The challenges of low power design

Karen Yorav

The challenges of low power design

What this tutorial is NOT about:

- Electrical engineering
- CMOS technology

but also not

- Hand waving nonsense about trends and politics of the semiconductor industry

It WILL be:

An overview of major low-power design techniques,
keeping verification in mind

Motivation

- Portability
  - Battery life
  - Increased functionality
  - Heat generation

- Huge server farms

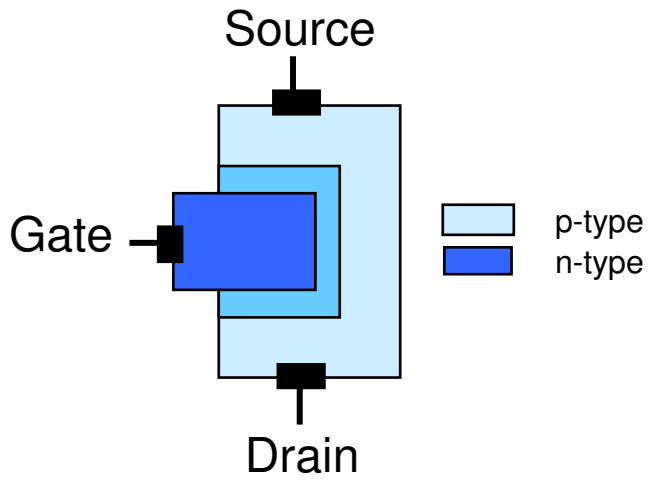
- Environmental awareness

Power is as important as performance

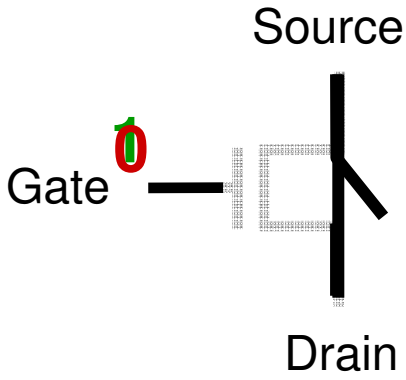
Background

A transistor

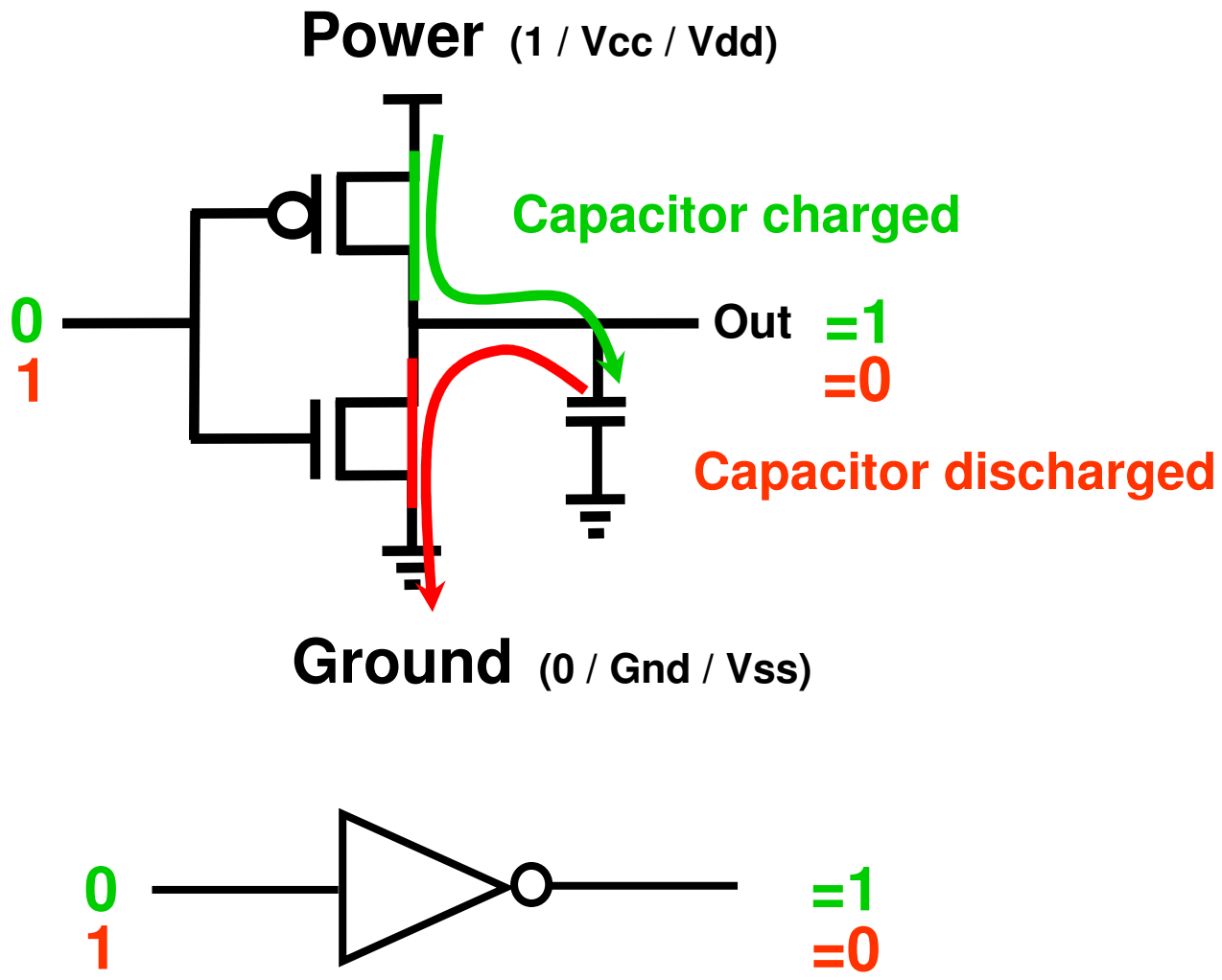
- Physically –
  layers of doped silicon



- Logically – think of it as a switch



A gate



Where is power consumed?

Dynamic power (switching power)

- Power consumed when the output of a gate changes value
- Quadratically dependent on voltage

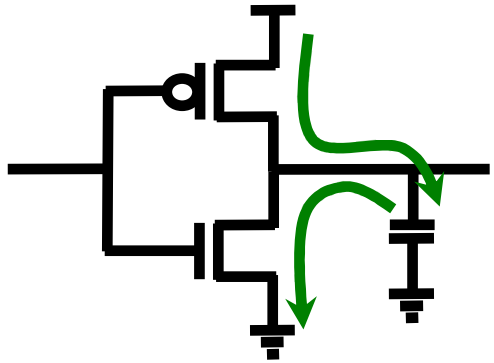
$$P_D = K \cdot C \cdot V_{dd}^2 \cdot F$$

K – Switching factor

C – Capacitence

V_{dd} – Supply voltage

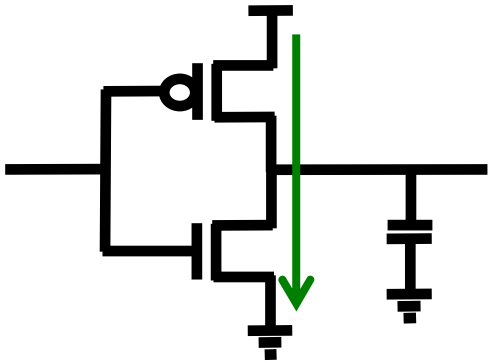
F – Frequency



Where is power consumed?

