110
Arithmetic

ENGR 3410 – Computer Architecture

Mark L. Chang

Fall 2007

Computer Arithmetic

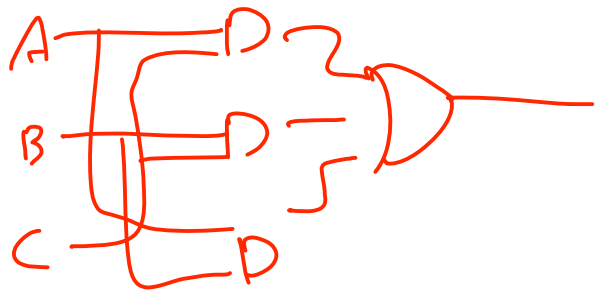
- Readings: Chapter 3
- Review binary numbers, 2's complement
- Develop Arithmetic Logic Units (ALUs) to perform CPU functions.
- Introduce multiplication, division, floating point.

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

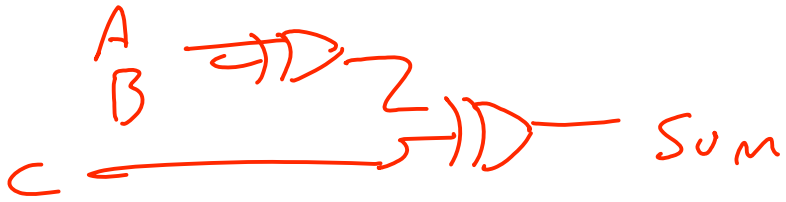
$CO = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

$= BC + AC + AB$



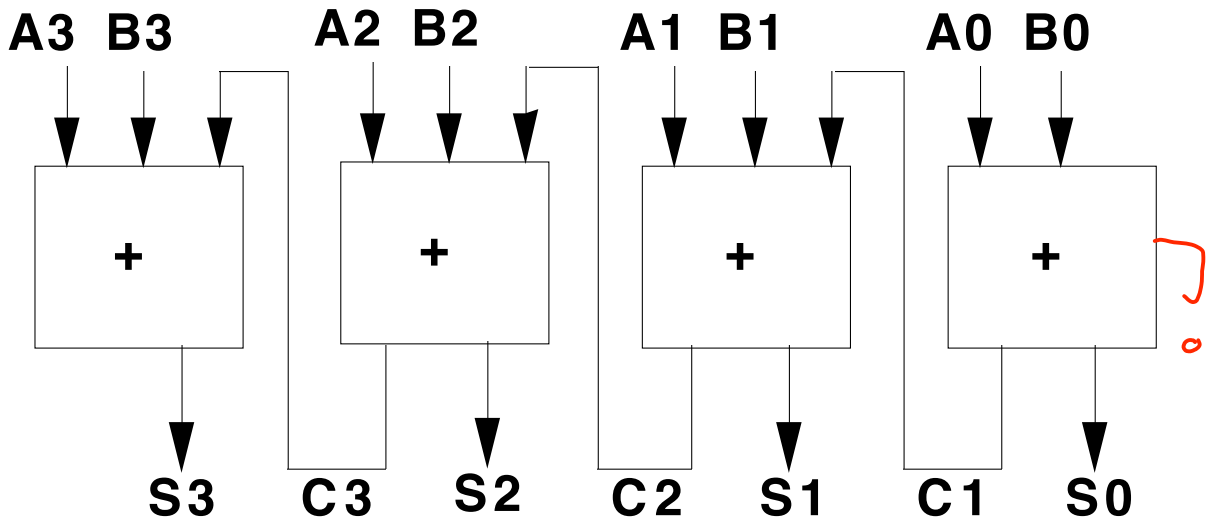
$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

