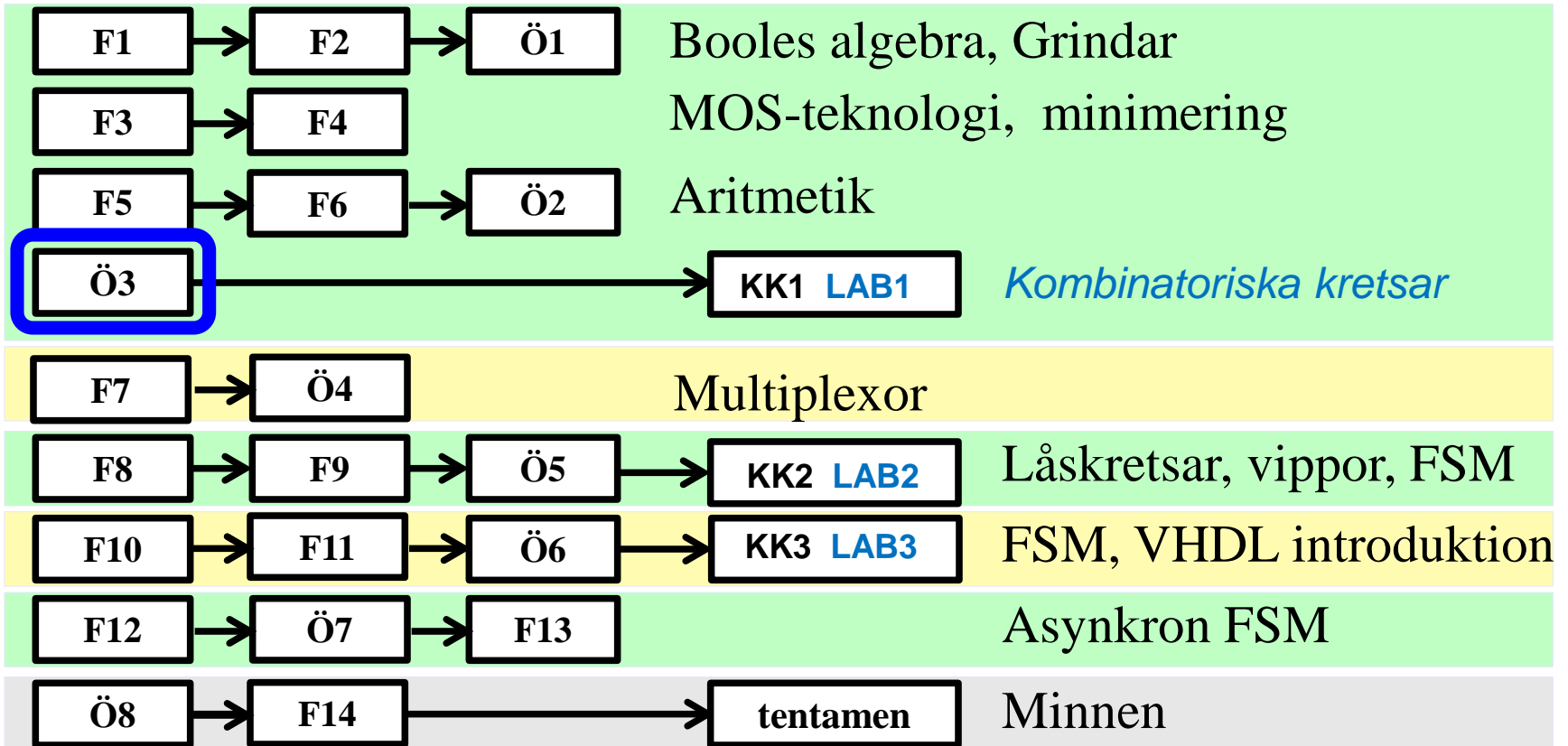


IE1204 Digital Design



*Föreläsningar och övningar bygger på varandra! Ta alltid igen det Du missat!
Läs på i förväg – delta i undervisningen – arbeta igenom materialet efteråt!*

ÖH 8.4 7-4-2-1 kod

Kodomvandlare 7-4-2-1-kod till BCD-kod.

Vid kodning av siffrorna 0...9 användes förr ibland en kod med vikterna 7-4-2-1 i stället för den binära kodens vikter 8-4-2-1.

I de fall då en siffras kodord kan väljas på olika sätt väljs det kodord som innehåller minst antal ettor.

(en variant av 7-4-2-1 koden används i dag till butikernas streck-kod)



	7	4	2	1		8	4	2	1
	x_7	x_4	x_2	x_1		y_8	y_4	y_2	y_1
(0)	0	0	0	0	0	0	0	0	0
(1)	0	0	0	1	1	0	0	0	1
(2)	0	0	1	0	2	0	0	1	0
(3)	0	0	1	1	3	0	0	1	1
(4)	0	1	0	0	4	0	1	0	0
(5)	0	1	0	1	5	0	1	0	1
(6)	0	1	1	0	6	0	1	1	0
					7	0	1	1	1
					8	1	0	0	0
					9	1	0	0	1

Ex 8.4 7-4-2-1 kod

Kodomvandlare 7-4-2-1-kod till BCD-kod.

Vid kodning av siffrorna 0...9 användes förr ibland en kod med vikterna 7-4-2-1 i stället för den binära kodens vikter 8-4-2-1.

I de fall då en siffras kodord kan väljas på olika sätt väljs det kodord som innehåller minst antal ettor.

(en variant av 7-4-2-1 koden används i dag till butikernas streck-kod)



	7	4	2	1		8	4	2	1
	x_7	x_4	x_2	x_1		y_8	y_4	y_2	y_1
(0)	0	0	0	0	0	0	0	0	0
(1)	0	0	0	1	1	0	0	0	1
(2)	0	0	1	0	2	0	0	1	0
(3)	0	0	1	1	3	0	0	1	1
(4)	0	1	0	0	4	0	1	0	0
(5)	0	1	0	1	5	0	1	0	1
(6)	0	1	1	0	6	0	1	1	0
(8)	1	0	0	0	7	0	1	1	1
(9)	1	0	0	1	8	1	0	0	0
					9	1	0	0	1

Ex 8.4 7-4-2-1 kod

Kodomvandlare 7-4-2-1-kod till BCD-kod.

Vid kodning av siffrorna 0...9 användes förr ibland en kod med vikterna 7-4-2-1 i stället för den binära kodens vikter 8-4-2-1.

I de fall då en siffras kodord kan väljas på olika sätt väljs det kodord som innehåller minst antal ettor.

(en variant av 7-4-2-1 koden används i dag till butikernas streck-kod)



	7	4	2	1		8	4	2	1
	x_7	x_4	x_2	x_1		y_8	y_4	y_2	y_1
(0)	0	0	0	0	0	0	0	0	0
(1)	0	0	0	1	1	0	0	0	1
(2)	0	0	1	0	2	0	0	1	0
(3)	0	0	1	1	3	0	0	1	1
(4)	0	1	0	0	4	0	1	0	0
(5)	0	1	0	1	5	0	1	0	1
(6)	0	1	1	0	6	0	1	1	0
(8)	1	0	0	0	7	0	1	1	1
(9)	1	0	0	1	8	1	0	0	0
(10)	1	0	1	0	9	1	0	0	1

8.4

	7	4	2	1		8	4	2	1
	x_7	x_4	x_2	x_1		y_8	y_4	y_2	y_1
(0)	0	0	0	0	0	0	0	0	0
(1)	0	0	0	1	1	0	0	0	1
(2)	0	0	1	0	2	0	0	1	0
(3)	0	0	1	1	3	0	0	1	1
(4)	0	1	0	0	4	0	1	0	0
(5)	0	1	0	1	5	0	1	0	1
(6)	0	1	1	0	6	0	1	1	0
(8)	1	0	0	0	7	0	1	1	1
(9)	1	0	0	1	8	1	0	0	0
(10)	1	0	1	0	9	1	0	0	1

y_8

x_7	x_4	x_2	x_1	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

y_4

x_7	x_4	x_2	x_1	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

y_2

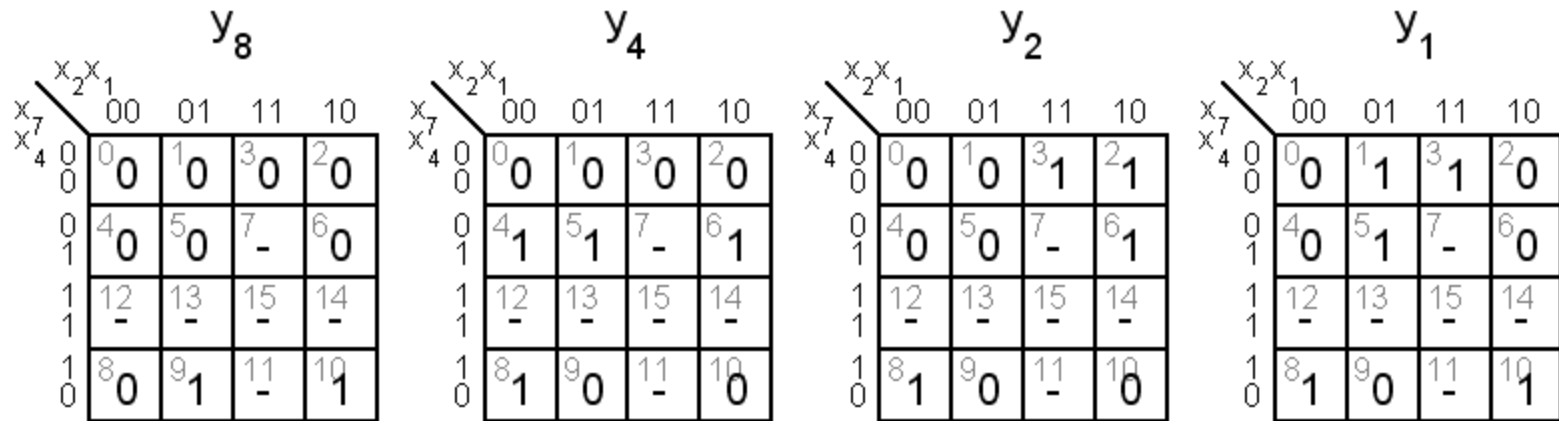
x_7	x_4	x_2	x_1	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

y_1

x_7	x_4	x_2	x_1	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

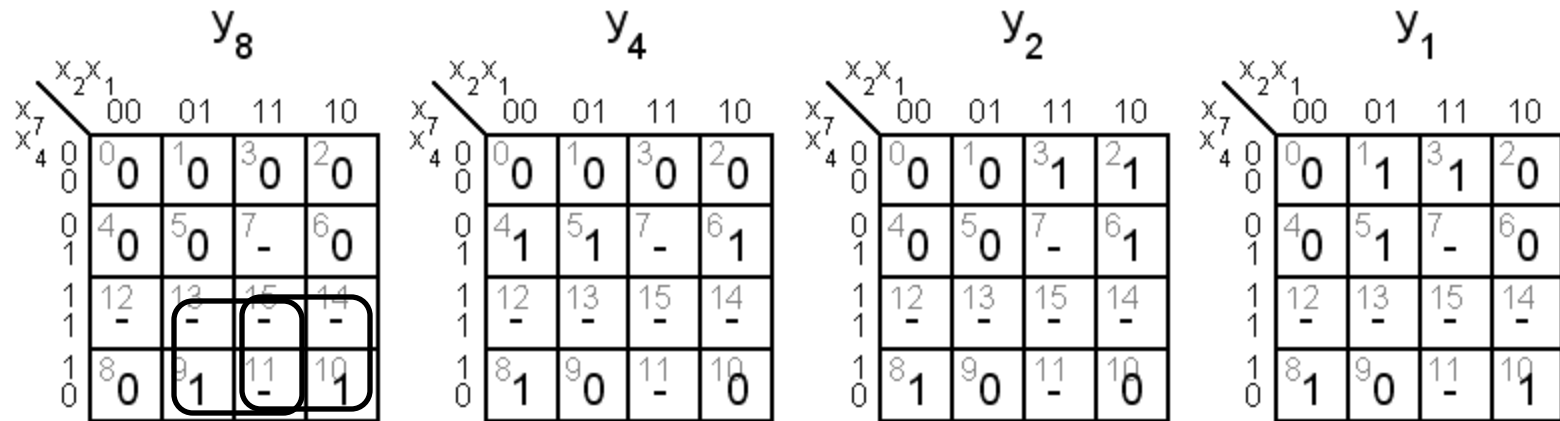
8.4

	7	4	2	1		8	4	2	1
	x_7	x_4	x_2	x_1		y_8	y_4	y_2	y_1
(0)	0	0	0	0	0	0	0	0	0
(1)	0	0	0	1	1	0	0	0	1
(2)	0	0	1	0	2	0	0	1	0
(3)	0	0	1	1	3	0	0	1	1
(4)	0	1	0	0	4	0	1	0	0
(5)	0	1	0	1	5	0	1	0	1
(6)	0	1	1	0	6	0	1	1	0
(8)	1	0	0	0	7	0	1	1	1
(9)	1	0	0	1	8	1	0	0	0
(10)	1	0	1	0	9	1	0	0	1



