

NMK20603

Computer Architecture

Describing Combinational Logic

Tool installed?

Homework?

Coding style:

- ▶ camelcase signal
- ▶ prefix i,o,t

Recap:

- ▶ module description
- ▶ bitwise operators: $\sim$, $\&$, $|$
- ▶ concatenation op: $\{ \}$

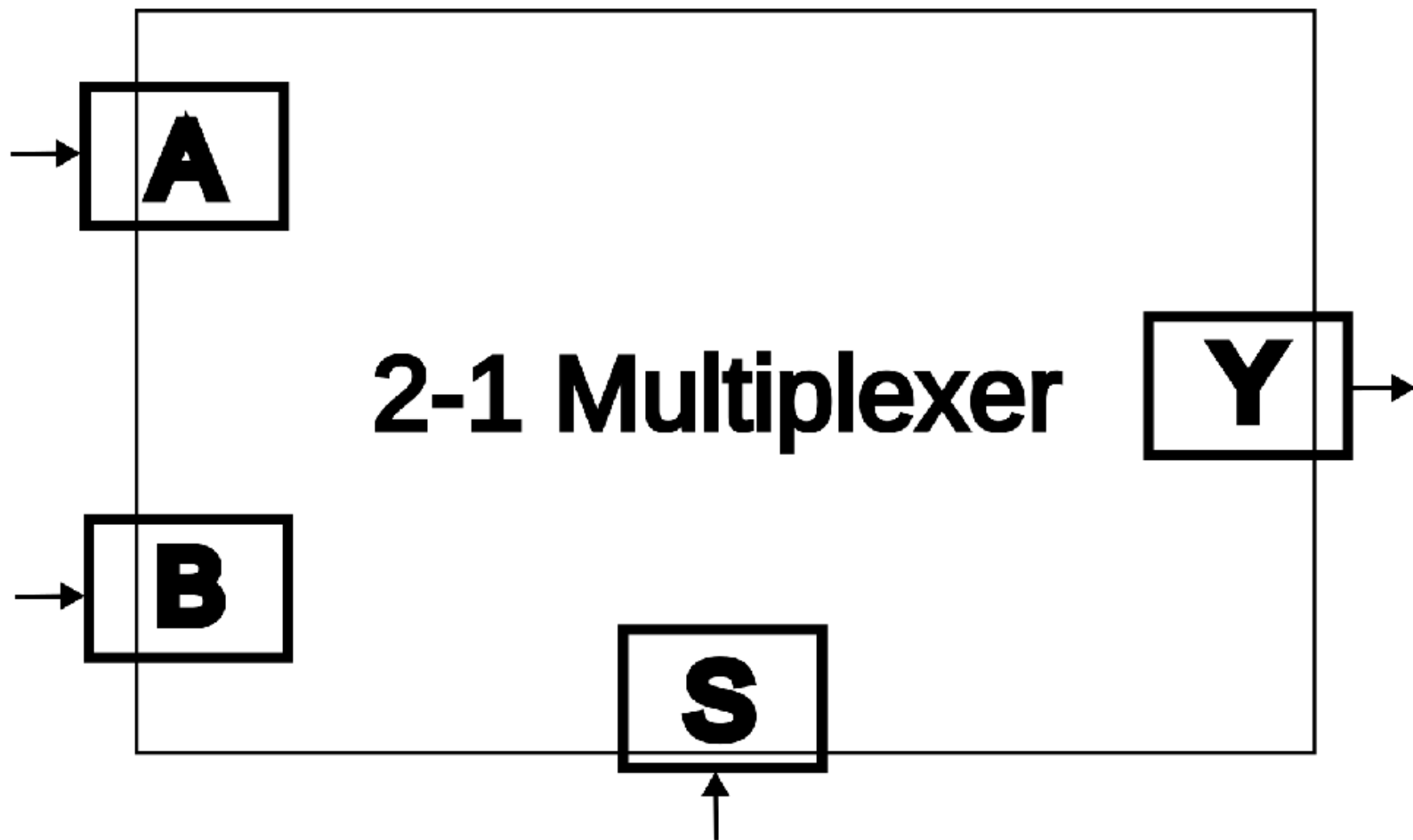
Let's do
CA component!

Hardware

if-else

2-to-1 (@2-1)

Multiplexer



$\Rightarrow$ if $S=0$: $Y=A$

$\Rightarrow$ if $S=1$: $Y=B$

Truth table

⇒ 'simplified'

S	Y
$\emptyset$	A
1	B

AND/OR logic

revisited

⇒ as inhibitor

AND gate

⇒ can force 0

⇒ enabler!

