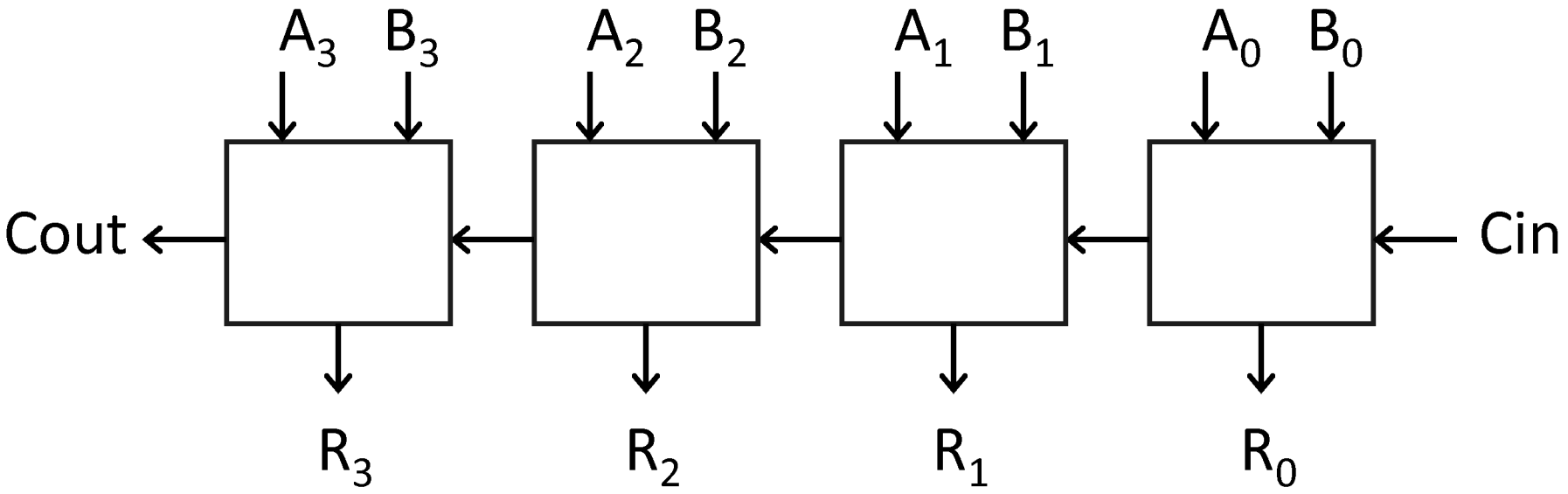
Arithmetic

Kevin Walsh
CS 3410, Spring 2010
Computer Science
Cornell University

See: P&H Chapter 3.1-3, C.5-6



- Adds two 4-bit numbers, along with carry-in
- Computes 4-bit result and carry out

Addition with negatives:

- pos + pos → add magnitudes, result positive
- neg + neg → add magnitudes, result negative
- pos + neg → subtract smaller magnitude,
  keep sign of bigger magnitude

First Attempt: Sign/Magnitude Representation

- 1 bit for sign (0=positive, 1=negative)
- N-1 bits for magnitude

Better: Two's Complement Representation

- Leading 1's for negative numbers
- To negate any number:
  - complement *all* the bits
  - then add 1

Non-negatives (as usual):	Negatives (two's complement: flip then add 1):	
+0 = 0000	~0 = 1111	-0 = 0000
+1 = 0001	~1 = 1110	-1 = 1111
+2 = 0010	~2 = 1101	-2 = 1110
+3 = 0011	~3 = 1100	-3 = 1101
+4 = 0100	~4 = 1011	-4 = 1100
+5 = 0101	~5 = 1010	-5 = 1011
+6 = 0110	~3 = 1001	-6 = 1010
+7 = 0111	~7 = 1000	-7 = 1001
+8 = 1000	~8 = 0111	-8 = 1000

Signed two's complement

- Negative numbers have leading 1's
- zero is unique: +0 = - 0
- wraps from largest positive to largest negative

