A MIPS ALU

Tom Kelliher, CS 240

Feb. 22, 2002

1 Administrivia

Announcements

No class Monday.

Assignment

For Monday, e-mail assembly exercise solution to me. Read 3.10–16.

For Wednesday, read 4.5 — carry lookahead addition.

From Last Time

String handling, summary of addressing modes, program build process.

Outline

1. Building an ALU.

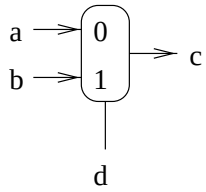
Coming Up

Carry lookahead addition.

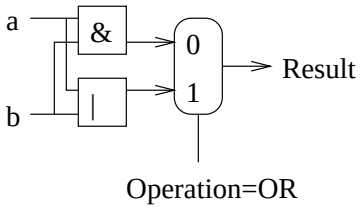
2 Building an ALU

Important design principle: reuse.

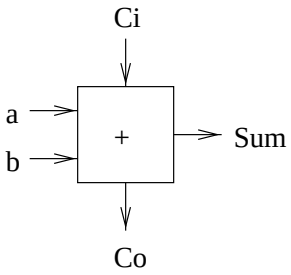
2-1 mux symbol:



A one-bit two function logical unit:

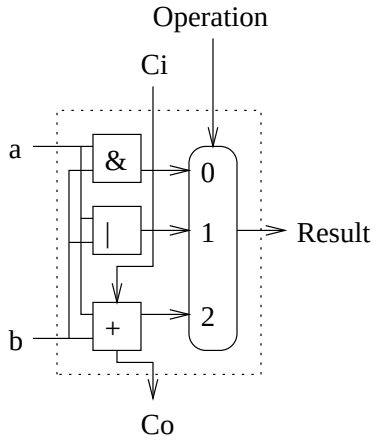


Recall block diagram for full adder:

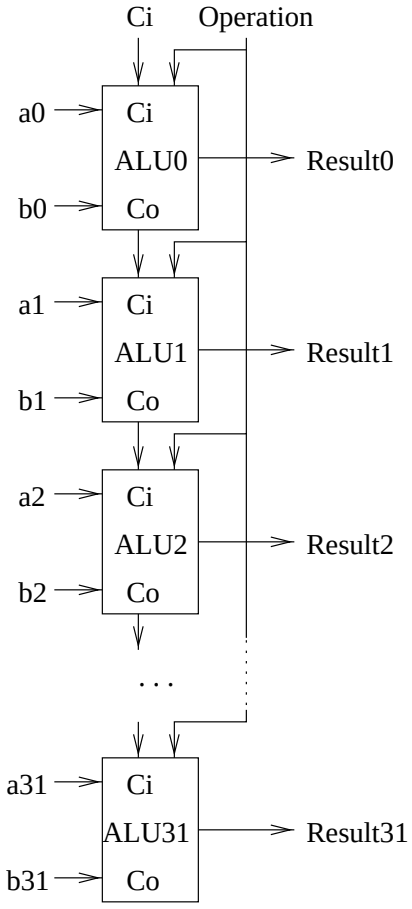


2.1 An ALU

A one-bit three operation ALU:



A 32-bit three operation ALU:



Ripple carry addition.

2.2 A MIPS ALU

What additional functionality needed?

1. Subtraction.
2. `slt` instruction.
3. `beq`/`bne` comparison.
4. Overflow detection.

How do we modify the basic ALU to handle these operations?

Notes:

1. Semantics of `slt`:

   ```
   slt $t0, $t1, $t2

   $t0 = ($t1 < $t2);
   ```

2. What is overflow?

   (a) Adding two unsigned numbers.

   Can the difference of two unsigned numbers result in an overflow?

   (b) Adding two signed (both positive or both negative) numbers.

   Difference?

   (c) Detecting overflow.

2.2.1 Control Summary

Three input lines:

1. Subtract `b`.

2. Mux select: two lines.

3. Eight possibilities, five used: and, or, add, subtract, set on less than.