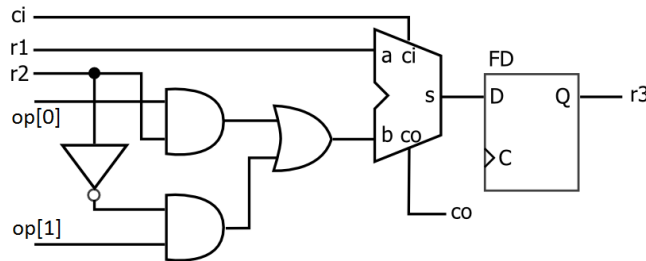


4. vaja: Računska enota z registrom

4.1 Registrska celica



Naredi registrsko celico za izračun osnovnih aritmetičnih operacij nad enobitnim vhodnim podatkom. Dvobitni signal op določa vrsto operacije:

op	operacija	
00	$r3 = r1 + ci$	prenos r1 ali r1+1
01	$r3 = r1 + r2 + ci$	vsota r1+r2 s prenosom
10	$r3 = r1 + (\text{not } r2) + ci$	razlika r1-r2 pri ci=1
11	$r3 = r1 - 1 + ci$	prenos r1 ali r1-1

Uporabi operatorja seštevanja in združevanja za opis polnega seštevalnika:

```
assign {co, s} = a + b + ci;
```

Podatkovni flip-flop opišemo s proceduralnim blokom, ki se proži npr. ob pozitivnem robu ure oz. naraščajoči fronti:

```
always @(posedge clk) begin
```

Napiši testno strukturo z generatorjem ure za simulacijo registrske celice:

```
always begin
  clk = 0; #10;
  clk = 1; #10;
end

initial begin
  {op, r2, r1, ci} = 5'b 00000; #11;
  {op, r2, r1, ci} = 5'b 00010; #20;
  {op, r2, r1, ci} = 5'b 01110; #20;
  {op, r2, r1, ci} = 5'b 01111; #20;
  {op, r2, r1, ci} = 5'b 10011;
  ...
end
```

4.2 Računska enota

Poveži štiri registrske celice v 4-bitno računsko enoto z izhodnim registrom. Delovanje vezja preizkusi s testno strukturo v kateri nastavi uro in testne vektorje za vhode r1, r2 in ci. Določi vhodne vrednosti, pri katerih bi preizkusil delovanje vezja za vsako izmed štirih operacij, npr. za preizkus operacije 01 (vsota) uporabi:

r1	r2	ci	r3	co
0000	1111	0		
0001	1111	0		
0010	1110	0		
0100	1100	0		
1000	1000	0		
1111	0000	1		
1110	0000	1		
1101	0001	1		
1011	0011	1		
0111	0111	1		