

Digitale Technieken

Les 5: Codering van getallen + codes

Inhoud

- **Codering van getallen (p. 61 t.e.m. p.66 en deel enkel in presentatie)**
 - Natuurlijke getallen
 - Gehele getallen
 - Rationale getallen
- **Codes (p. 21 t.e.m. p. 41)**
 - Cijfercodes: de Gray-code, BCD-code, XS₃-code
 - + Eigenschappen van cijfercodes
 - Alfnumerieke codes
 - Technische codes
 - Andere codes

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)

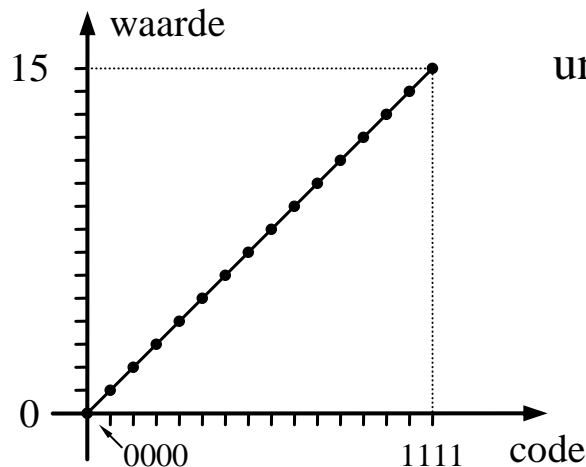
- = SM = *Signed mode* (of *signed numbers*)
- Tekendragende getallen!

Rationale getallen

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

De weergave van natuurlijke getallen

In zijn basisvorm geeft het 'natuurlijke' binaire talstelsel alleen maar absolute waarden weer; dit zijn dus tekenloze getallen (*unsigned numbers*). De 16 bitpatronen die bijvoorbeeld in een 4-bit systeem kunnen gevormd worden van 0000 tot 1111 stellen de positieve getallen van 0 tot 15 voor. Zo'n tekenloze 'absolute' getalweergave noemt men met een vakterm de '*unsigned mode*'.