N bits can be used to represent

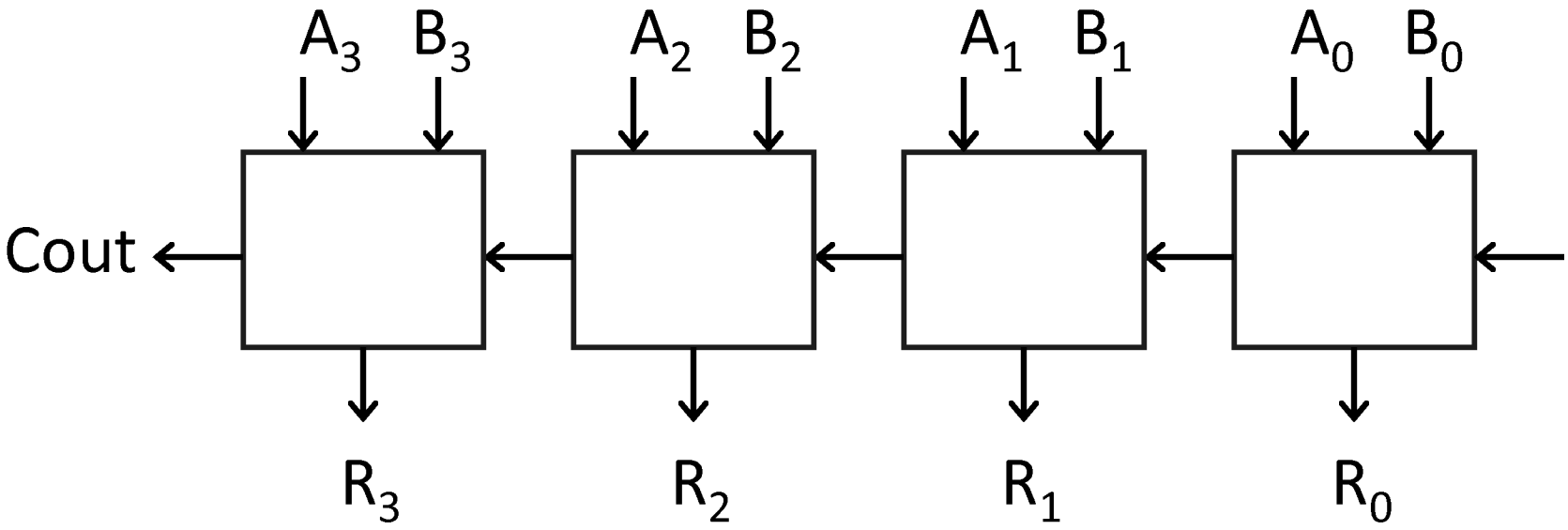
- unsigned:
  - eg: 8 bits $\Rightarrow$
- signed (two's complement):
  - ex: 8 bits $\Rightarrow$

Extending to larger size

Truncate to smaller size

Addition with two's complement signed numbers

- Perform addition as usual, regardless of sign (it just works)



How does that work?