$= A \oplus B \oplus C$



Multi-Bit Addition

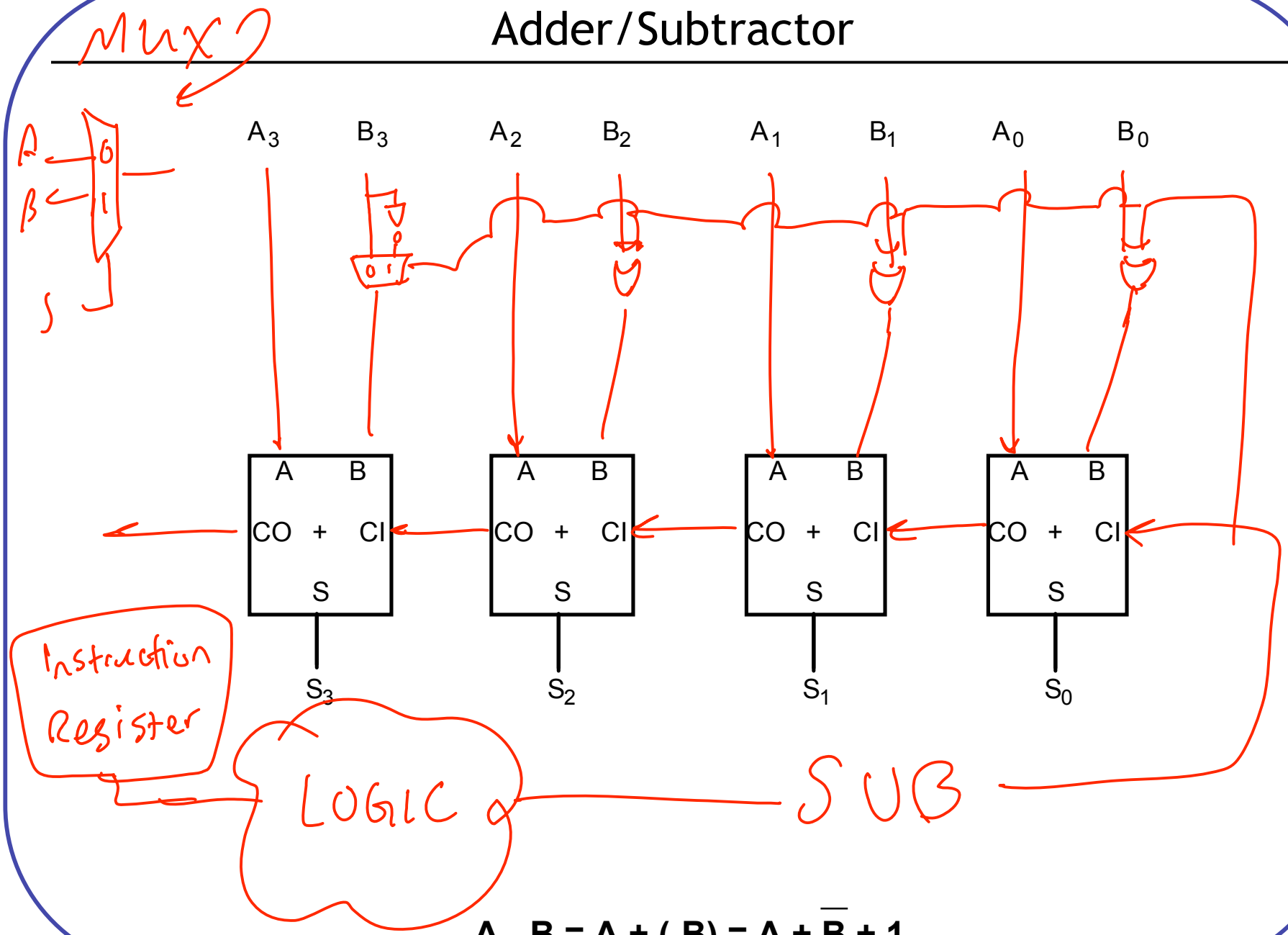
Cascaded Multi-bit Adder



usually interested in adding more than two bits

