“Note that each of the six control bits instructs the ALU to carry out a certain *elementary* operation. *Taken together, the combined effects of these operations* cause the ALU to computer a variety of useful functions. … Of course, this does not happen miraculously, it’s the result of careful design.” (Nisan 34)

“First we made a list of all the primitive operations that we wanted our computer to be able to perform (right column in figure 2.6). Next, we used backward reasoning to figure out how x, y and out can be manipulated in binary fashion in order to carry out the desired operations. … The resulting ALU is simple and elegant. And in the hardware business simplicity and elegance imply inexpensive and powerful computer systems.” (Nisan 38)

These bits instruct how to preset the x input		These bits instruct how to preset the y input		This bit selects between + / And	This bit inst. how to postset out	Resulting ALU output
zx	nx	zy	ny	f	no	out=
if zx then x=0	if nx then x=!x	if zy then y=0	if ny then y=!y	if f then out=x+y else out=x&y	if no then out=!out	f(x,y)=
1	0	1	0	1	0	0
1	1	1	1	1	1	1
1	1	1	0	1	0	-1
0	0	1	1	0	0	x
1	1	0	0	0	0	y
0	0	1	1	0	1	!x
1	1	0	0	0	1	!y
0	0	1	1	1	1	-x
1	1	0	0	1	1	-y
0	1	1	1	1	1	x+1
1	1	0	1	1	1	y+1
0	0	1	1	1	0	x-1
1	1	0	0	1	0	y-1
0	0	0	0	1	0	x+y
0	1	0	0	1	1	x-y
0	0	0	1	1	1	y-x
0	0	0	0	0	0	x&y
0	1	0	1	0	1	x|y

Figure 2.6 The ALU truth table. Taken together, the binary operations coded by the first six columns effect the function listed in the right column (we use the symbols !, &, and | to represent the operators Not, And, and Or, respectively, performed bit-wise). The complete ALU truth table consists of sixty-four rows, of which only the eighteen presented here are of interest.

No way! Do you really believe that combining four simple operations in a two’s complement binary system can yield the eighteen functions above?

Prove the table to yourself by procedurally working through the implementations of each function (applied to 4-bit numbers for simplicity). Write the decimal equivalent of the arbitrarily provided binary inputs and outputs. Show each step of your work (use the margin if necessary when summing or and’ing).

f(x,y) = 0 0000	1010 -6	0001 1
set x to zero	0000	
don't negate x	0000	
set y to zero		0000
don't negate y		0000
add x and y	0000	
don't negate result	0000 0... duh	

f(x,y) = 1 0001	1010	0001
zx=1		
nx=1		
zy=1		
ny=1		
x+y		
!out		

f(x,y) = -1	1010	0001

Prove to yourself that the answers are correct. First calculate the operation by hand and write the result in binary and decimal. Compare this calculation to the procedural output.

f(x,y) = x 0100 4	0100	0101

f(x,y) = y	1010	0011

f(x,y) = !x	1010	0101

Are you remembering to write the decimal values for each of the inputs and the target answer (upper-left box)?

f(x,y) = !y	1010	0101

f(x,y) = -x	0010	1000

"Computers in the future may weigh no more than 1.5 tons."
- Popular Mechanics, 1949

f(x,y) = -y	1010	0001

f(x,y) = x+1	0001	0001

f(x,y) = y+1	1010	1111

f(x,y) = x-1	0110	0001

"But what...is it good for?"
- Advanced Computing Systems Division of IBM, 1968, commenting on the microchip

f(x,y) = y-1	0001	1111

f(x,y) = x+y	0010	0101

f(x,y) = x-y	0111	0010

"Let's go trashing," someone in the back seat said. "There's a C.O. in Astoria." It's cool to be talking in a kind of hacker's code. The word trashing means climbing around in garbage, where you hope to find computer printouts that list secret passwords and logons. And C.O., as everybody in the Supra knows, means Central Office. As in New York Telephone's Central Office, in Astoria, Queens.
Masters of Deception by Michele Slatalla

f(x,y) = y-x	1101	1111

<table>
<tr><td>f(x,y) = x&y</td><td>1011</td><td>1000</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><td></td><td colspan="2"></td></tr>
</table>

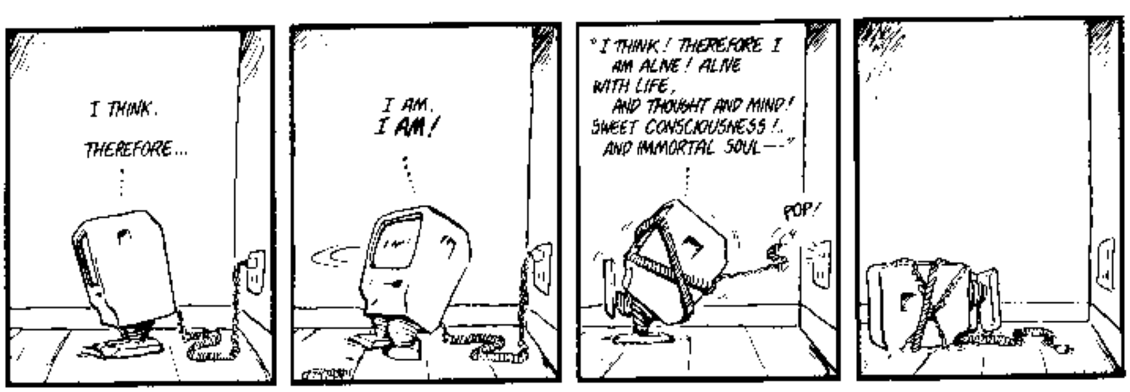
<table>
<tr><td>f(x,y) = x|y</td><td>1111</td><td>1010</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><td></td><td colspan="2"></td></tr>
</table>

My dear creative, emotional, sometimes foolish, opinionated human,

You should now see that the characteristics of binary numbers in the two's complement system coupled with a combination of four simple binary/Boolean operations (zeroing, bitwise negation, adding, or'ing) provides us with at least eighteen simple arithmetic functions.

true,
Banana Jr. 2000

PS. Now go build your ALU.



<u>Bloom County Babylon</u> by Berke Breathed