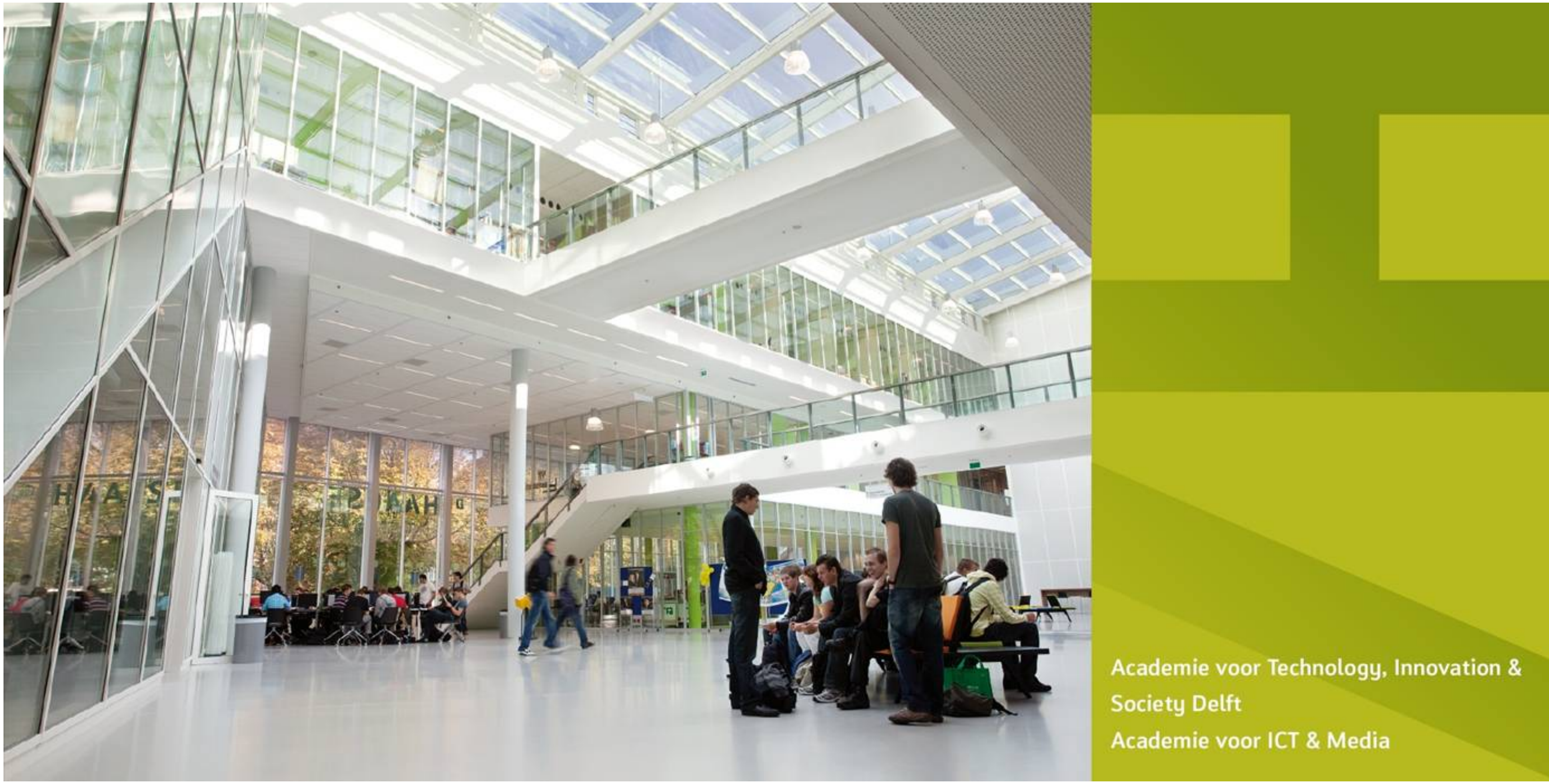


Inleiding Digitale Techniek

Week 2 – Binaire getallen, BCD, Gray, ASCII, 7-segment
Jesse op den Brouw
INLDIG/2018-2019

DE HAAGSE HOGESCHOOL

Talstelsels

- Wij mensen zijn opgegroeid met het rekenen in het *tientallig* of *decimaal* talstelsel, simpelweg omdat we tien vingers hebben.

- Het lag voor de hand dat de rekenapparaten die in de loop der jaren gebouwd werden ook rekenden met het decimale stelsel.

- De mechanische apparaten konden daar goed mee overweg maar met de elektronische versies had men toch veel problemen. Een geheugenelement dat tien verschillende cijfers moest kunnen onthouden was lastig te maken.

Talstelsels

- Veel makkelijker (en kleiner) te maken is een geheugenelement dat slechts twee verschillende cijfers kan onthouden, een zogenaamd binair geheugenelement. Vandaar dat alle rekenapparaten (computers) tegenwoordig rekenen in het *tweetallig* of *binaire* talstelsel.

- Soms worden ook wel een drietallig of ternair talstelsel gebruikt maar dit is uitzonderlijk.

- We bekijken eerst niet-negatieve getallen. Negatieve getallen worden later besproken.

