Computer Arithmetic

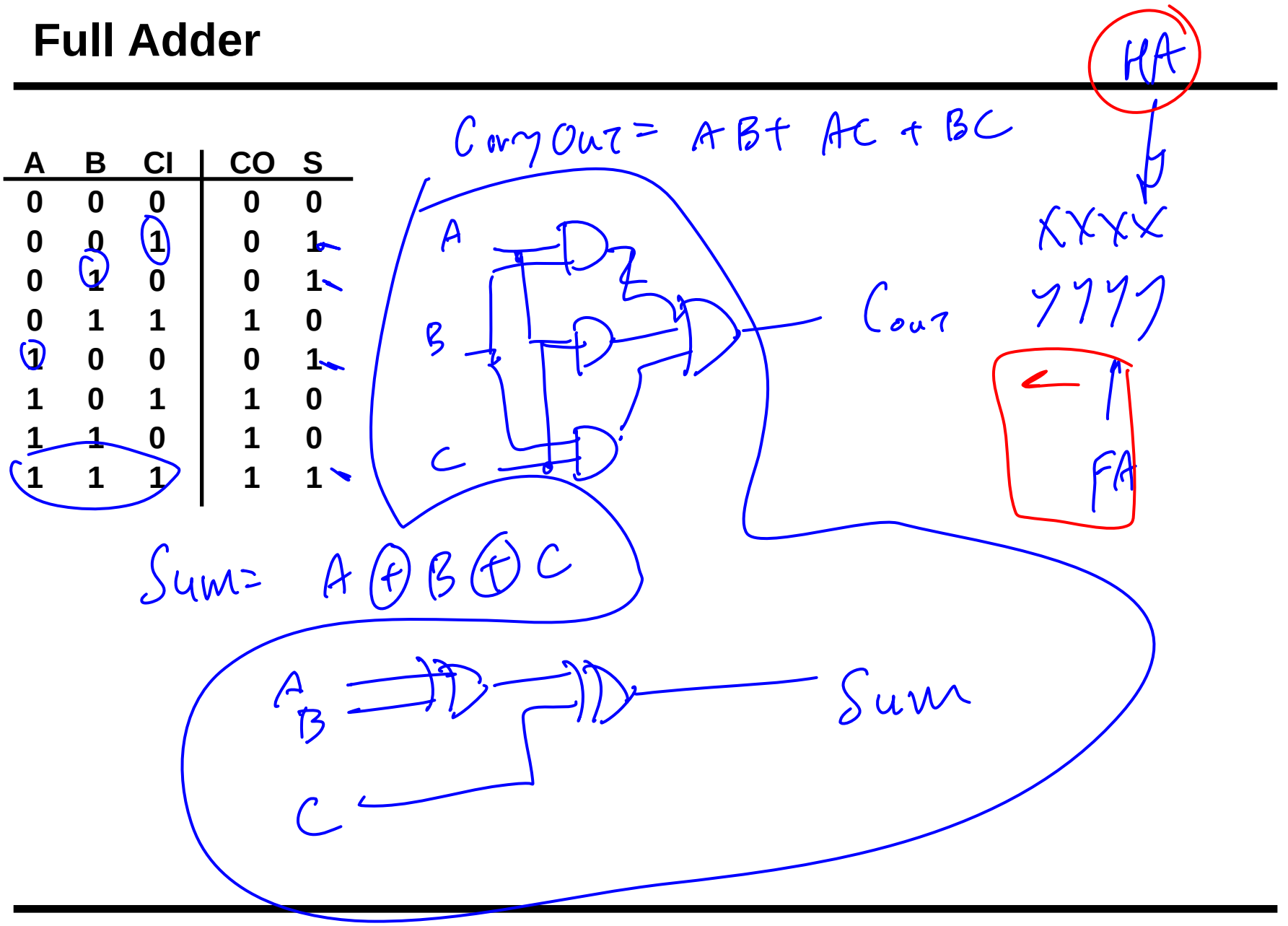
Readings: Chapter 3

Develop Arithmetic Logic Units (ALUs) to perform CPU functions.

Introduce multiplication, division, floating point.

