Getallenrepresentatie

Processoren
17 november 2014

Wat is een getal?

- Getallen zitten in je hoofd.
- Een getallenrepresentatie is nodig om erover te praten / schrijven / ze op te slaan.
- Een getallenrepresentatie interpreteer je om zijn waarde te bepalen

Mogelijke getalrepresentaties

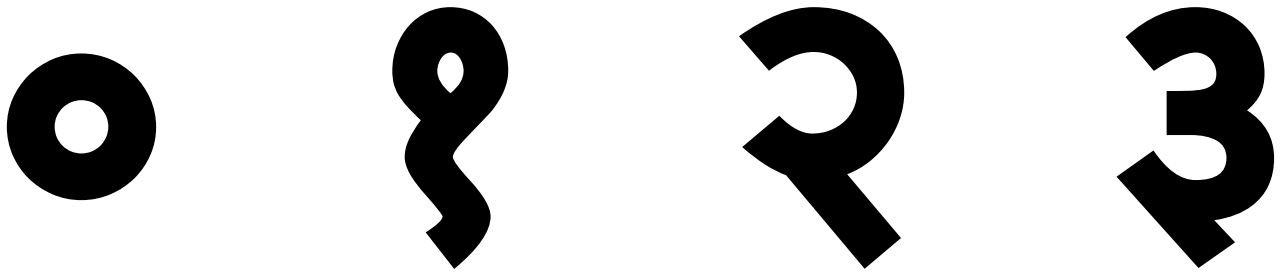
Mogelijke getalrepresentaties

MDCXIV

Mogelijke getalrepresentaties

א ב ג

Mogelijke getalrepresentaties



Mogelijke getalrepresentaties

0 1 2 3

Mogelijke getalrepresentaties

- standaard in het dagelijks leven: cijfers, (meestal) decimaal stelsel
- standaard in de computer: bitpatronen, altijd binair stelsel
- In beide gevallen: positionele notatie

Getalstelsels: decimaal

- algemeen schema voor getalrepresentatie:

$$\ldots d_2\ d_1\ d_0\ ,\ d_{-1}\ d_{-2}\ \ldots$$

- waarde van deze decimale representatie:

$$\ldots + 10^2 d_2 + 10^1 d_1 + 10^0 d_0 + 10^{-1} d_{-1} + 10^{-2} d_{-2} + \ldots$$

- Radix of grondtal = 10

Getalstelsels: binair

- algemeen schema voor getalrepresentatie:

$$\ldots d_2\ d_1\ d_0\ ,\ d_{-1}\ d_{-2}\ \ldots$$

- waarde van deze binaire representatie:

$$\ldots + 2^2 d_2 + 2^1 d_1 + 2^0 d_0 + 2^{-1} d_{-1} + 2^{-2} d_{-2} + \ldots$$

- Radix of grondtal = 2

Getalstelsels voor willekeurige grondtallen

- algemeen schema voor getalrepresentatie:

$$\ldots\ d_3\ d_2\ d_1\ d_0\ ,\ d_{-1}\ d_{-2}\ \ldots$$

- waarde van deze representatie:

$$\ldots + a^3d_3 + a^2d_2 + a^1d_1 + a^0d_0 + a^{-1}d_{-1} + a^{-2}d_{-2} + \ldots$$

- Radix of grondtal a

Veel gebruikte stelsels

naam	grondtal	gebruik
decimaal	10	dagelijks leven
binair	2	computergeheugen
octaal	8	spreken over computers
hexadecimaal	16	spreken over computers
sexagesimaal	60	minuten, seconden
vigesimaal	20	1 £ = 20 s (tot 1971)
duodecimaal	12	1 s = 12 d (tot 1971)

Omrekenen

- Algemene methode:
  delen door het gewenste grondtal, met rest
  - Voorbeeld: 21_{dec} in binair.
- ander stelsel → decimaal:
  tel de machten bij elkaar op
  - Voorbeeld: $1\ 1101\ 0001_{bin}$ in decimaal.
- binair ↔ octaal of hexadecimaal:
  per cijfer omzetten
  - Voorbeeld: $1\ 1101\ 0001_{bin}$ in octaal.

