Chapter 4
Low-Power VLSI Design

Jin-Fu Li

Advanced Reliable Systems (ARES) Lab.

Department of Electrical Engineering

National Central University

Jhongli, Taiwan

Outline

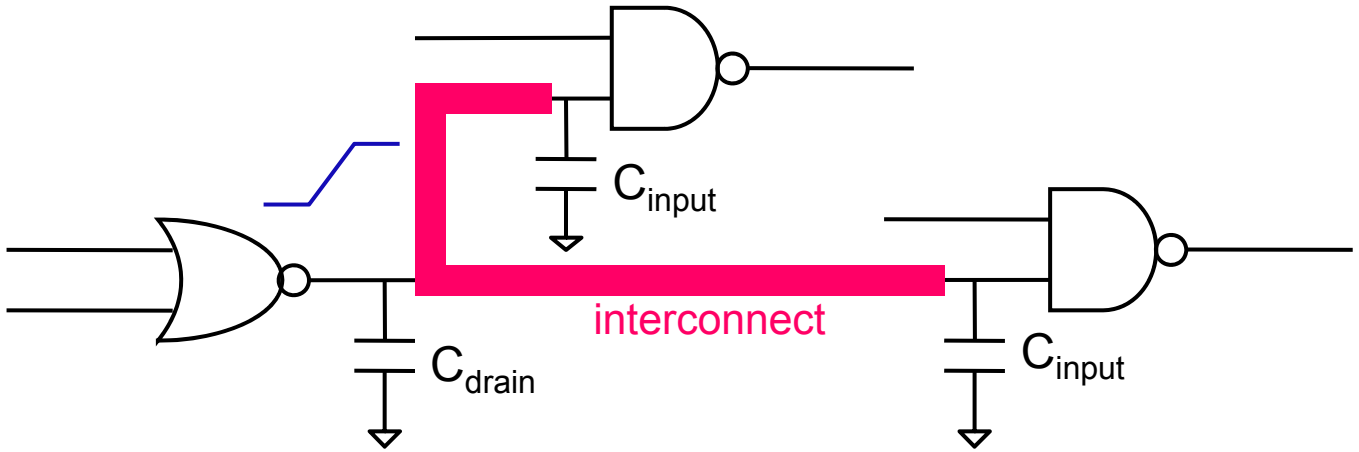
- Introduction
- Low-Power Gate-Level Design
- Low-Power Architecture-Level Design
- Algorithmic-Level Power Reduction
- RTL Techniques for Optimizing Power

Introduction

- Most SOC design teams now regard power as one of their top design concerns
- Why low-power design?
  - Battery lifetime (especially for portable devices)
  - Reliability
- Power consumption
  - Peak power
  - Average power

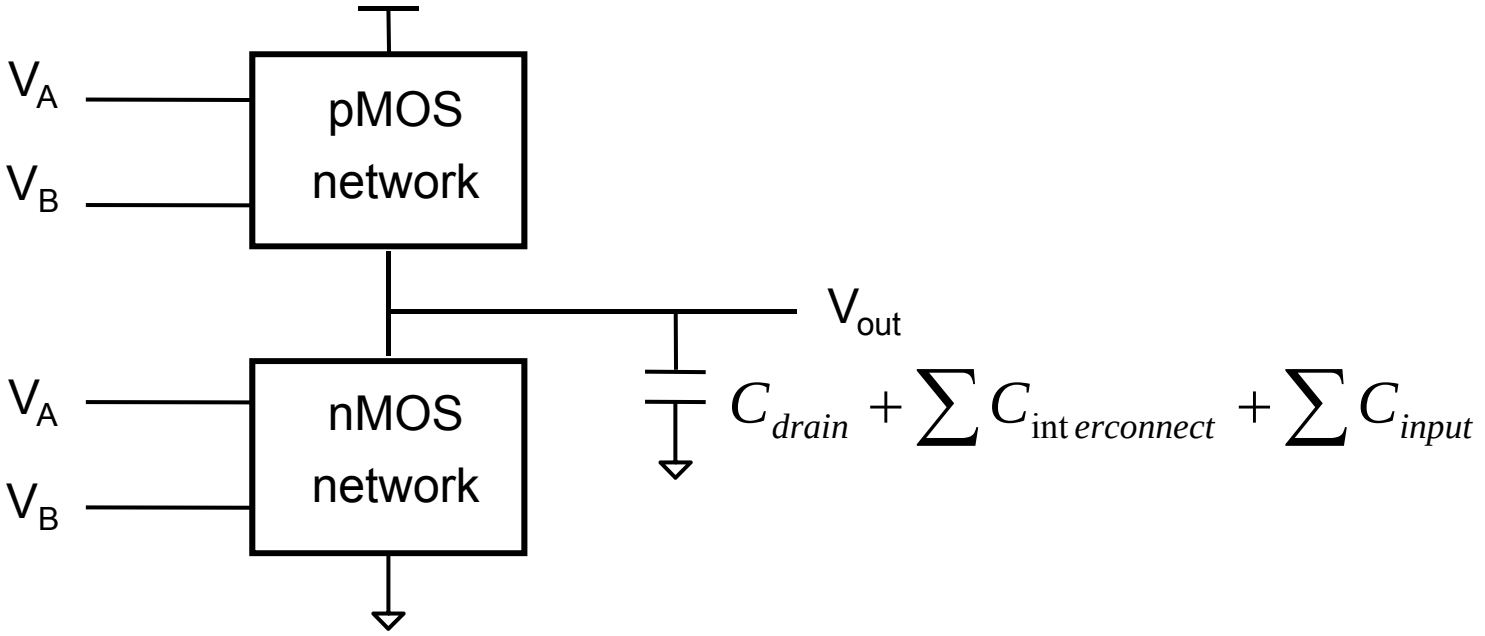
Overview of Power Consumption

- Average power consumption
  - Dynamic power consumption
  - Short-circuit power consumption
  - Leakage power consumption
  - Static power consumption
- Dynamic power dissipation during switching



Overview of Power Consumption

- Generic representation of a CMOS logic gate for switching power calculation



$$P_{avg} = \frac{1}{T}\left[\int_0^{T/2} V_{out}\left(-C_{load}\frac{dV_{out}}{dt}\right)dt + \int_{T/2}^{T}(V_{DD}-V_{out})\left(C_{load}\frac{dV_{out}}{dt}\right)dt\right]$$

Overview of Power Consumption

- The average power consumption can be expressed as

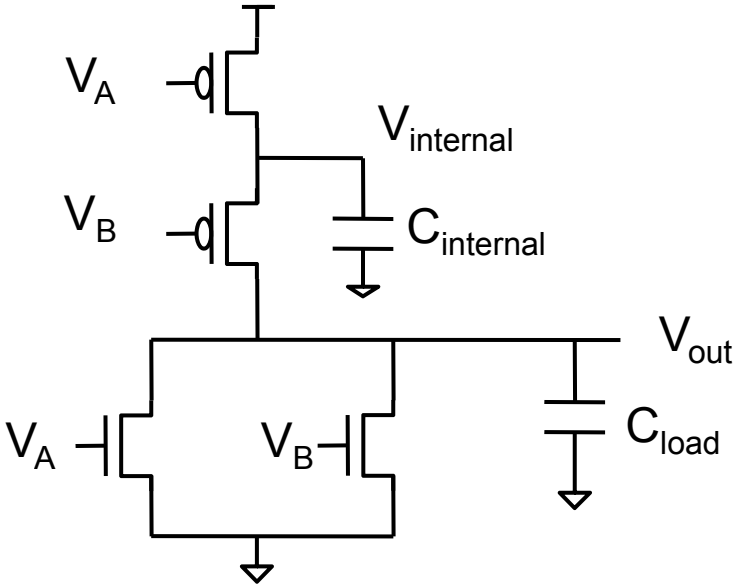
$$P_{avg} = \frac{1}{T} C_{load} V_{DD}^2 = C_{load} V_{DD}^2 f_{CLK}$$

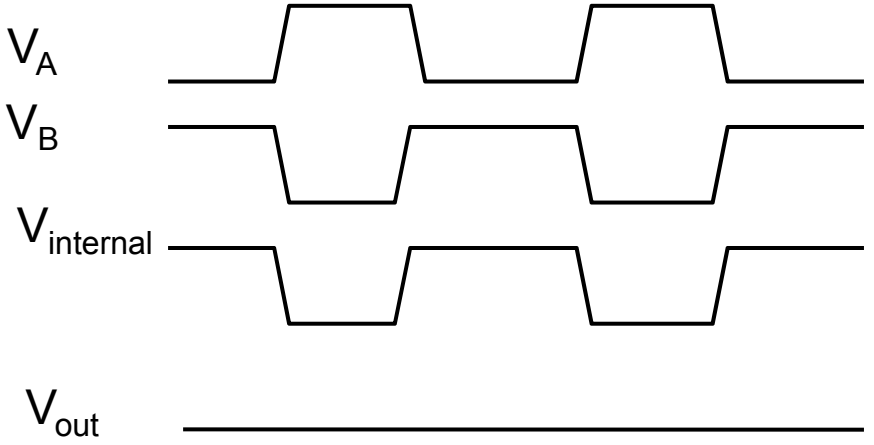
- The node transition rate can be slower than the clock rate. To better represent this behavior, a node transition factor (α_T) should be introduced

$$P_{avg} = \alpha_T C_{load} V_{DD}^2 f_{CLK}$$

- The switching power expressed above are derived by taking into account the output node load capacitance

Overview of Power Consumption





The generalized expression for the average power dissipation can be rewritten as

$$P_{avg} = \left(\sum_{i=1}^{\#ofnodes} \alpha_{Ti} C_i V_i \right) V_{DD} \, f_{CLK}$$

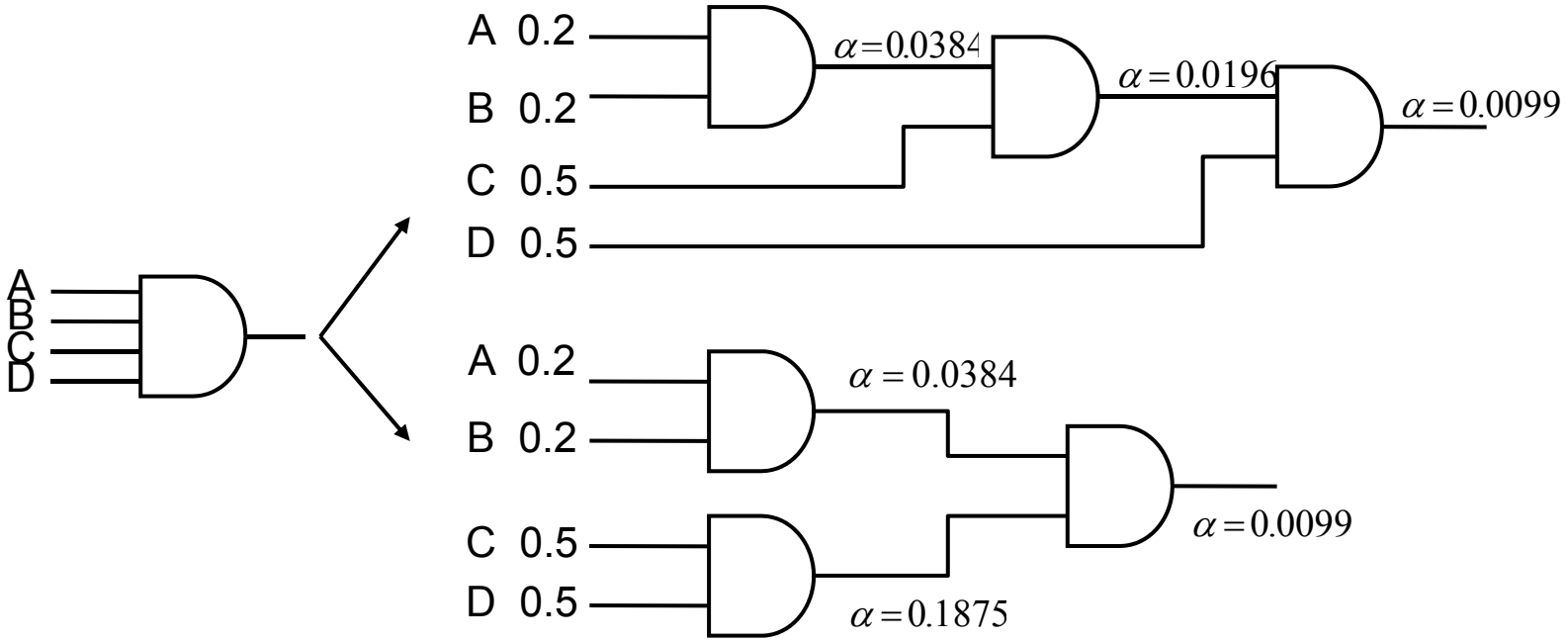
Gate-Level Design – *Technology Mapping*

- The objective of logic minimization is to reduce the boolean function.
- For low-power design, the signal switching activity is minimized by restructuring a logic circuit
- The power minimization is constrained by the delay, however, the area may increase.
- During this phase of logic minimization, the function to be minimized is

