

EEC 216 Lecture #2: Metrics and Logic Level Power Estimation

**Rajeevan Amirtharajah
University of California, Davis**

Outline

- Announcements
- **Aside: Environmental Impact of Electronics**
- Finish Lecture 1 Topics
- Intersignal Correlations
- Glitches
- High Level Power Estimation
- Behavioral Level Power Estimation
- Architectural Level Power Estimation

Power Consumption at the System Level

- **500 MHz Pentium III with 17 in. Monitor: 150-200 W**
- **Server: 300 W**
- **Mainframe: 10-20 kW**
- **Author's home office: 2 CPUs, 2 monitors, laptop, 3 printers, scanner, plus misc. peripherals**
 - Nameplate rating: 2.4 kW
 - Max power (incl. peripherals): 700 W
 - Typical usage: 150-170 W
 - Average over 10 days: 77 W (9% of total consumption)

From B. Hayes, "The Computer and the Dynamo", *American Scientist*, Sep-Oct 2001

Power Consumption in the US (2001)

- **All office equipment consumes 74 TW-hours / year (about 2% US total)**
- **Adding telecoms increases total to 3.2 %**
- **Power down and sleep modes could save 23-40 TWh**
- **Technology has dramatically improved power cost of computing**
 - ENIAC (1940s): 18,000 vacuum tubes, 174 kW, roughly 10 W / tube
 - Today's microprocessors: 100 M transistors, 100 W, roughly 1 μ W / transistor (would draw 10 GW if no change from 1940s)

From B. Hayes, "The Computer and the Dynamo", *American Scientist*, Sep-Oct 2001

Worldwide Carbon Emissions

- **Information & communications technology accounts for 2% of global CO₂ emissions in 2008**
 - 49% from PCs and printers, 37% from telecoms networks and devices, 14% from data centers
 - Comparable to aviation industry
- **In 2020, projected 1.43B tons CO₂ equivalent**
 - 57% from PCs and printers, 25% from telecoms networks and devices, 18% from data centers
 - ...but, can enable ~8B tons emissions reductions through increased efficiency in other areas, e.g. logistics, travel

From The Economist, “More Silicon, Less Carbon”, *The World in 2009*, Nov. 19, 2008

Outline

- Announcements
- Aside: Environmental Impact of Electronics
- **Finish Lecture 1 Topics**
- Intersignal Correlations
- Glitches
- High Level Power Estimation
- Behavioral Level Power Estimation
- Architectural Level Power Estimation

Outline

- Announcements
- Aside: Environmental Impact of Electronics
- Finish Lecture 1 Topics
- **Intersignal Correlations**
- Glitches
- High Level Power Estimation
- Behavioral Level Power Estimation
- Architectural Level Power Estimation

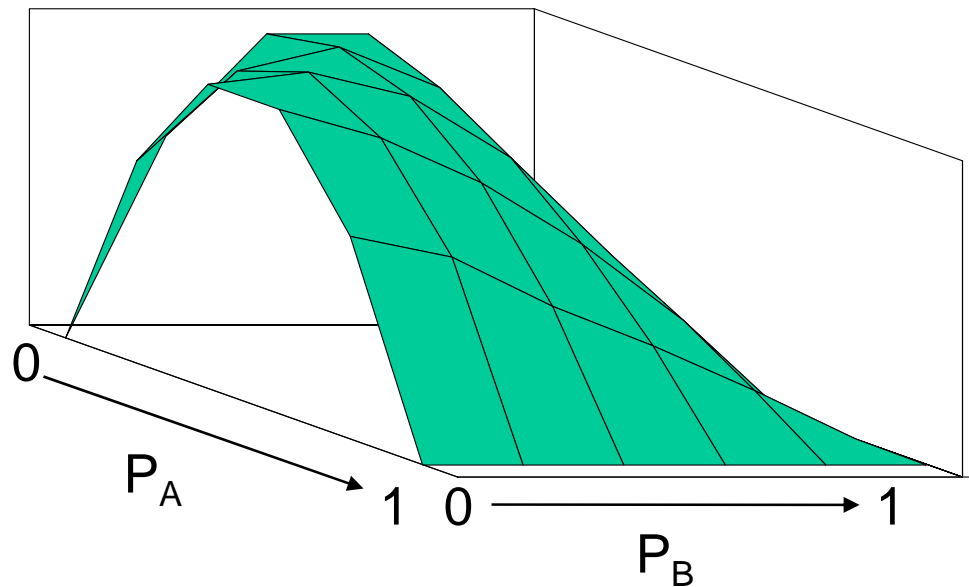
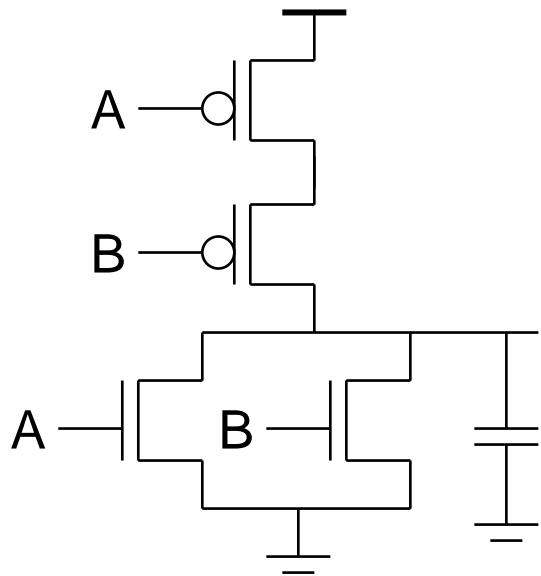
Permissions to Use Conditions & Acknowledgment

- **Permission is granted to copy and distribute this slide set for educational purposes only, provided that the complete bibliographic citation and following credit line is included: "Copyright 2002 J. Rabaey et al." Permission is granted to alter and distribute this material provided that the following credit line is included: "Adapted from (complete bibliographic citation). Copyright 2002 J. Rabaey et al." This material may not be copied or distributed for commercial purposes without express written permission of the copyright holders.**
- **Slides 9-20, 22-5 Adapted from CSE477 VLSI Digital Circuits Lecture Slides by Vijay Narayanan and Mary Jane Irwin, Penn State University**

NOR Gate Transition Probabilities

- **Switching activity is a strong function of the input signal statistics**

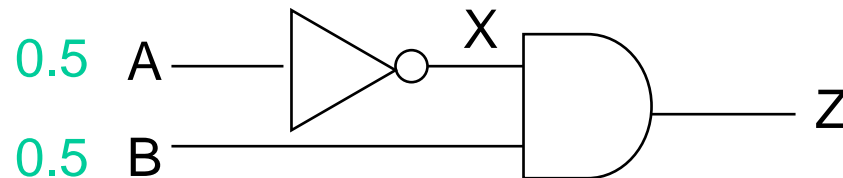
– P_A, P_B are the probabilities that inputs A, B are 1



$$P_{0 \rightarrow 1} = P_0 \times P_1 = (1 - (1 - P_A)(1 - P_B)) (1 - P_A)(1 - P_B)$$

Transition Probabilities for Basic Gates

	$P_{0 \rightarrow 1} = P_{\text{out}=0} \times P_{\text{out}=1}$
NOR	$(1 - (1 - P_A)(1 - P_B)) \times (1 - P_A)(1 - P_B)$
OR	$(1 - P_A)(1 - P_B) \times (1 - (1 - P_A)(1 - P_B))$
NAND	$P_A P_B \times (1 - P_A P_B)$
AND	$(1 - P_A P_B) \times P_A P_B$
XOR	$(1 - (P_A + P_B - 2P_A P_B)) \times (P_A + P_B - 2P_A P_B)$