8.4

	7	4	2	1		8	4	2	1
	x_7	x_4	x_2	x_1		y_8	y_4	y_2	y_1
(0)	0	0	0	0		0	0	0	0
(1)	0	0	0	1		1	0	0	1
(2)	0	0	1	0		2	0	0	1
(3)	0	0	1	1		3	0	0	1
(4)	0	1	0	0		4	0	1	0
(5)	0	1	0	1		5	0	1	0
(6)	0	1	1	0		6	0	1	1
(8)	1	0	0	0		7	0	1	1
(9)	1	0	0	1		8	1	0	0
(10)	1	0	1	0		9	1	0	1



$$y_8 = x_7 x_2 + x_7 x_1$$

8.4

	7	4	2	1		8	4	2	1
	x_7	x_4	x_2	x_1		y_8	y_4	y_2	y_1
(0)	0	0	0	0	0	0	0	0	0
(1)	0	0	0	1	1	0	0	0	1
(2)	0	0	1	0	2	0	0	1	0
(3)	0	0	1	1	3	0	0	1	1
(4)	0	1	0	0	4	0	1	0	0
(5)	0	1	0	1	5	0	1	0	1
(6)	0	1	1	0	6	0	1	1	0
(8)	1	0	0	0	7	0	1	1	1
(9)	1	0	0	1	8	1	0	0	0
(10)	1	0	1	0	9	1	0	0	1

y_8

x_7	x_4	x_2	x_1	
0	0	01	11	10
0	0	0	0	0
0	0	1	0	0
1	4	5	7	6
1	0	0	-	0
1	12	13	15	14
1	-	-	-	-
1	8	9	11	10
0	0	1	-	1

y_4

x_7	x_4	x_2	x_1	
0	0	01	11	10
0	0	0	0	0
0	0	1	0	0
1	4	5	7	6
1	1	1	-	1
1	12	13	15	14
1	-	-	-	-
1	8	9	11	10
0	1	0	-	0

y_2

x_7	x_4	x_2	x_1	
0	0	01	11	10
0	0	0	0	0
0	0	1	0	0
1	4	5	7	6
1	0	0	-	1
1	12	13	15	14
1	-	-	-	-
1	8	9	11	10
0	1	0	-	0

y_1

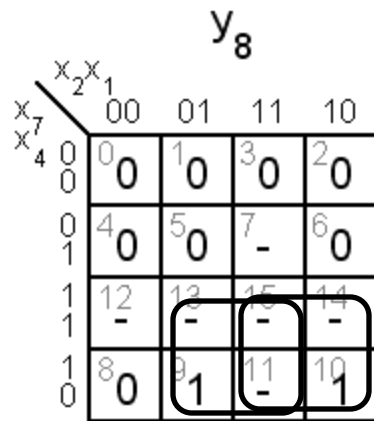
x_7	x_4	x_2	x_1	
0	0	01	11	10
0	0	0	0	0
0	0	1	0	0
1	4	5	7	6
1	0	1	-	0
1	12	13	15	14
1	-	-	-	-
1	8	9	11	10
0	1	0	-	1

$$y_8 = x_7 x_2 + x_7 x_1$$

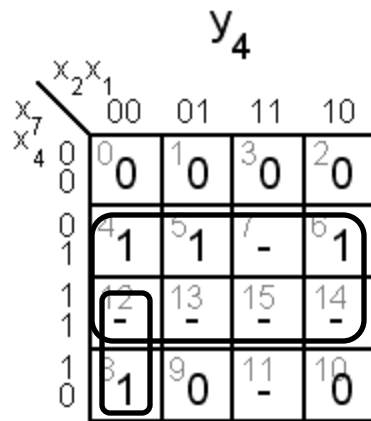
$$y_4 = x_4 + x_7 x_2 x_1$$

8.4

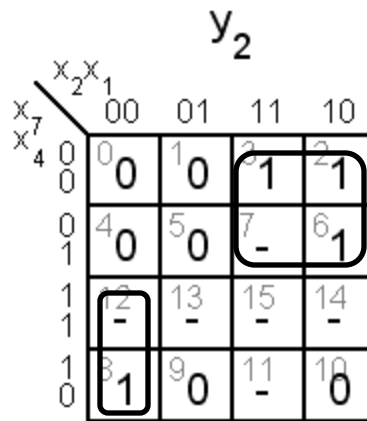
	7	4	2	1		8	4	2	1
	x_7	x_4	x_2	x_1		y_8	y_4	y_2	y_1
(0)	0	0	0	0	0	0	0	0	0
(1)	0	0	0	1	1	0	0	0	1
(2)	0	0	1	0	2	0	0	1	0
(3)	0	0	1	1	3	0	0	1	1
(4)	0	1	0	0	4	0	1	0	0
(5)	0	1	0	1	5	0	1	0	1
(6)	0	1	1	0	6	0	1	1	0
(8)	1	0	0	0	7	0	1	1	1
(9)	1	0	0	1	8	1	0	0	0
(10)	1	0	1	0	9	1	0	0	1



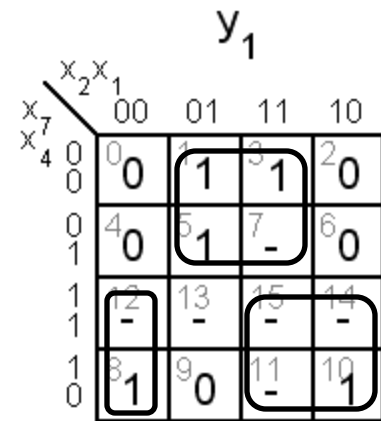
$$y_8 = x_7 x_2 + x_7 x_1$$



$$y_4 = x_4 + x_7 x_2 x_1$$



$$y_2 = x_7 x_2 + x_7 x_2 x_1$$



$$y_1 = x_7 x_1 + x_7 x_2 + x_7 x_2 x_1$$

8.4

Gemensamma hoptagningar kan ge delade grindar!

	7	4	2	1		8	4	2	1
	x_7	x_4	x_2	x_1		y_8	y_4	y_2	y_1
(0)	0	0	0	0	0	0	0	0	0
(1)	0	0	0	1	1	0	0	0	1
(2)	0	0	1	0	2	0	0	1	0
(3)	0	0	1	1	3	0	0	1	1
(4)	0	1	0	0	4	0	1	0	0
(5)	0	1	0	1	5	0	1	0	1
(6)	0	1	1	0	6	0	1	1	0
(8)	1	0	0	0	7	0	1	1	1
(9)	1	0	0	1	8	1	0	0	0
(10)	1	0	1	0	9	1	0	0	1

y_8

x_7	x_4	x_2	x_1	
0	0	0	0	00
0	0	1	0	01
0	1	0	0	11
0	1	0	1	10
1	0	0	0	4
1	0	0	1	5
1	0	1	0	7
1	0	1	1	6
1	1	0	0	12
1	1	0	1	13
1	1	1	0	15
1	1	1	1	14
1	0	0	0	8
1	0	0	1	9
1	0	1	0	11
1	0	1	1	10

$$y_8 = x_7 x_2 + x_7 x_1$$

y_4

x_7	x_4	x_2	x_1	
0	0	0	0	00
0	0	1	0	01
0	1	0	0	11
0	1	0	1	10
1	0	0	0	4
1	0	0	1	5
1	0	1	0	7
1	0	1	1	6
1	1	0	0	12
1	1	0	1	13
1	1	1	0	15
1	1	1	1	14
1	0	0	0	8
1	0	0	1	9
1	0	1	0	11
1	0	1	1	10

$$y_4 = x_4 + x_7 x_2 x_1$$

y_2

x_7	x_4	x_2	x_1	
0	0	0	0	00
0	0	1	0	01
0	1	0	0	11
0	1	0	1	10
1	0	0	0	4
1	0	0	1	5
1	0	1	0	7
1	0	1	1	6
1	1	0	0	12
1	1	0	1	13
1	1	1	0	15
1	1	1	1	14
1	0	0	0	8
1	0	0	1	9
1	0	1	0	11
1	0	1	1	10

$$y_2 = x_7 x_2 + x_7 x_2 x_1$$

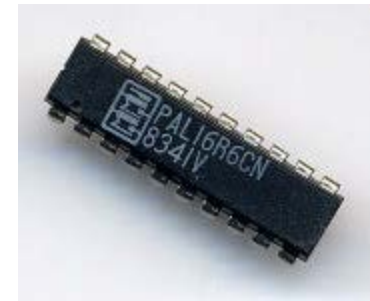
y_1

x_7	x_4	x_2	x_1	
0	0	0	0	00
0	0	1	0	01
0	1	0	0	11
0	1	0	1	10
1	0	0	0	4
1	0	0	1	5
1	0	1	0	7
1	0	1	1	6
1	1	0	0	12
1	1	0	1	13
1	1	1	0	15
1	1	1	1	14
1	0	0	0	8
1	0	0	1	9
1	0	1	0	11
1	0	1	1	10

$$y_1 = x_7 x_1 + x_7 x_2 + x_7 x_2 x_1$$

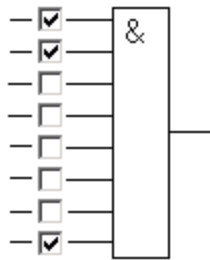
8.4

PLA-kretsar innehåller programmerbara AND och OR grindar. (Detta visade sig vara onödigt komplext, så det vanliga blev PAL-kretsar med endast AND-nätet programmerbart).

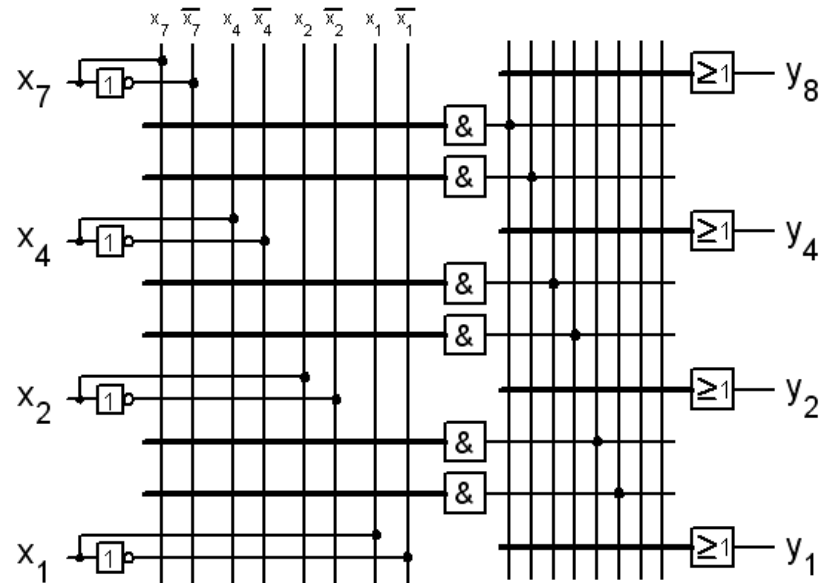


Grindarna har många programmerbara ingångsanslutningar. De många ingångarna ritas därför oftast med ett "förenklat" ritsätt.

