

- Most popular method for power reduction of clock signals and functional units
- Gate off clock to idle functional units
- Logic for generation of disable signal necessary
  - Higher complexity of control logic
  - Higher power consumption
  - Critical timing critical for avoiding of clock glitches at OR gate output
  - Additional gate delay on clock signal



Recap: V_{DD} versus Delay and Power



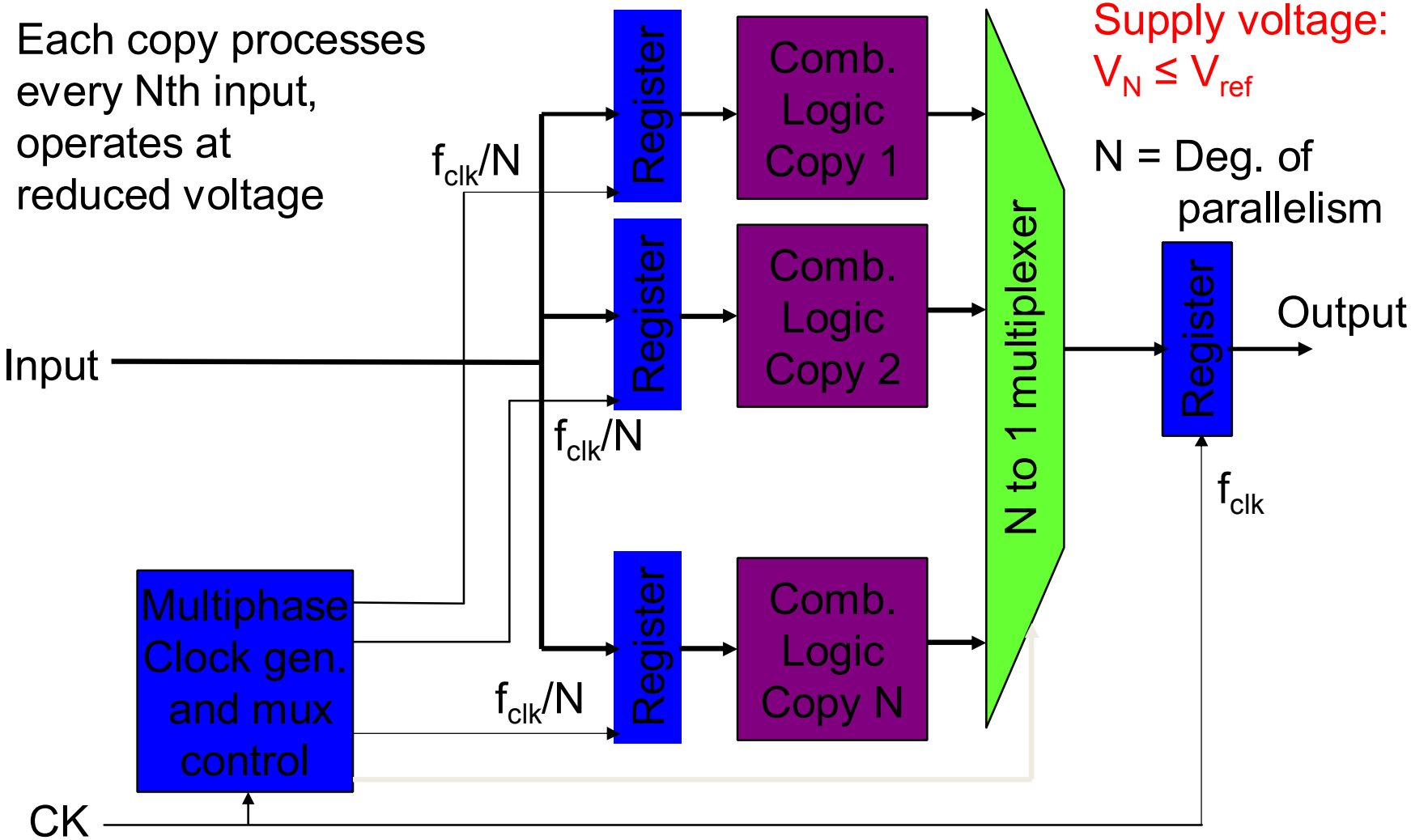
Dynamic Power can be traded by delay

Parallel Architecture

