Computer Organization

Multi-Bit & Multi-Way Logic Gates

Multi-Style Gates

- We can use our Elementary & Composite Gates to create even more complex gates that can help short-hand a lot of our future work

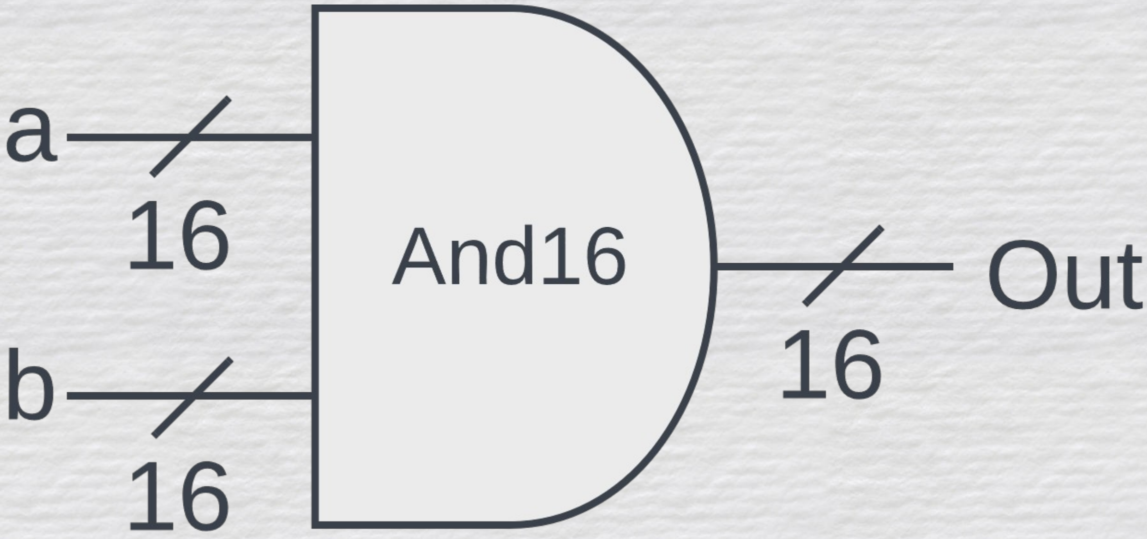
- Multi-Bit Gates
  - Takes a sequence of bits and passes it through a logical gate
- Multi-Way Gates
  - Extends the inputs/outputs of basic gates (4-Way, 8-Way16, etc.)

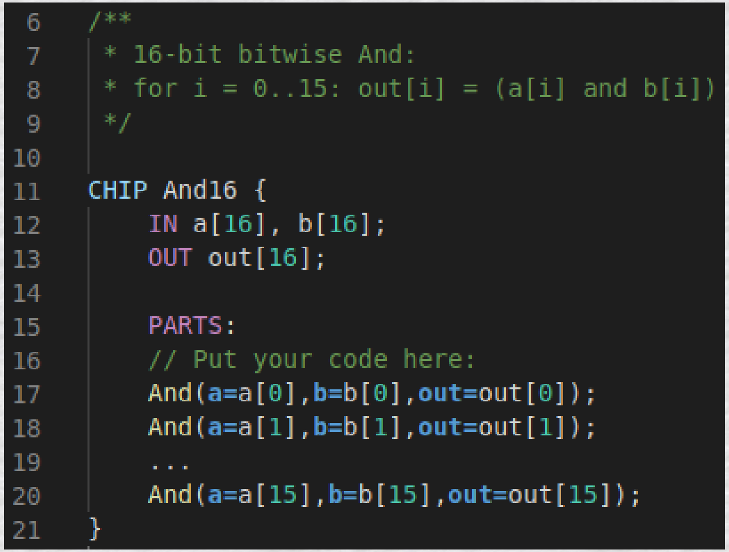
Multi-Bit gates

- Takes sequences of bits as inputs and returns a new sequence of bits based on the logical gate type
  - And, Or, Xor, Not, Mux, DMux

a[16]	1	0	1	0	1	0	1	1	0	1	0	1	1	1	0	0
b[16]	0	0	1	0	1	1	0	1	0	0	1	0	1	0	1	0
a[16] And b[16]	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0