$$\begin{array}{r} -154 \\ +283 \\ \hline \end{array}$$

Overflow

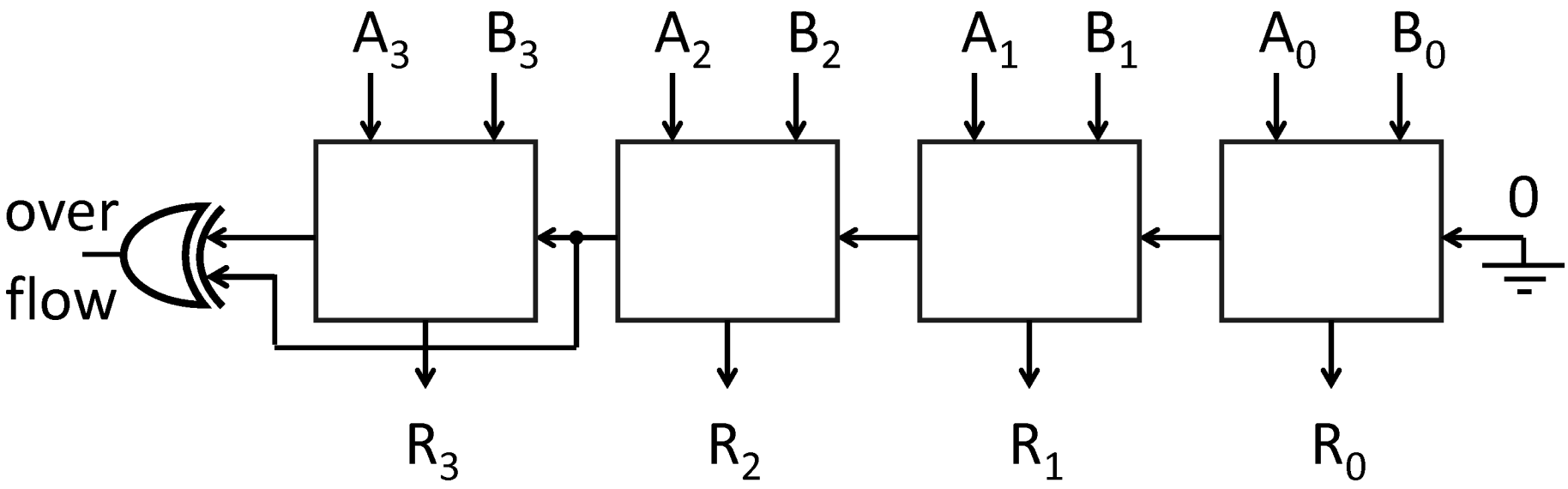
- adding a negative and a positive?
- adding two positives?
