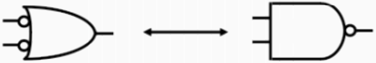
Digital logic

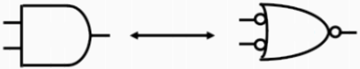
CS 211: Computer Architecture

Applying DeMorgan's Laws to gates

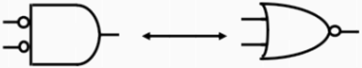
$$\overline{X} + \overline{Y} = \overline{XY}$$



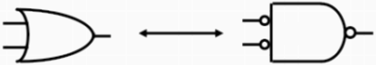
$$XY = \overline{\overline{X} + \overline{Y}}$$



$$\overline{X}\,\overline{Y} = \overline{X + Y}$$



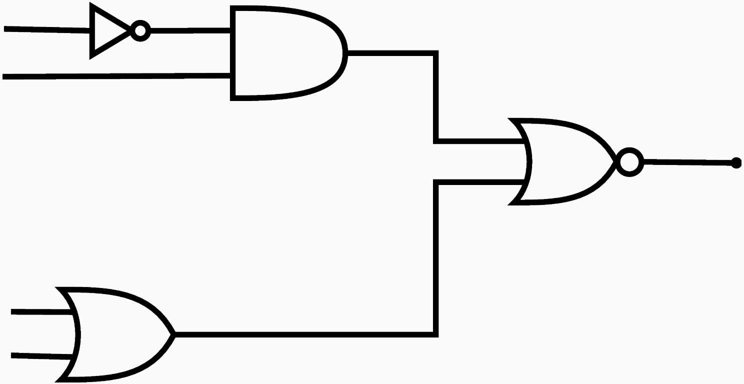
$$X + Y = \overline{\overline{X}\,\overline{Y}}$$



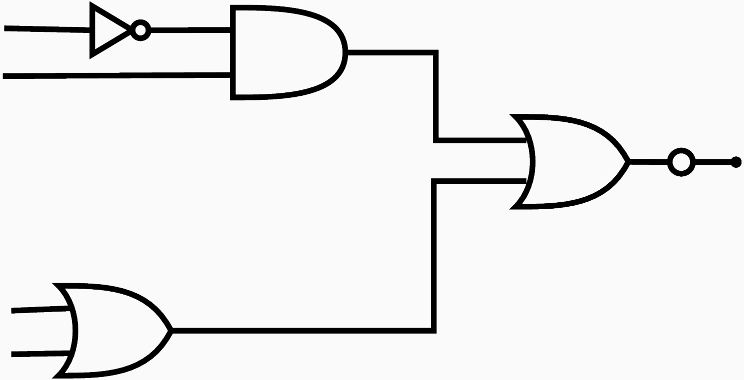
Converting circuits to NAND gates

To convert a circuit to all NAND gates, use the previous gate conversions, plus the NOT conversion below. Start from the output and work backwards towards the inputs.