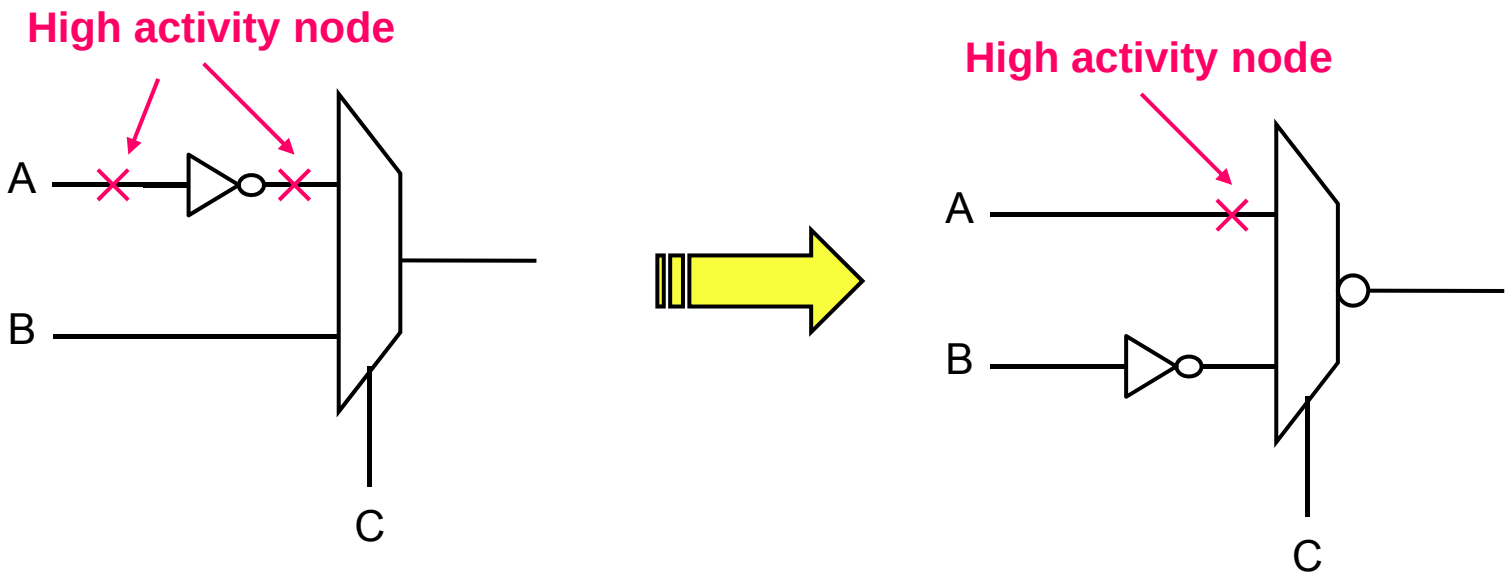
$$\sum_i P_i (1 - P_i) C_i$$

Gate-Level Design – *Technology Mapping*

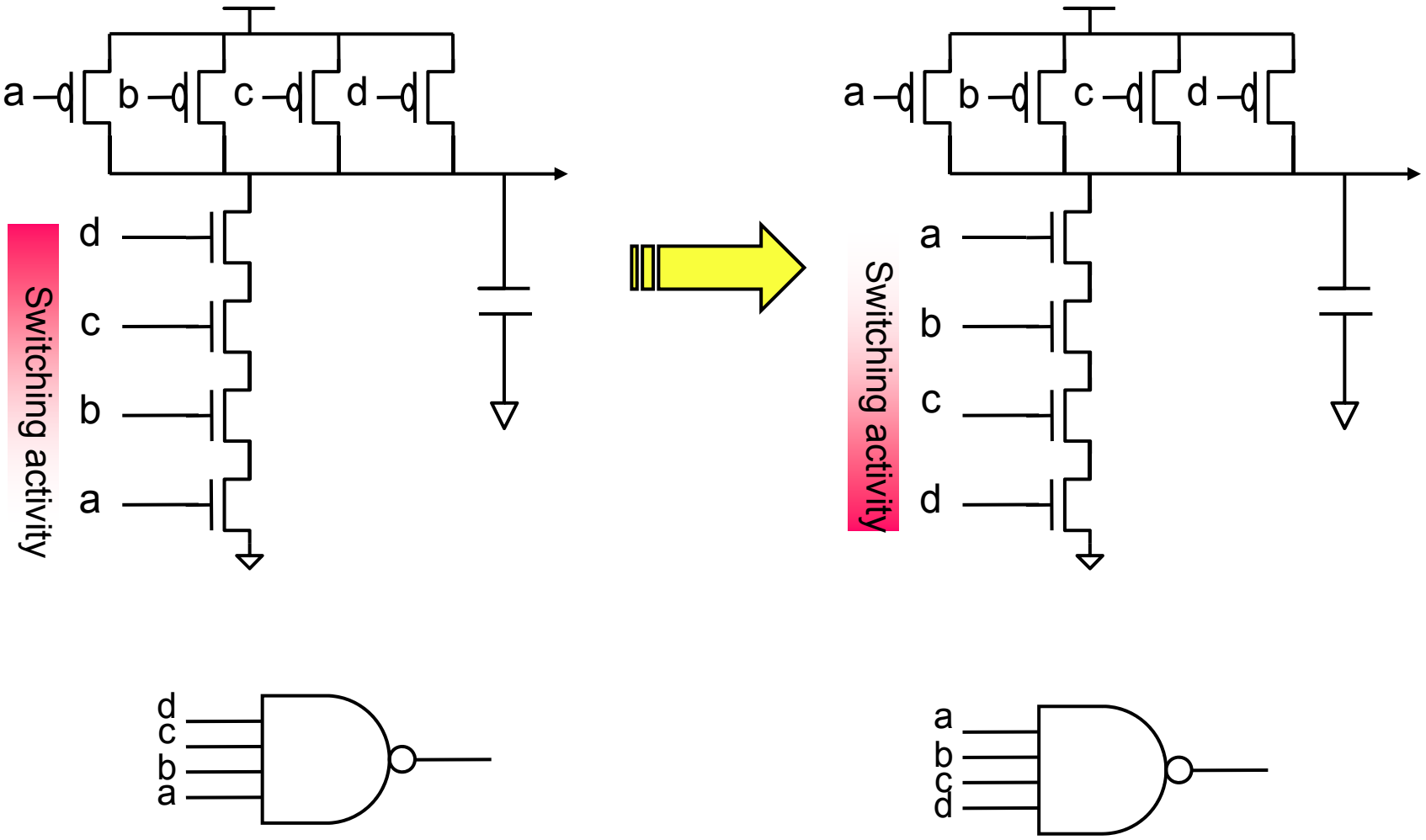
- The first step in technology mapping is to decompose each logic function into two-input gates
- The objective of this decomposition is to minimizing the total power dissipation by reducing the total switching activity



Gate-Level Design – *Phase Assignment*

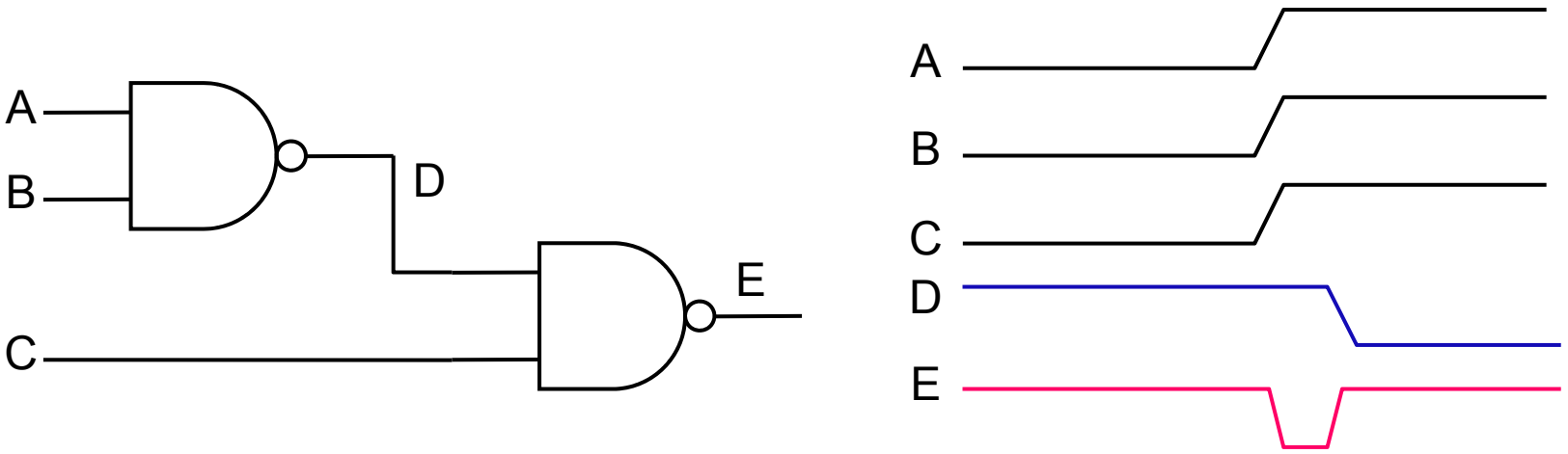


Gate-Level Design – *Pin Swapping*



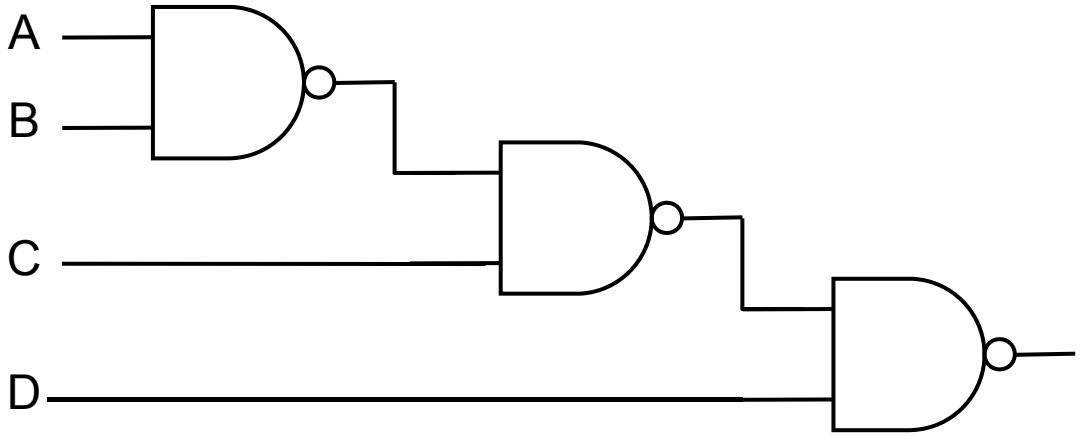
Gate-Level Design – *Glitching Power*

- Glitches
  - spurious transitions due to imbalanced path delays
- A design has more balanced delay paths
  - has fewer glitches, and thus has less power dissipation
- Note that there will be no glitches in a dynamic CMOS logic

