Processoren

Marc Seutter & David N. Jansen

10 November 2014

Leerdoelen

- Inzicht krijgen in de opbouw van de hardware van een computer en de instructies van een processor.

- je construeert een (eenvoudige) processor
- je leert (korte) assembly-programma's schrijven
- je kunt uitleggen:
  beginselen van computer-organisatie;
  relatie tussen organisatie en instructies;
  performance verbetering.

Werkvorm

- Hoorcollege op maandagmiddag
  - Lees het boek bij het betreffende college
- Werkcollege op dinsdag-, woensdag- of vrijdagmiddag
  - Wekelijkse huiswerkopgaven: electronisch inleveren op maandag
  - voor- en nabespreken in het werkcollege.

Studentassistenten en werkcolleges

Drie studentassistenten:

- Gerben van der Lubbe, gerben.vanderlubbe@student.ru.nl
- Jordi Riemens, jordi.riemens@student.ru.nl
- Niek Janssen, niek.janssen01@gmail.com

Werkcolleges:

grp2	dinsdag	15:30-17:30	GN6	Niek
grp1	woensdag	13:30-15:30	HG00.068	Gerben
grp3	vrijdag	15:30-17:30	HG00.071/GN1	Jordi

Huiswerkopgaven

- De webpagina van dit vak is te vinden op:
  - https://lab.cs.ru.nl/algemeen/Processoren

- In het rooster vind je per week de slides van het college en de huiswerkopgave van die week met de uiterste inleverdatum.

- Werk zoveel mogelijk als paren. Dit geldt ook voor het inleveren van je practicum-opgave.

Huiswerk inleveren per email

- Degene van de twee wiens achternaam het eerste in het alfabet voorkomt, bepaalt bij welke studentassistent je je uitwerking inlevert en wie je bij vragen kunt aanspreken
  - A t/m C: Jordi Riemens, jordi.riemens@student.ru.nl
  - D t/m K: Niek Janssen, niek.janssen01@gmail.com
  - L t/m Z: Gerben van der Lubbe, gerben.vanderlubbe@student.ru.nl
- We ontvangen jullie uitwerking graag als pdf of platte tekst (dus geen MsWord), in een email met subject: [Proc] Week1, etc.

Beoordeling

- Het cijfer van het vak wordt bepaald door het afgeronde gemiddelde van het resultaat voor de practicum-opdracht en het schriftelijke tentamen mits voor dat laatste tenminste een 5 is gehaald.

- Voor dit schriftelijke tentamencijfer kan middels de huiswerkopgaven een bonus verkregen worden.

Bonusregeling voor huiswerk

- Er komen 7 in te leveren huiswerkopgaven.

- Als je tenminste 5 van de 7 inlevert, waarbij elke ingeleverde huiswerkopgave voor tenminste 60% goed was, krijg je een bonus van +1 op het (schriftelijk) tentamencijfer.

Tijdsbesteding

Hoorcolleges + werkcolleges	$8 \times 2 + 7 \times 2 =$	30
Lezen	$8 \times 1 =$	8
Huiswerk	$7 \times 1 =$	7
Practicumopdracht	$4 \times 8 =$	32
Tentamen	$5 + 2 =$	7
Totaal (3 ec)	$3 \times 28 =$	84

Overzicht van de stof

- In week 1 t/m 3 bouwen we (bottom-up) de kennis op over de hardware in een processor:
  - Booleaanse algebra, poorten, basisschakelingen.
  - Getallenrepresentatie, binair optellen, aftrekken, vermenigvuldigen en delen.
  - Eindige automaten, counters, registers, geheugen.

Overzicht van de stof, vervolg

- In week 4 en 5 zullen we (historisch/topdown) de opbouw van een processor bestuderen.
  - De von Neumann architectuur
  - De structuur van een CPU (Central Processing Unit, processor)
  - Adres-, control- en databus
  - De werking van een CPU
  - De CPU in detail: ALU, Registers, Vlaggen (Flags), Control, Instructie Set Architectuur en Addressing Modi.
  - De typische machine instructies (De practicum processor en 8086 als voorbeelden).