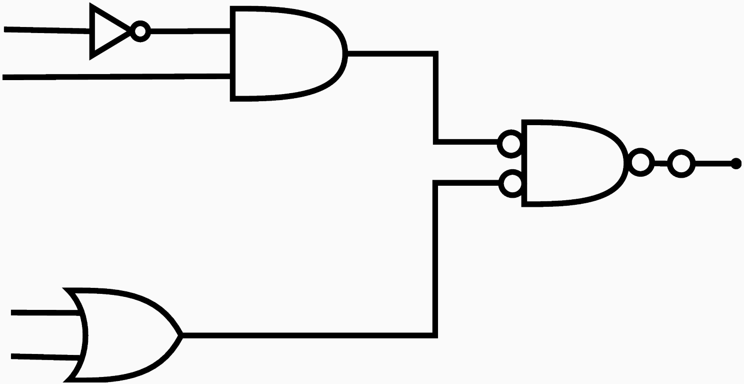
Converting circuits to NAND gates



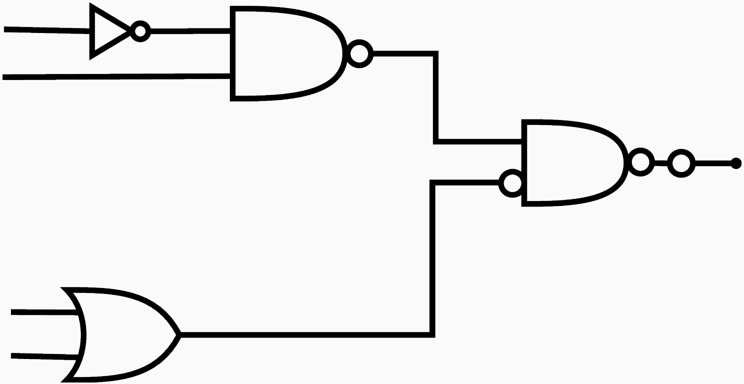
Converting circuits to NAND gates



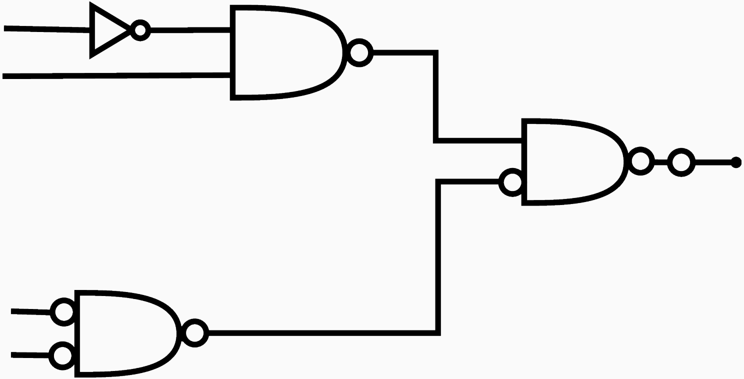
Converting circuits to NAND gates



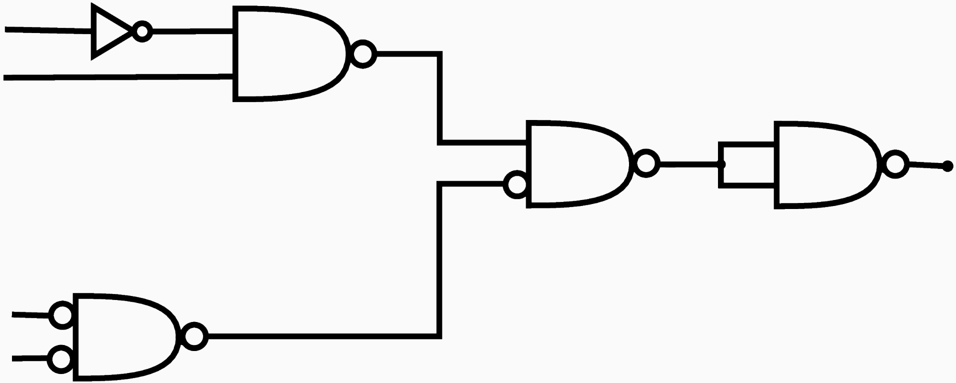
Converting circuits to NAND gates



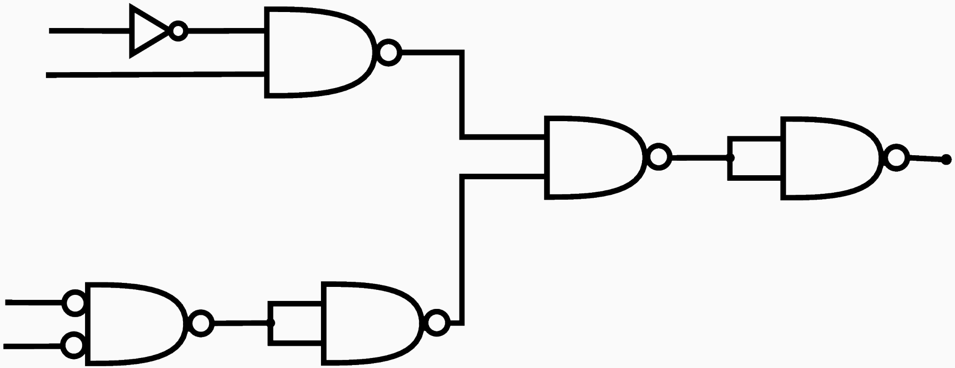
Converting circuits to NAND gates