Programmerbar logik



förenklat ritsätt för 8 ingångars grind



8.4

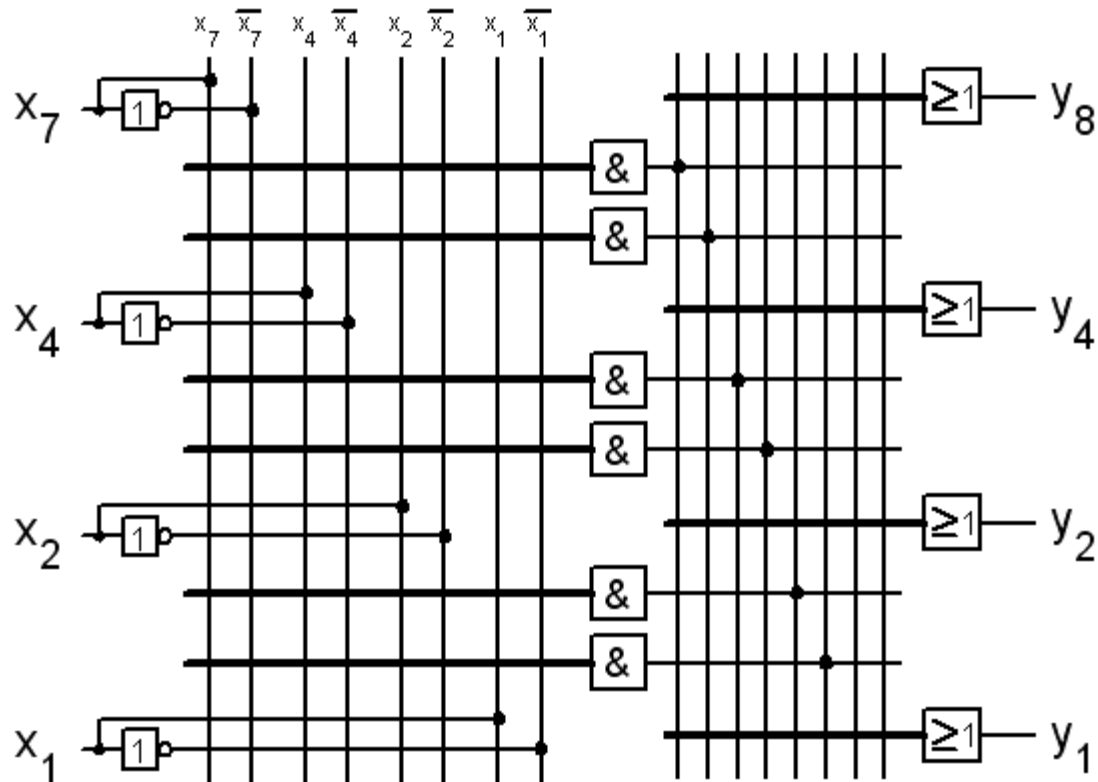
$$y_8 = x_7 x_2 + x_7 x_1$$

$$y_4 = x_4 + x_7 x_2 x_1$$

$$y_2 = \overline{x_7} x_2 + x_7 \overline{x_2} x_1$$

$$y_1 = \overline{x_7} x_1 + x_7 x_2 + x_7 \overline{x_2} x_1$$

Grind-delning!



8.4

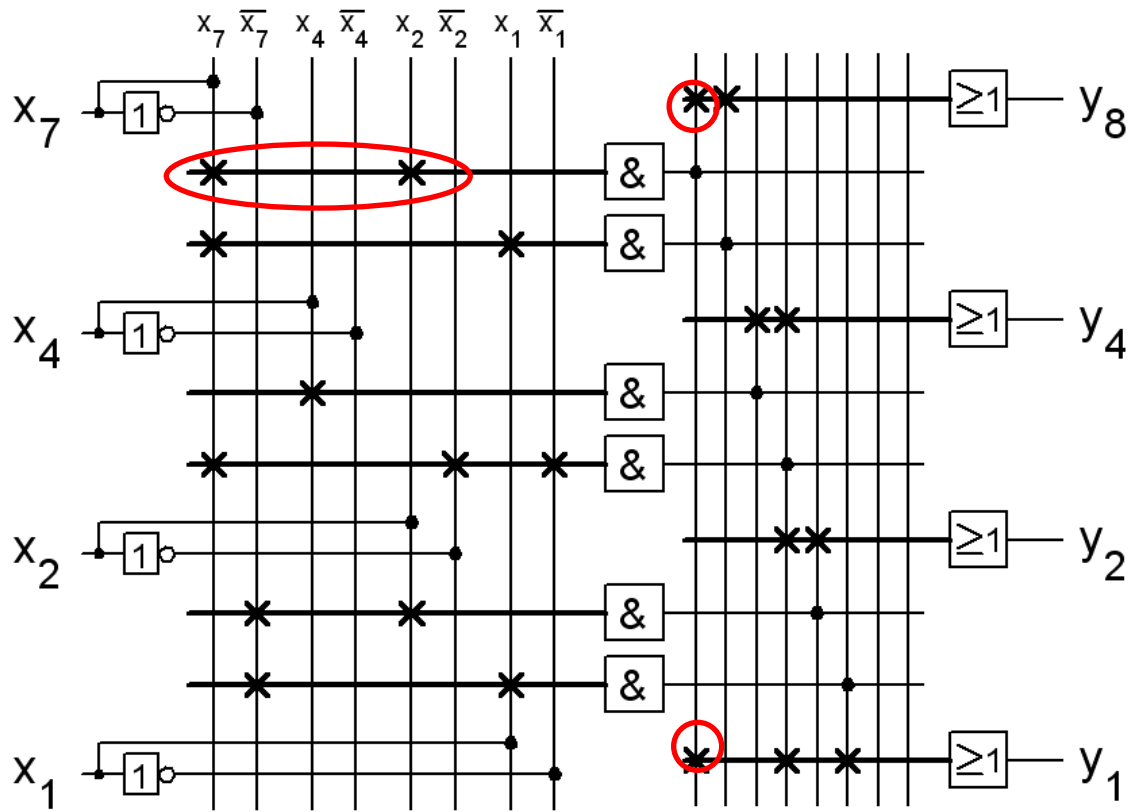
$$y_8 = x_7 x_2 + x_7 x_1$$

$$y_4 = x_4 + x_7 \overline{x_2} x_1$$

$$y_2 = \overline{x_7} x_2 + x_7 \overline{x_2} x_1$$

$$y_1 = \overline{x_7} x_1 + x_7 x_2 + x_7 \overline{x_2} x_1$$

Grind-delning!



8.4

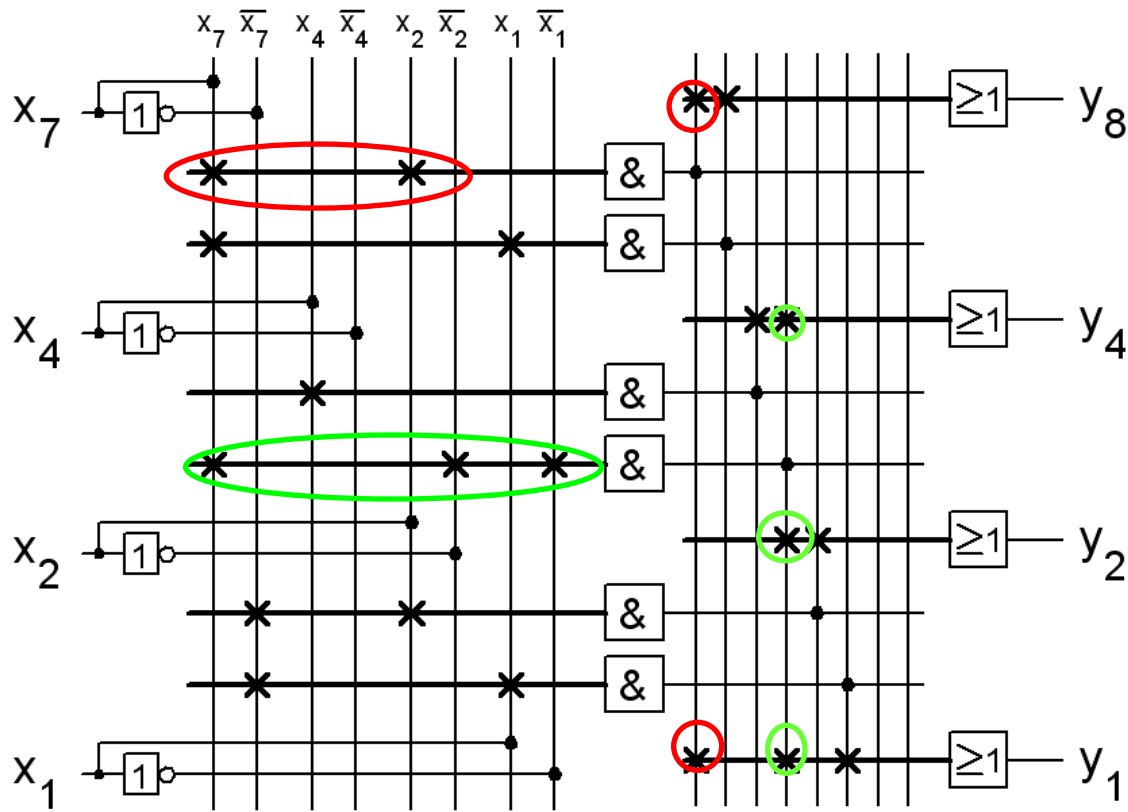
$$y_8 = x_7 x_2 + x_7 x_1$$

$$y_4 = x_4 + x_7 \overline{x_2} x_1$$

$$y_2 = \overline{x_7} x_2 + x_7 \overline{x_2} x_1$$

$$y_1 = \overline{x_7} x_1 + x_7 x_2 + x_7 \overline{x_2} x_1$$

Grind-delning!



Reella tal

Decimalkomma ”,” och Binärpunkt ”.”

$$10,3125_{10} = 1010.0101_2$$

Bin \rightarrow Dec

1 0 1 0 . 0 1 0 1

2^3 2^2 2^1 2^0 . 2^{-1} 2^{-2} 2^{-3} 2^{-4}

8 4 2 1 0,5 0,25 0,125 0,0625

$$8 + 0 + 2 + 0 + 0 + 0,25 + 0 + 0,0625 = 10,3125$$

Ex 1.2b

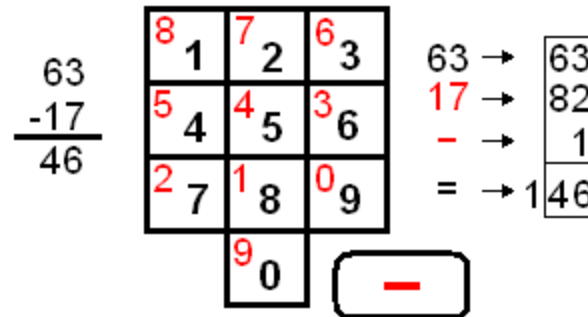
$$110100.010_2 =$$

Ex 1.2b

$$\begin{aligned} 110100.010_2 &= \\ &= (2^5 + 2^4 + 2^2 + 2^{-2} = 32 + 16 + 4 + 0.25) = \\ &= 52,25_{10} \end{aligned}$$

Komplementräkning

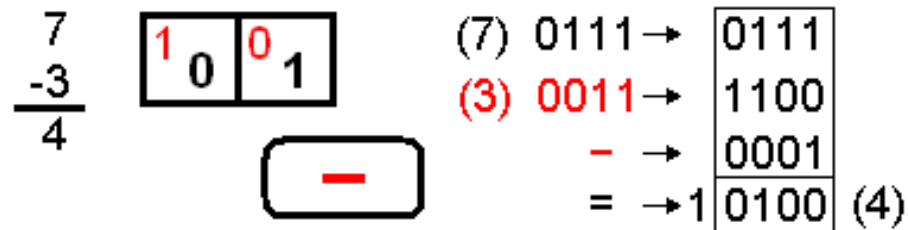
Subtraktion med en additionsmaskin = komplementräkning



$$63 - 17 = 46$$

Talet -17 slås in som med röda siffror 17 och blir då 82. När - tangenten trycks in adderas 1. Resultatet blir: $63+82+1 = 146$. Om bara två siffror visas: 46

2-komplement



Binäertalet 3, 0011, blir negativt -3 genom att man inverterar alla bitar och lägger till ett, 1101.

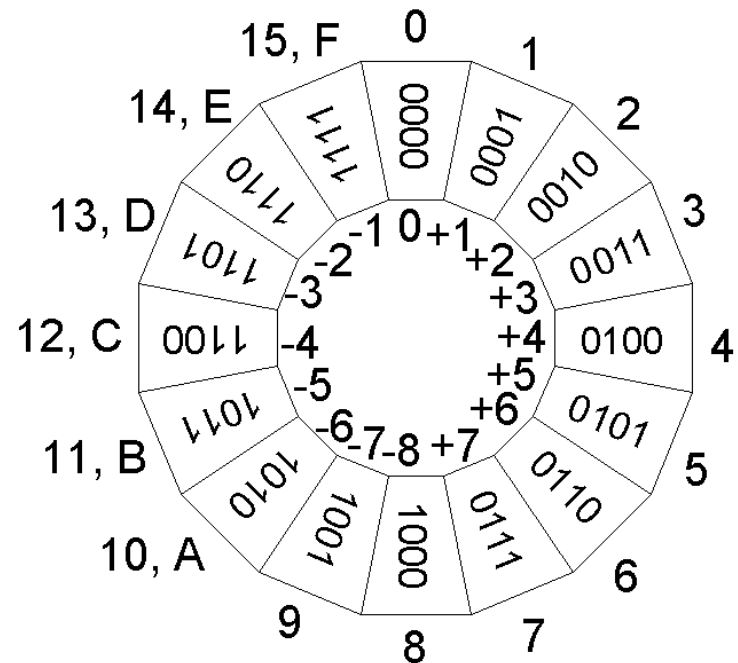
Registeraritmetik

- Datorregister är ”ringar”

 Ett fyra bitars register rymmer $2^4 = 16$ tal.

Antingen 8 positiva (+0...+7) och 8 negativa (-1...-8) tal ”med tecken”, eller 16 (0...F) ”teckenlösa” tal.