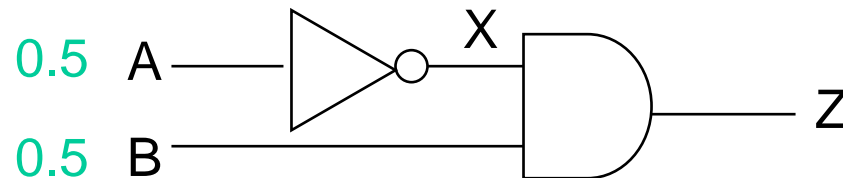


For X: $P_{0 \rightarrow 1} =$

For Z: $P_{0 \rightarrow 1} =$

Transition Probabilities for Basic Gates

	$P_{0 \rightarrow 1} = P_{\text{out}=0} \times P_{\text{out}=1}$
NOR	$(1 - (1 - P_A)(1 - P_B)) \times (1 - P_A)(1 - P_B)$
OR	$(1 - P_A)(1 - P_B) \times (1 - (1 - P_A)(1 - P_B))$
NAND	$P_A P_B \times (1 - P_A P_B)$
AND	$(1 - P_A P_B) \times P_A P_B$
XOR	$(1 - (P_A + P_B - 2P_A P_B)) \times (P_A + P_B - 2P_A P_B)$

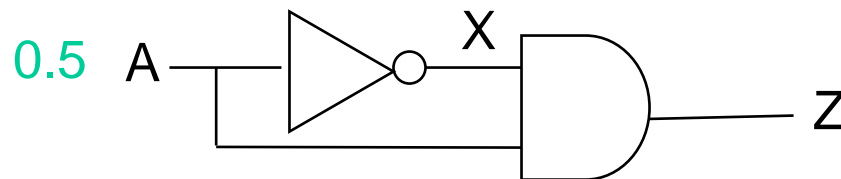


For X: $P_{0 \rightarrow 1} = P_0 \times P_1 = (1 - P_A) P_A = 0.25$

For Z: $P_{0 \rightarrow 1} = P_0 \times P_1 = (1 - P_X P_B) P_X P_B = 3/16$

Problems With Input-to-Output Evaluation

- **Signal and transition probabilities are progressively evaluated from input to output node**
 - Straightforward approach determined by circuit topology
 - Does not deal with feedback (for example in sequential circuits)
 - Assumes input signal probabilities are independent

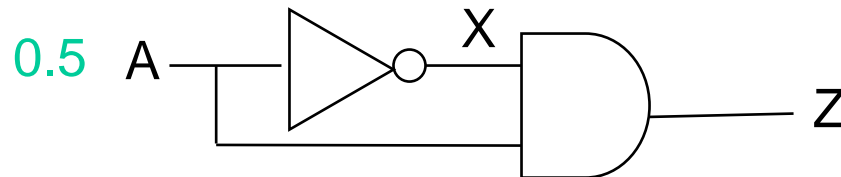


For X: $P_{0 \rightarrow 1} =$

For Z: $P_{0 \rightarrow 1} =$

Problems With Input-to-Output Evaluation

- **Signal and transition probabilities are progressively evaluated from input to output node**
 - Straightforward approach determined by circuit topology
 - Does not deal with feedback (for example in sequential circuits)
 - Assumes input signal probabilities are independent

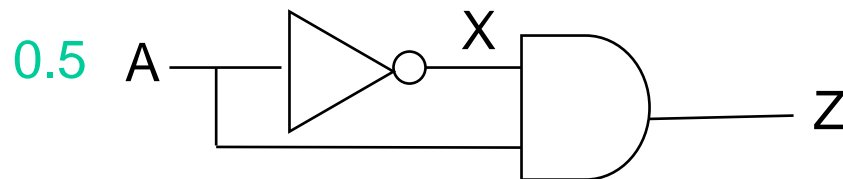


$$\text{For X: } P_{0 \rightarrow 1} = P_0 \times P_1 = (1 - P_A) P_A = 0.25$$

$$\text{For Z: } P_{0 \rightarrow 1} = P_0 \times P_1 = (1 - P_X P_A) P_X P_A = 3/16 ?$$

Problems With Input-to-Output Evaluation

- **Signal and transition probabilities are progressively evaluated from input to output node**
 - Straightforward approach determined by circuit topology
 - Does not deal with feedback (for example in sequential circuits)
 - Assumes input signal probabilities are independent



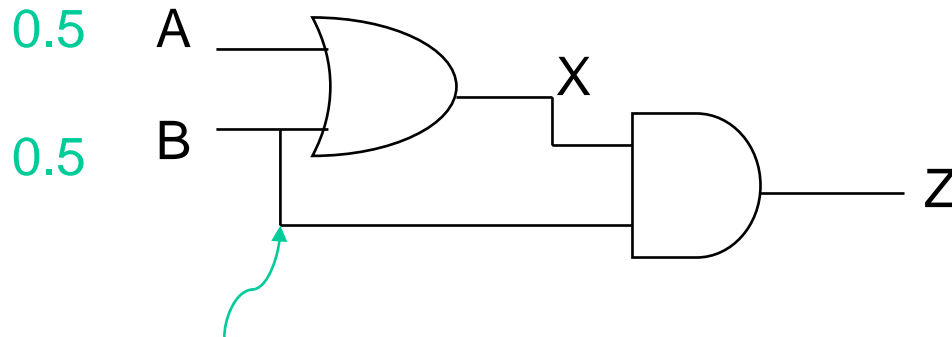
$$\text{For X: } P_{0 \rightarrow 1} = P_0 \times P_1 = (1 - P_A) P_A = 0.25$$

$$\text{For Z: } P_{0 \rightarrow 1} = P_0 \times P_1 = (1 - P_X P_A) P_X P_A = 0!$$

$$Z = A \bullet \overline{A} = 0$$

Intersignal Correlations

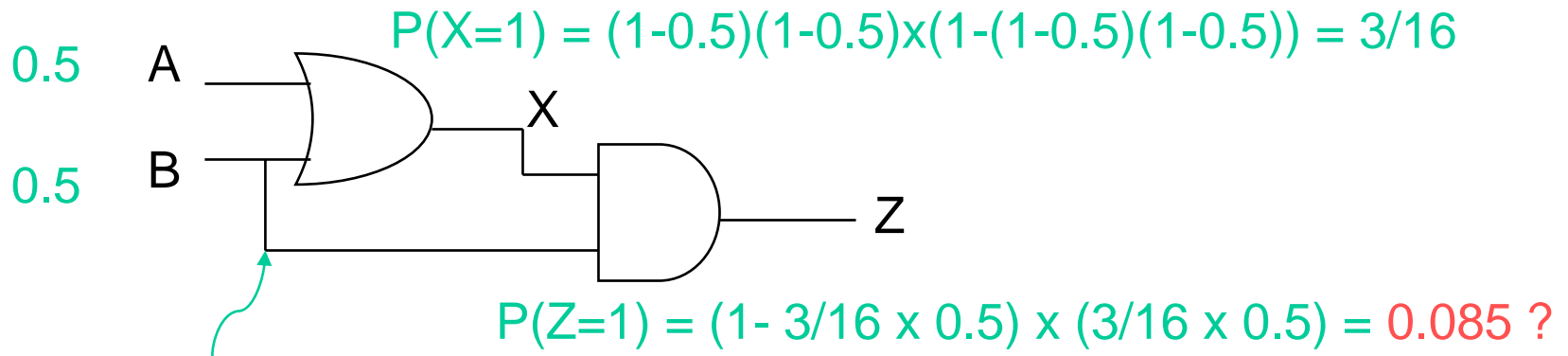
- **Determining switching activity is complicated by the fact that signals exhibit correlation in space and time, e.g. by circuits with reconvergent fanout**