Each copy processes every Nth input, operates at reduced voltage

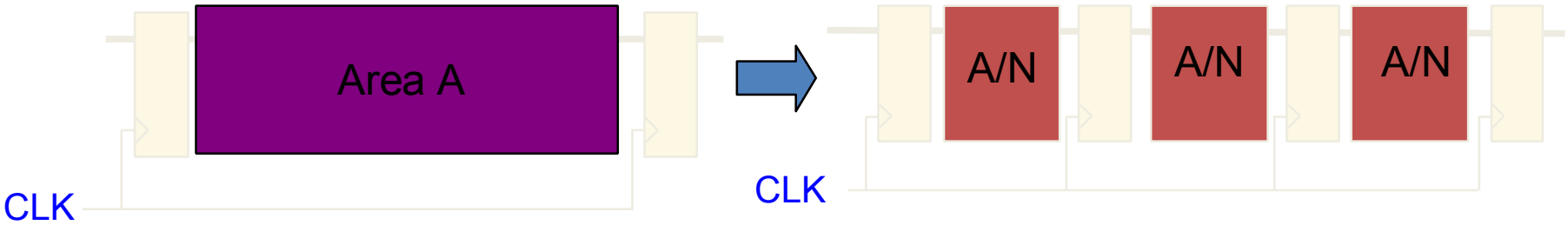
Supply voltage: $V_N \leq V_{ref}$

N = Deg. of parallelism

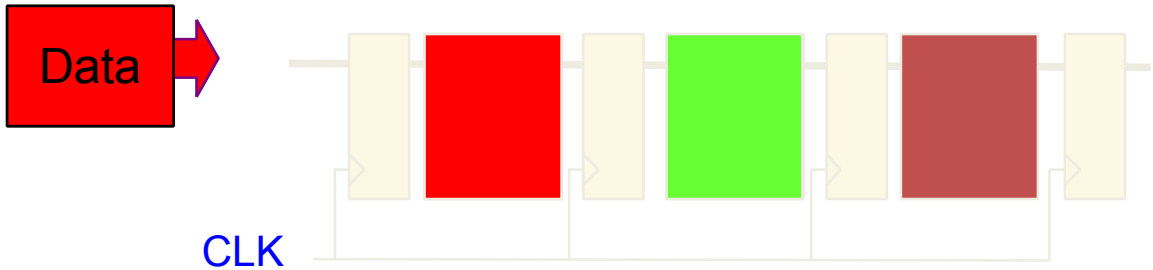


Pipelined Architecture

- Reduces the propagation time of a block by factor N
  ➔ Voltage can be reduced at constant clock frequency
- Constant throughput

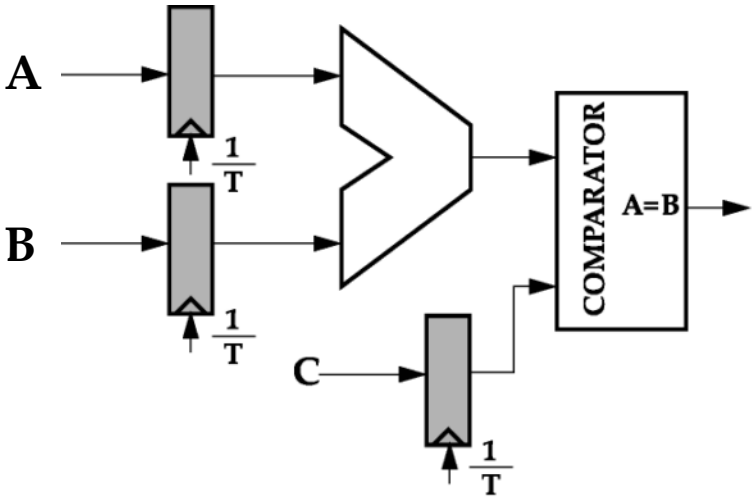


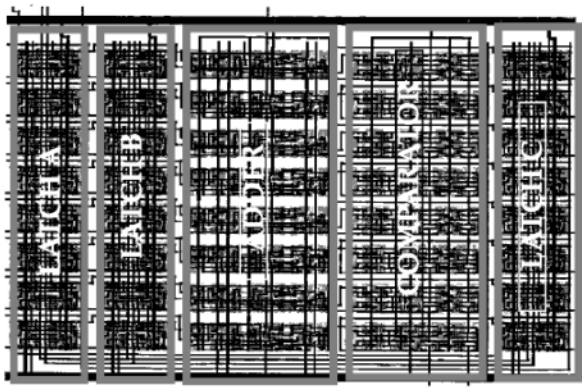
- Functionality:



Parallel Architecture: Example

- Reference Data path (for example)