Static power (leakage)

- Power consumed by each element at all times
  - From several sources
- Grows exponentially when voltage is reduced
- Increases as transistor size shrinks



$$I_{sub} = \mu \cdot C_{ox} \cdot V_{th}^2 \cdot \frac{W}{L} \cdot e^{\frac{V_{GS} - V_T}{n \cdot V_{th}}}$$

V_{GS} – gate-source voltage

V_T – threshold voltage

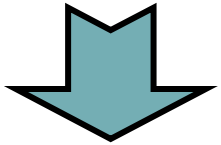
Until recently, leakage was negligible

soon to be 50% of overall power dissipation, and worse

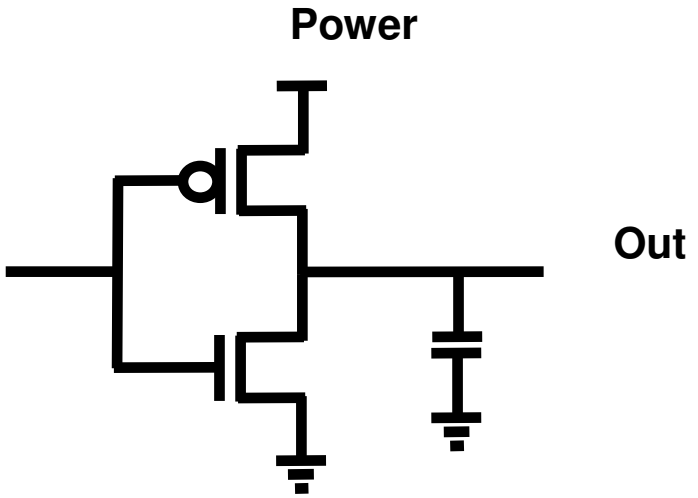
Frequency and Voltage

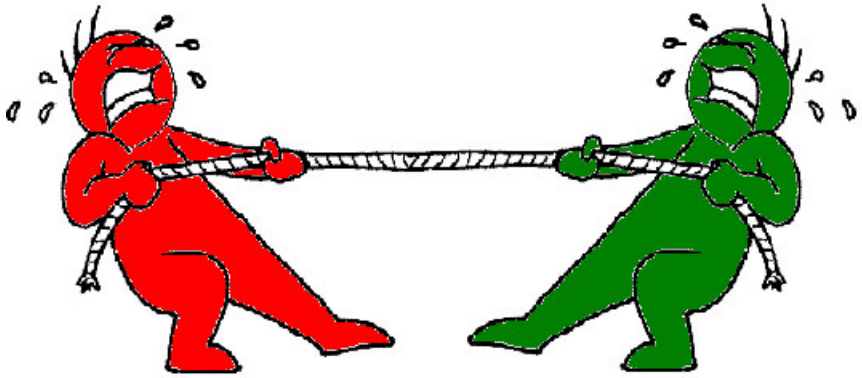
lower the voltage

increase delay

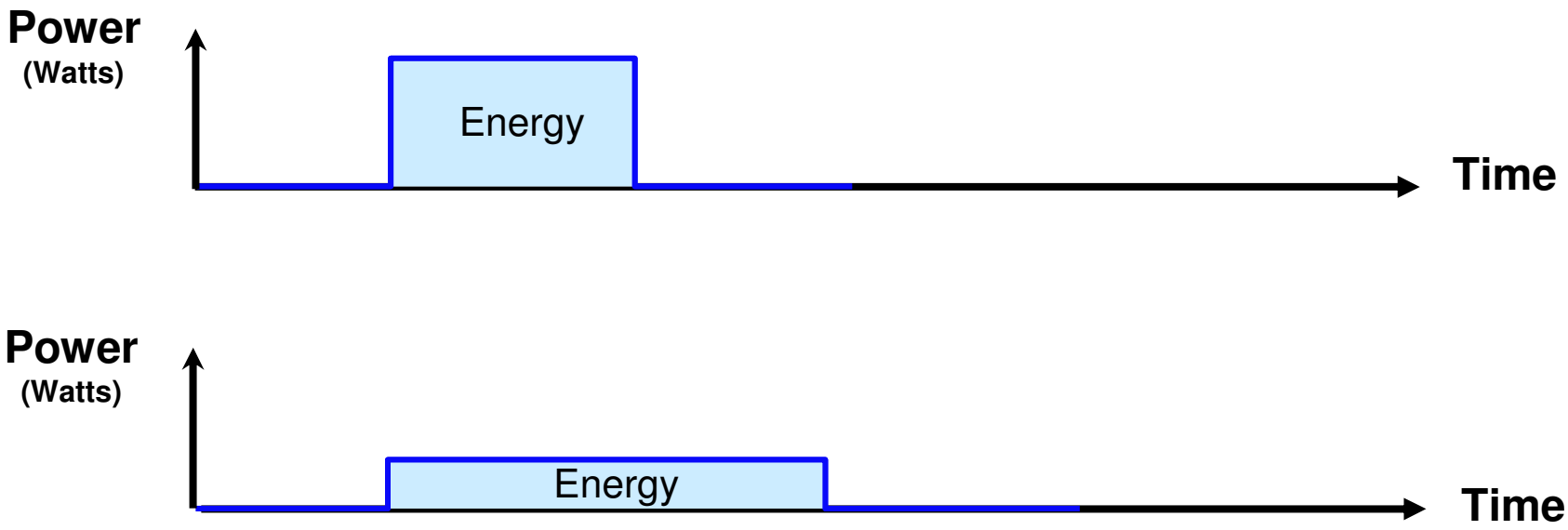


reduce frequency





Power Vs. Energy



Power Estimation

- Power consumption depends on:
  - Number of elements
  - Toggling factor of each element
  - Specific cells used
    - (size, shape, technology etc.)
  - Manufacturing variance
  - Operating temperature
- Difficult to estimate, large error margin
- Existing tools use:
  - Measurements from previous designs
  - Use models
  - Complicated formulae
  - …

Power Verification – what is the question?