And16 Gate



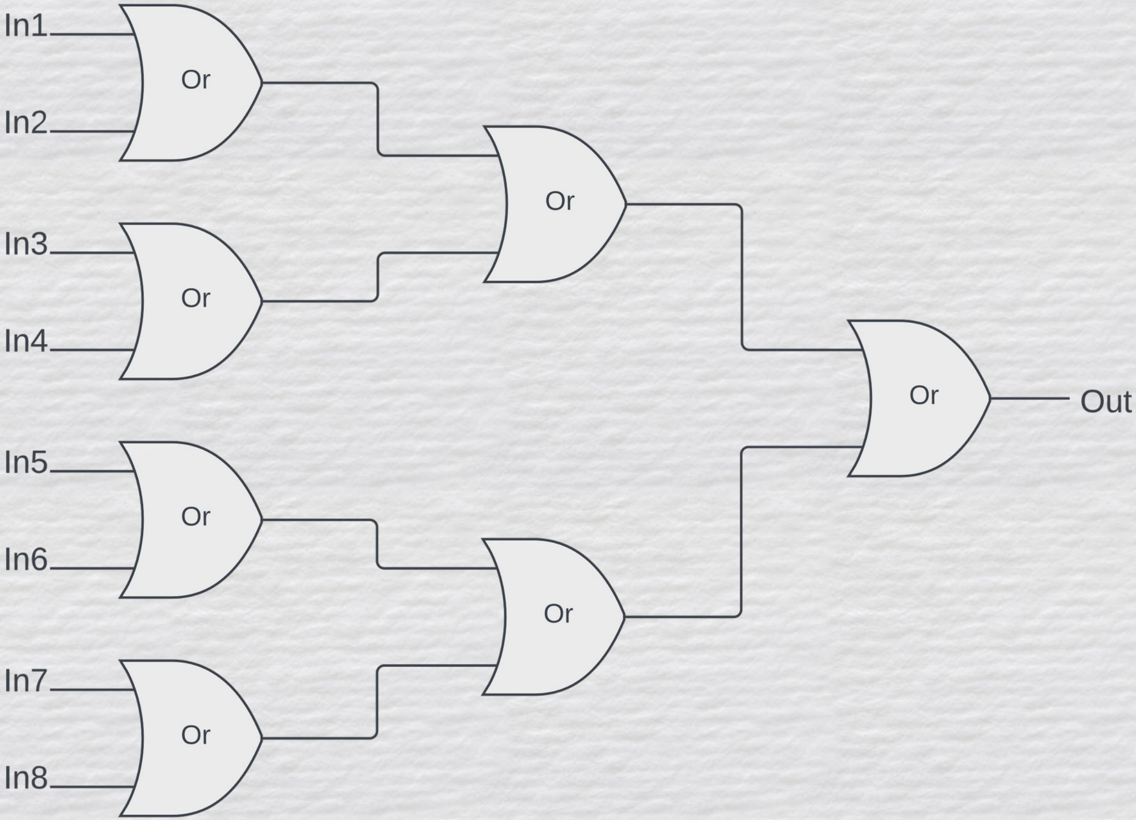


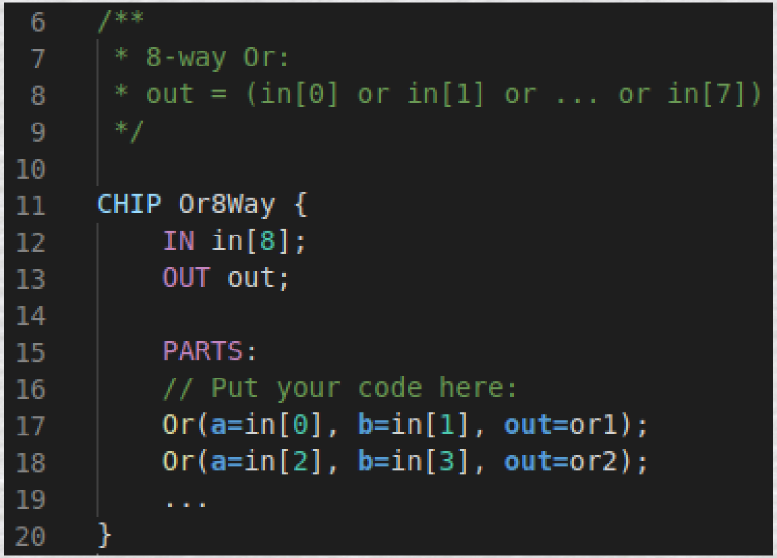
```
/**
 * 16-bit bitwise And:
 * for i = 0..15: out[i] = (a[i] and b[i])
 */

CHIP And16 {
    IN a[16], b[16];
    OUT out[16];

    PARTS:
    // Put your code here:
    And(a=a[0],b=b[0],out=out[0]);
    And(a=a[1],b=b[1],out=out[1]);
    ...
    And(a=a[15],b=b[15],out=out[15]);
}
```