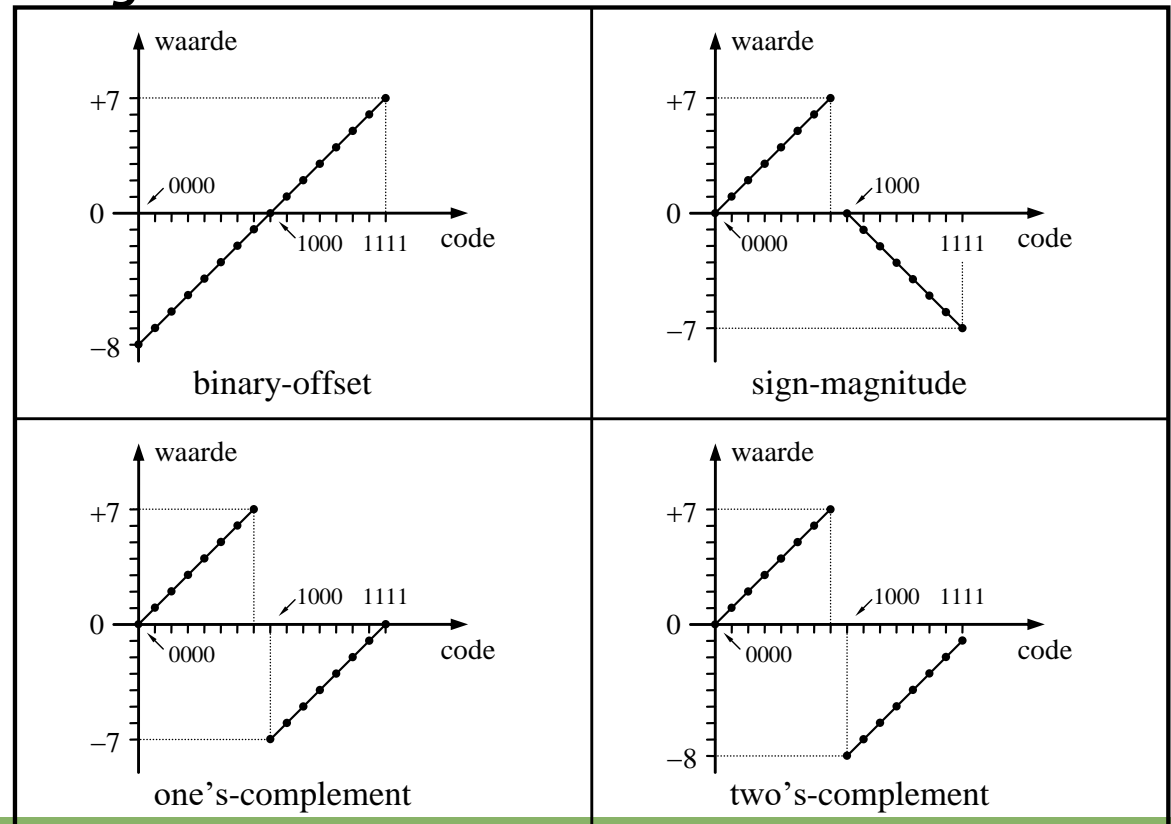


unsigned number

In de natuurlijke binaire code
stellen de bitpatronen van 0000 tot 1111
de positieve getalwaarden voor van 0 tot 15.

De weergave van gehele getallen

- Negatieve getallen weergeven → volgorde van waarden herschikken
- Bovenste helft positieve waarden opofferen → getalbereik ongeveer gelijk verdelen over positieve en negatieve waarden
- = tekendragende getallen = *signed numbers* signed number
- 4 mogelijkheden:



De weergave van gehele getallen

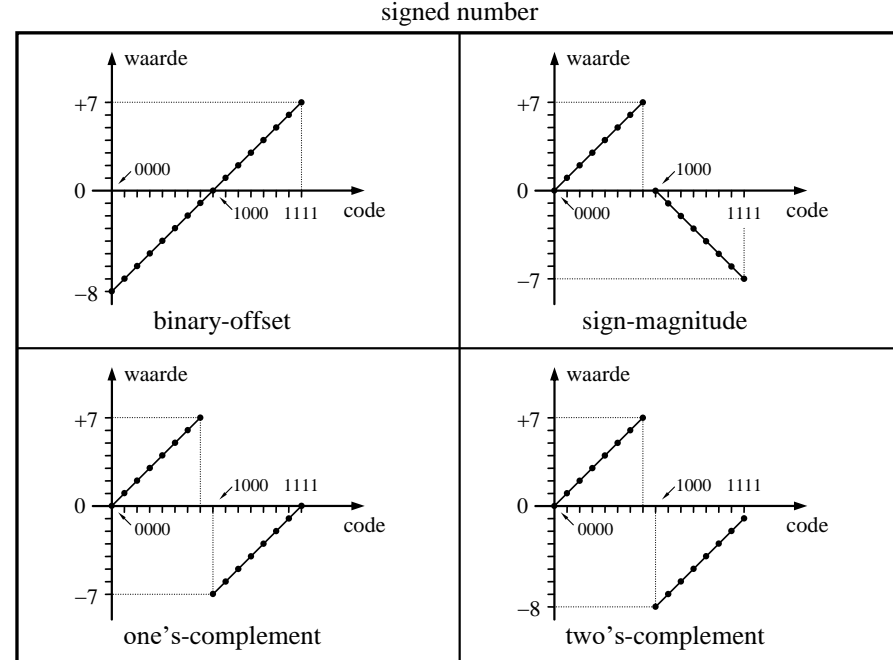
Algemeen:

Om de getalwaarde van een bitstring te kennen moet er apart uitdrukkelijk vermeld worden welke interpretatie er van toepassing is: zonder die expliciete informatie kan men onmogelijk weten of het om een '*signed*' of een '*unsigned*' codering gaat.

In geval van '*signed*' codering moet de juiste methode ook gekend/gegeven zijn (1 van de 4).

De weergave van gehele getallen: tekendragende 8-bit getallen

- Welk codesysteem is nu het beste?
- Linker bit steeds teken + of –
- 7 bits over voor getalwaarde
- Meestal +127 tot -127 (soms -128)



- **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
 - two's complement: probleem opgelost door de waarde van alle negatieve getallen met één te verlagen, zodat de lijst van -1 tot -128 loopt

De weergave van gehele getallen: tekendragende 8-bit getallen

Het belangrijkste criterium om een **tekendragend codeersysteem** voor getallen te beoordelen is dat het **wiskundig consistent** is. Het is immers ontoelaatbaar dat een rekenkundige bewerking tot een onzinnig resultaat zou leiden. Tot zijn essentie herleid komt dit er op neer dat **de optelling van twee tegengestelde waarden als som het getal nul moet opleveren**.