1. Are there going to be electrical problems?
   - In-rush currents, capacitance, voltage variance, etc.

2. How much power will the design consume?
   - This is ***Power Characterization***, or ***Power Estimation***

3. Is the power scheme implemented as planned?
   - Verifying the power control circuitry

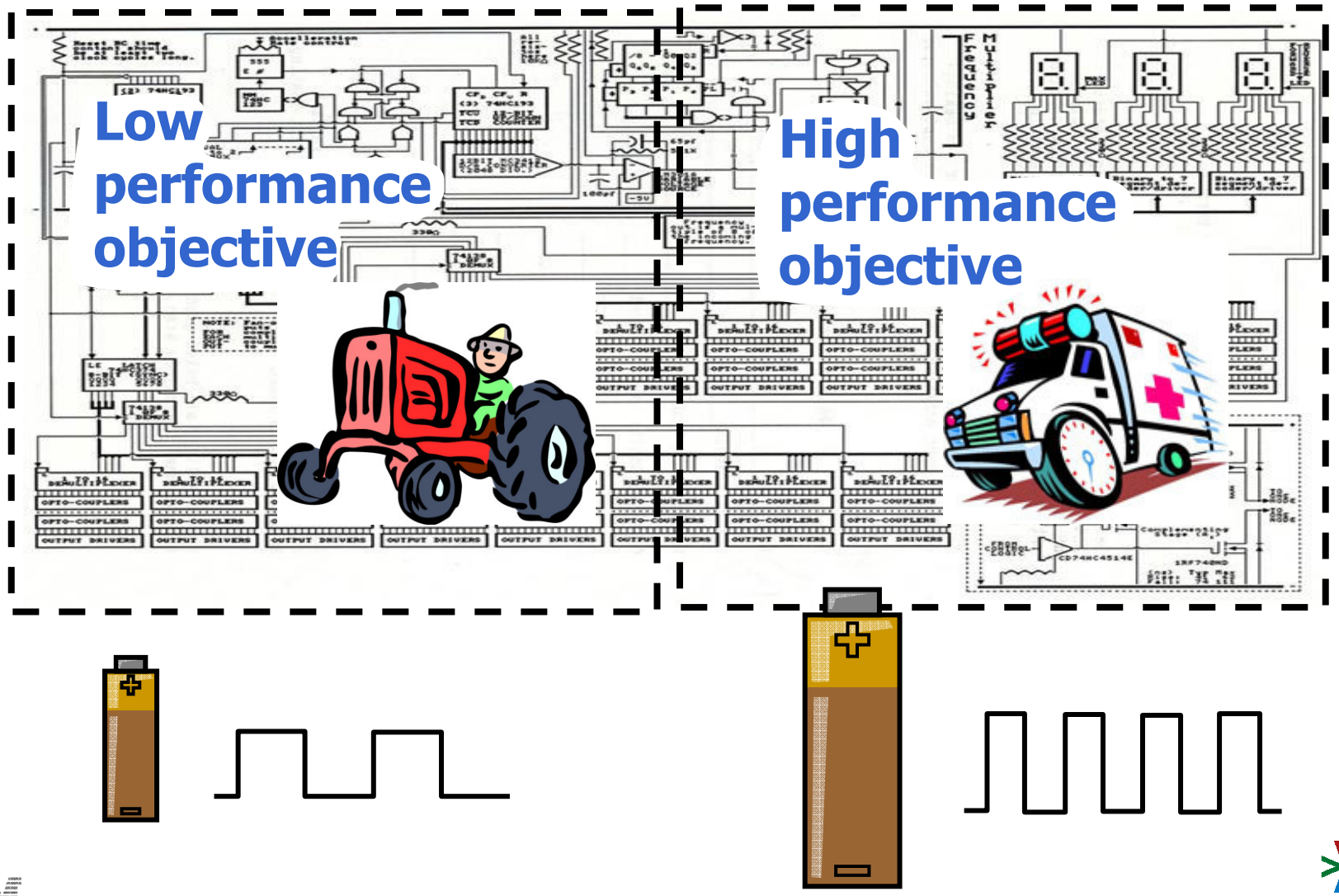
4. Is the design functioning correctly with the implemented power scheme?
   - Verifying functional correctness of circuits that employ low power design techniques

Power Saving Techniques

1. Multiple power domains
2. Frequency and voltage scaling
3. Clock gating
4. Power gating
5. Architectural exploration

Multiple Power Domains

Different components run with different voltage/frequency



Multiple Power Domains – challenges and solutions

- Verification issue:

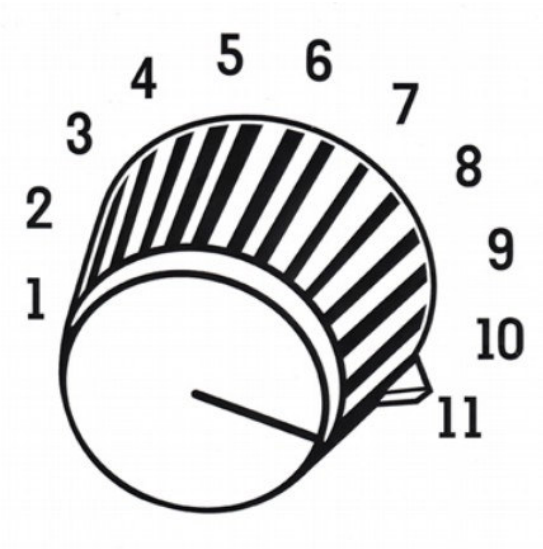
  asynchronous interfaces

- Formal solution
  - model the clocks

- Simulation solution
  - model the clocks

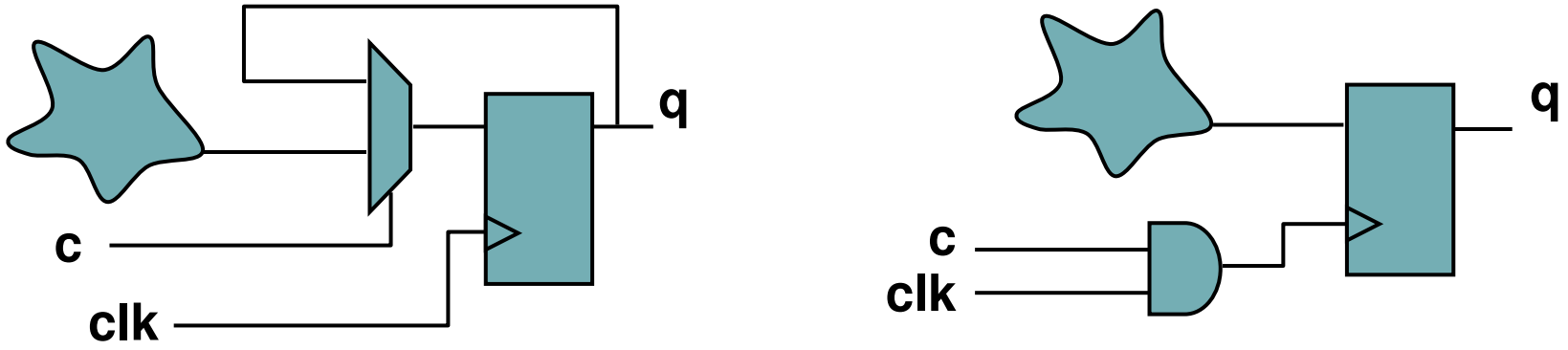
Frequency and Voltage Scaling

- Dynamically control the voltage and frequency
  - Increase voltage when high performance is required
  - Reduce the voltage when not needed
- On-chip power circuitry controls the voltage and frequency
- Several electrical issues
  - if all goes well electrically there is no effect on functionality



Clock Gating

- Prevent the clock from ticking
  - save dynamic power
    - both inside latches and on the clock distribution network

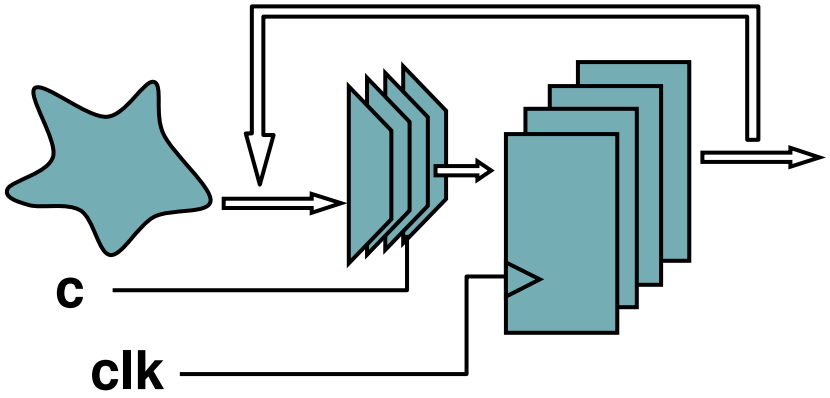


Feedback Loop Elimination

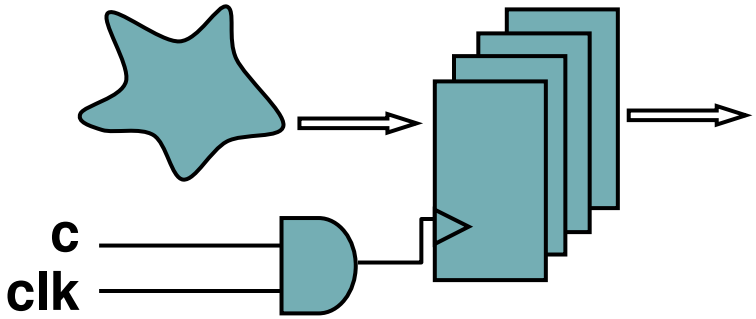
A special purpose cell, which costs a lot more