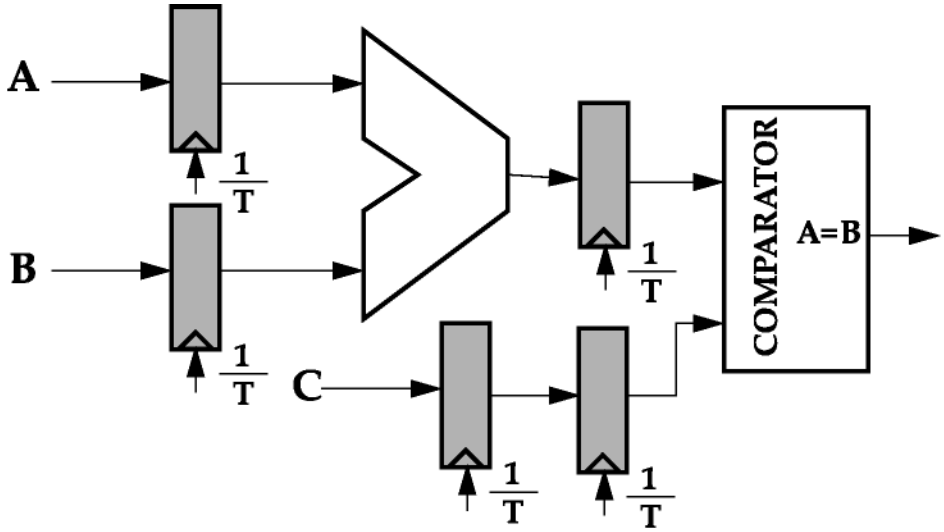


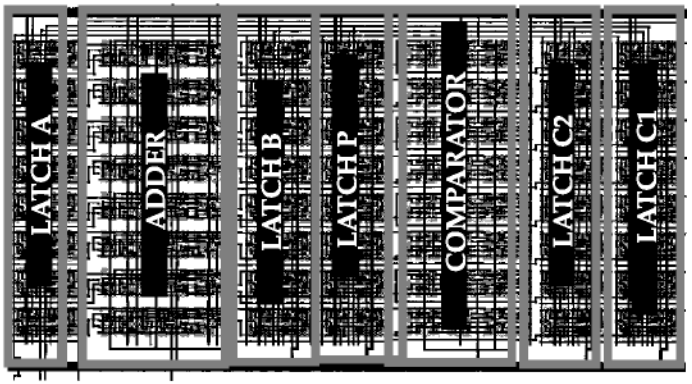


Area = 636 x 833 u^2

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

Pipelined Architecture: Example





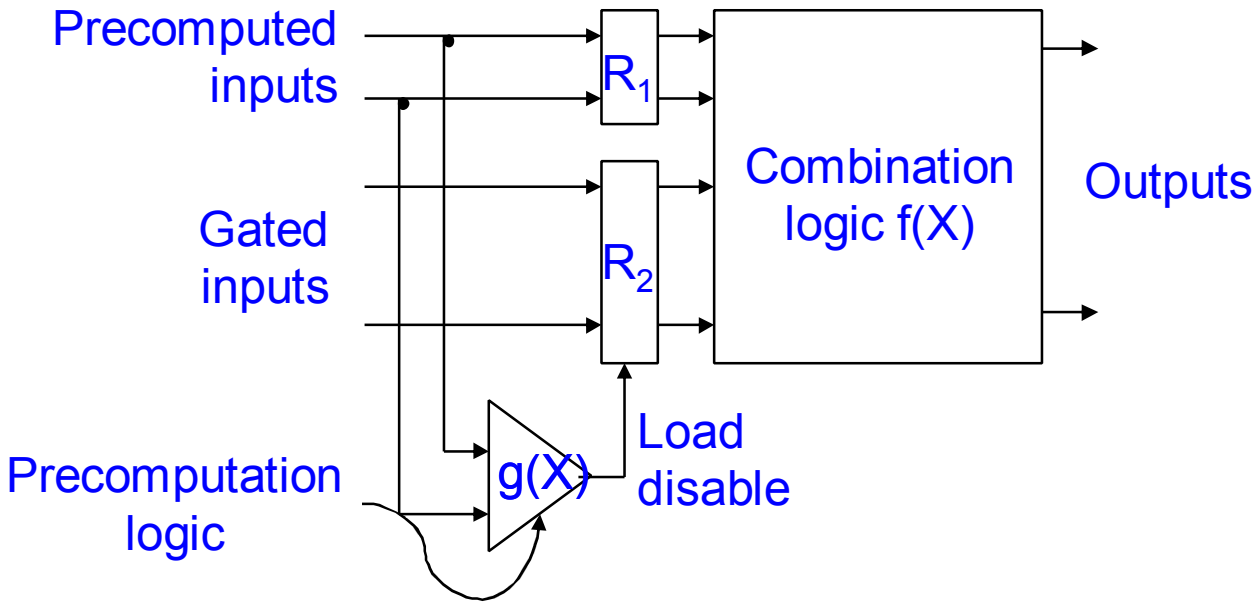
Area = 640 x 1081 u^2

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

Approximate Trend

	N-parallel proc.	N-stage pipeline proc.
Capacitance	$N*C_{ref}$	C_{ref}
Voltage	V_{ref}/N	V_{ref}/N
Frequency	f_{ref}/N	f_{ref}
Dynamic Power	$C_{ref}V_{ref}^2f_{ref}/N^2$	$C_{ref}V_{ref}^2f_{ref}/N^2$
Chip area	N times	10-20% increase

Precomputation



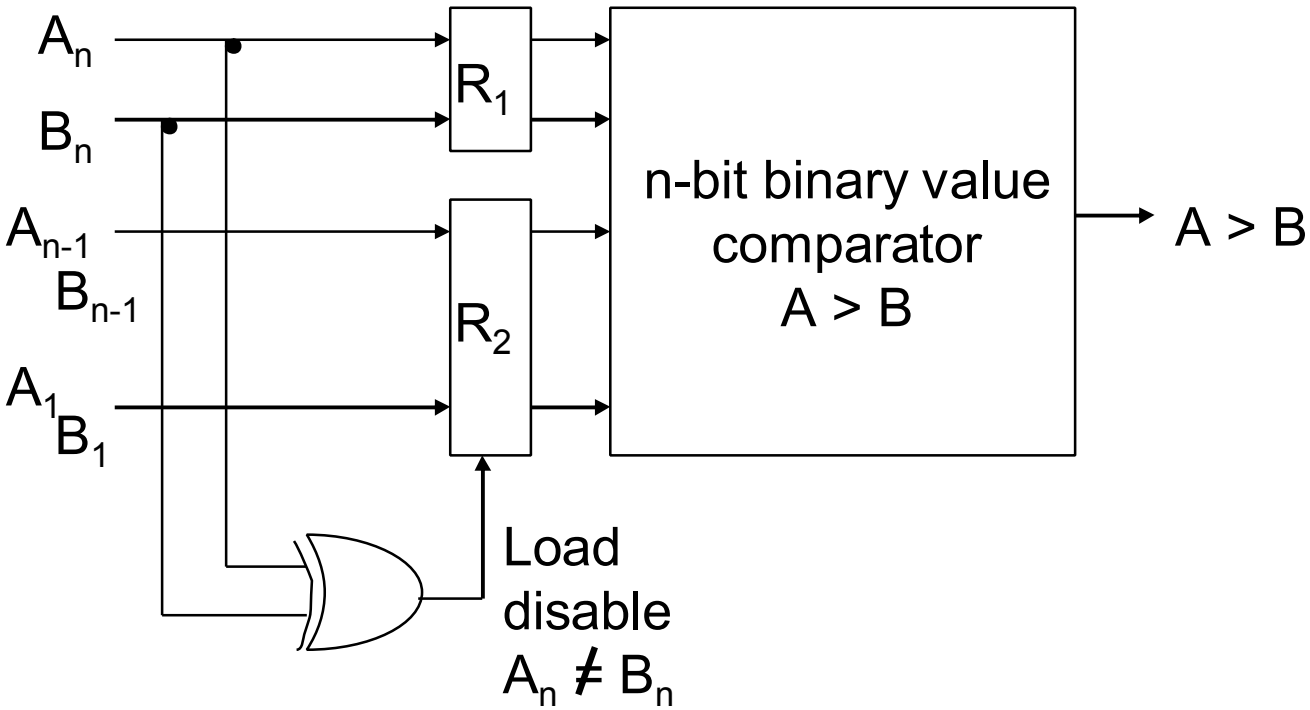
- Identify logical conditions at inputs that are invariant to the output
  - Since those inputs don't affect output, disable input transitions
  - Trade area for energy