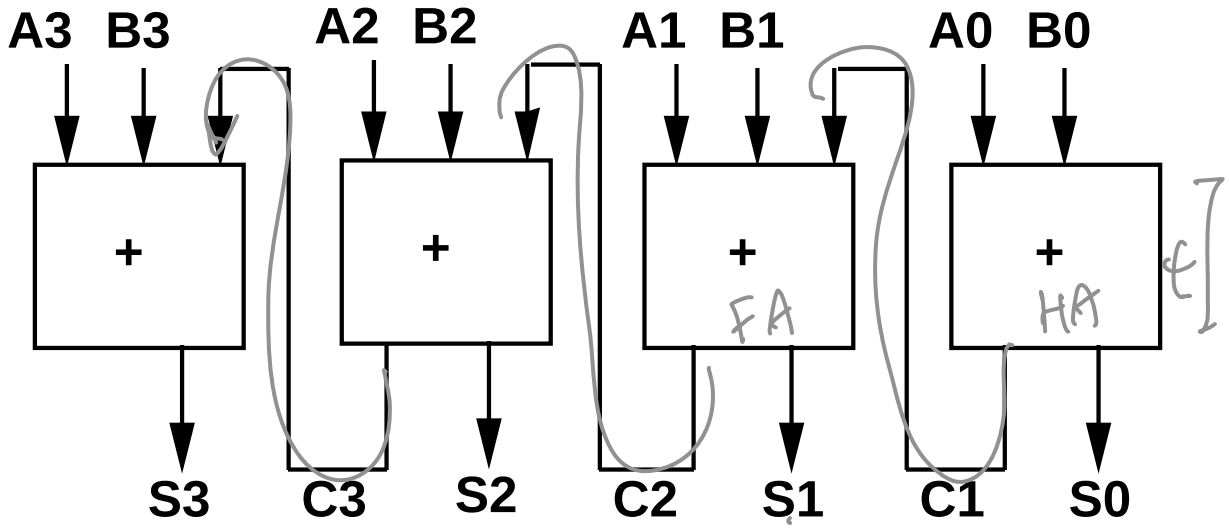
Full Adder

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



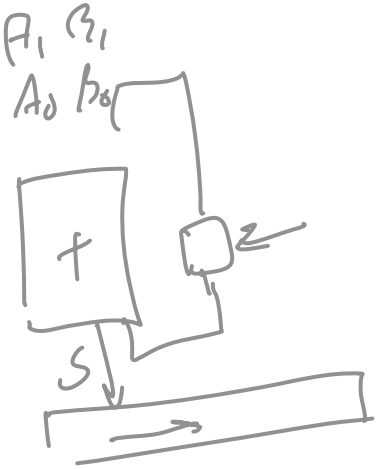
Multi-Bit Addition

Cascaded Multi-bit Adder