Clock Gating

Much more effective if the same gating function is applied to large sets of registers



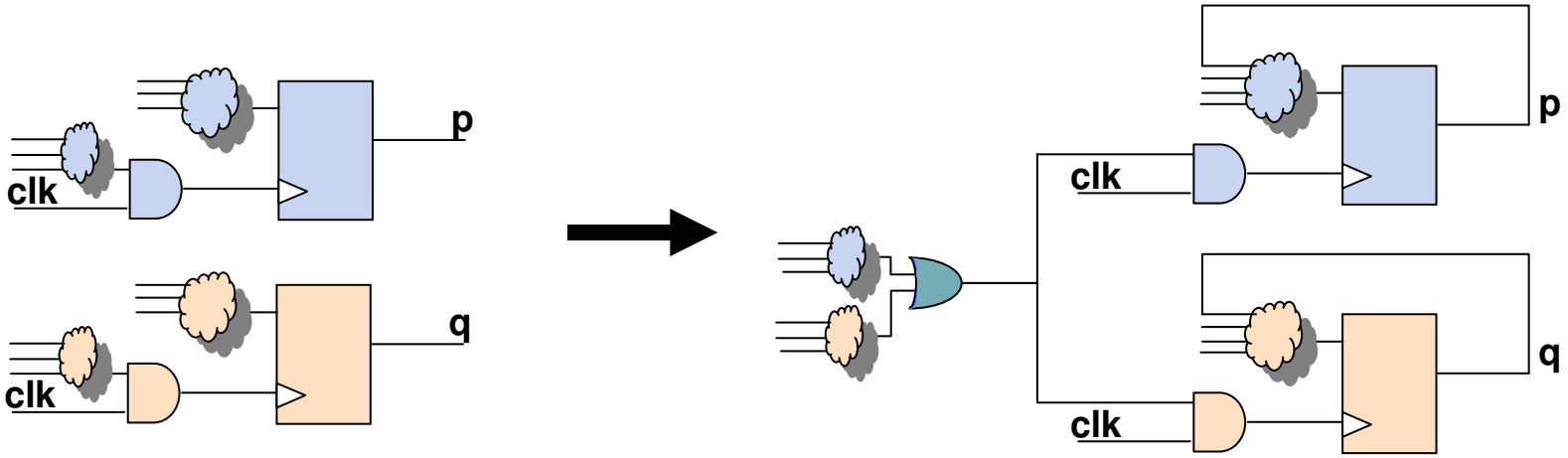
Before



After

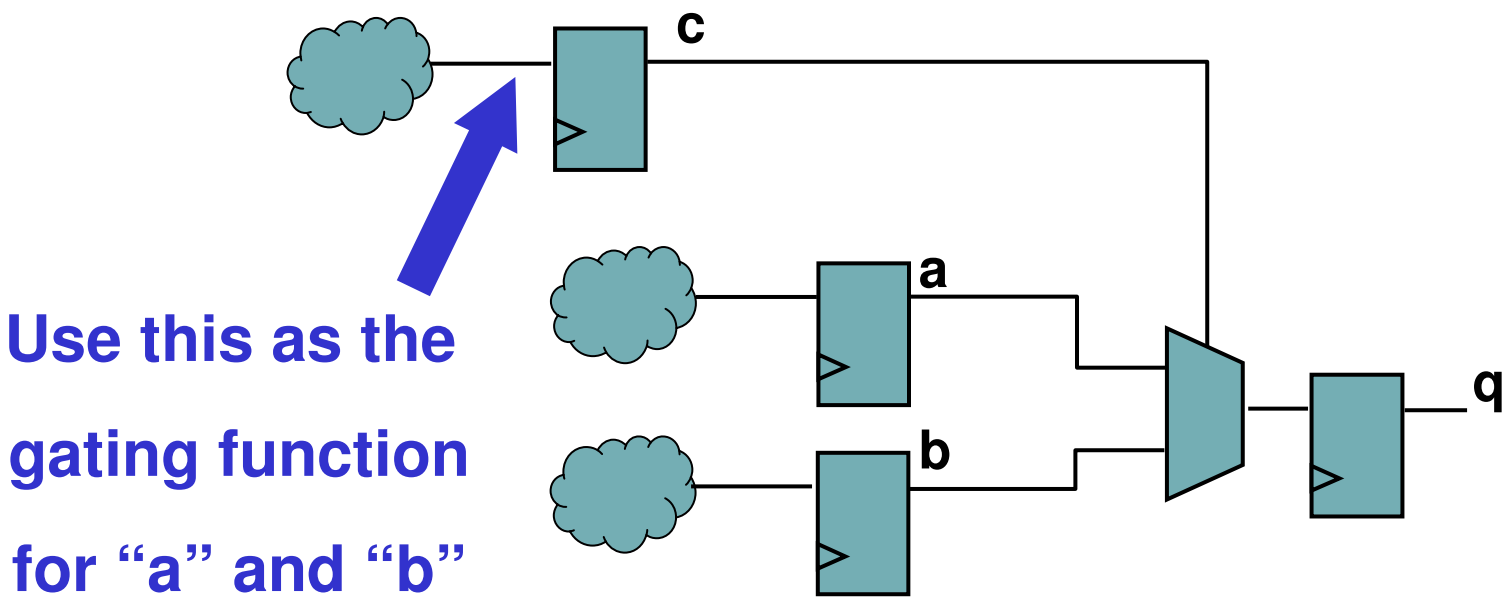
Approximation of Gating Functions

- Approximation enables gating more latches with the same function
  - Gates less often, but perhaps saves more
  - Requires adding back feedback loop



Clock Gating – Cont.

Using observability don't care conditions



No longer (Boolean) equivalent to original!