Decimaal talstelsel

- Ons talstelsel is een zogenaamd *positioneel talstelsel*.
- Een getal bestaat uit de rij cijfers.
- Elk cijfer is één van de tien cijfers 0 t/m 9.
- Het grondtal is 10 (decimaal).
- De *positie* van het cijfer bepaalt het *gewicht*.
- Het gewicht is een macht van 10.

Decimaal talstelsel

- Neem als voorbeeld het getal 547:

- Dit kan gelezen worden als:

cijfer

$$547 = 5 \cdot 100 + 4 \cdot 10 + 7 \cdot 1$$

gewicht

- Of als machten van 10:

exponent

minst significant

$$547 = 5 \cdot 10^2 + 4 \cdot 10^1 + 7 \cdot 10^0$$

grondtal

Mooi woord voor grondtal is *radix*

Binair talstelsel

- Het binaire talstelsel bestaat slechts uit twee cijfers, 0 en 1.
- Het grondtal is 2.
- Een voorbeeld: 11011010
- Om aan te geven dat het om een binair getal gaat, wordt een index toegevoegd:

11011010_2

11011010_{BIN}

Binair naar decimaal

- Het omrekenen van een binair getal naar het decimale equivalent gaat eenvoudig.

- Bij elk binair cijfer hoort een gewicht: dat is een 2-macht. De som van alle cijfers vermenigvuldigd met de bijbehorende 2-machten levert het decimale equivalent.

$$\begin{aligned} 1101_2 &= 1\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 \\ &= 8 \quad + 4 \quad + 0 \quad + 1 \\ &= 13_{10} \end{aligned}$$

grondtal

$$\begin{aligned} 11011010_2 &= 1\cdot 2^7 + 1\cdot 2^6 + 1\cdot 2^4 + 1\cdot 2^3 + 1\cdot 2^1 \\ &= 128 + 64 + 16 + 8 + 2 \\ &= 218_{10} \end{aligned}$$

Decimaal naar decimaal

- Hoe zet je een decimaal getal om in een decimaal getal?
- Door herhaald delen door 10:

23561 ÷ 10	=	2356 rest	1
2356 ÷ 10	=	235 rest	6
235 ÷ 10	=	23 rest	5
23 ÷ 10	=	2 rest	3
2 ÷ 10	=	0 rest	2
<u>0</u>			

uitlezen

- Deze techniek toepassen voor omzetting naar binaire getallen.

Decimaal naar binair

- Als voorbeeld wordt 53_{10} omgezet:

minst significante cijfer

53 ÷ 2	=	26 rest 1
26 ÷ 2	=	13 rest 0
13 ÷ 2	=	6 rest 1
6 ÷ 2	=	3 rest 0
3 ÷ 2	=	1 rest 1
1 ÷ 2	=	0 rest 1
<u>0</u>		

uitlezen

meest significante cijfer

- Het binaire equivalent is dan 110101_2

Decimaal naar binair

- Voor het omzetten van een decimaal getal naar het binaire equivalent is herhaalde deling door 2 nodig:

  1) Ga uit van het decimale getal dat omgezet moet worden.
  2) Deel het gehele getal door 2.
  3) Schrijf de restwaarde (0 of 1) op en ga verder met het hele deel.
  4) Is het gehele deel groter dan 0? Dan naar stap 2.
  5) Lees het binaire getal af uit de restwaarden.

Er is ook nog een andere methode (“passen en meten”), die wordt niet besproken.

Opgaven

- Zet om van binair naar decimaal:

  1111_2 110011_2 01111111_2 1010010110100101_2

- Zet om van decimaal naar binair:

  25_{10} 10_{10} 128_{10} 27543_{10}

- Met tien vingers kan je ook van 0 t/m 1023 tellen door binaire codering te gebruiken. Een 0 is dan een gebogen vinger en een 1 is een gestrekte vinger. Waarom levert het getal 132_{10} toch problemen op in het dagelijks gebruik?

Hexadecimaal

- Over binaire getallen kunnen wat opmerkingen gemaakt worden:
- De lengte van binaire getallen wordt snel groter (factor 3,3 t.o.v. decimale getallen),
- Grote binaire getallen zijn lastig te overzien, gevoelig voor fouten,
- Binaire getallen worden vaak geschreven in blokken van 4 bits, gescheiden door een punt. Leidende nullen worden dan ook geschreven.
- Om binaire getallen goed te kunnen overzien wordt een ander talstelsel gebruikt: hexadecimaal.

Hexadecimaal

- Het hexadecimale talstelsel is een 16-tallig stelsel.
- Er zijn dus 16 “cijfers”.*)
- De eerste tien cijfers zijn identiek aan die van het decimale systeem.
- De overige zes cijfers worden weergegeven door de eerste 6 letters uit het alfabet.

*) een mooi woord hiervoor is “symbolen”.

Hexadecimaal

- Dus:

  0 t/m 9 hebben de waarden 0 t/m 9

  A heeft de waarde 10
  B heeft de waarde 11
  C heeft de waarde 12
  D heeft de waarde 13
  E heeft de waarde 14
  F heeft de waarde 15

Hexadecimaal naar decimaal

- Een voorbeeld van een hexadecimaal getal:

  $3FA5_{16}$

- Dit kan worden omgezet naar een decimaal getal:

  $3FA5_{16} = 3 \cdot 16^3 + 15 \cdot 16^2 + 10 \cdot 16^1 + 5 \cdot 16^0 = 16293_{10}$

- Merk op: hetzelfde systeem als decimaal en binair, nu echter met grondtal 16.