Rekenen met binaire getalrepresentatie

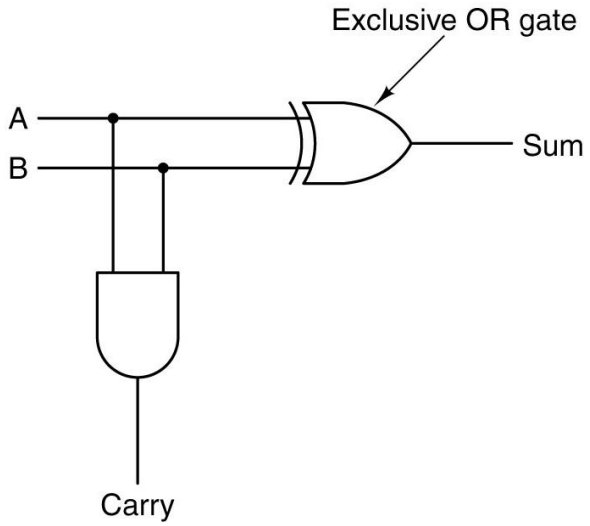
- bijna als schriftelijk optellen / aftrekken
  - voorbeeld: $81_{dec} + 54_{dec}$

- 0+0 = 0
  0+1 = 1+0= 1
  1+1 = 0 en 1 onthouden
  1+1+1 = 1 en 1 onthouden
- 1–0 = 1
  1–1 = 0–0= 0
  0–1 = 1 en 1 lenen
  0–1–1 = 0 en 1 lenen

