Add16(a = , b = , out = );

ALU(x = , y = , zx = , nx = , zy = , ny = , f = , no = , out = , zr = , ng = );

And16(a = , b = , out = );

And(a = , b = , out = );

ARegister(in = , load = , out = );

Bit(in = , load = , out = );

CPU(inM = , instruction = , reset = , outM = , writeM  = , addressM = , pc = );

DFF(in = , out = );

DMux4Way(in = , sel = , a = , b = , c = , d = );

DMux8Way(in = , sel = , a = , b = , c = , d = , e = , f = , g = , h = );

DMux(in = , sel = , a = , b = );

DRegister(in = , load = , out = );

FullAdder(a = , b = , c = , sum = , carry = );

HalfAdder(a = , b = , sum = , carry = );

Inc16(in = , out = );

Keyboard(out = );

Memory(in = , load = , address = , out = );

Mux16(a = , b = , sel = , out = );

Mux4Way16(a = , b = , c = , d = , sel = , out = );

Mux8Way16(a = , b = , c = , d = , e = , f = , g = , h = , sel = , out = );

Mux(a = , b = , sel = , out = );

Nand(a = , b = , out = );

Not16(in = , out = );

Not(in = , out = );

Or16(a = , b = , out = );

Or8Way(in = , out = );

Or(a = , b = , out = );

PC(in = , load = , inc = , reset = , out = );

RAM16K(in = , load = , address = , out = );

RAM4K(in = , load = , address = , out = );

RAM512(in = , load = , address = , out = );

RAM64(in = , load = , address = , out = );

RAM8(in = , load = , address = , out = );

Register(in = , load = , out = );

ROM32K(address = , out = );

Screen(in = , load = , address = , out = );

Xor(a = , b = , out = );