Cijfervoorbeeld: is $(+2) + (-2) = 0$?

binary offset	1000 0010	+	0111 1110	=	(1) 0000 0000	niet nul!
signed magnitude	0000 0010	+	1000 0010	=	(0) 1000 0100	niet nul!
one's complement	0000 0010	+	1111 1101	=	(0) 1111 1111	niet nul!
two's complement	0000 0010	+	1111 1110	=	(1) 0000 0000	som is NUL

Bij binary offset is (1) 0000 0000 niet nul, maar ... ?

De betekenis van de 9^e bit: 'overflow'

De 'consistentievoorwaarde' wordt slechts door één systeem vervuld: het *two's complement*.

Codering van gehele getallen: sign-magnitude (tekenbit)

Codering d.m.v. een tekenbit = *sign-magnitude*

– Methode:

- De MSB wordt gebruikt als tekenbit: 0 = + 1 = -
- De overblijvende bits stellen de waarde voor volgens het binaire stelsel
- De woordlengte dient vooraf bepaald te zijn

vb. **+**112 = **0**111 0000

- 83 = **1**101 0011

– Nadelen:

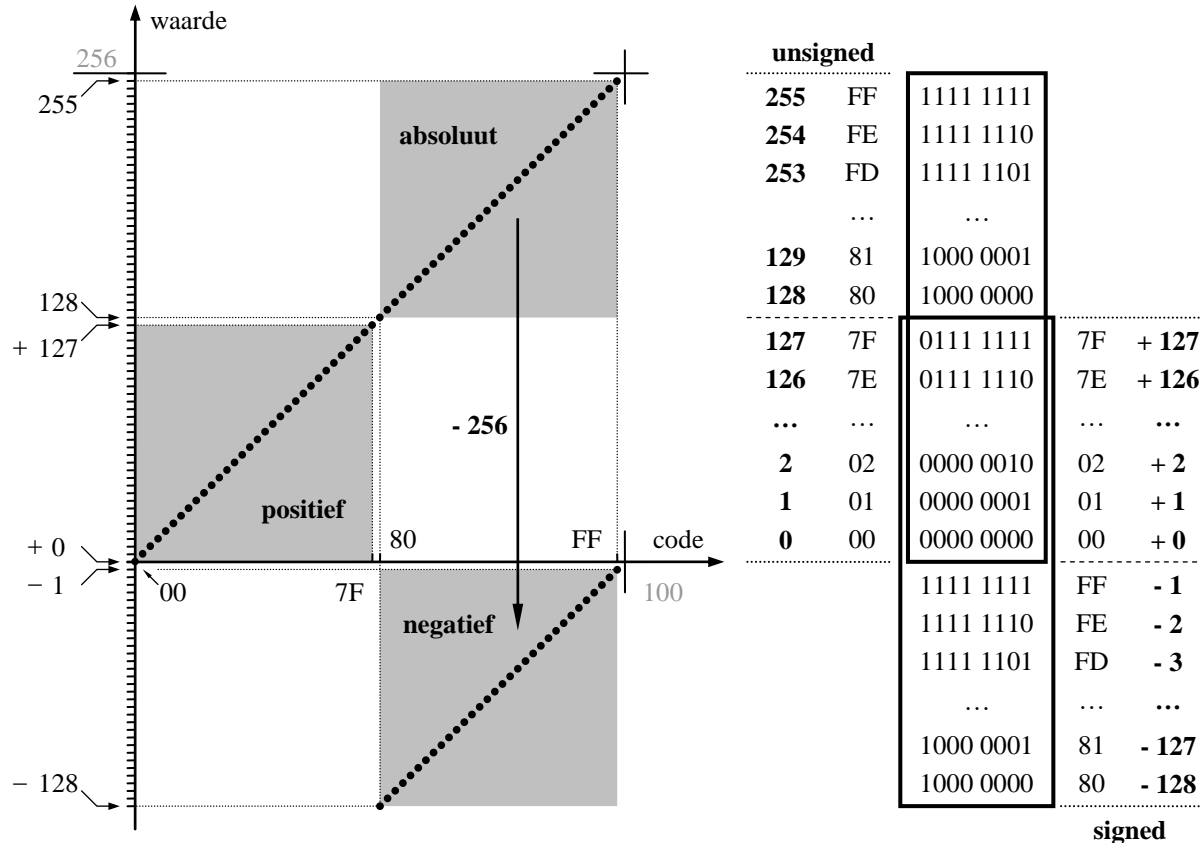
- Het getal 0 kan op 2 manieren gecodeerd worden
0000 0000 en 1000 000
- De binaire rekenregels gelden niet
 - De binaire som van een positief en negatief getal is fout
 $12 + (-6) = +6$
 $0000\ 1100 + 1000\ 0110 = \mathbf{1}001\ 0010 = -18$ (volgens deze coderingsmethode)
MSB = 1 dus een negatief getal
 - De som van hetzelfde positieve en negatieve getal is niet nul

Codering van gehele getallen: 2's complement

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

Codering van gehele getallen: 2's complement

De binaire codering van negatieve getallen in **2'S COMPLEMENT** (8-bit).