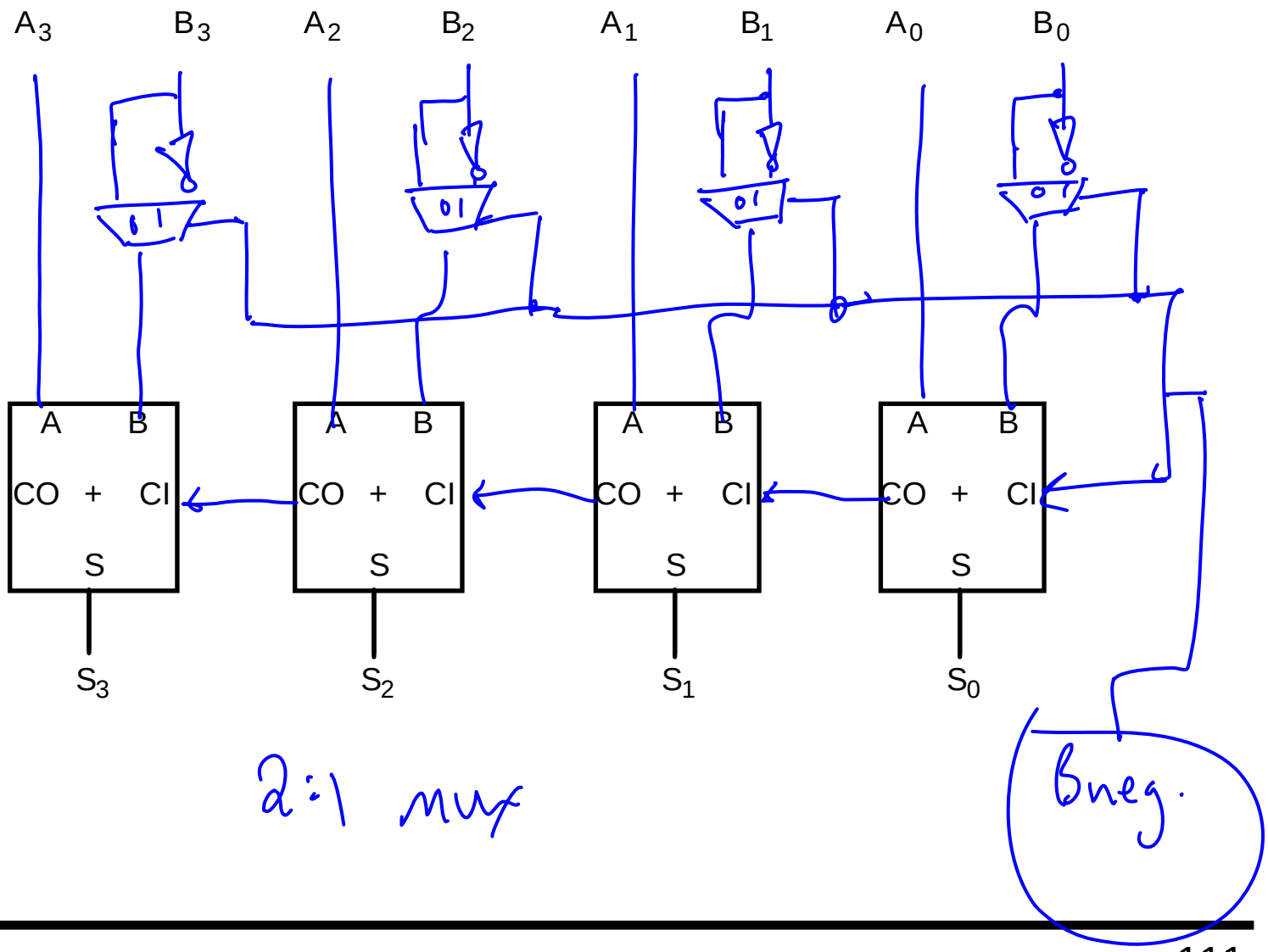


usually interested in adding more than two bits

this motivates the need for the full adder



Adder/Subtractor



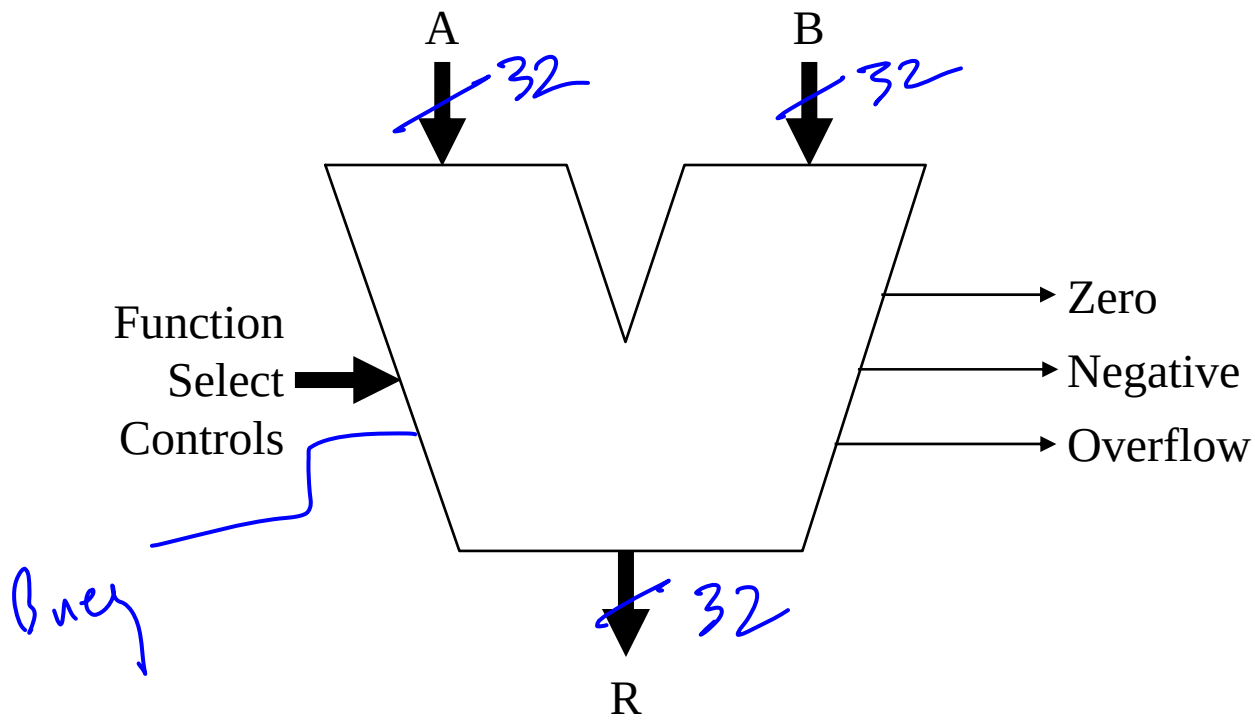
