

Circuits

- and gate



- or gate

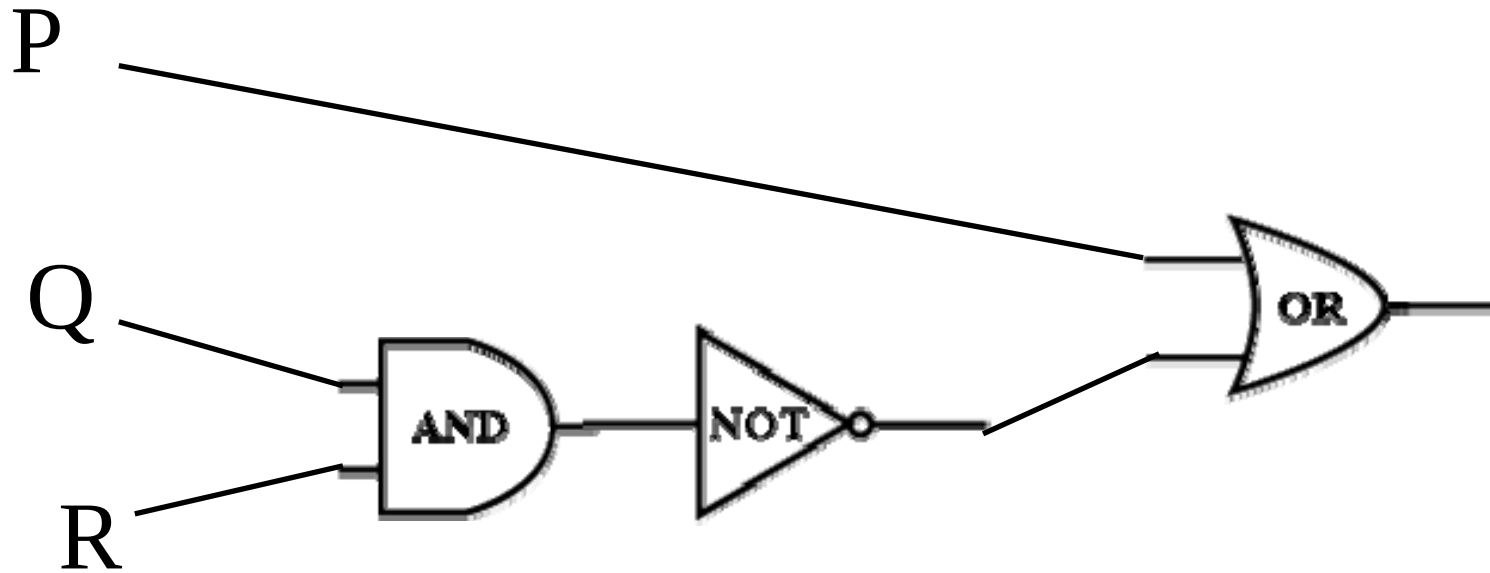


- not gate



Combining & Determining I/O Relationship

- $P \vee \sim (Q \wedge R)$



Draw the Circuit for:

P,Q&R

are

inputs

Simplify

before

building

the circuit

P	Q	R	Output
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

Number Conversions

- Base of the Number System
 - 10 (decimal), 2 (binary), 8 (octal), 16 (hexadecimal)
 - tells how many different numerals are used
 - determines the value of each place
- Conversions from anything to Base 10
 - use the definition of the number system
- Conversions from Base 10 to anything
 - use repeated integer division

Addition of Binary Numbers

- Carry if the number would be too large for the number system -- if it is greater than 1

1001

+ 10

1001

+ 11

1011

+ 11

1101

+111

Addition of Oct and Hex Numbers

- Carry if the number would be too large for the number system (larger than 7 or 15)