Clock Gating

- There are tools to automatically apply clock gating at the net-list level
  - Rely on heuristics to determine benefit
- Hand crafted clock-gating can be more powerful
  - but error-prone

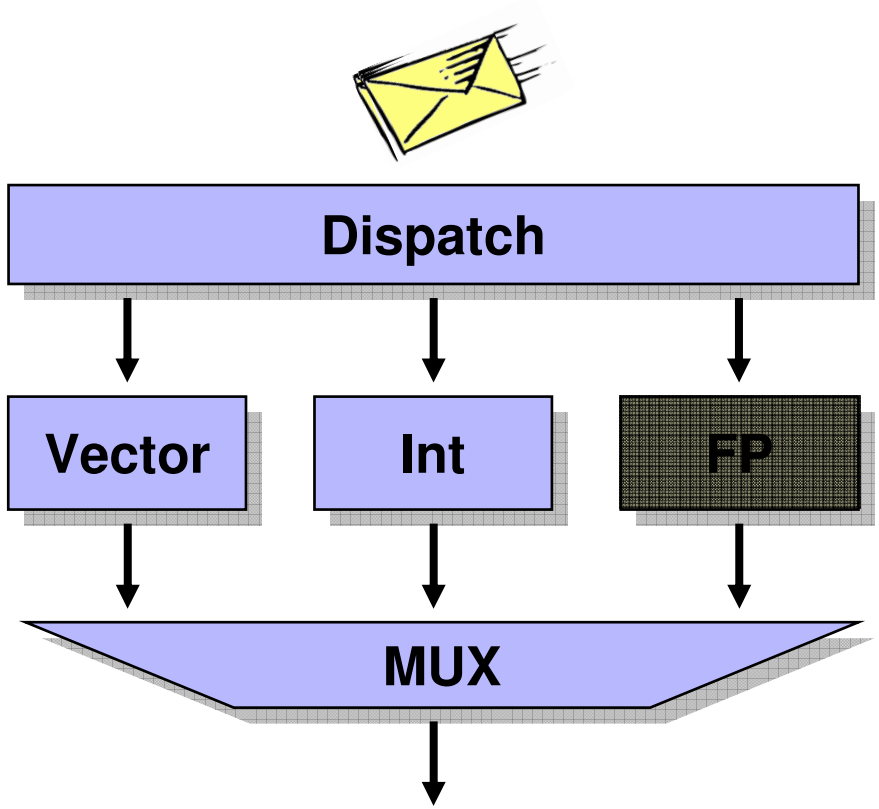
Clock gating is reported to reduce power consumption by 20% - 60%

Clock Gating – Challenges and Solutions

- Where and when can we gate the clock?
  - Find gating functions suitable for many latches
- When done manually
  - Boolean / Sequential equivalence checking
- When applied to whole blocks / unit (functional gating)
  - Scalability of sequential equivalence tools
- Do we really reduce power?
  - Need better power estimation capabilities

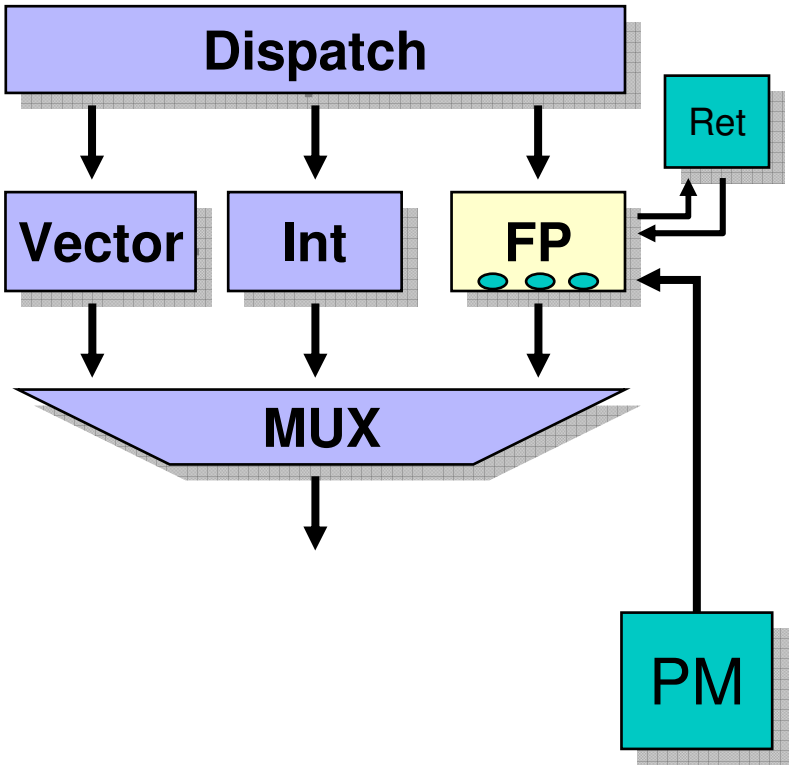
Power Gating

- Completely turn off the power to parts of the design when they are not being used



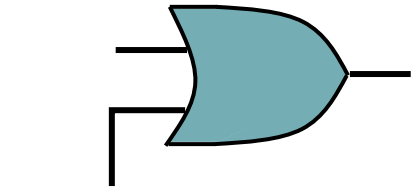
Power gating

- Isolation
  - A powered-off gate must not drive powered-on gate *OR ELSE!*
- State Retention
  - Powered-off FFs lose their state, when turned on they will have arbitrary values
  - Save content of important FFs in an “always-on” power domain
  - Copy back the state when power is restored
- Power Management
  - Control power-off and power-on sequences
- We are ignoring many electrical issues!



Modeling power-gated designs

- Electrically:
  - Isolation elements prevent back current
  - During power-on latches have un-defined values (X value)
  - When power stabilizes latches have arbitrary values



isolation activated

- Functionally:
  - Isolation elements are simple gates
  - Powered-off Flip-Flops are non-deterministic

Verification of Power Gated Designs

- Isolation
  - Is it possible for a powered-off element to drive a powered-on element?
  - Fundamentally a structural check
- Control
  - Does the power management machine obey all constraints?
- Interaction
  - Between power-gated and always-on domains
  - Between power-gated domains that are switched independently

Verification of Power Gated Designs

In simulation:

- Extra mode of operation
  - increase in coverage space
- Non-determinism to model power-on state
  - dramatic increase in coverage space
- Solution: use 3-valued simulation
  - slight abstraction
  - simulation is significantly slower
  - cover more space with a single run

Interrupt – a word on X

Different communities use different X differently

- **The electrical X**
  An un-determined voltage that may be interpreted as either 0 or 1
  - if Vdd=1v then 0.5 is undetermined,
    if Vdd=5v then 0.5 is a definite zero

- **The logical X**
  Either 0 or 1, we do not know which
  - $X \wedge 0 = 0 \quad X \wedge 1 = X \qquad X \oplus X = X$
  - An abstraction of a constant, loses information
  - (In STE – this would be the “top” value)