OR gate

⇒ can force 1

⇒ selector!

Build equation!

Truth table

⇒ 'full'

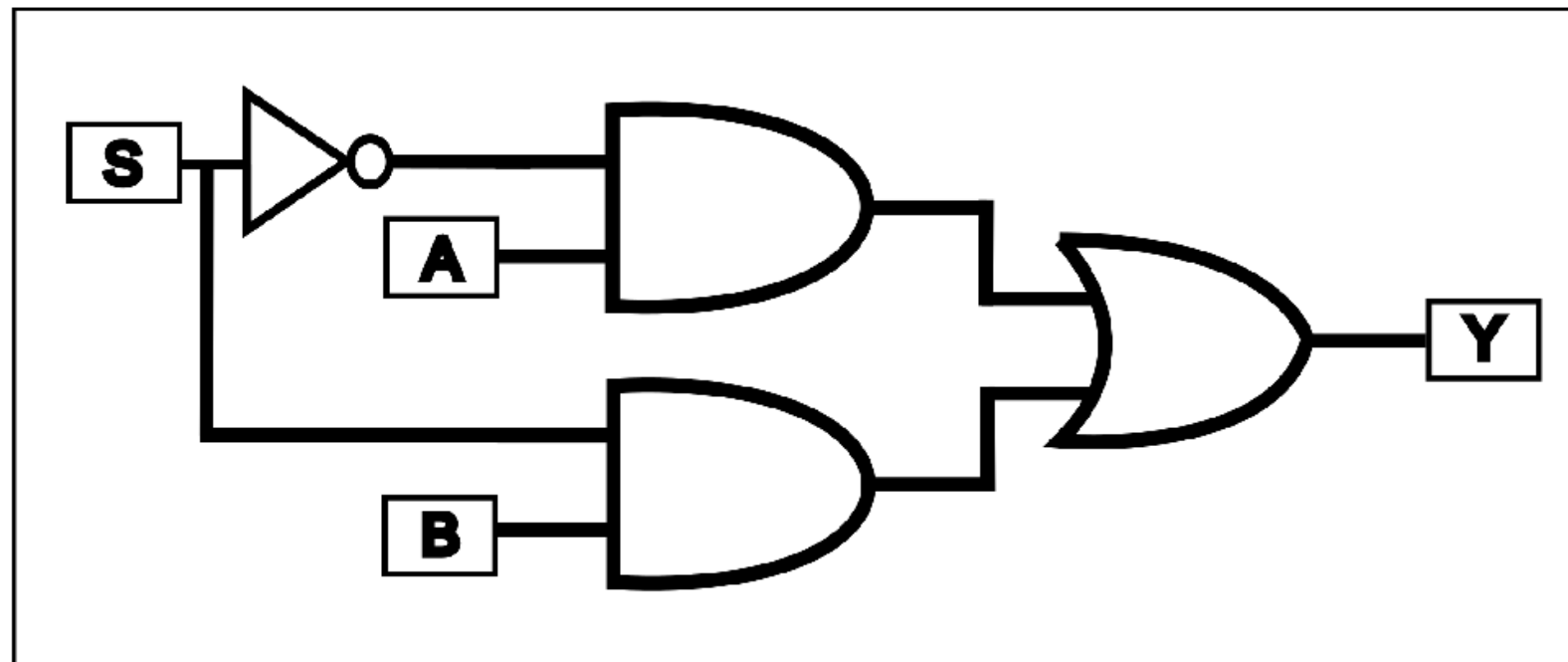
S	A	B	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Get SOP equation!

- More than 1
method to build!

Schematic!

mux21



Describe

mux21_1b

```
module mux21_1b (iS, iA, iB, oY);  
  input iS, iA, iB;  
  output oY;  
  wire oY, tA, tB;  
  assign tA = ~iS&iA;  
  assign tB = iS&iB;  
  assign oY = tA | tB;  
endmodule
```

Testbench:

- ▶ mux21_1b_tb

⇒ refer to 3i1o tb

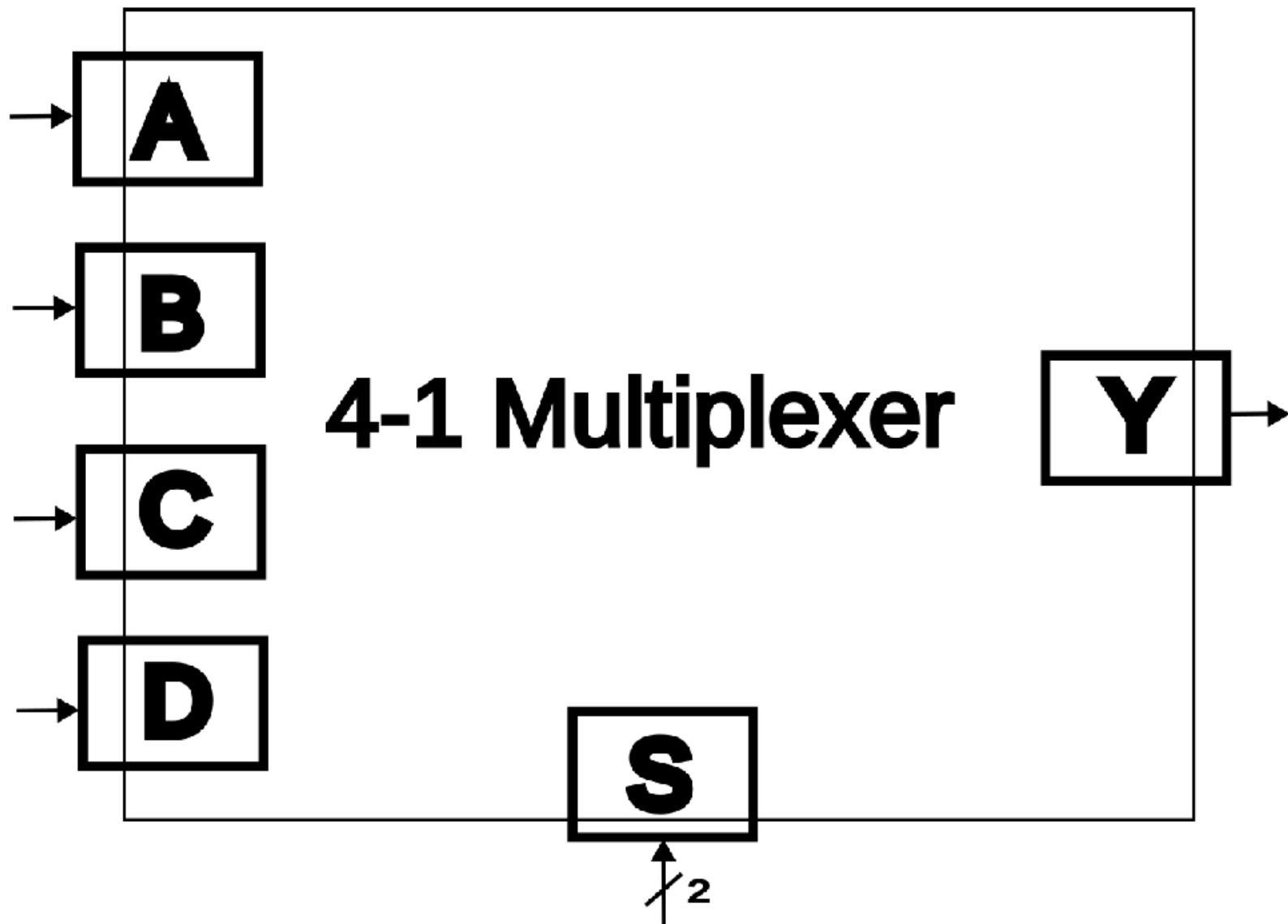
Describe

mux21_1b_tb

```
module mux21_1b_tb ();  
  reg dS,dA,dB;  
  wire mY;  
  integer loop;  
  initial  
  begin  
    for (loop=0; loop<8; loop=loop+1)  
    begin  
      {dS,dA,dB} = loop;  
      #10;  
    end  
  end  
  mux21_1b dut (dS,dA,dB,mY);  
endmodule
```

4-to-1 (@4-1)

Multiplexer



Truth table

⇒ 'simplified'

S1	S0	Y
0	0	A
0	1	B
1	0	C
1	1	D

Notice patterns?

⇒ mux21_1b?