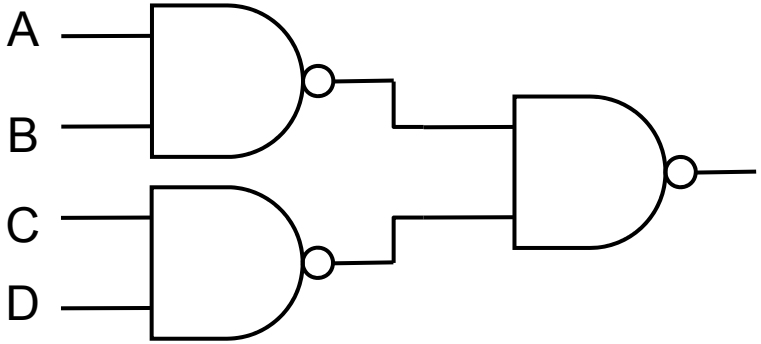


Gate-Level Design – *Glitching Power*

- A chain structure has more glitches
- A tree structure has fewer glitches

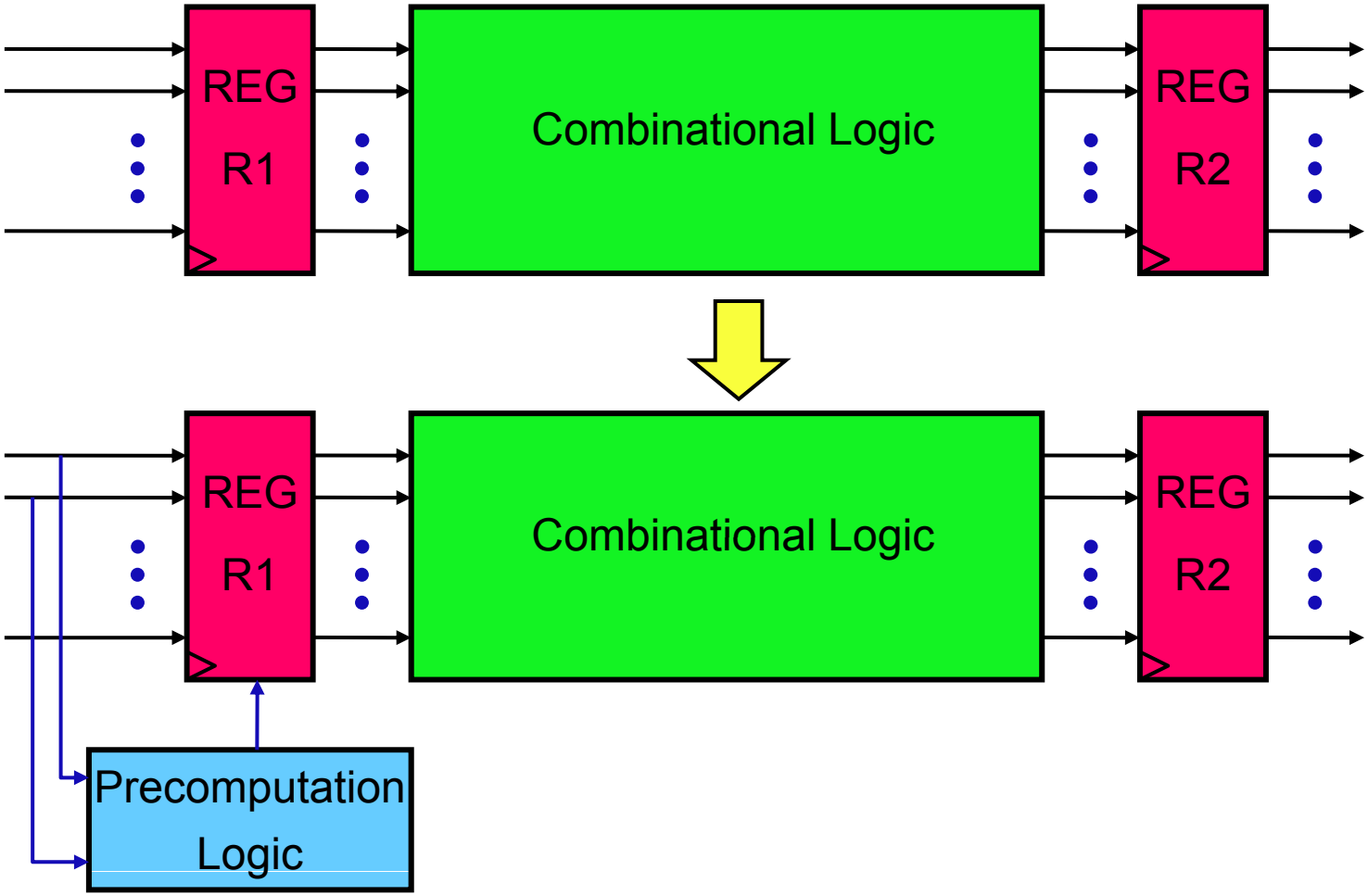


Chain structure

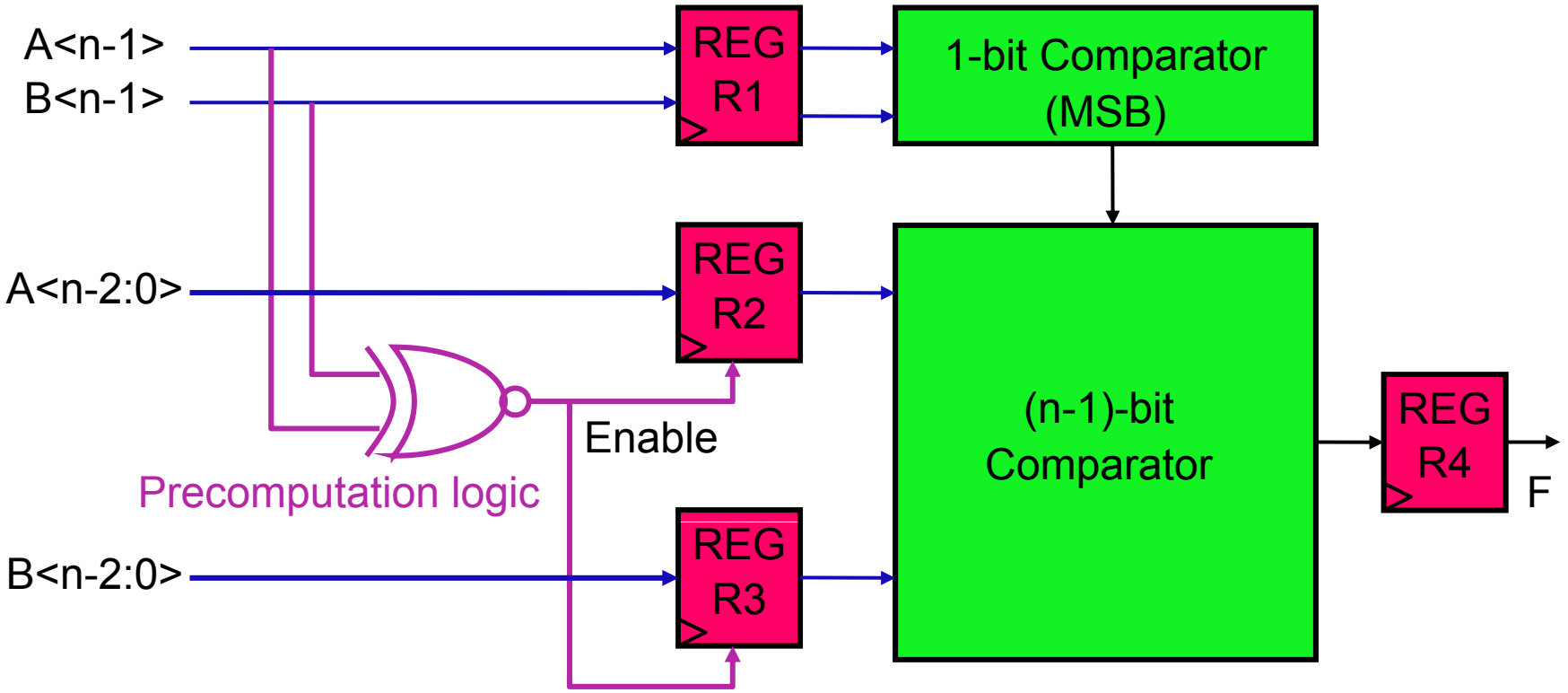


Tree structure

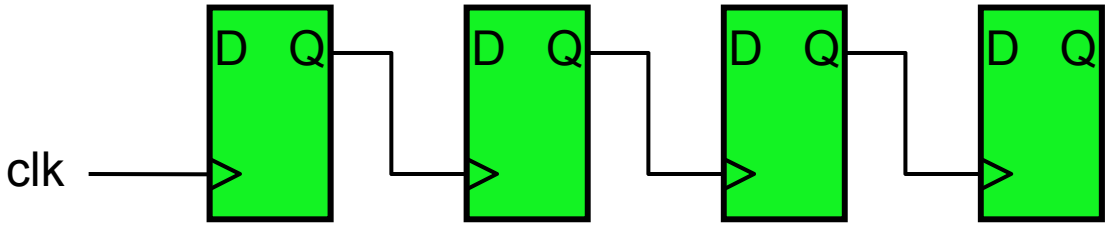
Gate-Level Design – *Precomputation*



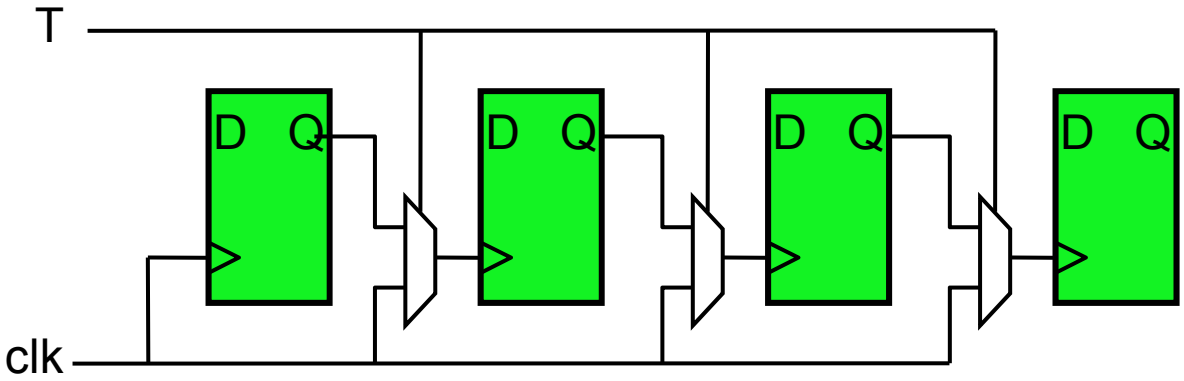
Gate-Level Design – *Precomputation*



Gate-Level Design – *Gating Clock*

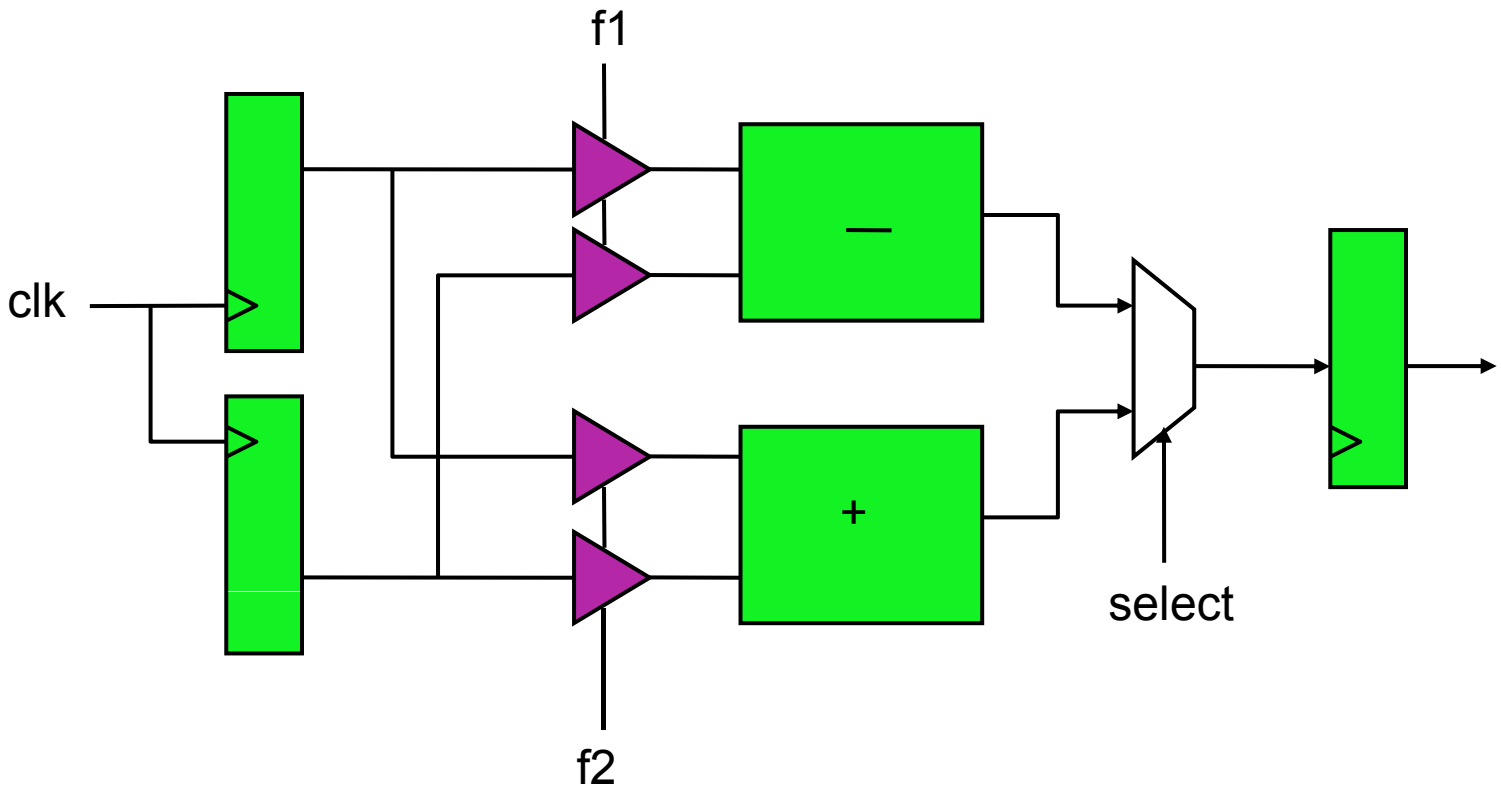


Fail DFT rule checking

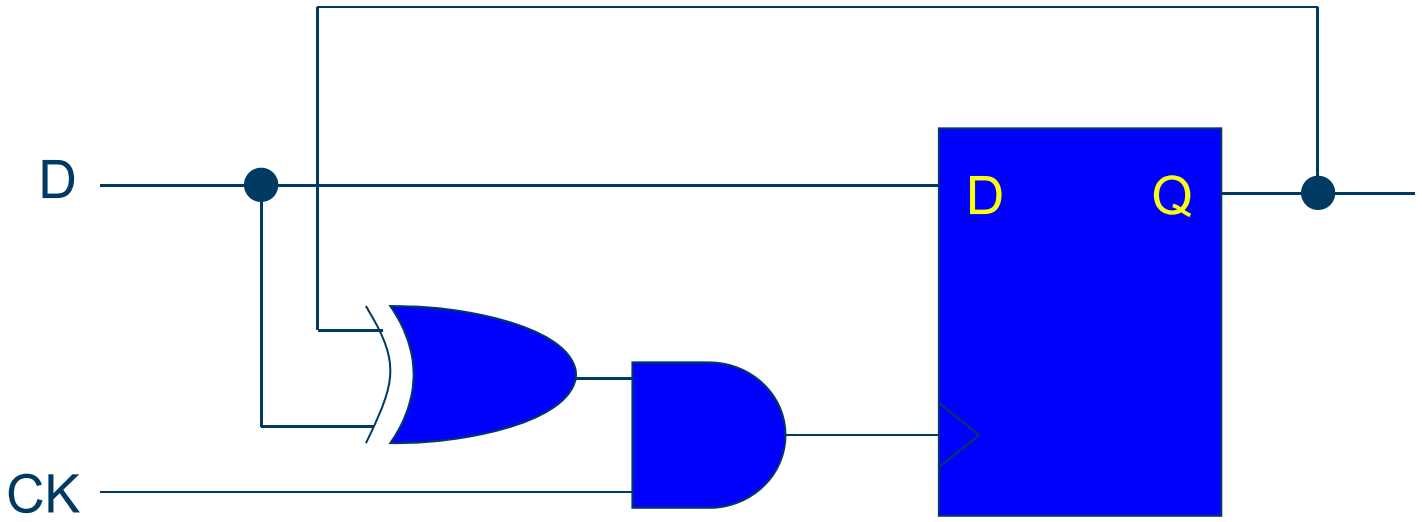


Add control pin to solve DFT violation problem

Gate-Level Design – *Input Gating*

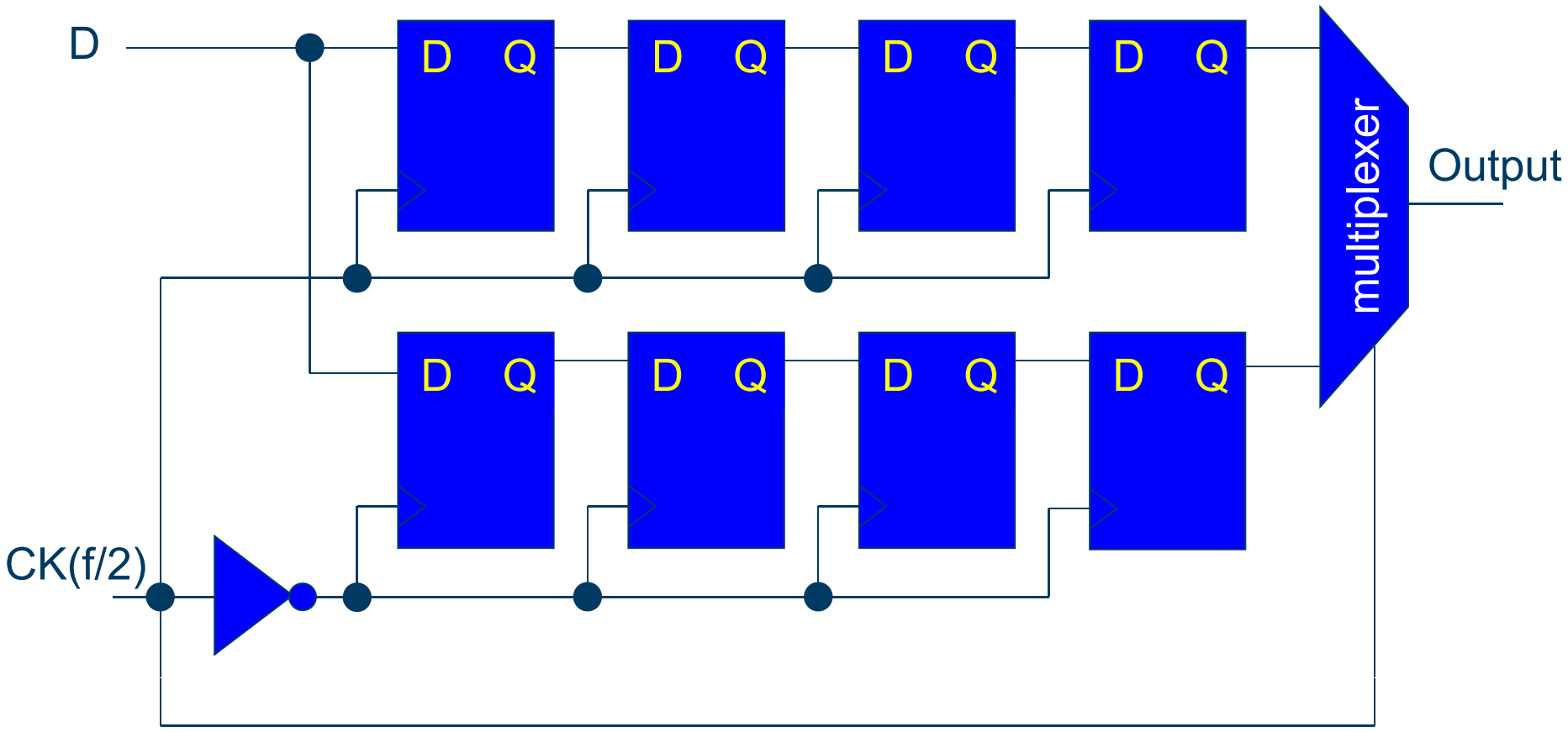


Clock-Gating in Low-Power Flip-Flop



Source: Prof. V. D. Agrawal

Reduced-Power Shift Register



