Raul Aguilar

CS 220 2367

September 6, 2019

CHIP Mux {

IN a, b, sel;

OUT out;

PARTS:

Not(in=sel, out=nsel);

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

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

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

}

CHIP DMux {

IN in, sel;

OUT a, b;

PARTS:

Not(in=sel, out=nsel);

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

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

}

CHIP Not16 {

IN in[16];

OUT out[16];

PARTS:

Not(in=in[0], out=out[0]);

Not(in=in[1], out=out[1]);

Not(in=in[2], out=out[2]);

Not(in=in[3], out=out[3]);

Not(in=in[4], out=out[4]);

Not(in=in[5], out=out[5]);

Not(in=in[6], out=out[6]);

Not(in=in[7], out=out[7]);

Not(in=in[8], out=out[8]);

Not(in=in[9], out=out[9]);

Not(in=in[10], out=out[10]);

Not(in=in[11], out=out[11]);

Not(in=in[12], out=out[12]);

Not(in=in[13], out=out[13]);

Not(in=in[14], out=out[14]);

Not(in=in[15], out=out[15]);

}

CHIP And16 {

IN a[16], b[16];

OUT out[16];

PARTS:

And(a=a[0], b=b[0], out=out[0]);

And(a=a[1], b=b[1], out=out[1]);

And(a=a[2], b=b[2], out=out[2]);

And(a=a[3], b=b[3], out=out[3]);

And(a=a[4], b=b[4], out=out[4]);

And(a=a[5], b=b[5], out=out[5]);

And(a=a[6], b=b[6], out=out[6]);

And(a=a[7], b=b[7], out=out[7]);

And(a=a[8], b=b[8], out=out[8]);

And(a=a[9], b=b[9], out=out[9]);

And(a=a[10], b=b[10], out=out[10]);

And(a=a[11], b=b[11], out=out[11]);

And(a=a[12], b=b[12], out=out[12]);

And(a=a[13], b=b[13], out=out[13]);

And(a=a[14], b=b[14], out=out[14]);

And(a=a[15], b=b[15], out=out[15]);

}

CHIP Or16 {

IN a[16], b[16];

OUT out[16];

PARTS:

// Put your code here:

Or(a=a[0], b=b[0], out=out[0]);

Or(a=a[1], b=b[1], out=out[1]);

Or(a=a[2], b=b[2], out=out[2]);

Or(a=a[3], b=b[3], out=out[3]);

Or(a=a[4], b=b[4], out=out[4]);

Or(a=a[5], b=b[5], out=out[5]);

Or(a=a[6], b=b[6], out=out[6]);

Or(a=a[7], b=b[7], out=out[7]);

Or(a=a[8], b=b[8], out=out[8]);

Or(a=a[9], b=b[9], out=out[9]);

Or(a=a[10], b=b[10], out=out[10]);

Or(a=a[11], b=b[11], out=out[11]);

Or(a=a[12], b=b[12], out=out[12]);

Or(a=a[13], b=b[13], out=out[13]);

Or(a=a[14], b=b[14], out=out[14]);

Or(a=a[15], b=b[15], out=out[15]);

}

CHIP Mux16 {

IN a[16], b[16], sel;

OUT out[16];

PARTS:

// Put your code here:

Mux(a=a[0], b=b[0], sel=sel, out=out[0]);

Mux(a=a[1], b=b[1], sel=sel, out=out[1]);

Mux(a=a[2], b=b[2], sel=sel, out=out[2]);

Mux(a=a[3], b=b[3], sel=sel, out=out[3]);

Mux(a=a[4], b=b[4], sel=sel, out=out[4]);

Mux(a=a[5], b=b[5], sel=sel, out=out[5]);

Mux(a=a[6], b=b[6], sel=sel, out=out[6]);

Mux(a=a[7], b=b[7], sel=sel, out=out[7]);

Mux(a=a[8], b=b[8], sel=sel, out=out[8]);

Mux(a=a[9], b=b[9], sel=sel, out=out[9]);

Mux(a=a[10], b=b[10], sel=sel, out=out[10]);

Mux(a=a[11], b=b[11], sel=sel, out=out[11]);

Mux(a=a[12], b=b[12], sel=sel, out=out[12]);

Mux(a=a[13], b=b[13], sel=sel, out=out[13]);

Mux(a=a[14], b=b[14], sel=sel, out=out[14]);

Mux(a=a[15], b=b[15], sel=sel, out=out[15]);

}

CHIP Or8Way {

IN in[8];

OUT out;

PARTS:

// Put your code here:

Or(a=in[0], b=in[1], out=or1);

Or(a=in[2], b=in[3], out=or2);

Or(a=in[4], b=in[5], out=or3);

Or(a=in[6], b=in[7], out=or4);

Or(a=or1, b=or2, out=or5);