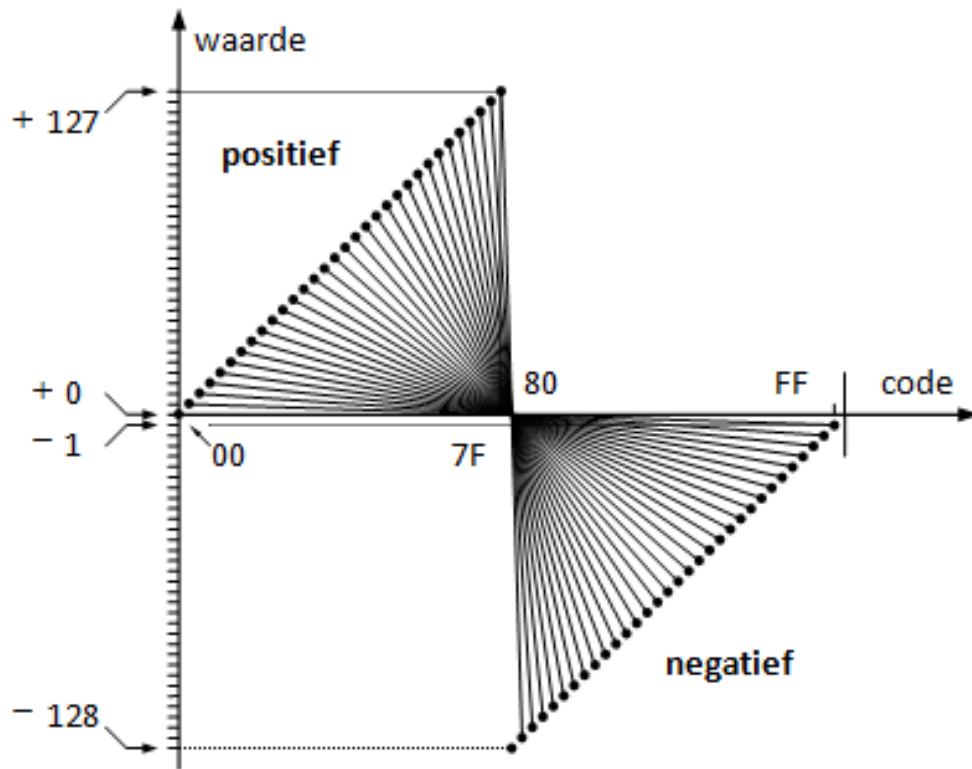


Bemerk: na 0000 0000 komt 1111 1111 (vergelijk met kilometerteller)



Codering van gehele getallen: 2's complement

- Berekenen van de tegengestelde waarde: compinc!