this motivates the need for the full adder

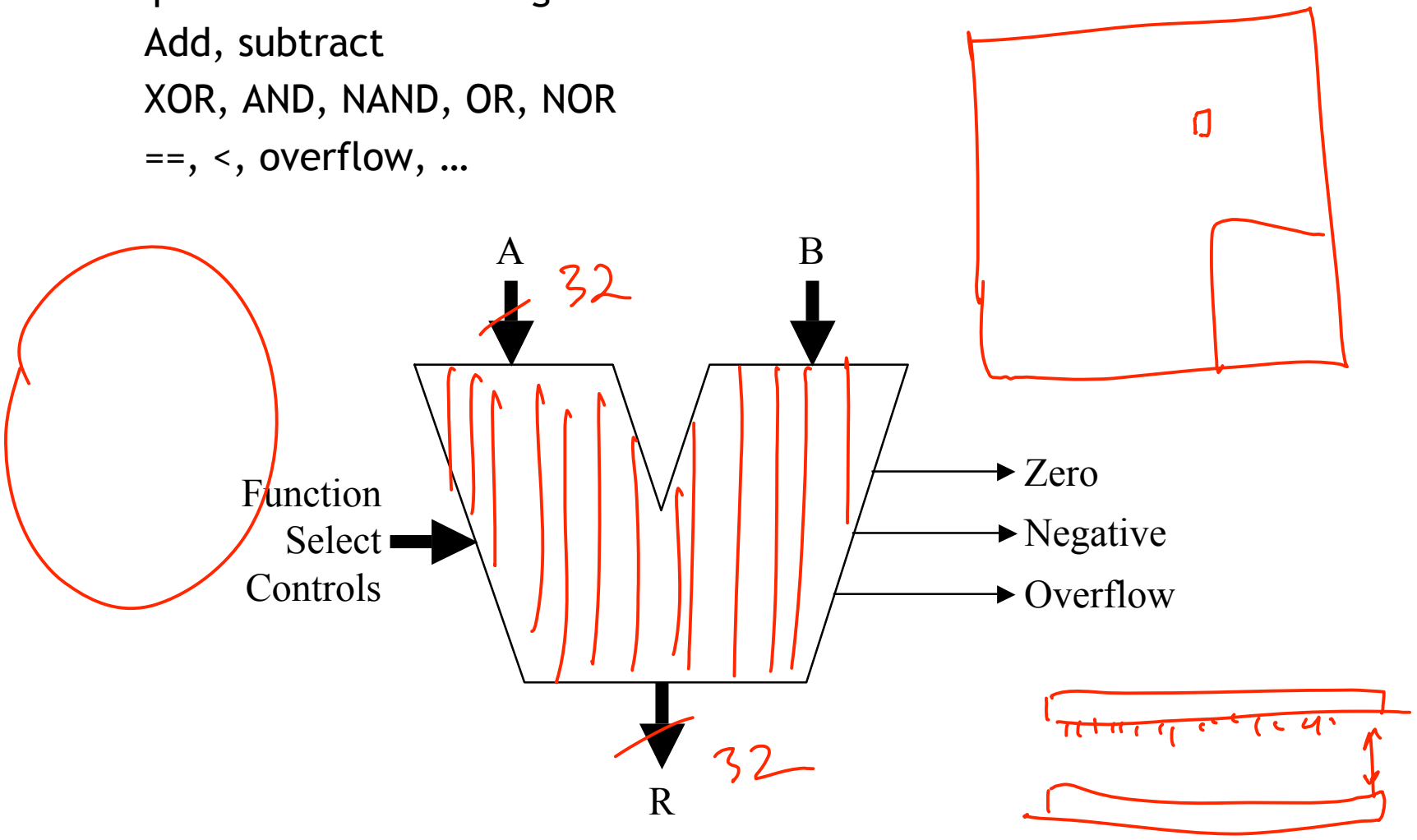
Adder/Subtractor



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