- adding two negatives?

Rule of thumb:

Overflow happened iff

carry into msb != carry out of msb

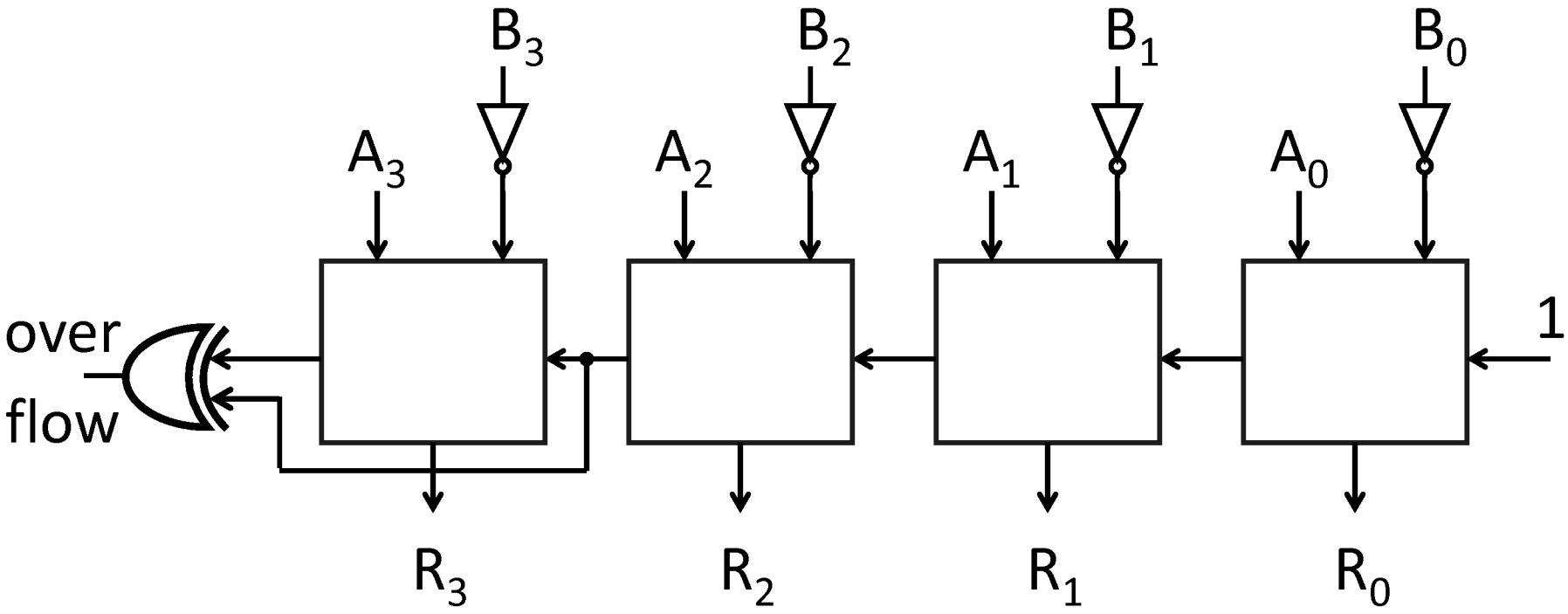
Two's Complement Adder with overflow detection