Overflow

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

ALU: Arithmetic Logic Unit

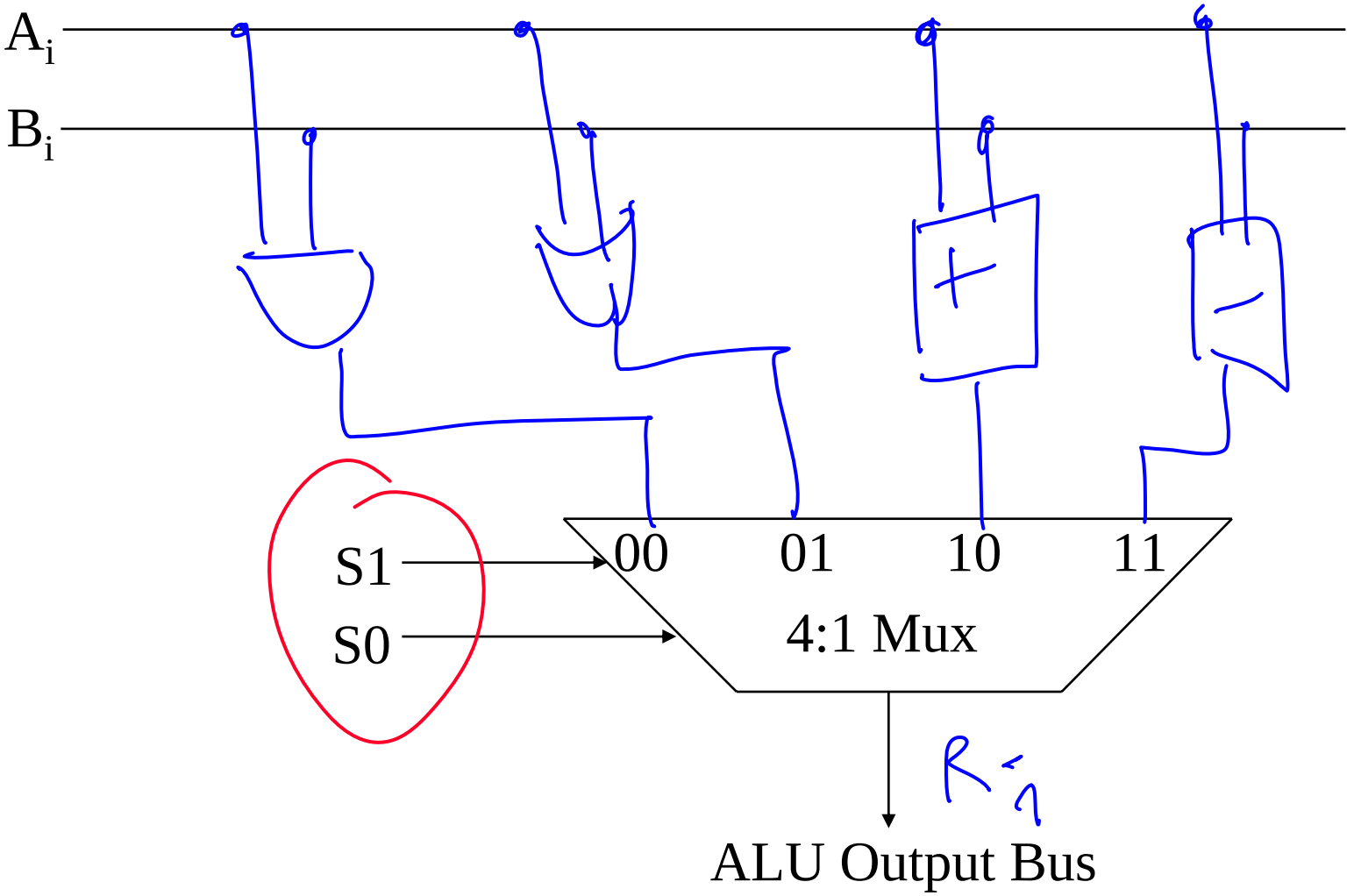
Computes arithmetic & logic functions based on controls

