

Codering van getallen

Natuurlijke getallen

(positieve gehele getallen)

- Codering volgens het binaire getalstelsel
- Het bereik van de te coderen getallen wordt beperkt door de woordlengte van de gebruikte bitstring

vb.

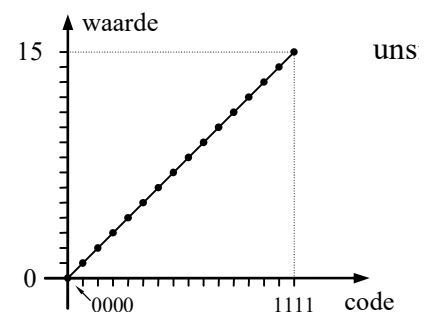
Met een byte: alle natuurlijke getallen van 0 tem 255

Met een woord (16 bits): van 0 tem 65535

- = UM = **Unsigned mode** = alles wat tot nu toe gezien is

•

- **We noteren geen teken!**



Gehele getallen

(positieve en negatieve)

- = SM = **Signed mode** (of signed numbers)
- **Tekendragende getallen!**

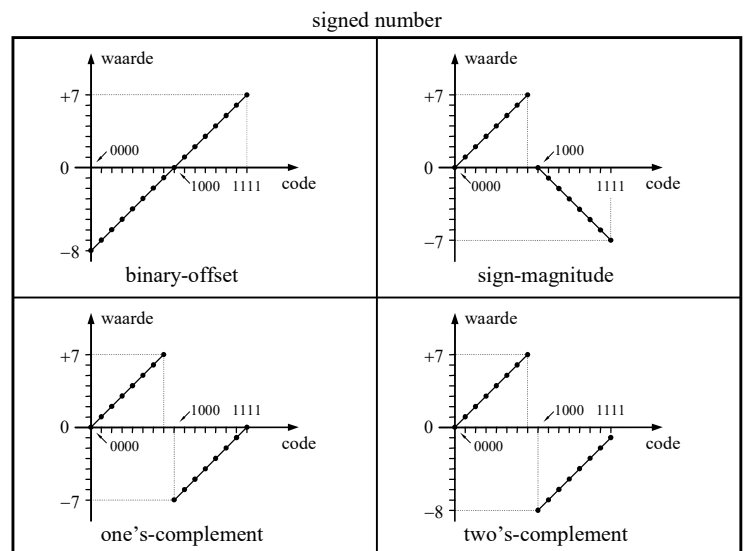
(Verpicht een teken, + of -)

Vb

- +101101001
- -101100101

Nadelen:

- **binary-offset**: de bitpatronen voor positieve getallen (signed mode) zijn niet de zelfde als voor de absolute waarden (unsigned mode)
- **sign-magnitude** en **one's complement**: de waarde 0 op twee manieren kunnen weergeven - een nodeloze complicatie (+0 = 0000 of -0 = 1000)
- **two's complement**: probleem opgelost door de waarde van alle negatieve getallen met één te verlagen, zodat de lijst van -1 tot -128 loopt



Rationale getallen

- Elk decimaal (komma)getal met **eindig veel** decimalen
- Het quotient van 2 gehele getallen (**deling van 2 gehele getallen**)