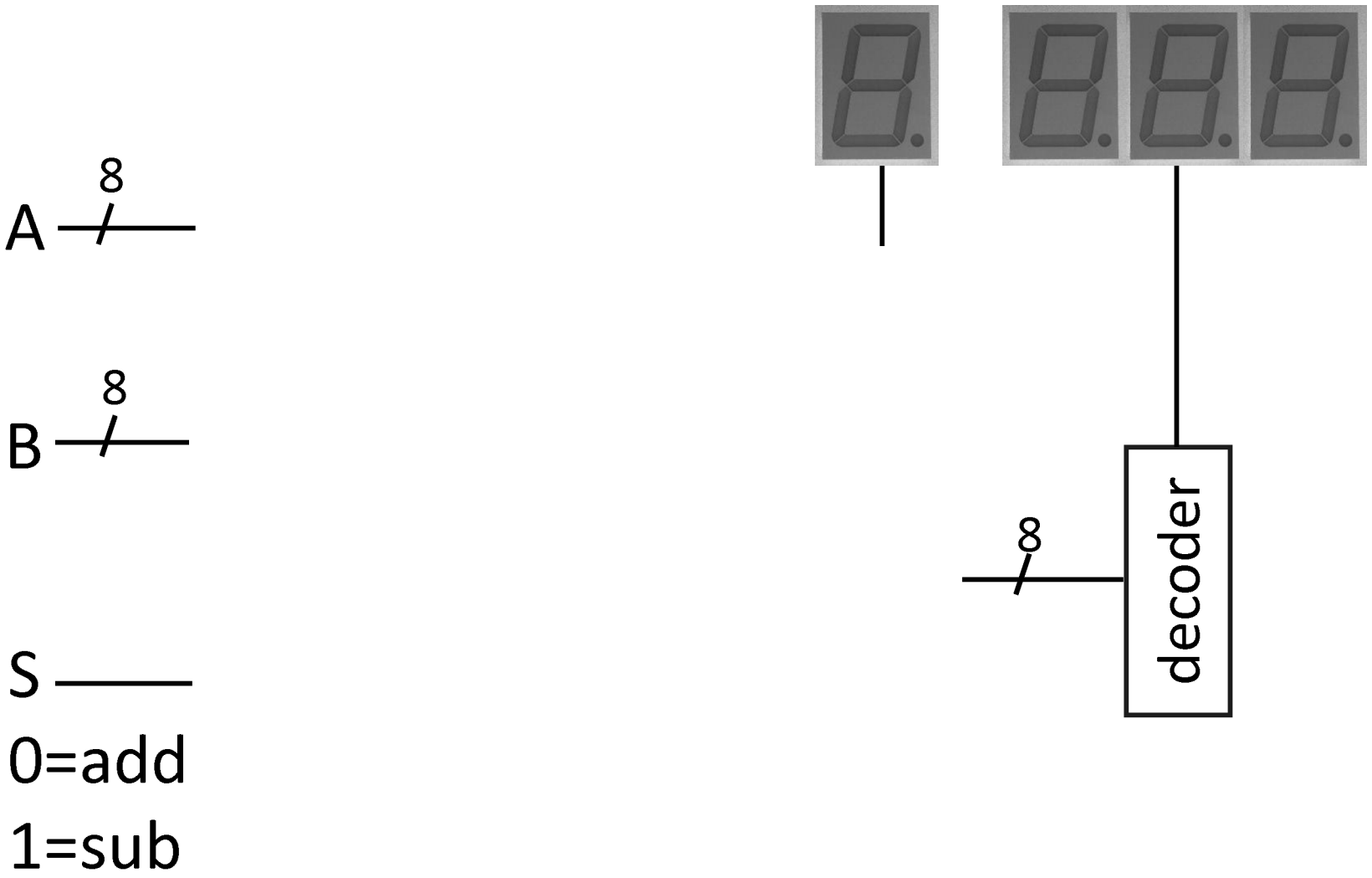
Two's Complement Subtraction

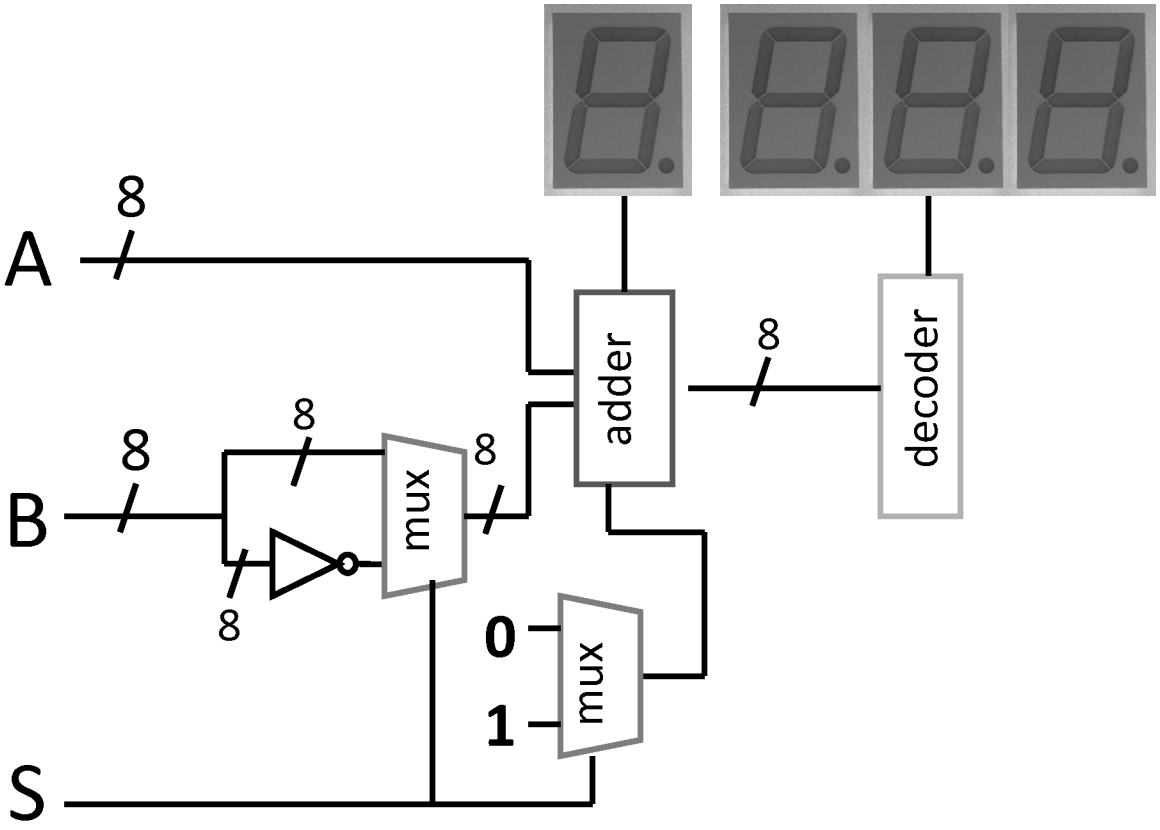
Lazy approach