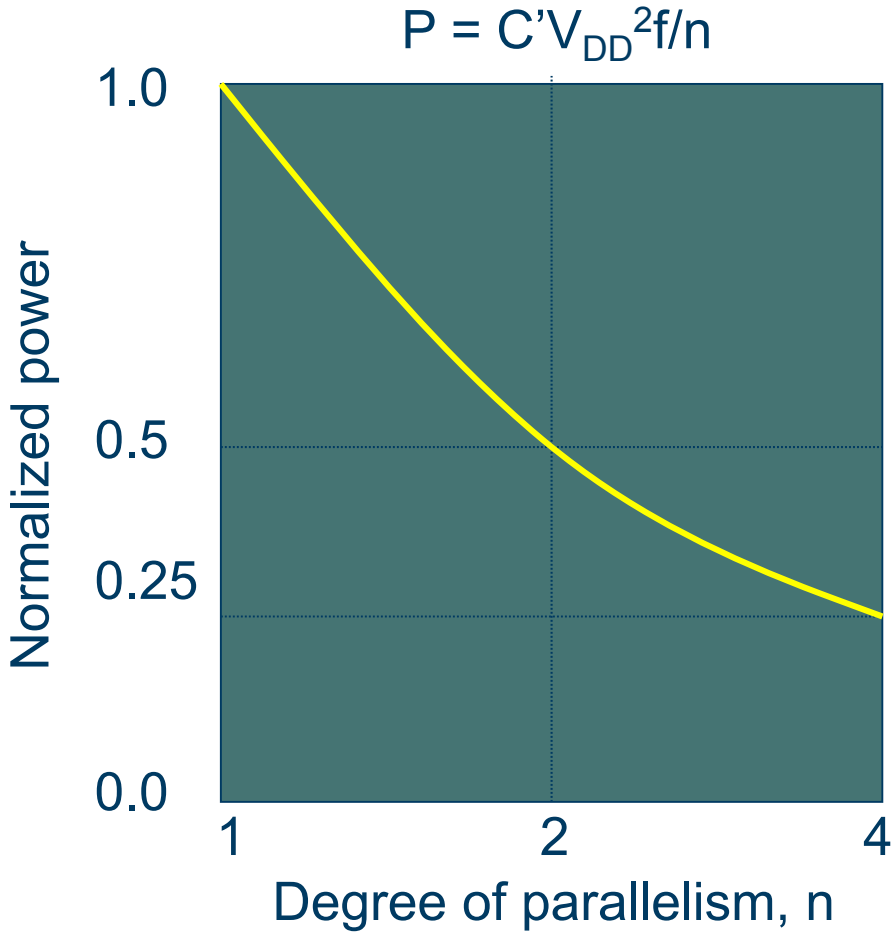
Flip-flops are operated at full voltage and half the clock frequency.

Source: Prof. V. D. Agrawal

Power Consumption of Shift Register

16-bit shift register, 2μ CMOS

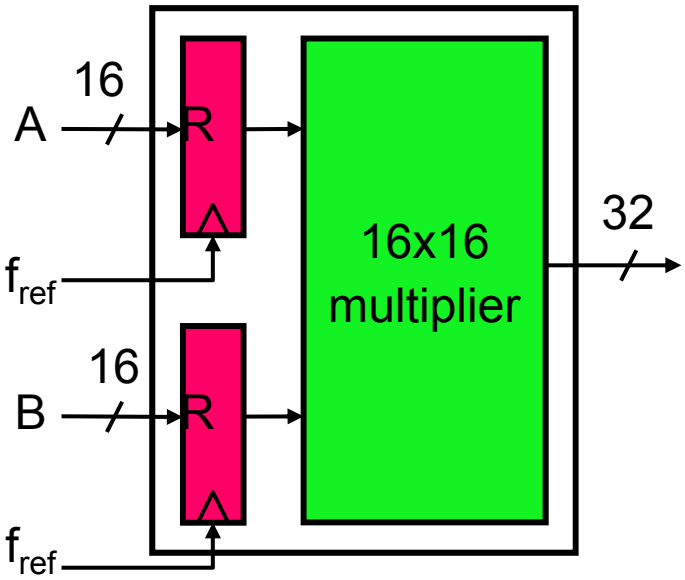
Deg. Of parallelism	Freq (MHz)	Power (μW)
1	33.0	1535
2	16.5	887
4	8.25	738



C. Piguet, "Circuit and Logic Level Design," pages 103-133 in W. Nebel and J. Mermet (ed.), *Low Power Design in Deep Submicron Electronics*, Springer, 1997.

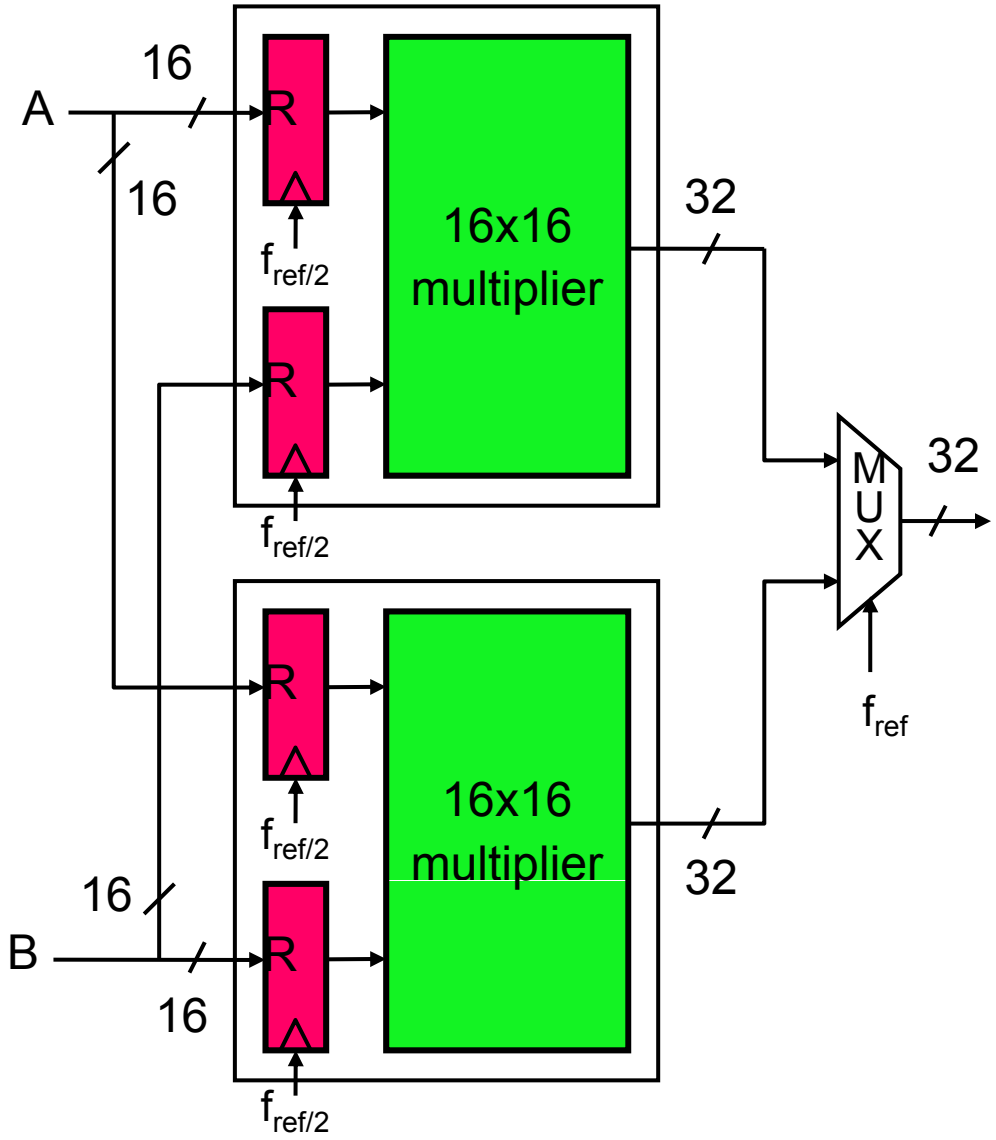
Source: Prof. V. D. Agrawal

Architecture-Level Design – *Parallelism*



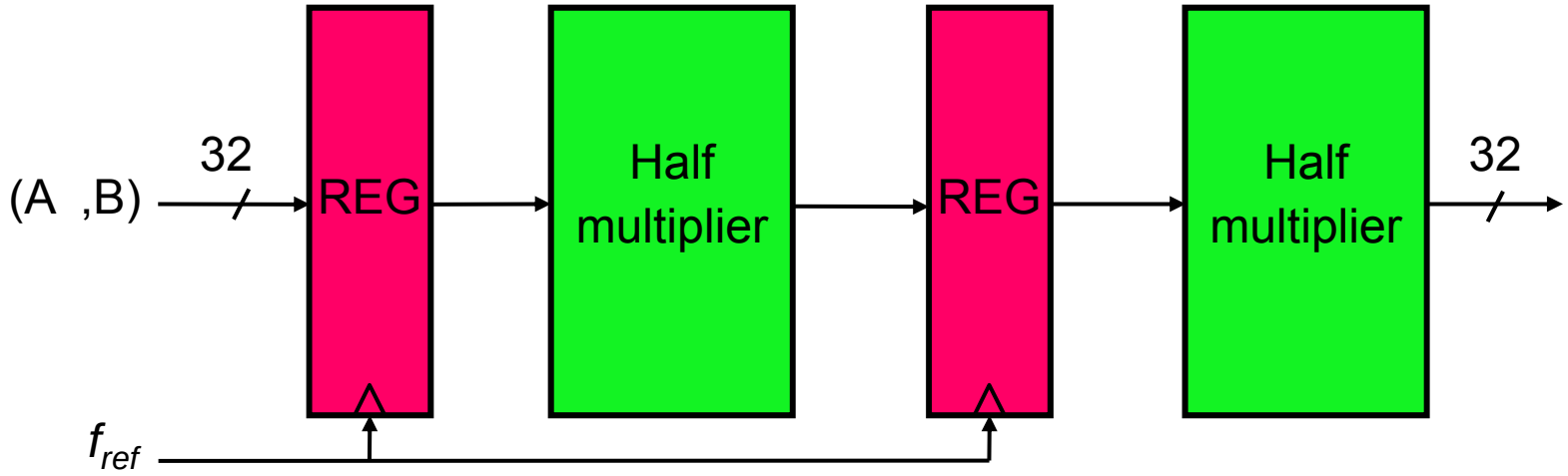
Assume that With the same 16x16 multiplier, the power supply can be reduced from V_{ref} to $V_{ref}/1.83$.

$$P_{parallel} = 2.2C_{ref}\left(\frac{V_{ref}}{1.83}\right)^2 \frac{f_{ref}}{2} = 0.33P_{ref}$$