Precomputation: Design Issues

- Design steps
  1. Selection of precomputation architecture
  2. Determination of precomputed and gated inputs (Register R1 should be much smaller than R2)
  3. Search good implementation for g(X)
  4. Evaluation of potential energy savings based on input statistics (if savings not sufficient go to step 2 or 3 and try again)
- Also works for multiple output functions where g(X) is the product of gj(X) over all j

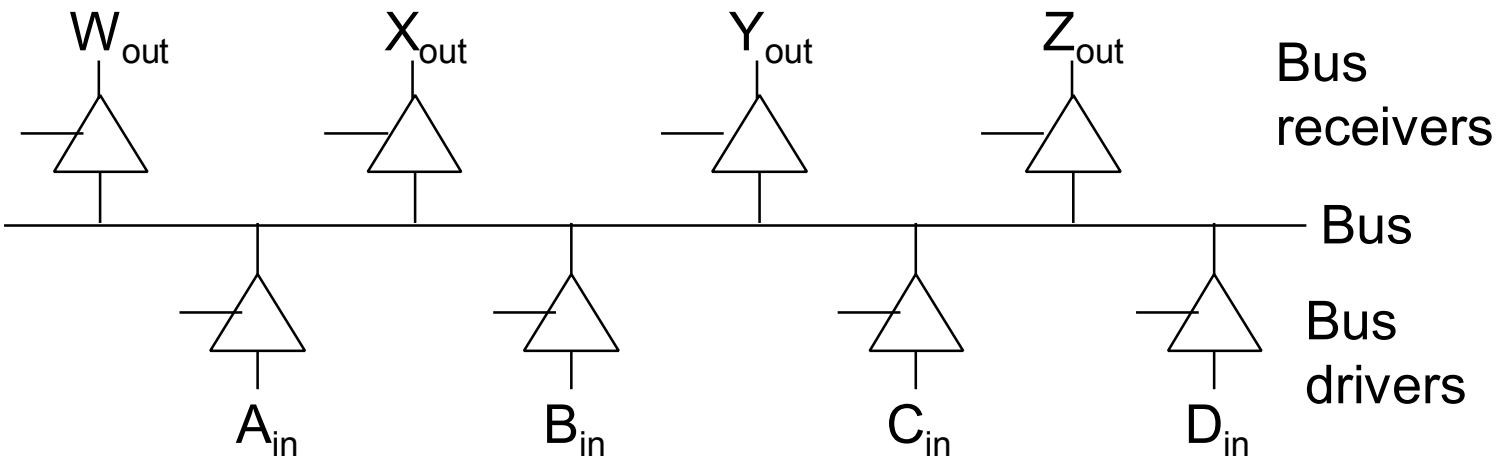
Precomputation: Example

- Binary Comparator



Bus Power

- Buses are significant source of power dissipation
  - 50% of dynamic power for interconnect switching (Magen, SLIP 04)
  - MIT Raw processor's on-chip network consumes 36% of total chip power (Wang et al. 2003)
- Caused by:
  - High switching activities
  - Large capacitive loading