- **Don’t care**
  Meaning this value is not important

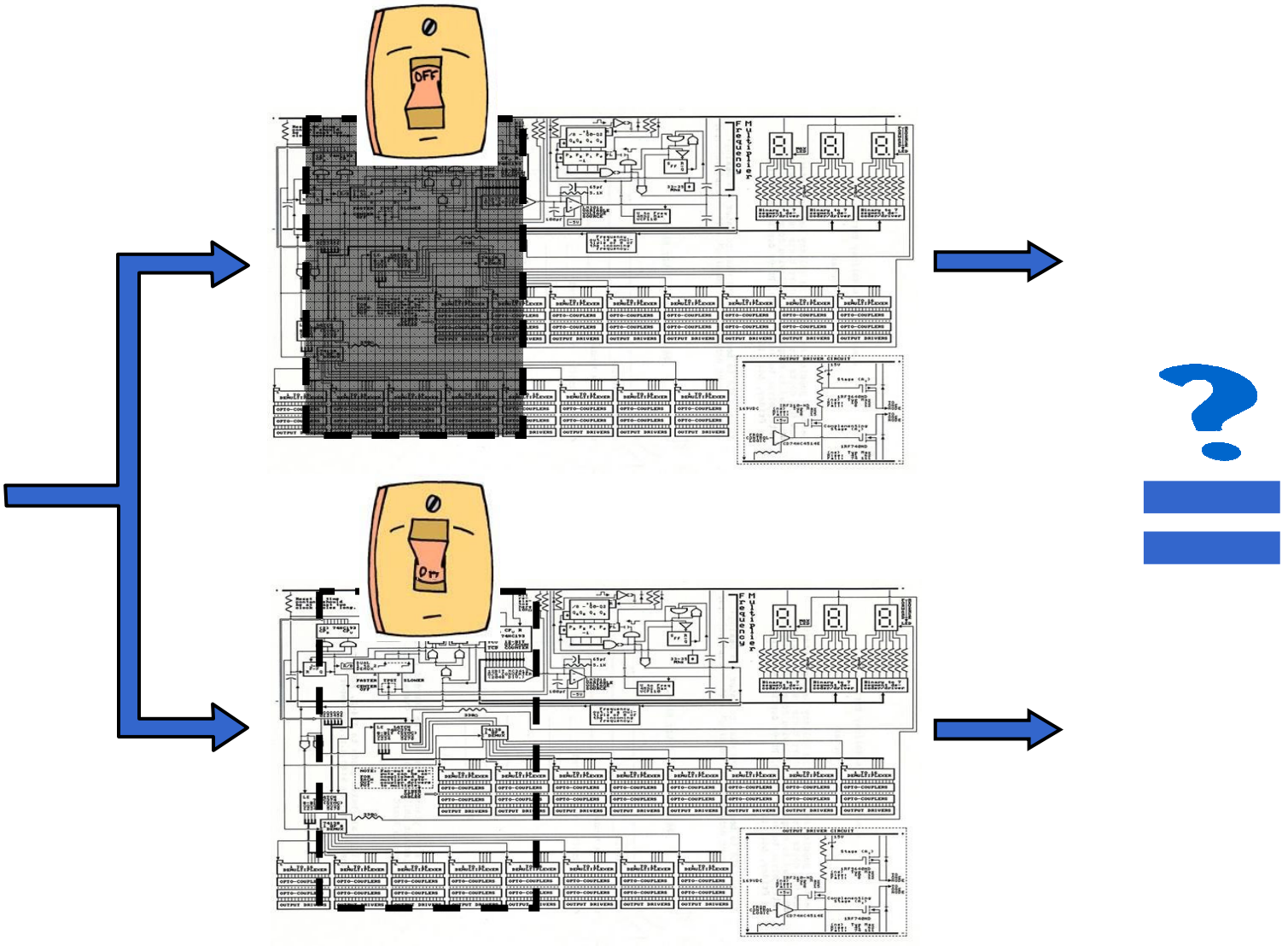
Verification of Power Gated Designs

Formal Verification:

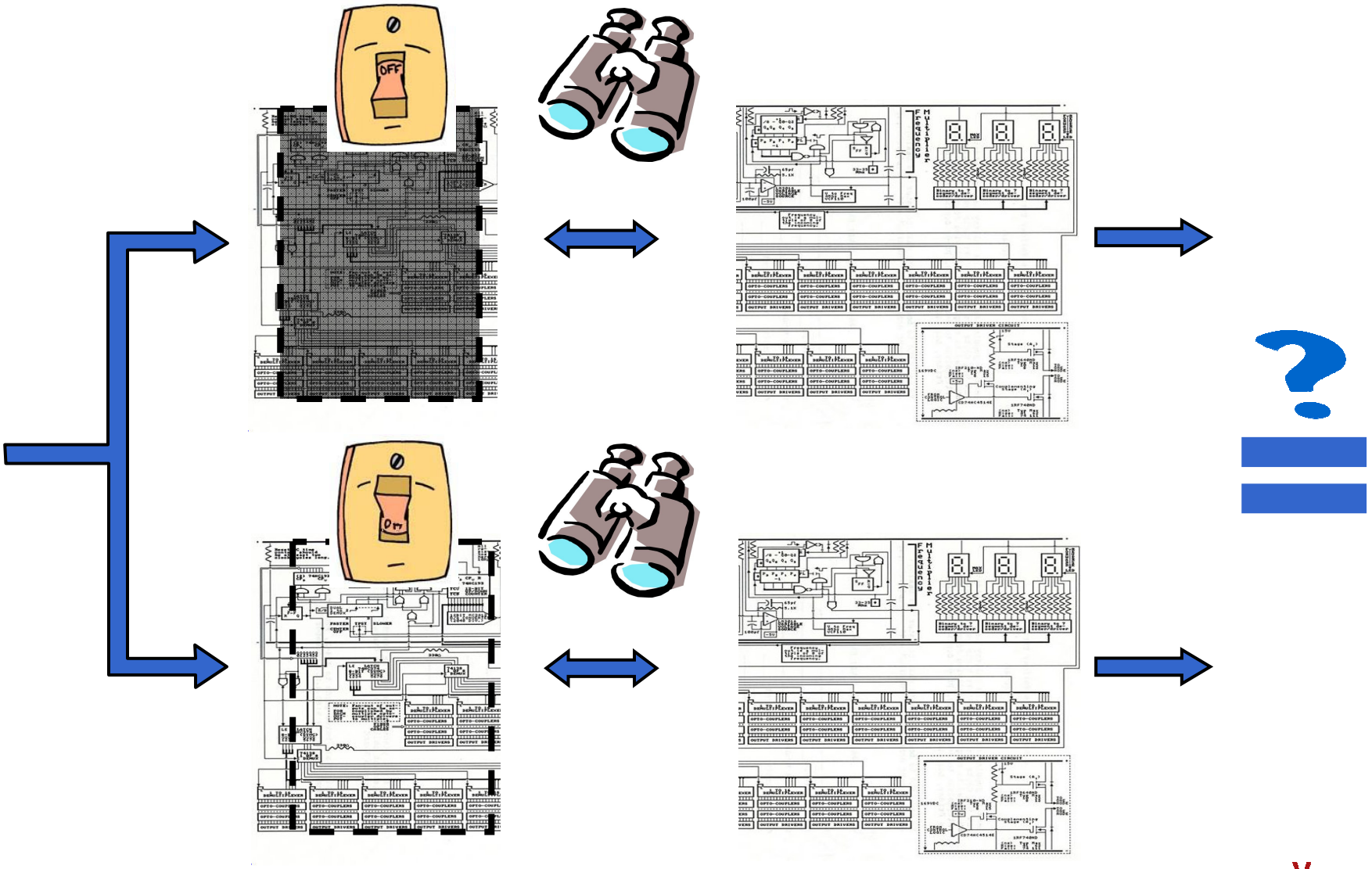
- Power gating is normally done at the unit level
  - typically too large for model checking
- Increased state-space due to non-determinism
- Recently developed methodology:
  - Perform functional verification with power gating disabled
  - Use *sequential equivalence checking* to prove that enabling power gating does not change the functional behavior
  - Sequential equivalence is much easier in this setup

"Functional Verification of Power Gated Designs by Compositional Reasoning", CAV'08