Overzicht van de stof, vervolg 2

- In week 6 en 7 gaan we in op het programmeren in machine code en welke tools daarvoor zijn:
  - Basis machinecode instructies
  - Stack operaties, subroutine call en return, recursie in machinecode.
  - Assembly
  - Assembler
  - Linker
  - Object en executable bestanden
  - Libraries, shared libraries en DLLs.

Overzicht van de stof, slot

- In week 8 gaan we in op twee methoden om processoren sneller te maken:
  - Pipelining (en zijn problemen)
  - Caching (en de problemen daarvan)

Beide technieken zijn ook in software te gebruiken, hetgeen een goede reden is om ze te bespreken.

Booleaanse algebra

Processoren
10 november 2014

Booleaanse algebra

- $(B, \wedge, \vee, \neg)$
  - B is een verzameling met een partiele ordening $\leq$
  - $a \wedge b$ = grootste element $\leq a$ en $\leq b$ conjunctie
  - $a \vee b$ = kleinste element $\geq a$ en $\geq b$ disjunctie
  - $\neg a$ = het element met $a \wedge \neg a = 0$ negatie
    en $a \vee \neg a = 1$
- Bij ons is altijd $B = \{0, 1\}$. Vanwege deze beperking zullen we de symbolen van de schakelalgebra gebruiken nl: ., +, $\overline{}$

Waarheidstabellen

a	b	$a \,.\, b$	$a + b$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

a	$\bar{a}$
0	1
1	0

- Idee: 0 = onwaar 1 = waar
  · = en (and) + = of (or)
  $\overline{}$ = niet (not)

Wetten voor Booleaanse algebra

Wet	Conjunctieve vorm	Disjunctieve vorm
Identiteit	$1 \,.\, a = a$	$0 + a = a$
Nulelement	$0 \,.\, a = 0$	$1 + a = 1$
Idempotentie	$a \,.\, a = a$	$a + a = a$
Inverse	$a \,.\, \bar{a} = 0$	$a + \bar{a} = 1$
Commutativiteit	$a \,.\, b = b \,.\, a$	$a + b = b + a$
Associativiteit	$(a \,.\, b) \,.\, c = a \,.\, (b \,.\, c)$	$(a + b) + c = a + (b + c)$
Distributiviteit	$a + (b \,.\, c) = (a+b).(a+c)$	$a \,.\, (b + c) = (a.b)+(a.c)$
Absorptie	$a \,.\, (a + b) = a$	$a + (a \,.\, b) = a$
De Morgan	$\overline{a \,.\, b} = \bar{a} + \bar{b}$	$\overline{a + b} = \bar{a} \,.\, \bar{b}$

Booleaanse functie

- functie van B^n naar B^m
  voorbeelden: and, or, binair optellen, aftrekken,...
  - algemeen: functies met een waarheidstabel

- Stelling (compleetheid van ., +, $\overline{}$):
  elke Booleaanse functie kan geschreven worden met ., +, $\overline{}$

- Conventie: . heeft prioriteit over + en mag als shorthand ook weggelaten worden.

Booleaanse functies vereenvoudigen

- gebruik de wetten van de Booleaanse algebra
- disjunctieve normaalvorm (som van producten):
  niet altijd de kortste vorm,
  maar eenvoudig te construeren en gebruiken
- Karnaugh-diagrammen:
  handmatige methode om een minimale DNV te vinden

Disjunctieve normaalvorm

- disjunctie van conjuncties van 1 of meer literals (variabele of zijn inverse)
  - voorbeelden:
    $a \,.\, b + a \,.\, \bar{c} + \bar{a} \,.\, \bar{b} \,.\, c$
  - geen voorbeelden:
    $a \,.\, (a + b \,.\, c)$
    $a \,.\, b \,.\, \bar{c} + a \,.\, \bar{b} \,.\, \bar{c} + \overline{a.\bar{b}.c}$

Functies in DNV opschrijven

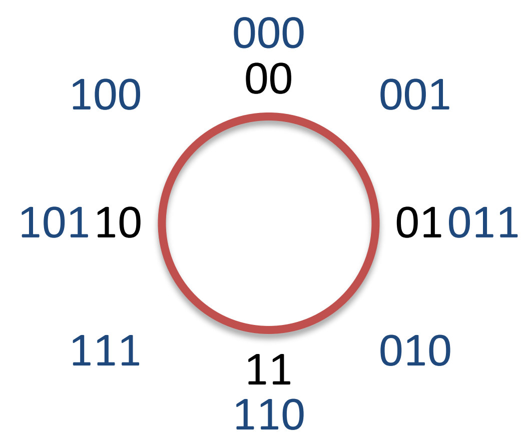
- Gebruik De Morgan [$\overline{a+b} = \bar{a} \, . \, \bar{b}$] om alle inversen naar literals te brengen