Architecture-Level Design – *Pipelining*

The hardware between the pipeline stages is reduced then the reference voltage V_{ref} can be reduced to V_{new} to maintain the same worst case delay. For example, let a 50MHz multiplier is broken into two equal parts as shown below. The delay between the pipeline stages can be remained at 50MHz when the voltage V_{new} is equal to $V_{ref}/1.83$

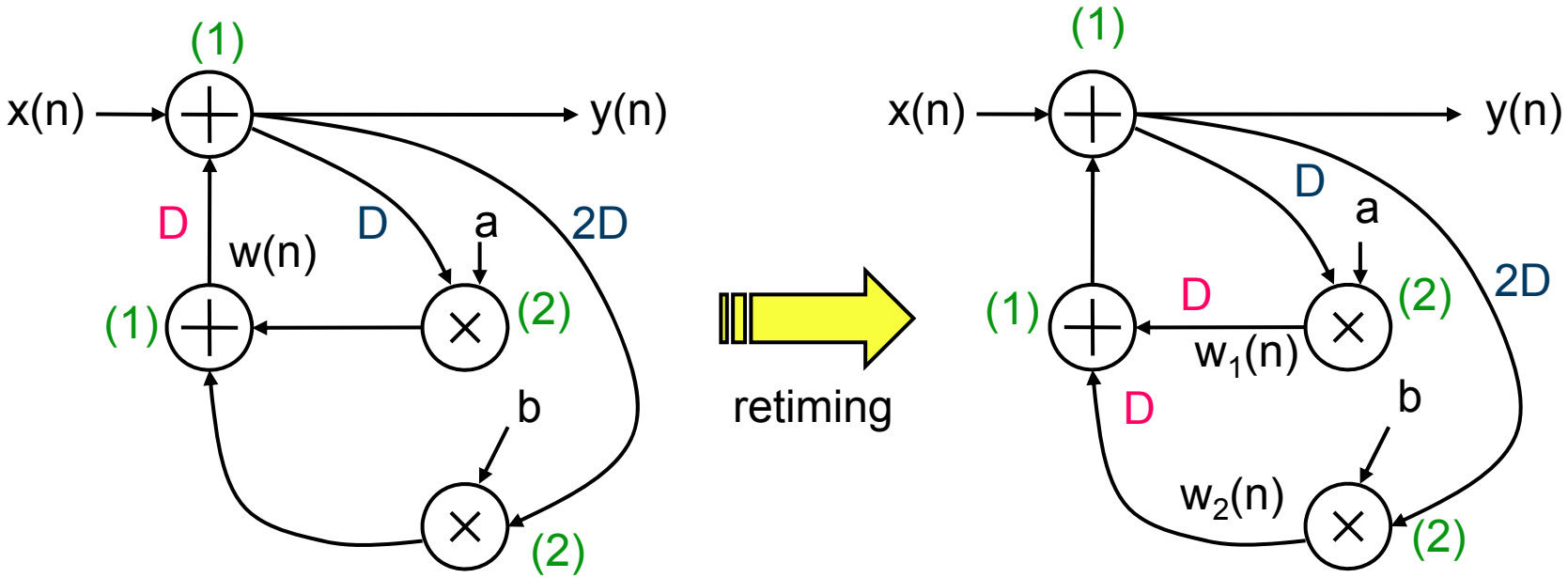


$$P_{pipeline} = 1.2C_{ref}\left(\frac{V_{ref}}{1.83}\right)^2 f_{ref} = 0.36P_{ref}$$

Architecture-Level Design – *Retiming*

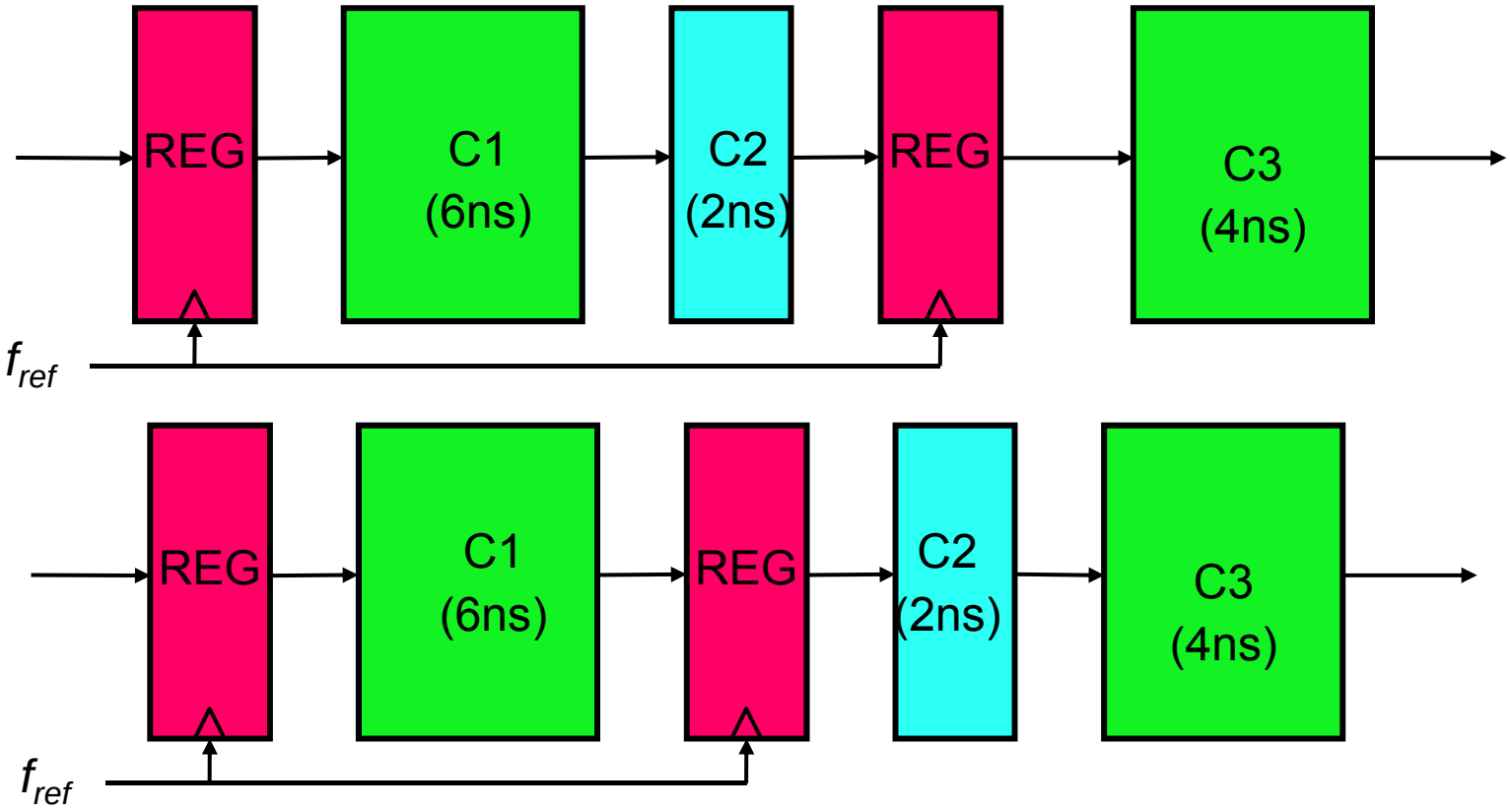
Retiming is a transformation technique used to change the locations of delay elements in a circuit without affecting the input/output characteristics of the circuit.

Two versions of an IIR filter.

