

## Lösningsförslag övningsuppgifter 2023-03-30

1. Genomför följande aritmetiska operationer på 4-bitars binär form:
- a)  $6 + 3$
  - b)  $6 - 3$
  - c)  $3 * 3$
  - d)  $9 / 3$

**Lösningsförslag:**

Binära operationer kan genomföras på samma sätt som för decimala tal. I detta fall används liggande stolen för division.

1. a)  $6 + 3 = 9 \Rightarrow$

	0	1	1	0
+	0	0	1	1
	1	0	0	1

Figur 1: Binär addition.

1. b)  $6 - 3 = 6 + (2^4 - 3) = 6 + 13 = 19$   
 $\Rightarrow 19 = 3 \text{ mod carry} \Rightarrow$

1	1	0	1	0
1	1	1	0	1
0	0	1	1	

Figur 2: Binär subtraktion med 4-bitars 2-komplement.

1. c)  $3 \cdot 3 = 9 \Rightarrow$

$\begin{array}{r} 0011 \\ \cdot 0011 \\ \hline 0011 \\ 0011 \\ \hline 1001 \end{array}$

$+ 0011$

Figur 3: Binär multiplikation.

1. d)  $\frac{9}{3} = 3 \Rightarrow$

$9 \rightarrow$

$$\begin{array}{r} 0011 \\ \underline{1001} \phantom{11} \\ 11 \\ \underline{11} \\ 0 \end{array}$$

$r \rightarrow$

Figur 4: Binär division med liggande stolen.

Division med liggande stolen sker från vänster till höger, där kvot  $q$  samt rest  $r$  beräknas:

- $l < ll \Rightarrow q_3 = 0$ , lägg till biten till höger  $\Rightarrow$  division med 10 i stället för 1
- $10 < 11 \Rightarrow q_2 = 0$ , lägg till biten till höger  $\Rightarrow$  nästa gång division med 100 i stället för 10
- $100 / 11 = 4 / 3 \Rightarrow q_1 = 1, r_1 = 1 \Rightarrow$  Placera  $r = 1$  nedanför 100, lägg till nästa bit 1  $\Rightarrow$  nästa gång division med 11
- $11 / 11 = 3 / 3 \Rightarrow q_0 = q, r_0 = 0 \Rightarrow$  Inga fler bitar till höger, beräkningen är klar

Därmed gäller att kvoten  $q = 0011 \Rightarrow 9/3 = 1001 / 0011 = 0011$

## Hårdvarunära programmering

2. Ett grindnät, där tre slide-switchar  $\text{switch}[2:0]$  används som insignaler och två lysdioder  $\text{led}[1:0]$  används som utsignaler, ska realiseras för hand samt i VHDL. Lysdioderna ska visa antalet höga slide-switchar – 1 binärt. Sambandet mellan antalet höga slide-switchar samt utsignalerna på lysdioder  $\text{led}[1:0]$  visas nedan:

Antal höga slide-switchar	led[1:0]
0	11
1	00
2	01
3	10

Tabell 1: Sambandet mellan antalet höga insignaler samt utsignalerna för grindnätet.

- Rita en sanningstabell för insignaler  $\text{switch}[2:0] = ABC$  samt utsignaler  $\text{led}[1:0] = XY$ .
- Ur sanningstabellen, ta fram minimerade ekvationer för utsignaler XY och realisera grindnätet.
- Skapa en testbänk döpt *vhdl\_exercise1\_tb* för simulering av modulen *vhdl\_exercise1* i ModelSim. Testa varje kombination av insignaler  $\text{switch}[2:0]$  under 10 ns var. Den totala simuleringstiden ska därmed uppgå till 80 ns.
- Instansiera modulen *vhdl\_exercise1* och kontrollera att modulen fungerar som tänkt.
- Realisera konstruktionen i VHDL via en modul döpt *vhdl\_exercise1*:
  - Välj FPGA-kort Terasic DE0 (enhet 5CEBA4F23C7)
  - Koppla  $\text{switch}[2:0]$  till T13, V13 och U13
  - Koppla  $\text{led}[1:0]$  till AA1 samt AA2

Se till att koden är kompilerad och in- och utportarna har anslutits enligt ovan så att validering enkelt kan genomföras genom att programmera FPGA-kortet.