Decimaal naar hexadecimaal

- Omzetten van decimaal naar hexadecimaal gaat op eenzelfde wijze als omzetten van decimaal naar binair, alleen moet er nu gedeeld worden door 16:

$40796 \div 16$	=	2549 rest 12	$\rightarrow$	C	
$2549 \div 16$	=	159 rest 5			
$159 \div 16$	=	9 rest 15	$\rightarrow$	F	uitlezen
$9 \div 16$	=	0 rest 9			
$\underline{0}$					

- Het hexadecimale equivalent is dus $9F5C_{16}$

DE HAAGSE HOGESCHOOL

Tabel

- De eerste 17 getallen in decimaal, binair en hexadecimaal.

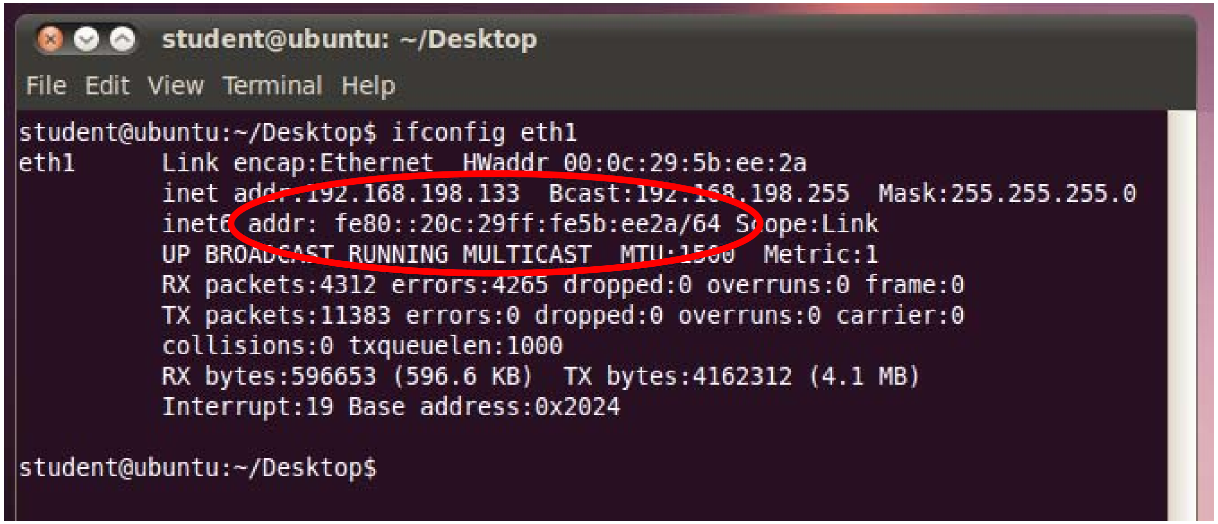
decimaal	binair	hexadecimaal
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10

Hexadecimaal in dagelijks gebruik?

- Tot nu toe wordt het hexadecimale stelsel voornamelijk gebruikt in de computer-industrie: hardware, low-level software.
- Maar binnenkort komt daar verandering in.
- Rond deze tijd zijn namelijk de IPv4 adressen “op”.
- Er is al een nieuw systeem: IPv6 met 128-bit adressen!
- De adressen worden genoteerd als hexadecimale getallen, dus met 32 hexadecimale cijfers per adres.

Hexadecimaal in dagelijks gebruik?

- Het IPv6-adres wordt opgesplitst in eenheden van 16 bits:
- Voorbeeld: 0123:4567:89AB:CDEF:0123:4567:89AB:CDEF
- Het hele IPv4-adresbereik past 2^{96} keer in het IPv6-adresbereik!



DE HAAGSE HOGESCHOOL

Hexadecimaal en binair

- Het hexadecimale systeem heeft als grondtal 16. Dit is een macht van 2 ($2^4 = 16$).

- Het is dus eenvoudig om een hexadecimaal getal om te zetten in een binair getal.

- Elk hexadecimaal cijfer kan worden geschreven met precies vier bits.

leidende nullen worden geschreven

$3F5A_{16} = 0011.1111.0101.1010_2$

punten ter bevordering van het lezen

DE HAAGSE HOGESCHOOL

Opgaven

- Zet om van hexadecimaal naar decimaal:

  $3FF_{16}$ $4D52_{16}$ $CAFE_{16}$

- Zet om van decimaal naar hexadecimaal:

  255_{10} 57005_{10} 32768_{10}

- Hoeveel unieke adressen zijn mogelijk met IPv6?

- Hoeveel unieke IPv6-adressen zijn er mogelijk per vierkante meter aardoppervlakte?

Binaire breuken

- Binaire breuken zijn eenvoudig te verwerken. Het werkt hetzelfde als met gehele getallen, alleen zijn de exponenten nu negatief.

$$0{,}1101_2 = 2^{-1} + 2^{-2} + 2^{-4} = 0{,}8125_{10}$$

- De komma geeft de scheiding aan tussen het gehele deel en de fractie.