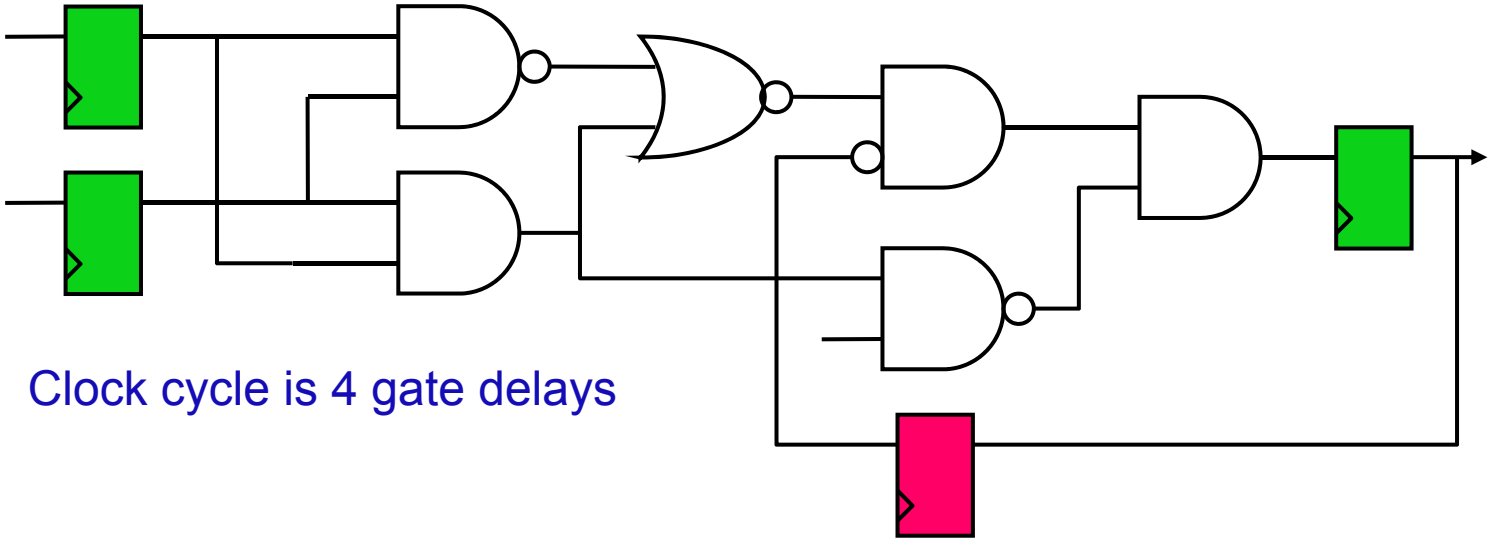


Architecture-Level Design – *Retiming*

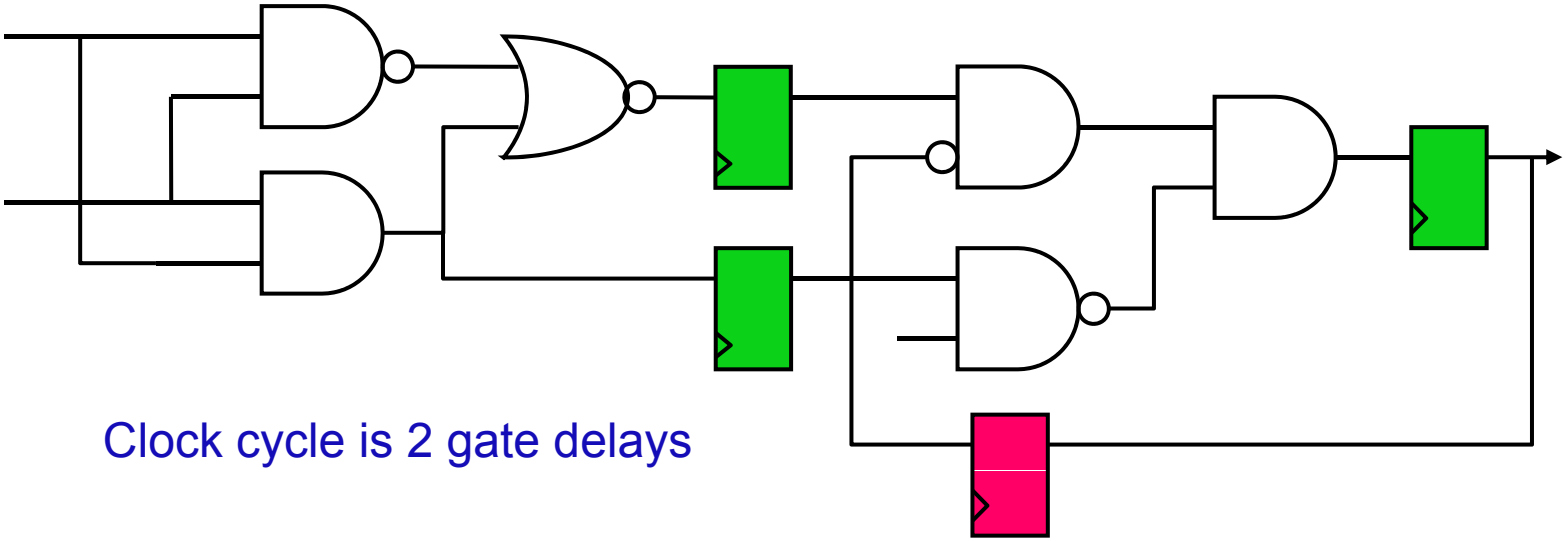
Retiming for pipeline design



Architecture-Level Design – *Retiming*

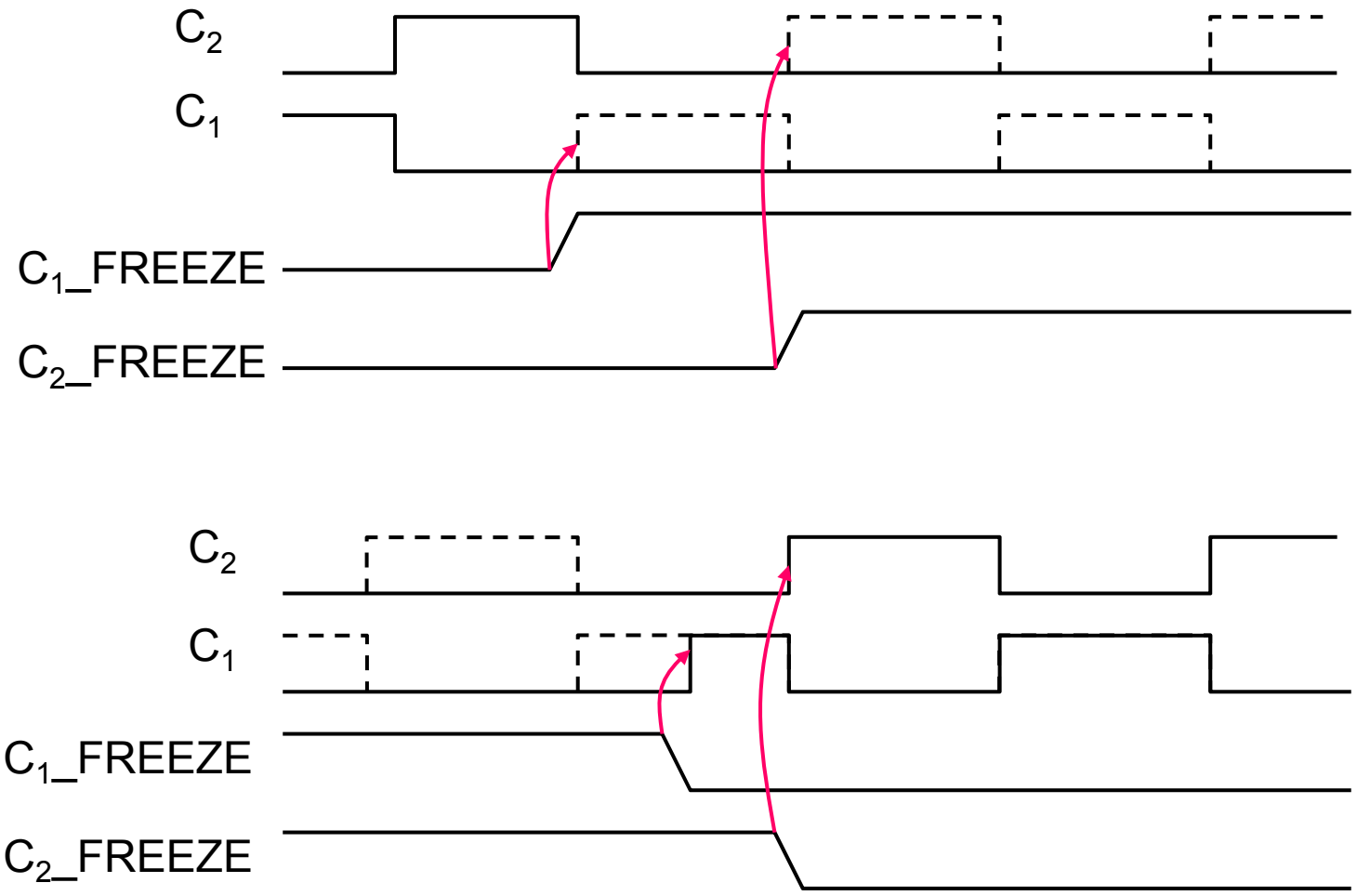


Clock cycle is 4 gate delays



Clock cycle is 2 gate delays

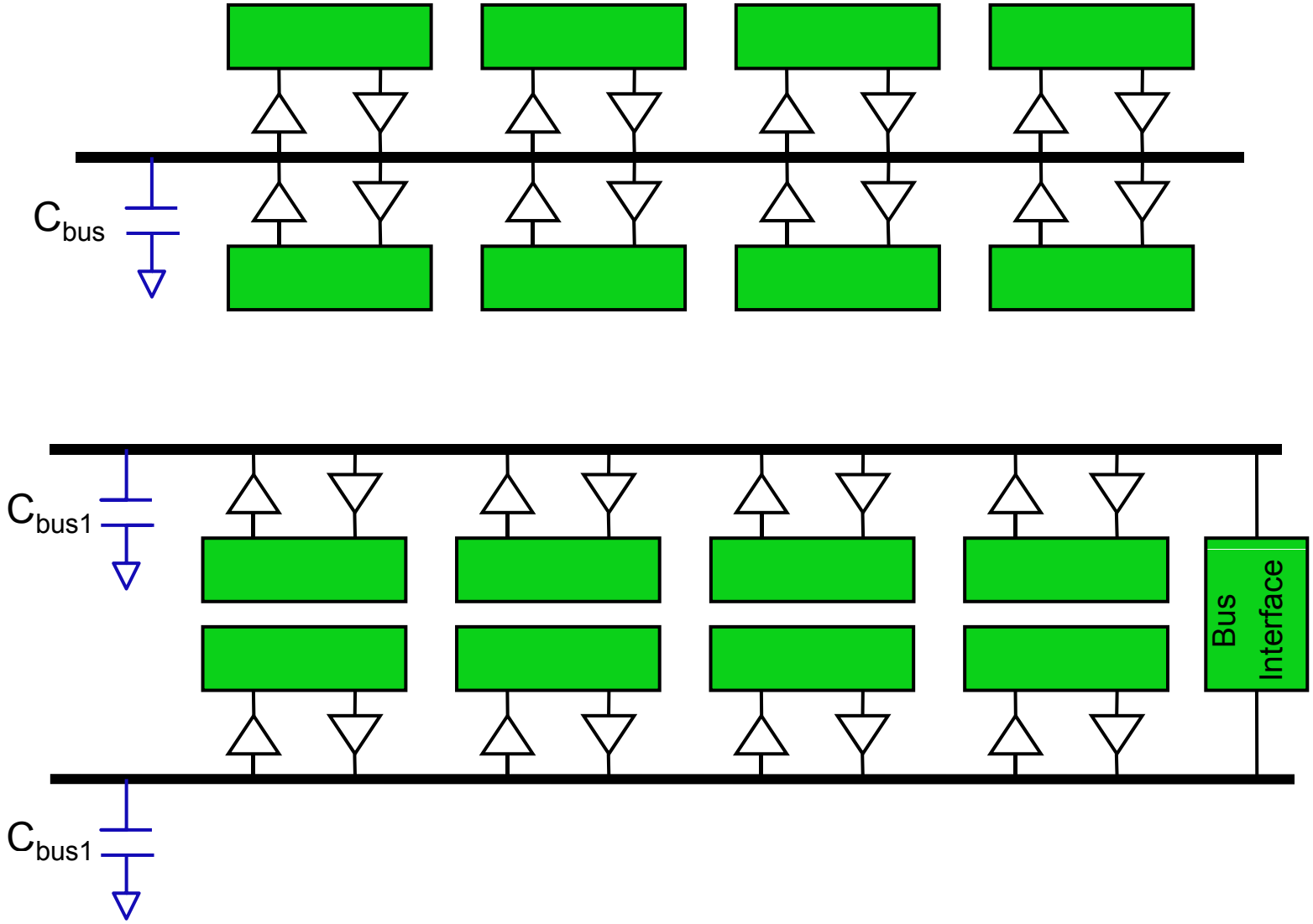
Architecture-Level Design – *Power Management*



Architecture-Level Design – *Bus Segmentation*

- Avoid the sharing of resources
  - Reduce the switched capacitance
- For example: a global system bus
  - A single shared bus is connected to all modules, this structure results in a large bus capacitance due to
    - The large number of drivers and receivers sharing the same bus
    - The parasitic capacitance of the long bus line
- A segmented bus structure
  - Switched capacitance during each bus access is significantly reduced
  - Overall routing area may be increased

Architecture-Level Design – *Bus Segmentation*



Algorithmic-Level Design – $f_{activity}$ Reduction

Minimization the switching activity, at high level, is one way to reduce the power dissipation of digital processors.

One method to minimize the switching signals, at the algorithmic level, is to use an appropriate coding for the signals rather than straight binary code.

The table shown below shows a comparison of 3-bit representation of the binary and Gray codes.

Binary Code	Gray Code	Decimal Equivalent
000	000	0
001	001	1
010	011	2
011	010	3
100	110	4
101	111	5
110	101	6
111	100	7

State Encoding for a Counter

- Two-bit binary counter:
  - State sequence, 00 → 01 → 10 → 11 → 00
  - Six bit transitions in four clock cycles
  - 6/4 = 1.5 transitions per clock
- Two-bit Gray-code counter
  - State sequence, 00 → 01 → 11 → 10 → 00
  - Four bit transitions in four clock cycles
  - 4/4 = 1.0 transition per clock
- Gray-code counter is more power efficient.

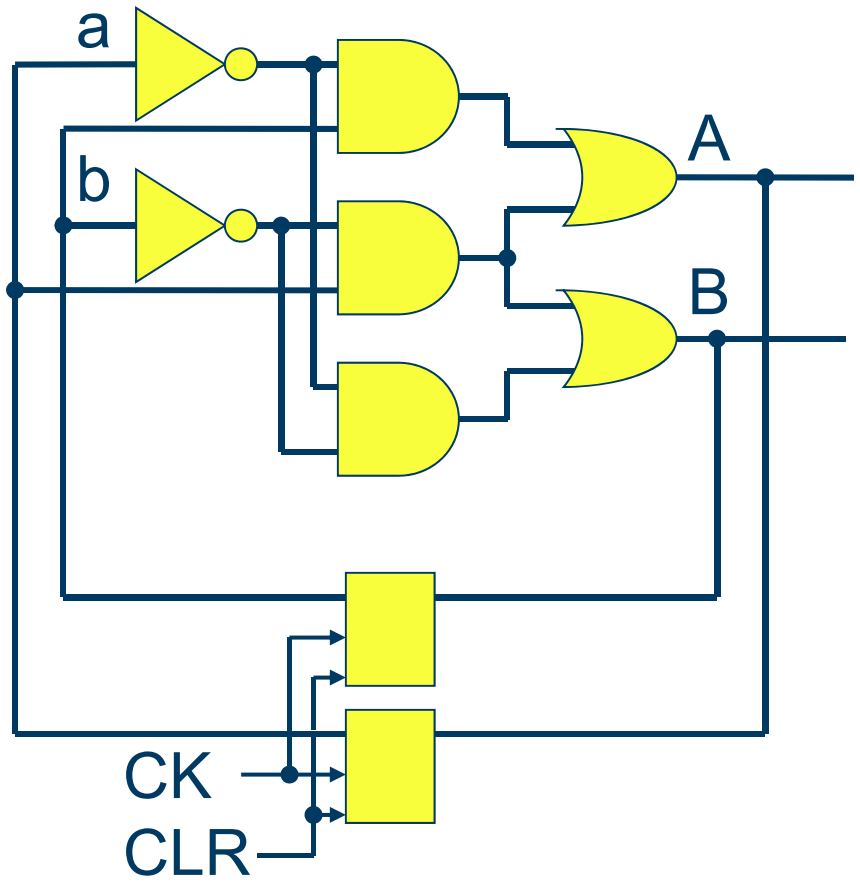
G. K. Yeap, *Practical Low Power Digital VLSI Design*, Boston: Kluwer Academic Publishers (now Springer), 1998.

Source: Prof. V. D. Agrawal

Binary Counter: Original Encoding

Present state		Next state	
a	b	A	B
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

A = a'b + ab'
B = a'b' + ab'



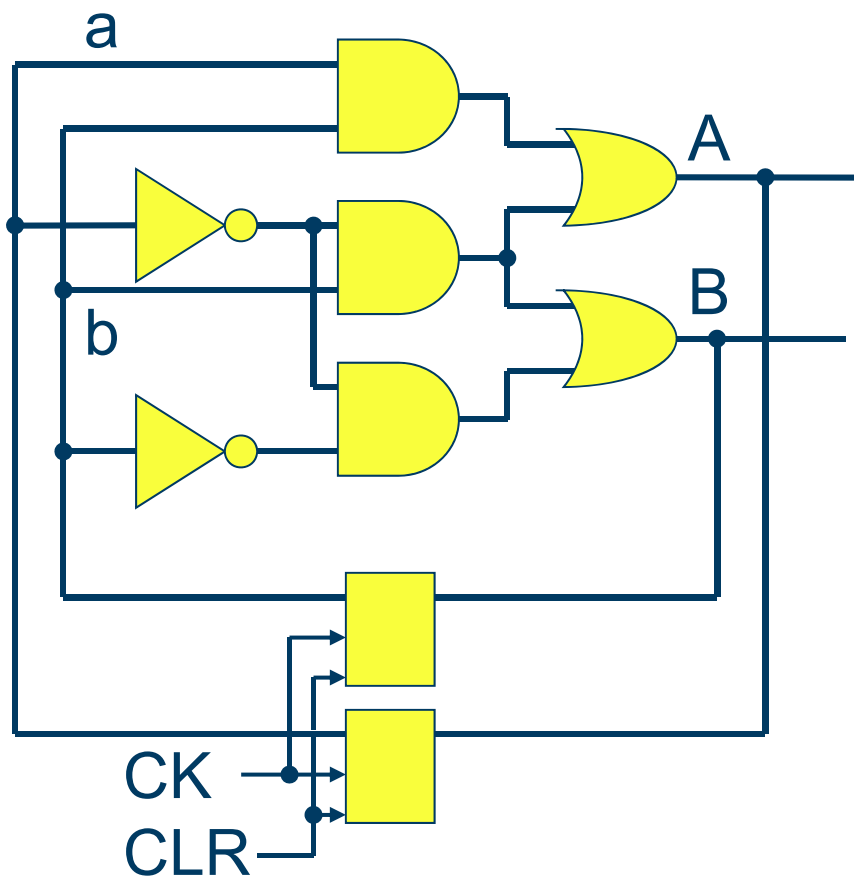
Source: Prof. V. D. Agrawal

Binary Counter: Gray Encoding

Present state		Next state	
a	b	A	B
0	0	0	1
0	1	1	1
1	0	0	0
1	1	1	0

A = a'b + ab

B = a'b' + a'b



Source: Prof. V. D. Agrawal

Three-Bit Counters

Binary		Gray-code	
State	No. of toggles	State	No. of toggles
000	-	000	-
001	1	001	1
010	2	011	1
011	1	010	1
100	3	110	1
101	1	111	1
110	2	101	1
111	1	100	1
000	3	000	1
Av. Transitions/clock = 1.75		Av. Transitions/clock = 1	