Multi-Way Gates

- Used to expand the number of inputs/outputs a logical gate can compute
    - Or 8-Way
    - Mux 4-Way16
    - DMux 4-Way

Or 8-Way Gate





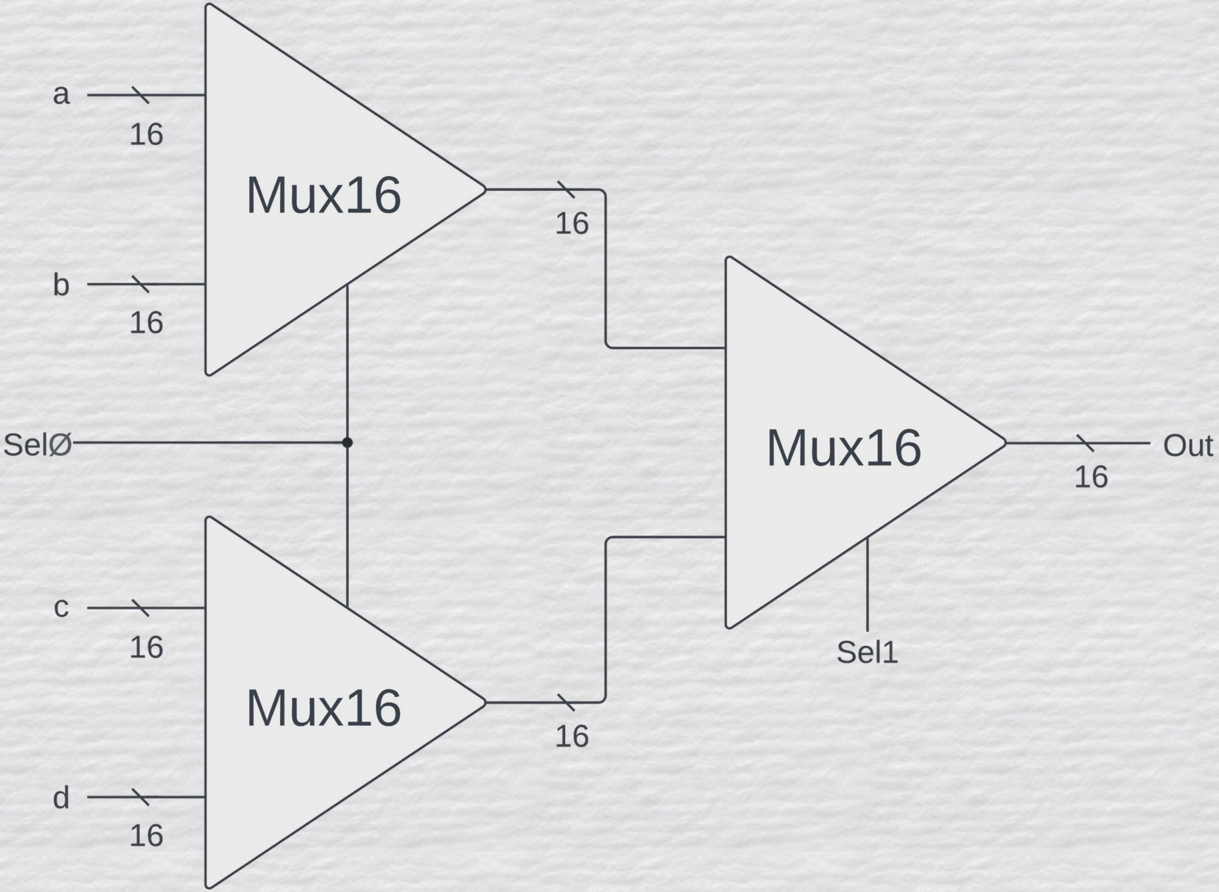
```
/**
 * 8-way Or:
 * out = (in[0] or in[1] or ... or in[7])
 */

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);
    ...
}
```