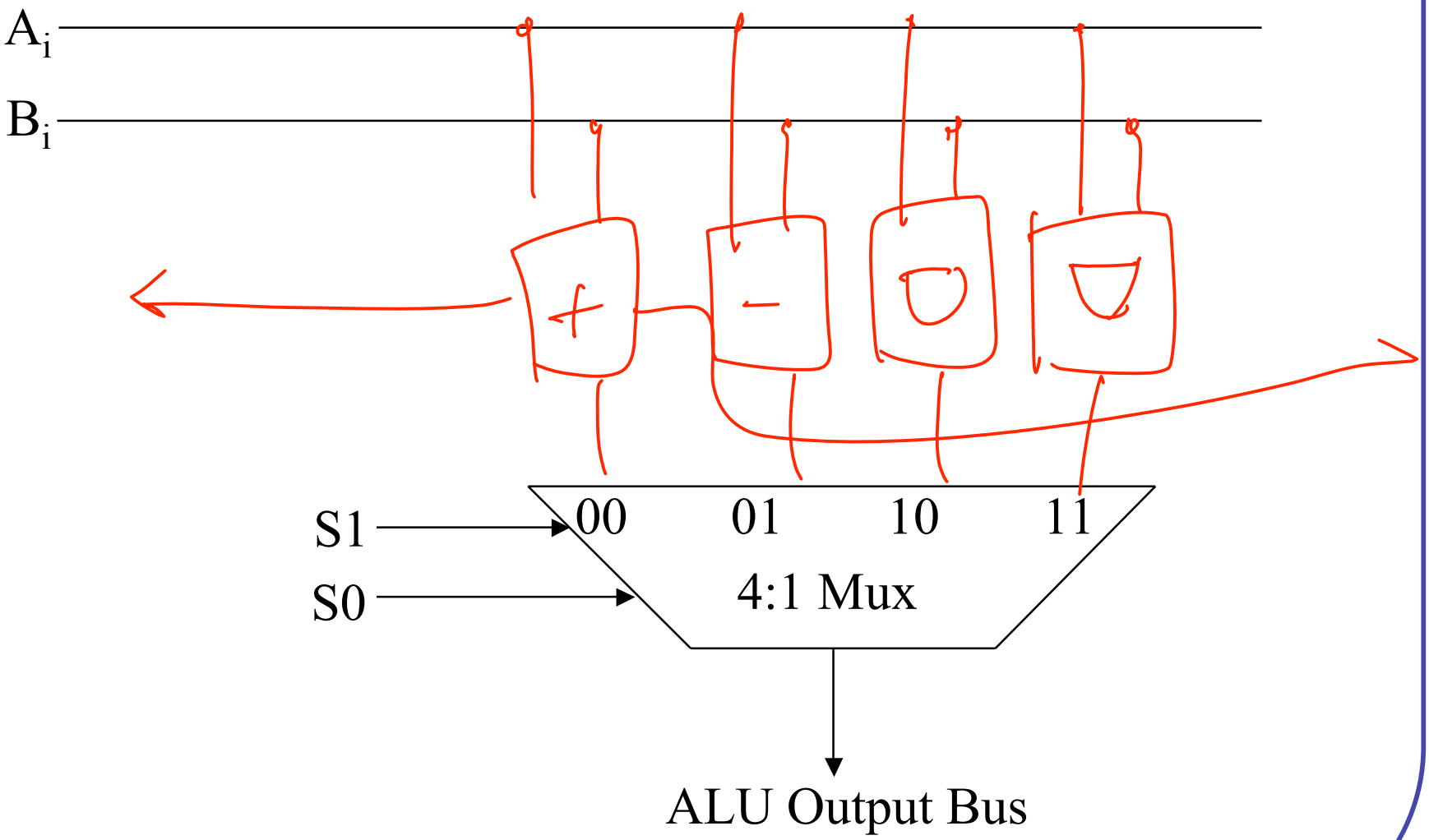
ALU: Arithmetic Logic Unit

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



