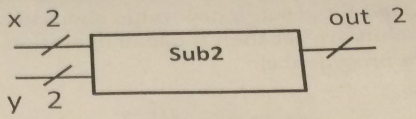
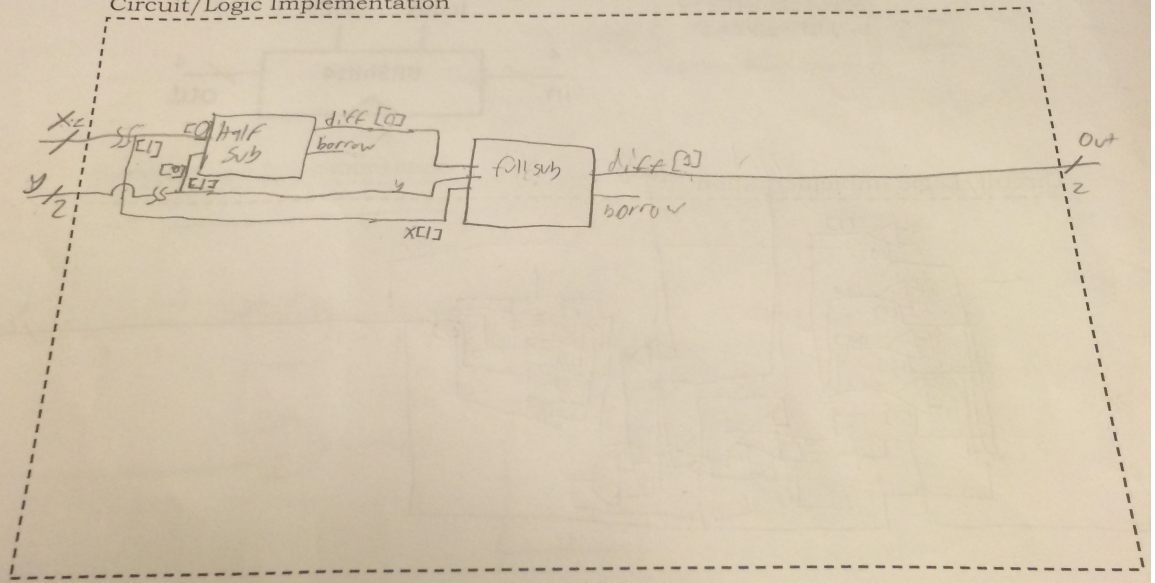
Sub2

A sub2 takes 2 2-bit unsigned numbers, x and y, and takes the difference of the two (i.e., calculates x-y). Implement a Sub2 using a Half Subtractor and a Full Subtractor.

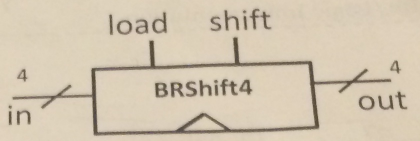


Circuit/Logic Implementation

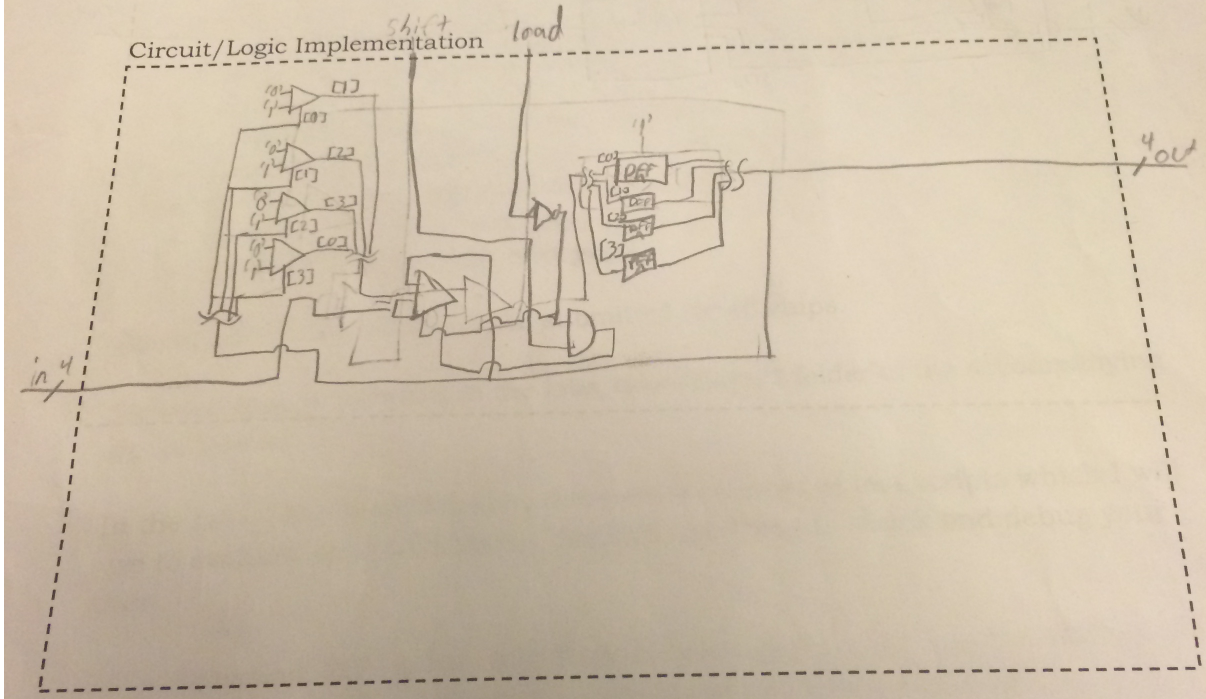


BRShift4

The BRShift4 is a 4 bit register with an additional flag for a shift operation. When shift is set, every bit in the 4 bit register should be shifted right (i.e., bit[i] = bit[i+1] and bit[3] = bit[0]). As an example 1110 would become 0111. When the load bit is set, a new value should be load in to the register. If both load and shift are set then the load operation should be performed (i.e., the load bit takes precedence).