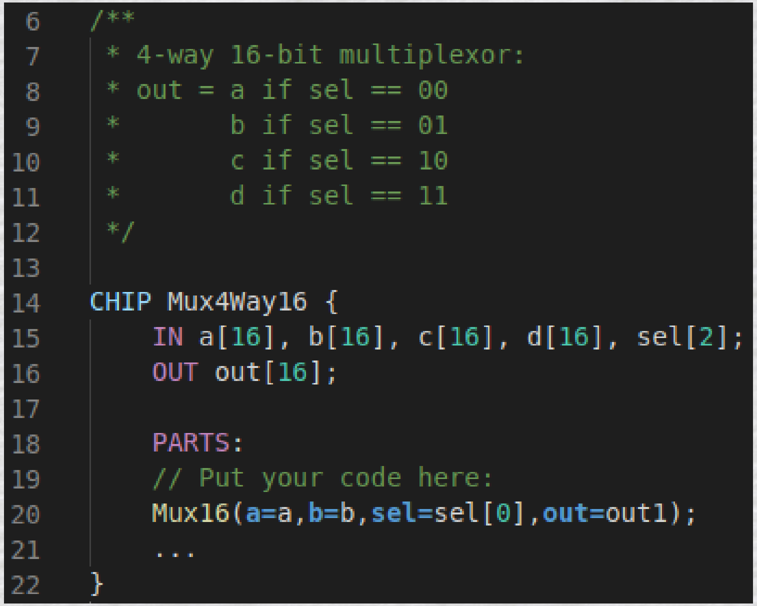
Multi-Way Mux & DMux

- Works just like regular Muxes and DMuxes, but the Select pins are very important regarding their ordering

- N = Total Number of Select Pins
  - Multi-Way Mux
    - Count Select Pins: 0, 1, … N
  - Mutli-Way DMux
    - Count Select Pins: N, N – 1, ... 0