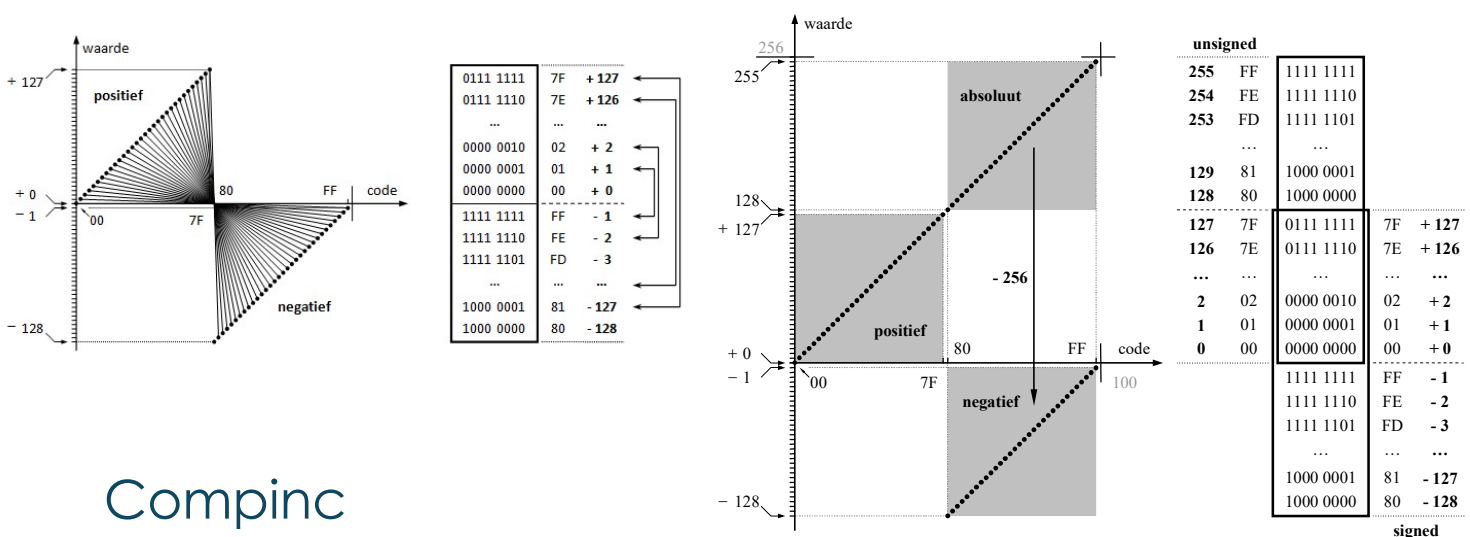
Vb

- +101101.10010
- -101101.1010
- +0101

Codering van gehele getallen:

2's complement

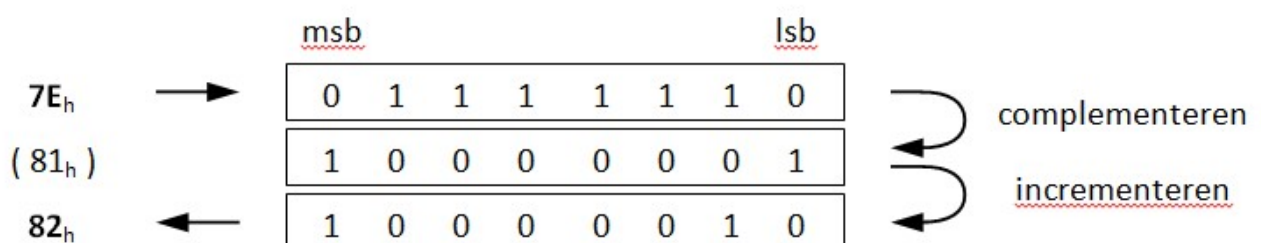
- **Two's complement = 2's complement**
- **De gebieden van unsigned mode en signed mode overlappen elkaar.**



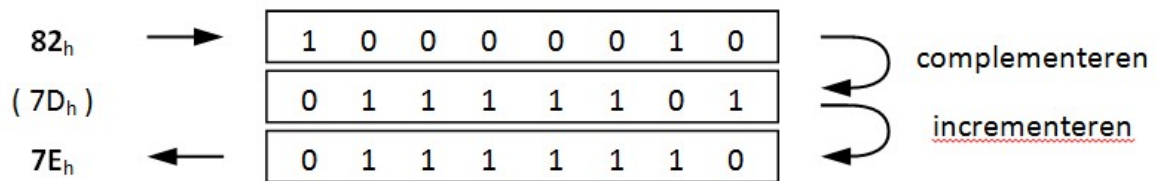
Compinc

- **comp = complementeren (= eerst bits inverteren)**
- **inc = incrementeren (= dan +1)**

van positief naar negatief



van negatief naar positief



Codes

Eigenschappen

gewogen codes (*weighted code*)

= codes waarbij de decimale waarde teruggevonden wordt door aan elke bit een gewicht toe te kennen en de producten van de gewichten met hun overeenkomstige bits te sommeren (= geziene "gewicht-methode")

Vb : De standaard BCD-code is een gewogen code

→ "BCD8421" → de cijfers specificeren het gewicht van elke bit

Vb : De Gray-code is geen gewogen code,

→ dus niet praktisch voor rekenkundige bewerkingen

Zelfcomplementerende code (self complementary code, self complementing code)

= codes waarbij het inverse van het gecodeerde getal het 9-complement oplevert van dit getal