Circuit/Logic Implementation

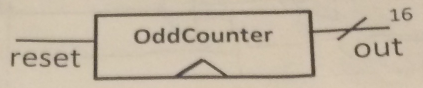



```
Mux(a=false, b=true, sel=in[0], out=in[1]);
Mux(a=false, b=true, sel=in[1], out=in[2]);
Mux(a=false, b=true, sel=in[2], out=in[3]);
Mux(a=false, b=true, sel=in[3], out=in[0]);
```

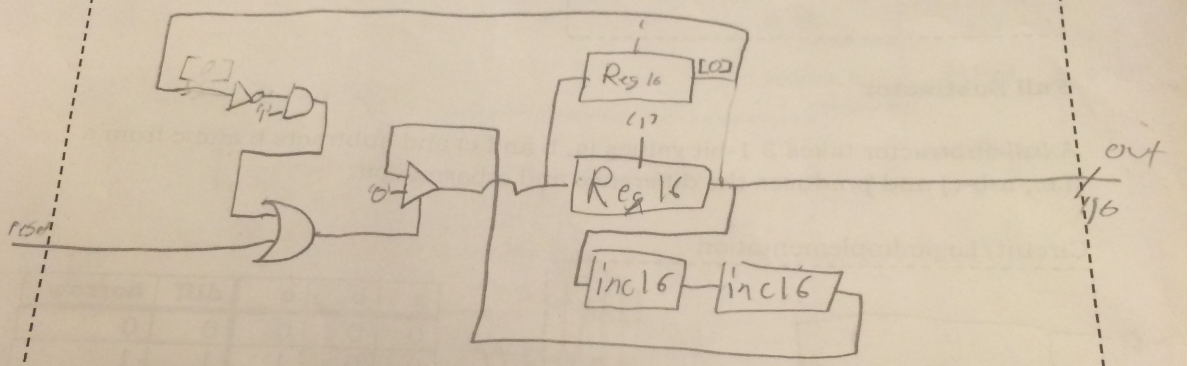
OddCounter

The OddCounter should continually count from 1 to 65535 in unsigned binary. On each clock cycle, the value of the count should be incremented by 2. The counter should have a reset bit to restart the counter at 1 and also automatically restart at 1 if the value of the counter is an even number. The output is a 16 bit value.

[15]=1

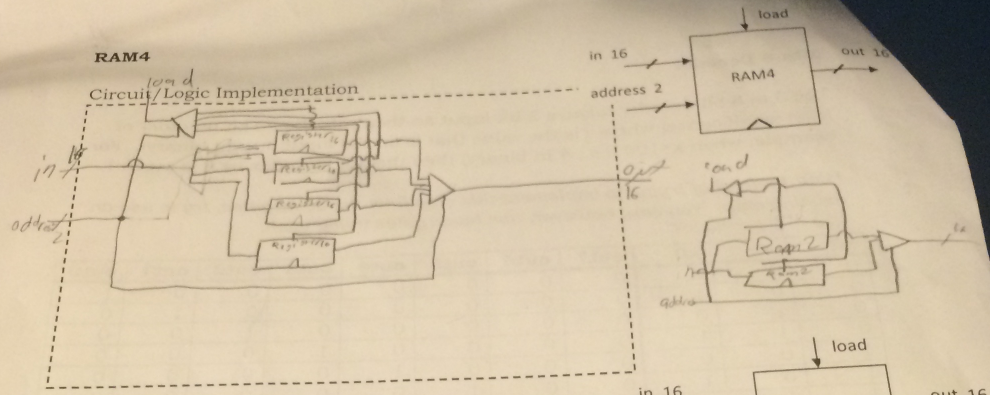


Circuit/Logic Implementation



RAM4

Circuit/Logic Implementation



RAM32

Circuit/Logic Implementation

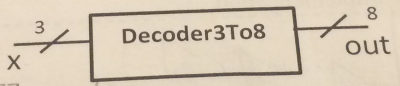
Note: For the RAM32 you may leave some the RAM4 chips out the diagram as long your logic is clear.

3-to-8 Decoder

The 3 to 8 bit decoder takes a 3 bit input an then sets the *ith* bit (in order of least significance) where *i* is the value that is on the input bus (in binary). For example, when x=100 (i.e., 4 in binary) then the out bit index by 4 is set to 1.

Hint: Instead of trying to implement this chip from the basic gates, try to use an existing chip. You can, however, use basic gates if you wish.

x2	x1	x0	out7	out6	out5	out4	out3	out2	out1	out0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0



x[0..2] and out[0..7] negative

Circuit/Logic Implementation