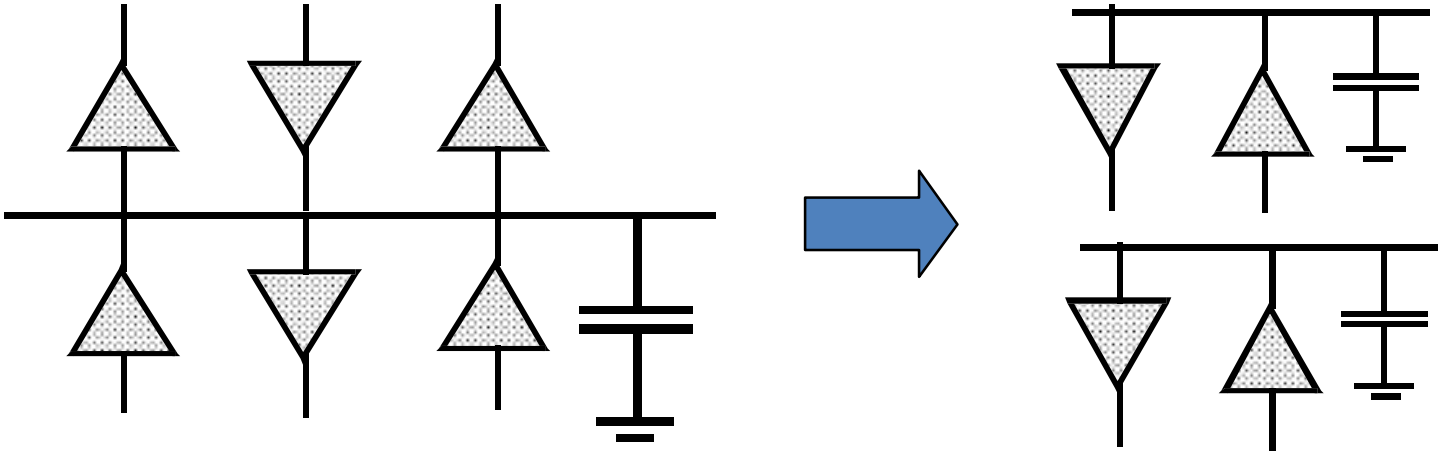


Bus Power Reduction

- For an n-bit bus: $P_{bus} = n^* \alpha f_{Clk} C_{load} V_{DD}^2$
- Alternative bus structures
  - Segmented buses (lower C_{load})
  - Charge recovery buses
  - Bus multiplexing (lower f_{Clk} possible)
- Minimizing bus traffic (n)
  - Code compression
  - Instruction loop buffers
- Minimization of bit switching activity (f_{clk}) by data encoding
- Minimize voltage swing (V_{DD}^2) using differential signaling

Reducing Shared Resources

- Shared resources incur switching overhead
- Local bus structures reduce overhead

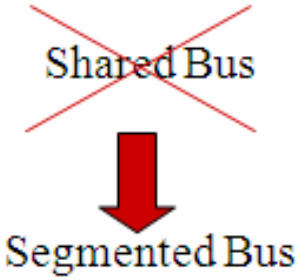


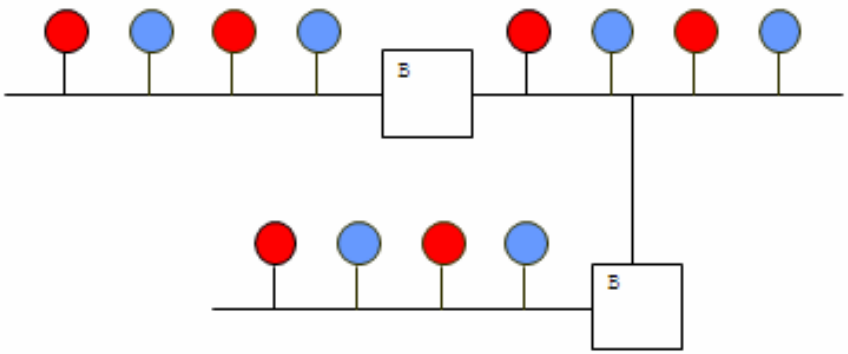
Global bus architecture

Local bus architecture

Reducing Shared Resources cont'd

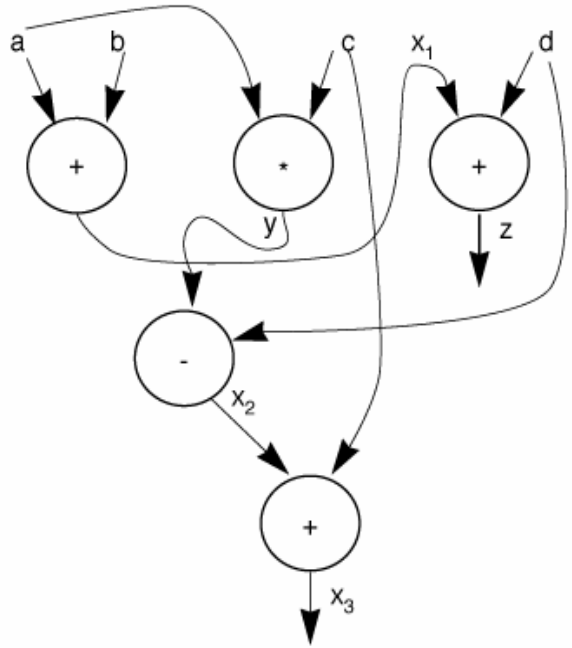
- Bus segmentation
  - Another way to reduce shared buses
  - Control of bus segment by controller blocks (B)





Design Layer: Algorithm Level

- Base elements:
  - Functions
  - Procedures
  - Processes
  - Control structures
- Description of design behavior



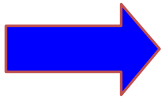
Coding styles

- Use processor-specific instruction style:
  - Variable types
  - Function calls style
  - Conditionalized instructions (for ARM)
- Follow general guidelines for software coding
  - Use table look-up instead of conditionals
  - Make local copies of global variables so that they can be assigned to registers
  - Avoid multiple memory look-ups with pointer chains