Reconvergent fanout:

Intersignal Correlations

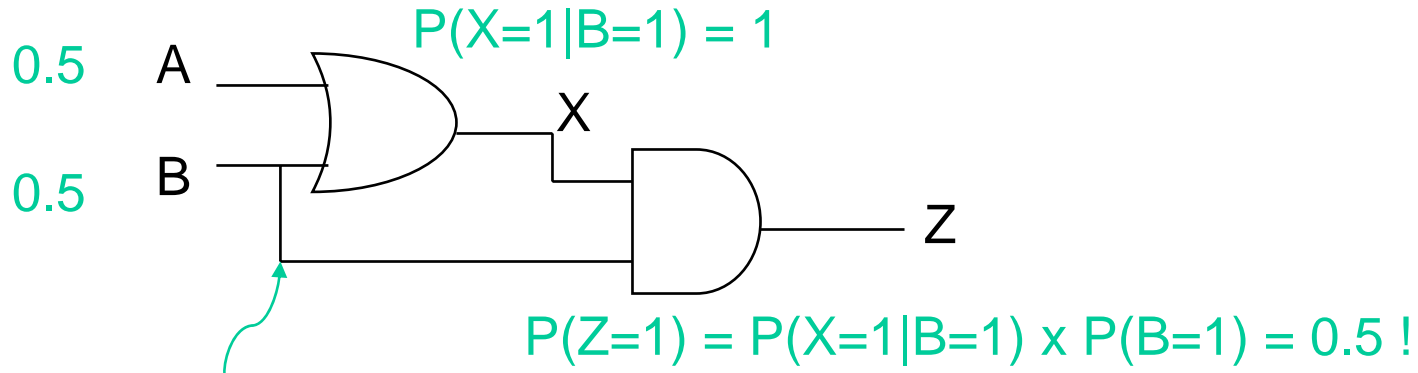
- Determining switching activity is complicated by the fact that signals exhibit correlation in space and time, e.g. by circuits with reconvergent fanout



Reconvergent fanout:

Intersignal Correlations

- Determining switching activity is complicated by the fact that signals exhibit correlation in space and time, e.g. by circuits with reconvergent fanout



Reconvergent fanout:

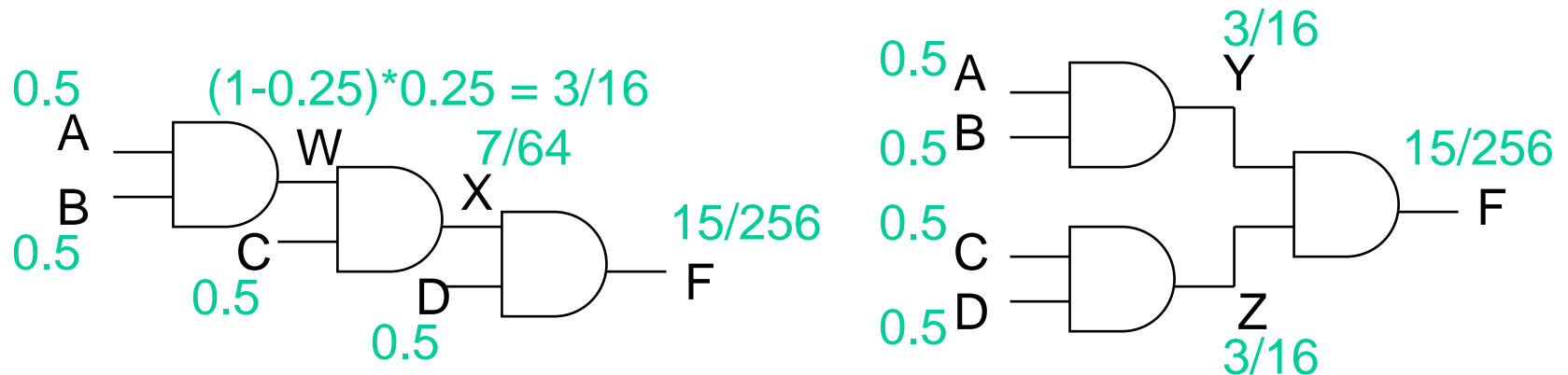
$$P(Z=1) = P(B=1) \times P(X=1 | B=1)$$

- Have to use **conditional probabilities!**

Logic Restructuring

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

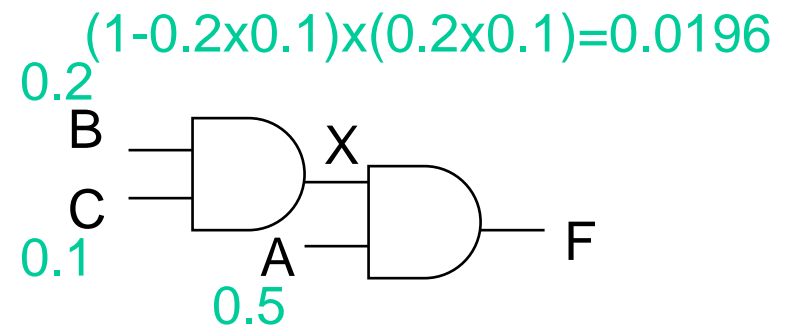
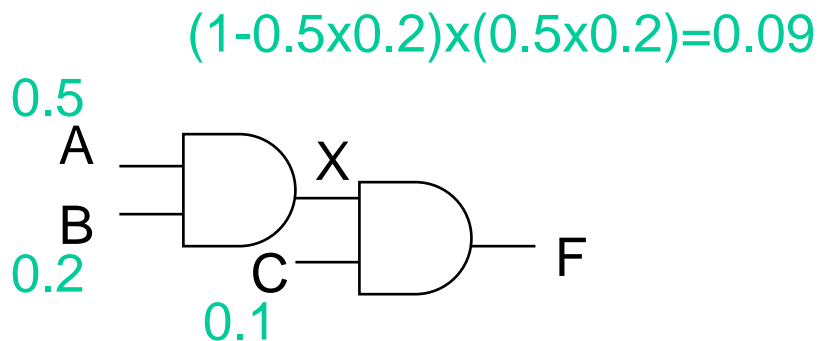
$$\text{AND: } P_{0 \rightarrow 1} = P_0 \times P_1 = (1 - P_A P_B) \times P_A P_B$$



- **Chain implementation has a lower overall switching activity than the tree implementation for random inputs**
- **Ignores glitching effects**