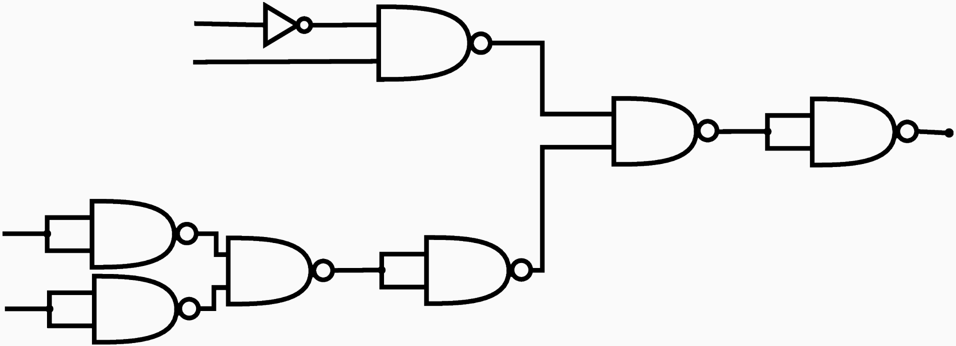
Converting circuits to NAND gates



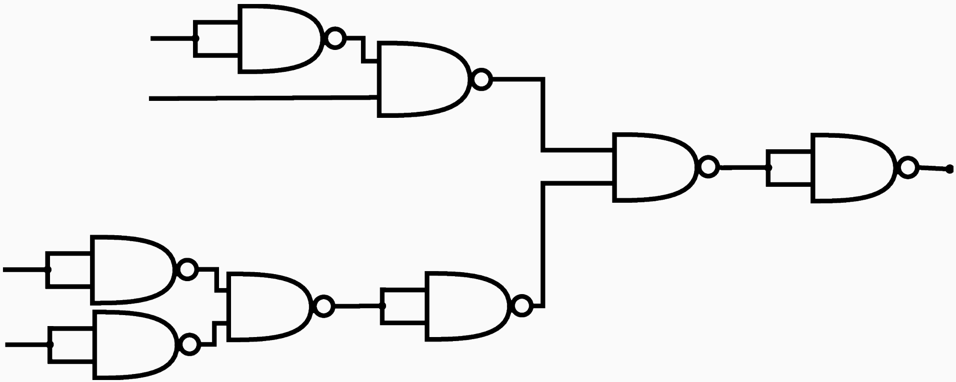
Converting circuits to NAND gates



Converting circuits to NAND gates

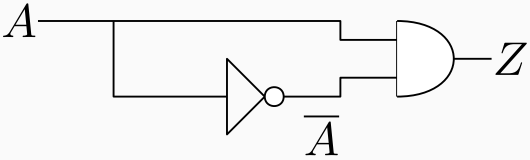


Converting circuits to NAND gates



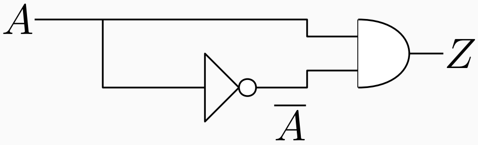
Timing

In this circuit, what value should Z have?