Source-code Transformations

- Minimize power-consuming activity:
  - Computation

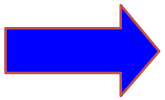
```
A*B+A*C
```



```
A*(B+C)
```

  - Communication

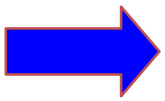
```
for (c = 1..N)
    receive (A)
    B=c*A
```



```
receive (A)
for (c = 1..N)
        B=c*A
```

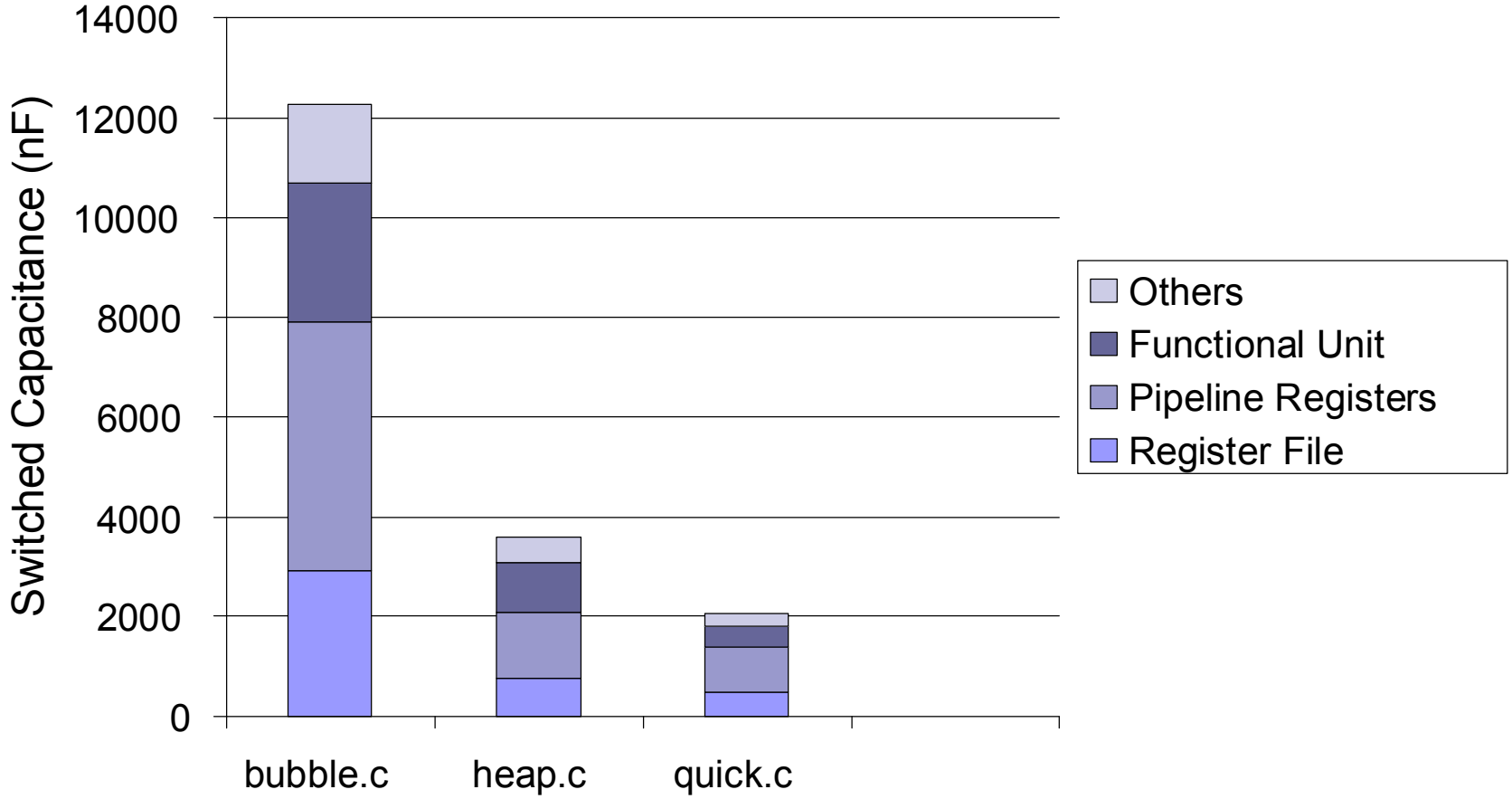
  - Storage

```
for (c = 1..N)
   B[c] = A[c]*D[c]
for (c = 1..N)
   F[c] = B[c]-1
```



```
for (c = 1..N)
F[c] = A[c]*D[c]-1
```