- Add, subtract
- XOR, AND, NAND, OR, NOR
- ==, <, overflow, …



Bit Slice ALU Design

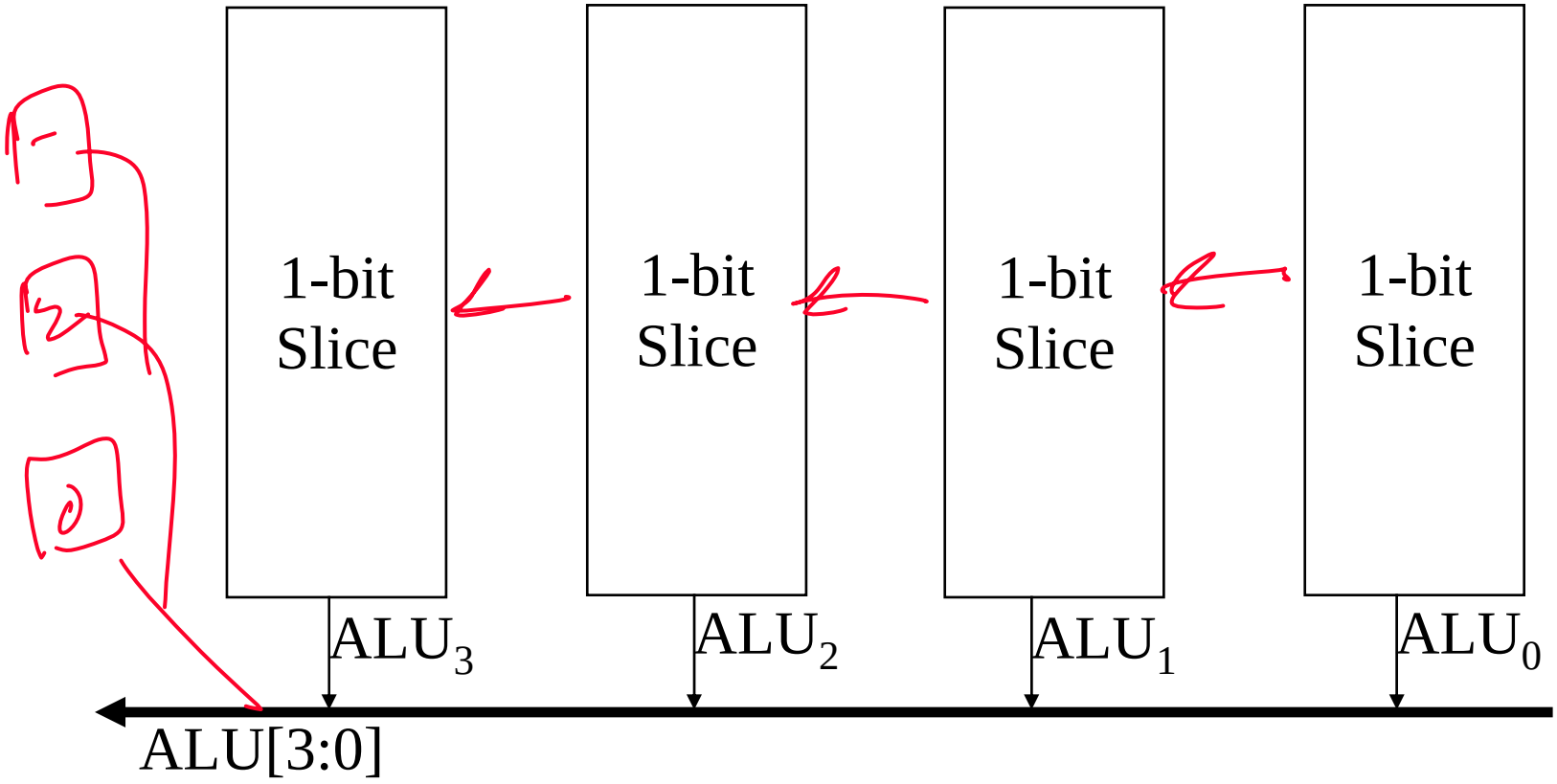
Add, Subtract, AND, OR



Bit Slice ALU Design (cont.)

