Binary subtraction

Intro

Binary subtraction is interesting in that it uses negative numbers to arrive at a result. For example, if you start with 7 and subtract 5, it's the same thing as adding 7 to -5.

$$7 - 5 = 2$$

is the same thing as:

$$7 + (-5) = 2$$

It's just a different way of skinning a cat, and a concept that wasn't available until the zero was fully understood. In fact, it wasn't until 1657 that a mathematician (John Hudde) used a single variable to represent either a positive or a negative number. For all those years until 1657, positive and negative numbers were handled as separate special cases. The reason is because we couldn't conceive of there being less than nothing.

In electronics, a subtractor can be designed using the same approach as that of an adder.

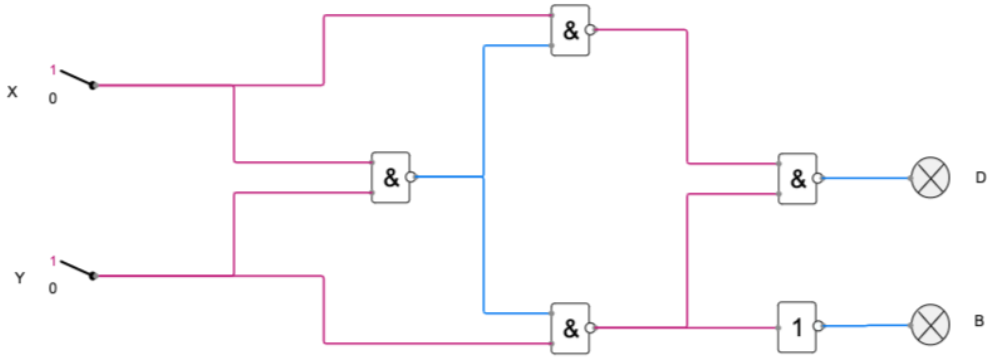
As with an adder, in the general case of calculations on multi-bit numbers, three bits are involved in performing the subtraction for each bit of the difference: the minuend X_i, subtrahend Y_i, and a borrow in from the previous (less significant) bit order position B_i.

The outputs are the difference bit D_i and borrow bit B_{i+1}.

The subtractor is best understood by considering that the subtrahend and both borrow bits have negative weights, whereas the X and D bits are positive.

Half Substractor

The half subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs, the minuend X and subtrahend Y and two outputs the difference D and borrow out B_{out}. The borrow out signal is set when the subtractor needs to borrow from the next digit in a multi-digit subtraction.

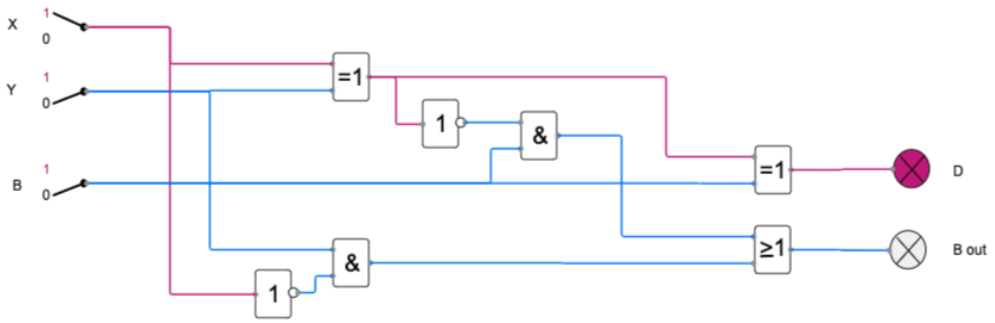


The truth table of the half substractor is:

X	Y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Full subtractor

The full subtractor is a combinational circuit which is used to perform subtraction of three input bits: the minuend X, subtrahend Y, and borrow in B^{in}. The full subtractor generates two output bits: the difference D and borrow out B^{out}. B^{in} is set when the previous digit is borrowed from X. Thus, B^{in} is also subtracted from X as well as the subtrahend Y.