- Gebruik distributiviteit [$a.(b+c) = (a.b)+(a.c)$] om een disjunctie van conjuncties te krijgen

- We noemen een vorm een volledige DNV als elke literal exact een keer in een conjunctie voorkomt

Karnaugh-diagram

- Gray-code = cirkel/lijst van binaire getallen waarin twee opeenvolgende getallen maar 1 bit verschillen



- maak waarheidstabel in volgorde van Gray-code

- tabel-items onder of naast elkaar kun je dan samenvoegen

Karnaugh-diagram: voorbeeld 1

- F = $\bar{a}.\bar{b}.\bar{c} + \bar{a}.b.\bar{c} + a.b.\bar{c} + \bar{a}.\bar{b}.c +$
  $a.\bar{b}.c + \bar{a}.b.c + a.b.c$
- F = $\bar{a} + b + c$

	ab =			
	00	01	11	10
c=0	1	1	1	0
c=1	1	1	1	1

Karnaugh-diagram: voorbeeld 2

- f(a,b,c,d) = $\bar{a}.\bar{b}.\bar{c}.\bar{d}$ + $\bar{a}.\bar{b}$.c.$\bar{d}$ + $\bar{a}.b.\bar{c}.\bar{d}$ + $\bar{a}$.b.$\bar{c}$.d + $\bar{a}$.b.c.$\bar{d}$ + $a.\bar{b}.\bar{c}.\bar{d}$ + $a.\bar{b}.c.\bar{d}$ + $a.b.c.\bar{d}$ + $a.b.c.d$

$\bar{a}.b.\bar{c}$
+ $a.b.c$
+ $c.\bar{d}$
+ $\bar{b}.\bar{d}$

		a=0		a=1		a=0	
		b=0	b=1		b=0	b=0	
c=0	d=0	1	1	0	1	1	1
	d=1	0	1	0	0	0	1
c=1		0	0	1	0	0	0
	d=0	1	1	1	1	1	1
c=0	d=0	1	1	0	1	1	1
	d=1	0	1	0	0	0	1

Karnaugh-diagram: voorbeeld 2

- f(a,b,c,d) = $\bar{a}.\bar{b}.\bar{c}.\bar{d}$ + $\bar{a}.\bar{b}.\text{c}.\bar{d}$ + $\bar{a}.b.\bar{c}.\bar{d}$ + $\bar{a}.\text{b}.\bar{c}.\text{d}$ + $\bar{a}.b.c.\bar{d}$ + $a.\bar{b}.\bar{c}.\bar{d}$ + $a.\bar{b}.c.\bar{d}$ + $a.b.c.\bar{d}$ + $a.b.c.d$

$\bar{a}.b.\bar{c}$
\+ $a.b.c$
\+ $\bar{a}.\bar{d}$
\+ $\bar{b}.\bar{d}$

		a=0	a=0	a=1	a=1	a=0	a=0
		b=0	b=1	b=1	b=0	b=0	
c=0	d=0	1	1	0	1	1	1
c=0	d=1	0	1	0	0	0	1
c=1	d=1	0	0	1	0	0	0
c=1	d=0	1	1	1	1	1	1
c=0	d=0	1	1	0	1	1	1
c=0		0	1	0	0	0	1

Aan de slag:
teken je eigen Karnaugh-diagram

f(a,b,c,d) = $a.b.\bar{c}.\bar{d} + \bar{a}.\bar{b}.\bar{c}.d +$
$a.b.\bar{c}.d + a.\bar{b}.\bar{c}.d +$
$\bar{a}.\bar{b}.c.d + \bar{a}.b.c.d +$
$a.\bar{b}.c.d$

$= a.b.\bar{c}\ +$
$\bar{a}.c.d +$
$\bar{b}.d$

		a=0		a=1	
		b=0	b=1	b=1	b=0
c=0	d=0	0	0	1	0
c=0	d=1	1	0	1	1
c=1	d=1	1	1	0	1
c=1	d=0	0	0	0	0

Karnaugh diagrammen, slot

- Middels Karnaugh diagrammen kun je voor simpele Booleaanse functies (≤ 6 variabelen) een minimale som van producten van literals opstellen.

