Lecture Overview

• Shifters
• Low-power ALU intro
• CAD4 due Tues
• Quiz next Thurs in class
  – Example to be posted soon
Some Insights from ISSCC 2009

- Intel → first fully static datapath for a high performance part in >20 years on Nehalem
  - 8 cores, pairs of caches/cores can be deactivated if defects occur
  - Top-level metal is 7-8um thick for extremely low resistance
  - Significant error correction capabilities
- Micro fuel cell deposited on chip, 4x4mm
- Cyborg moth radio – UWB, piezoelectric energy scavenging

Last Time

- Multipliers
  - Made primarily of arrays of adders
  - Can be very slow, must be carefully optimized
- Modified Booth recoding → map multiplier to a smaller encoded version and reduce partial product count by 2X
  - 2s complement follows naturally
Shift Types

- Arithmetic vs. logical shift (start with 1101)
  - Logical shift (baseline op for us)
    - Logical shift right by 1: 0110  \(\text{Shift in 0s for logical right shifts}\)
    - Logical shift right by 2: 0011
    - Logical shift left by 1: 1010  \(\text{Shift in 0s for left shifts}\)
    - Logical shift left by 2: 0100
  - Arithmetic shift (not baseline)
    - Arithmetic left shifts same as logical
    - Arithmetic right shift by 1: 1110  \(\text{Repeat sign of MSB for arithmetic right shifts}\)
    - Arithmetic right shift by 2: 1111

The Binary Shifter Concept

The Binary Shifter Concept
Barrel Shifter

Area Dominated by Wiring

Requires decoding of shift amount given in instruction word

4x4 barrel shifter

Only 1 shift bit high at any time

\[ \text{Width}_{\text{barrel}} \approx 2 \, p_m \, N \]

N is max shift width (15 for us), \( p_m \) is metal pitch
Logarithmic Shifter

\[ A_3 \rightarrow Sh1 \rightarrow B_3 \]
\[ A_2 \rightarrow Sh2 \rightarrow B_2 \]
\[ A_1 \rightarrow Sh4 \rightarrow B_1 \]
\[ A_0 \rightarrow Sh4 \rightarrow B_0 \]

0-7 bit Logarithmic Shifter

\[ \text{width}_{\text{log}} \cdot p_m \left( 2^K + \left\{ \frac{1}{2^1} + \frac{1}{2^2} + \cdots + \frac{1}{2^K} \right\} \right) = p_m \left( 2^K + 2^{K-1} \right) \]
\[ K = \log_2 N \]
Other Spins on Logarithmic Shifters

- \( \log_4 \) instead of \( \log_2 \)
  - Advantage: Fewer stages of pass transistors
  - Disadvantage: must re-encode the control bits
  - Ex: 16 bits \( \rightarrow \) 4 stages of pass transistors for \( \log_2 \) vs. only 2 stages for \( \log_4 \)
- Can use CMOS transmission gates instead of NMOS pass transistors
- Rotate instruction – could be an ISA addition
  - 11010101 \( \rightarrow \) rotate right by 5 bits \( \rightarrow \) 10101110
- Reverse order shifters – do the big shifts first
  - Helpful because the big shifts have larger wire capacitances
  - Elmore delay is reduced by having large caps charged through the least resistance (fewest pass transistors)

Active Power Reduction

\[
P \sim \alpha \cdot C_L \cdot V_{swing} \cdot V_{DD} \cdot f \quad E \sim \alpha \cdot C_L \cdot V_{swing} \cdot V_{DD}
\]

- Reducing load capacitance
  - Technology scaling
  - Gate sizing, logic minimization, better placement tools
  - Logic families (pass transistor logic, …)
- Reducing supply voltage
  - Quadratic impact on power
  - Impact on delay – how to maintain throughput?
- Reducing frequency – performance penalty
- Reducing switching probability (\( \alpha \))
  - Architecture
  - Glitching power reduction (15-20%)
Glitching in Static CMOS Networks

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

![Gate Diagram]

<table>
<thead>
<tr>
<th>ABC</th>
<th>X</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Unit Delay

Glitching in a Ripple Carry Adder

![Adder Diagram]

![Voltage Waveform Graph]
Balanced Delay Paths to Reduce Glitching

- Glitching is due to mismatches in path lengths in a logic network
- If all input signals of a gate change simultaneously, no glitching occurs

Equalize the lengths of timing paths through logic

Circuit-Level Activity Encoding

Conditional inversion coding for buses:
If more than ½ of bits switch, switch the inverse set to save power

Increases delay due to encode/decode elements
Clock Gating

- Most popular method for power reduction of clock signals and functional units
- Turn off clock to idle functional units
  - Ex. floating point units
  - Need logic to generate \textbf{disable} signal
    - Increases complexity of control logic
    - Consumes power
    - Timing critical to avoid clock glitches at OR gate output
  - Additional gate delay on clock signal
    - OR gate can replace a buffer in the clock distribution tree

Summary

- Barrel shifters are area-intensive but have only 1 pass transistor per path
  - Lots of junction capacitance though
- Log shifters are more versatile for wider data
  - Various choices: log base, reverse order, pass transistor vs. T-gate, buffering
- Low-power design
  - Can focus on reducing any # of things from switching activity to cap to voltage
  - Extremely important today; everyone cares about power