Input Ordering

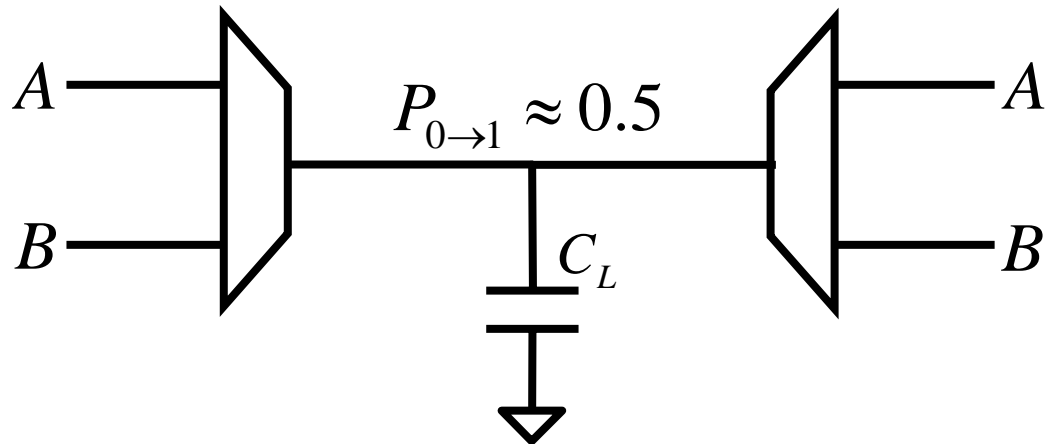
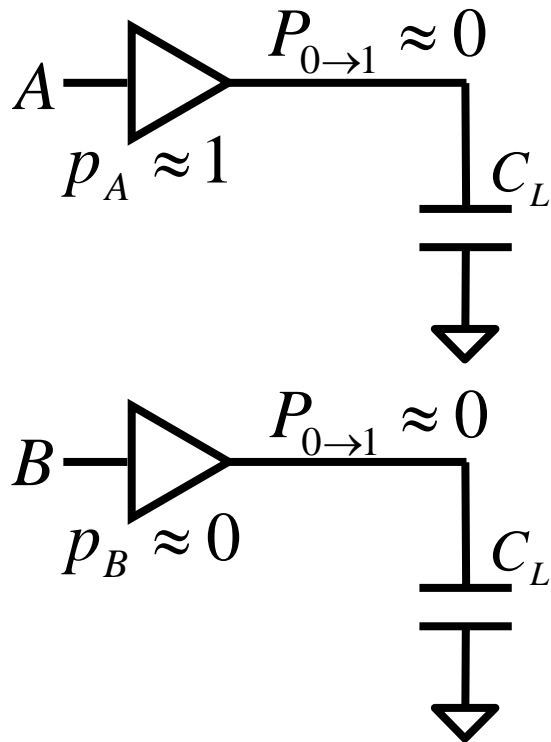
Two alternate implementations with inputs reordered, same function so output stats identical



Beneficial to postpone the introduction of signals with a **high transition rate (signals with signal probability close to 0.5)**

Time Multiplexing Resources

- Time multiplexing can conserve area at the expense of increased switched capacitance
- Consider the case of parallel vs. serial data transmission: more transitions due to muxing

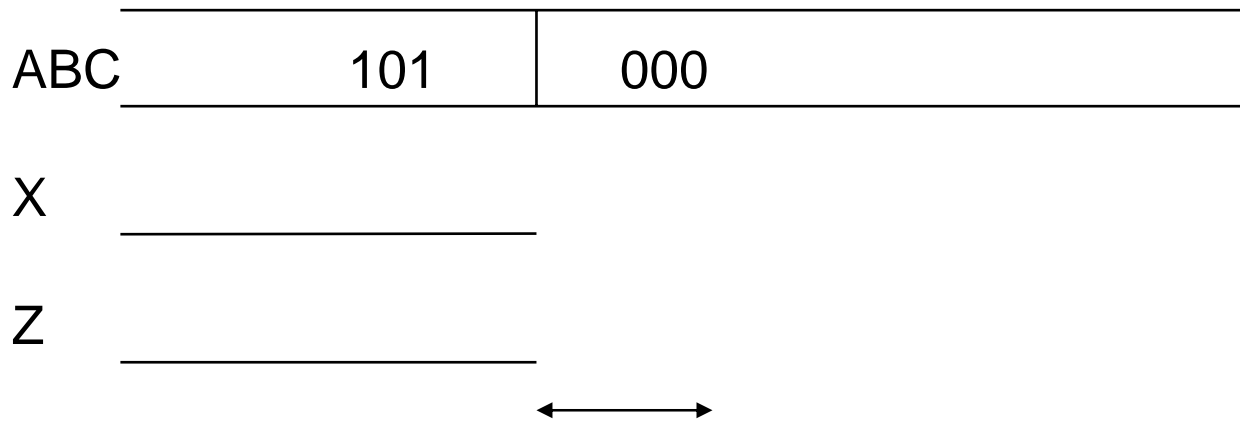
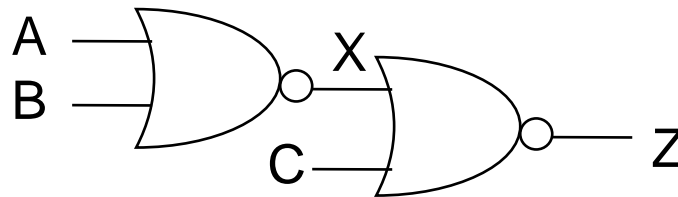


Outline

- Announcements
- Aside: Environmental Impact of Electronics
- Finish Lecture 1 Topics
- Intersignal Correlations
- **Glitches**
- High Level Power Estimation
- Behavioral Level Power Estimation
- Architectural Level Power Estimation

Glitching in Static CMOS Networks

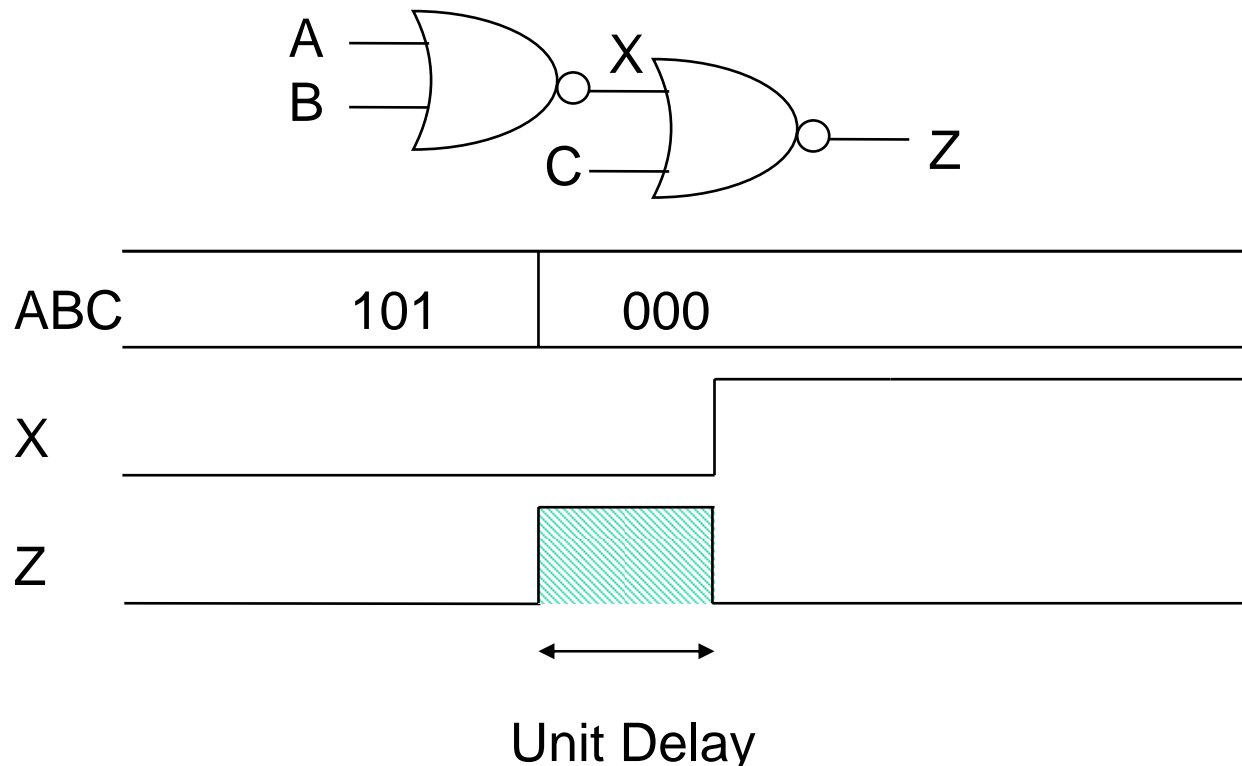
- **Gates have a nonzero propagation delay resulting in spurious transitions or **glitches** (dynamic hazards)**
 - Glitch: node exhibits multiple transitions in a single cycle before settling to the correct logic value