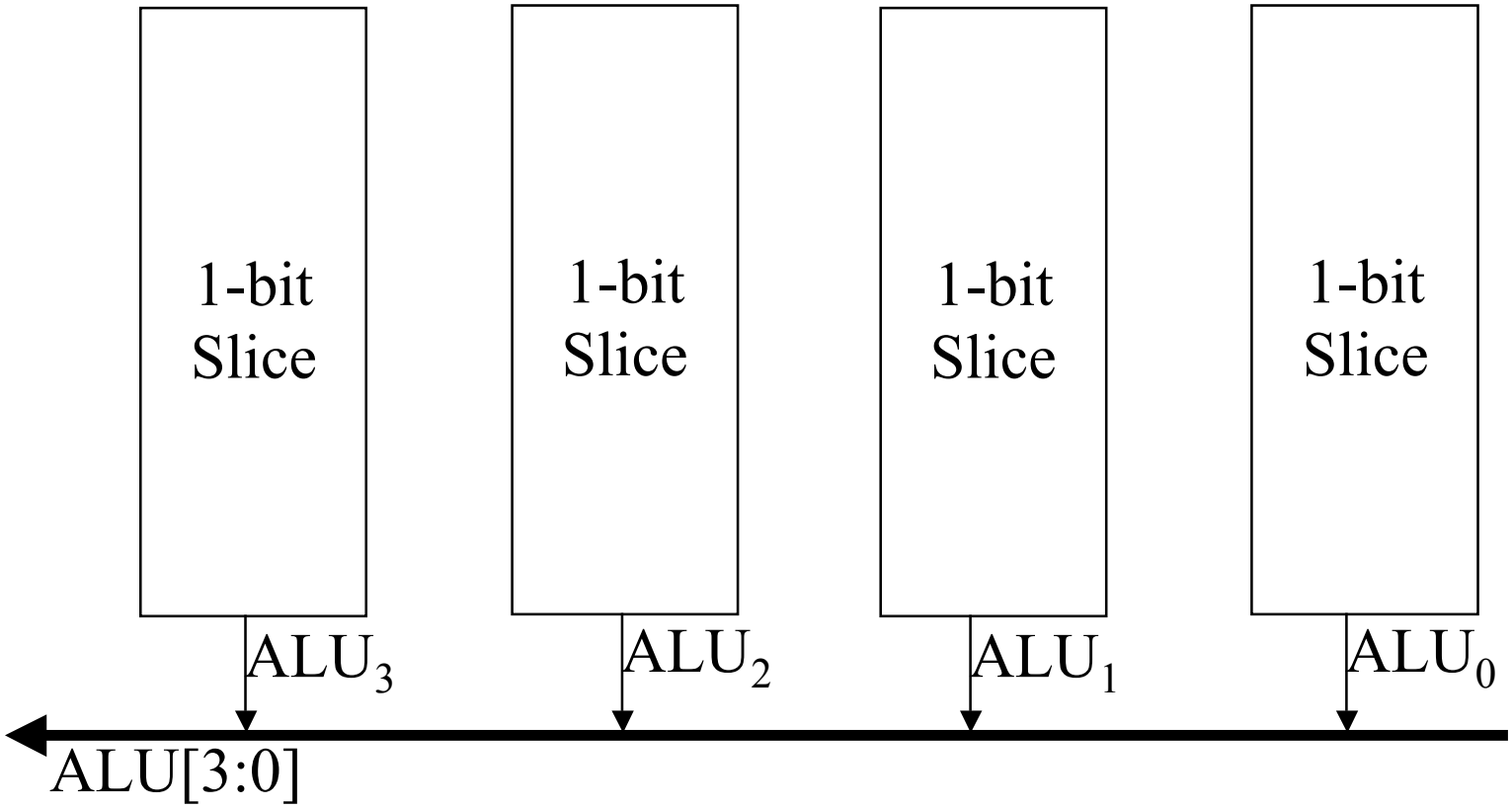
Bit Slice ALU Design

- Add, Subtract, AND, OR



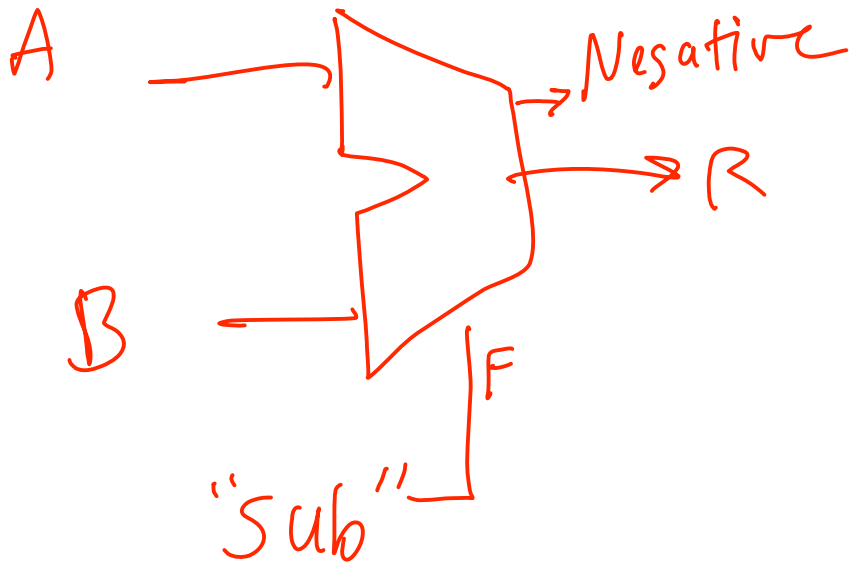
Bit Slice ALU Design (cont.)

