Phys 396 Sp 2026

Lecture 2

Friday: Read

Mon: HW 1 by 5pm

Binary addition

Any number, x, can be represented in binary form

$$x = x_{n-1} x_{n-2} \ldots x_1 x_0$$

where each bit x_j can take values 0 or 1. This is set up so that

$$x = 2^{n-1} x_{n-1} + 2^{n-2} x_{n-2} + \ldots + 2^1 x_1 + 2^0 x_0$$

One can add two numbers

$$x = x_{n-1} x_{n-2} \ldots x_1 x_0$$

$$y = y_{n-1} y_{n-2} \ldots y_1 y_0$$

purely using operations on the bits that represent the numbers. So if

$$z = x + y$$

then

$$z = z_n z_{n-1} \ldots z_1 z_0$$

and the bits for z must be obtained from the bits for x and y. So we have

Inputs

$x_0, x_1, x_2, \ldots x_{n-1}$

$y_0, y_1, y_2, \ldots, y_{n-1}$

$\rightarrow$

Outputs

z_0 involves x_0, y_0

z_1 " x_1, y_1 and x_0, y_0

$\vdots$

How can we construct an algorithm for adding numbers in a bitwise fashion? We need two basic operations.

Multiplication of two bits

Let x, y be two single bits. Then the product of these is a map

$$x, y \to xy$$

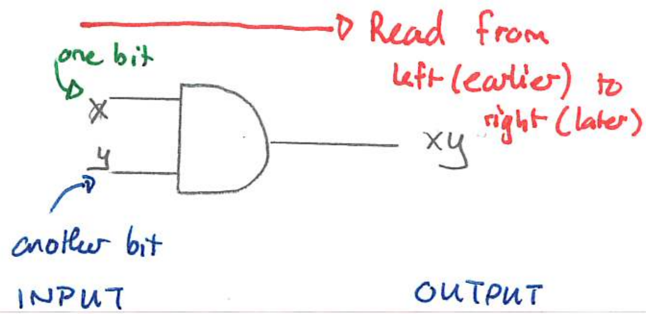
and we can construct a table of all possibilities. In logic, this combination is called the "AND" operation. We say

$$x \text{ AND } y \quad \text{or} \quad x \wedge y$$

"truth table"

x	y	$x \wedge y \equiv xy$
0	0	0
0	1	0
1	0	0
1	1	1

We can represent this as a circuit using an "AND" gate.



In electronics, we can construct a piece of hardware that accomplishes this operation and we can then string multiple such operations together. We could use this to multiply any number of single bits.

Modular addition of two bits

Consider adding

$$x_k 2^k + y_k 2^k = s_k 2^k + c_k 2^{k+1}$$

what coefficients appear for $s_k, c_k, \ldots$??

1 Addition of two single bits

Consider adding $x2$ and $y2$ where $x, y \in \{0, 1\}$ are bits. The sum should be of the form

$$x2 + y2 = s2 + c2^2$$

where $s \in \{0, 1\}$ is the sum bit and $c \in \{0, 1\}$ the carry bit. The aim of this exercise is to determine expressions for s and c in terms of x and y.

a) Create a table of all possible values for x and y. For each row determine s and c.

b) Use the table to find an expression for the carry c in terms of x and y.

c) What gate could be used to determine the carry?

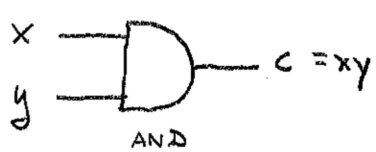
d) Use the table to determine an expression for the sum s in terms of x and y.

Answer a)

x	y	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

b) One can see that $c = xy$ (binary multiplication)

c) The AND gate accomplishes this:



d) This is less obvious and will require some type of addition where $1 + 1 = 0$?

To accommodate this, we require a new type of addition. This is called "addition modulo 2" or modular addition. The idea is

* add two digits
* divide by 2 and determine the remainder
* the remainder is the result of the addition process.

