Parallel Requests
- Assigned to the computer
  - Example: Search "Katz"

Parallel Threads
- Assigned to the core
  - Example: Lookup, Ads

Parallel Instructions
- >1 instruction @ one time
  - Example: 5 pipelined instructions

Parallel Data
- >1 data item @ one time
  - Example: Add of 4 pairs of words

Hardware Descriptions
- All gates @ one time

Review
- Truth table can be mapped to gates for combinational logic design
- Boolean algebra allows minimization of gates
- Sequential vs. Combinational Logic
- Unique configurations of a digital system: State and State Machines

Type of Circuits
- Synchronous Digital Systems consist of two basic types of circuits:
  - Combinational Logic (CL) circuits
    - Output is a function of the inputs only, not the history of its execution
    - E.g., circuits to add A, B (ALUs)
  - Sequential Logic (SL)
    - Circuits that "remember" or store information
    - Aka “State Elements”
    - E.g., memories and registers (Registers)
Conceptual MIPS Datapath

Agenda
- Multiplexer
- ALU Design
- And in Conclusion, ...

Data Multiplexer
(e.g., 2-to-1 x n-bit-wide)

N Instances of 1-bit-Wide Mux

How Do We Build a 1-bit-Wide Mux (in Logisim)?
4-to-1 Multiplexer

How many rows in TT?

\[ e = \overline{s_1}s_0a + \overline{s_1}s_0b + s_1\overline{s_0}c + s_1s_0d \]

Alternative Hierarchical Approach (in Logisim)

Subcircuits
- Subcircuit: Logisim equivalent of procedure or method
  - Every project is a hierarchy of subcircuits

Logisim

N-bit-wide Data Multiplexer (in Logisim + tunnel)

Administrivia
Agenda

• Multiplexer
• ALU Design
• And, in Conclusion, ...

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

\[
\begin{align*}
S &= S_0 = a_0 \oplus b_0 \\
c_1 &= a_0 \land b_0
\end{align*}
\]

Adder/Subtractor: One-bit adder Least Significant Bit

\[
\begin{array}{cccccc}
a_0 & b_0 & a_1 & b_1 & a_2 & b_2 & a_3 & b_3 & s_0 & c_1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

Adder/Subtractor: One-bit adder (1/2)

Adder/Subtractor: One-bit Adder (2/2)

\[
\begin{align*}
s_i &= a_i \oplus b_i \\
c_{i+1} &= \text{MAJ}(a_i, b_i, c_i)
\end{align*}
\]
N x 1-bit Adders $\Rightarrow$ 1 N-bit Adder

Connect Carry Out $i-1$ to Carry in $i$:

Twos Complement Adder/Subtractor

Critical Path
- When setting clock period in synchronous systems, must allow for worst case
- Path through combinational logic that is worst case called “critical path”
  - Can be estimated by number of “gate delays”:
    Number of gates must go through in worst case
- Idea: Doesn’t matter if speedup other paths if don’t improve the critical path
- What might critical path of ALU?

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
- Arithmetic circuits are a kind of combinational logic