- Binaire getallen kunnen een samenstel zijn van een geheel deel en een fractie:

$$011{,}1101_2 = 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-4} = 3{,}8125_{10}$$

Binaire breuken

- Van decimaal naar binair is lastiger, hier is herhaald vermenigvuldigen nodig.
- Een voorbeeld:

noteer de gehele getallen na vermenigvuldiging,

ga door met de fractie

0,8125 x 2	=	1,625	$\rightarrow$ 1
0,625 x 2	=	1,25	$\rightarrow$ 1
0,25 x 2	=	0,5	$\rightarrow$ 0
0,5 x 2	=	1,0	$\rightarrow$ 1
0			

uitlezen

stop indien 0

- Het antwoord is dan $0{,}1101_2$.

DE HAAGSE HOGESCHOOL

Opgaven

- Zet de volgende binaire breuken om in decimale breuken:

  $0{,}1001_2$ $0{,}11111111_2$

- Zet de volgende decimale breuken om in een binaire breuken:

  $0{,}75_{10}$ $0{,}3_{10}$ $0{,}8_{10}$

- Zet de volgende getallen om:

  $11{,}7_{10} \rightarrow$ hex $1F{,}3C_{16} \rightarrow$ dec

  $3F{,}EA_{16} \rightarrow$ bin $110110{,}110111_2 \rightarrow$ hex

Binaire breuken

- Niet alle eindige decimale breuken kunnen precies worden weergegegeven in het binaire systeem:

0,6 x 2 =	1,2	$\rightarrow$ 1	
0,2 x 2 =	0,4	$\rightarrow$ 0	uitlezen
0,4 x 2 =	0,8	$\rightarrow$ 0	
0,8 x 2 =	1,6	$\rightarrow$ 1	
0,6 ...			

- Het antwoord is dan $0{,}100110011001\ldots_2$.

- Dus $0{,}6_{10}$ *kan niet exact worden weergegeven in het binaire systeem!*

DE HAAGSE HOGESCHOOL

Afronden binaire breuken

- Afronden vindt plaats op basis van het eerste niet weer te geven cijfer na de komma:

  0: afronden naar beneden

  1: afronden naar boven

- $0{,}6_{10} = 0{,}1|\underline{0}\ldots = 0{,}1_2$ (na 1 bit)
  $0{,}6_{10} = 0{,}100|\underline{1}\ldots = 0{,}101_2$ (na 3 bits)
  $0{,}6_{10} = 0{,}10011001|\underline{1}\ldots = 0{,}10011010_2$ (na 8 bits)

- $0{,}6_{10} \approx 0{,}10011010_2 \rightarrow 0{,}10011010_2 = 0{,}6015625_{10}$ (+0,26%)

Noot: afronden volgens de Citogroep

DE HAAGSE HOGESCHOOL

Opgaven

- Hoewel zelden toegepast, komt in de praktijk ook het octale of achttallig talstelsel voor. Hierbij worden alleen de cijfers 0 t/m 7 gebruikt. Wat is de decimale waarde van 377_8 en 127_8? Wat is de binaire waarde van deze twee octale getallen?
- In een bepaald talstelsel blijkt het getal 41 gelijk te zijn aan het kwadraat van 5. In welk talstelsel staat dit getal geschreven?
- Op de planeet Mars is een getal ontdekt in een onbekend talstelsel. Na veel puzzelen komen de onderzoekers er achter dat het marsiaanse getal 234 overeenkomt met aardse getal 123_{10}. In welk talstelsel zijn de marsiaanse getallen genoteerd?

Benodigde aantal bits voor specifiek getal

- Om het decimale getal M weer te geven zijn tenminste *n* bits nodig zodanig dat:

$$M \leq 2^n - 1$$

- Na enig rekenwerk volgt dat:

$$n = \frac{\log(M+1)}{\log 2}$$

- Het resultaat moet worden afgerond op het laagste gehele getal dat groter is dan of gelijk is aan het oorspronkelijke getal:

$$n = \left\lceil \frac{\log(M+1)}{\log 2} \right\rceil$$

Benodigde aantal bits

- Om een decimaal getal van *m* cijfers weer te geven zijn tenminste *n* bits nodig zodanig dat:

$$10^m \leq 2^n$$

- Daaruit volgt dat:

$$n \geq \frac{m}{\log 2}$$

- Het resultaat moet worden afgerond op het laagste gehele getal dat groter is dan of gelijk is aan het oorspronkelijke getal:

$$n = \left\lceil \frac{m}{\log 2} \right\rceil$$

Bereik getallen

- In de onderstaande opsomming een aantal voorkomende bitbreedtes.

4 bits	$0 \leq g \leq 15$
8 bits	$0 \leq g \leq 255$
10 bits	$0 \leq g \leq 1.023$
12 bits	$0 \leq g \leq 4.095$
16 bits	$0 \leq g \leq 65.535$
24 bits	$0 \leq g \leq 16.777.215$
32 bits	$0 \leq g \leq 4.294.967.295$
64 bits	$0 \leq g \leq 18.446.744.073.709.551.615$
128 bits	$0 \leq g \leq 340.282.366.920.938.463.463.374.607.431.768.211.455$

Modulo rekenen

- In digitale systemen worden getallen opgeslagen met een eindig aantal bits.
- Dat betekent dat slechts een eindig aantal getallen kunnen worden weergegeven.
- Het bereik van een *n*-bits getal *g* is $0 \leq g \leq 2^n - 1$.
- De bijbehorende bitcombinaties zijn 0...0 t/m 1...1, beide met *n* bits.
- Het aantal combinaties is 2^n.