Om registret är fullt gör ”+1” att det ”slår runt”.



Registerlängd

- 4 bitar kallas **Nibble**. Registret rymmer $2^4 = 16$ tal.
0...15, -8...+7
- 8 bitar kallas **Byte**. Registret rymmer $2^8 = 256$ tal.
0...255, -128...+127
- 16 bitar kallas **Word**. $2^{16} = 65536$ tal.
0...65535, -32768...+32767

Vanliga registerstorlekar är idag 32 bitar (DoubleWord) och 64 bitar (QuadWord).

Dessa storleksbenämningar är de som används av Windows-programmet Calculator. Word kan ofta vara 32 bitar i stället.

Ex 1.8

Skriv följande tal "med tecken" med två-komplementsnotation, $x = (x_6, x_5, x_4, x_3, x_2, x_1, x_0)$.

a) -23

b) -1 =

c) +38 =

d) -64 =

Ex 1.8

Skriv följande tal "med tecken" med två-komplementsnotation, $x = (x_6, x_5, x_4, x_3, x_2, x_1, x_0)$.

a) $-23 = (+23_{10} = 0010111_2 \rightarrow -23_{10} = 1101000_2 + 1_2) = 1101001_2$
 $= 105_{10}$

b) $-1 =$

c) $+38 =$

d) $-64 =$

Ex 1.8

Skriv följande tal "med tecken" med två-komplementsnotation, $x = (x_6, x_5, x_4, x_3, x_2, x_1, x_0)$.

$$\text{a) } -23 = (+23_{10} = 0010111_2 \rightarrow -23_{10} = 1101000_2 + 1_2) = 1101001_2 = 105_{10}$$

$$\text{b) } -1 = (+1_{10} = 0000001_2 \rightarrow -1_{10} = 1111110_2 + 1_2) = 1111111_2 = 127_{10}$$

$$\text{c) } +38 =$$

$$\text{d) } -64 =$$

Ex 1.8

Skriv följande tal "med tecken" med två-komplementsnotation, $x = (x_6, x_5, x_4, x_3, x_2, x_1, x_0)$.

$$\text{a) } -23 = (+23_{10} = 0010111_2 \rightarrow -23_{10} = 1101000_2 + 1_2) = 1101001_2 = 105_{10}$$

$$\text{b) } -1 = (+1_{10} = 0000001_2 \rightarrow -1_{10} = 1111110_2 + 1_2) = 1111111_2 = 127_{10}$$

$$\text{c) } +38 = (32_{10} + 4_{10} + 2_{10}) = 0100110_2 = 38_{10}$$

$$\text{d) } -64 =$$

Ex 1.8

Skriv följande tal "med tecken" med två-komplementsnotation,
 $x = (x_6, x_5, x_4, x_3, x_2, x_1, x_0)$.

$$\text{a) } -23 = (+23_{10} = 0010111_2 \rightarrow -23_{10} = 1101000_2 + 1_2) = 1101001_2 = 105_{10}$$

$$\text{b) } -1 = (+1_{10} = 0000001_2 \rightarrow -1_{10} = 1111110_2 + 1_2) = 1111111_2 = 127_{10}$$

$$\text{c) } +38 = (32_{10} + 4_{10} + 2_{10}) = 0100110_2 = 38_{10}$$

$$\text{d) } -64 = (+64_{10} = 1000000_2 \text{ är ett för stort positivt tal!} \\ \text{men fungerar ändå } -64_{10} \rightarrow 0111111_2 + 1_2) = 1000000_2 = 64_{10}$$

Ex 2.1

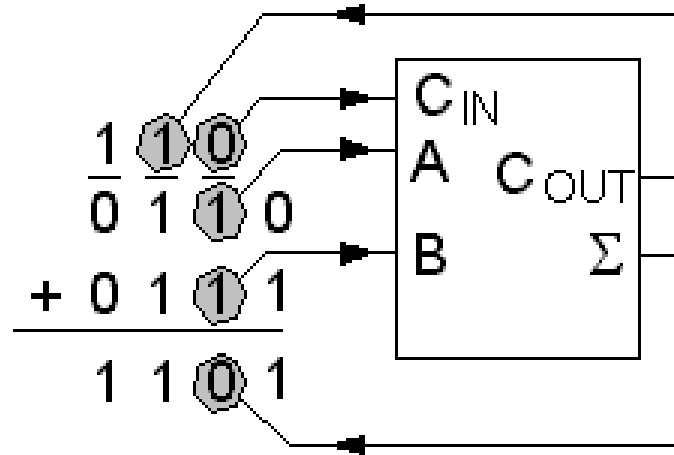
a) $110 + 010$ b) $1110 + 1001$

c) $11\ 0011.01 + 111.1$ d) $0.1101 + 0.1110$

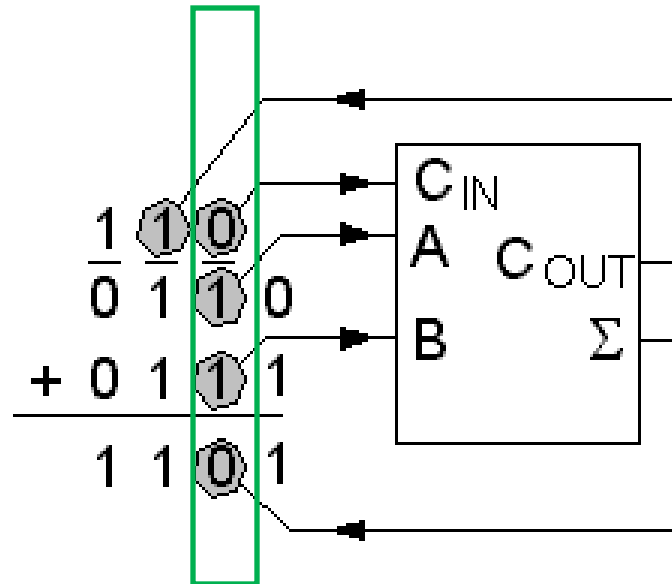
$$\begin{array}{r} \text{a)} \quad \begin{array}{r} \frac{1}{1} \frac{1}{1} 1 0 \\ + 0 1 0 \\ \hline 1 0 0 0 \end{array} \quad \text{b)} \quad \begin{array}{r} \frac{1}{1} 1 1 1 0 \\ + 1 0 0 1 \\ \hline 1 0 1 1 1 \end{array} \end{array}$$

$$\begin{array}{r} \text{c)} \quad \begin{array}{r} 1 1 \frac{1}{0} \frac{1}{0} \frac{1}{1} 1.0 1 \\ + \quad \quad \quad 1 1 1.1 \\ \hline 1 1 1 0 1 0.1 1 \end{array} \quad \text{d)} \quad \begin{array}{r} \frac{1}{0.1} \frac{1}{1} 1 0 1 \\ + 0.1 1 1 0 \\ \hline 1.1 0 1 1 \end{array} \end{array}$$

Heladderaren



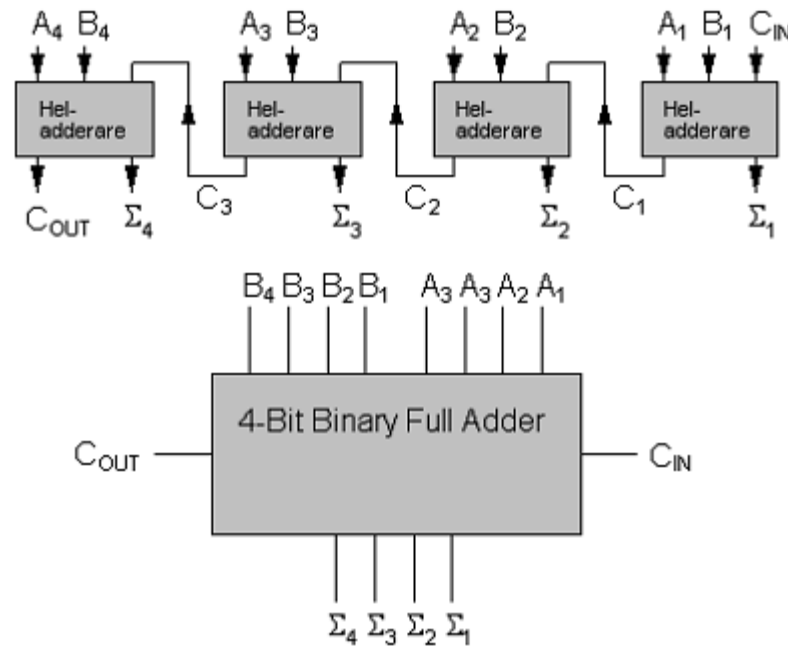
Heladderaren



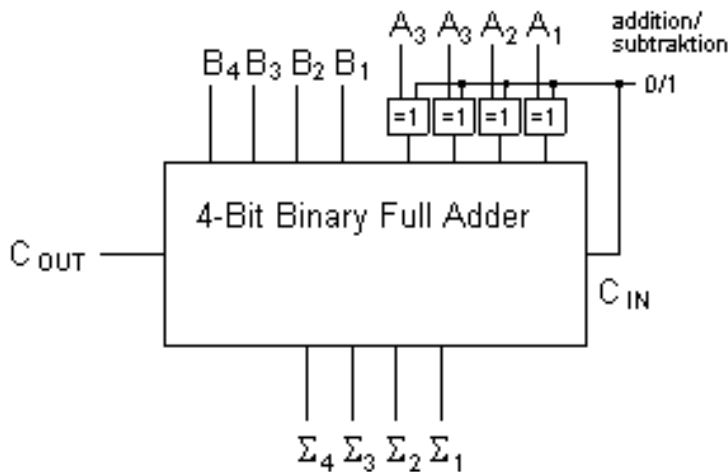
Ett grindnät som gör en binär addition på en valfri bitposition med två binära tal kallas för en **Heladderare**.

4-bits adderare

En additionskrets för binära fyrbitstal består således av fyra heladderarkretsar.



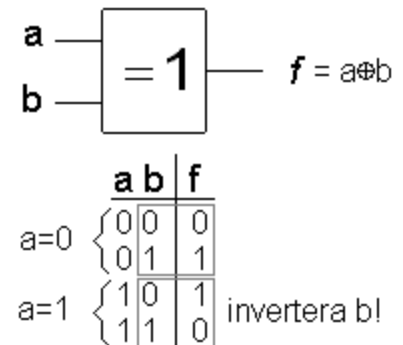
Subtraktion?



Subtraktion av binära tal kan ske genom sk. komplementräkning. Negativa tal representeras då av sannkomplementet, vilket innebär att alla bitar inverteras och en etta adderas till talet.

Man utnyttjar då additionskretsen även till subtraktion.

Rent kretsmässigt kan man lösa inverteringen med XOR-grindar, och man adderar en etta till talet genom att låta $C_{IN} = 1$.