The truth table of the full substractor is:

X	Y	B in	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0

1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Binary subtraction

Intro

Binary subtraction is interesting in that it uses negative numbers to arrive at a result. For example, if you start with 7 and subtract 5, it's the same thing as adding 7 to -5.

$$7 - 5 = 2$$

is the same thing as:

$$7 + (-5) = 2$$

It's just a different way of skinning a cat, and a concept that wasn't available until the zero was fully understood. In fact, it wasn't until 1657 that a mathematician (John Hudde) used a single variable to represent either a positive or a negative number. For all those years until 1657, positive and negative numbers were handled as separate special cases. The reason is because we couldn't conceive of there being less than nothing.

In electronics, a subtractor can be designed using the same approach as that of an adder.

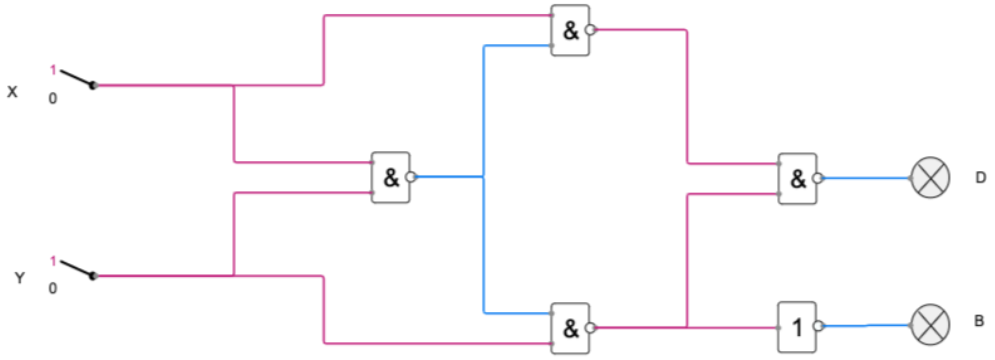
As with an adder, in the general case of calculations on multi-bit numbers, three bits are involved in performing the subtraction for each bit of the difference: the minuend X_i, subtrahend Y_i, and a borrow in from the previous (less significant) bit order position B_i.

The outputs are the difference bit D_i and borrow bit B_{i+1}.

The subtractor is best understood by considering that the subtrahend and both borrow bits have negative weights, whereas the X and D bits are positive.

Half Substractor

The half subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs, the minuend X and subtrahend Y and two outputs the difference D and borrow out B_{out}. The borrow out signal is set when the subtractor needs to borrow from the next digit in a multi-digit subtraction.

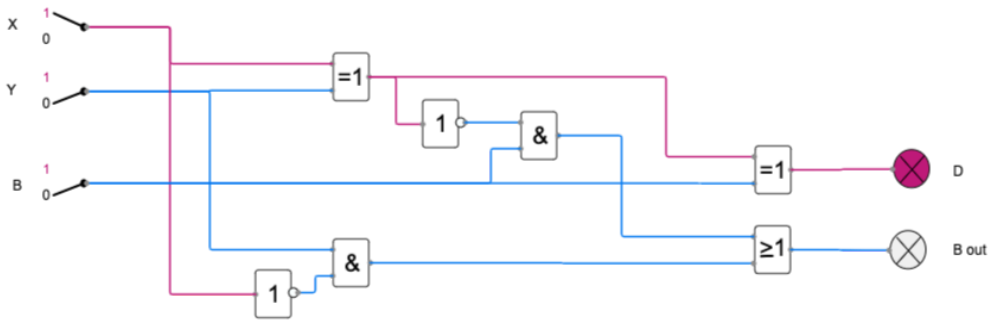


The truth table of the half substractor is:

X	Y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Full subtractor

The full subtractor is a combinational circuit which is used to perform subtraction of three input bits: the minuend X, subtrahend Y, and borrow in B^{in}. The full subtractor generates two output bits: the difference D and borrow out B^{out}. B^{in} is set when the previous digit is borrowed from X. Thus, B^{in} is also subtracted from X as well as the subtrahend Y.



The truth table of the full substractor is:

X	Y	B in	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0

1	0	1	0	0
1	1	0	0	0
1	1	1	1	1