NMK20603

Computer Architecture

MY1

Describing Combinational Logic (cont.)

Codes so far:

- ✔ mux21_1b, mux21_1b_tb
- ✔ mux41_1b, mux41_1b_tb
- ✔ mux21_2b, mux21_2b_tb
- ✔ mux41_2b, mux41_2b_tb
- ✔ dmux21, dmux21_tb (only this matters)

MY1

Homework?

⇒ dmux41

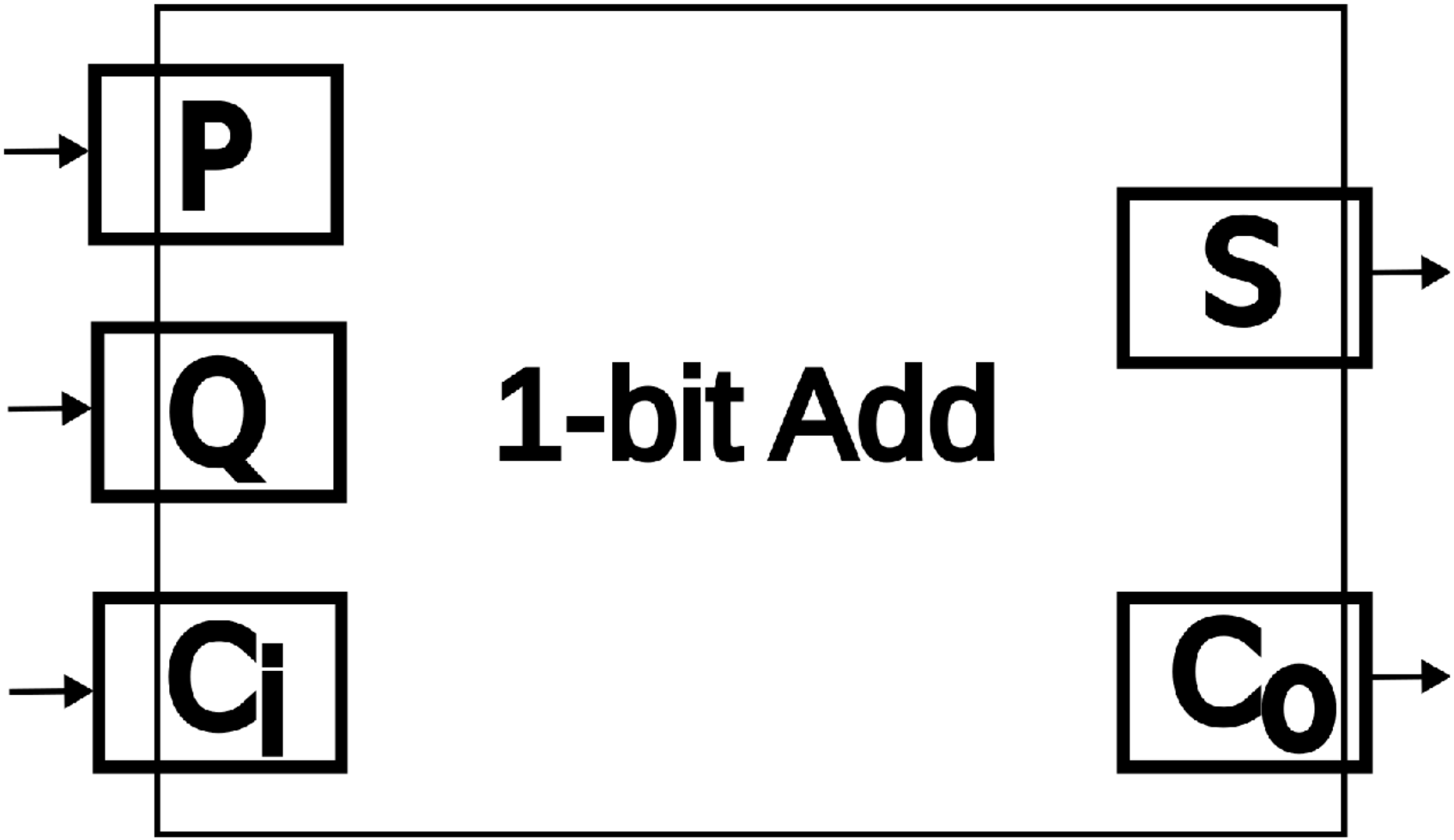
⇒ dmux41_tb

MY1

Let's do arithmetic!