Modulo rekenen

- Getallen groter dan 1...1 (met *n* bits) worden *afgebeeld* in het bereik 0...0 t/m 1...1 (met *n* bits).

- Voorbeeld: de getallen 23_{10} en 7_{10} zijn identiek* als ze worden afgebeeld met 4 bits:

$$23_{10} \xrightarrow{\text{omzetting}} 10111_2 \xrightarrow{\text{met 4 bits}} 0111_2 \xrightarrow{\text{omzetting}} 7_{10}$$

- De getallen worden *modulo* 2^n afgebeeld. Dat houdt in dat de *rest na deling door* 2^n wordt opgeslagen.

- Voorbeeld: $39 \bmod 2^4 = 23 \bmod 2^4 = 7 \bmod 2^4 = 7$

*) congruent modulo

Opgaven

- Een ontwerper van een digitale schakeling heeft een getal opgeslagen in een 8 bits variabele. Bij het aanzetten van de schakeling wordt de variabele gereset (op 0 gezet). Wat is de waarde van deze variabele als hij 653 maal met 1 is verhoogd?

- Geef een algemene uitdrukking voor de verzameling getallen die opgeslagen in *n* bits het getal *M* oplevert, met $0 \leq M \leq 2^{n}-1$.

- Hoeveel bits zijn er nodig om een 8-cijferig decimaal getal op te slaan?

- Hoeveel bits zijn er nodig om de decimale getallen 365, 1024, 7987 en 176890 op te slaan?

BCD-code

- Het nadeel van binaire getallen is dat deze niet direct kunnen worden afgebeeld als decimale getallen, immers een computer werkt met binaire getallen.

- Bij het omzetten van een binair getal naar een decimaal getal moet herhaald gedeeld worden door $10_{10} = 1010_2$ en de restwaarde stelt dan één decimaal cijfer voor.

- Een digitale schakeling voor dit probleem kost veel poorten.

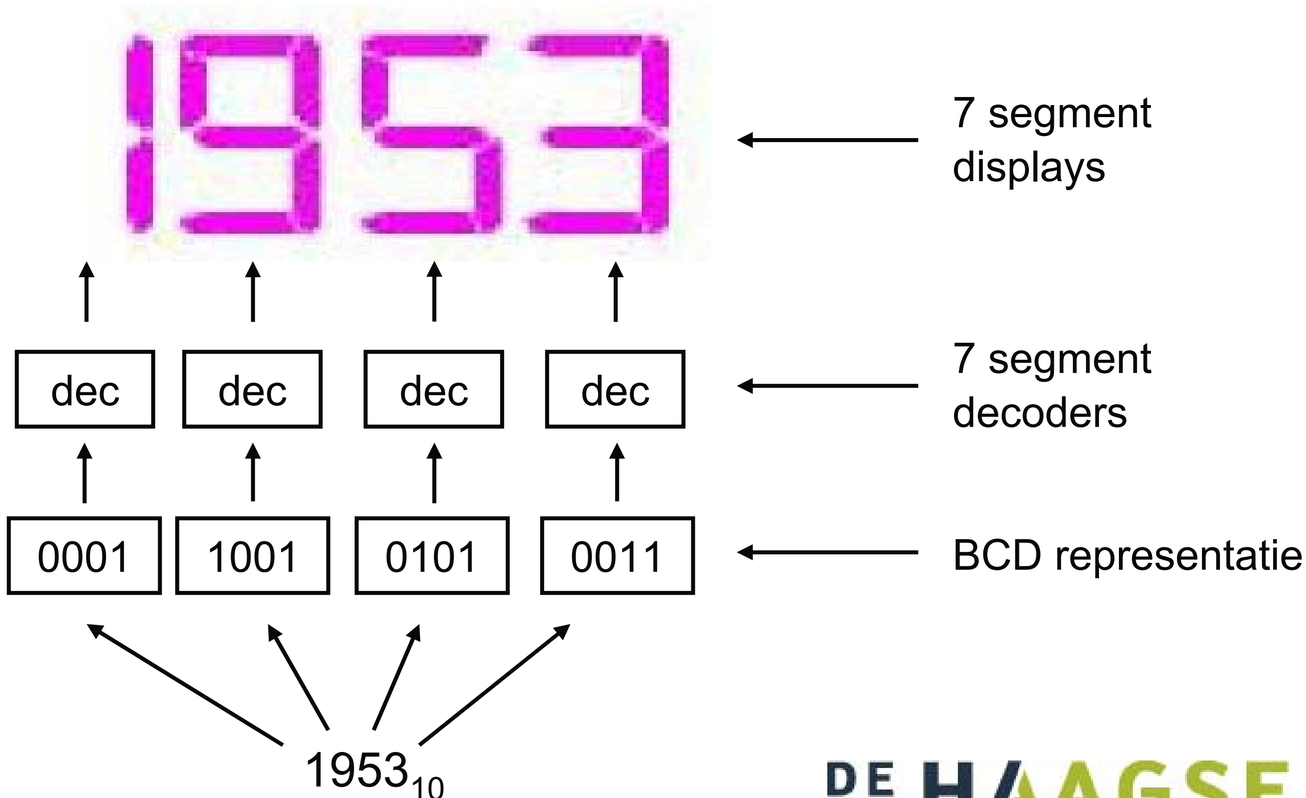
- Slimmer is om een codering te gebruiken waarbij een decimaal cijfer wordt opgeslagen in vier bits.