Or(a=or3, b=or4, out=or6);

Or(a=or5, b=or6, out=out);

}

CHIP Mux4Way16 {

IN a[16], b[16], c[16], d[16], sel[2];

OUT out[16];

PARTS:

// Put your code here:

Mux(a=a[0],b=b[0],sel=sel[0],out=muxab0);

Mux(a=c[0],b=d[0],sel=sel[0],out=muxcd0);

Mux(a=muxab0,b=muxcd0,sel=sel[1],out=out[0]);

Mux(a=a[1],b=b[1],sel=sel[0],out=muxab1);

Mux(a=c[1],b=d[1],sel=sel[0],out=muxcd1);

Mux(a=muxab1,b=muxcd1,sel=sel[1],out=out[1]);

Mux(a=a[2],b=b[2],sel=sel[0],out=muxab2);

Mux(a=c[2],b=d[2],sel=sel[0],out=muxcd2);

Mux(a=muxab2,b=muxcd2,sel=sel[1],out=out[2]);

Mux(a=a[3],b=b[3],sel=sel[0],out=muxab3);

Mux(a=c[3],b=d[3],sel=sel[0],out=muxcd3);

Mux(a=muxab3,b=muxcd3,sel=sel[1],out=out[3]);

Mux(a=a[4],b=b[4],sel=sel[0],out=muxab4);

Mux(a=c[4],b=d[4],sel=sel[0],out=muxcd4);

Mux(a=muxab4,b=muxcd4,sel=sel[1],out=out[4]);

Mux(a=a[5],b=b[5],sel=sel[0],out=muxab5);

Mux(a=c[5],b=d[5],sel=sel[0],out=muxcd5);

Mux(a=muxab5,b=muxcd5,sel=sel[1],out=out[5]);

Mux(a=a[6],b=b[6],sel=sel[0],out=muxab6);

Mux(a=c[6],b=d[6],sel=sel[0],out=muxcd6);

Mux(a=muxab6,b=muxcd6,sel=sel[1],out=out[6]);

Mux(a=a[7],b=b[7],sel=sel[0],out=muxab7);

Mux(a=c[7],b=d[7],sel=sel[0],out=muxcd7);

Mux(a=muxab7,b=muxcd7,sel=sel[1],out=out[7]);

Mux(a=a[8],b=b[8],sel=sel[0],out=muxab8);

Mux(a=c[8],b=d[8],sel=sel[0],out=muxcd8);

Mux(a=muxab8,b=muxcd8,sel=sel[1],out=out[8]);

Mux(a=a[9],b=b[9],sel=sel[0],out=muxab9);

Mux(a=c[9],b=d[9],sel=sel[0],out=muxcd9);

Mux(a=muxab9,b=muxcd9,sel=sel[1],out=out[9]);

Mux(a=a[10],b=b[10],sel=sel[0],out=muxab10);

Mux(a=c[10],b=d[10],sel=sel[0],out=muxcd10);

Mux(a=muxab10,b=muxcd10,sel=sel[1],out=out[10]);

Mux(a=a[11],b=b[11],sel=sel[0],out=muxab11);

Mux(a=c[11],b=d[11],sel=sel[0],out=muxcd11);

Mux(a=muxab11,b=muxcd11,sel=sel[1],out=out[11]);

Mux(a=a[12],b=b[12],sel=sel[0],out=muxab12);

Mux(a=c[12],b=d[12],sel=sel[0],out=muxcd12);

Mux(a=muxab12,b=muxcd12,sel=sel[1],out=out[12]);

Mux(a=a[13],b=b[13],sel=sel[0],out=muxab13);

Mux(a=c[13],b=d[13],sel=sel[0],out=muxcd13);

Mux(a=muxab13,b=muxcd13,sel=sel[1],out=out[13]);

Mux(a=a[14],b=b[14],sel=sel[0],out=muxab14);

Mux(a=c[14],b=d[14],sel=sel[0],out=muxcd14);

Mux(a=muxab14,b=muxcd14,sel=sel[1],out=out[14]);

Mux(a=a[15],b=b[15],sel=sel[0],out=muxab15);

Mux(a=c[15],b=d[15],sel=sel[0],out=muxcd15);

Mux(a=muxab15,b=muxcd15,sel=sel[1],out=out[15]);

}

CHIP Mux8Way16 {

IN a[16], b[16], c[16], d[16],

e[16], f[16], g[16], h[16],

sel[3];

OUT out[16];

PARTS:

// Put your code here:

Mux4Way16(a=a, b=b, c=c, d=d, sel[1]=sel[1], sel[0]=sel[0], out=muxabcd);

Mux4Way16(a=e, b=f, c=g, d=h, sel[1]=sel[1], sel[0]=sel[0], out=muxefgh);

Mux16(a=muxabcd, b=muxefgh, sel=sel[2], out=out);

}