110
Arithmetic

ENGR 3410 – Computer Architecture

Mark L. Chang

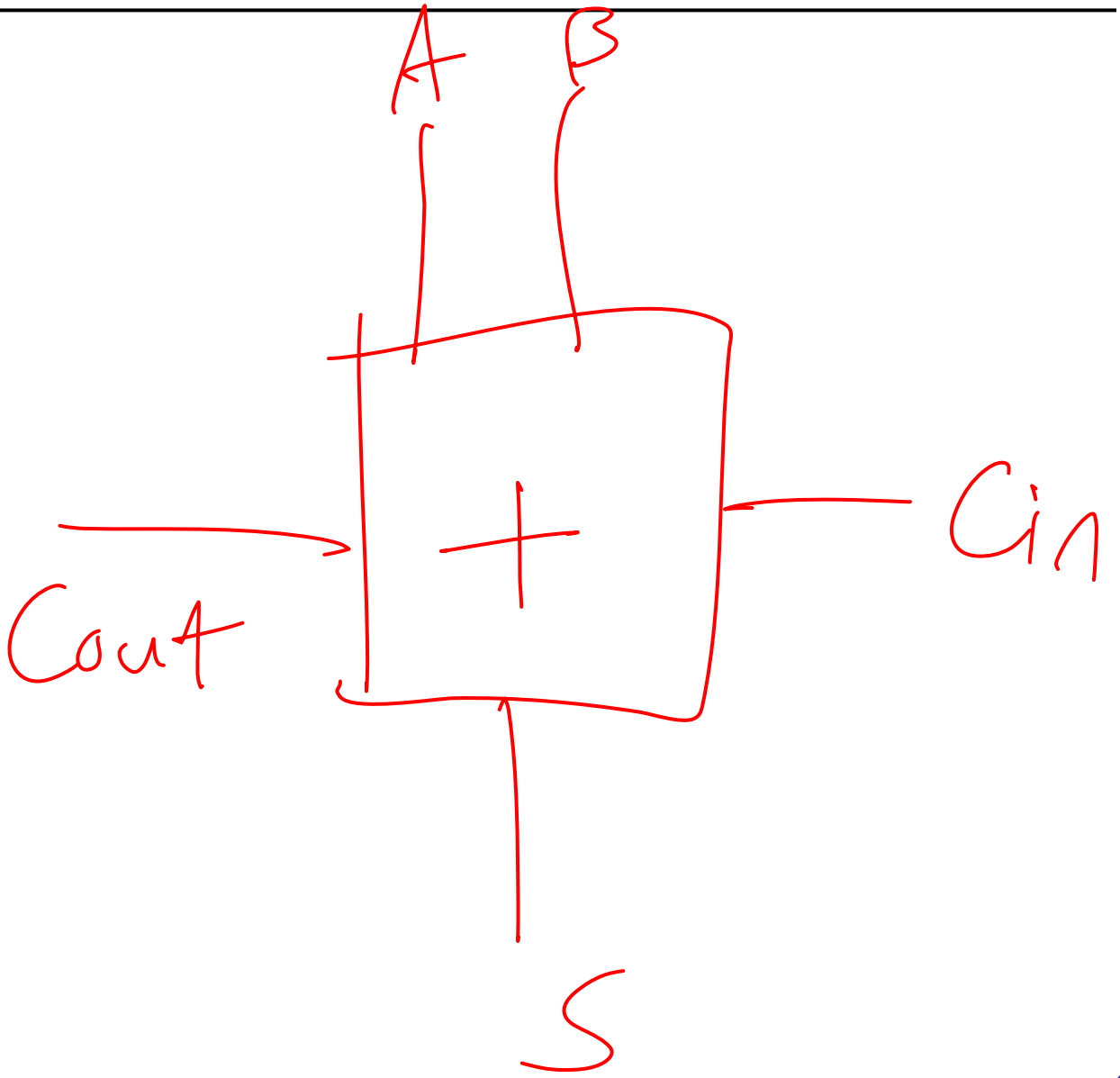
Fall 2006

Computer Arithmetic

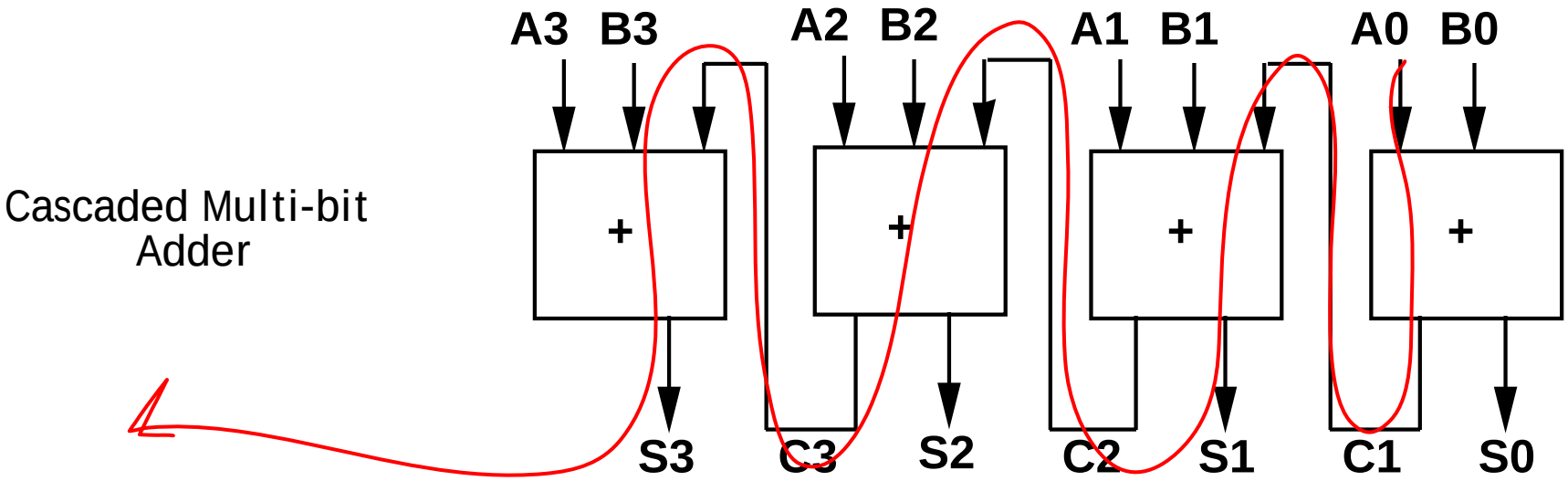
- Readings: Chapter 3
- Review binary numbers, 2's complement
- Develop Arithmetic Logic Units (ALUs) to perform CPU functions.
- Introduce multiplication, division, floating point.