9-complement = het verschil tussen 9, 99, 999, ... en dit getal

De standaard **BCD8421-code** is niet zelfcomplementerende code

→ Het 9-complement van $0011_2 (3_{10})$ is $1100_2 \neq 6_{10} \quad (9-3=6)$

De standaard **excess-3 code** is zelfcomplementerende code

→ het 9-complement van $0011_2 (0_{10})$ is $1100_2 (9_{10})=9 \quad (9-0=9)$

→ het 9-complement van $0100_2 (1_{10})$ is $1011_2 (8_{10})=8 \quad (9-1=8)$

Eén-wisselcode of cyclische code

= code waarbij 2 opeenvolgende codewoorden (\approx waarden) slechts in één bit van elkaar verschillen

Vb gray-code

De standaard **gray-code** is Eén-wisselcode of cyclische code

dec.	binair	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Reflecterende code

= een soort “gespiegelde code” in de opbouw van de tabel

Vb gray-code

BCD-code

De BCD-code: (Binary Coded Decimal) is een code waarbij men elk cijfer van een decimaal getal voorstelt door 4 bits.

Vb: $3196_{10} = 0011\ 0001\ 1001\ 0110 = (3 = 0011) (1 = 0001) (9 = 1001) (6 = 0110)$

- Van de 16 combinaties die met 4 bits mogelijk zijn, worden er in de BCD-code 6 niet gebruikt. (Gebruik 0 \rightarrow 9, niet gebruik 10 \rightarrow 15)
- De BCD-code is dus minder efficiënt dan de zuivere binaire codering.
- Ze wordt vooral toegepast bij digitale uitlezingen.

excess-3 code

- 3-teveel code

BCD-code \rightarrow excess-3 code bij elke bit 0011 (3) bij te tellen

Vb

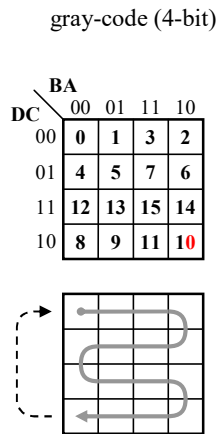
DCB-code	Som	Excess-3	Dec.
48	$0010 + 0011 = 0111 = 7$ $1000 + 0011 = 1011 = 11$	7B	123

dec.	excess-3
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

Gray-code

'minimale verschil code' genoemd (tussen naburige waarden).

G	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0



G	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

dec.	binair	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

samenvatting

	gewogen codes	Zelfcomplementerende code	Eén-wisselcode of cyclische code	Reflecterende code
BCD-code	✓	✗		
excess-3 code		✓		
Gray-code	✗		✓	✓

Combinatorische schakelingen

codeomvormer

Deze transformeert gecodeerde informatie naar een andere code

Niet-gecodeerde informatie = **begrijpbaar voor de mens**.

decoder

De **decoder** transformeert gecodeerde informatie naar niet-gecodeerde informatie.

VB

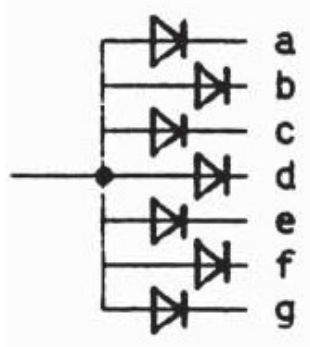
→ **7-segment decoder**

→ **1 uit 4 decoder** = uitcodeerschakeling

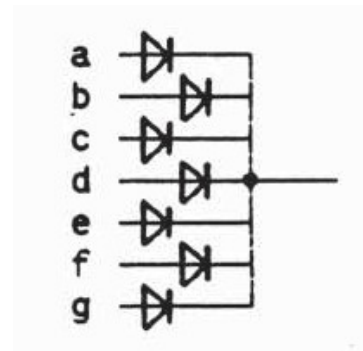
7 – segment



CA = common anode



CC = common cathode

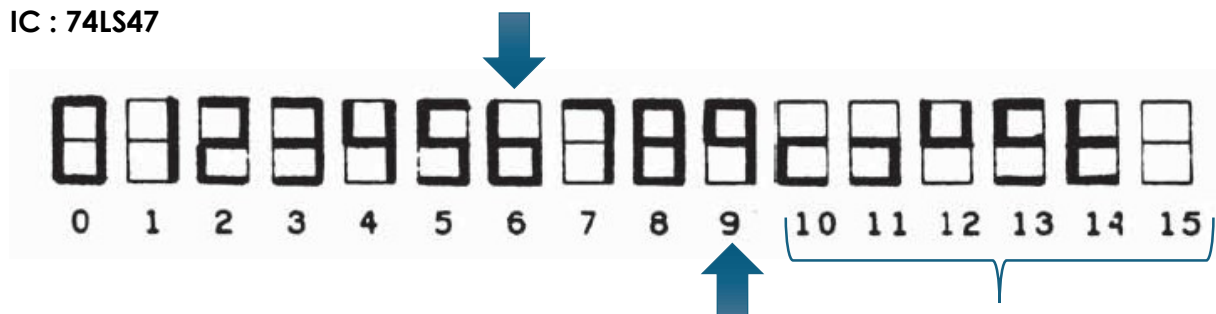


INGANGEN				UITGANGEN						
D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0