⇒ 1-bit Add

MY1



MY1

iC	iP	iQ	oC	oS
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Get SOP equation!

Describe:

add_1b

```
module add_1b (iC,iP,iQ,oC,oS);
input iC,iP,iQ;
output oC, oS;
wire oC, oS, tP, tG, tP1, tP2, tS1, tS2, tX;
// just to show order DOES NOT matter
assign oS = tS1 | tS2; // iC ^ tP;
assign oC = tG | tX;
assign tG = iP & iQ;
assign tP1 = iP & ~iQ;
assign tP2 = ~iP & iQ;
assign tP = tP1 | tP2; // iP ^ iQ;
assign tS1 = iC & ~tP;
assign tS2 = ~iC & tP;
assign tX = iC & tP;
endmodule
```

MY1

Describe:

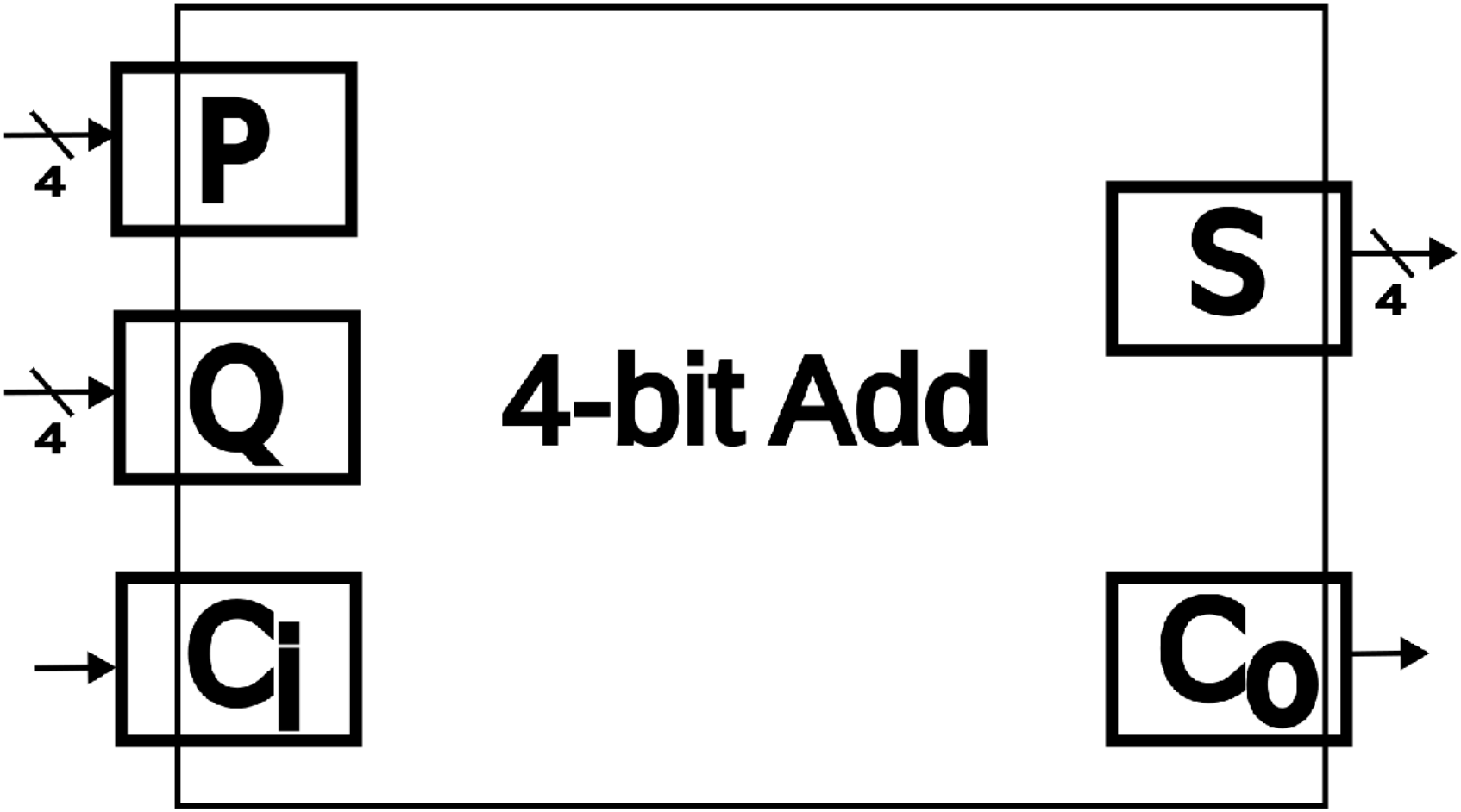
add_1b_tb

MY1

```
module add_1b_tb ();
reg dC,dP,dQ;
wire mC,mS;
reg cC,cS;
integer loop,ecnt;
initial begin
  ecnt = 0;
  for (loop=0;loop<8;loop=loop+1) begin
    {dC,dP,dQ} = loop;
    #5;
    {cC,cS} = dC+dP+dQ; // or, do a truth table!
    if (mC!==cC||mS!==cS) begin
      ecnt = ecnt + 1;
    end
    #5;
  end
  if (ecnt==0) begin
    $display("-- Module add_1b verified!");
  end
  else begin
    $display("** Module add_1b with error(s)! (%d)",ecnt);
  end
  $stop;
end
add_1b dut (dC,dP,dQ,mC,mS);
endmodule
```