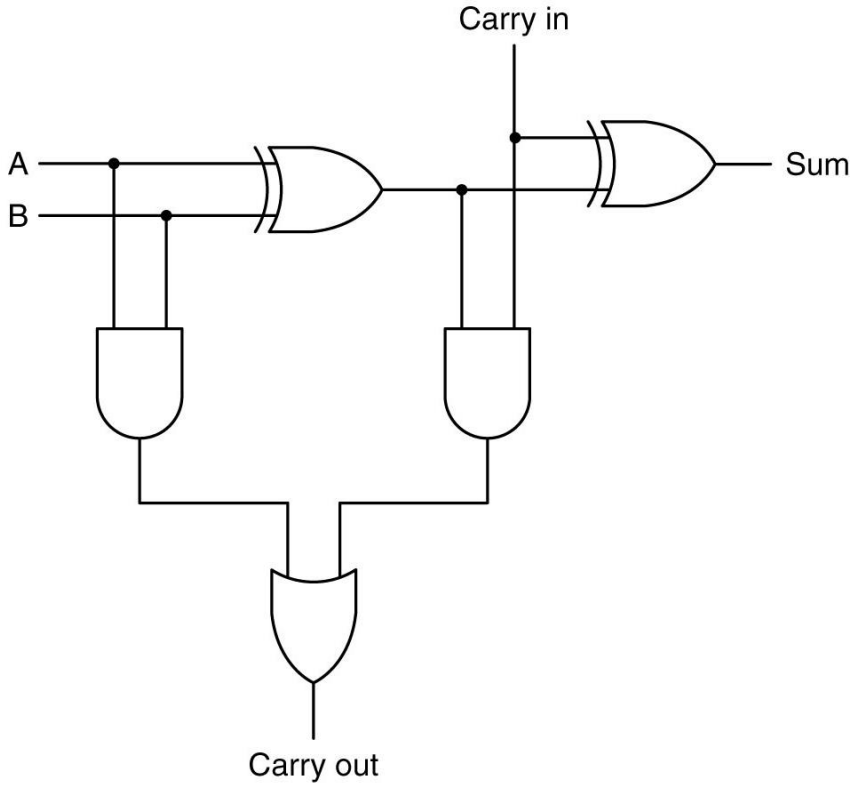
- Net zo: binair vermenigvudigen

Optel hardware: Adders

Half adder

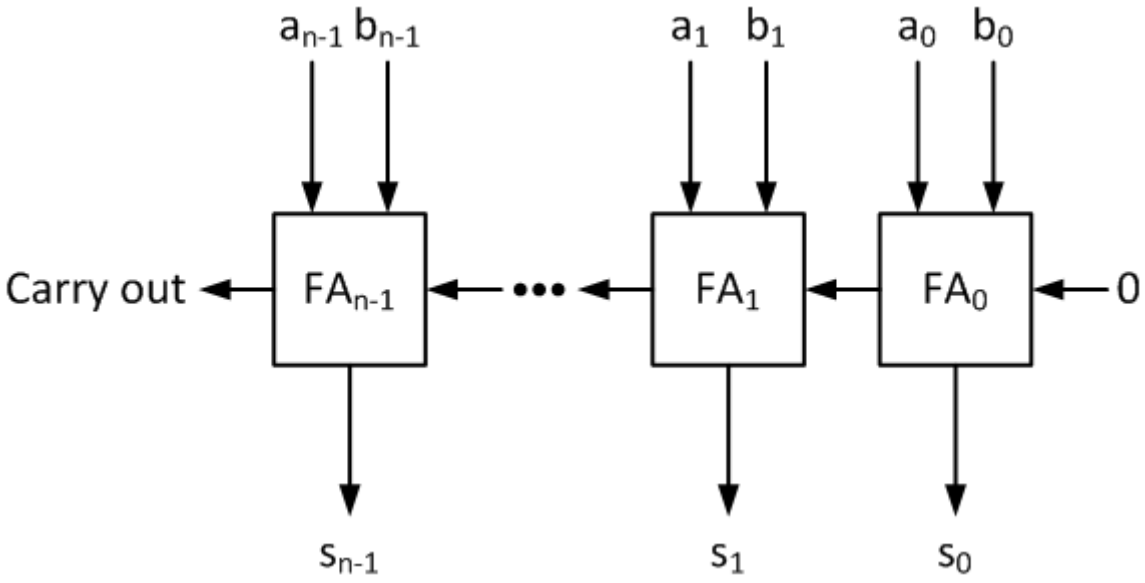


Full adder



Optel hardware: Adders 2

- Door n full adders te combineren krijgen we een opteller voor n bits getallen:



Beperkt geheugen/oppervlak

- er zijn oneindig veel getallen
- er zijn oneindig veel getallenrepresentaties

- geheugen van een computer is beperkt
  → beperkt aantal
  onderscheidbare representaties

- computers rekenen met een beperkt aantal bits
  bv. 32 of 64 bits.

Gelijkheid modulo *n*

- Twee getallen zijn **gelijk modulo *n*** als hun rest na deling door *n* hetzelfde is

  (dus als hun verschil een veelvoud van *n* is).

- voorbeelden:

$$60 \equiv 42 \pmod 6$$
$$13 \equiv 6 \pmod 7$$
$$405 \equiv 1429 \pmod{256}$$
$$-1 \equiv 255 \pmod{256}$$

Restklasse modulo *n*

- Een **restklasse modulo *n*** is een verzameling van alle getallen die gelijk modulo *n* zijn.

- voorbeelden:
  - $[0]_6 = \{ 0, 6, 42, 60, -6, \ldots \}$ restklasse modulo 6
  - $[1]_6 = \{ 1, 7, 13, -5, \ldots \}$ restklasse modulo 6
  - $[405]_{256} = \{ 149, 405, 661, 1429, -103, \ldots \}$ restklasse modulo 256

Operaties met restklassen

- operaties op gehele getallen uitbreiden naar restklassen:
  - kies een voorbeeldgetal/representant (:= een element uit de restklasse)
  - voer de operatie op het voorbeeldgetal uit
  - neem de restklasse van het resultaat
- vereist bewijs dat de resultaat-restklasse niet van het voorbeeldgetal afhangt

Bijna hetzelfde woord

- getallenrepresentatie:
  systeem van waarneembare symbolen
  (om over getallen te communiceren)
  - voorbeeld: א, ב, ג

- representant van een restklasse:
  voorbeeldgetal uit de restklasse
  - voorbeeld: –1 is een representant van
    $[255]_{256} = \{ -1, -257, 255, 511, 767, \ldots \}$

Optelling met restklassen

- restklassen modulo 6:
  { 0, 6, 12, ... } + { 1, 7, 13, ... } = ?
  - kies voorbeeldgetallen, b.v. 12 en 13
  - tel de voorbeeldgetallen op; resultaat is 25
  - neem de restklasse $[25]_6 = [1]_6$
  - Dus: $[0]_6 + [1]_6 = [1]_6$
  - Nog te bewijzen: andere voorbeeldgetallen leveren dezelfde resultaat-restklasse op ✓

een verzameling
van verzamelingen

Restklassenring

- $\mathbb{Z}/n\mathbb{Z}$:= verzameling van restklassen modulo n
- b.v. $\mathbb{Z}/6\mathbb{Z} = \{ [0]_6, [1]_6, \ldots, [5]_6 \}$