Route Carries

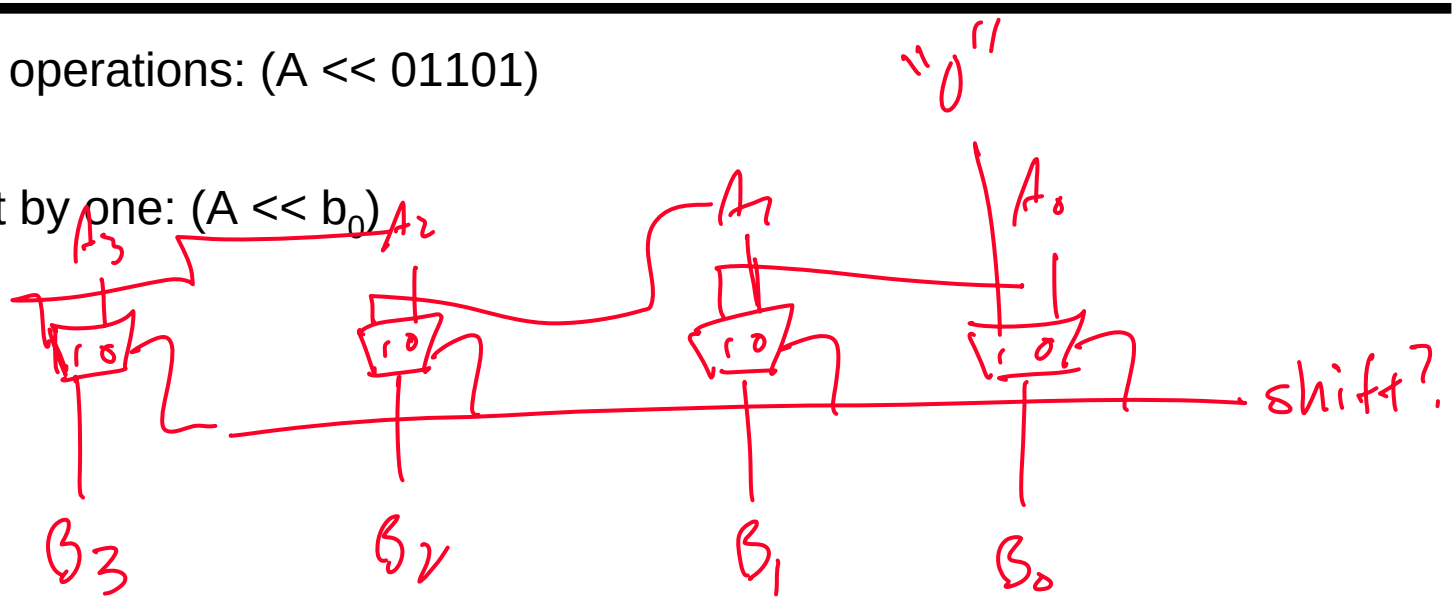
Overflow, zero, negative



Shifter

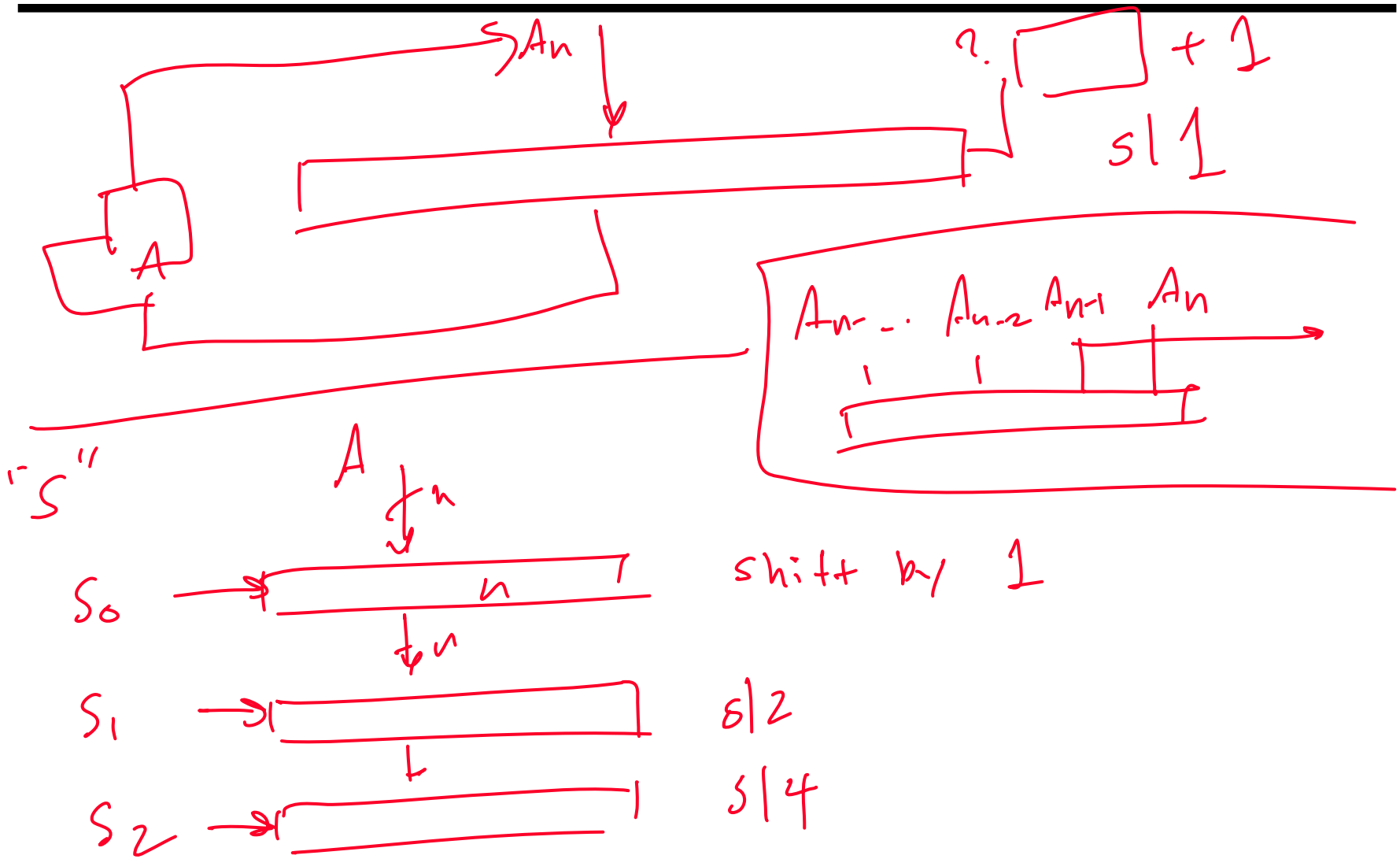
Support shift operations: (A << 01101)

Optional shift by one: (A << b_0)



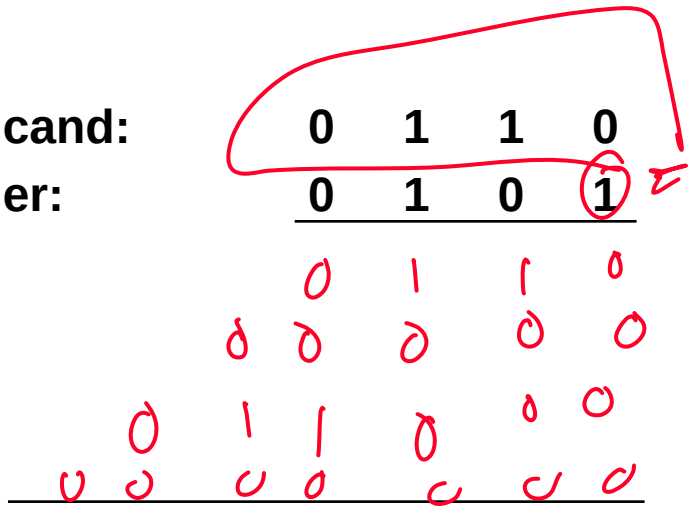
Optional shift by two: (A << b_1)

Shifter (cont.)