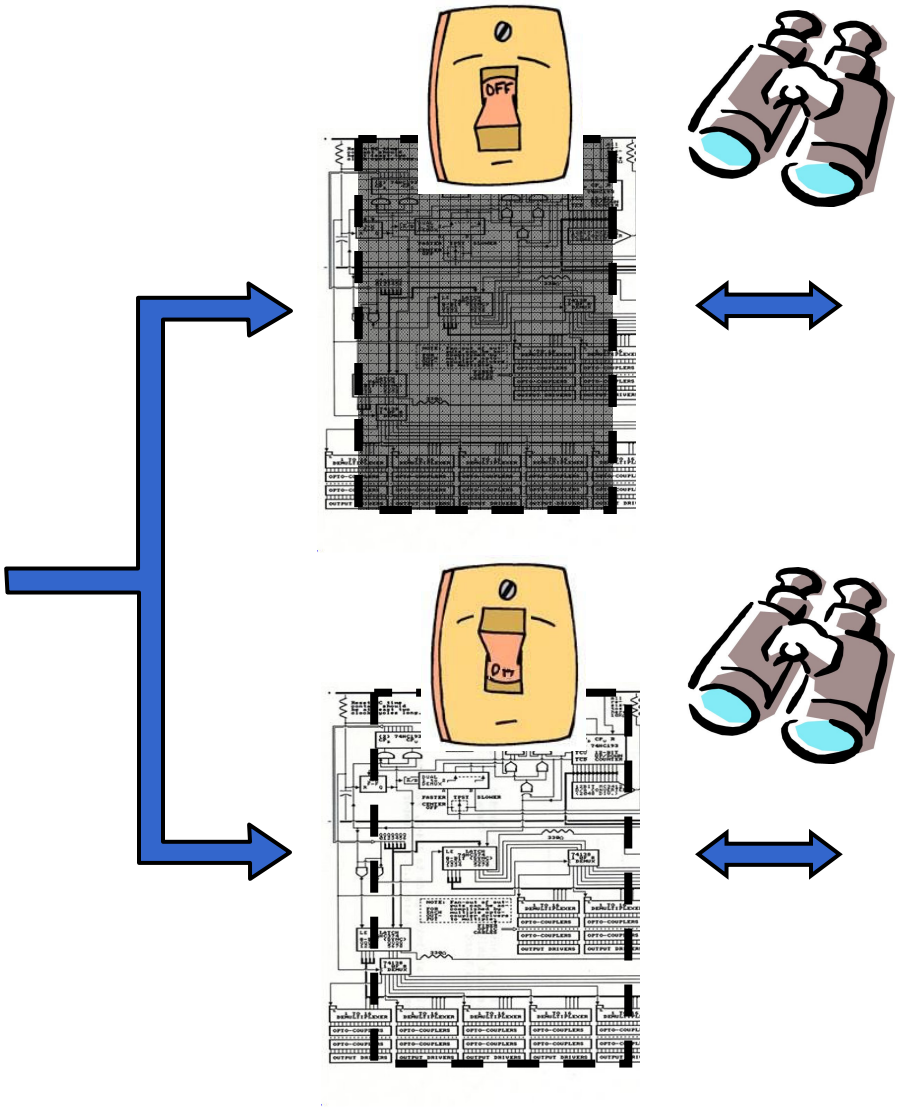
Sequential Equivalence for Power Gating



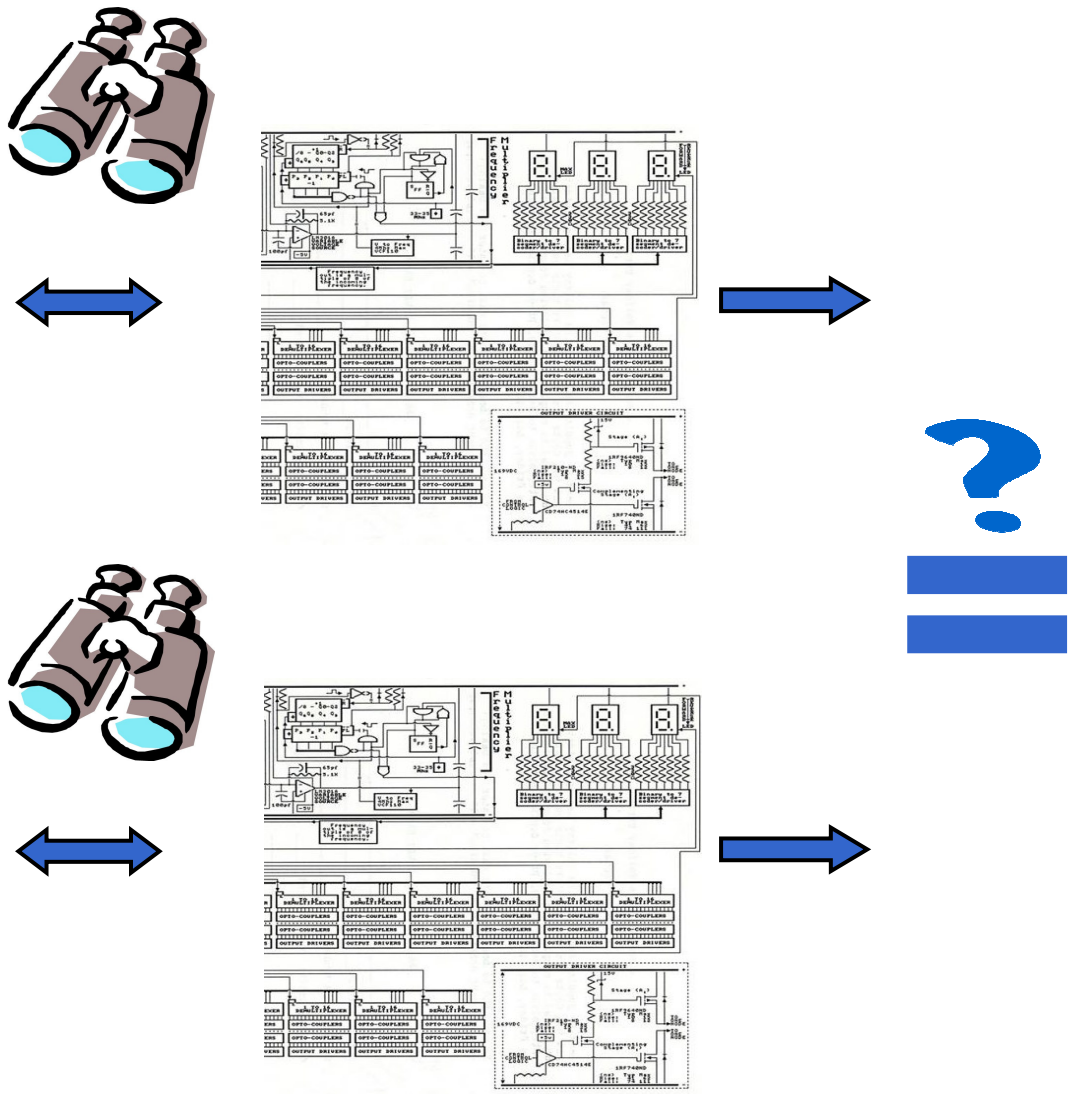
Sequential Equivalence for Power Gating



Sequential Equivalence for Power Gating



Sequential Equivalence for Power Gating



Other issues with Power Gating

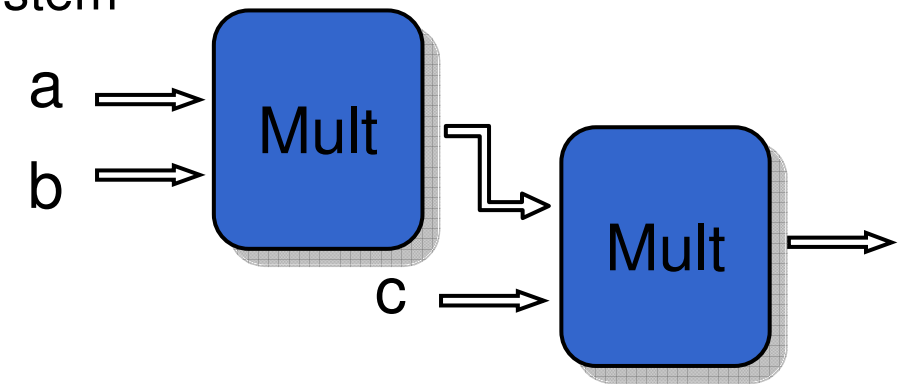
- Scan chains
- When is it beneficial to turn off a unit?
- Fine grain Vs. coarse grain power gating
- Reset Vs. state retention