Datapath Energy Consumption



➔ Algorithms can differ in power dissipation

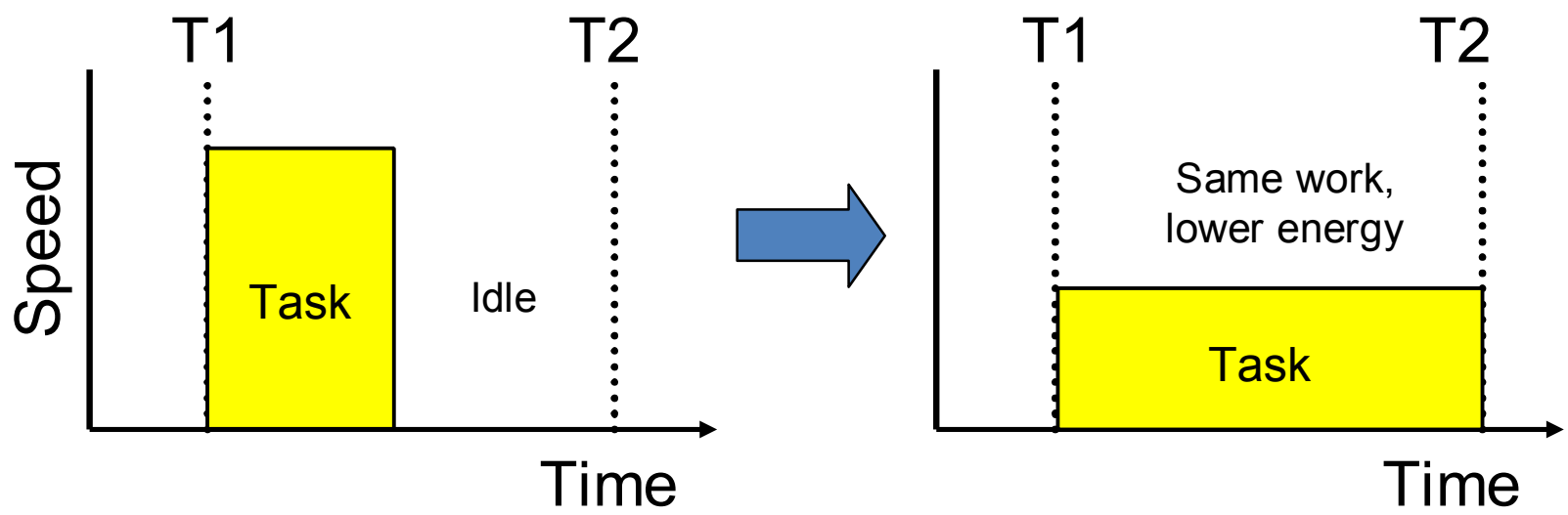
Adaptive Dynamic Voltage Scaling (DVS)

- Slow down processor to fill idle time
- More Delay ➔ lower operational voltage

- Runtime Scheduler determines processor speed and selects appropriate voltage
- Transitions delay for frequencies ~150μs
- Potential to realize 10x energy savings

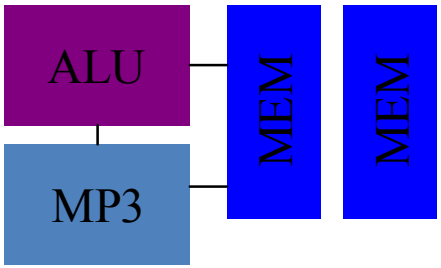
Adaptive DVS: Example

- Task with 100 ms deadline, requires 50 ms CPU time at full speed
  - Normal system gives 50 ms computation, 50 ms idle/stopped time
  - Half speed/voltage system gives 100 ms computation, 0 ms idle
  - Same number of CPU cycles but: $E = C (V_{DD}/2)^2 = E_{ref} / 4$
  - Dynamic Voltage Scaling adapts voltage to workload



Design Layer: System Level

- Basic Elements:
  - Complex modules
  - Processors
  - Calculation and control units
  - Sensors



Dynamic Power Management

- Systems are:
  - Designed to deliver peak performance, but …
  - Not needing peak performance most of the time
- Components are idle sometimes
- Dynamic power management (DPM):
  - Puts idle components in low-power non-operational states when idle
- Power manager:
  - Observes and controls the system
  - Power consumption of power manager is negligible

Processor Sleep Modes

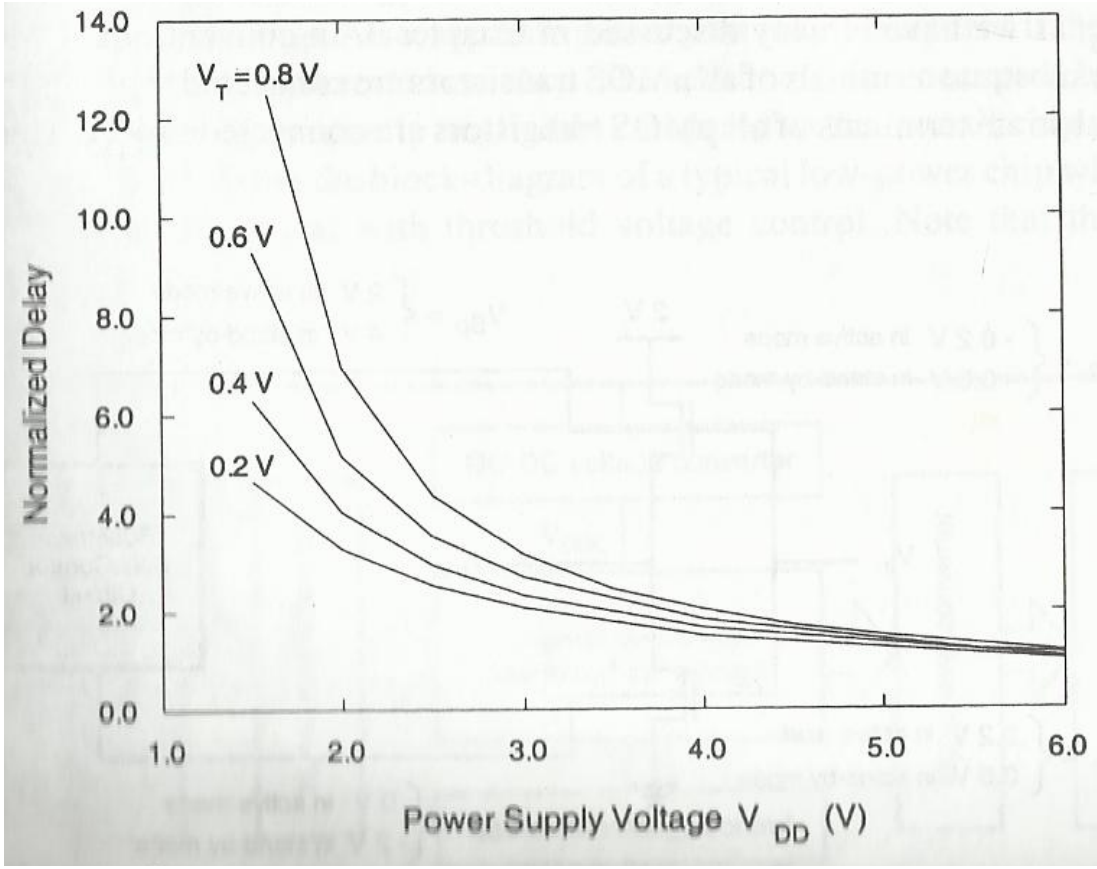
- Software power control - power management