- Een **ring** is een verzameling met operaties +, – en ×, die aan bepaalde axioma's voldoet
- Een **lichaam** is een ring met inverse.
- $\mathbb{Z}/n\mathbb{Z}$ is een ring (en soms een lichaam)

Voorbeeld: $\mathbb{Z}/1000\mathbb{Z}$

- $[0]_{1000} = \{ 0, 1000, 2000, -1000, -2000, \ldots \}$
- $[501]_{1000} + [630]_{1000} =$

- $[27]_{1000} - [13]_{1000} =$
- $[75]_{1000} \cdot [821]_{1000} =$

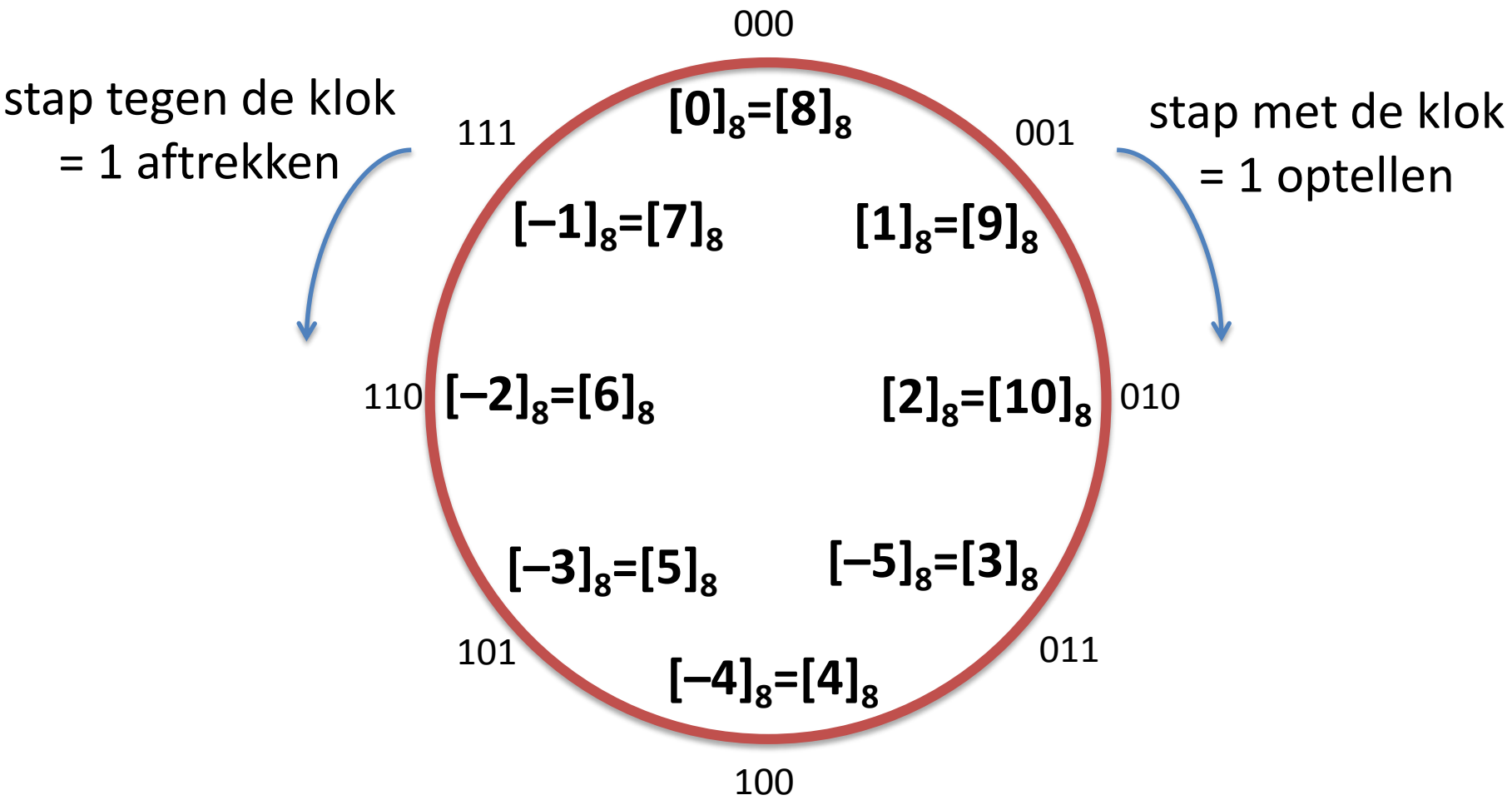
- $[460]_{1000} / [52]_{1000} =$

Hoe rekenen computers?