Summary of low-power design techniques

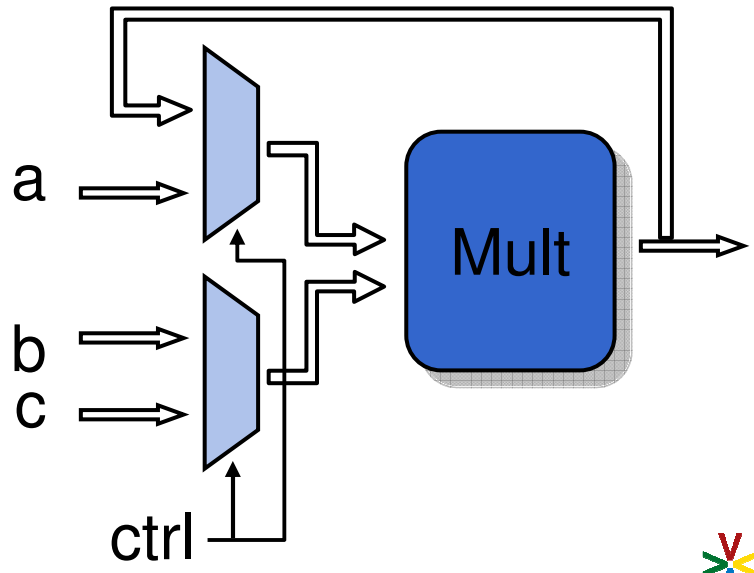
Power-reduction technique	Benefit	Timing penalty	Area Penalty	Methodology Impact			
				Arch.	Design	Verif.	Impl.
Clock gating	Medium	Little	Little	Low	Low	None	Low
Multi voltage	**Large**	Some	Little	High	Medium	Low	Medium
Frequency and voltage scaling	**Large**	Some	Some	High	High	High	High
Power gating	**Huge**	Some	Some	High	High	High	High

Architectural exploration

- The highest potential for reducing power consumption is at the system architecture level
  - system partitioning
  - bandwidth on busses
  - pipelining
  - redundancy
  - performance-critical blocks



- Use architectural exploration tools
  - power / performance tradeoff

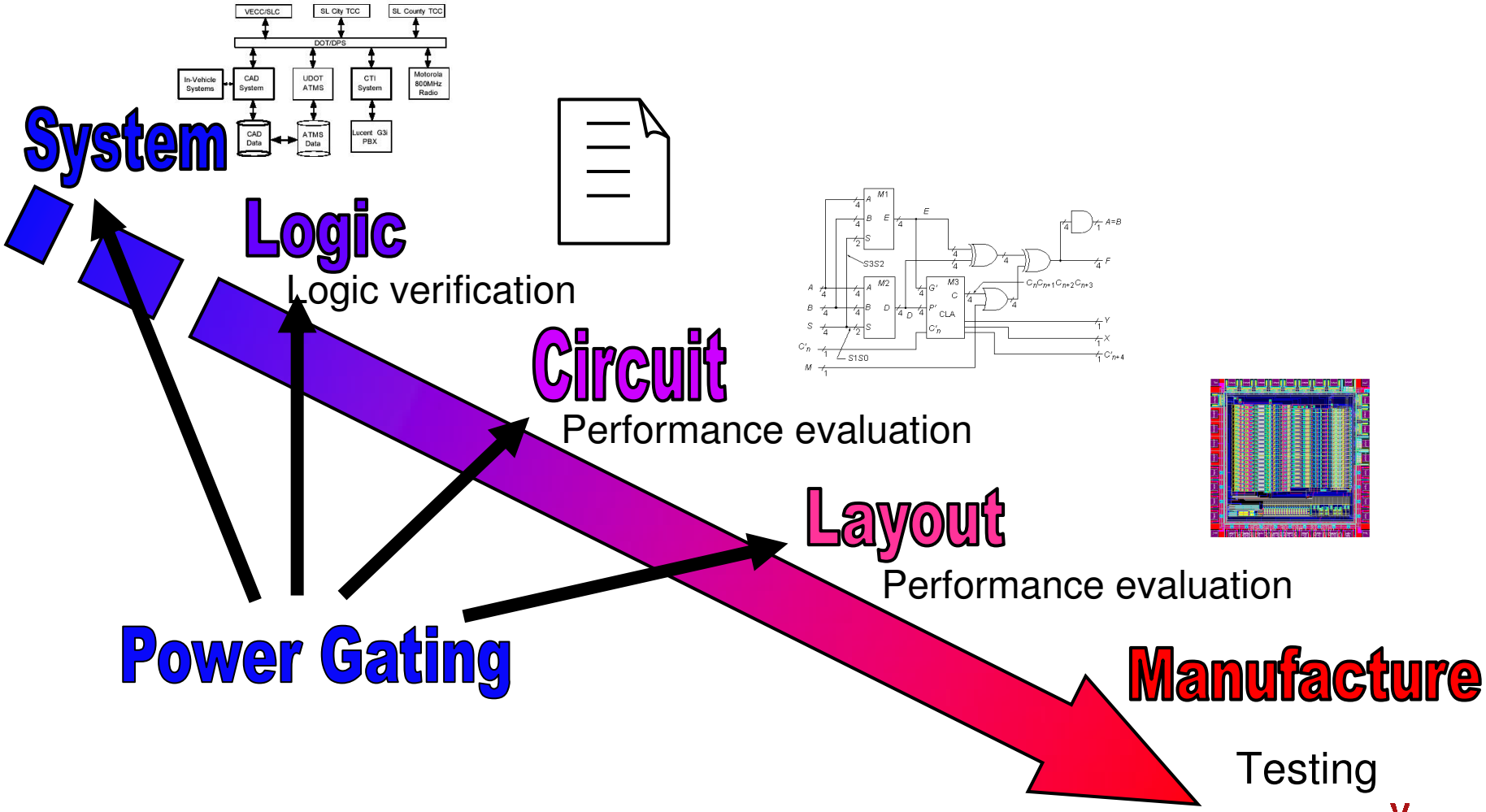


- Power estimation tools are not accurate

Design Flow of Low Power Applications

The design flow

"*It takes a village to do low-power design*" (Dylan McGrath, EETimes)



Power Specification

- A standard language for describing power design
  - power domains
  - power modes
  - power lines / switches / fences / retention registers
  - voltages
  - …
- CPF – Common Power Format
  - Developed as a standard by the Si2 organization
  - Donated by Cadence
- UPF – Unified Power Format
  - Approved as a standard by Accellera, now being worked on as an IEEE standard
  - Based on donations by Synopsys, Mentor Graphics, and Magma Design Automation

I recommend:

"Low power methodology manual for system-on-chip design"
Michael Keating, David Flynn, Robert Aitken, Alan Gibbons, Kaijian Shi

Many thanks to:

Eli Arbel, Sharon Barner, Cindy Eisner, Amir Nahir, Orna Raz, Giora Yorav