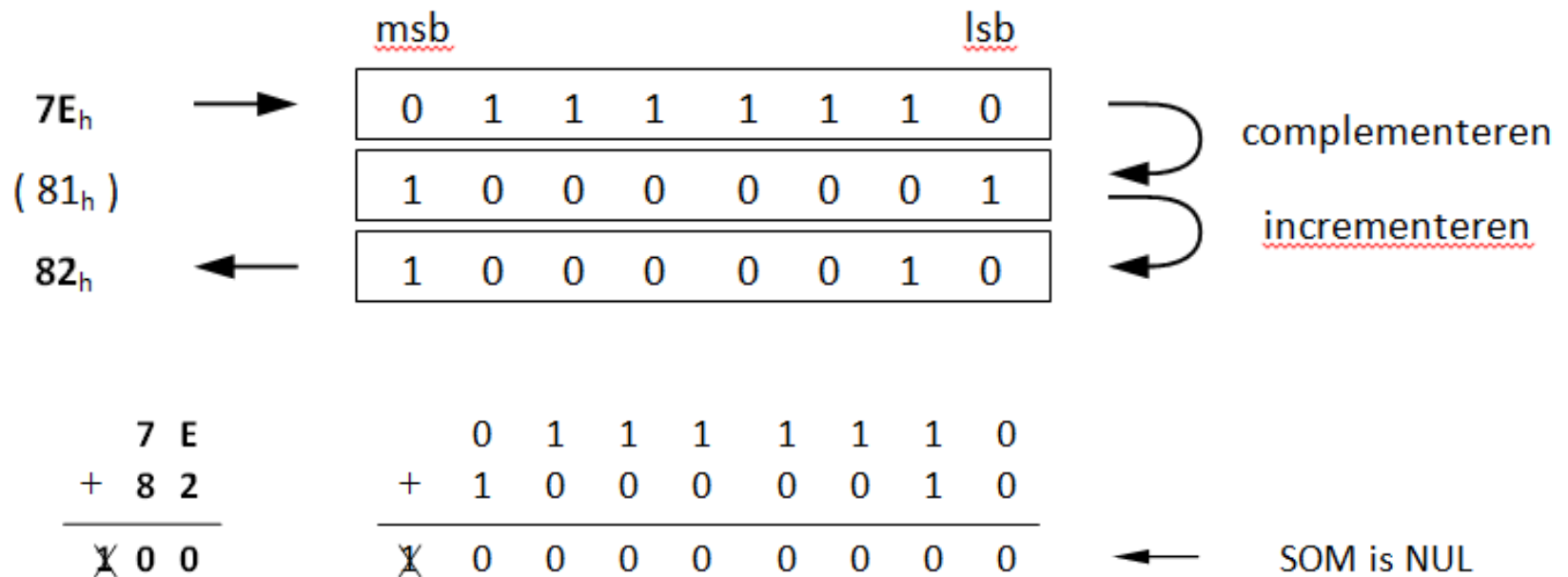


0111 1111	7F	+ 127	←
0111 1110	7E	+ 126	←
...	
0000 0010	02	+ 2	←
0000 0001	01	+ 1	←
0000 0000	00	+ 0	←
<hr/>			
1111 1111	FF	- 1	←
1111 1110	FE	- 2	←
1111 1101	FD	- 3	←
...	
1000 0001	81	- 127	←
1000 0000	80	- 128	←

- Bemerk: -128 heeft geen tegengestelde waarde (in het 8-bit systeem)

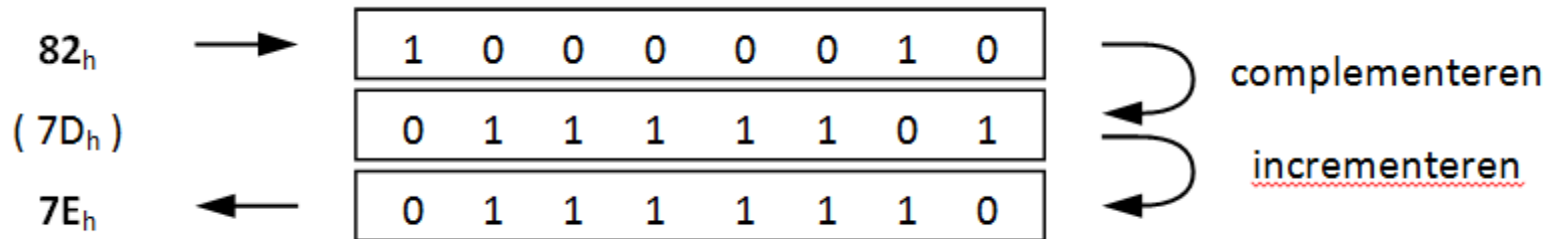
2's complement: "compinc" (van positief naar negatief)

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



Complementeren (van negatief naar positief)

- Omgekeerd voorbeeld: van neg \rightarrow pos



- = controle van vorige dia...
- Compinc regel werkt in twee richtingen: pos \leftrightarrow neg!
- Compinc mag je ook in hexvorm uitvoeren (zie bovenstaand voorbeeld)
- Merk op dat bij de bepaling van de getalwaarde in signed mode het minteken (b_7) niet terzijde gelaten wordt, maar meespeelt alsof het een volwaardige bit was! Ook in de hex notatie voert men de omzetbewerkingen uit alsof het betreffende getal gelegen is tussen 80₁₆ en FF₁₆ ...

Overzicht

Zie apart .pdf bestand!

Codering van getallen: rationale getallen (niet in handboek!)