BCD-code

- Deze codering wordt *Binary Coded Decimal* genoemd.
- Elk BCD-cijfer bestaat uit vier bits (*nibble*).
- Van de 16 combinaties worden er 10 gebruikt (0 t/m 9) en 6 niet (10 t/m 15).
- Rekenschakelingen voor BCD-gecodeerde getallen zijn lastig.
- Het afbeelden op bijvoorbeeld 7-segment displays gaat daarentegen weer heel gemakkelijk.

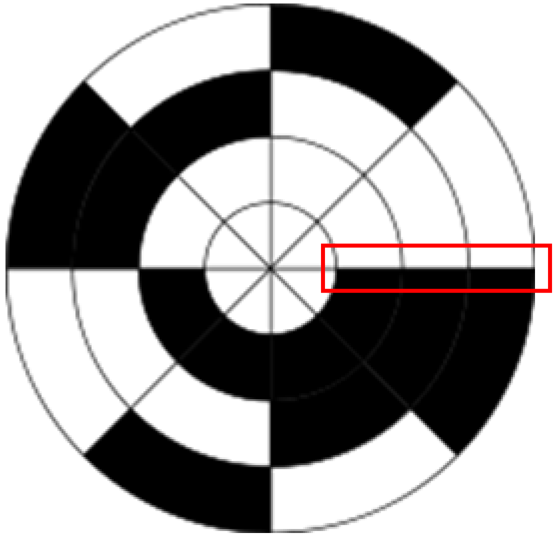
BCD-code

- BCD-code kan met behulp van een 7-segment decoder eenvoudig op een 7-segment display worden afgebeeld.



Gray-code

- In elektromechanische apparaten, zoals printers en kopieer-apparaten, is het soms nodig om de stand te weten van een roterend onderdeel.
- Dat kan gedaan worden met een zogenaamde codeerschijf.
- Dit levert problemen bij de randen van de zwarte en witte gebieden.



000 ?

010 ?

111 ?

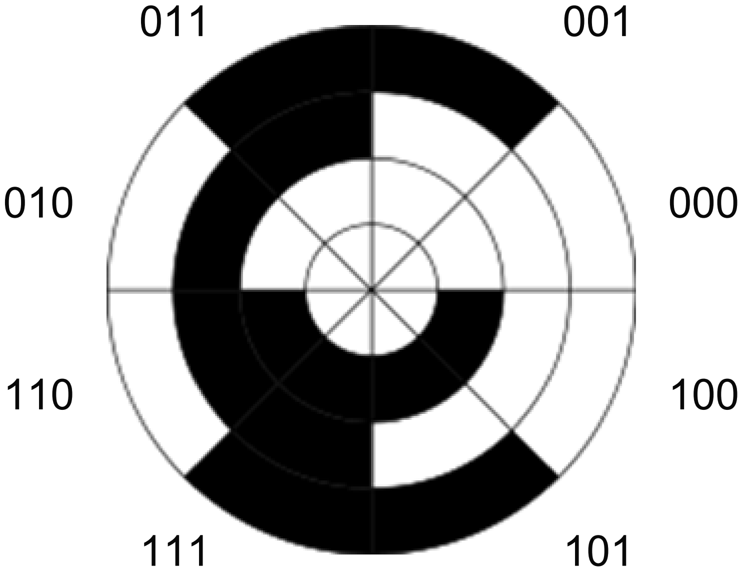
Gray-code

- Het probleem zit in het feit dat er meer dan één bitwisseling tegelijk moet plaatsvinden. Dat is praktisch nooit goed te regelen.

- Het probleem kan worden opgelost met de codeerschijf rechts.

- Hierbij wisselt steeds één bit per overgang.

- Dit wordt de *Gray-code* genoemd.



Gray-code

- De Gray-code is een zogenaamde *progressieve code*.
- Dit soort coderingen wordt veel toegepast bij toestands-toekenning (state assignment) bij asynchrone toestandsmachines.
- Hieronder de variant voor drie bits

decimaal	binair	Gray
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

DE HAAGSE HOGESCHOOL

ASCII-code

- Naast numerieke gegevens wisselen computers ook tekst uit. Dat moet op een gestandaardiseerde manier.
- De ASCII-code (1963, 7 bits) geeft de coderingen aan voor letters, cijfers en leestekens. Zo wordt de A gecodeerd als 41_{16} (65_{10}).
- Daarnaast wordt ook een aantal *besturingstekens* gecodeerd.
  - LF = $0A_{16}$, CR = $0D_{16}$, BS = 08_{16}

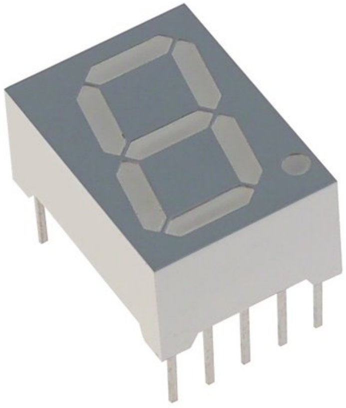
	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

Opgaven

- Hoeveel procent van de codecombinaties van een twee-cijferig BCD-getal kan nuttig worden gebruikt? En bij drie-cijferig?
- Ontwerp een schakeling die een (zuiver) 3-bits binaire code omzet in een 3-bits Gray-code. Hint: EXOR rules.
- Ontwerp een schakeling die een 3-bits Gray-code omzet in een (zuiver) 3-bits binaire code. EXOR rules again?
- Laat zien dat het eenvoudig is om in de ASCII-code hoofdletters te veranderen in kleine letters (en andersom).