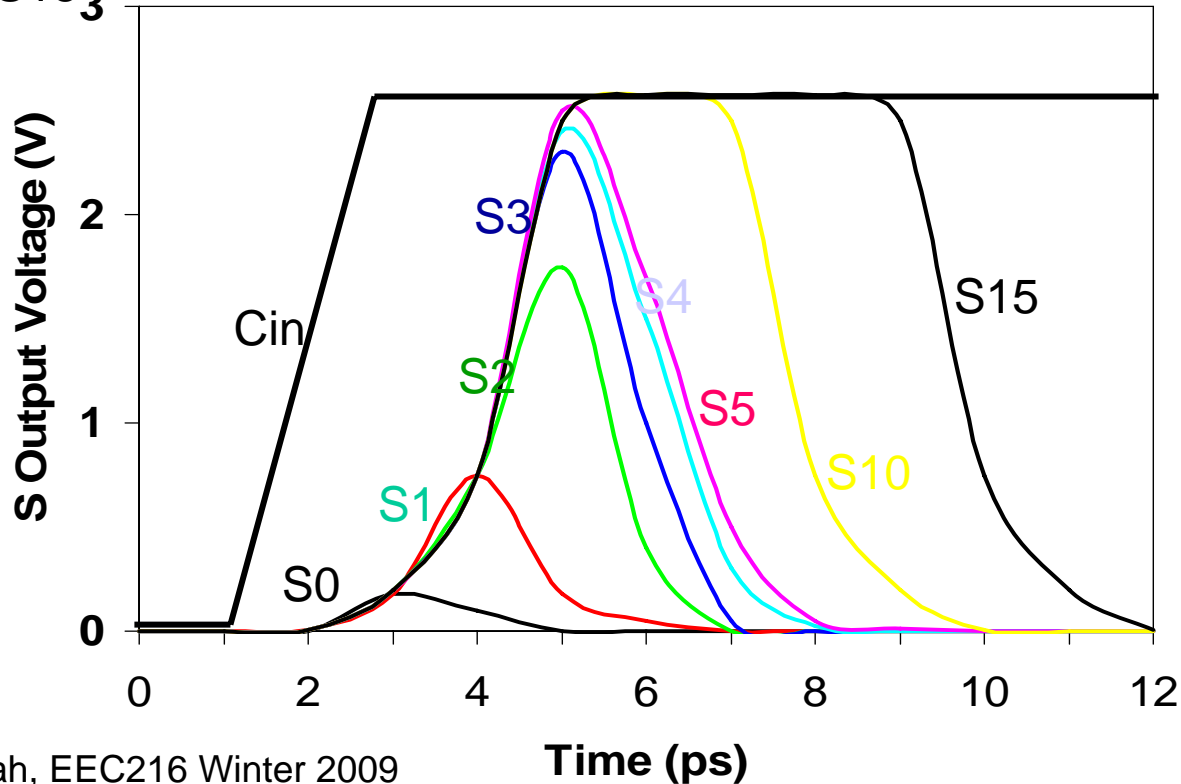
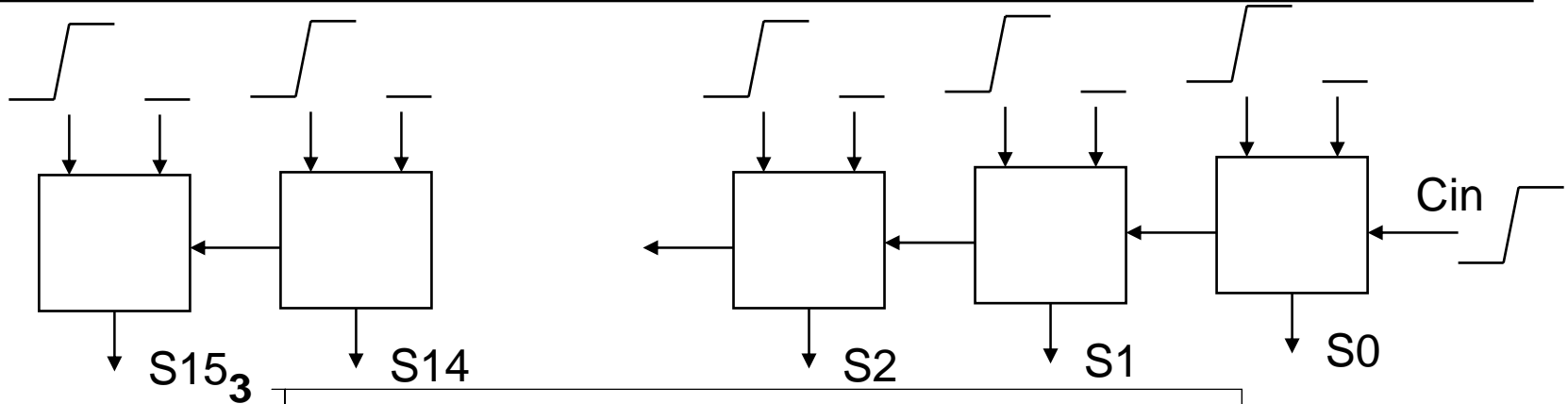
Unit Delay

Glitching in Static CMOS Networks

- **Gates have a nonzero propagation delay resulting in spurious transitions or **glitches** (dynamic hazards)**
 - Glitch: node exhibits multiple transitions in a single cycle before settling to the correct logic value

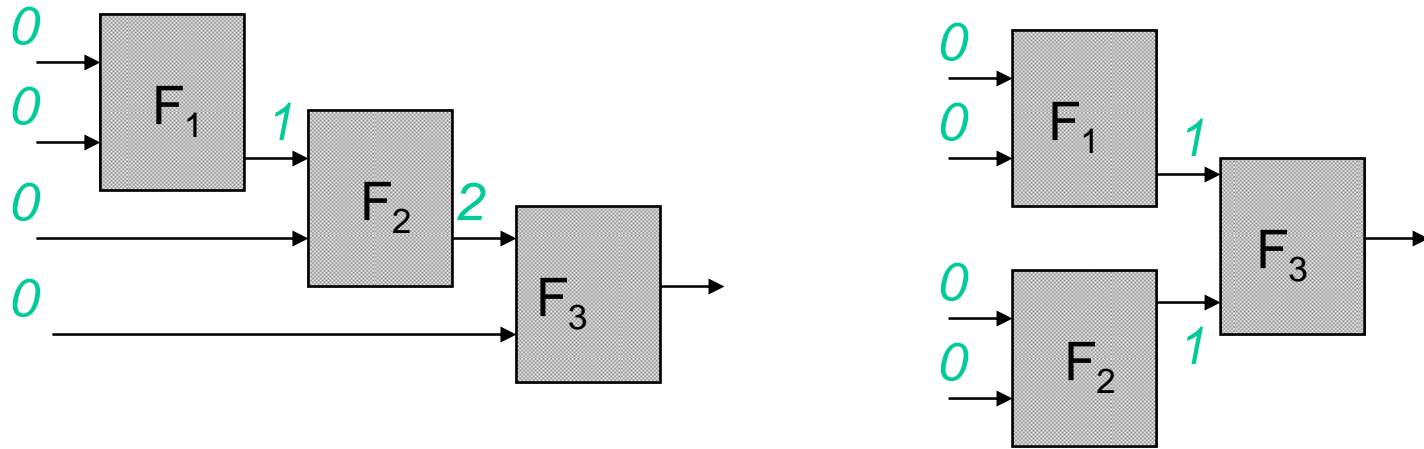


Glitching In A Ripple Carry Adder



Balanced Delay Paths to Reduce Glitching

- Glitching is due to a mismatch in the path lengths in the logic network; if all input signals of a gate change simultaneously, no glitching occurs