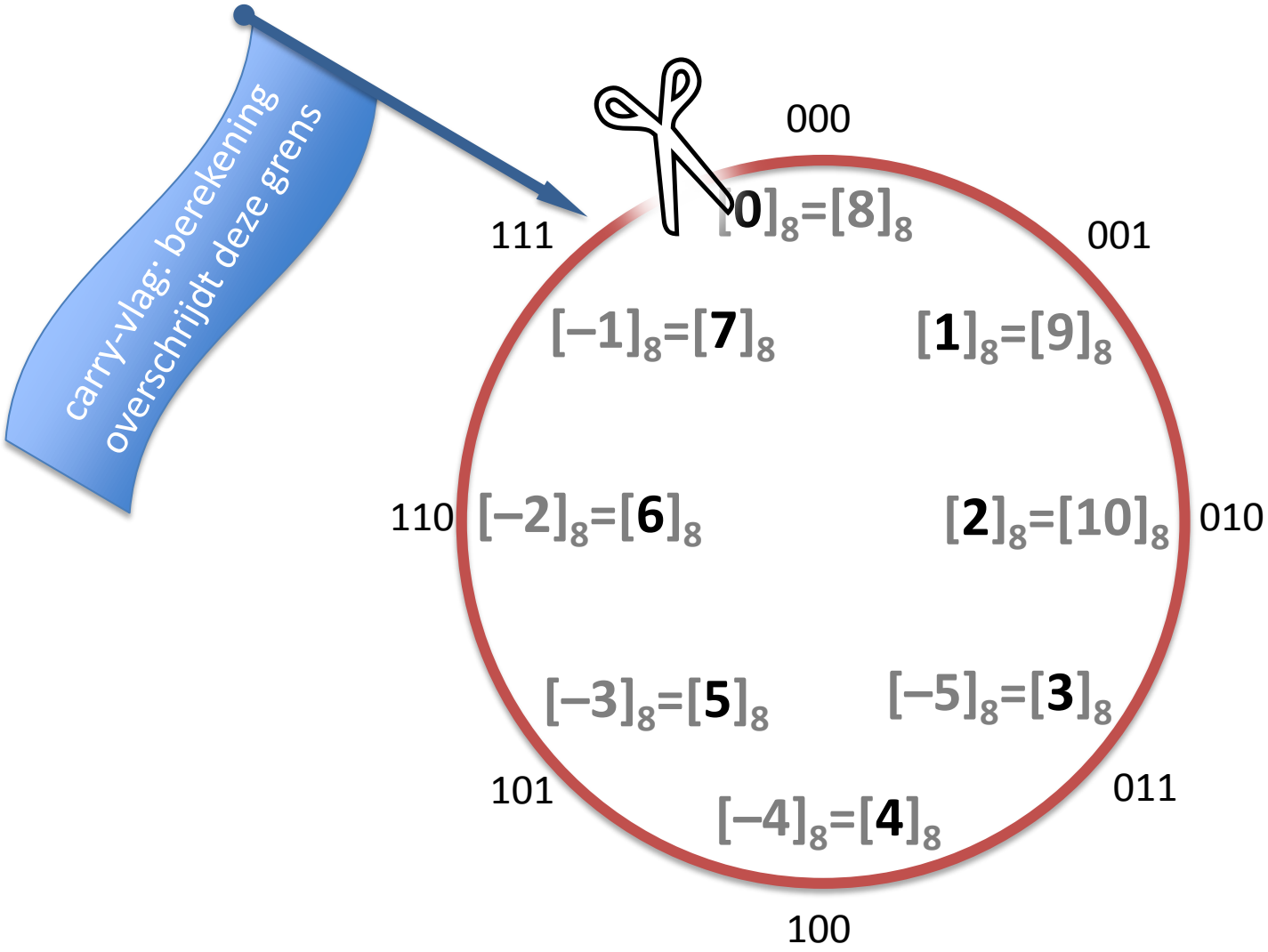
- restklassen modulo 2^8, 2^{16}, 2^{32} of 2^{64}

