A new company called Nanoscribe has developed a fabrication device that can create structures like the one at the right at the micro scale in minutes (instead of hours). The idea is that “tiny, ultrashort pulses from a near-infrared laser on a light-sensitive material solidifies on spot. Mirrors not motors.”

Review

• Use this table and techniques we learned to transform from 1 to another
Today

• Data Multiplexors
• Arithmetic and Logic Unit
• Adder/Subtractor
Data Multiplexor (here 2-to-1, n-bit-wide)
N instances of 1-bit-wide mux

How many rows in TT?

c = \overline{s}a\overline{b} + \overline{s}ab + s\overline{ab} + sab
= \overline{s}(a\overline{b} + ab) + s(\overline{a}b + ab)
= \overline{s}(a(\overline{b} + b)) + s((\overline{a} + a)b)
= \overline{s}(a(1) + s((1)b)
= \overline{s}a + sb
How do we build a 1-bit-wide mux?

\[
\overline{s}a + sb
\]
4-to-1 Multiplexor?

How many rows in TT?

\[ e = \overline{s_1 s_0} a + \overline{s_1 s_0} b + s_1 \overline{s_0} c + s_1 s_0 d \]
Is there any other way to do it?

Hint: NCAA tourney!

Ans: Hierarchically!
Arithmetic and Logic Unit

• Most processors contain a special logic block called “Arithmetic and Logic Unit” (ALU)

• We’ll show you an easy one that does ADD, SUB, bitwise AND, bitwise OR

\[
\begin{align*}
\text{when } S=00, & \quad R=A+B \\
\text{when } S=01, & \quad R=A-B \\
\text{when } S=10, & \quad R=A \text{ AND } B \\
\text{when } S=11, & \quad R=A \text{ OR } B
\end{align*}
\]
Our simple ALU
Adder/Subtractor Design -- how?

- Truth-table, then determine canonical form, then minimize and implement as we’ve seen before
- Look at breaking the problem down into smaller pieces that we can cascade or hierarchically layer
Adder/Subtractor – One-bit adder LSB...

\[\begin{array}{ccc}
  a_3 & a_2 & a_1 \\
  b_3 & b_2 & b_1 \\
  s_3 & s_2 & s_1
\end{array}\]

\[\begin{array}{c|c|c|c}
  a_0 & b_0 & s_0 & c_1 \\
  \hline
  0 & 0 & 0 & 0 \\
  0 & 1 & 1 & 0 \\
  1 & 0 & 1 & 0 \\
  1 & 1 & 0 & 1 \\
\end{array}\]

\[s_0 = \ L_1 \]

\[c_1 = \ L_2 \]
Adder/Subtracter – One-bit adder (1/2)...
Adder/Subtractor – One-bit adder (2/2)…

\[ s_i = \text{XOR}(a_i, b_i, c_i) \]
\[ c_{i+1} = \text{MAJ}(a_i, b_i, c_i) = a_i b_i + a_i c_i + b_i c_i \]
N 1-bit adders ⇒ 1 N-bit adder

What about overflow?
Overflow = \( c_n \)?
What about overflow?

- Consider a 2-bit signed # & overflow:
  - 10 = -2 + -2 or -1
  - 11 = -1 + -2 only
  - 00 = 0 NOTHING!
  - 01 = 1 + 1 only

- Highest adder
  - $C_1 = \text{Carry-in} = C_{\text{in}}$, $C_2 = \text{Carry-out} = C_{\text{out}}$
  - No $C_{\text{out}}$ or $C_{\text{in}} \Rightarrow$ NO overflow!

- $C_{\text{in}}$, and $C_{\text{out}} \Rightarrow$ NO overflow!
  - $C_{\text{in}}$, but no $C_{\text{out}} \Rightarrow A, B \text{ both } > 0$, overflow!
  - $C_{\text{out}}$, but no $C_{\text{in}} \Rightarrow A, B \text{ both } < 0$, overflow!
What about overflow?

• Consider a 2-bit signed # & overflow:

\[
\begin{align*}
10 &= -2 \\
11 &= -1 \\
00 &= 0 \\
01 &= 1
\end{align*}
\]

• Overflows when…

\[
\begin{align*}
&\text{\( C_{\text{in}}, \) but no \( C_{\text{out}} \) } \Rightarrow \ A, B \text{ both } > 0, \text{ overflow!} \\
&\text{\( C_{\text{out}}, \) but no \( C_{\text{in}} \) } \Rightarrow \ A, B \text{ both } < 0, \text{ overflow!}
\end{align*}
\]

\text{overflow} = c_n \hspace{1em} \text{XOR} \hspace{1em} c_{n-1}
Extremely Clever Subtractor

\[ b_{n-1} \quad a_{n-1} \]

\[ b_1 \quad a_1 \quad b_0 \quad a_0 \]

XOR serves as conditional inverter!

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>XOR(a,b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Peer Instruction

1) Truth table for mux with 4-bits of signals has $2^4$ rows

2) We could cascade N 1-bit shifters to make 1 N-bit shifter for sll, srl
Peer Instruction Answer

1) Truth table for mux with 4-bits of signals controls 16 inputs, for a total of 20 inputs, so truth table is $2^{20}$ rows... **FALSE**

2) We could cascade N 1-bit shifters to make 1 N-bit shifter for sll, srl ... **TRUE**

1) Truth table for mux with 4-bits of signals is $2^4$ rows long

2) We could cascade N 1-bit shifters to make 1 N-bit shifter for sll, srl
“And In conclusion…”

- Use muxes to select among input
  - $S$ input bits selects $2^S$ inputs
  - Each input can be $n$-bits wide, indep of $S$
- Can implement muxes hierarchically
- ALU can be implemented using a mux
  - Coupled with basic block elements
- $N$-bit adder-subtractor done using $N$ 1-bit adders with XOR gates on input
  - XOR serves as conditional inverter