Full Adder

A	B	CI	CO	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Multi-Bit Addition

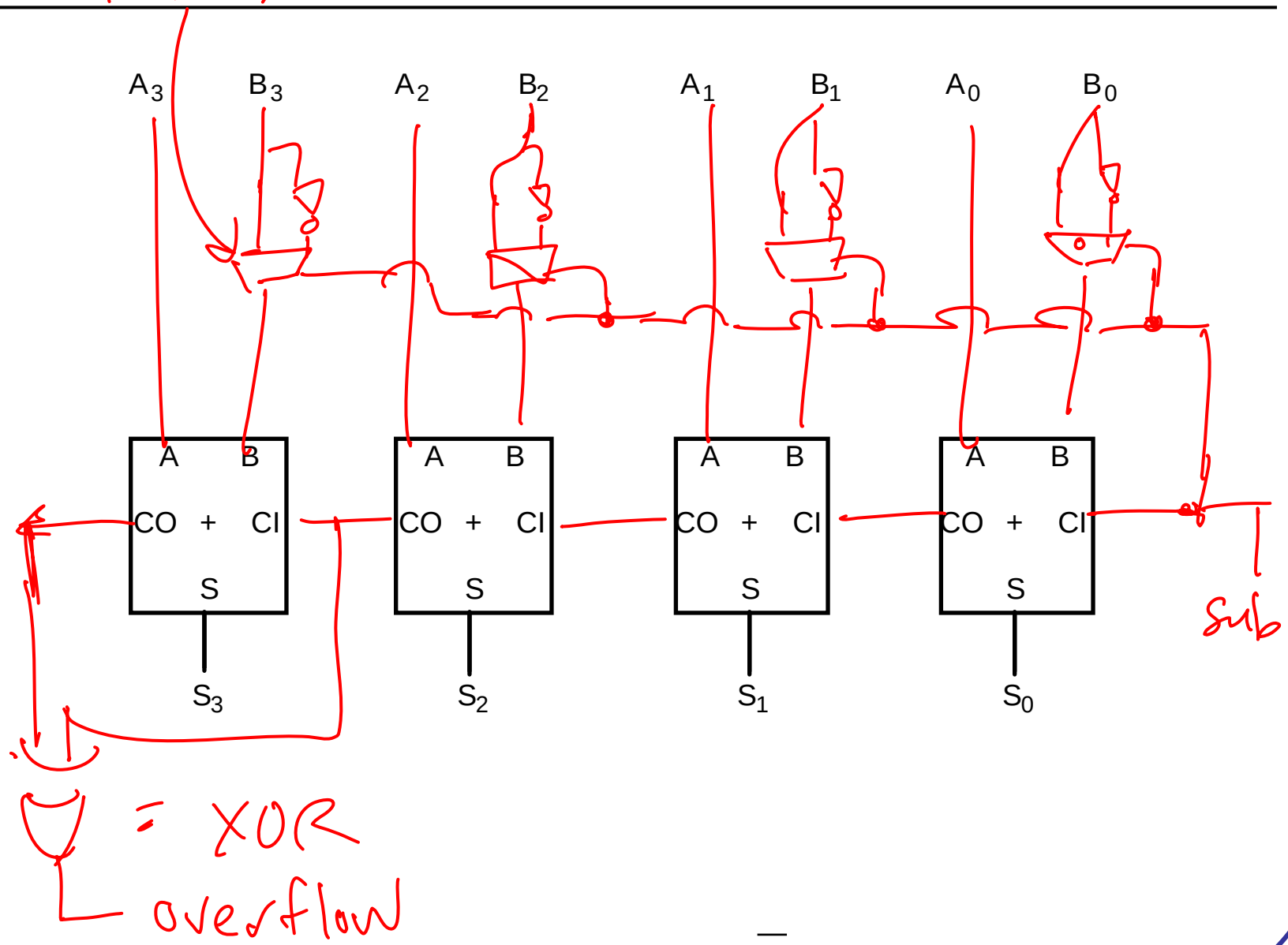


Cascaded Multi-bit Adder

usually interested in adding more than two bits