Binaire ingang 0001 = 1 → B & C moeten branden → deze zijn 0

In labo

IC : 74LS47

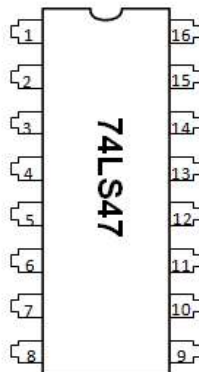
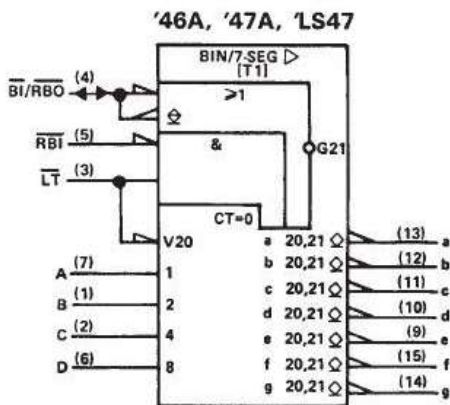


Bemerk:

- Weergave van cijfer 6: geen bovenste streepje
- Weergave van cijfer 9: geen onderste streepje
- Extra symbolen van 10 t.e.m. 15

IC 74LS47

INGANGEN								UITGANGEN						
Dec	\overline{LT}	\overline{RBI}	D	C	B	A	$\overline{BI}/\overline{RBO}$	\overline{a}	\overline{b}	\overline{c}	\overline{d}	\overline{e}	\overline{f}	\overline{g}
0	1	1	0	0	0	0	1	0	0	0	0	0	0	1
1	1	X	0	0	0	1	1	1	0	0	1	1	1	1
2	1	X	0	0	1	0	1	0	0	1	0	0	1	0
3	1	X	0	0	1	1	1	0	0	0	0	1	1	0
4	1	X	0	1	0	0	1	1	0	0	1	1	0	0
5	1	X	0	1	0	1	1	0	1	0	0	1	0	0
6	1	X	0	1	1	0	1	0	1	0	0	0	0	0
7	1	X	0	1	1	1	1	0	0	0	1	1	1	1
8	1	X	1	0	0	0	1	0	0	0	0	0	0	0
9	1	X	1	0	0	1	1	0	0	0	0	1	0	0
10	1	X	1	0	1	0	1	1	1	1	0	0	1	0
11	1	X	1	0	1	1	1	1	1	0	0	1	1	0
12	1	X	1	1	0	0	1	1	0	1	1	1	0	0
13	1	X	1	1	0	1	1	0	1	1	0	1	0	0
14	1	X	1	1	1	0	1	1	1	1	0	0	0	0
15	1	X	1	1	1	1	1	1	1	1	1	1	1	1
BI	X	X	X	X	X	X	0	1	1	1	1	1	1	1
RBI	1	0	0	0	0	0	0	1	1	1	1	1	1	1
LT	0	X	X	X	X	X	1	0	0	0	0	0	0	0



3 extra ingangen: LT, RBI en BI/RBO

! led in table staan inverse

LT = Lamp Test

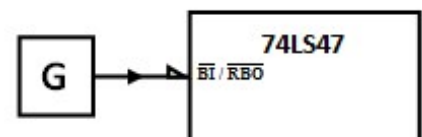
(als deze laag is alle led aan → test leds)