- aanpassingen om het meer te laten lijken op rekenen met kleine getallen
  - niet-negatieve getallen
  - twee-complement representatie (ook negatieve getallen)
  - vlaggen: carry, overflow (later)

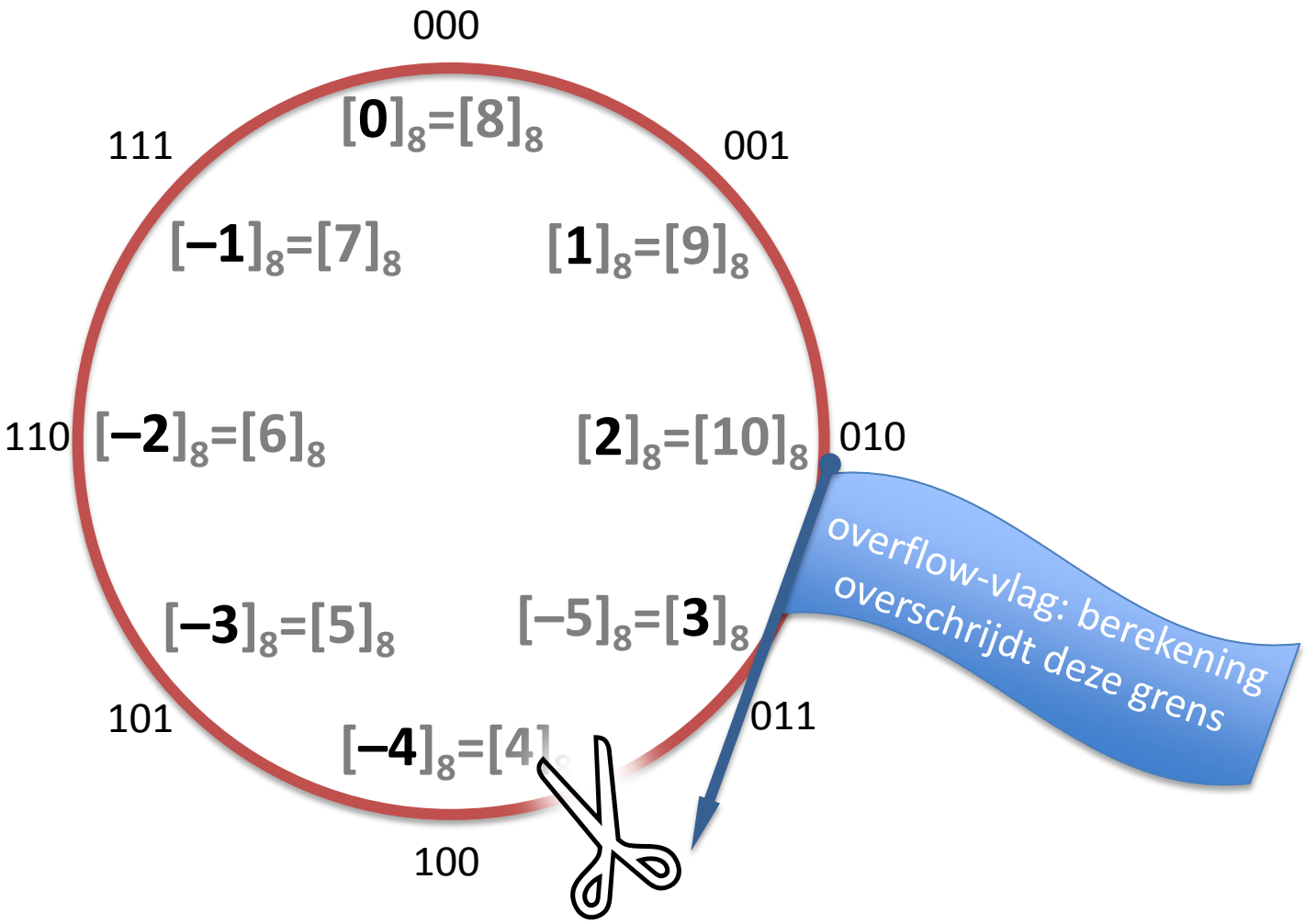
Cirkel van bitpatronen



~~Cirkel van~~ niet-negatieve getallen



~~Cirkel van~~ twee-complement



Twee-complement representatie

- *k* bits voor getallenrepresentatie beschikbaar
  - als gewone (unsigned) representatie: 0 ... $2^k - 1$
  - als twee-complement (signed) representatie: eerste helft als bij de unsigned representatie, tweede helft voor negatieve getallen

