pre-setting the x input		pre-setting the y input		computing + or &	post-setting the output	resulting ALU output
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	out(x,y) =
zx	**nx**	**zy**	**ny**	**f**	**no**	**out**
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

```
Add16(a= ,b= ,out= ) /* Adds up two 16-bit two's complement values */
ALU(x= ,y= ,zx= ,nx= ,zy= ,ny= ,f= ,no= ,out= ,zr= ,ng= ) /* Hack ALU */
And(a= ,b= ,out= ) /* And gate */
And16(a= ,b= ,out= ) /* 16-bit And */
ARegister(in= ,load= ,out= ) /* Address register (built-in) */
Bit(in= ,load= ,out= ) /* 1-bit register */
CPU(inM= ,instruction= ,reset= ,outM= ,writeM= ,addressM= ,pc= ) /* Hack CPU */
DFF(in= ,out= ) /* Data flip-flop gate (built-in) */
DMux(in= ,sel= ,a= ,b= ) /* Routes the input to one out of two outputs */
DMux4Way(in= ,sel= ,a= ,b= ,c= ,d= ) /* Routes the input to one out of four outputs */
DMux8Way(in= ,sel= ,a= ,b= ,c= ,d= ,e= ,f= ,g= ,h= ) /* Routes the input to one out of 8 outputs */
DRegister(in= ,load= ,out= ) /* Data register (built-in) */
HalfAdder(a= ,b= ,sum= , carry= ) /* Adds up two bits */
FullAdder(a= ,b= ,c= ,sum= ,carry= ) /* Adds up three bits */
Inc16(in= ,out= ) /* Sets out to in + 1 */
Keyboard(out= ) /* Keyboard memory map (built-in) */
Memory(in= ,load= ,address= ,out= ) /* Data memory of the Hack platform (RAM)  */
Mux(a= ,b= ,sel= ,out= ) /* Selects between two inputs */
Mux16(a= ,b= ,sel= ,out= ) /* Selects between two 16-bit inputs */
Mux4Way16(a= ,b= ,c= ,d= ,sel= ,out= ) /* Selects between four 16-bit inputs */
Mux8Way16(a= ,b= ,c= ,d= ,e= ,f= ,g= ,h= ,sel= ,out= ) /* Selects between eight 16-bit inputs */
Nand(a= ,b= ,out= ) /* Nand gate (built-in) */
Not(in= ,out= ) /* Not gate */
Not16(in= ,out= ) /* 16-bit Not */
Or(a= ,b= ,out= ) /* Or gate */
Or16(a= ,b= ,out= ) /* 16-bit Or */
Or8Way(in= ,out= ) /* 8-way Or */
PC(in= ,load= ,inc= ,reset= ,out= ) /* Program Counter */
RAM8(in= ,load= ,address= ,out= ) /* 8-word RAM */
RAM64(in= ,load= ,address= ,out= ) /* 64-word RAM */
RAM512(in= ,load= ,address= ,out= ) /* 512-word RAM */
RAM4K(in= ,load= ,address= ,out= ) /* 4K RAM */
RAM16K(in= ,load= ,address= ,out= ) /* 16K RAM */
Register(in= ,load= ,out= ) /* 16-bit register */
ROM32K(address= ,out= ) /* Instruction memory of the Hack platform (ROM, built-in) */
Screen(in= ,load= ,address= ,out= ) /* Screen memory map (built-in) */
Xor(a= ,b= ,out= ) /* Xor gate */
```