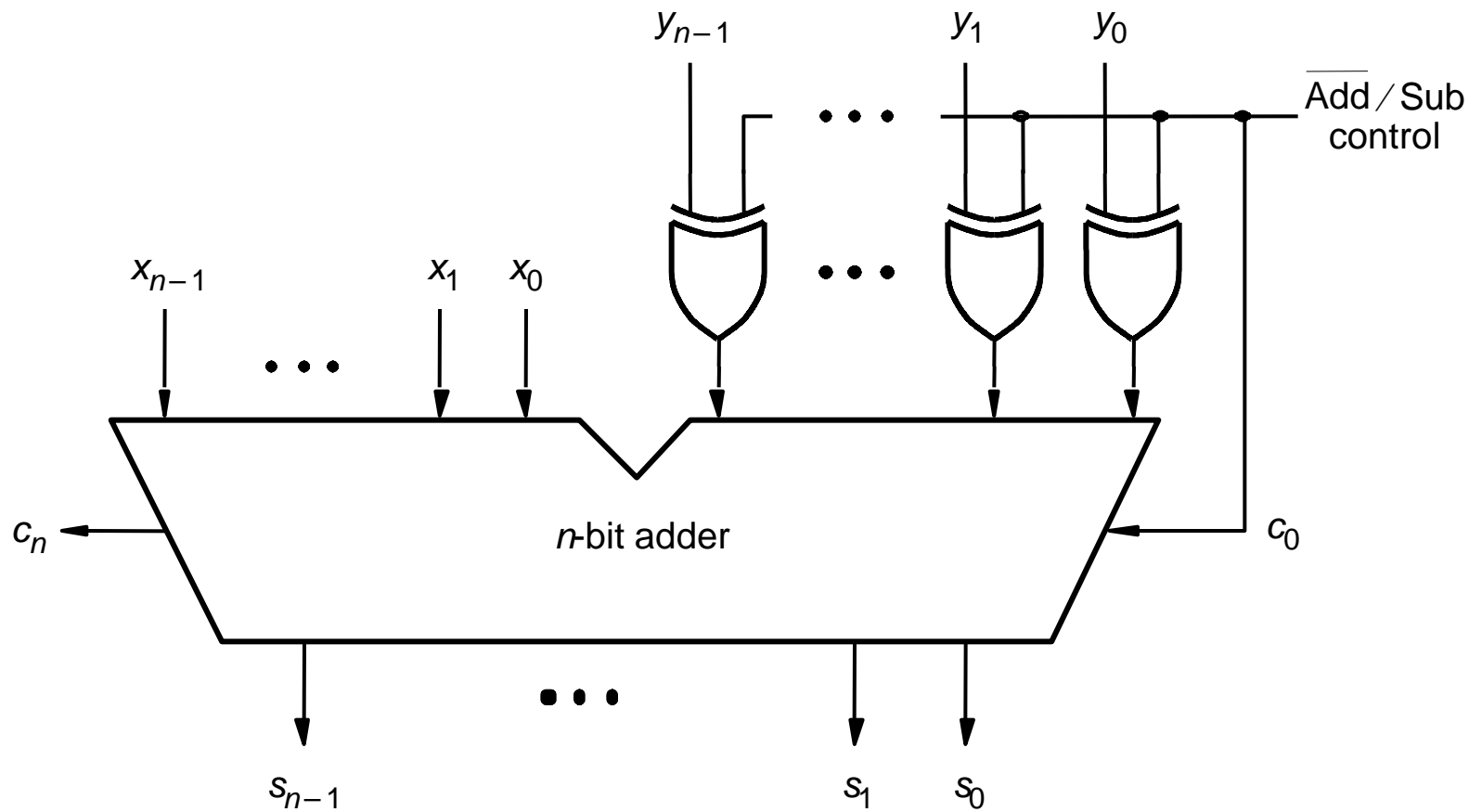
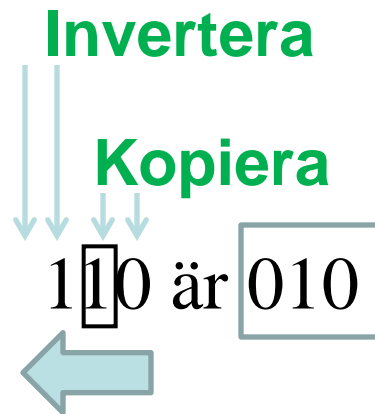


Figure 5.13. Adder/subtractor unit.

2-komplementet ”snabbt”

- För att lätt ta fram 2-komplementet av ett binärtal kan man använda följande förfarande:
 - Börja från höra sidan
 - Kopiera alla bitar från binärtalet som är 0 och den första 1:an
 - Invertera alla andra bitar

Exempel: 2-komplement från



Ex 2.2

Addera eller subtrahera (addition med motsvarande negativa tal) nedanstående tal. Talen skall representeras som binära 4-bitstal (Nibble) på två-komplementform.

a) $1 + 2$ b) $4 - 1$ c) $7 - 8$ d) $-3 - 5$

Exemplets negativa tal:

$$-1_{10} = (+1_{10} = 0001_2 \rightarrow -1_{10} = 1110_2 + 1_2) = 1111_2$$

$$-8_{10} = (+8_{10} = 1000_2 \rightarrow -8_{10} = 0111_2 + 1_2) = 1000_2$$

$$-3_{10} = (+3_{10} = 0011_2 \rightarrow -3_{10} = 1100_2 + 1_2) = 1101_2$$

$$-5_{10} = (+5_{10} = 0101_2 \rightarrow -5_{10} = 1010_2 + 1_2) = 1011_2$$

2.2

$$-1_{10} = 1111_2$$

$$-8_{10} = 1000_2$$

$$-3_{10} = 1101_2$$

$$-5_{10} = 1011_2$$

$$1+2=3$$

a)

$$\begin{array}{r|l} 0 & 0 & 0 & 1 & =1 \\ + & 0 & 0 & 1 & 0 & =2 \\ \hline & 0 & 0 & 1 & 1 & =3 \end{array}$$

$$4-1=3$$

b)

$$\begin{array}{r|l} 1 & 1 & & & & \\ \hline 0 & 1 & 0 & 0 & =4 \\ + & 1 & 1 & 1 & 1 & =-1 \\ \hline \cancel{1} & 0 & 0 & 1 & 1 & =3 \end{array}$$

$$7-8=-1$$

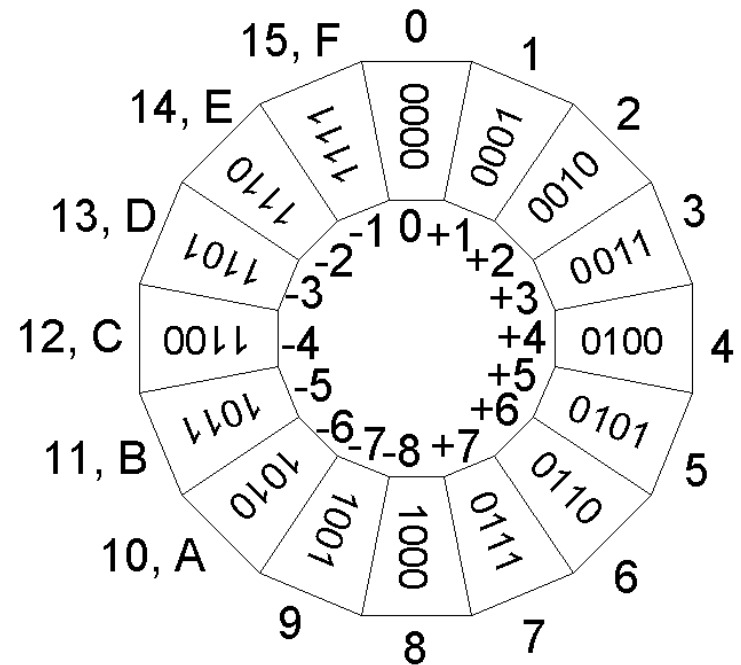
c)

$$\begin{array}{r|l} 0 & 1 & 1 & 1 & =7 \\ + & 1 & 0 & 0 & 0 & =-8 \\ \hline & 1 & 1 & 1 & 1 & =-1 \end{array}$$

$$-3-5=-8$$

d)

$$\begin{array}{r|l} 1 & 1 & 1 & 1 & 1 & \\ \hline 1 & 1 & 0 & 1 & =-3 \\ + & 1 & 0 & 1 & 1 & =-5 \\ \hline \cancel{1} & 1 & 0 & 0 & 0 & =-8 \end{array}$$



Ex 2.3 a,b

Multipluera för hand följande par av teckenlösa binära tal.

a) $110 \cdot 010$ b) $1110 \cdot 1001$

$$110 \cdot 010 = (6 \cdot 2 = 12) = 1100$$

$$1110 \cdot 1001 = 1111110$$

a)

$$\begin{array}{r} 110 \\ \times 010 \\ \hline 000 \\ 110 \\ + 000 \\ \hline 01100 = 12 \end{array}$$

b)

$$\begin{array}{r} 1110 \\ \times 1001 \\ \hline 1110 \\ 0000 \\ 0000 \\ + 1110 \\ \hline 1111110 = 126 \end{array}$$

Ex 2.3 c,d

Multiplitera för hand följande par av teckenlösa binära tal.

$$\begin{array}{r} 110011.01 \cdot 111.1 = \\ = 110000000.011 \\ \text{c) } \begin{array}{r} 110011\boxed{01} \\ \times \quad 111\boxed{1} \\ \hline 11001101 \\ 11001101 \\ 11001101 \\ + 11001101 \\ \hline 110000000\boxed{011} \end{array} \\ = 110000000.011 \end{array}$$

$$(51,25 \cdot 7,5 = 384,375)$$

$$\begin{array}{r} 0.1101 \cdot 0.1110 = \\ = 0.10110110 \\ \text{d) } \begin{array}{r} \quad \quad \boxed{1101} \\ \times \quad \boxed{1110} \\ \hline \quad \quad 0000 \\ \quad \quad 1101 \\ \quad \quad 1101 \\ + \quad 1101 \\ \hline \boxed{10110110} \end{array} \\ = 0.10110110 \end{array}$$

$$(0,8125 \cdot 0,875 = 0.7109375)$$

Fixpunktsberäkning är en "heltalsmultiplikation", binärpunkten sätts in först i resultatet.

Ex 2.4

Dividera för hand följande par av teckenlösa binära tal.

Trappan:

$$110/010=(6/2=3)=011$$

a)

$$\begin{array}{r} \overline{) 110} \\ \underline{10} \\ 10 \\ \underline{ 10} \\ 0 \end{array}$$

$$1110/1001=(14/9)=1.10\dots$$

b)

$$\begin{array}{r} \overline{) 1110} \\ \underline{1001} \\ 1010 \\ \underline{ 1001} \\ 1000 \\ \underline{ 1001} \\ 10000 \\ \underline{ 1001} \\ 111\dots \end{array}$$

Vid heltalsdivision blir svaret i stället 1.

Ex 2.4

Dividera för hand följande par av teckenlösa binära tal.

Kort division:

a) $110/010=(6/2=3)=011$

$$\frac{\boxed{110}}{10} = \frac{110}{10} = 1 \quad \frac{\begin{array}{c} \boxed{1} \\ 110 \end{array}}{10} = 11$$

Ex 2.4

Dividera för hand följande par av teckenlösa binära tal.

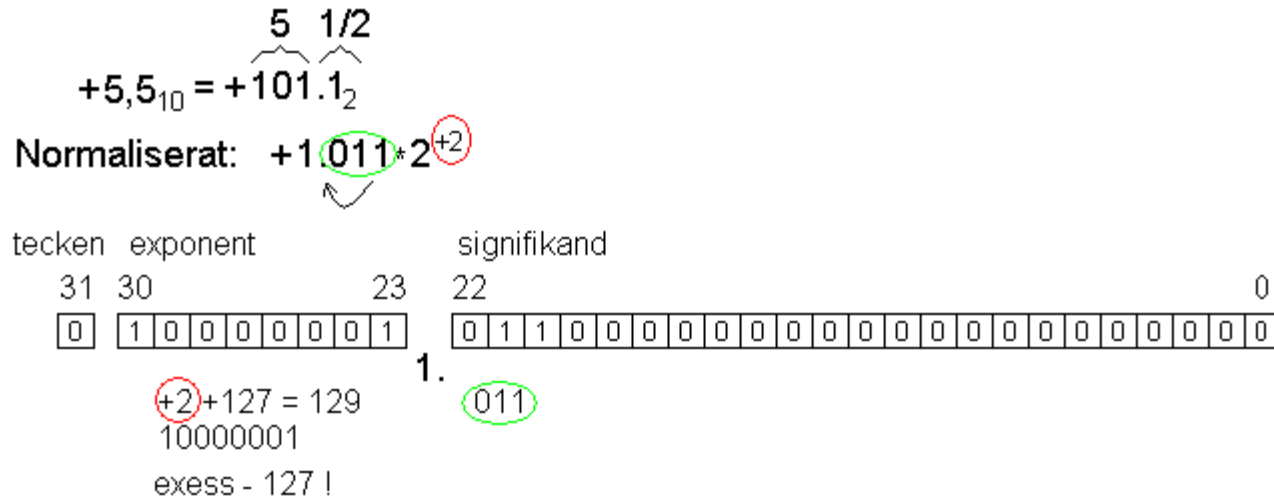
Kort division:

$$\text{b) } 1110/1001=(14/9=1,55\dots)=1.10\dots$$

$$\frac{1110}{1001} = \frac{1110}{1001} = 1 \quad \frac{1110.}{1001} = 1. \quad \frac{1110.}{1001} = 1.1 \quad \dots$$

Vid heltalsdivision blir svaret i stället 1.

IEEE – 32 bit float



Genom att exponenteten skrivs excess-127 kan flyttal storleksorteras med vanlig heltalsaritmetik!

[Dec → IEEE-754](#)

2.5 Flyttalsformat

IEEE 32 bit flyttal

```
s  eeeeeeee  ffffffffffffffffffffffffffffffff
31 30      23 22                                0
```

2.5 Flyttalsformat

IEEE 32 bit flyttal

```
s  eeeeeeee  ffffffffffffffffffffffffffffffff
31 30      23 22                                0
```

Vad blir:

```
    4    0    C    8    0    0    0    0
0100000011001000000000000000000000000000
```


2.5 Flyttalsformat

IEEE 32 bit flyttal

```
s  eeeeeeee  ffffffffffffffffffffffffffffffff
31 30      23 22                                0
```

Vad blir:

```
    4    0    C    8    0    0    0    0
01000000110010000000000000000000
```

```
0 10000001 100100000000000000000000
```

```
+ 129-127      1 + 0.5+0.0625
```

$$+1,5625 \cdot 2^2 = +6,25$$

IEEE-754 Floating-Point Conversion from 32-bit Hexadecimal to Floating-Point - Mozilla Firefox

Arkiv Redigera Visa Historik Bokmärken Verktyg Hjälp

http://babbage.cs.qc.cuny.edu/IEEE-754/32bit.html

IEEE-754 Floating-Point Conversion f...