So equalize the lengths of timing paths through logic

Example: F_1 and F_2 have unit delay

Switching Activity in Sequential Circuits

- **Estimating activity more difficult for sequential than combinational circuits**
 - Due to activity of state nodes typically being unknown at beginning of estimation process
 - Compute activity assuming steady state
- **Assume sequential circuits implement FSM where states follow Markov process**
 - Solve for probabilities of states using state transition probabilities
- **Best implemented in CAD tools**

Some Notes on Tools

- **Switch-level simulators (e.g. IRSIM)**
 - Replace MOSFETs with ideal switch and equivalent resistance in series for driver
 - Uses capacitors as loads
 - Very fast, captures switching activity, not very accurate
- **Lookup Table simulators (e.g. PowerMill, Nanosim)**
 - Create lookup table for device I-V curve from Spice, rather than implement I_{DS} equations
 - Fast, more accurate than switches, sometimes fails in odd operating regimes like subthreshold
- **Spice-like analog circuit simulations**
 - Most accurate accuracy, most time consuming

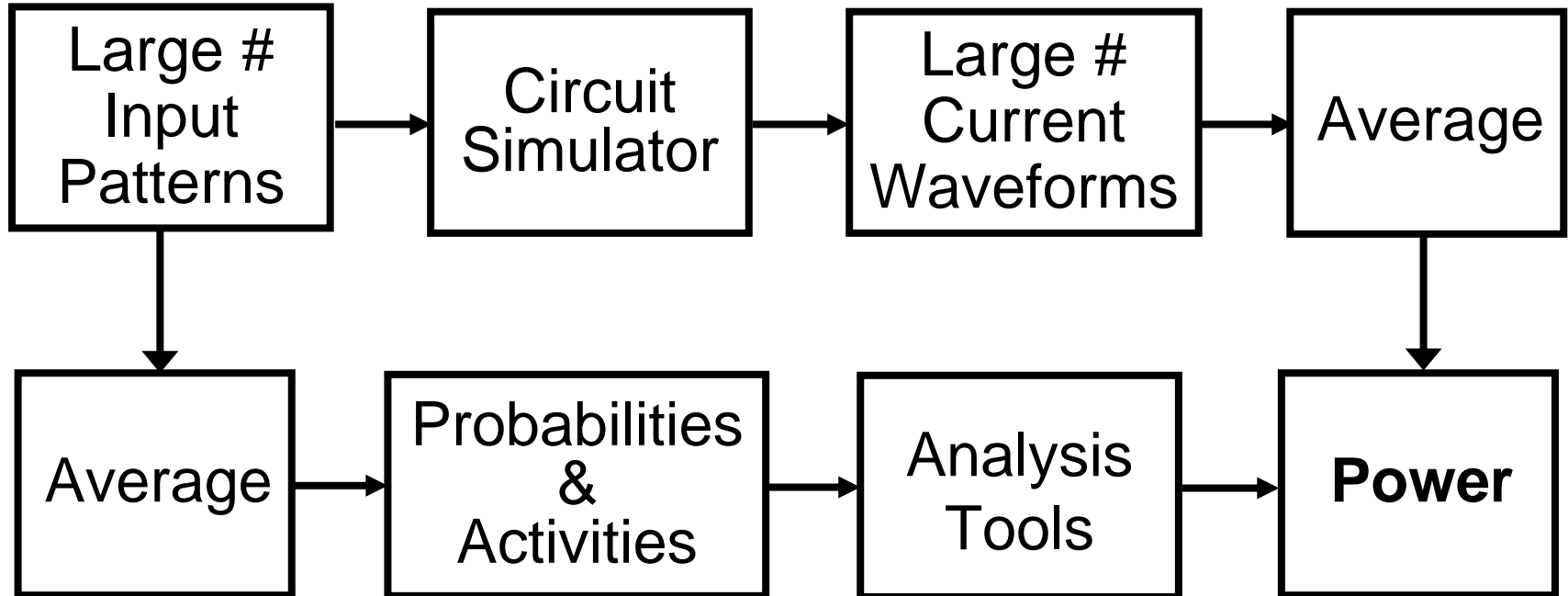
Tool Caveats

- **Beware of Garbage In-Garbage Out problems**
 - Rely heavily on process modeling for device I-V curves
 - Generating I-V curve lookup tables can introduce errors
 - Inconsistent simulations: e.g., circuit works in Spice but not in LUT-based tool
- **Sanity check simulation outputs**
 - Make sure peak currents consistent with total device width in design by estimating on resistance of every pullup/pulldown network in parallel
 - Convert power number to equivalent switched capacitance and compare to hand estimate of total capacitance in design

Outline

- Announcements
- Aside: Environmental Impact of Electronics
- Finish Lecture 1 Topics
- Intersignal Correlations
- Glitches
- **High Level Power Estimation**
- Behavioral Level Power Estimation
- Architectural Level Power Estimation

Alternative Power Estimation Flows



- **Top approach very compute-intensive**
 - Difficult to determine minimum number of simulation runs, long time to simulate large designs
 - Provides greatest accuracy

High-Level Power Estimation

- **Estimate power early in design process**
 - Feedback on system partitioning, architecture, instruction set choice
 - Rough estimate enables optimization of major power consuming modules
 - Only relative accuracy required
- **Final design stages use device-based power estimation, absolute accuracy required**
 - Logic synthesis or custom approach generates design
 - Switch-level simulations
 - Lookup table based simulations
 - SPICE-like analog circuit simulations

Top Level Power Estimation

- **System Level Power Estimation**
 - Allows designer to analyze power impact of system partitioning among software, FPGAs, ASICs, etc.
 - Spreadsheet analysis using library of models for entire components
 - Models created from measurements, low-level power estimation
- **Instruction Level Power Estimation**
 - Run assembly instructions on target processor and measure power
 - Create database of power cost for individual instructions, pairs, maybe entire frequently used traces
 - Add in cache misses, pipeline stalls, etc.

Outline

- Announcements
- Aside: Environmental Impact of Electronics
- Finish Lecture 1 Topics
- Intersignal Correlations
- Glitches
- High Level Power Estimation
- **Behavioral Level Power Estimation**
- Architectural Level Power Estimation

Behavioral Level Power Estimation

- **Start at the behavioral specification or algorithm level**
 - Typically assume some architecture for execution
 - Must predict memory configuration, number of accesses, bus architecture, average wire length, number of bus transactions, control path complexity
 - Components of power include capacitance and activity
- **Physical capacitance can be computed using estimated gate count, previous design experiences**
- **Two styles of activity prediction**
 - Static: based on estimating access frequency of resources
 - Dynamic: based on profiling simulations with user supplied inputs

Static Activity Prediction

- **Start with behavioral specification of function (HDL, C)**
 - Static analysis of control-data flow graph to determine frequency of resource accesses f_r
 - Fast: requires only one analysis pass through program
 - Ignores data dependencies, must use approximations or profiling (dynamic activity prediction) results to incorporate these

$$P = \sum_{r \in \{all\ datapath, control, memory, bus\ resources\}} C_r V_{DD}^2 f_r$$

- C_r determined by empirical models (for example, benchmarking previous designs)
- Low absolute accuracy but captures general trends
- Dynamic prediction slow since requires many sims