723_8

$+12_8$

265_8

$+ 33_8$

ABC_{16}

$+ 12_{16}$

CDE_{16}

$+ED_{16}$

Using a Circuit for Addition

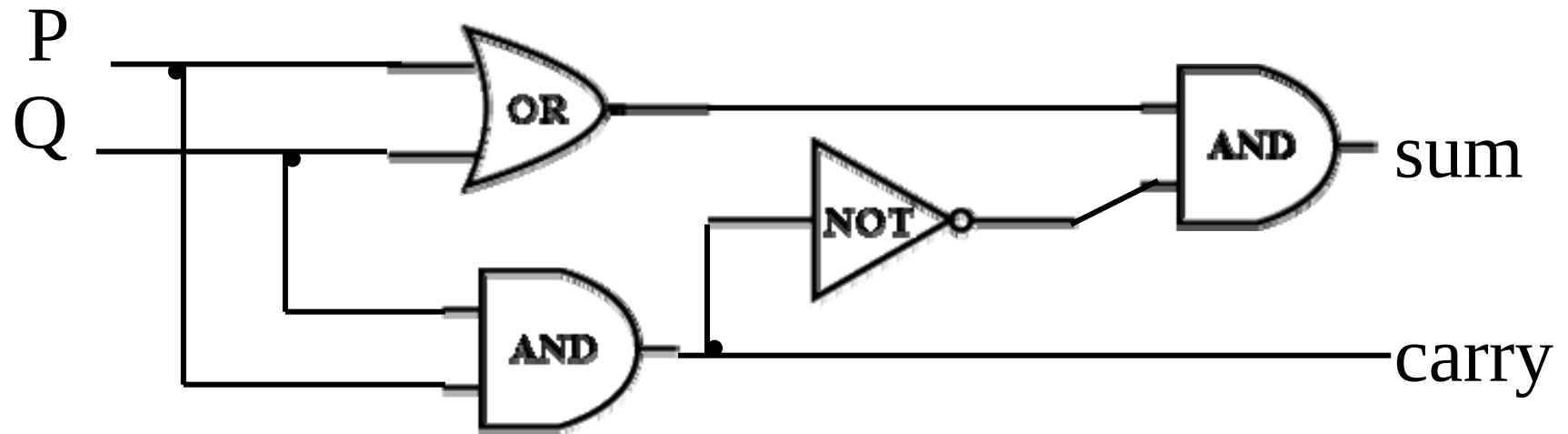
- Write as a Logic Expression
- Translate to Circuits

input

output

P	Q	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Half Adder

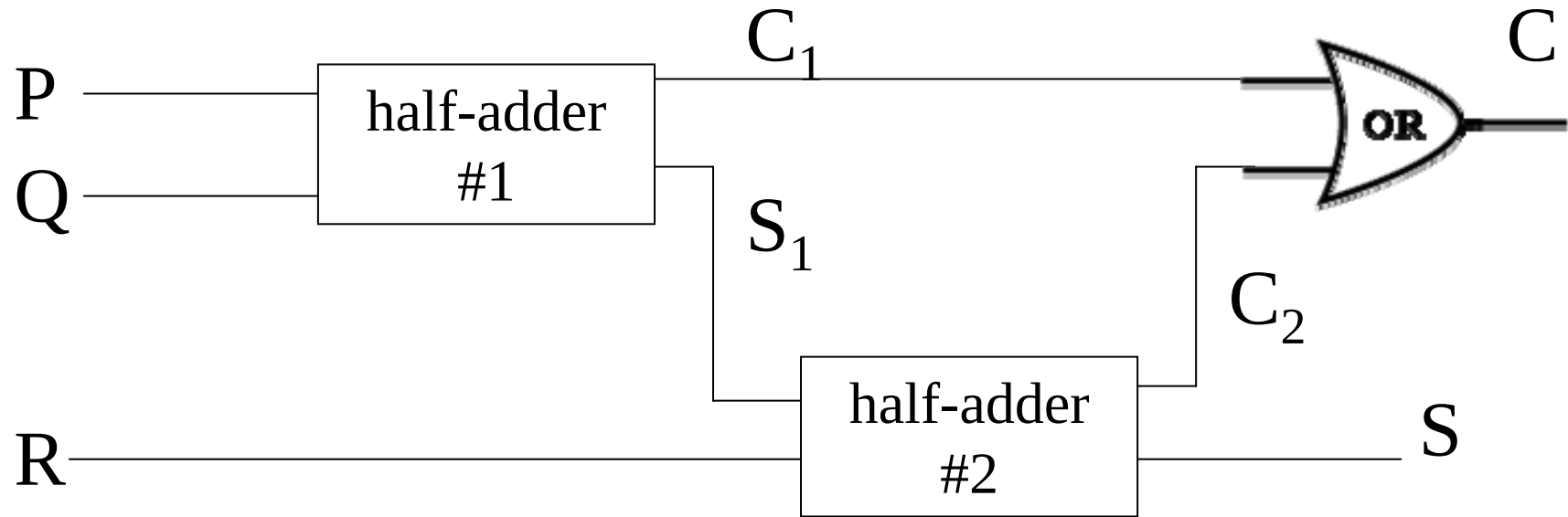


P and Q are binary values (1 bit each)