7-segment display

- De 7-segment display*) is al jaren lang het middel om getallen af te beelden.
- De display bestaat uit zeven segmenten (meestal uitgevoerd als leds) en een punt (ook uitgevoerd als led).
- De zeven segmenten zijn zo gecon-strueerd dat ze samen de tien cijfers kunnen weergeven.
- Het is zelfs mogelijk een aantal letters weer te geven.



*) http://en.wikipedia.org/wiki/Seven-segment_display

7-segment display

- Om tien verschillende cijfers weer te kunnen geven zijn vier bits nodig:

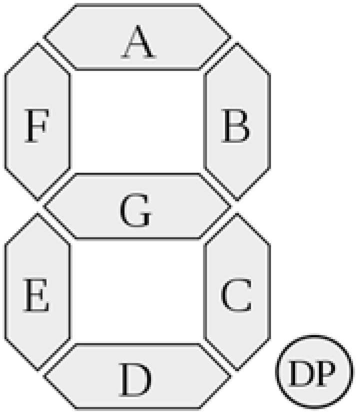
$$n = \lceil \log_2 10 \rceil = 4$$

- De bitcombinaties 0000 t/m 1001 worden hiervoor gebruikt.
- De bitcombinaties 1010 t/m 1111 worden niet gebruikt*).
- Dit komt precies overeen met één BCD-cijfer.

*) Voor deze combinaties kunnen ook letters worden gebruikt of worden de segmenten uitgeschakeld. Dit wordt *blanking* genoemd.

7-segment decoder

- Om cijfers weer te geven moeten de segmenten worden aangestuurd.
- Elk segment kan aan of uit staan, dus is een digitale schakeling te ontwerpen die dit doet.
- Dit is te realiseren met een schakeling die zeven (acht) uitgangen heeft.
- Hiervoor wordt aan de segmenten een letter toegekend, zie figuur.



7-segment decoder

- Vervolgens kunnen de tien cijfers worden geconstrueerd.

- Voor het segment A wordt dit op de volgende slides uitgewerkt.
- Aanname: led brandt bij logische 1.

7-segment decoder

- We stellen een waarheidstabel op met de ingangscombinaties 0000 t/m 1001.
- Aan elke ingangscombinatie kennen we een variabele y toe met indexnummer (y_0 t/m y_9).
- Bij elke ingangscombinatie wordt de waarde van segment A genoteerd.
- De combinaties 1010 t/m 1111 doen niet mee.

#	x_3	x_2	x_1	x_0	Y	A
0	0	0	0	0	y_0	1
1	0	0	0	1	y_1	0
2	0	0	1	0	y_2	1
3	0	0	1	1	y_3	1
4	0	1	0	0	y_4	0
5	0	1	0	1	y_5	1
6	0	1	1	0	y_6	1
7	0	1	1	1	y_7	1
8	1	0	0	0	y_8	1
9	1	0	0	1	y_9	1

DE HAAGSE HOGESCHOOL

7-segment decoder

- We bouwen de schakeling in twee delen op.
- Het eerste deel zorgt voor de logische schakelingen voor y_0 t/m y_9.
- Het tweede deel zorgt voor de logische schakeling voor segment A.
- Deze laatste wordt opgebouwd uit y_0 t/m y_9.

#	x_3	x_2	x_1	x_0	Y	A
0	0	0	0	0	y_0	1
1	0	0	0	1	y_1	0
2	0	0	1	0	y_2	1
3	0	0	1	1	y_3	1
4	0	1	0	0	y_4	0
5	0	1	0	1	y_5	1
6	0	1	1	0	y_6	1
7	0	1	1	1	y_7	1
8	1	0	0	0	y_8	1
9	1	0	0	1	y_9	1

DE HAAGSE HOGESCHOOL

7-segment decoder

- De functie voor combinatie 0000 is:

$$y_0 = \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_0}$$

- En voor 0001:

$$y_1 = \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0$$

- Tot en met 1001:

$$y_9 = x_3 \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0$$

#	x_3	x_2	x_1	x_0	Y	A
0	0	0	0	0	y_0	1
1	0	0	0	1	y_1	0
2	0	0	1	0	y_2	1
3	0	0	1	1	y_3	1
4	0	1	0	0	y_4	0
5	0	1	0	1	y_5	1
6	0	1	1	0	y_6	1
7	0	1	1	1	y_7	1
8	1	0	0	0	y_8	1
9	1	0	0	1	y_9	1