DOZE	Most units stopped except on-chip cache memory (cache coherency)
NAP	Cache also turned off, PLL still on, time out or external interrupt to resume
SLEEP	PLL off, external interrupt to resume

Deeper sleep mode consumes less power

Deeper sleep mode requires more latency to resume

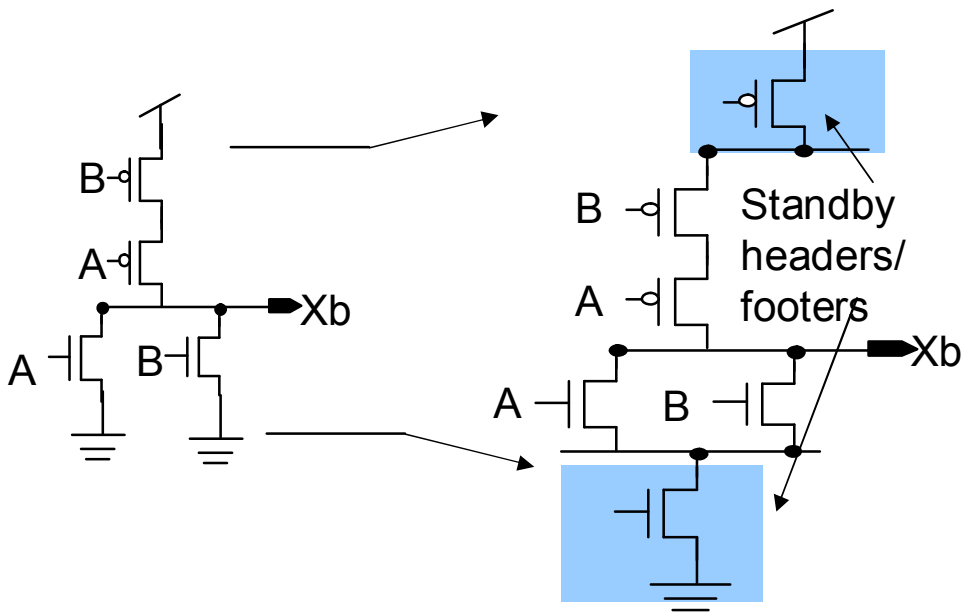
Relation between delay, supply voltage and threshold voltage



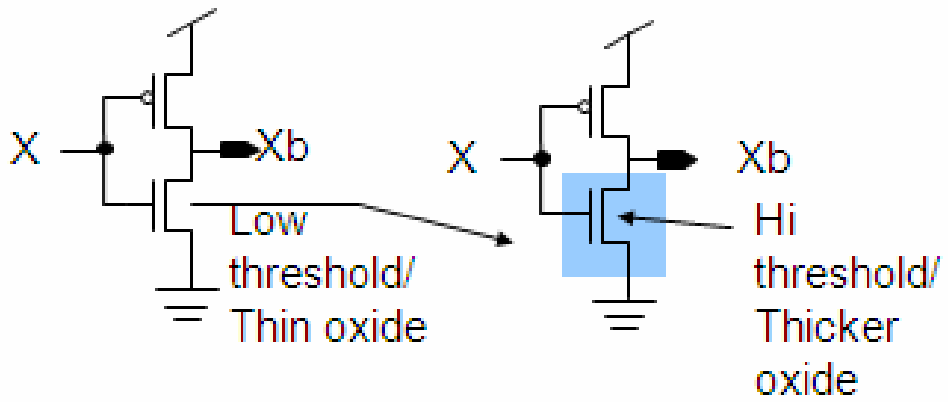
Source: Y. Leblebici, S.M. Kang, "CMOS Digital Integrated Circuits, McGraw Hill, 2002

MTCMOS

- Use header and/or footer switches to disconnect supplies when inactive.
- For performance, low-Vt for logic devices.
- 10-100x leakage improvement, ~5% perf overhead
- Loss of state when disconnected from supplies
- Large number of variants in the literature



Vt / Tox selection



- Low Vt devices on critical paths, rest high Vt
- 70-180mV higher Vt, 10-100x lower leakage, 5-20% slower
- Small fraction of devices low-Vt (1-5%)
- Thick oxide reduces gate leakage by orders of magnitude

THANK YOU