Source: Prof. V. D. Agrawal

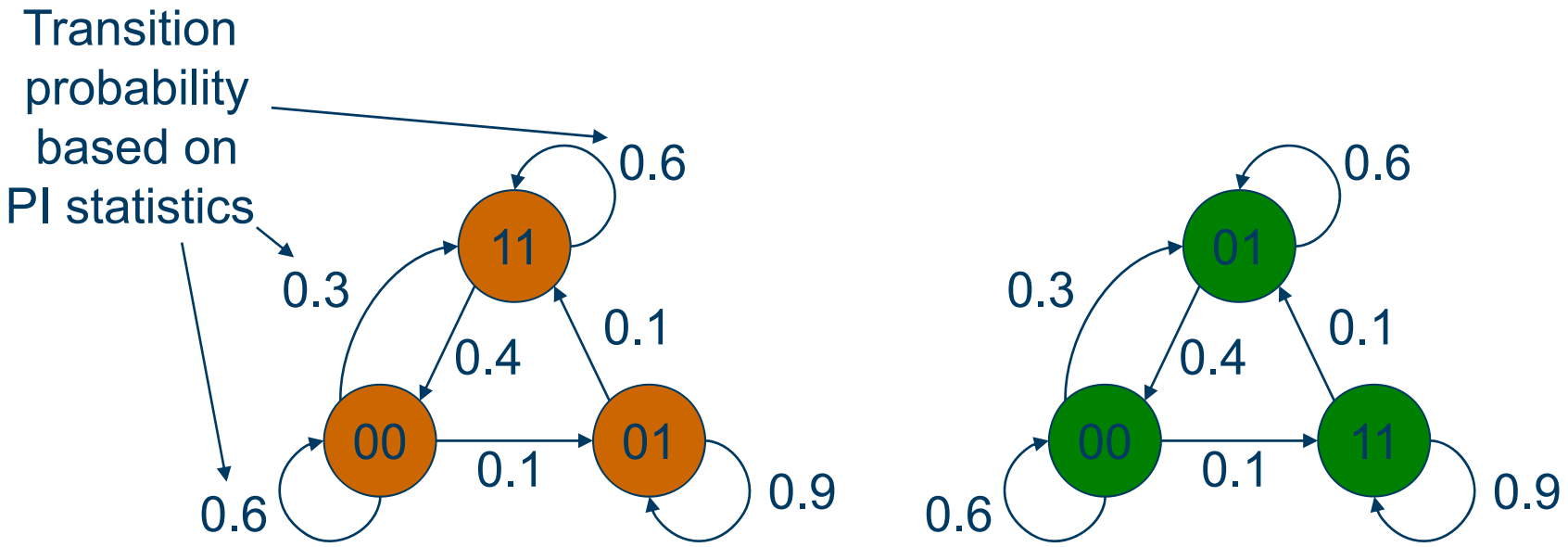
N-Bit Counter: Toggles in Counting Cycle

- Binary counter: $T(\text{binary}) = 2(2^N - 1)$
- Gray-code counter: $T(\text{gray}) = 2^N$
- $T(\text{gray})/T(\text{binary}) = 2^{N-1}/(2^N - 1) \rightarrow 0.5$

Bits	T(binary)	T(gray)	T(gray)/T(binary)
1	2	2	1.0
2	6	4	0.6667
3	14	8	0.5714
4	30	16	0.5333
5	62	32	0.5161
6	126	64	0.5079
∞	-	-	0.5000

Source: Prof. V. D. Agrawal

FSM State Encoding



Expected number of state-bit transitions:

2(0.3+0.4) + 1(0.1+0.1) = 1.6

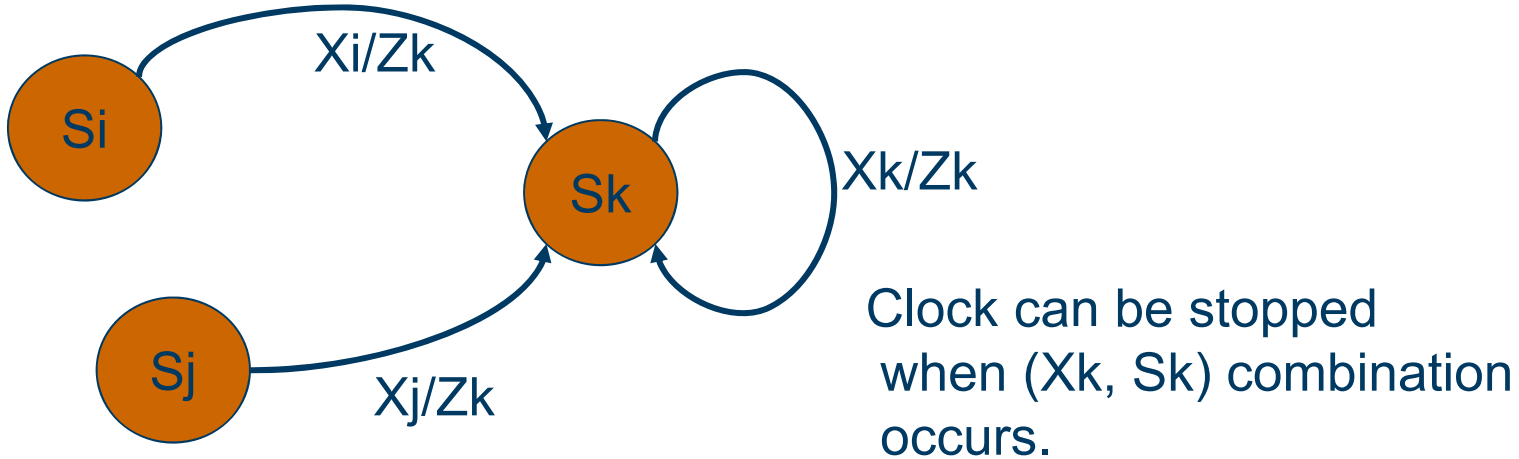
1(0.3+0.4+0.1) + 2(0.1) = 1.0

State encoding can be selected using a power-based cost function.

Source: Prof. V. D. Agrawal

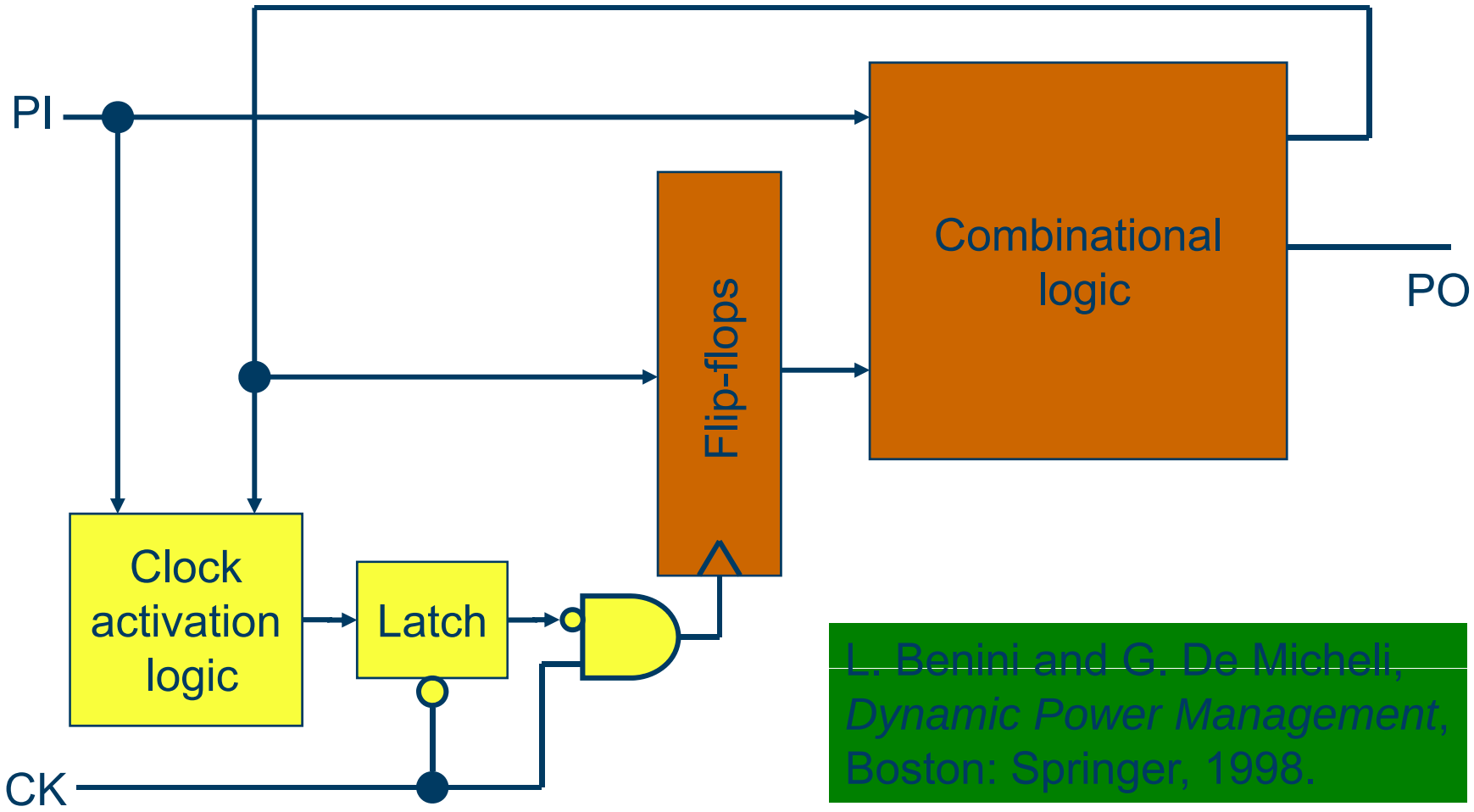
FSM: Clock-Gating

- Moore machine: Outputs depend only on the state variables.
  - If a state has a self-loop in the state transition graph (STG), then clock can be stopped whenever a self-loop is to be executed.