Multiplication

Example



Multiplicand: 0 1 1 0 — **6**

Multiplier: 0 1 0 1 — **5**

4 partial products

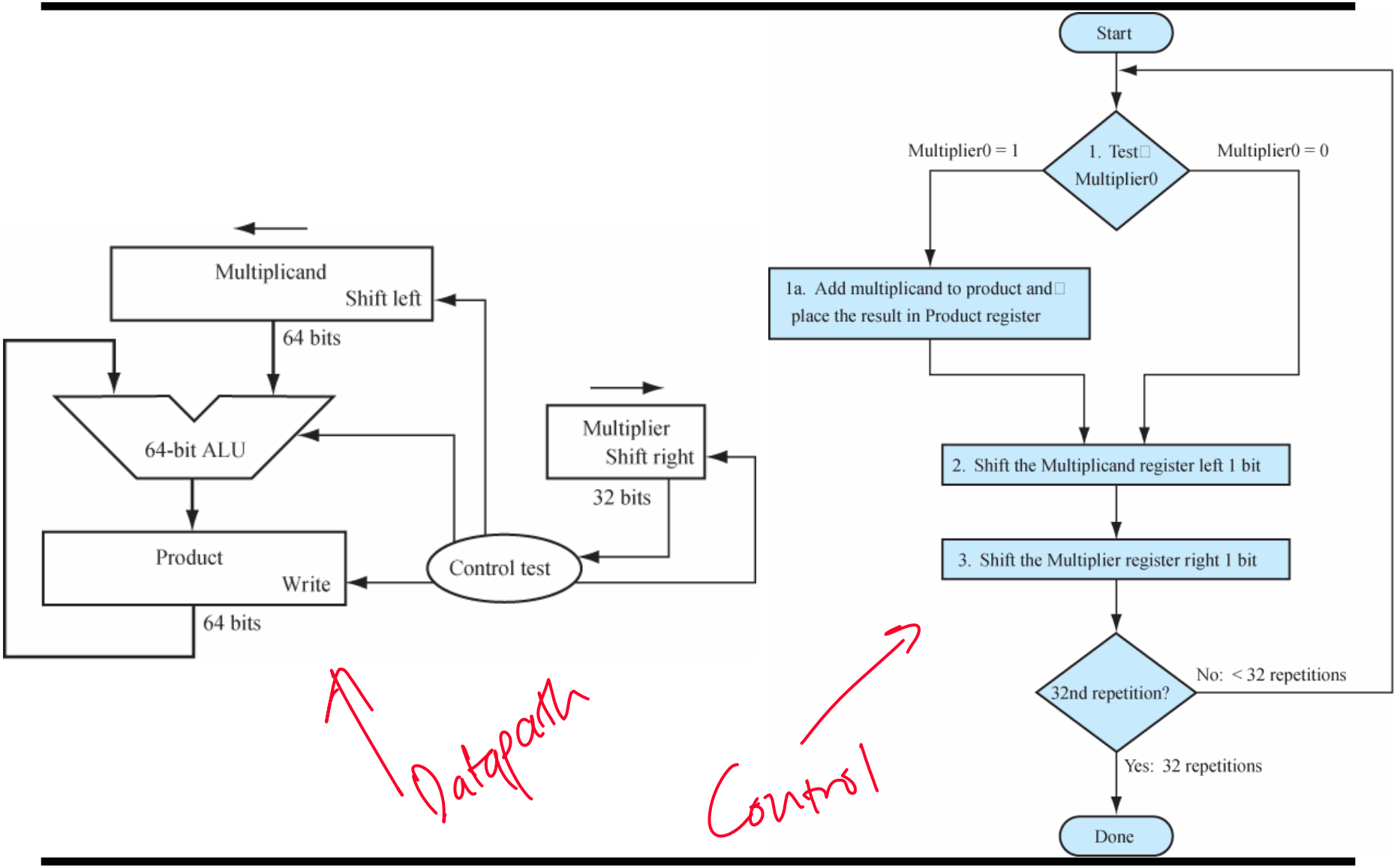
30

Repeat n times:

Compute partial product; shift; add

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

Sequential Multipliers

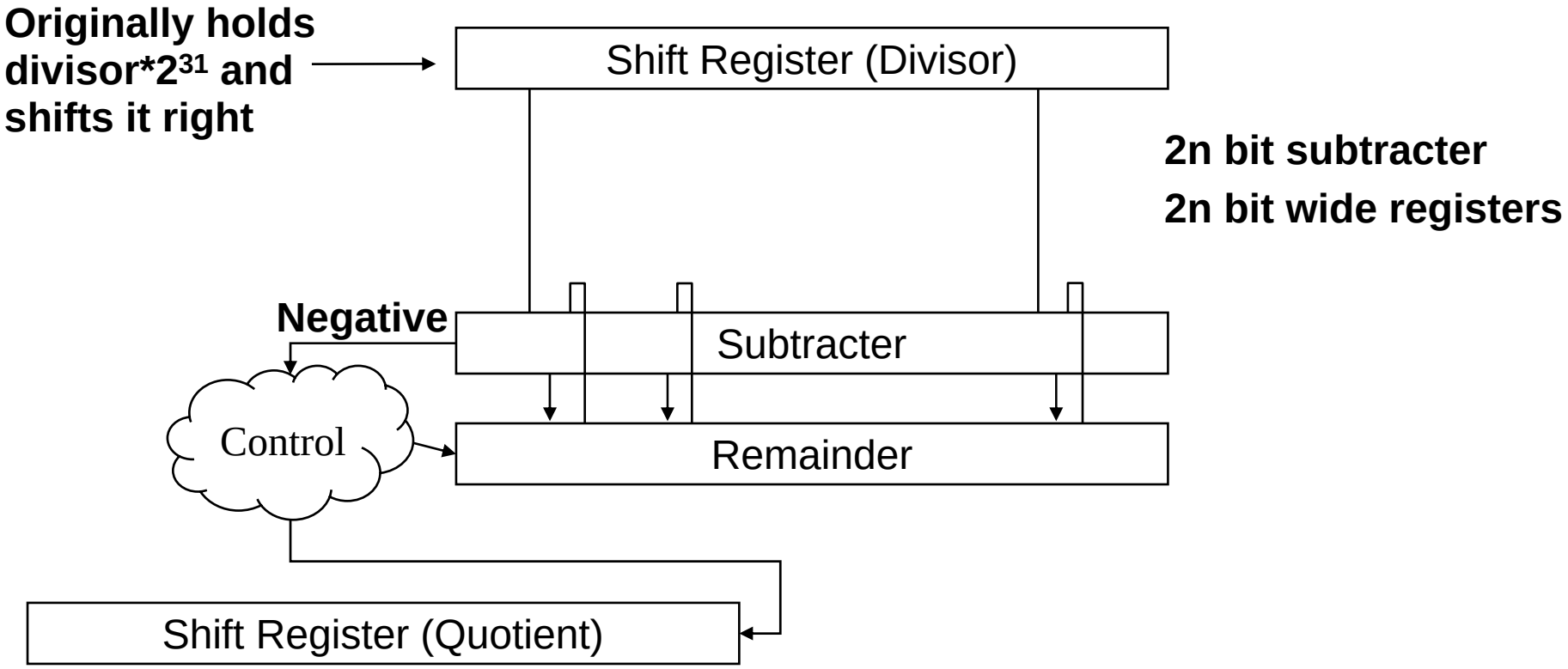


Division

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

Multiplication:

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

Sign: Different → Negative
Same → positive

(−)

Exponent: exp1 + exp2

(9)

Significand:

Sig1 x Sig2
if result >= base, div by base, exp++

10.2 → 1.02
e++

Floating Point Addition

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

$(5.38 \times 10^5 + 4.99 \times 10^5)$

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

$(-1.0 \times 10^5) + (9.99 \times 10^4)$

$(-1.0 \times 10^5) + (0.999 \times 10^5)$

Sign: (+)

(-)

Exponent: "stays the same" → from first / largest value.

(5)

(5)

Significand:

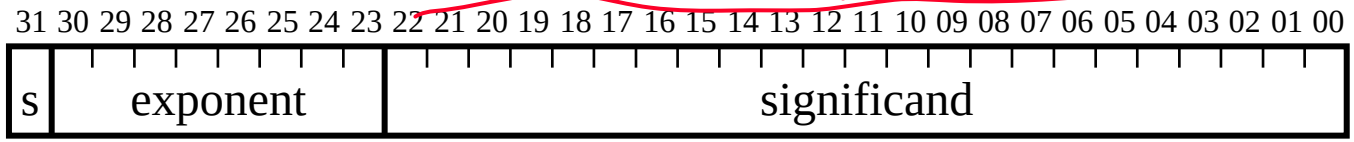
$10.37 \rightarrow 1.037 \times 10^6$

0.001

1.0×10^2

Floating Point Representation

Floating Point (Float) = $(-1)^s * (1.\text{significand})_2 * 2^{(\text{exponent}-127)}$



1.037

1.04

$(0.75)_{10} = \frac{1}{2} + \frac{1}{4}$

-0.75 in Floating Point?

sign: 1

sig: 0.11

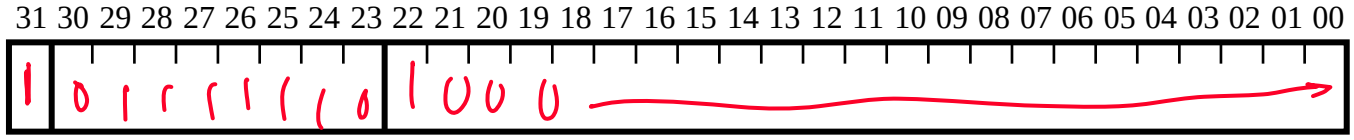
exp: -1

$(.5)_{10}$ $(.25)_{10} = 0.75$

sign: 1

exp: exp - 127 = -1 = 126

sig: 1.sig = "1."1

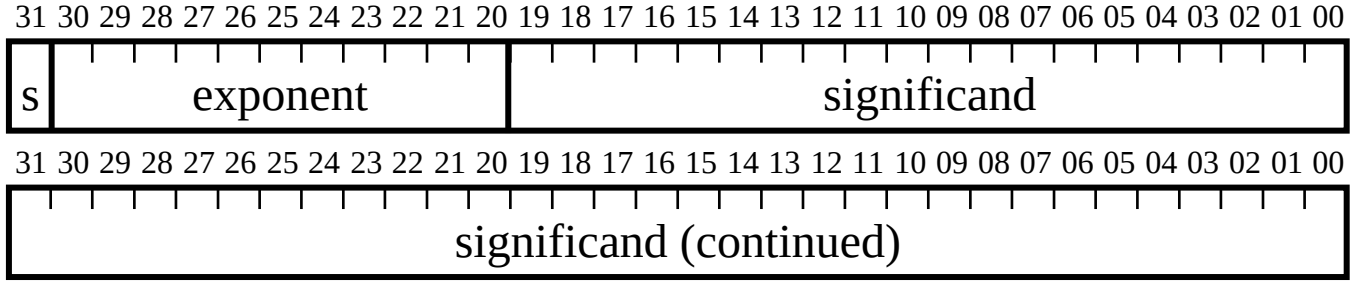


exponent: -1 = exp - 127

(1.)1001

Double Precision Representation

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



-0.75 in Double Precision?