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

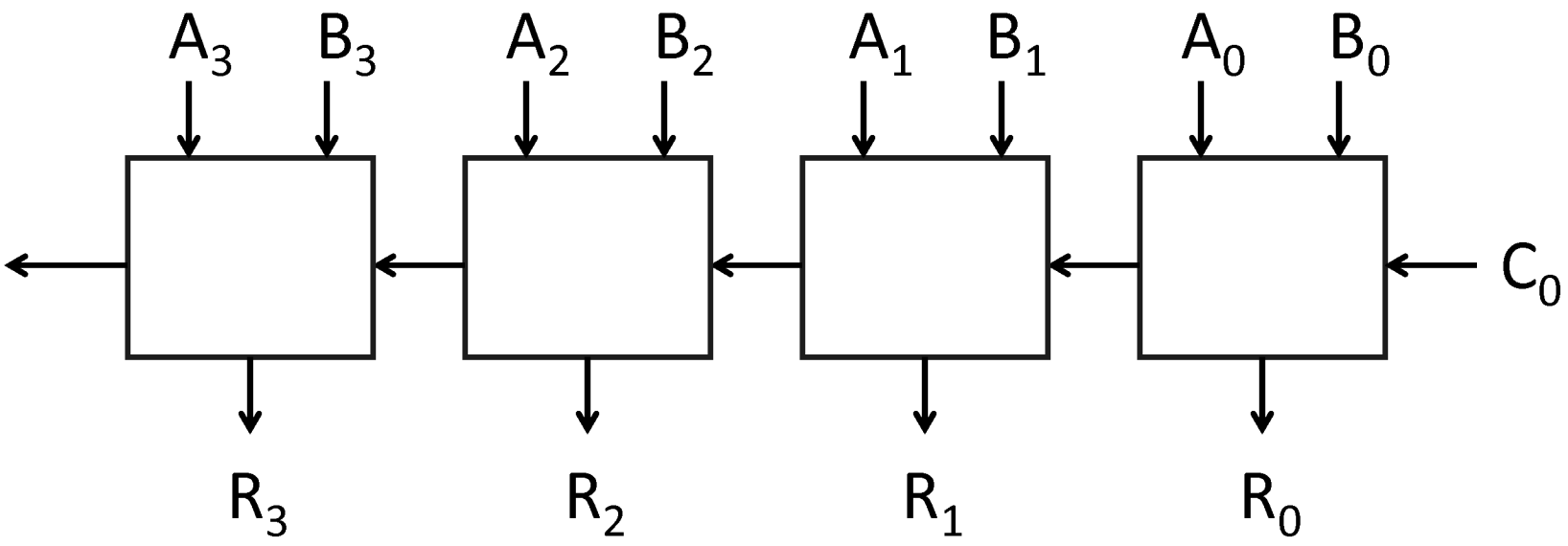


Q: What if (-B) overflows?