this motivates the need for the full adder

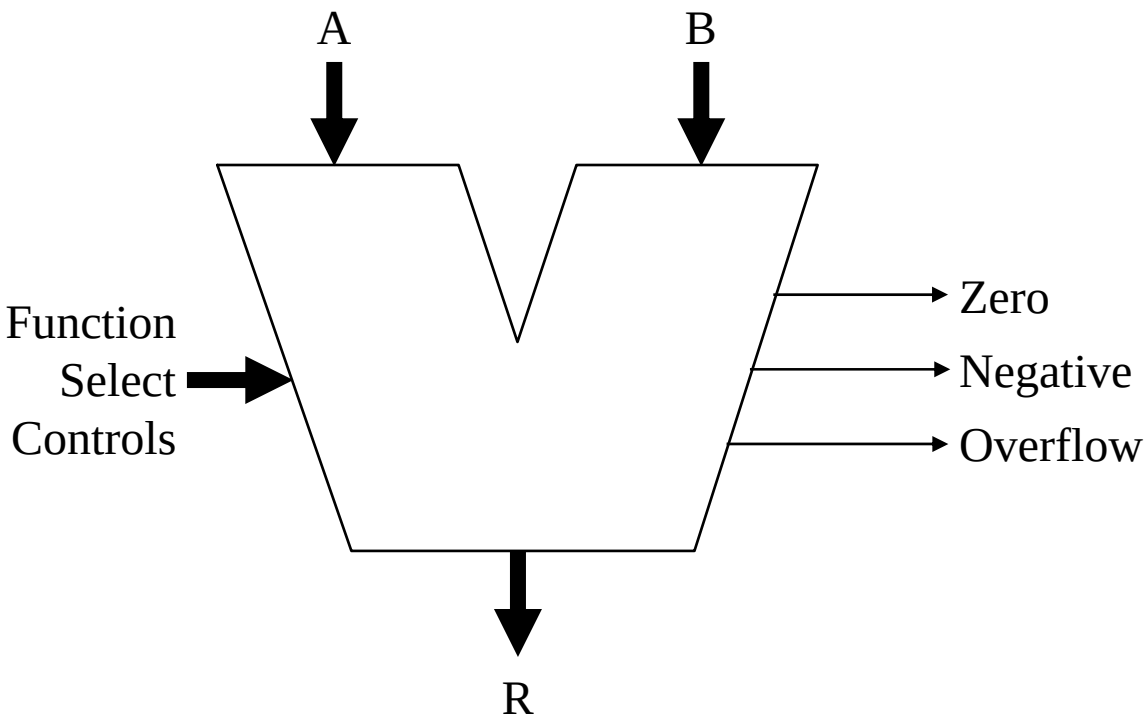
Adder/Subtractor



A - B = A + (-B) = A + $\overline{B}$ + 1

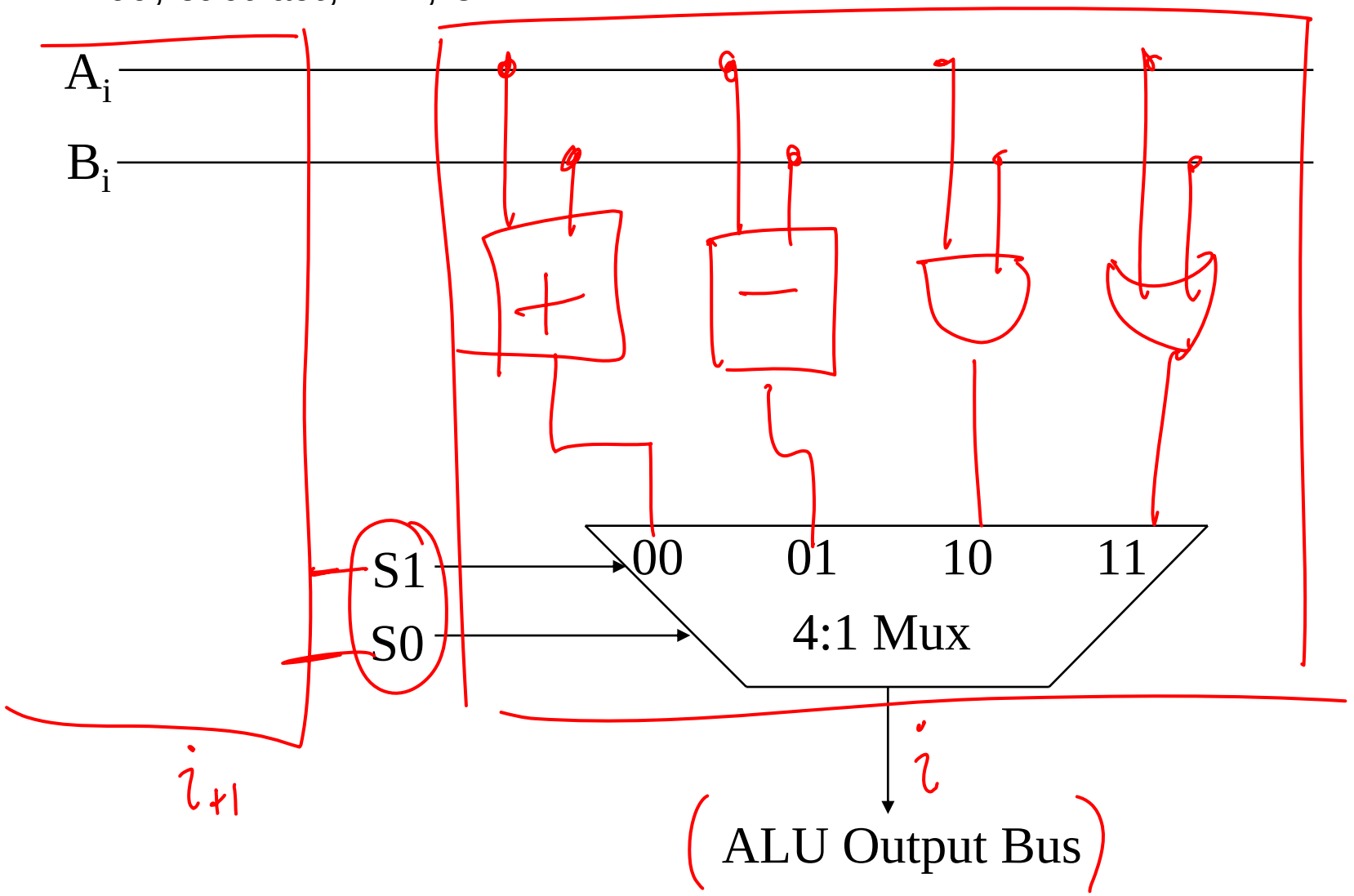
ALU: Arithmetic Logic Unit