BI = Blanking Input

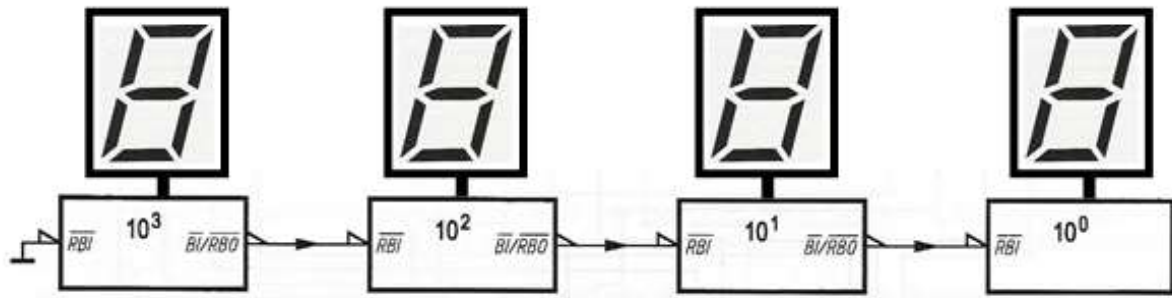
(als deze actief staat gaan alle leds uit)

BI = RBO = knipperen en/of intensiteitsregeling

- δ klein → donkerder display → U_{gem} laag
- δ groot → lichter display → U_{gem} hoog
- Knipperen bij lage frequentie (tot 30 à 50 Hz) voor het blote oog
- Intensiteitsregeling bij hogere frequentie (boven 50 Hz)



RBI = Ripple Blanking Input

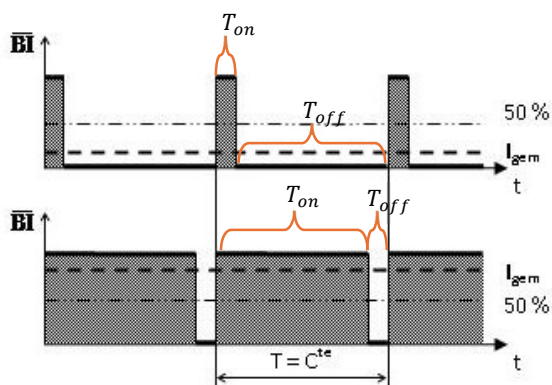


Doel van deze schakeling? Onderdrukking van leidende nullen! (kunnen uitleggen aan de hand van de WT)

Als $\overline{RBI} = 0$ en A, B, C en D=0 (m.a.w. getalwaarde 0) dan worden alle uitgangen 1 en doven de segmenten.

Daarbij wordt ook \overline{RBO} een uitgang en 0 → doorgeven aan volgende IC!!

duty-cycle δ



- 2 signalen met zelfde periode T, dus zelfde frequentie f, maar toch verschillende aantijd T_{on} en uittijd T_{off} !

- $T = T_{on} + T_{off}$

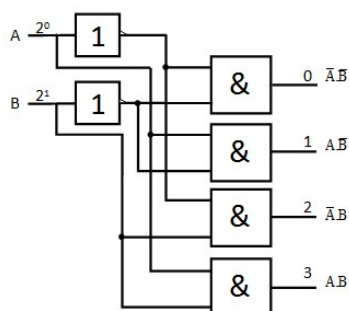
- δ = duty-cycle (δ uitspreken als 'delta') = gedeelte van de periode dat het signaal hoog is (in %)

- $\delta = \frac{T_{on}}{T} = \frac{T_{on}}{T_{on} + T_{off}}$

1 uit 4 decoder

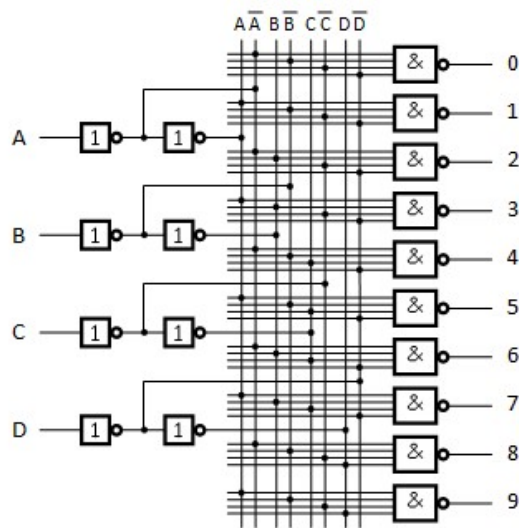
= "1 uit 2^n decoder" met n = aantal ingangen