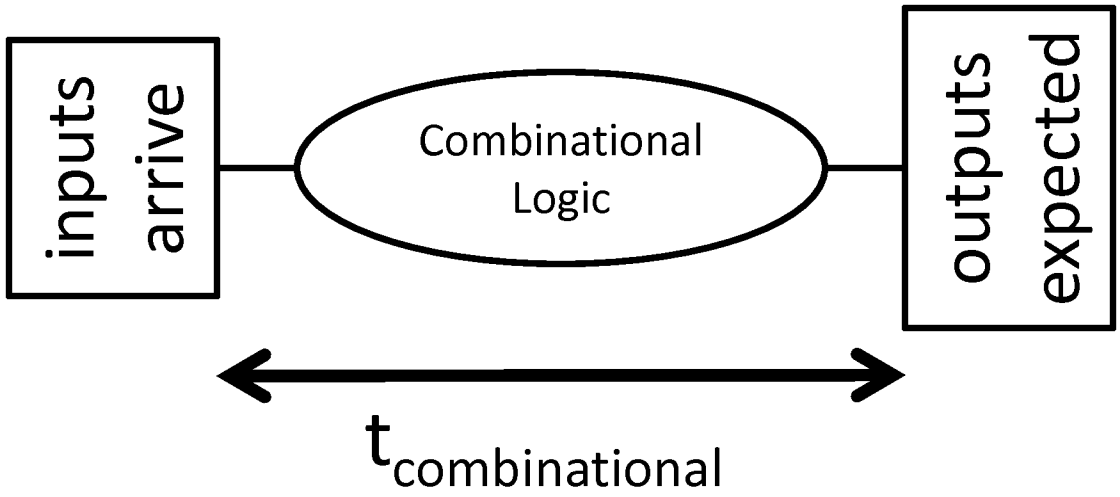

A
8
B
8
S
0=add
1=sub
8
decoder
8

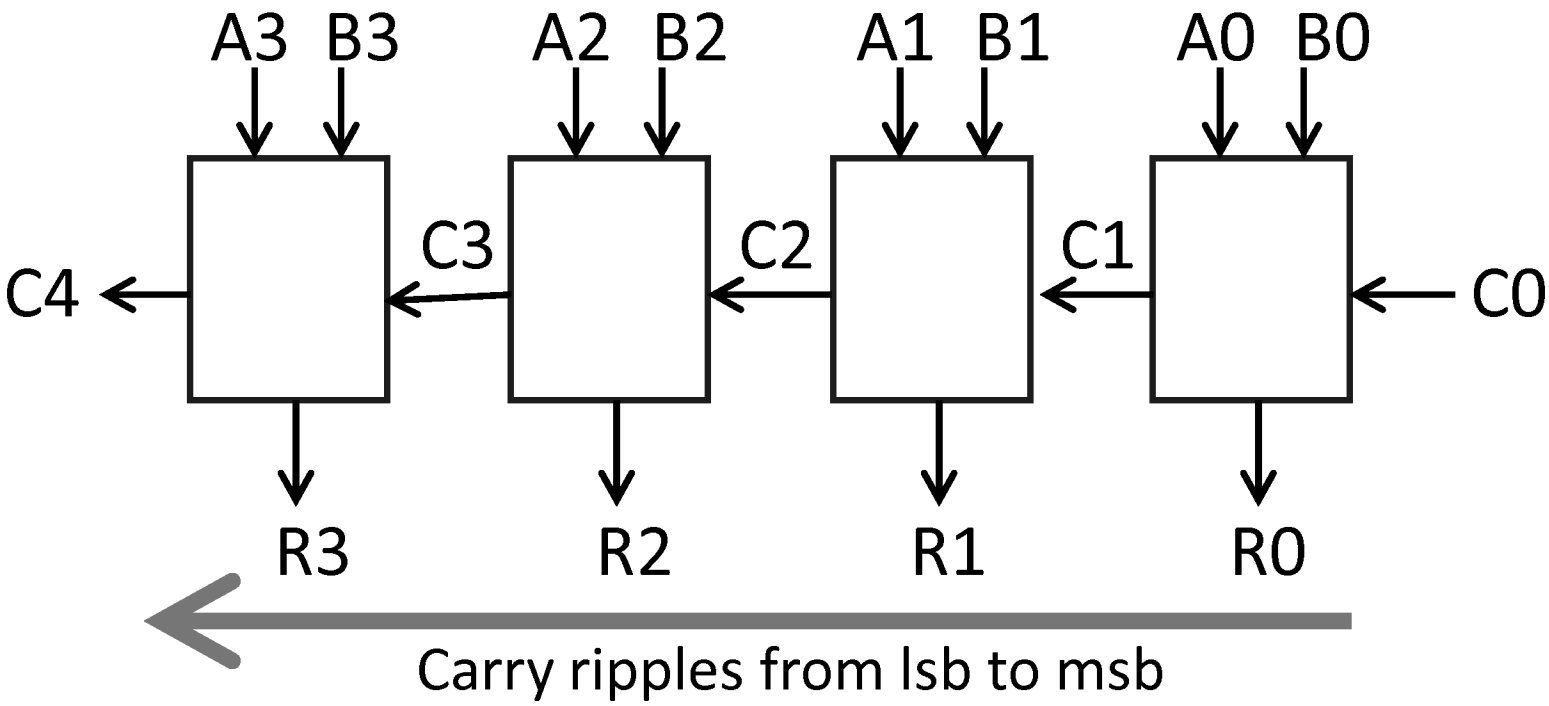

A
B
S
8
8
8
8
8
8
mux
mux
adder
decoder
0
1

- Is this design fast enough?
- Can we generalize to 32 bits? 64? more?



Speed of a circuit is affected by the number of gates in series (on the *critical path* or the *deepest level of logic*)





- First full adder, 2 gate delay
- Second full adder, 2 gate delay
- …