- Computes arithmetic & logic functions based on controls
  - Add, subtract
  - XOR, AND, NAND, OR, NOR
  - ==, <, overflow, …



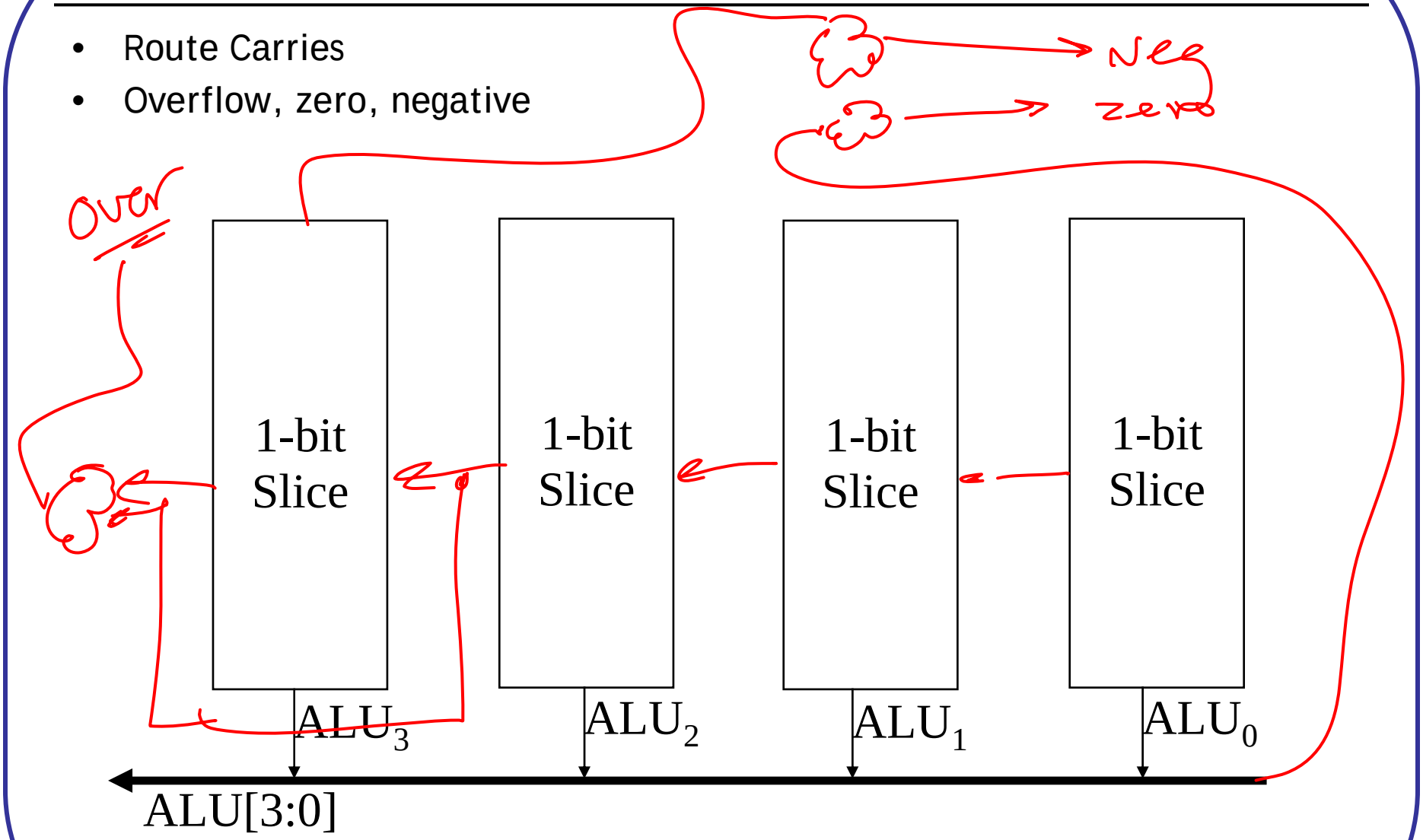
Bit Slice ALU Design

- Add, Subtract, AND, OR



Bit Slice ALU Design (cont.)