7-segment decoder

- Nu alle functies voor y_0 t/m y_9 bekend zijn, kan de functie voor segment A uitgewerkt worden.
- Segment A moet gaan branden voor elke ingangscombinatie waarvoor bij A in de waarheidstabel een logische 1 staat.
- Dat is dus bij y_0, y_2, y_3, y_5, y_6, y_7, y_8 en y_9. De functie wordt dan:

$$A = y_0 + y_2 + y_3 + y_5 + y_6 + y_7 + y_8 + y_9$$

#	x_3	x_2	x_1	x_0	Y	A
0	0	0	0	0	y_0	1
1	0	0	0	1	y_1	0
2	0	0	1	0	y_2	1
3	0	0	1	1	y_3	1
4	0	1	0	0	y_4	0
5	0	1	0	1	y_5	1
6	0	1	1	0	y_6	1
7	0	1	1	1	y_7	1
8	1	0	0	0	y_8	1
9	1	0	0	1	y_9	1

7-segment decoder

- Dit levert wel veel logica op:

  4x NOT, 8x AND4, 1x OR8

- Slimmer is om naar de 0-en te kijken. De schakeling is 0 bij y_1 en y_4. Dus:

$$A(0) = y_1 + y_4$$

- Om 1-en te krijgen met de functie geinverteerd worden:

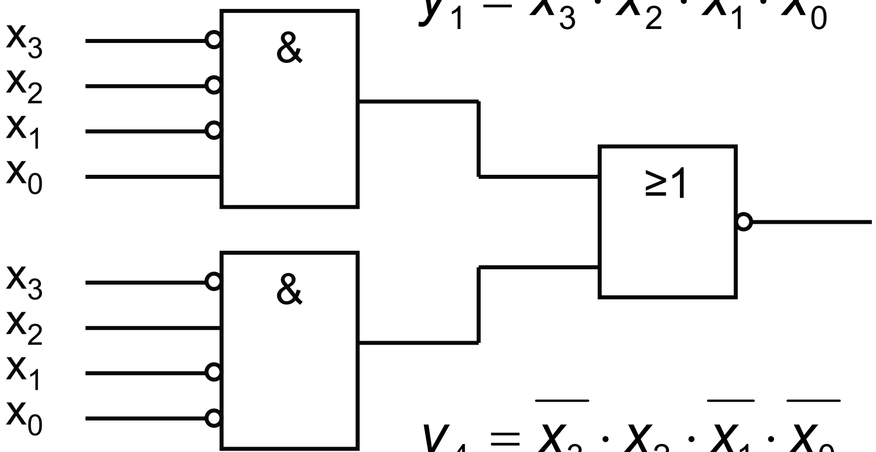
$$A = \overline{A(0)} = \overline{y_1 + y_4}$$

#	x_3	x_2	x_1	x_0	Y	A
0	0	0	0	0	y_0	1
1	0	0	0	1	y_1	0
2	0	0	1	0	y_2	1
3	0	0	1	1	y_3	1
4	0	1	0	0	y_4	0
5	0	1	0	1	y_5	1
6	0	1	1	0	y_6	1
7	0	1	1	1	y_7	1
8	1	0	0	0	y_8	1
9	1	0	0	1	y_9	1

7-segment decoder

- Het schema wordt dan:

$$y_1 = \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0$$



$$A = \overline{y_1 + y_4}$$

$$y_4 = \overline{x_3} \cdot x_2 \cdot \overline{x_1} \cdot \overline{x_0}$$

- NOT-poorten als geinverteerde ingangen.

7-segment display

- Een 4-bits code heeft in totaal 16 combinaties.
- Bij de BCD-code worden maar 10 combinaties gebruikt.
- De overige combinaties worden niet gebruikt bij het bepalen van de functies voor de segmenten.
- In de waarheidstabel wordt op deze plaatsen een *don't care* ingevuld.

#	x_3	x_2	x_1	x_0	A
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	-
11	1	0	1	1	-
12	1	1	0	0	-
13	1	1	0	1	-
14	1	1	1	0	-
15	1	1	1	1	-

Opgaven

- Met een 7-segment decoder kunnen de cijfers 0 t/m 9 worden gedecodeerd voor aansturing van een 7-segment display. De zes niet gebruikte combinaties kunnen toch zinvol worden gebruikt voor het afbeelden van A t/m F zodat ook hexadecimale cijfers kunnen worden afgebeeld. Hoe zouden die letters op en 7-segment display er uit zien?

- Ontwerp een schakeling voor segment B.

- Ontwerp een schakeling voor segment A indien don’t cares worden meegenomen.

Referenties

- De volgende boeken zijn gebruikt.

  Digitale techniek, van probleemstelling tot realisatie deel 1; A.P. Thijssen; 5e druk.

  Digital Design, Principles and Practices; J.F. Wakery; 3th Ed.

  Contemporary Logic Design; R.H. Katz; 2nd Ed.

  Dictaat hoofdstuk 2

- Internet:

  http://en.wikipedia.org/wiki/Seven-segment_display
  http://dandrake.livejournal.com/340063.html (IPv6 adr/m^2)
  http://en.wikipedia.org/wiki/ASCII
  http://nl.wikipedia.org/wiki/Afronden (afronden Citogroep)

Academie voor Technology, Innovation & Society Delft
Academie voor ICT & Media

De Haagse Hogeschool, Delft
+31-15-2606311
J.E.J.opdenBrouw@hhs.nl
www.dehaagsehogeschool.nl

DE HAAGSE HOGESCHOOL