Outline

- Announcements
- Aside: Environmental Impact of Electronics
- Finish Lecture 1 Topics
- Intersignal Correlations
- Glitches
- High Level Power Estimation
- Behavioral Level Power Estimation
- **Architectural Level Power Estimation**

Architecture Level Power Estimation

- **Start at register-transfer (RTL) level**
 - Primitives are functional units: adders, multipliers, controllers, register files, memory arrays
 - Gate, circuit, layout level details not yet specified
 - Floorplan may not be available, must estimate interconnect and clock distribution
- **Two strategies: analytical and empirical methods**
 - Each technique can be subdivided further
- **Goal: Relate power consumption of RTL to physical capacitance and switching activity**

Analytical Method 1: Complexity-Based

- **Complexity of a chip architecture can be described in terms of “gate equivalents”**
 - Primitives are specified as the number of reference gates required to implement
 - Example: 256 AND gates + 256 full adders = 16 x 16 array multiplier
 - Gate-equivalents specified in database or provided by user
- **Power estimated by multiplying gate equivalent by average power per gate**

$$P = \sum_{i \in \{functional\ units\}} GE_i \left(E_{typ} + C_L^i V_{DD}^2 \right) f A_i$$

Gate Equivalent Average Power

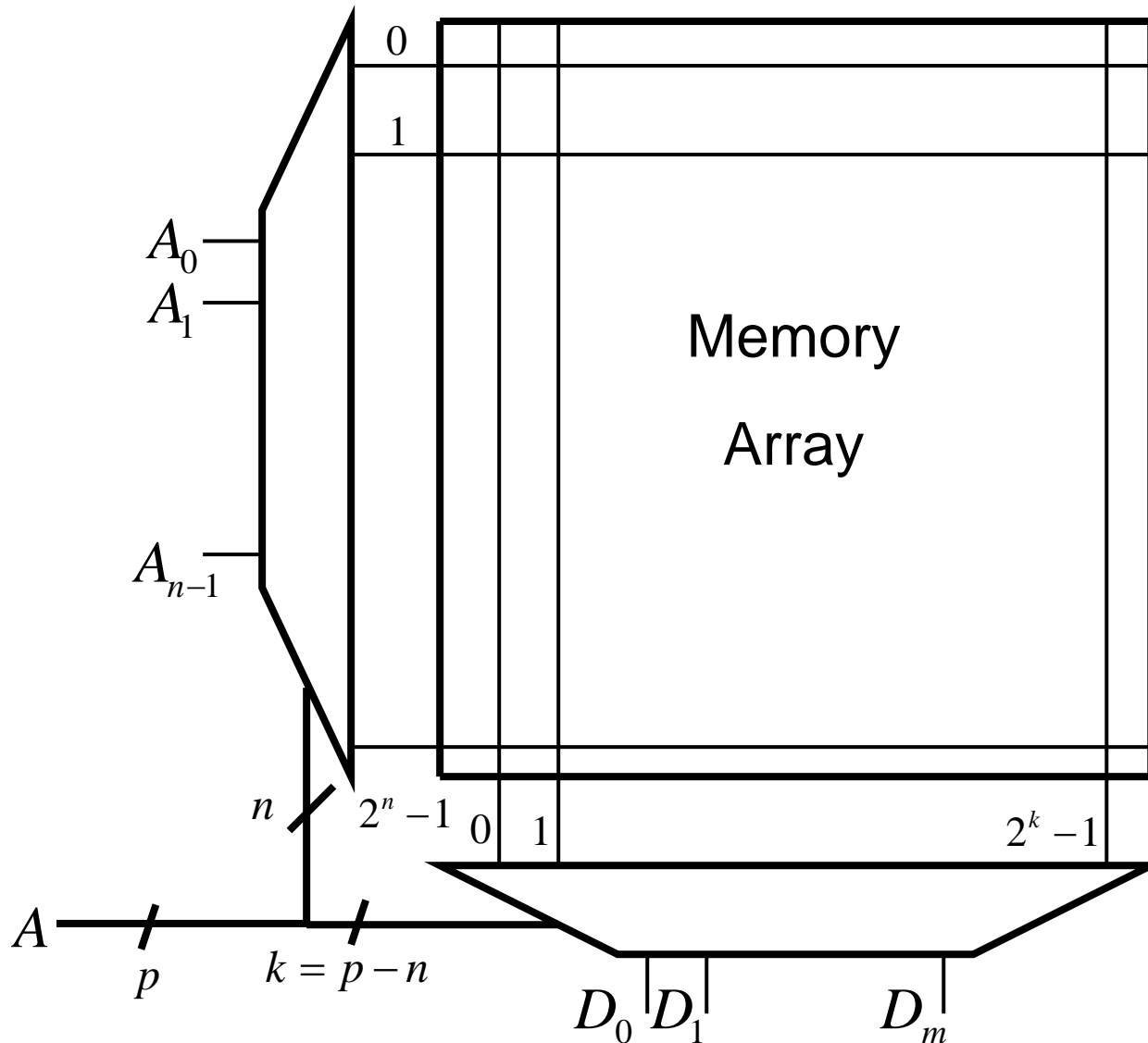
$$P = \sum_{i \in \{functional\ units\}} GE_i \left(E_{typ} + C_L^i V_{DD}^2 \right) f A_i$$

- **Model Parameters:**

- GE_i : number of gate equivalents in unit i
- E_{typ} : average energy consumed by gate
- C_L^i : average capacitive load incl. fanout & wire
- A_i : activity (average percentage of gates switching)

- **Improve accuracy by including models for different gates, circuit styles, layout techniques, special blocks**

Example: SRAM Array



Memory Cell Array Power Model

$$P_{memcell} = \frac{2^k}{2} (c_w l_{col} + 2^n C_{cd}) V_{DD} V_{bl} f$$

- **Model Parameters:**

- 2^k : Number of cells in a row
- C_w : bit line wire capacitance per unit length
- l_{col} : length of column (bit line)
- C_{cd} : wordline transistor drain capacitance
- V_{bl} : bit line voltage swing (often very small)

- **Can use similar models to optimize memory partitioning**

Tradeoffs of Complexity-Based Method

- **Requires very little information to obtain estimate**
 - Energy per gate based on technology parameters
 - Number of gate equivalents from library or experience
 - Activity factor supplied by user, possibly from experience
 - Interconnect length and capacitance modeled by derivative of Rent's Rule
- **Does not model activity very accurately**
 - Misses data dependence
 - User must supply good estimate per functional unit to analyze architectural tradeoffs accurately

Analytical Method 2: Activity-Based

- **Methods attempt to model activity more accurately than complexity methods**
 - Idea is to relate power of functional unit to amount of computation performed
 - Use information theory concept of entropy as metric for computational work
- **Power is proportional to:**
 - Capacitance x Activity
 - Area x Entropy
- **Output Entropy of Functional Unit with m outputs:**

$$H_o = \sum_{i=1}^{2^m} p_i \log_2 \frac{1}{p_i}$$

Area and Entropy Estimates

- **Functional unit with n inputs, m outputs:**

$$\text{Area} \propto \begin{cases} \frac{2^n}{n} H_o & , n \rightarrow \infty \\ 2^n H_o & , n \leq 10 \end{cases}$$

- **Average entropy of all nodes in functional unit (where H_i is input entropy, similarly defined):**

$$\text{Entropy} \approx \frac{2/3}{n+m} (H_i + 2H_o)$$

- Assumes entropy decreases quadratically with logic depth

Activity-Based Power Estimate

- **Methodology**
 1. Run many RTL simulations to measure input and output entropies
 2. Compute area and average node entropy
 3. Compute average power
- **No timing information so glitching power totally unaccounted for**
- **Implicitly assumes capacitance uniformly distributed over all circuit nodes**
- **Methods still topic of research, yet to demonstrate practical value**