- Route Carries
- Overflow, zero, negative



SLT

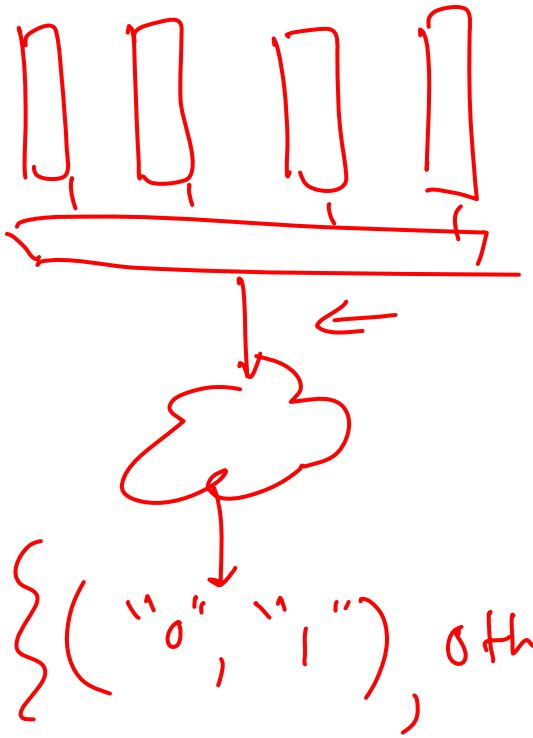
- Set less than: if (A<B) then R = 1, else R = 0
  - How do we know if (A<B)
  - Interaction w/overflow