- Elk rationaal getal R kan geschreven worden als

$$R = b_{n-1} \dots b_0 . b_{-1} \dots b_{-m} \quad (b_{n-1} \neq 0)$$

- Door het binaire punt n plaatsen naar links te verschuiven, kan men dit getal ook schrijven als

$$R = 0 . b_{n-1} \dots b_0 b_{-1} \dots b_{-m} \times 2^n$$

- Men noemt dit de **floating point** of glijdende komma (= zwevende, drijvende) voorstelling. Men heeft 2 binaire woorden nodig om op deze wijze een rationaal getal voor te stellen:

$$b_{n-1} \dots b_{-m} \quad = \text{de mantisse}$$

$$n = c_{p-1} \dots c_0 \quad = \text{de exponent}$$

- Zowel de mantisse als de exponent kunnen positief of negatief zijn!

Codes

- **Cijfercodes: de BCD-code, Excess-3 code, Gray-code**
 - + Eigenschappen van cijfercodes:
 - ✓ Gewogen code
 - ✓ Zelfcomplementerende code
 - ✓ Eén-wisselcode of cyclische code
 - ✓ Reflecterende code
- **Alfanumerieke codes**
- **Technische codes → zevensegment-code**

Cijfercodes: de BCD-code

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

Voorbeeld: $3196_{10} = 0011\ 0001\ 1001\ 0110$

- Van de 16 combinaties die met 4 bits mogelijk zijn, worden er in de BCD-code 6 niet gebruikt.
- De BCD-code is dus minder efficiënt dan de zuivere binaire codering.
- Ze wordt vooral toegepast bij digitale uitlezingen.

Cijfercodes: 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”)
- Voorbeeld

De standaard BCD-code is een gewogen code

→ “BCD₈₄₂₁” → de cijfers specificeren het gewicht van elke bit

Cijfercodes: zelfcomplementerende codes (*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
- Voorbeeld 1

De standaard BCD₈₄₂₁-code is niet zelfcomplementerend
→ Het 9-complement van 0011_2 (3_{10}) is $1100_2 \neq 6_{10}$

Cijfercodes: zelfcomplementerende codes (*self complementary code, self complementing code*)

- **Voorbeeld 2**

De excess-3 code (3-teveel code) van een decimaal cijfer vindt men door bij elk getal van de BCD₈₄₂₁ code binair 0011_2 (3_{10}) op te tellen (of +3 decimaal bij te tellen en dan naar het binaire stelsel om te zetten)

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

Cijfercodes: zelfcomplementerende codes (*self complementary code, self complementing code*)

- Voorbeeld 2 (vervolg)

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

- het complement van elk van deze codes levert het 9-complement op van het overeenkomstige decimale getal
→ wel een zelfcomplementerende code!
- geen gewogen code: het decimaal equivalent van een gecodeerd cijfer kan je niet terugvinden door aan elke bit een gewicht toe te kennen

Cijfercodes: één-wisselcodes of cyclische codes

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

Voorbeeld: Gray-code

Gray-code wordt ook een 'minimale verschil code' genoemd (tussen naburige waarden).

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

Cijfercodes: één-wisselcodes of cyclische codes

Voorbeeld: Gray-code (vervolg)

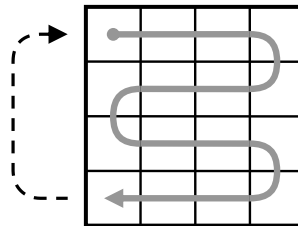
Bovendien een reflecterende code

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

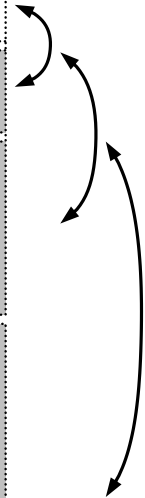
G	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

gray-code (4-bit)

		BA			
DC		00	01	11	10
		0	1	3	2
00		0	1	3	2
01		4	5	7	6
11		12	13	15	14
10		8	9	11	10



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



Cijfercodes: één-wisselcodes of cyclische codes

Voorbeeld: Gray-code (vervolg)

De Gray-code is geen gewogen code, dus niet praktisch voor rekenkundige bewerkingen.

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

Toepassingen:

1. Waar mechanische verplaatsingen worden omgezet naar een binaire code
2. In de telecommunicatie
3. Bij de Karnaughkaart