1 uit 4 decoder = uitcodeerschakeling = **slechts 1 uitgang hoog maken!**

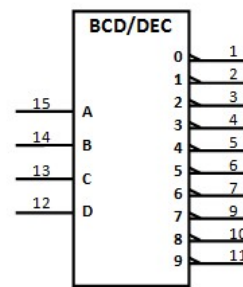


A	B	0	1	2	3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

BCD/DEC decoder



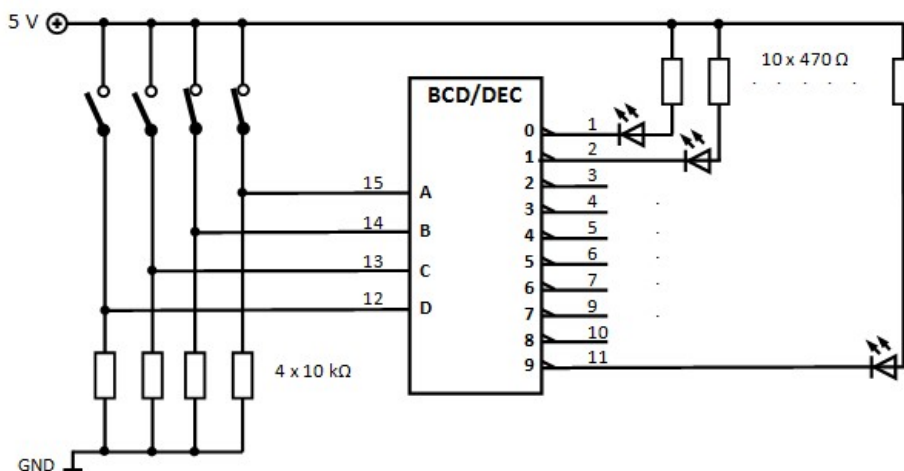
74HC/HCT42



Reden dubbele invertor?

Een ic die hier voor staat (voor A,B,C,D), deze mag niet aan een poort hangen met de veel ingangen voor de stroom, de invertor zorg voor een buffer.

No	INPUTS				OUTPUTS									
	D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	L	L	L	L	L	H	H	H	H	H	H	H	H	H
1	L	L	L	H	H	L	H	H	H	H	H	H	H	H
2	L	L	H	L	H	H	L	H	H	H	H	H	H	H
3	L	L	H	H	H	H	H	L	H	H	H	H	H	H
4	L	H	L	L	H	H	H	H	L	H	H	H	H	H
5	L	H	L	H	H	H	H	H	H	L	H	H	H	H
6	L	H	H	L	H	H	H	H	H	H	L	H	H	H
7	L	H	H	H	H	H	H	H	H	H	H	L	H	H
8	H	L	L	L	H	H	H	H	H	H	H	H	L	H
9	H	L	L	H	H	H	H	H	H	H	H	H	H	L
INVALID	H	L	H	L	H	H	H	H	H	H	H	H	H	H
	H	L	H	H	H	H	H	H	H	H	H	H	H	H
	H	H	L	L	H	H	H	H	H	H	H	H	H	H
	H	H	L	H	H	H	H	H	H	H	H	H	H	H
	H	H	H	L	H	H	H	H	H	H	H	H	H	H
	H	H	H	H	H	H	H	H	H	H	H	H	H	H



type ingang?

Active hoge ingang

Type uitgang?

Active lage uitgang

encoder

De **encoder** maakt van niet gecodeerde informatie gecodeerde informatie.