Source: Prof. V. D. Agrawal

Clock-Gating in Moore FSM

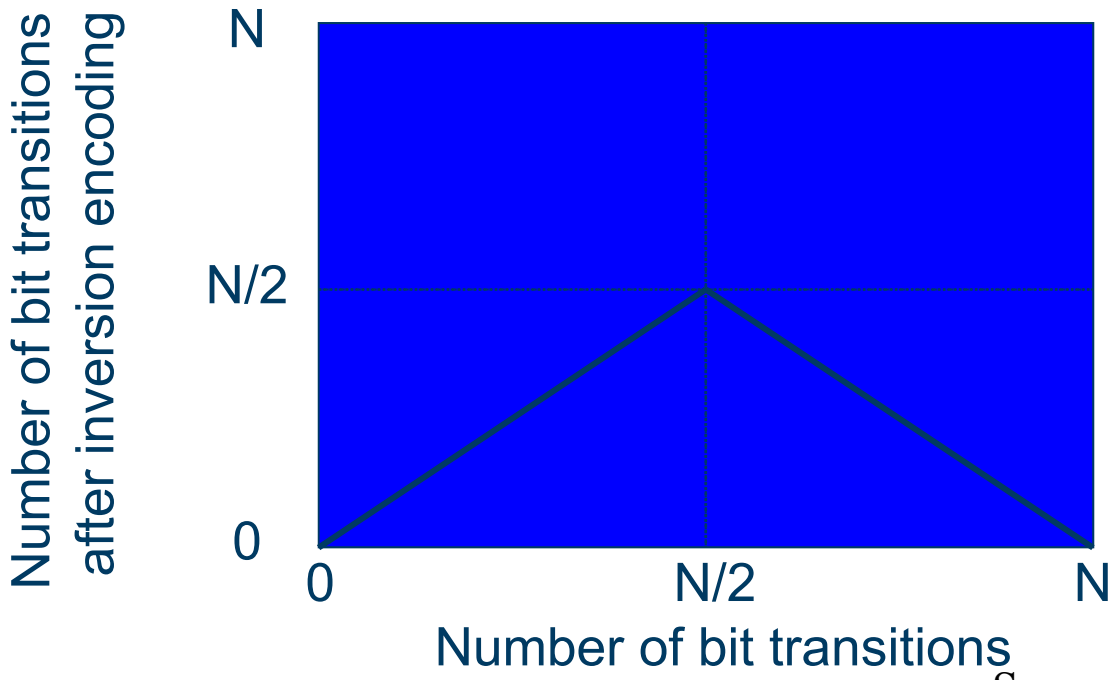


L. Benini and G. De Micheli, *Dynamic Power Management*, Boston: Springer, 1998.

Source: Prof. V. D. Agrawal

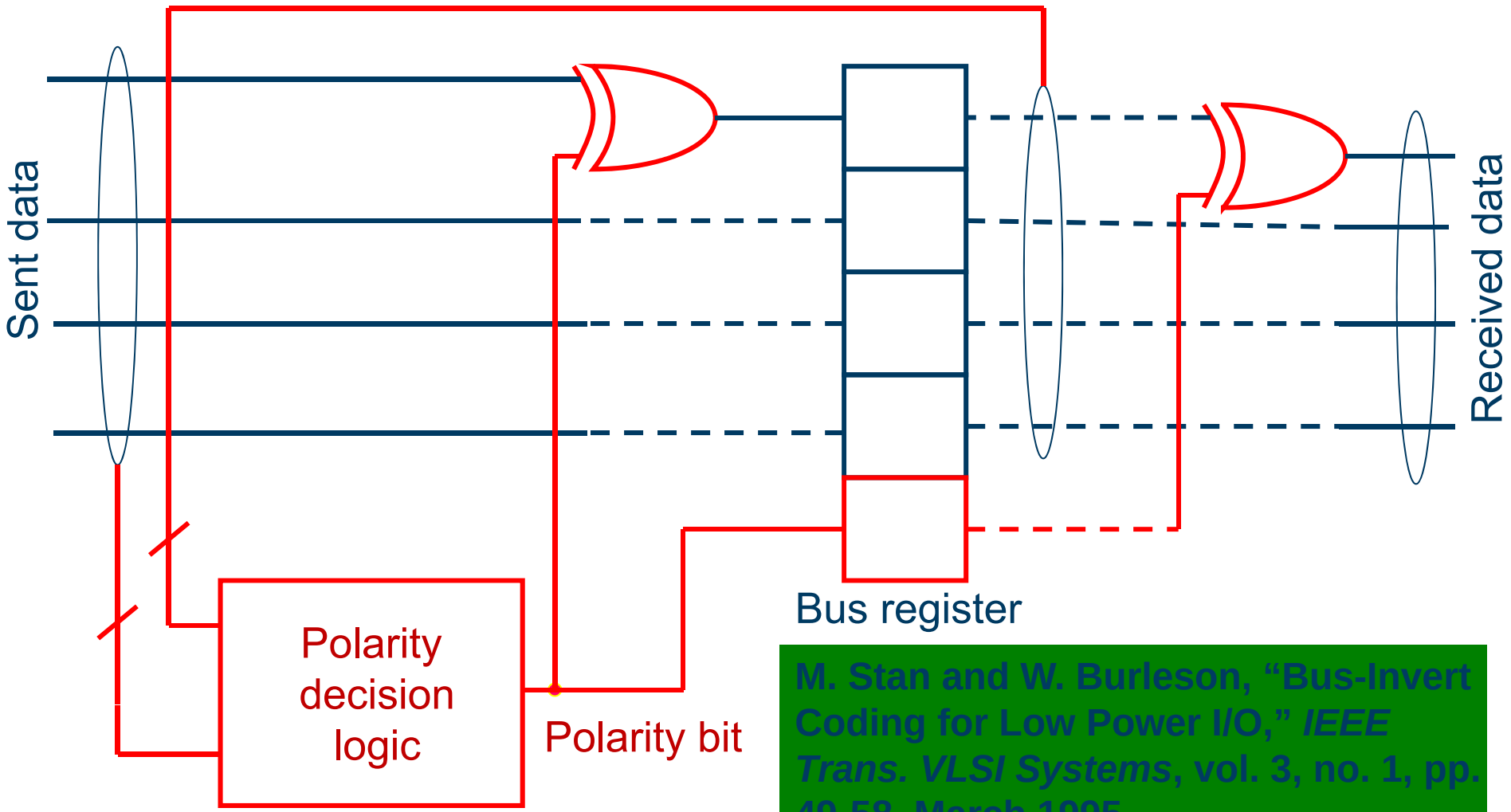
Bus Encoding for Reduced Power

- Example: Four bit bus
  - 0000 → 1110 has three transitions.
  - If bits of second pattern are inverted, then 0000 → 0001 will have only one transition.
- Bit-inversion encoding for N-bit bus:



Source: Prof. V. D. Agrawal

Bus-Inversion Encoding Logic



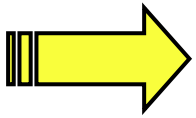
M. Stan and W. Burleson, "Bus-Invert Coding for Low Power I/O," *IEEE Trans. VLSI Systems*, vol. 3, no. 1, pp. 49-58, March 1995.

Source: Prof. V. D. Agrawal

RTL-Level Design – *Signal Gating*

Simple Decoder

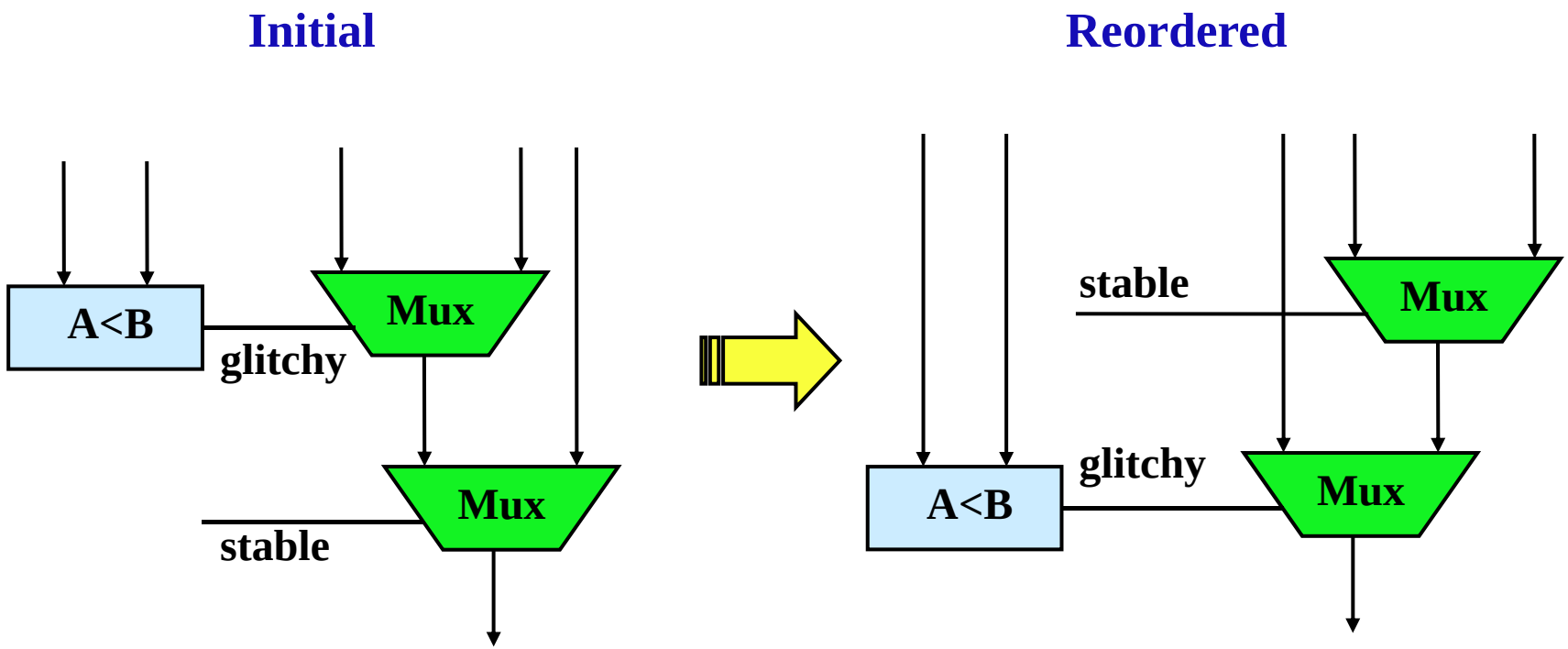
```
module decoder (a, sel);
    input  [1:0[ a;
    ouput [3:0] sel;
    reg    [3:0] sel;
    always @(a) begin
        case (a)
            2'b00: sel=4'b0001;
            2'b01: sel=4'b0010;
            2'b10: sel=4'b0100;
            2'b11: sel=4'b1000;
        endcase
    end
endmodule
```



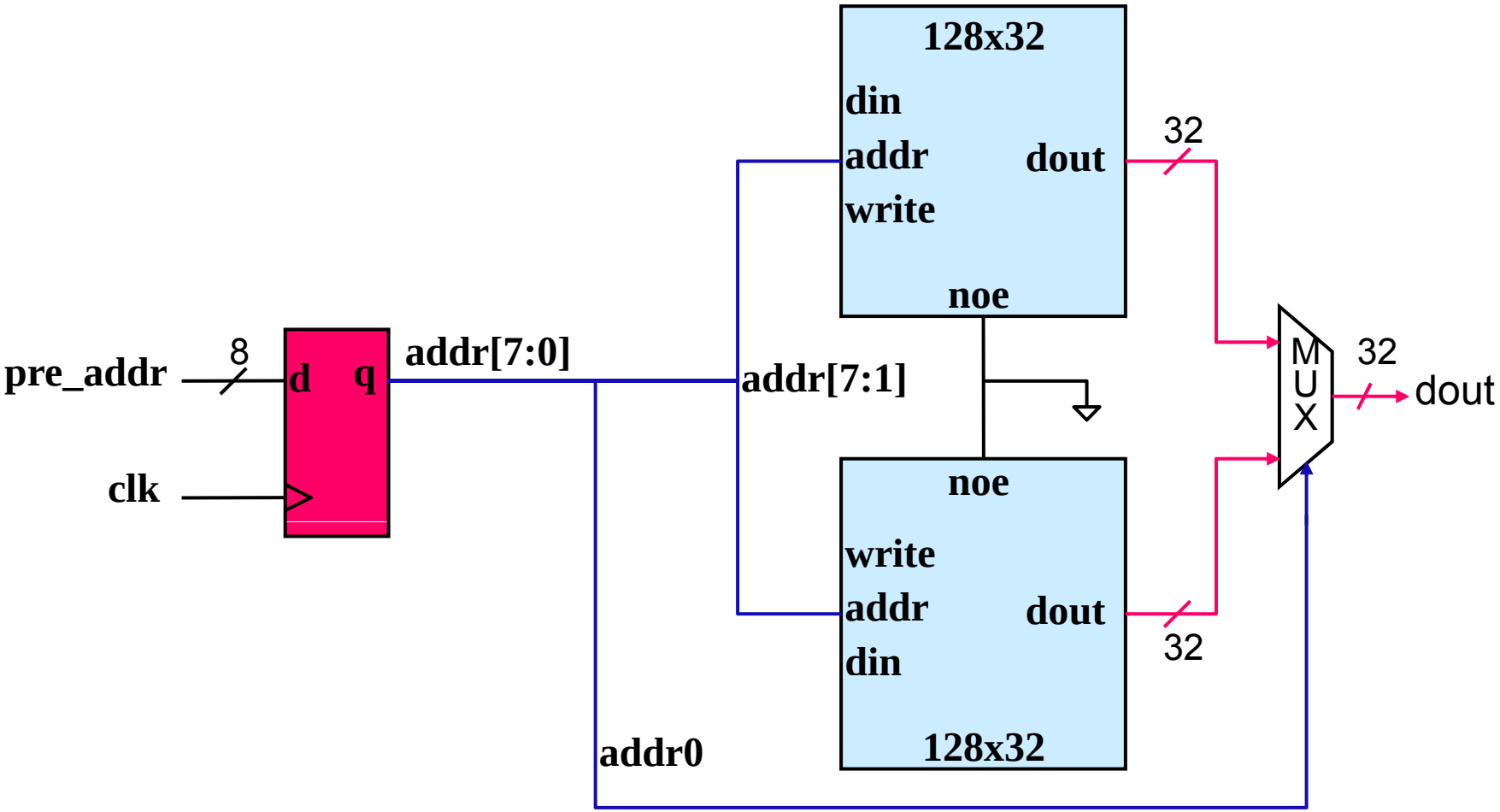
Decoder with enable

```
module decoder (en,a, sel);
    input          en;
    input  [1:0[ a;
    ouput [3:0] sel;
    reg    [3:0] sel;
    always @({en,a}) begin
        case ({en,a})
            3'b100: sel=4'b0001;
            3'b101: sel=4'b0010;
            3'b110: sel=4'b0100;
            3'b111: sel=4'b1000;
            default: sel=4'b0000;
        endcase
    end
endmodule
```

RTL-Level Design – *Datapath Reordering*

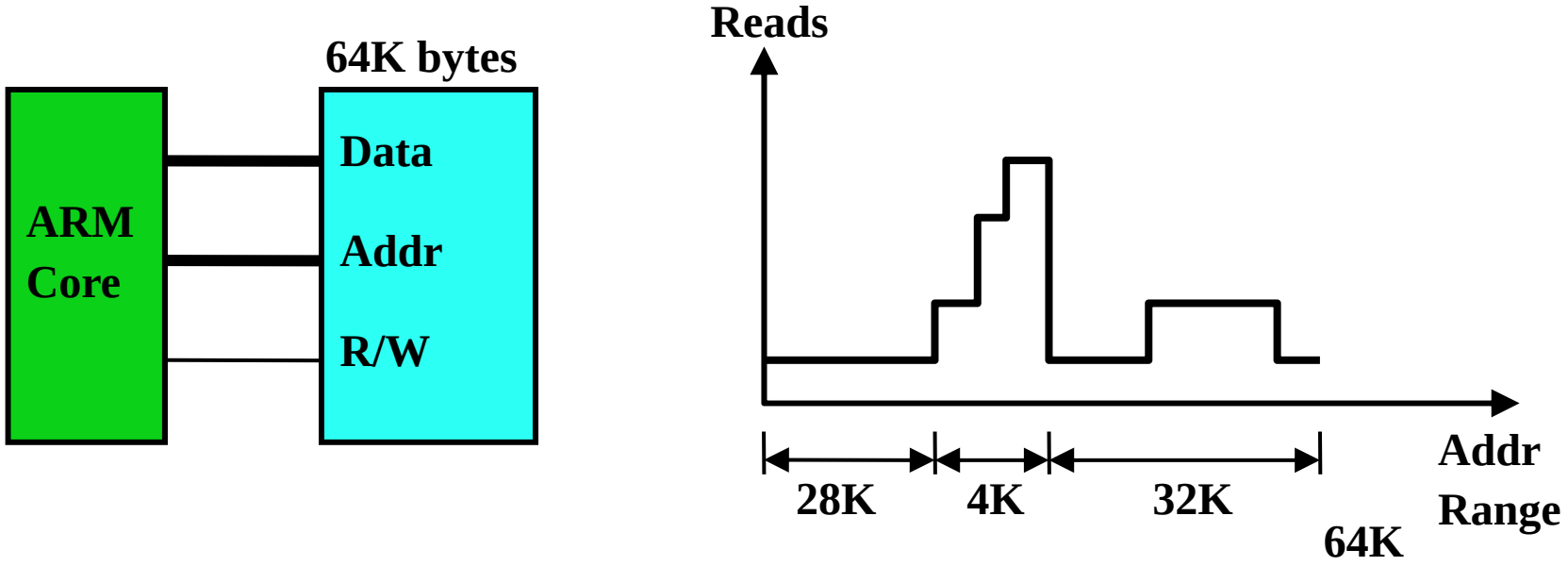


RTL-Level Design – *Memory Partition*



RTL-Level Design – *Memory Partition*

- Application-driven memory partition



RTL-Level Design – *Memory Partition*

- A power-optimal partitioned memory organization