COMP1036 Computer Fundamentals
Lab 4

---

1. Initially, use the DFF chip as discussed in class. Examine what happens at the output on the tick and tock steps when different inputs are used. Then, using the inbuilt DFF chip, implement the following circuits:

   (a) 1-bit register
   Note: The Hardware Simualtor doesn't like it if you connect the output of the gate you're constructing to the input of another gate within the gate you're constructing. Instead, specify an extra output from the first gate.

   Solution

```
CHIP Bit {
IN in, load;
OUT out;

PARTS:
Mux(a=t1, b=in, sel=load, out=w1);
DFF(in=w1, out=t1, out=out);
}
```

   (b) 16-bit register

```
If load[t-1]=1 then out[t] = in[t-1]
else out does not change (out[t] = out[t-1])
```

   Solution

```
CHIP Register {
IN in[16], load;
OUT out[16];

PARTS:
Bit(in=in[0], load=load, out=out[0]);
Bit(in=in[1], load=load, out=out[1]);
Bit(in=in[2], load=load, out=out[2]);
Bit(in=in[3], load=load, out=out[3]);
Bit(in=in[4], load=load, out=out[4]);
Bit(in=in[5], load=load, out=out[5]);
Bit(in=in[6], load=load, out=out[6]);
Bit(in=in[7], load=load, out=out[7]);
```

```
    Bit(in=in[8], load=load, out=out[8]);
    Bit(in=in[9], load=load, out=out[9]);
    Bit(in=in[10], load=load, out=out[10]);
    Bit(in=in[11], load=load, out=out[11]);
    Bit(in=in[12], load=load, out=out[12]);
    Bit(in=in[13], load=load, out=out[13]);
    Bit(in=in[14], load=load, out=out[14]);
    Bit(in=in[15], load=load, out=out[15]);
}
```

Next, implement the following circuits:

1. Inc16

```
16-bit incrementer.  out = in + 1 (16-bit addition).
Ignore the overflow.
```

Solution

```
CHIP Inc16 {
IN in[16];
OUT out[16];

PARTS:
Add16(a[0..15]=in[0..15], b[0]=true, b[1..15]=false, out[0..15]=out[0..15]);
}
```

2. PC

```
A 16-bit counter with load and reset control bits.
if      (reset[t]==1) out[t+1] = 0
else if (load[t]==1)  out[t+1] = in[t]
else if (inc[t]==1)   out[t+1] = out[t] + 1  (integer addition)
else                  out[t+1] = out[t]
```

Solution

```
CHIP PC {
IN in[16],load,inc,reset;
OUT out[16];

PARTS:
```

```
    // increment the output of the register
    Inc16(in = feedback, out = pc);
    Mux16(a = feedback, b = pc, sel = inc, out = w0);
    Mux16(a = w0, b = in, sel = load, out = w1);
    Mux16(a = w1, b = false, sel = reset, out = cout);
    // the output from the register also needs to get fed back through
    // the combinational logic to get processed for the next clock cycle.
    Register(in = cout, load = true, out = out, out = feedback);
}
```