IEEE-754 Floating-Point Conversion

From 32-bit Hexadecimal Representation To Decimal Floating-Point

Along with the Equivalent 64-bit Hexadecimal and Binary Patterns

Enter the 32-bit hexadecimal representation of a floating-point number here,
then click the **Compute** button.

Hexadecimal Representation:

Results:

Decimal Value Entered:

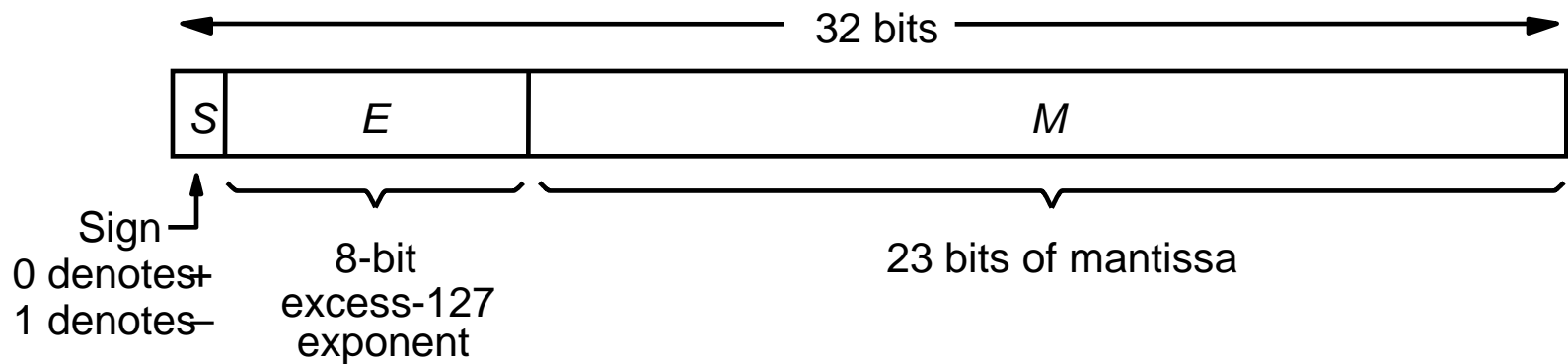
Single precision (32 bits):

Binary: Status:

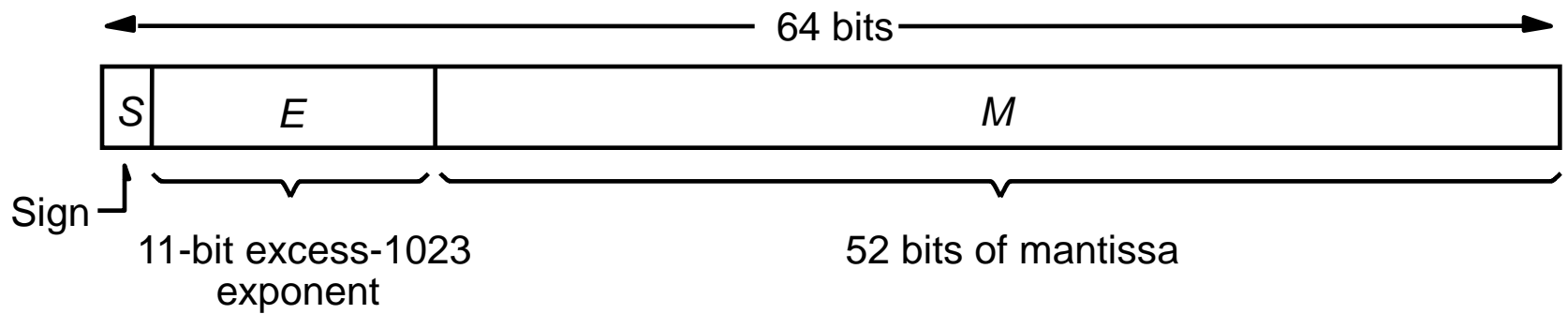
Bit 31 Sign Bit <input type="text" value="0"/> 0: + 1: -	Bits 30 - 23 Exponent Field <input type="text" value="10000001"/> Decimal value of exponent field and exponent <input type="text" value="129"/> - 127 = <input type="text" value="2"/>	Bits 22 - 0 Significand <input type="text" value="1.100100000000000000000000"/> Decimal value of the significand <input type="text" value="1.5625000"/>
--	--	---

DEC - IEEE-754

William Sandqvist william@kth.se



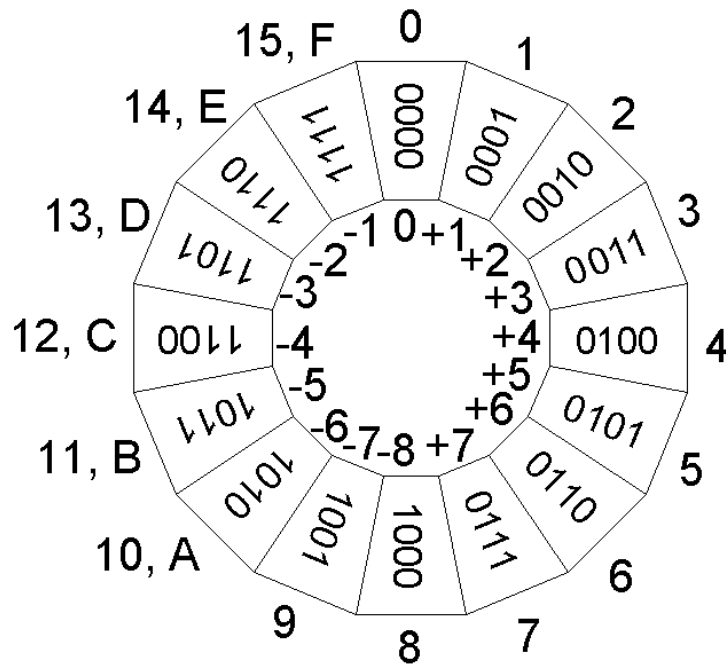
(a) Single precision



(b) Double precision

Figure 5.34. IEEE Standard floating-point formats.

Overflow



När man räknar med "tal med tecken" kan summan av två positiva tal *felaktigt* bli negativ (tex. "+4" + "+5" = "-7"), liksom summan av två negativa tal *felaktigt* kan bli positiv (tex. "-6" + "-7" = "+3").

Detta kallas för **Overflow**.

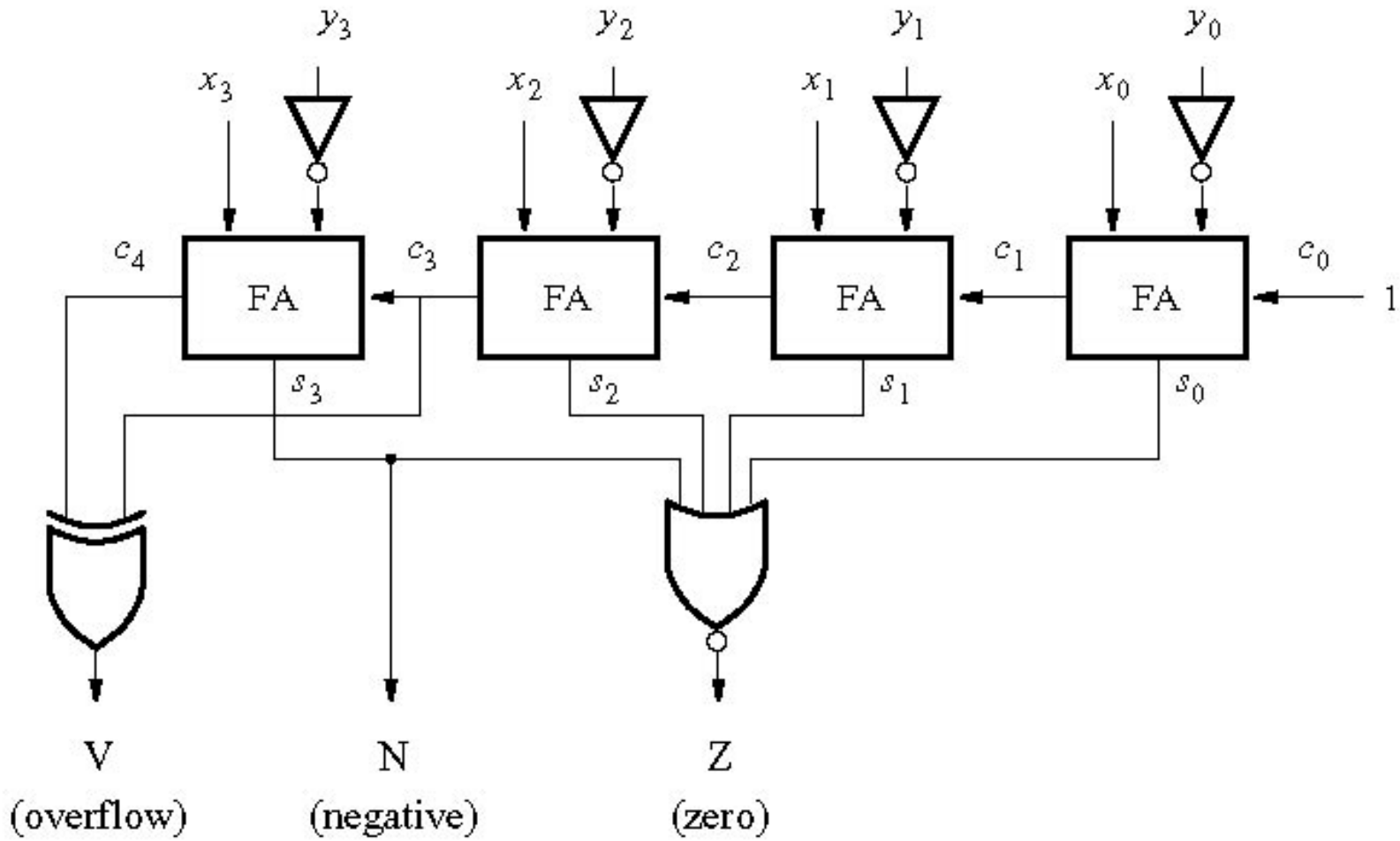
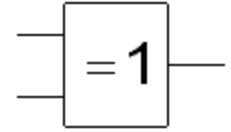


Figure 5.42. A comparator circuit.

Logik för att detektera overflow



*XOR testar
"olikhet"*

För 4-bit-tal

Overflow om c_3 och c_4 är *olika*

Annars är det inte overflow

$$\text{Overflow} = c_3 \bar{c}_4 + \bar{c}_3 c_4 = c_3 \oplus c_4$$

För n -bit-tal

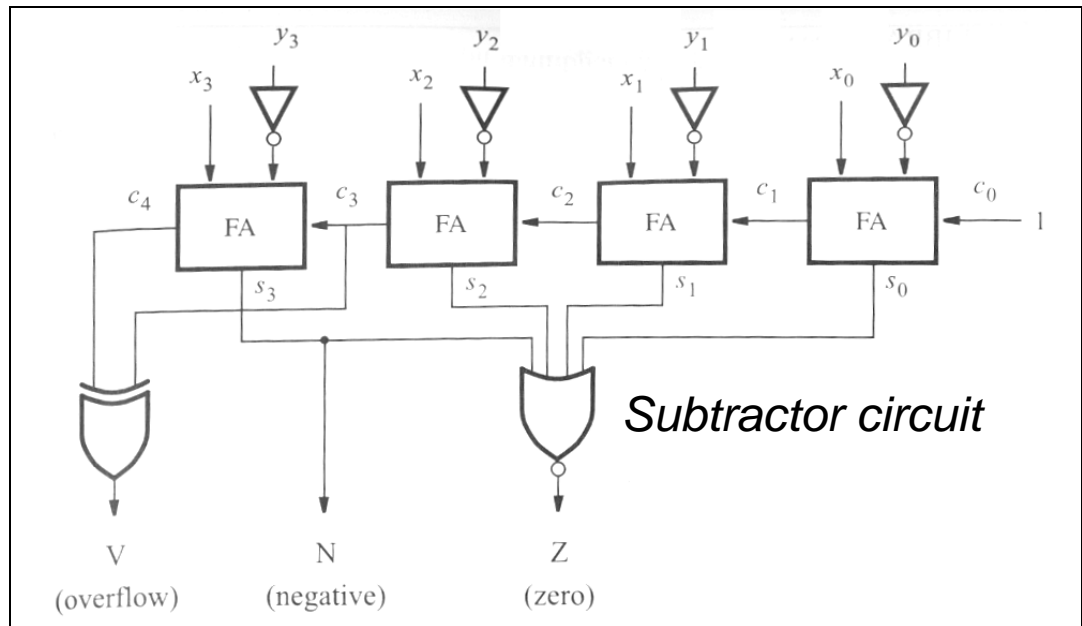
$$\text{Overflow} = c_{n-1} \oplus c_n$$

BV ex 5.10, < > =

Flags, Comparator. Two four-bit signed numbers, $X = x_3x_2x_1x_0$ and $Y = y_3y_2y_1y_0$, can be compared by using a subtractor circuit, which performs the operation $X - Y$. The three Flag-outputs denote the following:

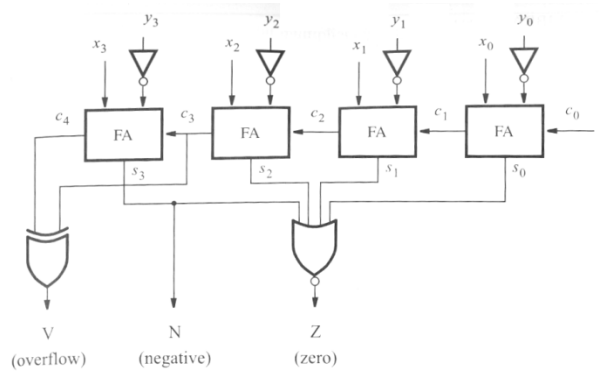
- $Z = 1$ if the result is 0; otherwise $Z = 0$
- $N = 1$ if the result is negative; otherwise $N = 0$
- $V = 1$ if arithmetic overflow occurs; otherwise $V = 0$

Show how Z , N , and V can be used to determine the cases $X = Y$, $X < Y$, $X > Y$.