Encoder = incodeerschakeling = omgekeerde van hiervoor

Toetsenbordencoder



cijfers	BCD-code			
	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

$$A = 1 + 3 + 5 + 7 + 9$$

$$\bar{A} = \bar{1}.\bar{3}.\bar{5}.\bar{7}.\bar{9}$$

$$B = 2 + 3 + 6 + 7$$

$$\bar{B} = \bar{2}.\bar{3}.\bar{6}.\bar{7}$$

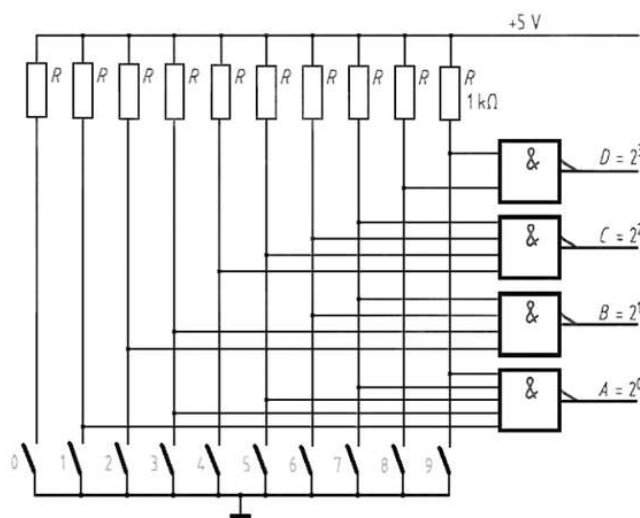
$$C = 4 + 5 + 6 + 7$$

$$\bar{C} = \bar{4}.\bar{5}.\bar{6}.\bar{7}$$

$$D = 8 + 9$$

$$\bar{D} = \bar{8}.\bar{9}$$

A = hoog als 1 of 3 of 5 of 7 of 9 is ingedrukt



$$\bar{A} = \bar{1}.\bar{3}.\bar{5}.\bar{7}.\bar{9}$$

$$\bar{B} = \bar{2}.\bar{3}.\bar{6}.\bar{7}$$

$$\bar{C} = \bar{4}.\bar{5}.\bar{6}.\bar{7}$$

$$\bar{D} = \bar{8}.\bar{9}$$

multiplexer

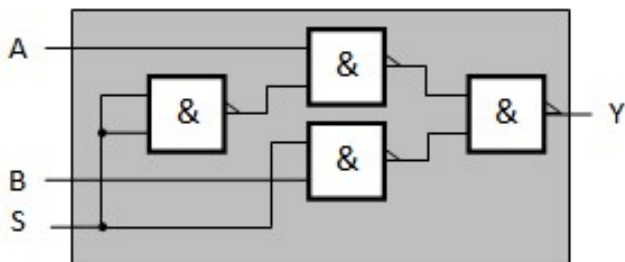
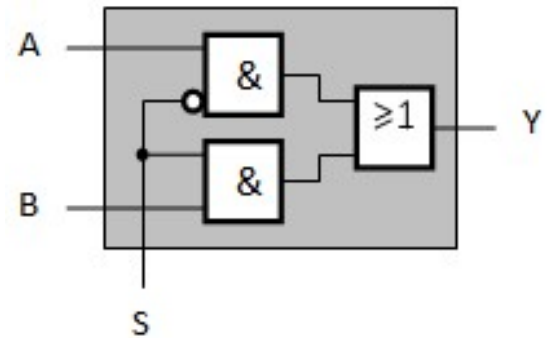
Een **multiplexer** werkt als een keuze schakelaar en verbindt meerdere ingangsklemmen met één uitgangsklem.

met 2 ingangen A en B en dus 1 selectielijn S

Formule: $Y = \bar{S}A + SB$

	A=0	A=1	
S=0	0	1	1
S=1	0	1	0
	B=0	B=1	B=0

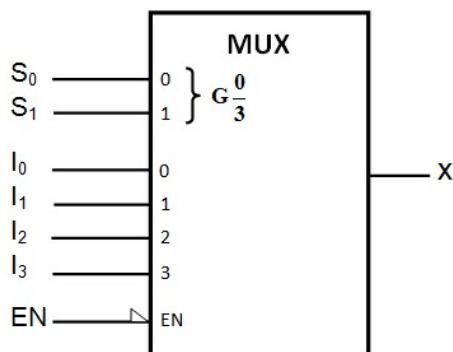
S	A	B	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



Zelfde als de ander schakeling maar alleen met NAND poorten, wat is het voordeel ?

Je kan de schakeling maken met 1 IC de vorige schakeling had je er minimum 2 nodig.

IEC-symbol



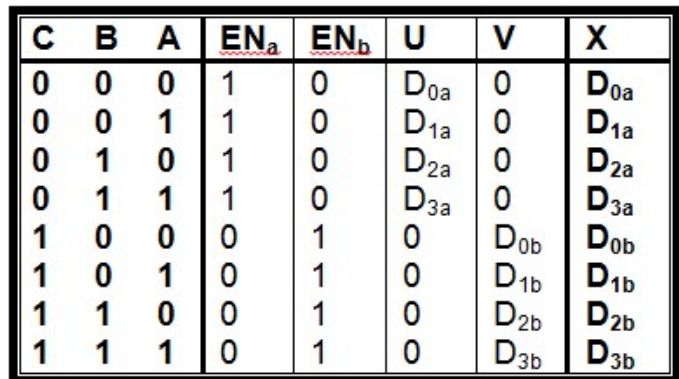
EN	S ₁	S ₀	X
1	x	x	0
0	0	0	I ₀
0	0	1	I ₁
0	1	0	I ₂
0	1	1	I ₃

disable

enable

$$X = I_0 \cdot \bar{S}_1 \cdot \bar{S}_0 + I_1 \cdot \bar{S}_1 \cdot S_0 + I_2 \cdot S_1 \cdot \bar{S}_0 + I_3 \cdot S_1 \cdot S_0$$