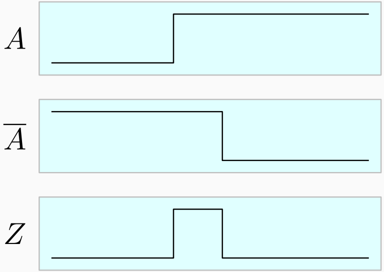


Timing

Circuit:



Voltages over time:

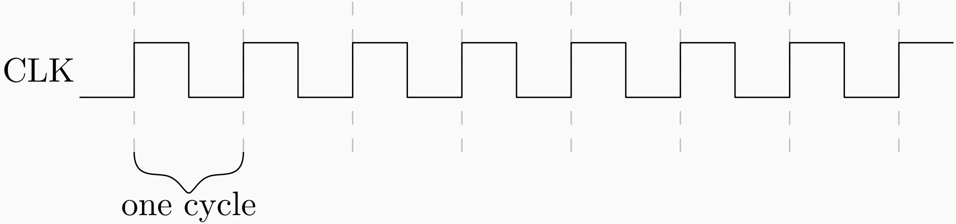


Timing

One way to deal with this is with synchronous circuits, that use a clock signal to know when to update.

1 Hz = 1 cycle / second

1 GHz = 1 billion cycles / second



Adding two bits

We can add two single bits:

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

Adding two bits

Let's also output the carry bit:

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Adding two 4-bit numbers

Let's add `1010` and `0011`:

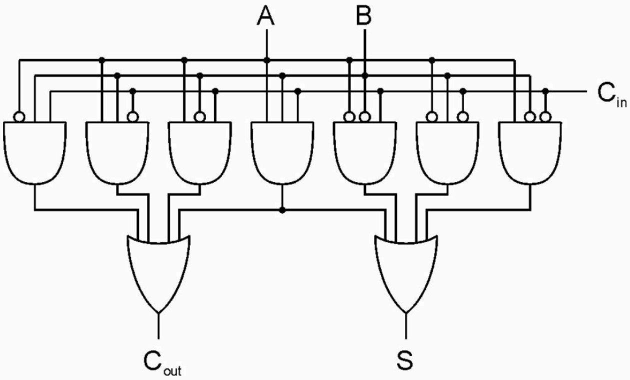
$$\begin{array}{ccccc} & & 1 & & \\ & 1 & 0 & 1 & 0 \\ + & 0 & 0 & 1 & 1 \\ \hline & 1 & 1 & 0 & 1 \end{array}$$

Adding two bits

Let's also add an *input* carry bit:

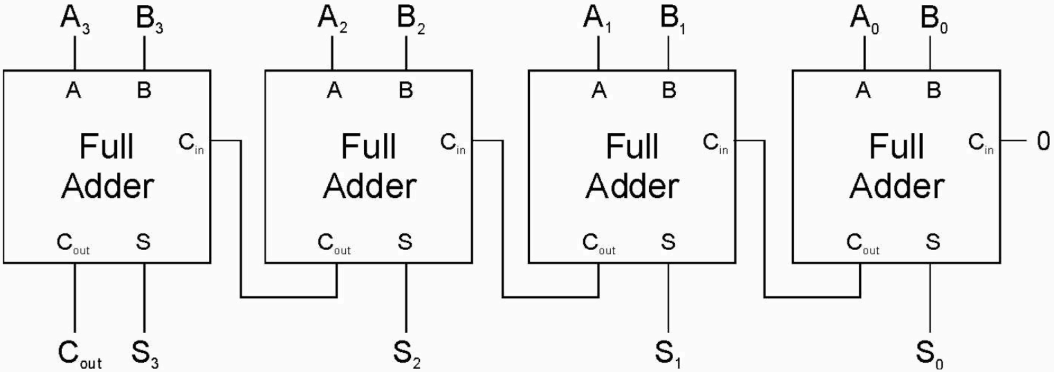
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Adding two bits

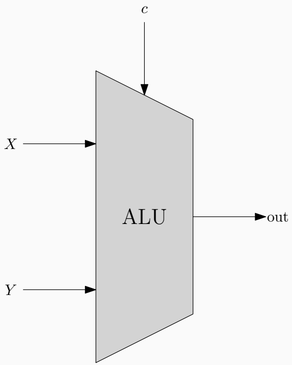


A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Adding two 4-bit numbers



Arithmetic logic unit (ALU)