BV ex 5.10

$X = Y ?$

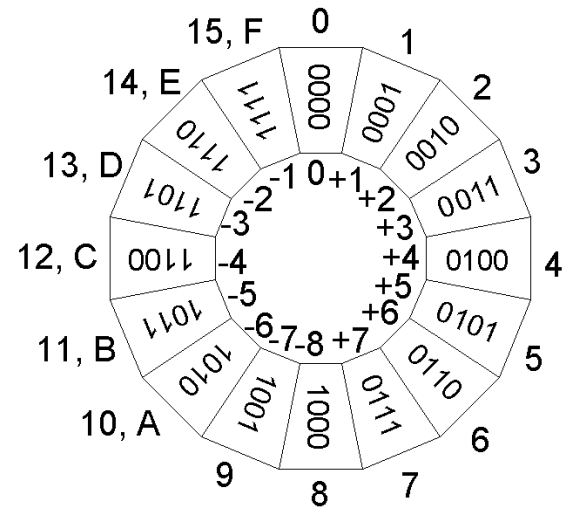


$X - Y$

$$V = c_4 \oplus c_3 \quad N = s_3$$

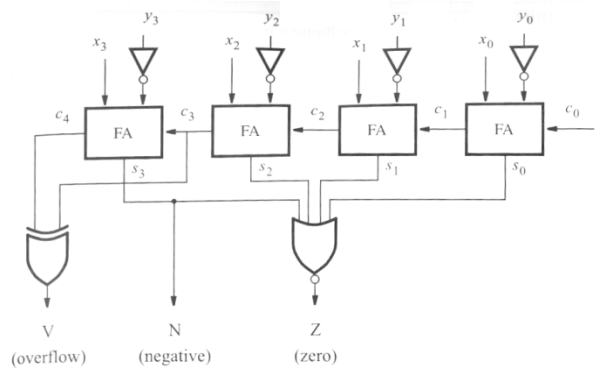
$$Z = \overline{(s_3 + s_2 + s_1 + s_0)}$$

$X = Y ?$



BV ex 5.10

$X = Y ?$



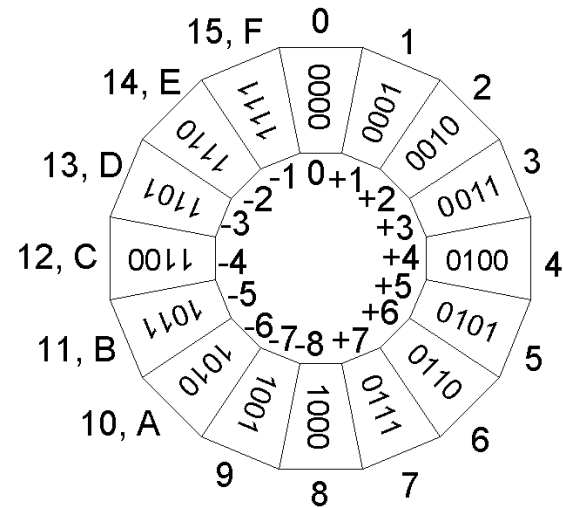
$$X - Y$$

$$V = c_4 \oplus c_3 \quad N = s_3$$

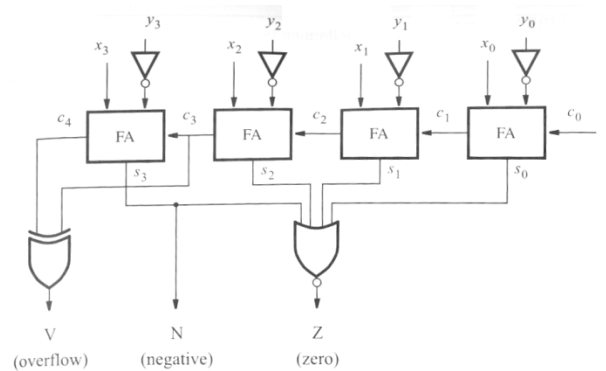
$$Z = (s_3 + s_2 + s_1 + s_0)$$

$X = Y ?$

$$X = Y \Rightarrow Z = 1$$



BV ex 5.10



$$X - Y$$

$$V = c_4 \oplus c_3 \quad N = s_3$$

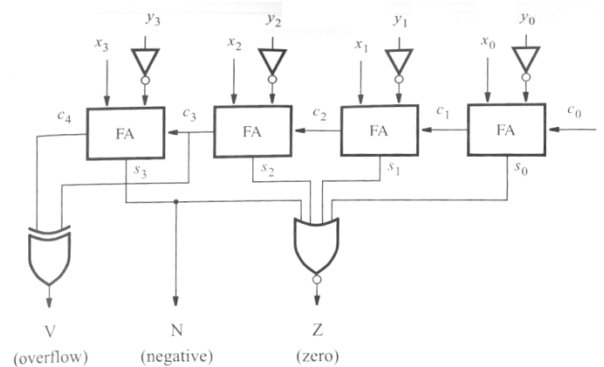
$$Z = \overline{(s_3 + s_2 + s_1 + s_0)}$$

$X < Y$?

Några testtal:

	$X < Y$	$X - Y$	V	N
3	4	$3 - 4 = -1$	0	1
-4	-3	$-4 - -3 = -1$	0	1
-3	4	$-3 - 4 = -7$	0	1
-5	4	$-5 - 4 = +7$	1	0

BV ex 5.10



$$X - Y$$

$$V = c_4 \oplus c_3 \quad N = s_3$$

$$Z = \overline{(s_3 + s_2 + s_1 + s_0)}$$

$X < Y$?

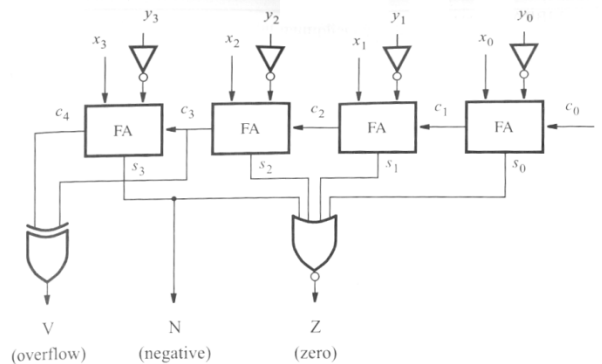
Om X och Y har samma tecken kommer $X - Y$ alltid att ligga inom talområdet. Dvs. $V = 0$. X, Y positiva tex. $3 - 4$ $N = 1$. X, Y negativa tex. $-4 - (-3)$ $N = 1$.

Om X neg och Y pos och $X - Y$ ligger inom talområdet, blir $V = 0$ och $N = 1$. Tex. $-3 - 4$.

Om X neg och Y pos men $X - Y$ ligger utanför talområdet, blir $V = 1$. Då blir $N = 0$. Ex. $-5 - 4$.

- Vid $X < Y$ blir flaggorna V och N således *alltid olika*. Detta kan indikeras med XOR.

BV ex 5.10



$$X - Y$$

$$V = c_4 \oplus c_3 \quad N = s_3$$

$$Z = (s_3 + s_2 + s_1 + s_0)$$

$X < Y$?

Om X och Y har samma tecken kommer $X - Y$ alltid att ligga inom talområdet. Dvs. $V = 0$. X, Y positiva tex. $3 - 4$ $N = 1$. X, Y negativa tex. $-4 - (-3)$ $N = 1$.

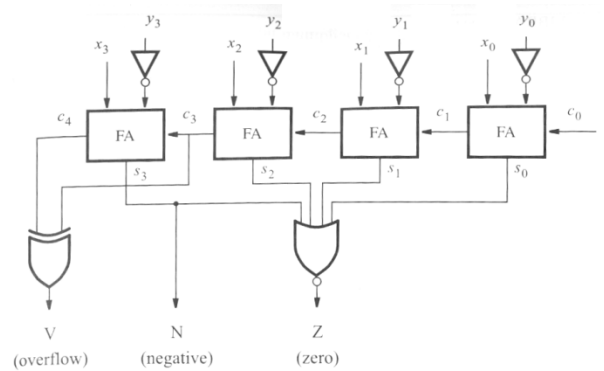
Om X neg och Y pos och $X - Y$ ligger inom talområdet, blir $V = 0$ och $N = 1$. Tex. $-3 - 4$.

Om X neg och Y pos men $X - Y$ ligger utanför talområdet, blir $V = 1$. Då blir $N = 0$. Ex. $-5 - 4$.

- Vid $X < Y$ blir flaggorna V och N således *alltid olika*. Detta kan indikeras med XOR.

$$X < Y \Rightarrow N \oplus V$$

BV ex 5.10



$$X - Y$$

$$V = c_4 \oplus c_3 \quad N = s_3$$

$$Z = \overline{(s_3 + s_2 + s_1 + s_0)}$$

$$X = Y \Rightarrow Z = 1$$

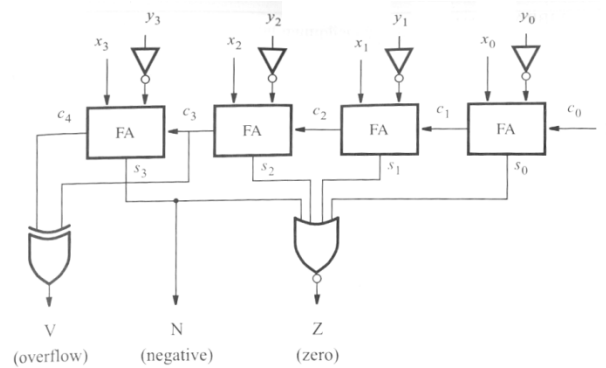
$$X < Y \Rightarrow N \oplus V$$

$$X \leq Y \Rightarrow$$

$$X > Y \Rightarrow$$

$$X \geq Y \Rightarrow$$

BV ex 5.10



$$X - Y$$

$$V = c_4 \oplus c_3 \quad N = s_3$$

$$Z = \overline{(s_3 + s_2 + s_1 + s_0)}$$

$$X = Y \Rightarrow Z = 1$$

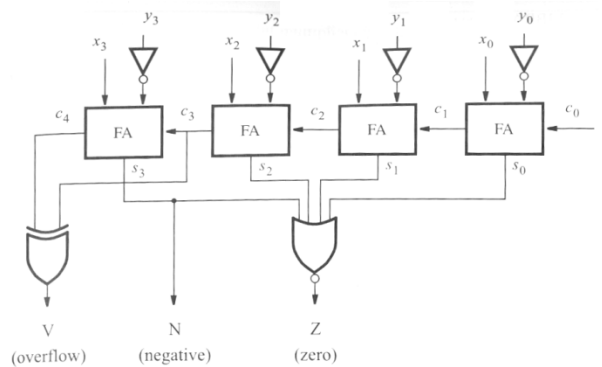
$$X < Y \Rightarrow N \oplus V$$

$$X \leq Y \Rightarrow Z + N \oplus V$$

$$X > Y \Rightarrow \overline{Z + N \oplus V} = \overline{Z} \cdot \overline{(N \oplus V)}$$

$$X \geq Y \Rightarrow \overline{N \oplus V}$$

BV ex 5.10



$$X - Y$$

$$V = c_4 \oplus c_3 \quad N = s_3$$

$$Z = \overline{(s_3 + s_2 + s_1 + s_0)}$$

$$X = Y \Rightarrow Z = 1$$

$$X < Y \Rightarrow N \oplus V$$

$$X \leq Y \Rightarrow Z + N \oplus V$$

$$X > Y \Rightarrow \overline{Z + N \oplus V} = \overline{Z} \cdot \overline{(N \oplus V)}$$

$$X \geq Y \Rightarrow \overline{N \oplus V}$$

*Så här kan en dator
göra de vanligaste
jämförelserna ...*

Träslöjds adderaren



Rippel carry kan åskådliggöras med denna video ...