    in plaats van 2^{k-1} ... $2^k - 1$: -2^{k-1} ... -1
  - bitpatronen met hoogste bit 1 zijn de negatieve getallen, die met een 0 de positieve getallen en 0

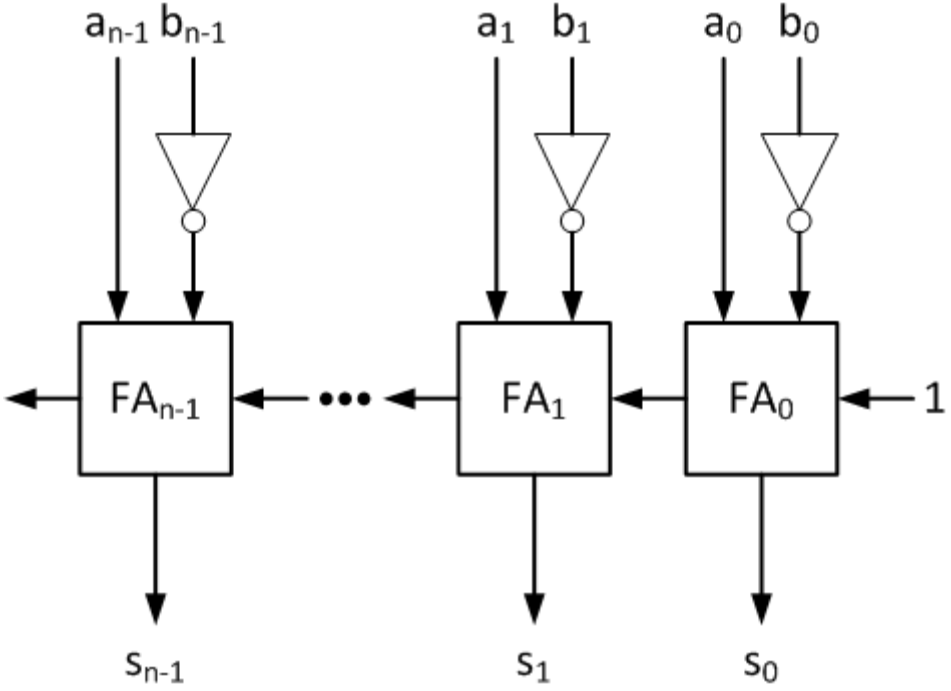
Twee-complement representatie 2

- Hoe vind je de twee-complement representatie van een negatief getal?
  - Bepaal de unsigned representatie van zijn inverse.
  - Inverteer alle bits.
  - Tel er 1 bij op (Vergeet die laatste stap niet).

- Voorbeeld (8 bits): 57_{dec} is 00111001_{bin}
  De 2-complement representatie van -57_{dec} is dan 11000110 + 1 oftewel 11000111_{bin}

Een n-bit subtractor

- Dit idee is ook te gebruiken om een n-bit aftrekschakeling mee te maken:



Voordelen van twee-complement

- Eenvoudig rekenen voor de computer
  - optellen en aftrekken (in de cirkel)
    werkt net als bij gewone representatie
  - eenvoudige test op negatief getal
    (daarom –4 en niet 4)
  - slechts één representatie van 0

  - voorbeelden

Optelling in de computer

- *k*-bit-patroon heeft drie betekenissen:
  - restklasse modulo 2^k
  - niet-negatief getal $\{ 0, \ldots, 2^k-1 \}$
  - getal in twee-complement $\{ -2^{k-1}, \ldots, 2^{k-1}-1 \}$