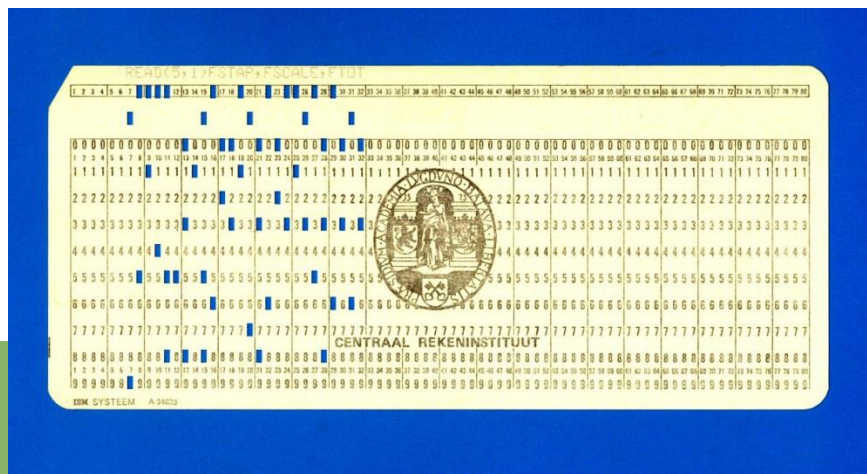
Alfanumerieke codes

Alfanumerieke gegevens = verzameling van letters (hoofd- en kleine letters), leestekens, cijfersymbolen en controlecodes voor datatransmissie

- ASCII-code = American Standard Code for Information Interchange
 - 7-bit code $\rightarrow 2^7 = 128$ bitcombinaties
 - 95 bitcombinaties voor leettertekens
 - Overige 33 zijn controletekens
 - 8^e bit bruikbaar als pariteitscontrole in datatransmissiesystemen
- ASCII-8 of extended ASCII
 - Uitbreiding van de 7-bit code
 - Sinds 1981
 - 8-bit code
 - Uitgebreider symbolenpallet
- EBCDIC = Extended Binary Coded Decimal Interchange
 - 8-bit code
 - Lezen als 'ebsidic'
 - Op vroegere computersystemen van IBM
 - Niet meer gebruikt

Alfanumerieke codes (vervolg)

- Unicode
 - 16-bit code → kan 65536 tekens bevatten
 - Ook leettertekens en karakters uit vreemde talen (Chinees, Japans, ...)
 - Ook Braille
- De hollerith-code
 - Toegepast op ponskaarten
 - Standaard ponskaart: 80 kolommen van 12 rijen
 - Ponskaart: cfr. principe draaiorgel



Alfanumerieke codes (vervolg)

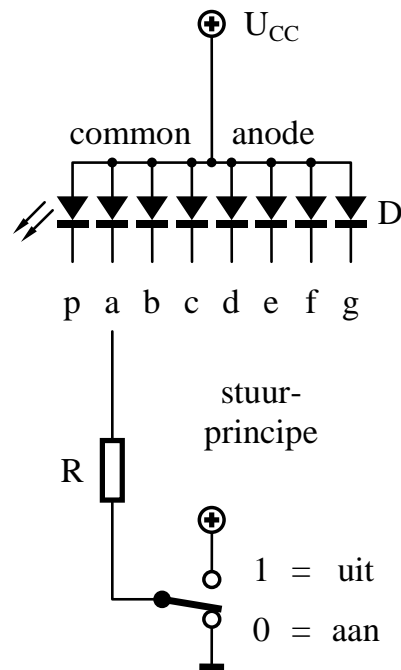
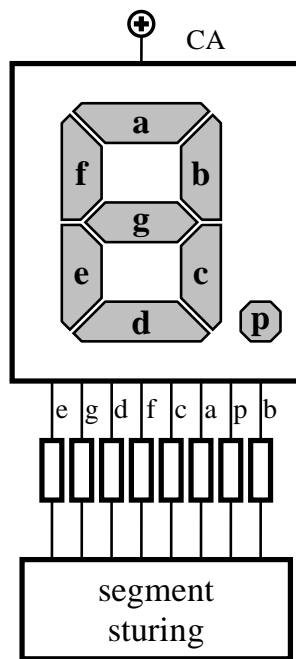
- Morsecode
 - Korte en lange piep (geluid), ook als lichtsignaal
 - Punten en strepen (geschreven)
 - Een geoefend oor (= telegrafist) haalt info uit ruis!
 - Vb. ... _ _ _ ... (= SOS)
 - Vb. ... _ _ ... (= SMS)
 - Heeft levens gered (scheepvaart)
 - In onbruik geraakt, maar nog steeds beoefend door radioamateurs

Technische codes

- = sets van bitpatronen waarmee elektronische schakelingen aangestuurd worden
- Eindeloos aantal toepassingen
- Enkele voorbeelden (zie handboek!):
 - Zevensegment-code (zie verder, gebruikt in lab!)
 - Dotmatrix code
 - Pixel-kleurcode
 - Andere codes: brailleschrift, weefgetouw van Jacquard, draaiorgels, muziekdozen, ...