Sub & look at neg bit

Do: A − B

if (neg) → set

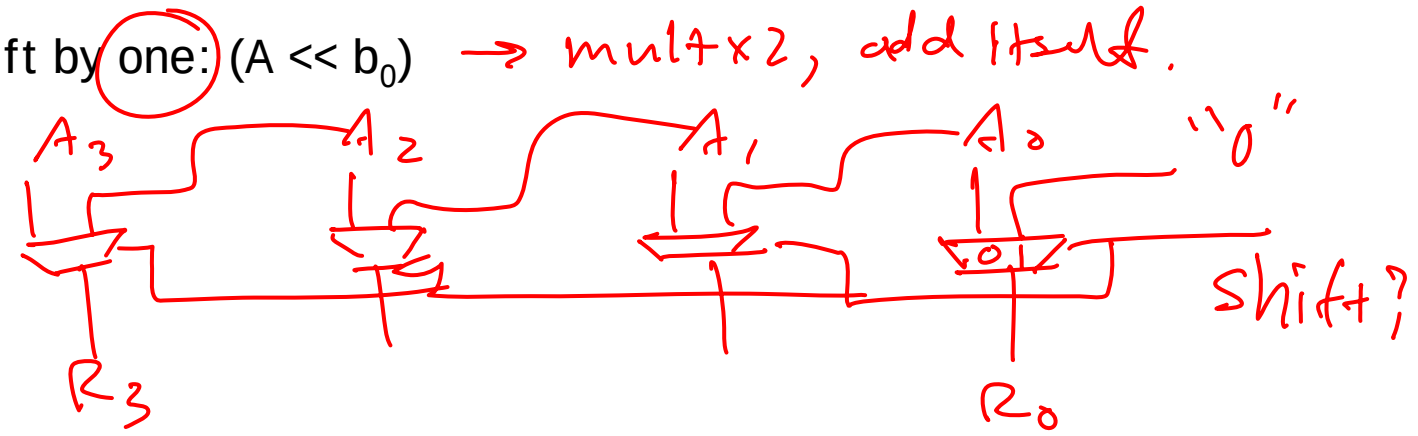
else → 0



Check overflow

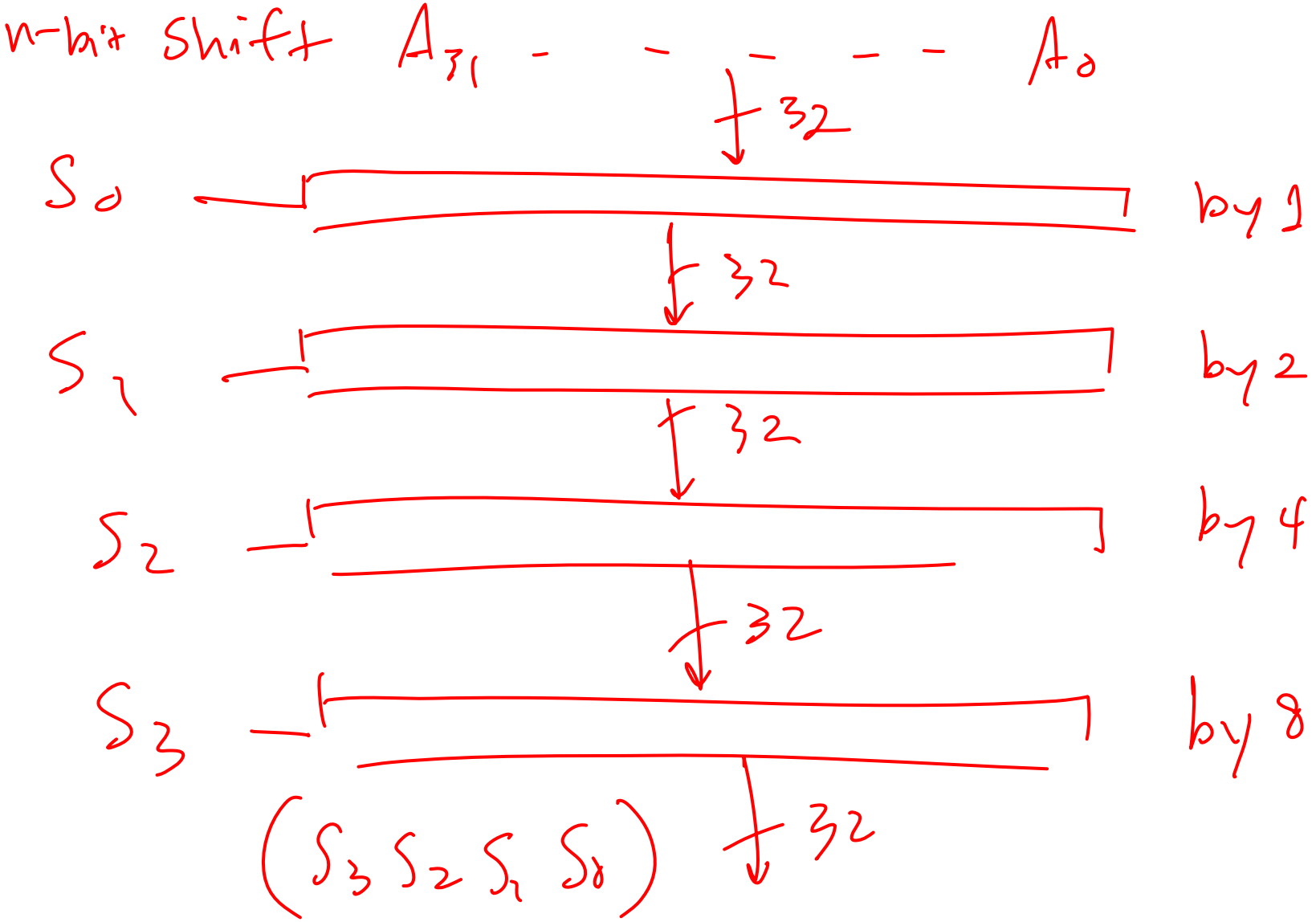
Shifter

- Support shift operations: (A << 01101)
- Optional shift by one: (A << b_0)



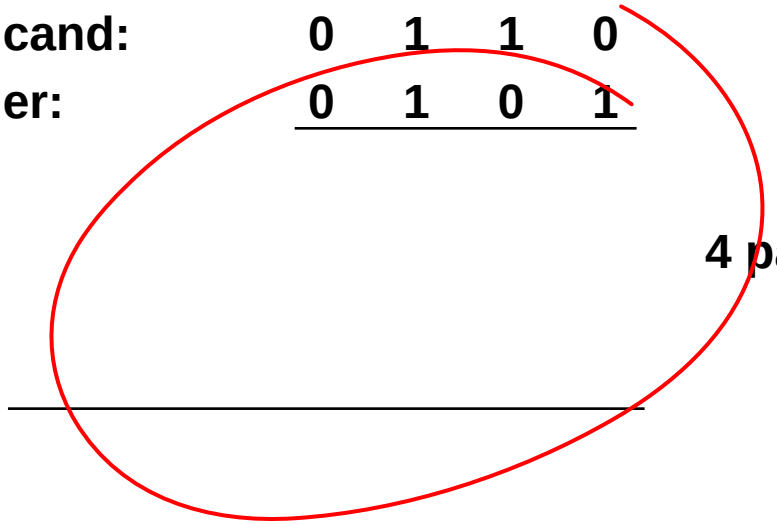
- Optional shift by two: (A << b_1)

Left Shifter (cont.)



