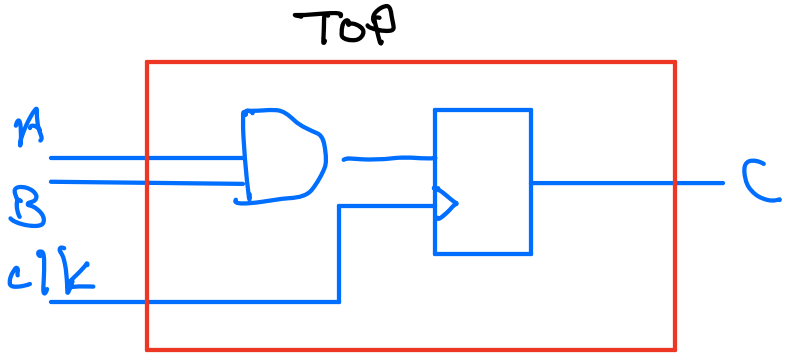
A "testbench" is a verilog program that can drive a verilog (or any) circuit, simulating the inputs to that circuit to test the response

Response: 1. "functional" - test the logic
2. timing - to make sure things happen when they should

#1 is easy

#2 is only as good as the ability to simulate actual timing

ex circuit to test:



verilog code:

```
module top(
  input A, B, clk,
  output reg C);
```

```
wire d = A & B;
always @(posedge clk) C <= d;
endmodule
```

verilog test bench:

```
`timescale 1ns/1ps
module top_tb;
```

`timescale a/b  (` = backquote, no ; at end)

a ≡ time unit

b ≡ precision

time units are used w/ # sign in verilog to add a delay

ex:
```
`timescale 1ns/1ps
a=0;
#1 a=1;
#5 a=0;
```

this means 1. set a=0

2. wait 1 "tick" (1ns) and then set a=1

3. wait 5ns and set a=0

Next: we need to define inputs & outputs to the circuit

circuit inputs: A, B, clk

we need to drive these from the testbench so they will be reg's

```
reg a, b, clock;
```

≠ names do not have to match!

circuit outputs: C

the testbench looks at C which is driven by the circuit

```
wire c;
```

now we instantiate the circuit:

```
top MYTOP (.A(a), .B(b), .clk(clock),
           .C(c));
```

circuit module name

instantiation name (anything!)

now we need to drive the inputs

lets say we want a 1kHz clock

$\Rightarrow$ period is $1\text{ms} = 10^6\text{ns}$ or 10^6 ticks

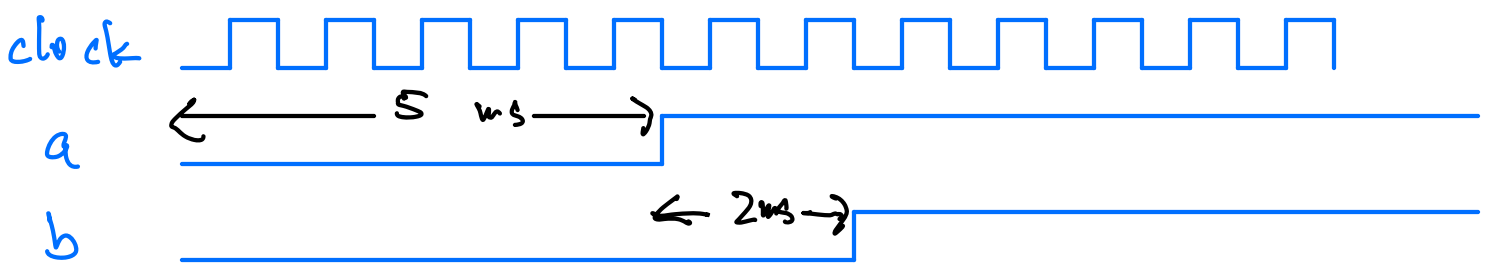
```
always begin
    clock = 0;          start at 0
  #500000 clock = 1;    set to 1 after ½ ms
  #500000 ;             wait ½ ms
end
```

} repeat forever

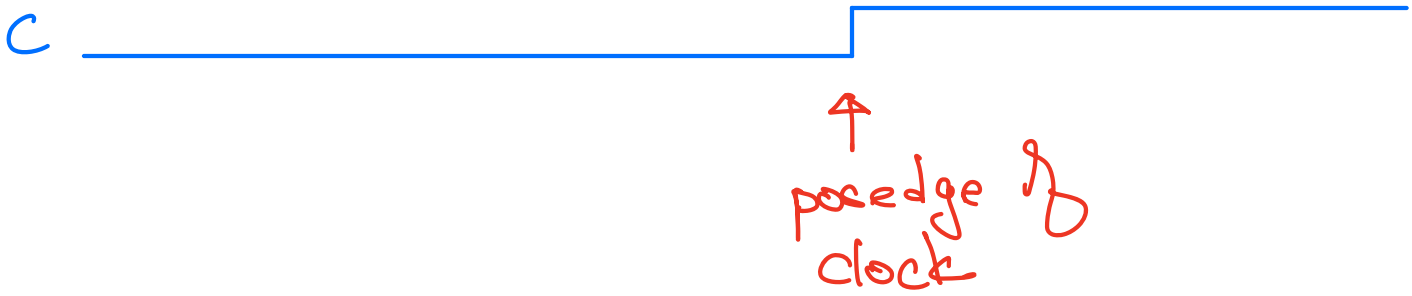
now we have to specify A, B inputs

```
initial begin
    a = 0;
    b = 0;
  #5000000 a=1;   after 5 ms, set a=1
  #2000000 b=1;   after 2 ms, set b=1
end
```

so input looks like this



then you should see C do this


C
posedge
clock