Empirical Method 1: Fixed Activity

- **Relate power consumption of RTL components to measured power of existing implementations (macromodeling)**
 - Best suited for library-based design implementation
- **Example: Power Factor Approximation**

$$P = \sum_{i \in \{functional\ units\}} k_i G_i f_i$$

- **Model Parameters:**
 - k_i : power factor constant
 - G_i : hardware complexity metric
 - f_i : activation frequency

Power Factor Approximation Example

- Try to estimate power for an $N \times N$ array multiplier
- Hardware complexity goes as square of input word length N
- Clock frequency f_{mult} specified by algorithm and performance constraints
- Power factor k_{mult} determined from past designs reported in literature (15 fW/bit²-Hz for 1.2 μ m CMOS at 5 V supply)

$$P = k_{mult} N^2 f_{mult}$$

- Ex: 32 x 32 multiply at 233 MHz consumes 3.58 mW
- Does not account for data dependence since power factor is assumed constant

Empirical Method 2: Activity Sensitive

- **Attempt to incorporate data activity statistics into power estimates**
 - Count number of bit transitions in input vector stream and multiply by experimentally determined constant fudge factor
 - Develop activity model empirically through RTL simulations for typical input streams
 - Construct multiple models depending on function, for example a datapath activity model and a control path activity model
- **Classic example: Dual-Bit Type (DBT) model**

Dual-Bit Type Model

- **Observation: fixed-point, 2's-complement data streams often characterized by two distinct activity regions**
 - Data bits (LSBs) exhibit activity similar to uniformly distributed white noise
 - Sign bits (MSBs) depend on sign transition probability, which is in turn related to temporal correlation ρ
- **Different empirically determined coefficients characterize switched (effective) capacitance and complexity in data (C_U and N_U , respectively) and sign (C_S and N_S) regions:**

$$P = (N_U C_U + N_S C_S) V_{DD}^2 f$$

DBT Data from Landman TCAD 96

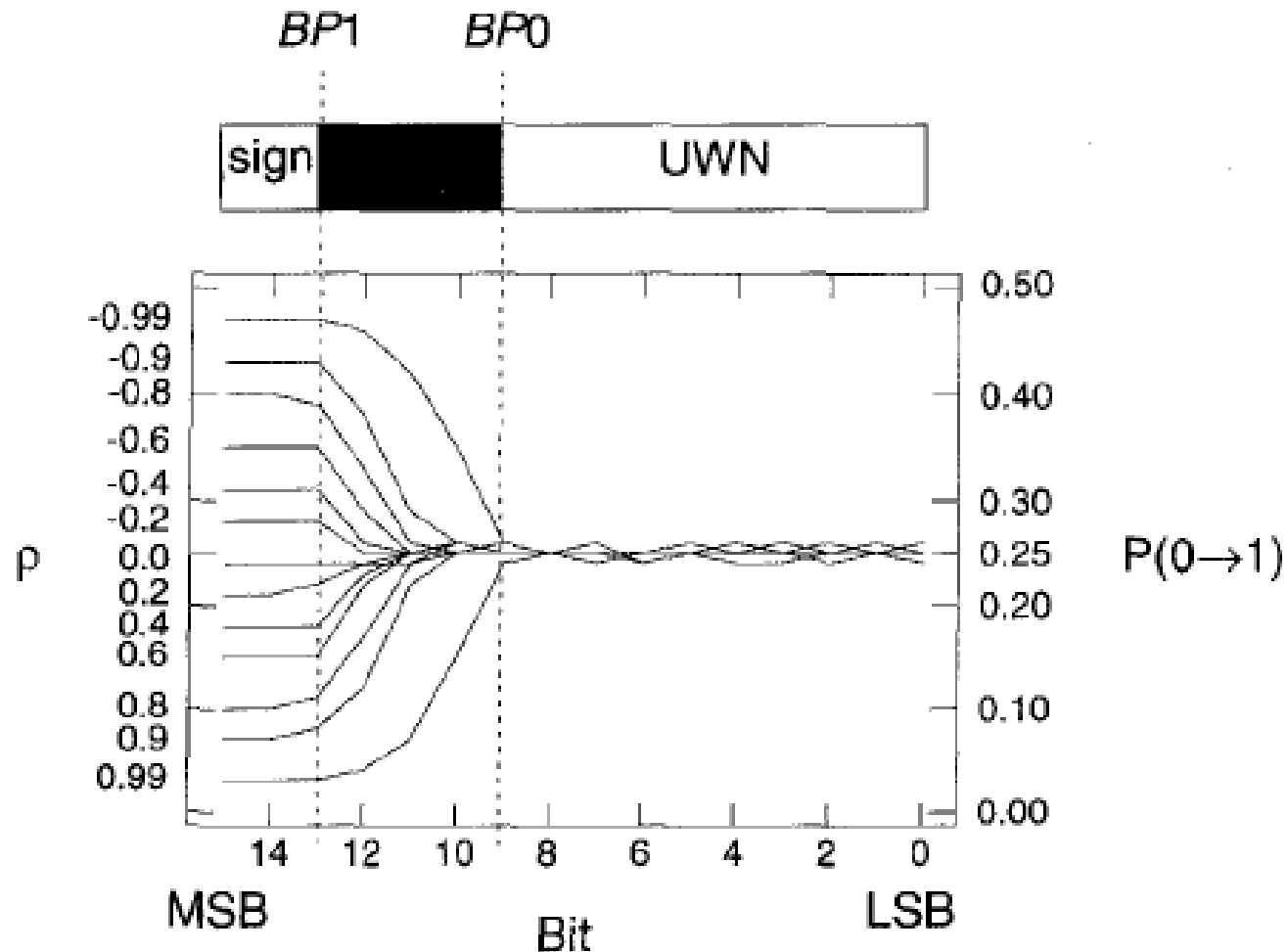


Fig. 3. Bit transition activity for data streams with varying temporal correlation.

Dual-Bit Type Model Breakpoints

- Can derive expressions for different bit region breakpoints based on data stream statistics such as mean (μ), variance (σ^2), and correlation (ρ):

$$\rho = \text{COV}(X_{t-1}, X_t) / \sigma^2$$

$$BP1 = \log_2 (|\mu| + 3\sigma)$$

$$BP0 = \log_2 \sigma + \Delta BP0$$

$$\Delta BP0 = \log_2 \left[\sqrt{1 - \rho^2} + \frac{|\rho|}{8} \right]$$

Control Path Activity Model

- **Observation: control path words often lack definite structure (unlike datapath)**
 - Words formed by concatenating independent fields or boolean flags, condition codes
 - Use transition probabilities and activity factors for complex gates like we analyzed earlier
 - Combine with complexity measurements as well
- **Different empirically determined coefficients characterize switched (effective) capacitance (C_I for inputs, C_O for outputs) and complexity:**

$$P = \left(N_I \alpha_I C_I N_M + N_O \alpha_O C_O N_M \right) V_{DD}^2 f$$

- N_I, N_O, N_M are complexity of input, output, and state transition logic

Summary: Architectural Power Estimation

- **Bridge gap between physical power estimation and conceptual level power estimation**
- **Start at register-transfer (RTL) level**
 - Primitives are functional units but gate, circuit, layout, floorplan details not yet specified
- **Analytical methods aim to estimate power from physical primitives**
 - Estimate capacitance by gate count, area
 - Determine activity from user input, entropy
- **Empirical methods based on measurements of previous designs or ensemble of simulations**
 - Model datapath and control activity separately

Next Topic: Interconnect Power

- **Look at estimating interconnect power and driving long wires**