Multiplication

- Example

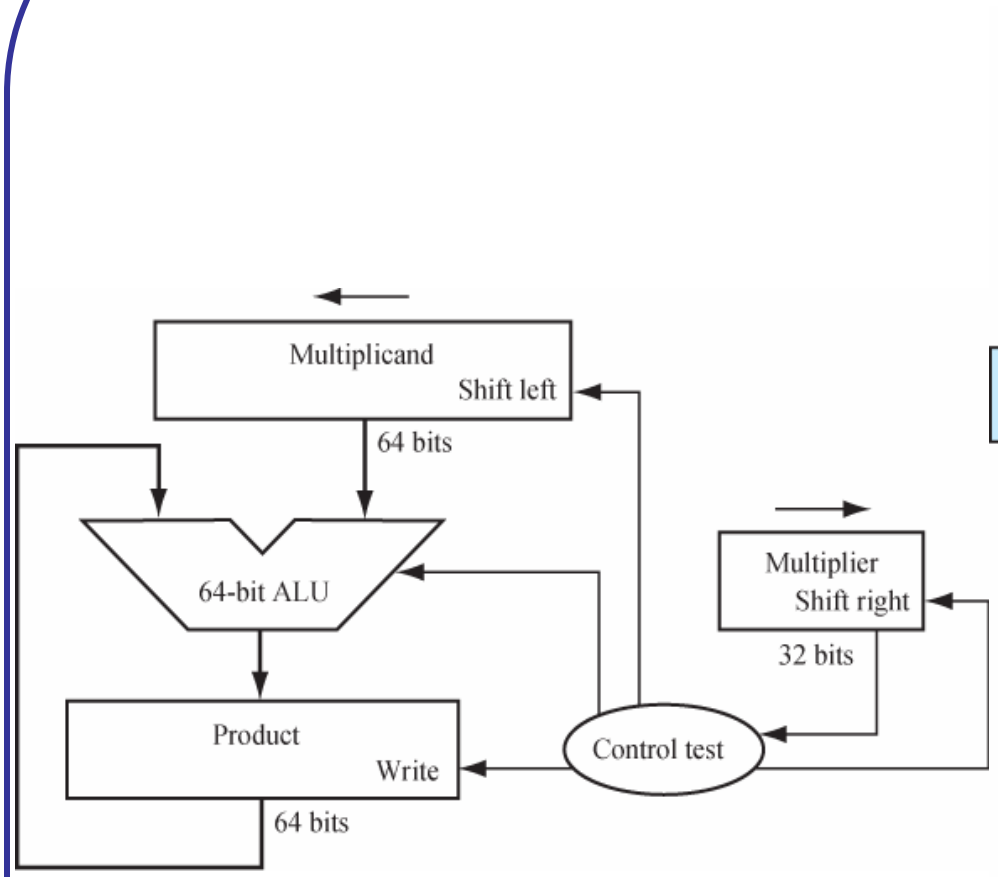


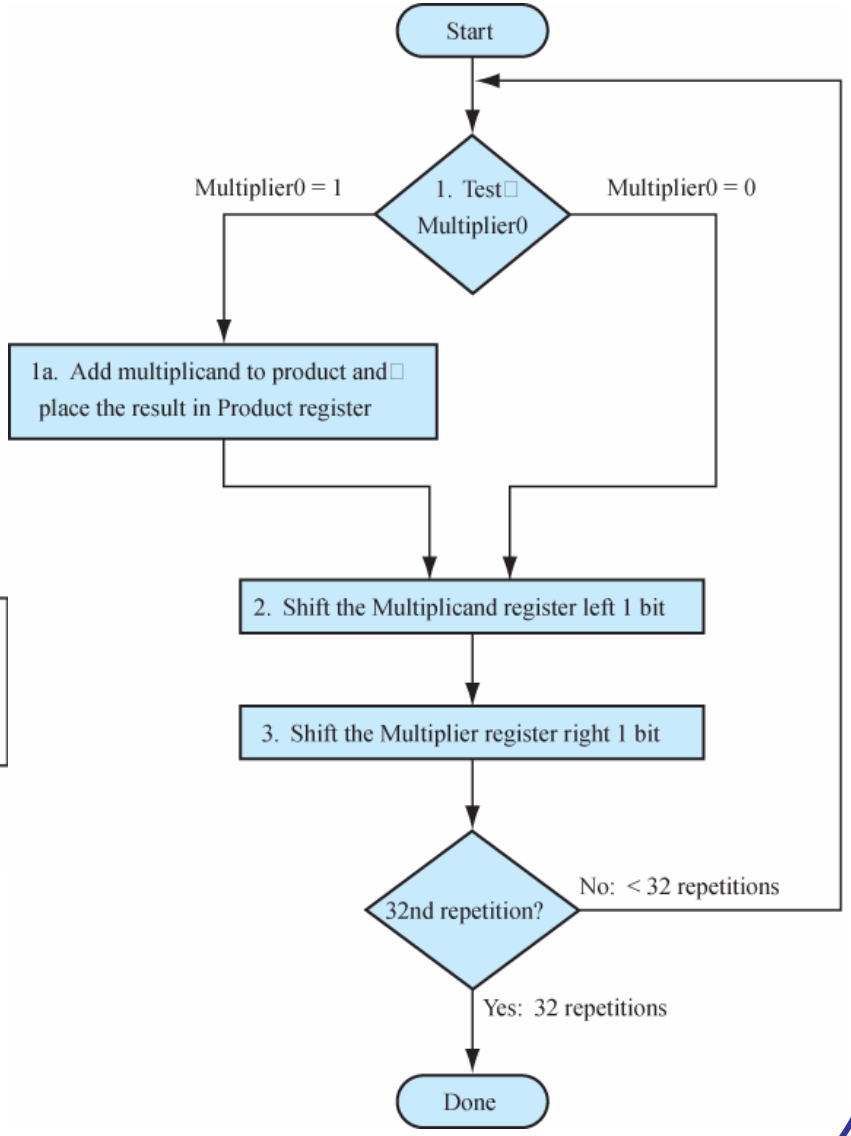
Repeat n times:

Compute partial product; shift; add

NOTE: Each bit of partial products is just an AND operation

Sequential Multipliers



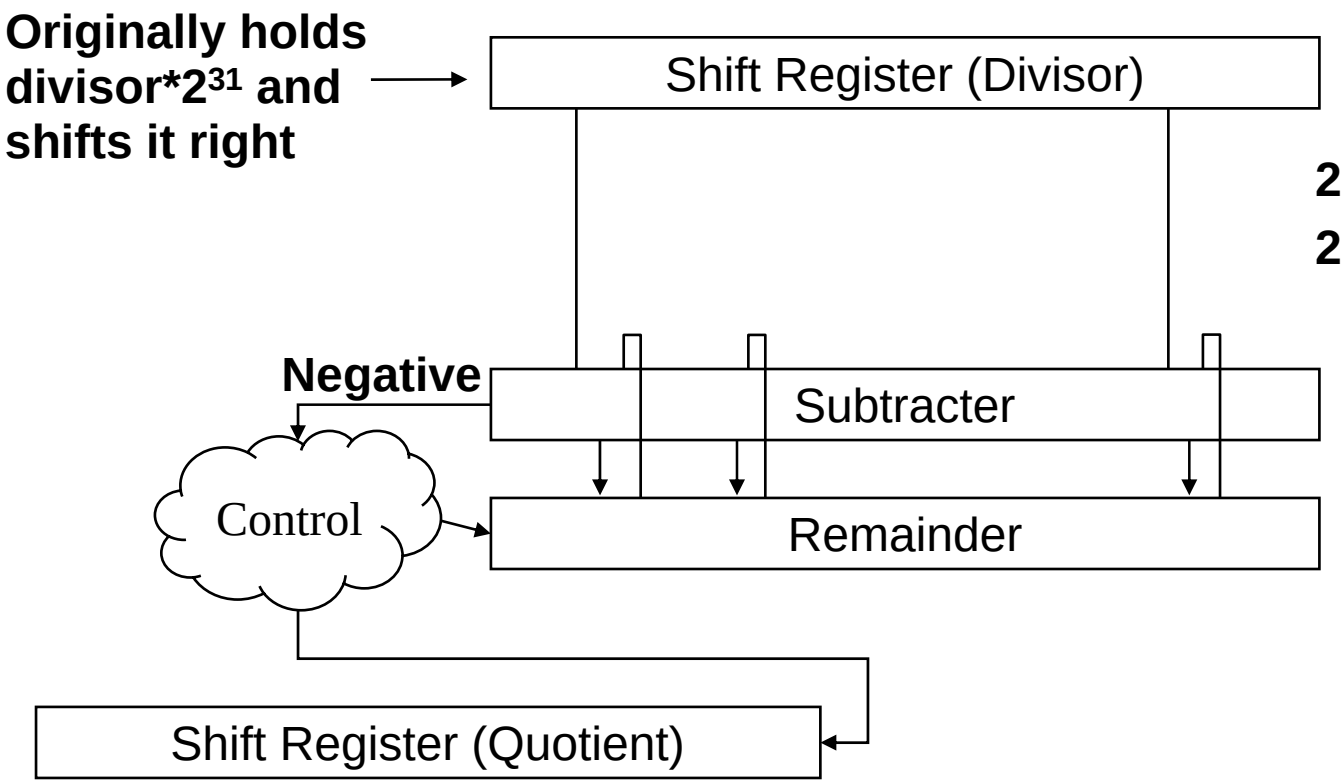


Division

- Example

Divisor: 0010 (2) $\overline{)\ 0\quad 1\quad 1\quad 0}$ **(6) Dividend**

Sequential Dividers



Remainder

Alternatively:

Shift remainder register to left

Use only n-bit subtracter

Floating Point

Want to represent numbers outside $2^{31}-1$ .. -2^{31}

Ideal: ~Scientific Notation

(+/-)Significand*Base^{Exponent} $5.439 * 10^{12}$ $1.010010 * 2^{100101}$

Multiplication:

$(5.1 * 10^{12}) * (-2.0 * 10^{-3})$

Sign: (xor) (-)

Exponent: exp1 + exp2 (signed) (9)

Significand: $S_1 \times S_2$ if ≥ Base, normalize

10.2 → -1.02×10^{10}

Floating Point Addition

Addition: $(5.38 * 10^5) + (4.99 * 10^5)$

Normalize first

$(9.99 * 10^4) + (-1.0 * 10^5)$

$-10 \times 10^4 + 9.99 \times 10^4$

Sign: whichever is bigger

\+

Exponent: Same 5

Significand: Do add/sub

10.37×10^5

1.037×10^6

\-

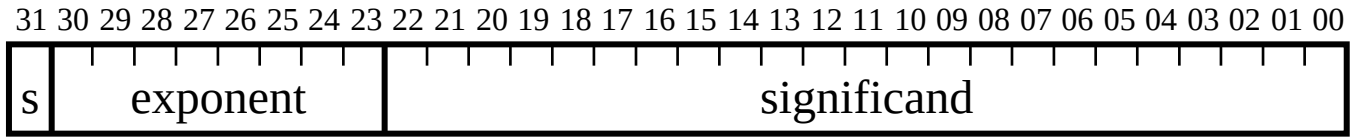
4

$-.01 \times 10^4$

-1.0×10^2

Floating Point Representation

- Floating Point (Float) = $(-1)^s * (1.\text{significand}) * 2^{(\text{exponent}-127)}$



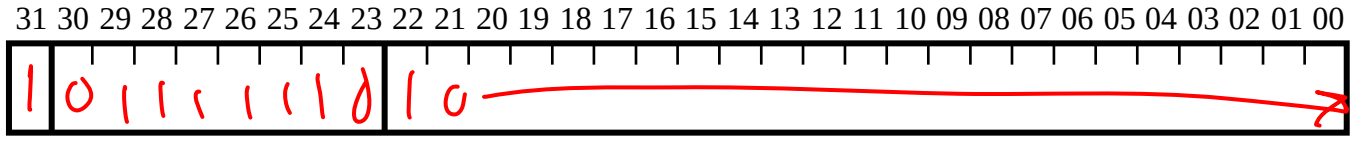
- -0.75 in Floating Point?

$-(2^{-1}+2^{-2})_{10} = -.75_{10} = -(0.11)_2 \times 2^0 = -(1.1)_2 \times 2^{-1}$

exp $(126)_{10}$

$2^{-1}+2^{-2}$

$\frac{1}{2}+\frac{1}{4}$

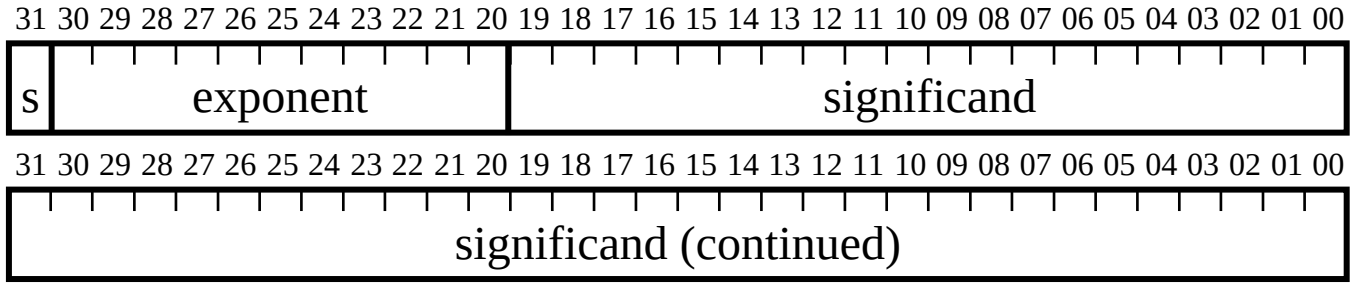


Sign

Sig =

Double Precision Representation

- Double Precision (double) = $(-1)^{s} * (1.\text{significand}) * 2^{(\text{exponent}-1023)}$



- -0.75 in Double Precision?