4-bit Add

MY1



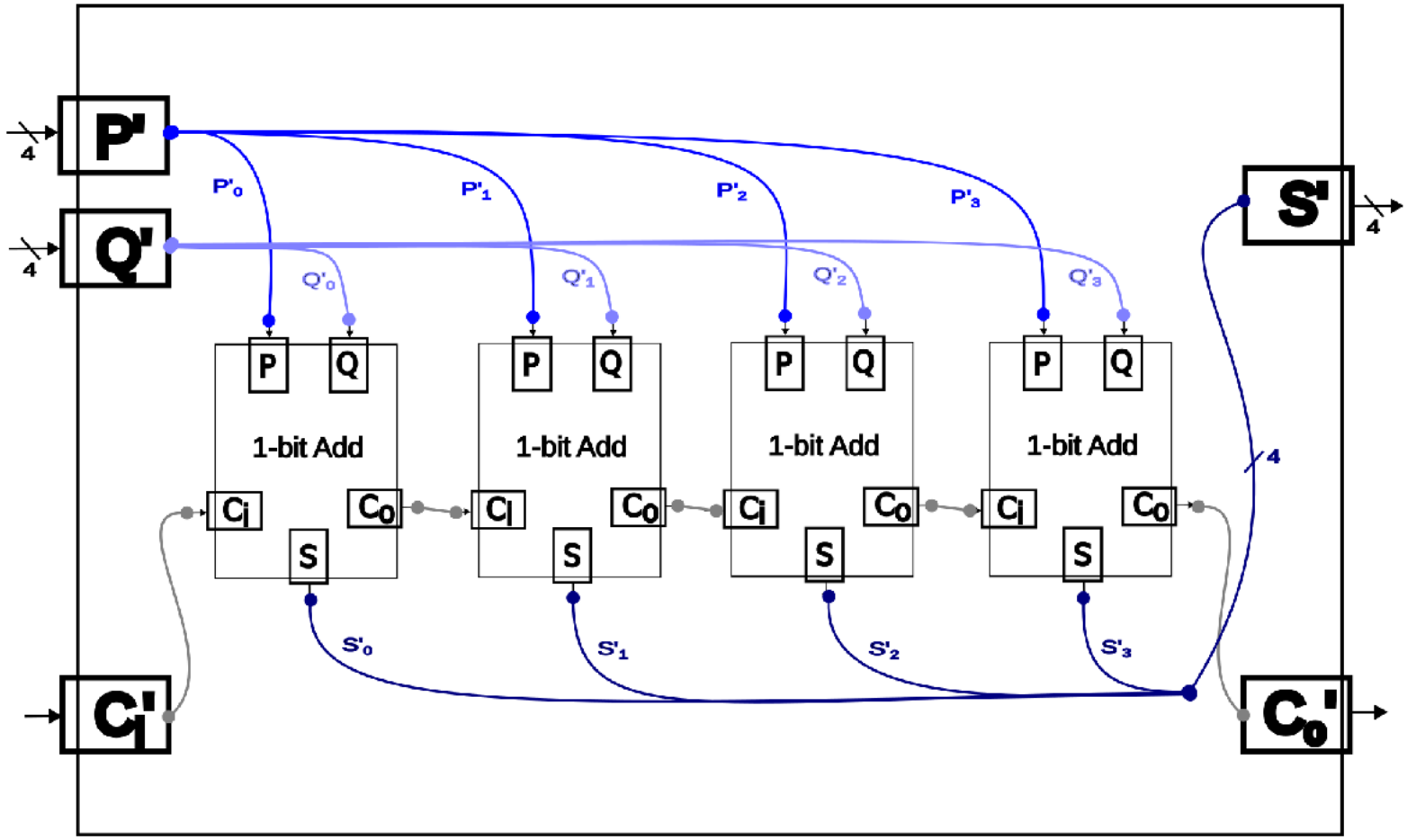
Equation?

⇒ Check on-paper

4-digit Addition!

MY1

4-bit Add



MY1

Ripple-Carry Adder!

MY1

Describe:

add_4b

MY1

```
module add_4b (iC,iP,iQ,oC,oS);
input iC;
input[3:0] iP,iQ;
output oC;
output[3:0] oS;
wire oC;
wire[3:0] oS;
wire[2:0] tC;
add_1b bit0 (iC,iP[0],iQ[0],tC[0],oS[0]);
add_1b bit1 (tC[0],iP[1],iQ[1],tC[1],oS[1]);
add_1b bit2 (tC[1],iP[2],iQ[2],tC[2],oS[2]);
add_1b bit3 (tC[2],iP[3],iQ[3],oC,oS[3]);
endmodule
```

MY1

Extra:

⇒ CLA?

MY1

Describe:

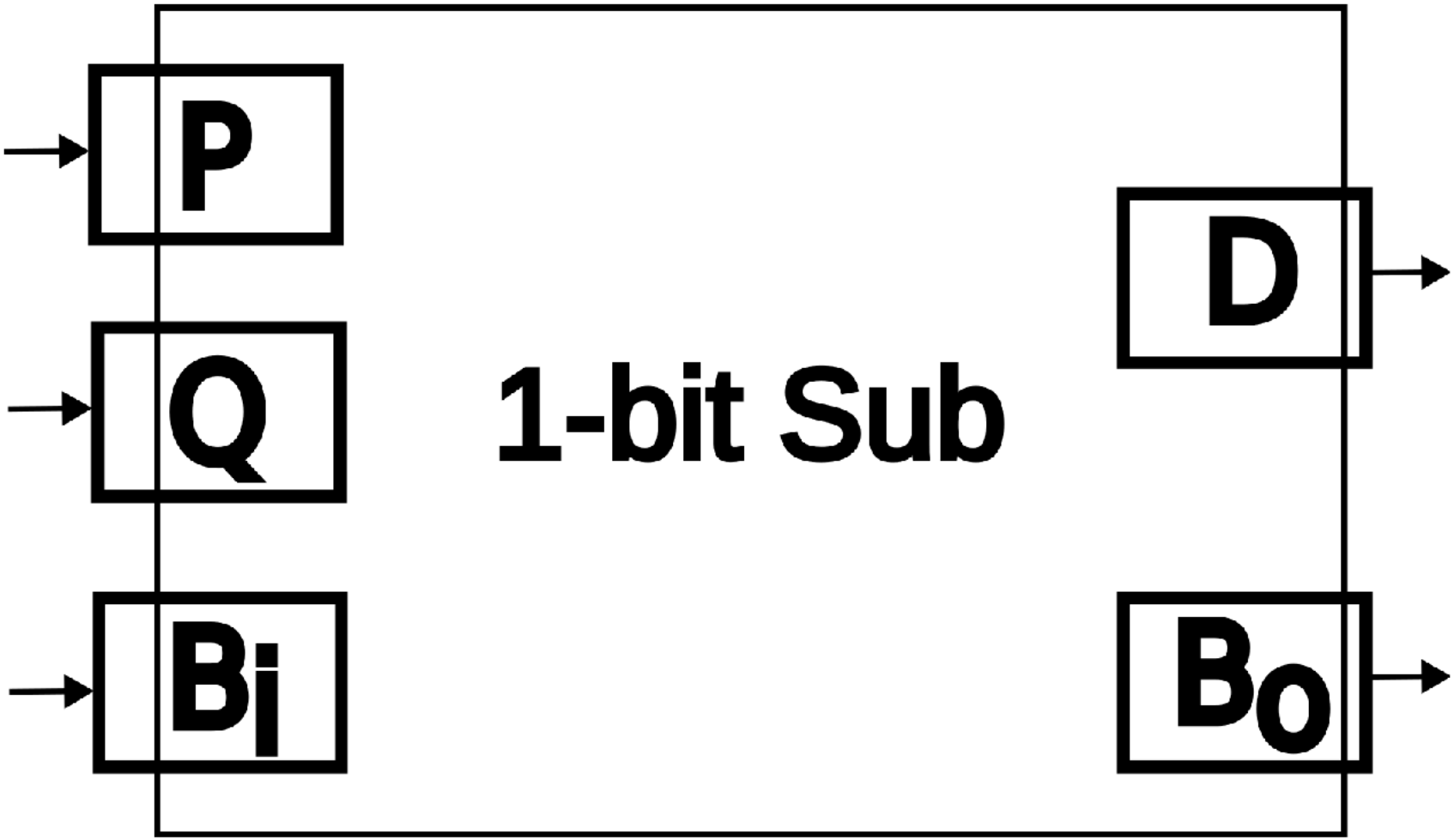
add_4b_tb

MY1

```
module add_4b_tb ();
reg dC;
reg[3:0] dP,dQ;
wire mC;
wire[3:0] mS;
reg cC;
reg[3:0] cS;
integer loop,ecnt;
initial begin
  ecnt=0;
  for (loop=0;loop<512;loop=loop+1) begin
    {dC,dP,dQ} = loop;
    #5;
    {cC,cS} = dP + dQ + dC;
    if (mC!==cC||mS!==cS) begin
      ecnt = ecnt + 1;
    end
    #5;
  end
  if (ecnt==0) begin
    $display("-- Module add_4b verified!");
  end
  else begin
    $display("** Module add_4b with error(s)! (%d)",ecnt);
  end
  $stop;
end
add_4b dut (dC,dP,dQ,mC,mS);
endmodule
```

1-bit Subtract

MY1



MY1

iB	iP	iQ	oB	oD
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

Get SOP equation!

Homework:

⇒ sub_1b, sub_1b_tb

⇒ sub_4b, sub_4b_tb

MY1