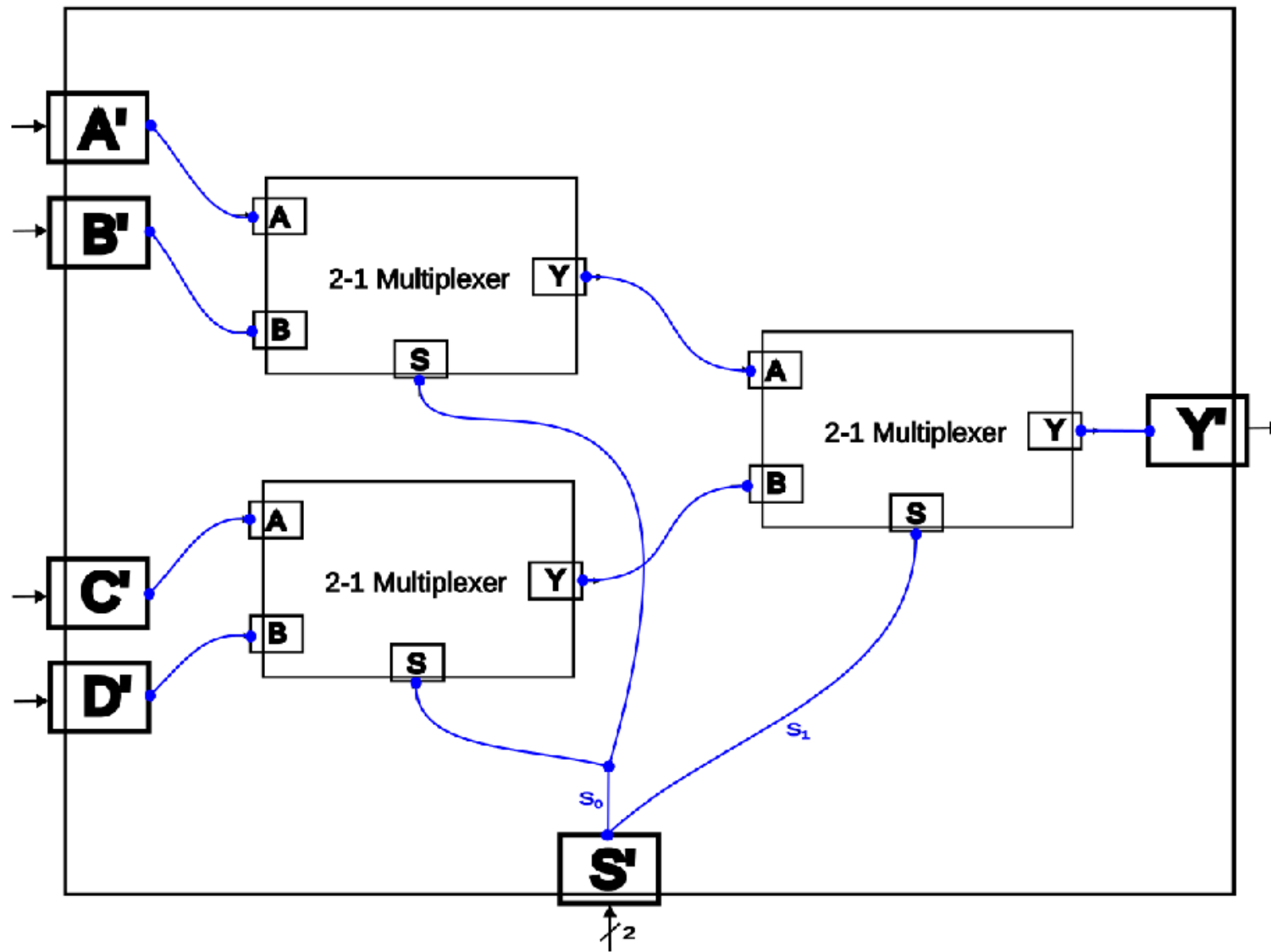
Instantiate

`mux21_1b`

in `mux41_1b`!

Schematic!

4-1 Multiplexer



Truth table

⇒ 'full'

We can, but

We would not!

Build equation!

⇒ 'look' for mux21

Note is...

⇒ 2-bits!

Bus @ Verilog arrays!

Bus size as
Indices [begin:end]

```
wire[1:0] is;
```

Describe

mux41_1b

```
module mux41_1b (iS, iA, iB, iC, iD, oY);  
  input[1:0] iS;  
  input iA, iB, iC, iD;  
  output oY;  
  wire oY, tP, tQ;  
  mux21_1b s00 (iS[0], iA, iB, tP);  
  mux21_1b s01 (iS[0], iC, iD, tQ);  
  mux21_1b s10 (iS[1], tP, tQ, oY);  
endmodule
```

- Make sure mux21_1b is in same project!

Testbench:

► mux41_1b_tb!

⇒ loop count?

Describe

mux41_1b_tb

```

module mux41_1b_tb ();
reg[1:0] dS;
reg dA,dB,dC,dD;
wire mY;
integer loop;
initial
begin
    for (loop=0;loop<64;loop=loop+1)
    begin
        {dS,dA,dB,dC,dD} = loop;
        #10;
    end
end
mux41_1b dut (dS,dA,dB,dC,dD,mY);
endmodule

```

Homework:

► 2-bits 4-1 mux!

⇒ mux41_2b

⇒ mux41_2b_tb