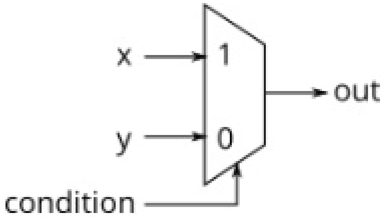
Always if ○

← alwaysblock2 ✓ Previous Next always_if2 ○ →

An `if` statement usually creates a 2-to-1 multiplexer, selecting one input if the condition is true, and the other input if the condition is false.



```
always @(*) begin
    if (condition) begin
        out = x;
    end
    else begin
        out = y;
    end
end
```

This is equivalent to using a continuous `assign`ment with a conditional operator:

```
assign out = (condition) ? x : y;
```

However, the procedural `if` statement provides a new way to make mistakes. The circuit is combinational only if `out` is always assigned a value.

A bit of practice

Build a 2-to-1 mux that chooses between a and b. Choose b if *both* `sel_b1` and `sel_b2` are true. Otherwise, choose a. Do the same twice, once using `assign` statements and once using a procedural if statement.

sel_b1	sel_b2	out_assign out_always
0	0	a
0	1	a
1	0	a
1	1	b