Technische codes: zevensegment-code

- nummering leds
- aansturing
- common anode (= CA) ↔ common cathode (= CC)



zevensegment - display

	p	a	b	c	d	e	f	g	
0	1	0	0	0	0	0	0	1	81
1	1	1	0	0	1	1	1	1	CF
2	1	0	0	1	0	0	1	0	92
3	1	0	0	0	0	1	1	0	86
4	1	1	0	0	1	1	0	0	CC
5	1	0	1	0	0	1	0	0	A4
6	1	0	1	0	0	0	0	0	A0
7	1	0	0	0	1	1	1	1	8F
8	1	0	0	0	0	0	0	0	80
9	1	0	0	0	0	1	0	0	84
digit	segmentpatroon								code

Oefeningen

Opgave 1: zoek het 2's complement van de volgende bytes.

a) 0100 1100₂

b) 0000 0000₂

c) 0000 0001₂

d) 1101 0000₂

e) 0111 1111₂

f) 1000 0000₂

Oefeningen

Opgave 2: zet om van decimaal naar een 8-bits binaire string (= een byte) volgens 2 mogelijkheden

- met tekenbit (=sign-magnitude):

BIN

$-35_{10} \rightarrow$

- met 2's complement:

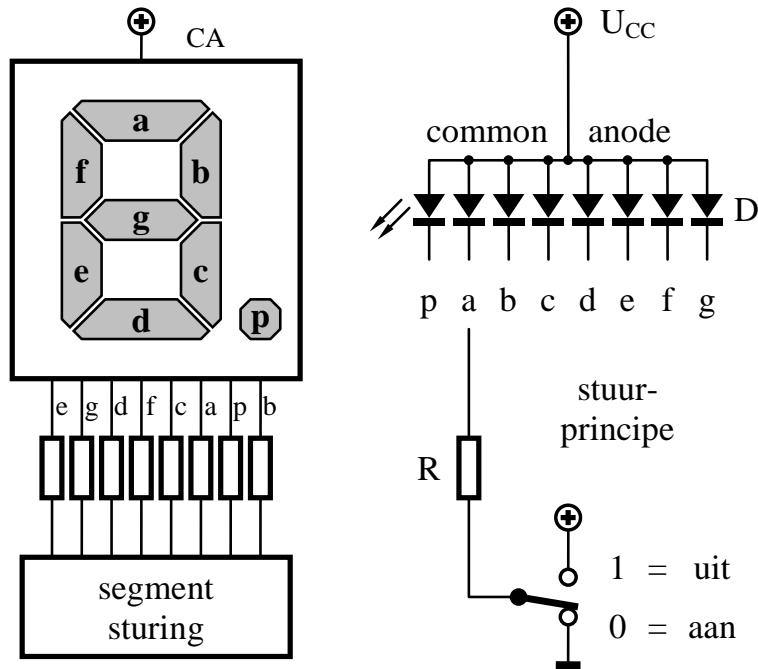
BIN

$-35_{10} \rightarrow$

Oefeningen

Opgave 3 (thuis): teken de aansluitingen van een common cathode (= CC) display (analoog als hieronder bij een CA).

0 1 2 3 4 5 6 7 8 9 A b C d E F



zevensegment - display

	p	a	b	c	d	e	f	g	
0	1	0	0	0	0	0	0	1	81
1	1	1	0	0	1	1	1	1	CF
2	1	0	0	1	0	0	1	0	92
3	1	0	0	0	0	1	1	0	86
4	1	1	0	0	1	1	0	0	CC
5	1	0	1	0	0	1	0	0	A4
6	1	0	1	0	0	0	0	0	A0
7	1	0	0	0	1	1	1	1	8F
8	1	0	0	0	0	0	0	0	80
9	1	0	0	0	0	1	0	0	84
digit	segmentpatroon								code

Sudietips en leerdoelen:

Je moet zeker in staat zijn om:

- Gehele getallen in *2's complement* en *sign-magnitude* voor te stellen
- Te verklaren hoe rationale getallen gecodeerd worden in het binaire stelsel
- Een decimaal getal in BCD-code voor te stellen
- De Gray-code, ... kunnen opstellen
- De zevensegment-code kunnen opstellen, uitleggen en toepassen (ook voor het labo)
- Uit te zoeken of een gegeven cijfercode gewogen, zelf-complementerend, cyclisch of reflecterend is
- Alle geziene codes
- Cijfercodes: de te kennen essentie staat in deze presentatie (in handboek meer uitgebreid)
- ...