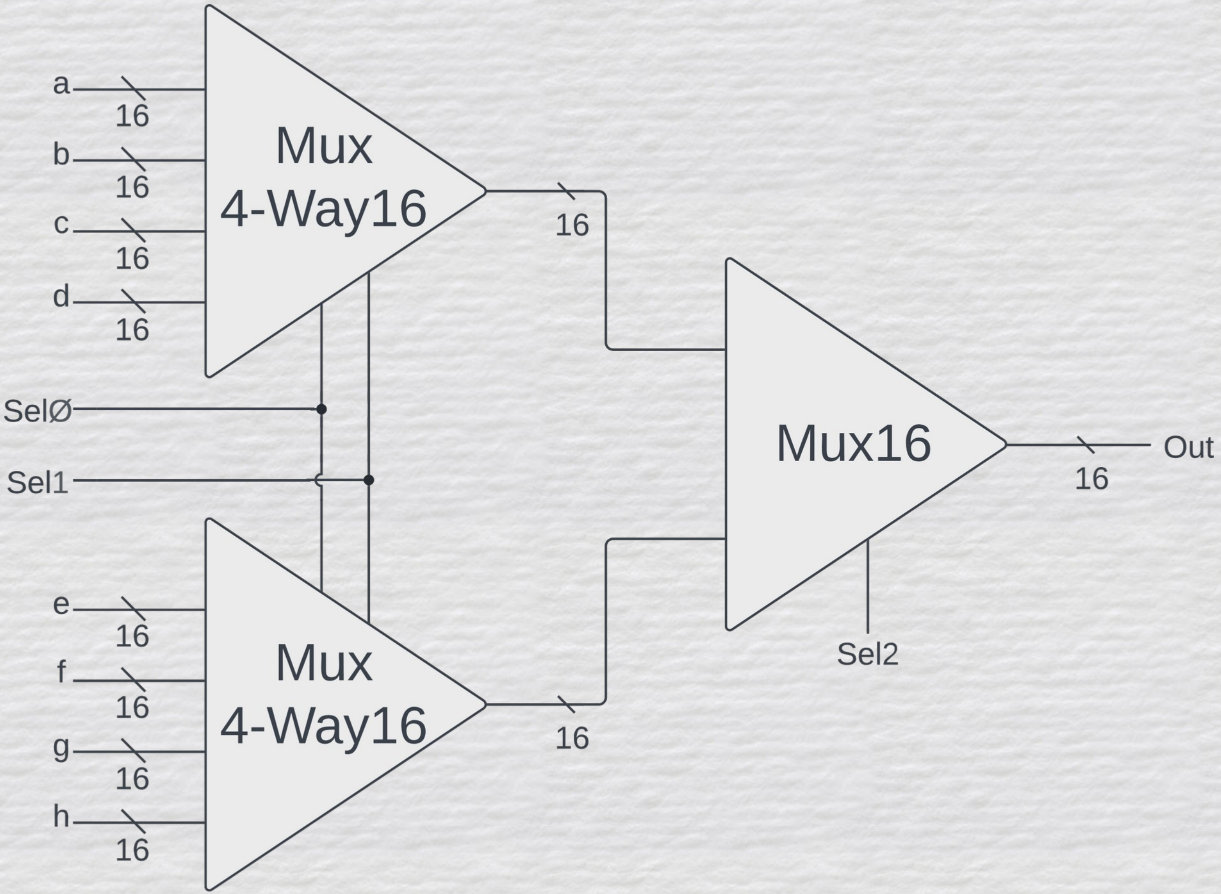
Mux 4-Way16

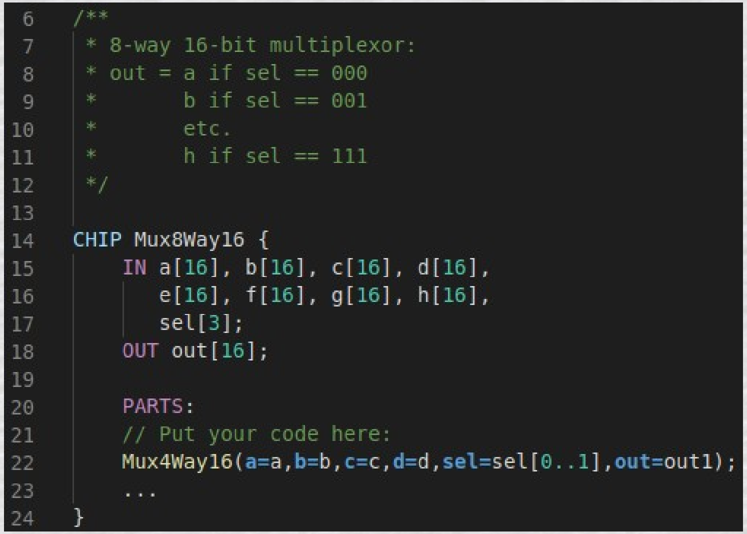




```
/**
 * 4-way 16-bit multiplexor:
 * out = a if sel == 00
 *       b if sel == 01
 *       c if sel == 10
 *       d if sel == 11
 */

CHIP Mux4Way16 {
    IN a[16], b[16], c[16], d[16], sel[2];
    OUT out[16];

    PARTS:
    // Put your code here:
    Mux16(a=a,b=b,sel=sel[0],out=out1);
    ...
}
```