More precisely

> Let x, y be two bits. Then addition modulo 2 gives
>
> $$x \oplus y$$
>
> where
> $$0 \oplus 0 = 0$$
> $$0 \oplus 1 = 1$$
> $$1 \oplus 0 = 1$$
> $$1 \oplus 1 = 0$$

Thus in the problem of

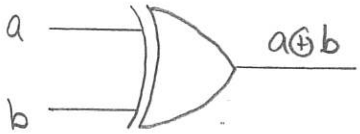
$$x\,2 + y\,2 = s\,2 + c\,2^2$$

We have $\qquad s = x \oplus y.$

So in general

> If x_k, y_k are bits then
>
> $$x_k 2^k + y_k 2^k = s_k 2^k + c_{k+1} 2^{k+1}$$
>
> where the sum is
>
> $$s_k = x_k \oplus y_k$$
>
> and the carry is
>
> $$c_{k+1} = x_k y_k.$$

Addition modulo 2 can be represented by the XOR gate.
The XOR gate maps



a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

In logic this is called exclusive or.

Thus the basic steps in addition using bits can be accomplished using AND and XOR gates. Note however that in practical addition of two numbers

$$\left.\begin{aligned} x &= x_{n-1}x_{n-2}\ldots x_2x_1x_0 \\ y &= y_{n-1}y_{n-2}\ldots y_2y_1y_0 \end{aligned}\right\} \to \quad x+y \equiv z = z_n z_{n-1}\ldots z_2z_1z_0$$

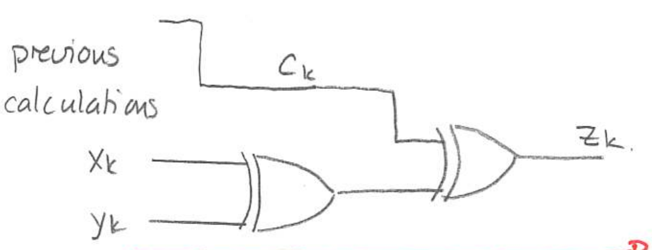
we have that

$$z_k$$

is determined from x_k, y_k and the carry c_k from the lower bits. Clearly

$$z_k = x_k \oplus y_k \oplus c_k$$

and modular arithmetic guarantees that the sum can be done in any order. This will require multiple XOR gates (a circuit)



read left to right

There would need to be a separate circuit to determine the next carry c_{k+1}.

We can see that

$$\boxed{\text{Binary addition can be accomplished by a succession of XOR and AND gates}}$$

In physical realizations each gate invocation requires:

1) hardware
2) time

Thus we can quantify the difficulty of the problem of adding binary numbers as follows:

$$\boxed{\text{The difficulty of adding two numbers is quantified by counting the number of basic XOR and AND gates required to do the procedure.}}$$

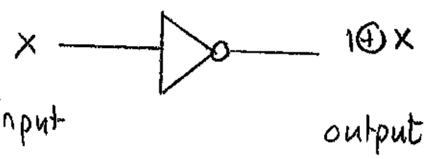
We can then ask how to optimize the process or algorithm for performing the addition task.

Other gates

There are various other gates that perform useful binary operations

1) NOT gate

This maps as follows

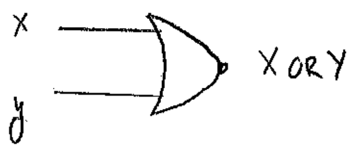


Input x	$1\oplus x$ output
0	1
1	0

This flips the bit value and is called negation

2) OR gate

This performs the logical "OR" operation that takes two inputs and produces an output of "1" provided at least one input is "1".



x	y	x or y
0	0	0
0	1	1
1	0	1
1	1	1

We can show that the OR gate maps

$$(x,y) \to 1\oplus\left[(1\oplus a)(1\oplus b)\right]$$

It turns out that the AND, OR and NOT gate can be used to construct and XOR gate. In fact

> The AND, OR and NOT gates are sufficient to construct any binary operation.

In this sense the set of such gates is <u>universal</u>