- Route Carries
- Overflow, zero, negative



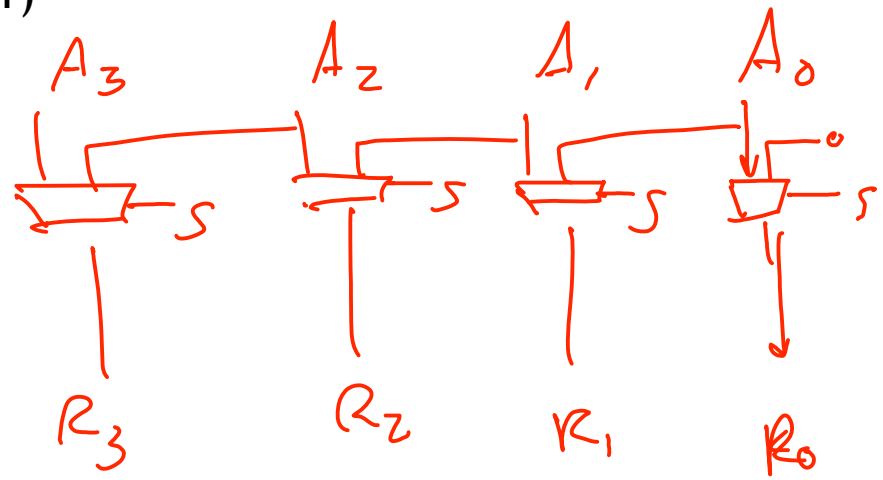
SLT

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



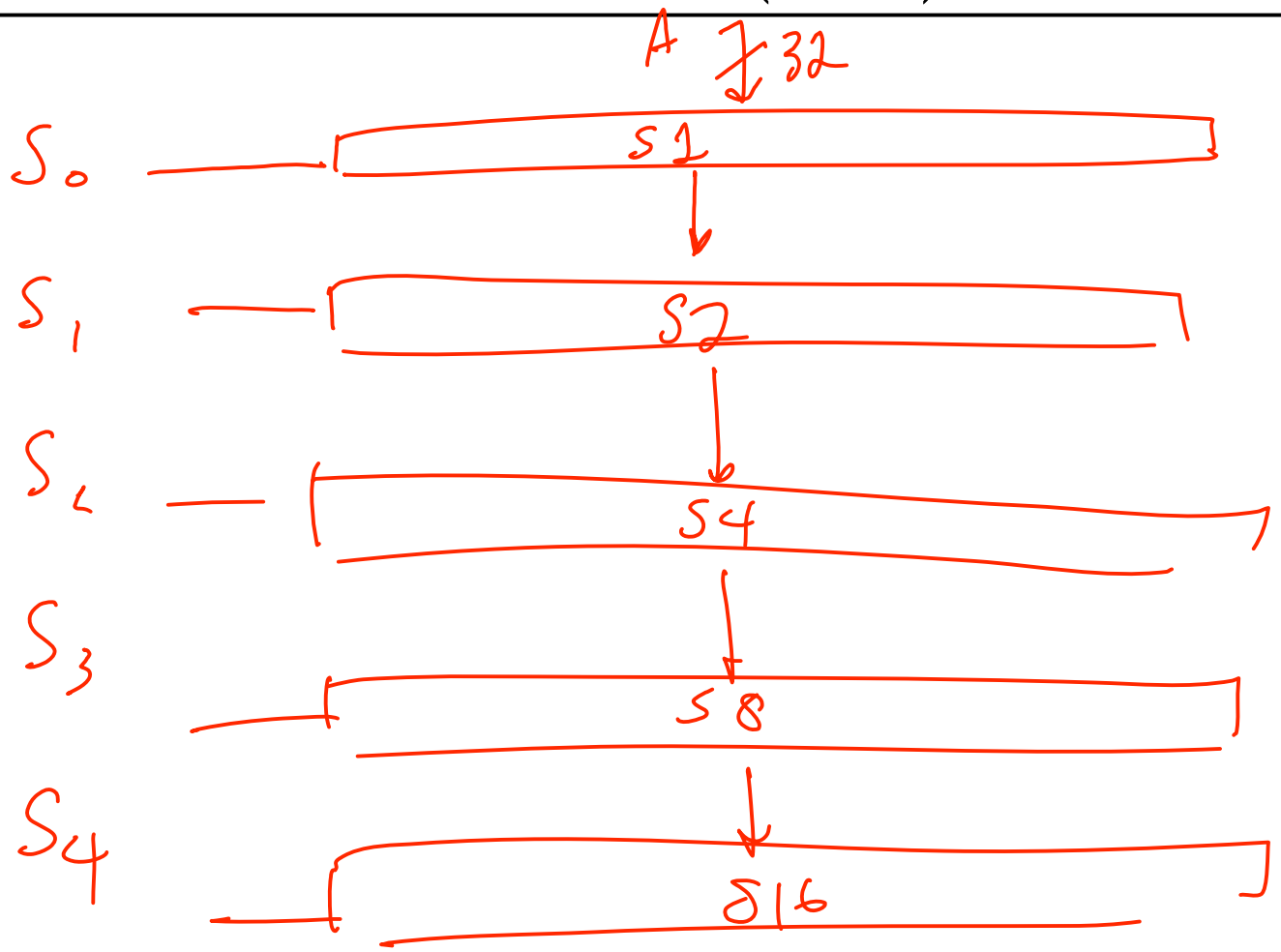
Shifter

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



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

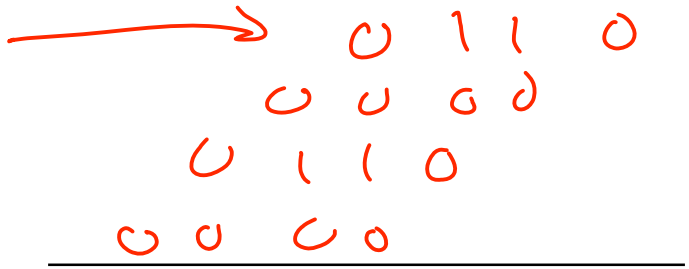
Shifter (cont.)