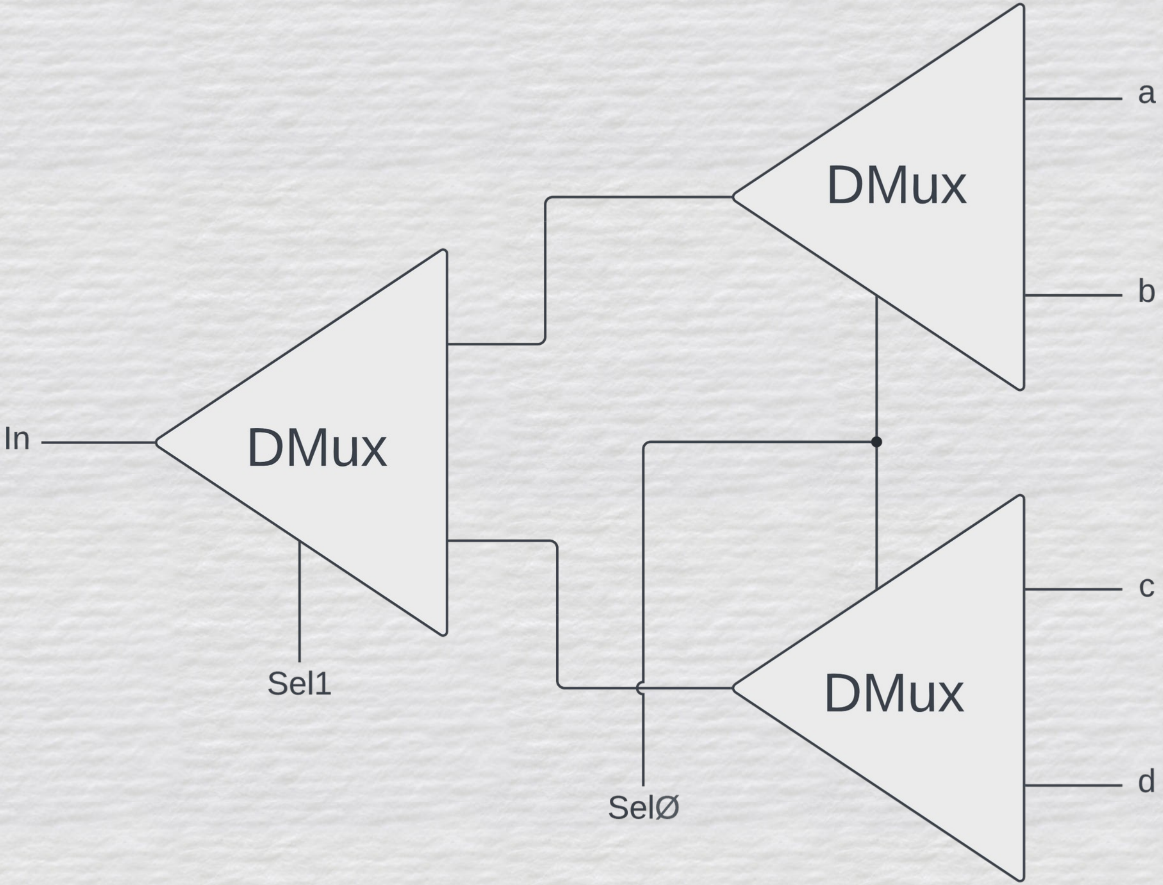
Mux 8-Way16

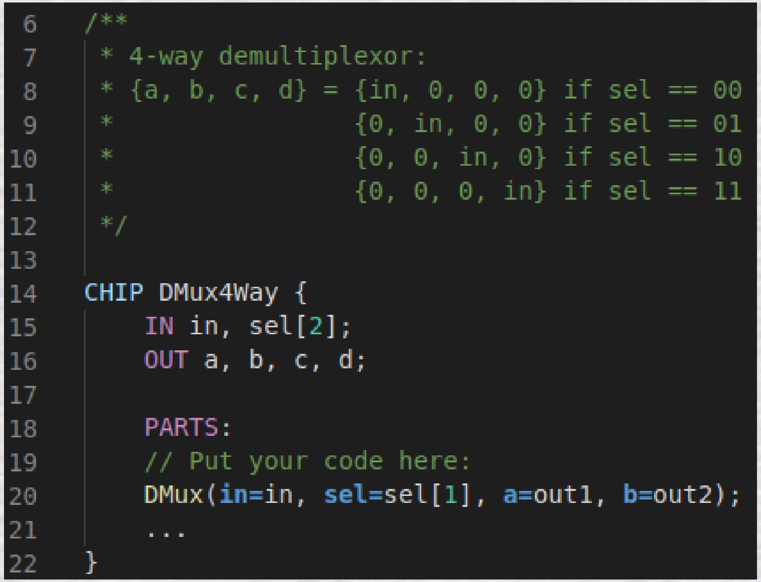




```
/**
 * 8-way 16-bit multiplexor:
 * out = a if sel == 000
 *       b if sel == 001
 *       etc.
 *       h if sel == 111
 */

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=sel[0..1],out=out1);
    ...
}
```

Dmux 4-Way





```
/**
 * 4-way demultiplexor:
 * {a, b, c, d} = {in, 0, 0, 0} if sel == 00
 *                {0, in, 0, 0} if sel == 01
 *                {0, 0, in, 0} if sel == 10
 *                {0, 0, 0, in} if sel == 11
 */

CHIP DMux4Way {
    IN in, sel[2];
    OUT a, b, c, d;

    PARTS:
    // Put your code here:
    DMux(in=in, sel=sel[1], a=out1, b=out2);
    ...
}
```