### Lösningsförslag:

- a) Genom att sätta att utsignaler XY är lika med antalet insignaler ABC – 1:

$$XY = ABC - 1$$

erhålls följande sanningstabell:

A	B	C	X	Y
0	0	0	1	1
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Tabell 2: Sanningstabell för grindnätet i uppgift 2.

- b) Av ovanstående tabell kan ekvationer för utportar X och Y härledas.

$X = 1$  då  $ABC = 000$  eller  $ABC = 111$ . Således kan följande ekvation härledas:

$$X = A'B'C' + ABC$$

X kan därmed realiseras via en AND-grind med  $A'$ ,  $B'$  och  $C'$  som insignaler, en AND-grind med ABC som insignaler samt en OR-grind med  $A'B'C'$  samt ABC som insignaler.

$Y = 1$  då  $ABC = 000$ ,  $ABC = 011$ ,  $ABC = 101$  eller  $ABC = 110$ . Således kan följande ekvation härledas:

$$Y = A'B'C' + A'BC + AB'C + ABC'$$

vilket kan förenklas till

$$Y = A'(B'C' + BC) + A(B'C + BC')$$

Eftersom  $B \wedge C = B'C + BC'$  samt  $(B \wedge C)' = B'C' + BC$  kan ekvationen förenklas till

$$B \wedge C = B'C + BC'$$

samt

$$(B \wedge C)' = (B'C' + BC)$$

Ersätt  $B \wedge C$  med  $D$ :

$$D = B \wedge C$$

så ser vi att

$$Y = A'D' + AD,$$

vilket är ekvivalent med

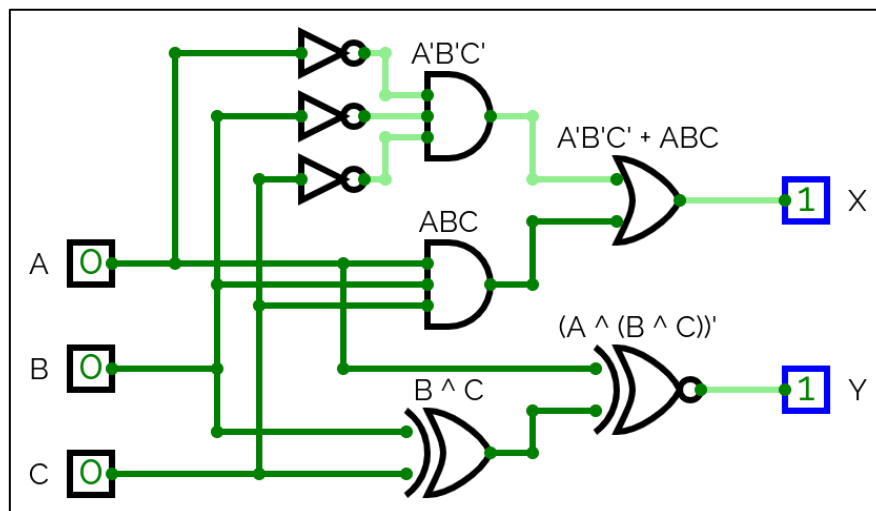
$$Y = (A \wedge D)'$$

Genom att ersätta  $D$  med  $B \wedge C$  erhålls följande ekvation:

$$Y = (A \wedge (B \wedge C))'$$

$Y$  kan således realiseras via en XOR-grind med  $B$  och  $C$  som insignaler samt en XNOR-grind med  $A$  och  $(B \wedge C)$  som insignaler.

Via härledda ekvationer för utportar  $X$  och  $Y$  kan grindnätet realiseras i enlighet med nedanstående figur:



Figur 3: Grindnät för konstruktionen i uppgift 2.

- c) Se filen `vhdl_exercise1.vhd`.
- d) Se filen `vhdl_exercise1_tb.vhd`.
- e) Ladda ned .qar-filen `vhdl_exercise1.qar`, kompiler och programmera FPGA-kortet Terasic DE0.  
**Notering:** Filen `vhdl_exercise1_alternative_design.qar` demonstrerar en alternativ konstruktion där konstruktionen implementeras via beskrivande kod i stället för logiska ekvationer.