Multiplication

- Example

Multiplicand:	**0 1 1 0**	**6**
Multiplier:	**0 1 0 1**	**5**



4 partial products

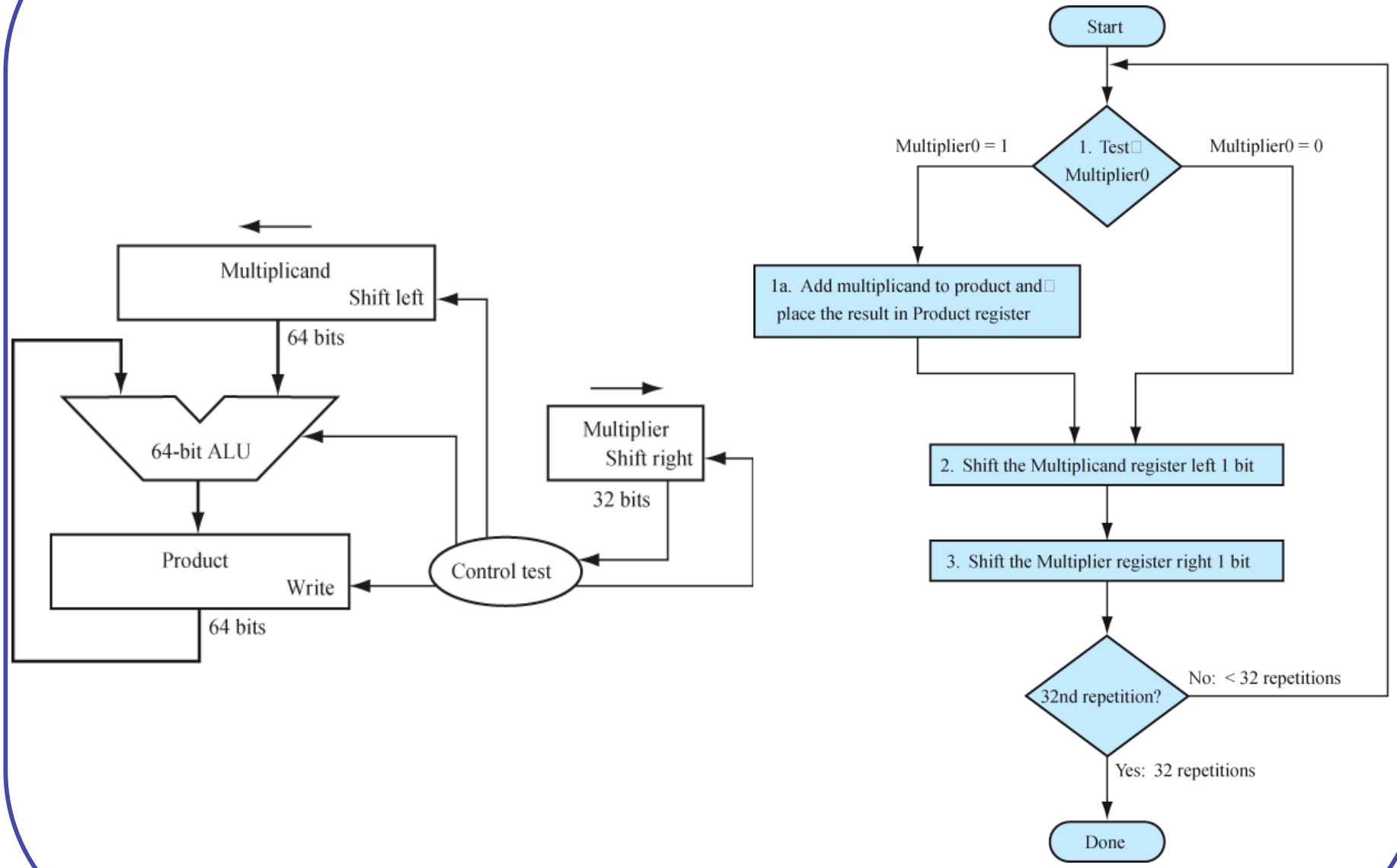
30

Repeat n times:

Compute partial product; shift; add

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

Sequential Multipliers

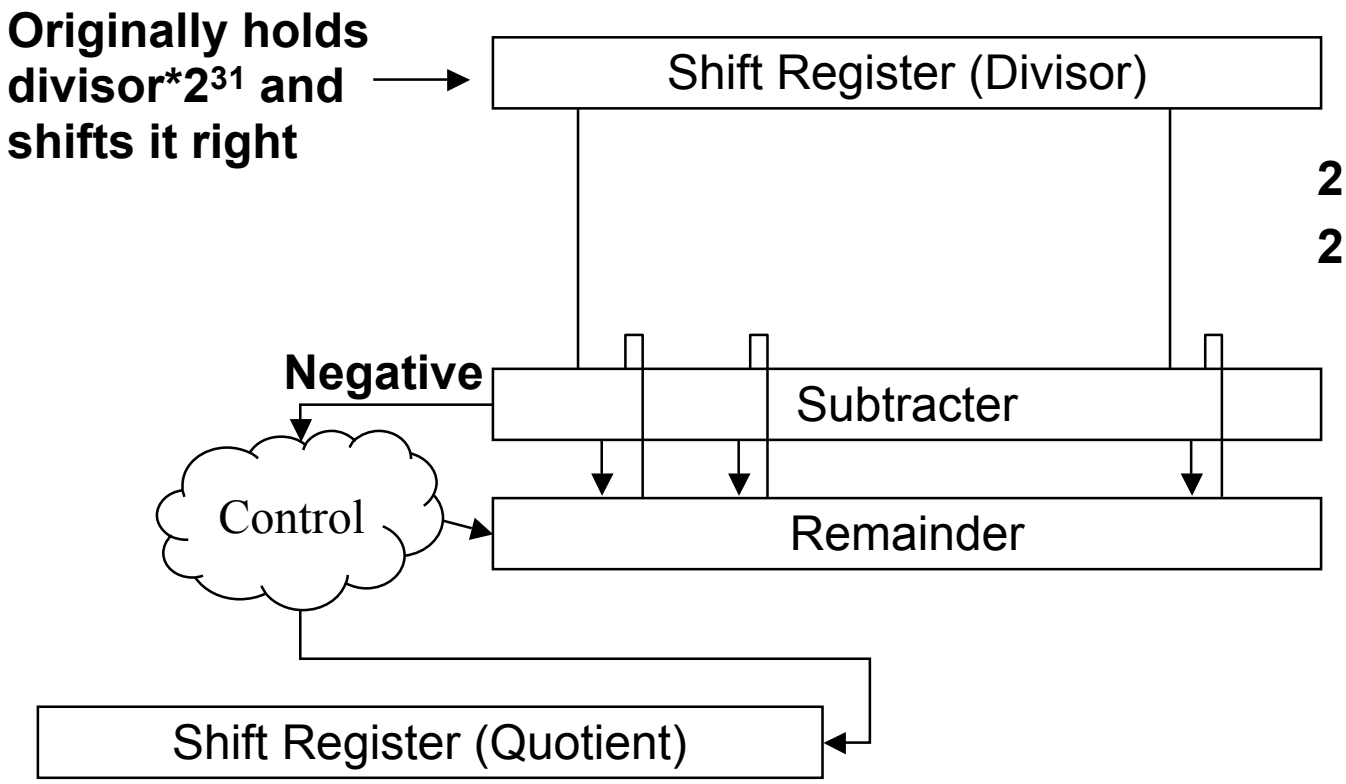


Division

- 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$^{\text{Exponent}}$ $5.439 * 10^{12}$ $1.010010 * 2^{100101}$

Multiplication:

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

Sign: (-)

Exponent: Add exponents = 9

Significand: $S_1 \times S_2 = 10.2 \rightarrow -1.02 \times 10^{10}$

Floating Point Addition

Addition: $(5.38 * 10^5) + (4.99 * 10^5)$ $(9.99 * 10^4) + (-1.0 * 10^5)$

$5.38 \times 10^5 + 4.99 \times 10^5$ $0.999 \times 10^5 + -1.0 \times 10^5$

Sign: Bigger value (+) (−)

Exponent: Bigger value 5 5

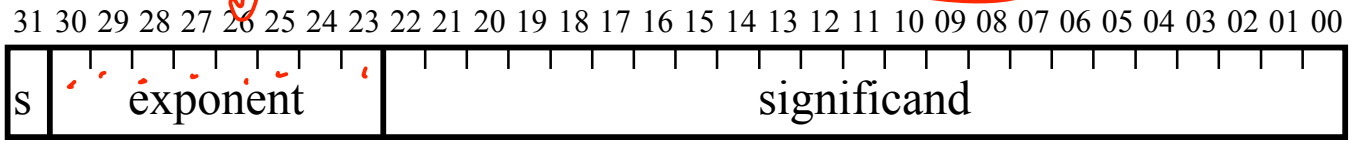
Significand: Do add/sub 0.001

10.37 1.0×10^2

1.037×10^6

Floating Point Representation

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



IEEE.754

- -0.75 in Floating Point?

Sign: 1

sig: 1.sig. 1.1000...

exp: -1 = exp-127

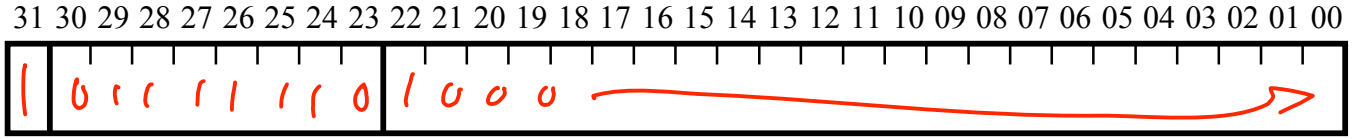
exp = 126

1.0

0.1 = 1/2

0.01 = 1/4

0.11 = .75 → 1.1×2^{-1}

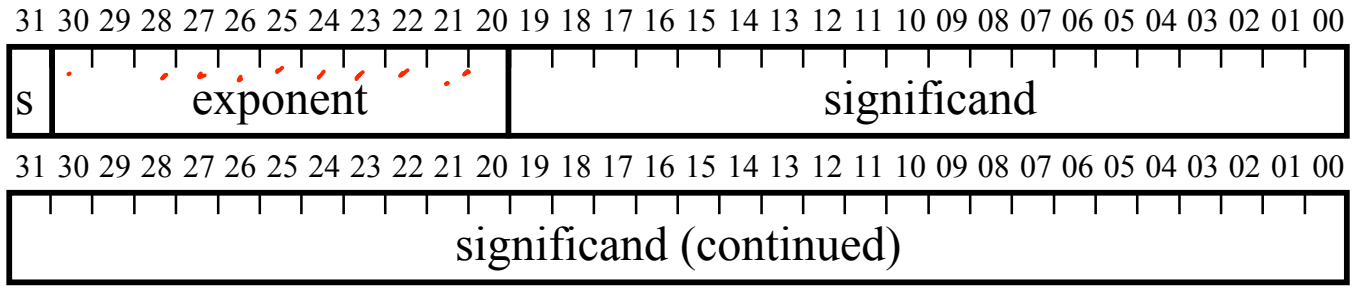


126_{10}

0

Double Precision Representation

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



- -0.75 in Double Precision?