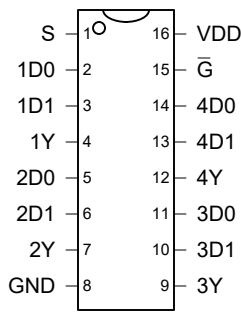


74153 4:1 Mux

Two 4:1 Multiplexers

D_{3:0}: data
S_{1:0}: select
Y: output
Gb: enable

```
always @(1Gb, S, 1D)
    if (1Gb) 1Y = 0;
    else 1Y = 1D[S];
always @(2Gb, S, 2D)
    if (2Gb) 2Y = 0;
    else 2Y = 2D[S];
```

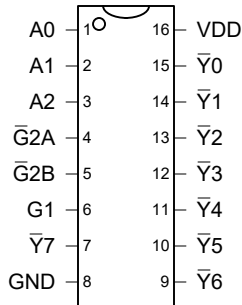


74157 2:1 Mux

Four 2:1 Multiplexers

D_{4:0}: data
S: select
Y: output
Gb: enable

```
always @(*)
    if (~Gb) 1Y = 0;
    else 1Y = S ? 1D[1] : 1D[0];
    if (~Gb) 2Y = 0;
    else 2Y = S ? 2D[1] : 2D[0];
    if (~Gb) 3Y = 0;
    else 3Y = S ? 3D[1] : 3D[0];
    if (~Gb) 4Y = 0;
    else 4Y = S ? 4D[1] : 4D[0];
```

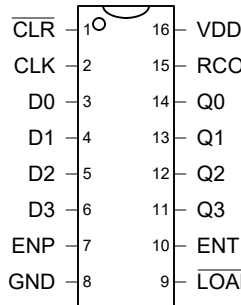


74138 3:8 Decoder

3:8 Decoder

A_{2:0}: address
Y_{b7:0}: output
G1: active high enable
G2: active low enables

G1	G2A	G2B	A2:0	Y7:0
0	x	x	xxx	11111111
1	1	x	xxx	11111111
1	0	1	xxx	11111111
1	0	0	000	11111110
1	0	0	001	11111101
1	0	0	010	11111011
1	0	0	011	11110111
1	0	0	100	11101111
1	0	0	101	11011111
1	0	0	110	10111111
1	0	0	111	01111111



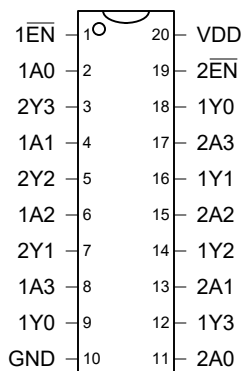
74161/163 Counter

4-bit Counter

CLK: clock
Q_{3:0}: counter output
D_{3:0}: parallel input
CLRb: async reset (161)
sync reset (163)
LOADb: load Q from D
ENP, ENT: enables
RCO: ripple carry out

```
always @(posedge CLK) // 74163
    if (~CLRb) Q <= 4'b0000;
    else if (~LOADb) Q <= D;
    else if (ENP & ENT) Q <= Q+1;

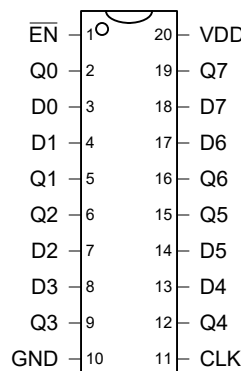
assign RCO = (Q == 4'b1111) & ENT;
```



8-bit Tristate Buffer

A_{3:0}: input
Y_{3:0}: output
ENb: enable

```
assign 1Y =
    1ENb ? 4'bzzzz : 1A;
assign
    2Y = 2ENB ? 4'bzzzz : 2A;
```



8-bit Enableable Register

CLK: clock
D_{7:0}: data
Q_{7:0}: output
ENb: enable

```
always @(posedge clk)
    if (~ENb) Q <= D;
```