- optelling werkt correct met restklassen
- soms is resultaat incorrect voor andere betekenissen

opteltabel
voor restklassen

+	$[0]_8$	$[1]_8$	$[2]_8$	$[3]_8$	$[4]_8$	$[5]_8$	$[6]_8$	$[7]_8$
$[0]_8$	$[0]_8$	$[1]_8$	$[2]_8$	$[3]_8$	$[4]_8$	$[5]_8$	$[6]_8$	$[7]_8$
$[1]_8$	$[1]_8$	$[2]_8$	$[3]_8$	$[4]_8$	$[5]_8$	$[6]_8$	$[7]_8$	$[0]_8$
$[2]_8$	$[2]_8$	$[3]_8$	$[4]_8$	$[5]_8$	$[6]_8$	$[7]_8$	$[0]_8$	$[1]_8$
$[3]_8$	$[3]_8$	$[4]_8$	$[5]_8$	$[6]_8$	$[7]_8$	$[0]_8$	$[1]_8$	$[2]_8$
$[4]_8$	$[4]_8$	$[5]_8$	$[6]_8$	$[7]_8$	$[0]_8$	$[1]_8$	$[2]_8$	$[3]_8$
$[5]_8$	$[5]_8$	$[6]_8$	$[7]_8$	$[0]_8$	$[1]_8$	$[2]_8$	$[3]_8$	$[4]_8$
$[6]_8$	$[6]_8$	$[7]_8$	$[0]_8$	$[1]_8$	$[2]_8$	$[3]_8$	$[4]_8$	$[5]_8$
$[7]_8$	$[7]_8$	$[0]_8$	$[1]_8$	$[2]_8$	$[3]_8$	$[4]_8$	$[5]_8$	$[6]_8$

opteltabel
voor bitpatronen

+	000	001	010	011	100	101	110	111
000	000	001	010	011	100	101	110	111
001	001	010	011	100	101	110	111	000
010	010	011	100	101	110	111	000	001
011	011	100	101	110	111	000	001	010
100	100	101	110	111	000	001	010	011
101	101	110	111	000	001	010	011	100
110	110	111	000	001	010	011	100	101
111	111	000	001	010	011	100	101	110

opteltabel
voor niet-negatieve getallen

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

opteltabel
voor niet-negatieve getallen

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

opteltabel
voor getallen in twee-complement

+	0	1	2	3	–4	–3	–2	–1
0	0	1	2	3	–4	–3	–2	–1
1	1	2	3	–4	–3	–2	–1	0
2	2	3	–4	–3	–2	–1	0	1
3	3	–4	–3	–2	–1	0	1	2
–4	–4	–3	–2	–1	0	1	2	3
–3	–3	–2	–1	0	1	2	3	–4
–2	–2	–1	0	1	2	3	–4	–3
–1	–1	0	1	2	3	–4	–3	–2

opteltabel
voor getallen in twee-complement

	0	1	2	3	–4	–3	–2	–1
0	0	1	2	3	–4	–3	–2	–1
1	1	2	3	–4	–3	–2	–1	0
2	2	3	–4	–3	–2	–1	0	1
3	3	–4	–3	–2	–1	0	1	2
–4	–4	–3	–2	–1	0	1	2	3
–3	–3	–2	–1	0	1	2	3	–4
–2	–2	–1	0	1	2	3	–4	–3
–1	–1	0	1	2	3	–4	–3	–2

Vermenigvuldiging

- vermenigvuldiging van restklassen: eenduidig

- vermenigvuldiging van getallen $\in\{ 0, 1, ..., 2^k-1 \}$ levert een getal $\in\{ 0, 1, ..., 2^{2k}-2^{k+1}+1 \}$

- vermenigvuldiging van getallen $\in\{ -2^{k-1}, ..., 2^{k-1}-1 \}$ levert een getal $\in\{ -2^{2k-2}+2^{k-1}, ..., 2^{2k-2} \}$

  → vaak resultaat te groot (carry/overflow)

Vermenigvuldiging zonder overflow

- in de praktijk: multiplicatie van *k*-bits-getallen produceert bitpatroon met 2*k* bits
- niet-negatief of twee-complement maakt dan wel verschil
- b.v. 8086-assembly:
  - MUL CL AX := AL · CL (niet-negatief)
  - IMUL CL AX := AL · CL (twee-complement)

Deling

- computer rekent niet met restklassen
- (gehele) deling met rest

Samenvatting

- een representatie is nodig om over getallen te praten
- binaire getallen (representatie)
- rekenregels voor binaire getallen
- beperkt geheugen → restklassen
- aanpassing voor negatieve getallen:
  - twee-complement representatie