$$\text{sum} = (P \vee Q) \wedge \sim(P \wedge Q)$$

$$\text{carry} = (P \wedge Q)$$

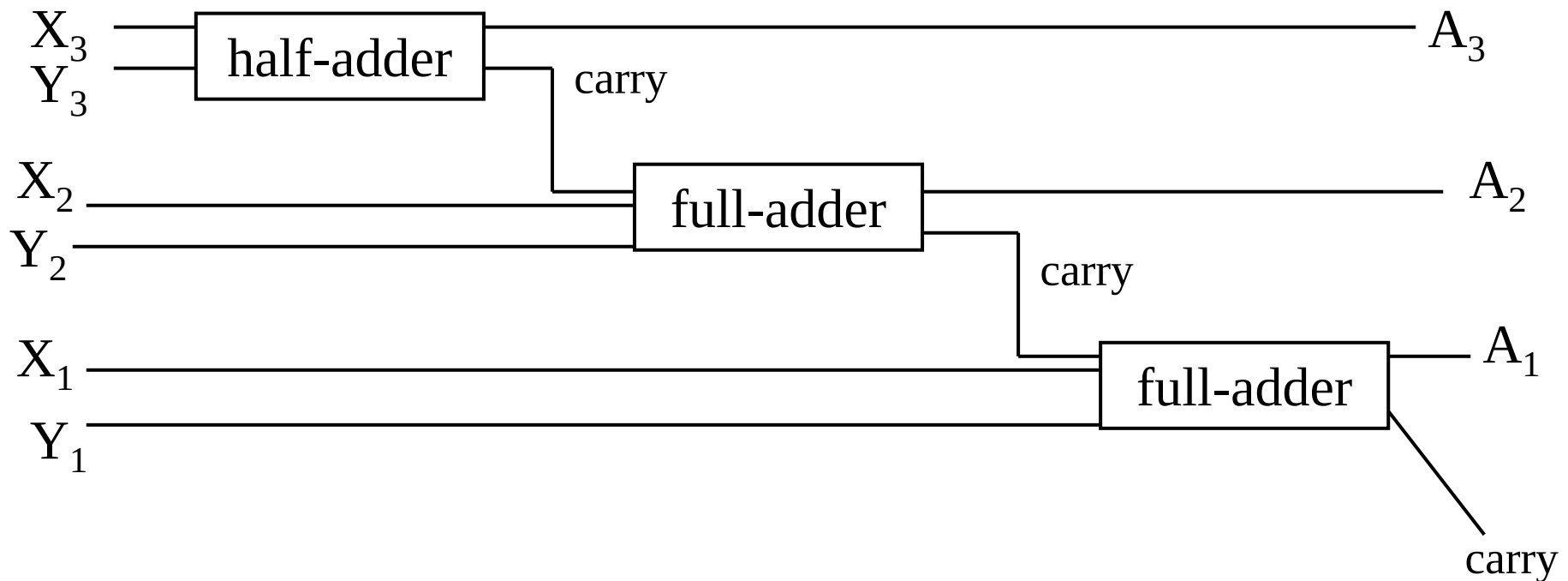
Full Adder



- P Q and R are binary digits
- $P + Q + R$ gives sum value and carry value

Parallel Adders

- Chain these half adders and full adders together for multi-bit addition
- $X_1X_2X_3 + Y_1Y_2Y_3 = CA_1A_2A_3$



2's Complement

To Represent Negative Values using Binary

1. Find the binary equivalent of the absolute value.
 2. Pad on the left to completely fill the bits.
 3. Switch all of the 1's to 0's and 0's to 1's.
 4. Add 1 to the result.
- i.e.: Find the 8-bit 2's complement representation of -43.

1. $43_{10} = 101011_2$
2. 00101011_2
3. 11010100_2
4. $11010101_2 = -43_{10}$