Cascadeschakeling: '= soort achtereenschakeling'
(geen serie- of parallelschakeling)



Een **demultiplexer** doet het omgekeerde van een multiplexer en verbindt één ingang met één van de uitgangsklemmen

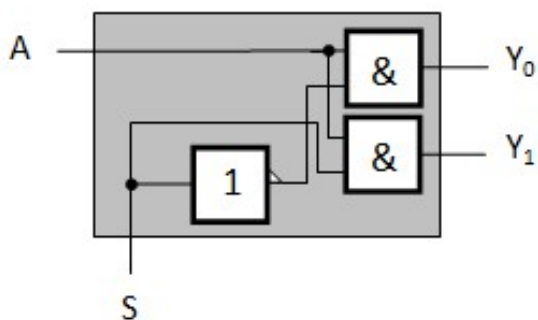
S	A	Y_0	Y_1
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

$$Y_0 = \bar{S}.A$$

Y_0	$A=0$	$A=1$
$S=0$	0	1
$S=1$	0	0

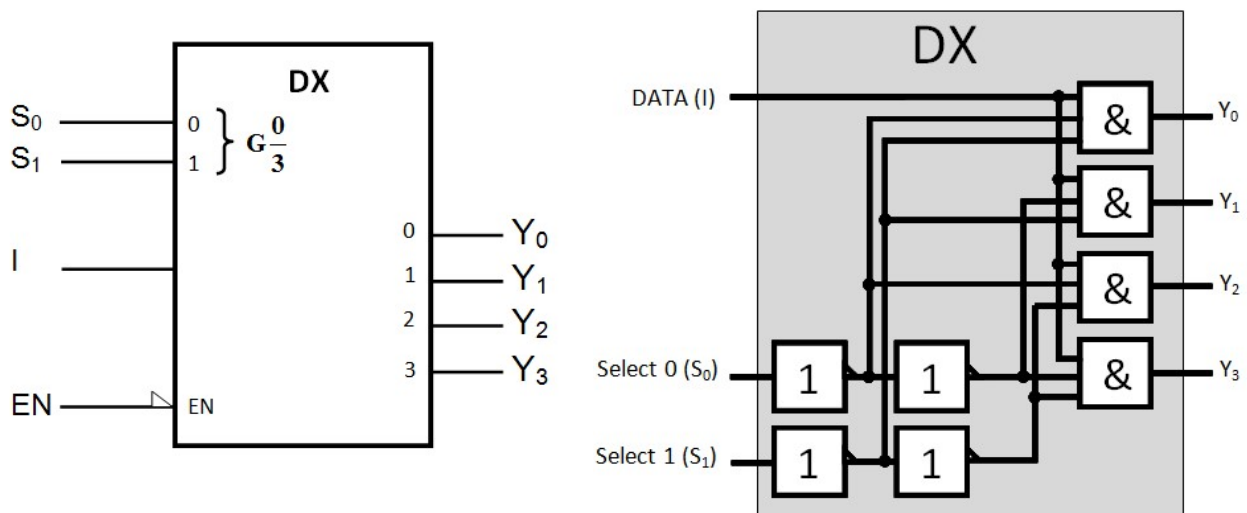
Y_1	$A=0$	$A=1$
$S=0$	0	0
$S=1$	0	1

$$Y_1 = S.A$$



Demultiplexers: met 4 uitgangen en 2 selectielijnen

\overline{EN}	S_1	S_0	Y_0	Y_1	Y_2	Y_3	
1	x	x	0	0	0	0	<u>disable</u>
0	0	0	1	0	0	0	<u>enable</u>
0	0	1	0	1	0	0	
0	1	0	0	0	1	0	
0	1	1	0	0	0	1	



Decodeerschakelingen zijn

een bijzonder type demultiplexers omdat ze uit een aantal ingangssignalen (dus niet 1 zoals bij een demultiplexer) meerdere uitgangssignalen afleiden.

Men spreekt van een **1 naar 4 lijn demultiplexer** (1 to 4 line demultiplexer) omdat er 1 ingang is en 4 uitgangen, maar ook van een **2 naar 4 lijn decoder** (2 to 4 line decoder) waarbij de 2 selectielijnen nu als ingangen gebruikt worden bij dezelfde 4 uitgangen.

De gebruikte IC is dus dezelfde, maar de toepassing ervan is verschillend.