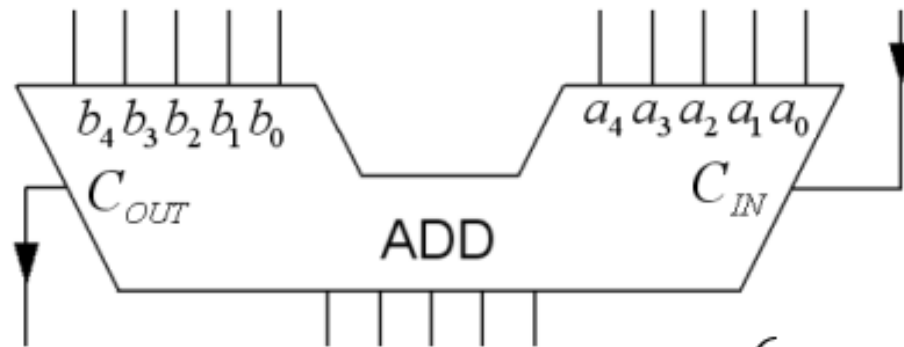


[Marble adding machine](#)

William Sandqvist william@kth.se

Ex 8.11 Multiply with 6 ?

x_3	x_2	x_1	x_0		1	0

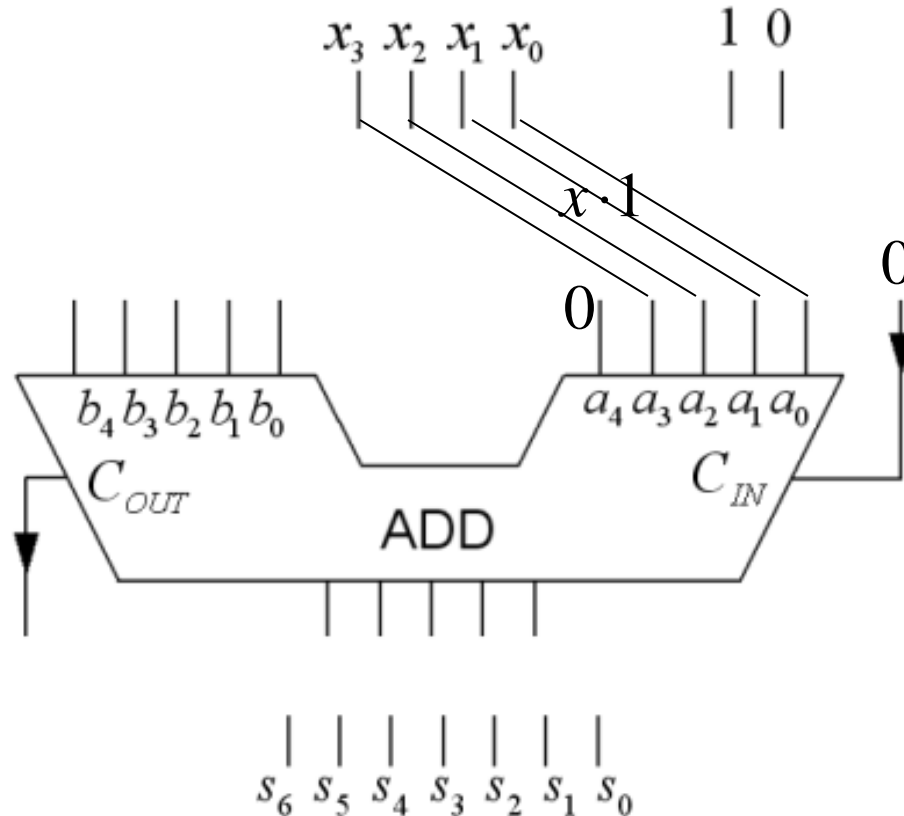


$$s = 6 \times x =$$

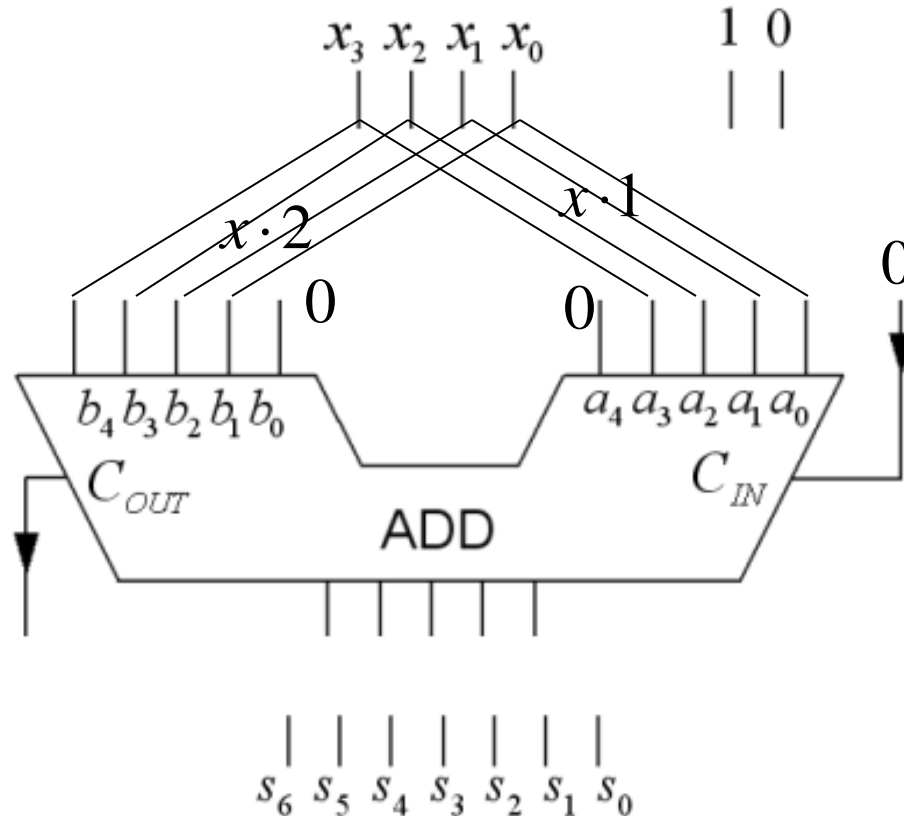
$$= 2 \times (2 \times x + 1 \times x)$$

s_6	s_5	s_4	s_3	s_2	s_1	s_0	

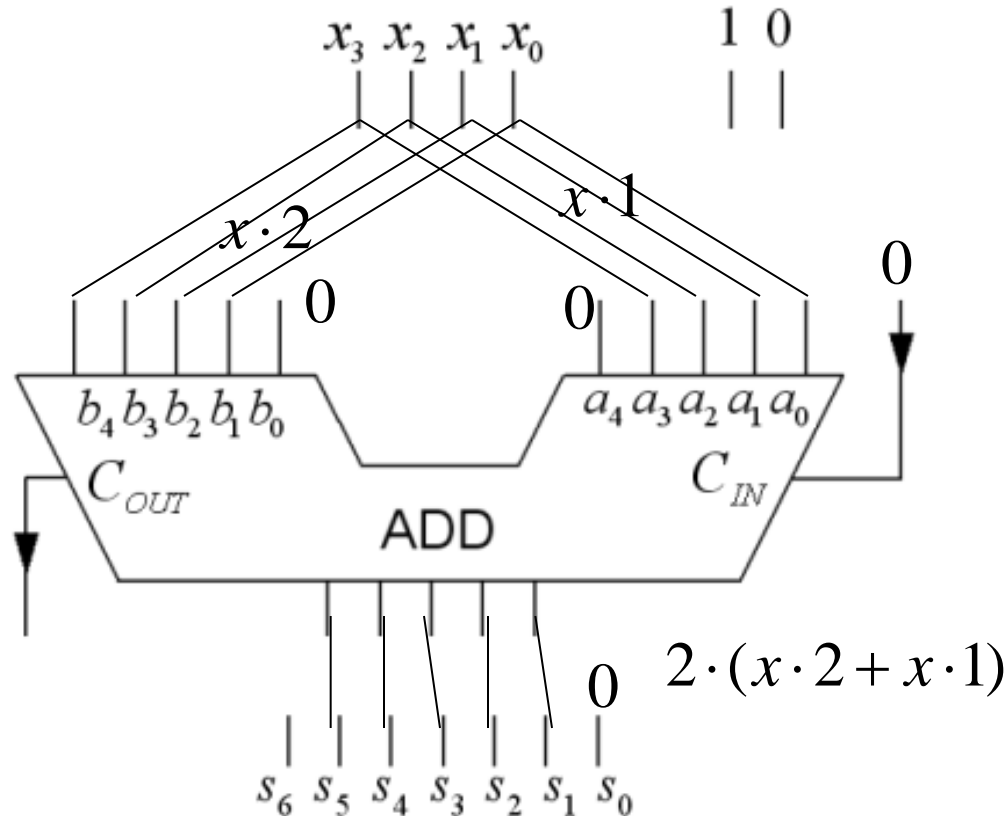
Ex 8.11 Multiply with 6 !



Ex 8.11 Multiply with 6 !



Ex 8.11 Multiply with 6 !



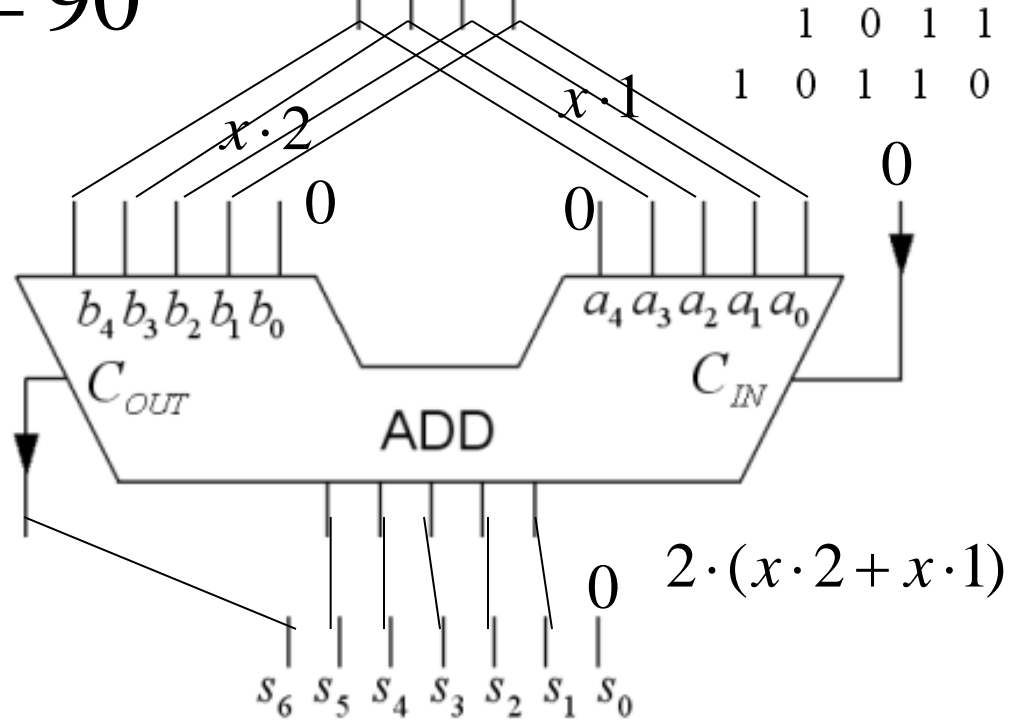
Ex 8.11 Multiply with 6 !

$$15 \cdot 6 = 90$$

$$1111 = 15$$

$x_3 \ x_2 \ x_1 \ x_0$

$$\begin{array}{r}
 \underline{1} \ \underline{1} \ \underline{1} \ \underline{1} \ \underline{0} \ \underline{0} \\
 0 \ \boxed{1 \ 1 \ 1 \ 1} \ 0 \quad 15 \times 2 \\
 + \ 0 \ 0 \ \boxed{1 \ 1 \ 1 \ 1} \quad 15 \times 1 \\
 \hline
 1 \ 0 \ 1 \ 1 \ 0 \ 1 \\
 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \quad \times 2
 \end{array}$$



$$1011010 = 90$$

William Sandqvist william@kth.se