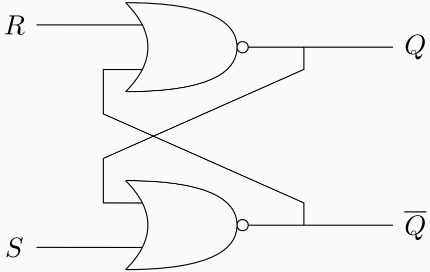


c	out
0	$X + Y$
1	$X - Y$
2	$X * Y$
3	X/Y

Sequential circuits

So far all circuits have been combinational, functions from input to output.

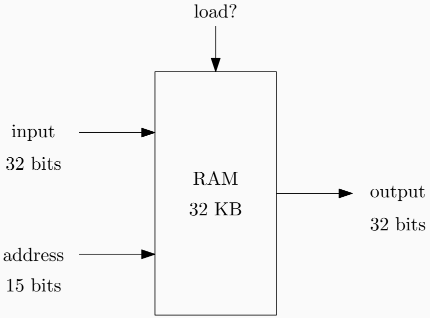
Sequential circuits also have a way to preserve state.



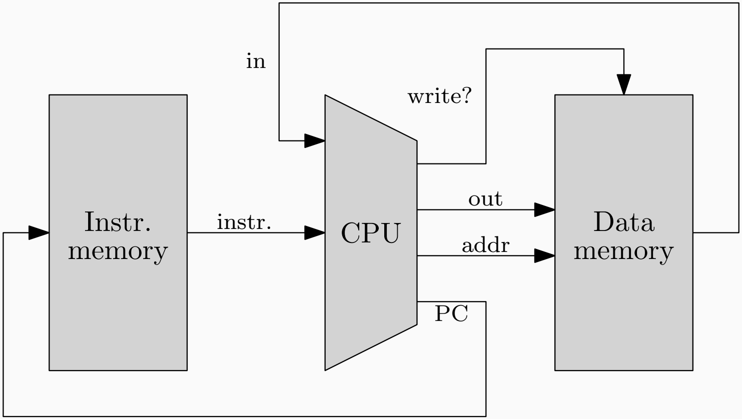
Memory

Once we have a circuit that can store one bit:

- combine 32 of them to make a 32-bit register
- combine registers to make a RAM module

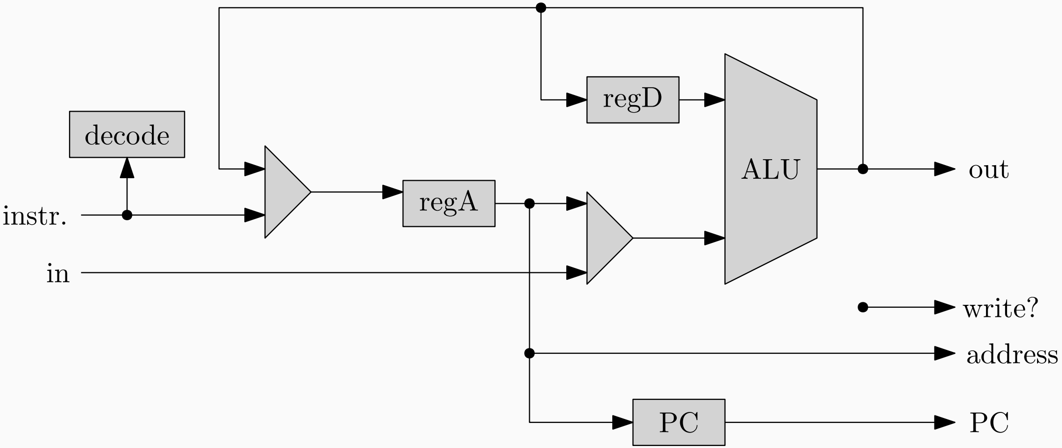


Computer overview



(Elements of Computer Systems)

Central processing unit (CPU)



(Elements of Computer Systems)