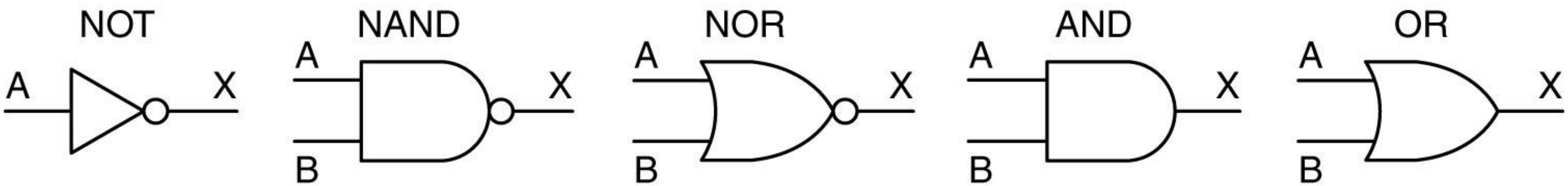
- Als je meer dan 1 output wil realiseren, hoeft het werken met Karnaugh diagrammen niet automatisch de kleinste oplossing te geven (Denk aan sharing van gemeenschappelijke expressies en meer dan 2 niveau diepe logica).

- Voor de liefhebber: Zoek maar eens naar het Quine-McCluskey- en het Espresso algoritme.

Wat heeft Booleaanse algebra te maken met computers?

Gates (poorten)

- grafische presentatie van Booleaanse functies



(a)

A	X
0	1
1	0

(b)

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(c)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(d)

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(e)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Gates in de computer

- Kies een voltage voor 0 en 1
  - meestal 0 = 0–1 V; 1 = 3–5 V (Op chip lager b.v. 2.5V)
- gates opbouwen uit transistoren
  - eenvoudigste gates: NOT, NAND, NOR
    4 transistoren schakeltijd: ca. 10 ns (TTL)
    2–4 transistoren schakeltijd: ca. 30 ps (CMOS,65nm)
  - ingewikkeldere gates: AND, OR
    8 transistoren schakeltijd: ca. 20 ns (TTL)
    6 transistoren schakeltijd: ca. 60 ps (CMOS,65nm)

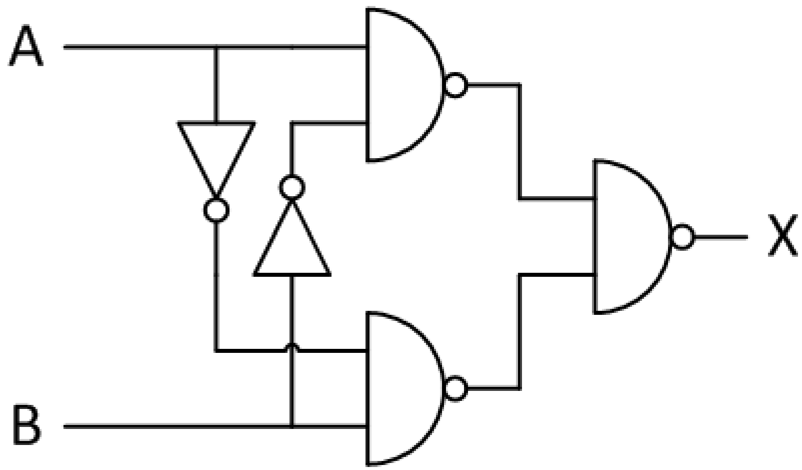
Voorbeelden van Booleaanse functies

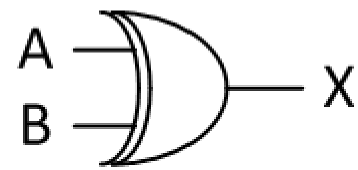
- Exor
- Multiplexer
- Decoder
- Adders
- En nog veel meer....

- De verzamelnaam voor dit soort circuits (realisaties van Booleaanse functies) is combinatorische logica.

EXOR

- Een veelgebruikt subcircuit is de exclusive or:

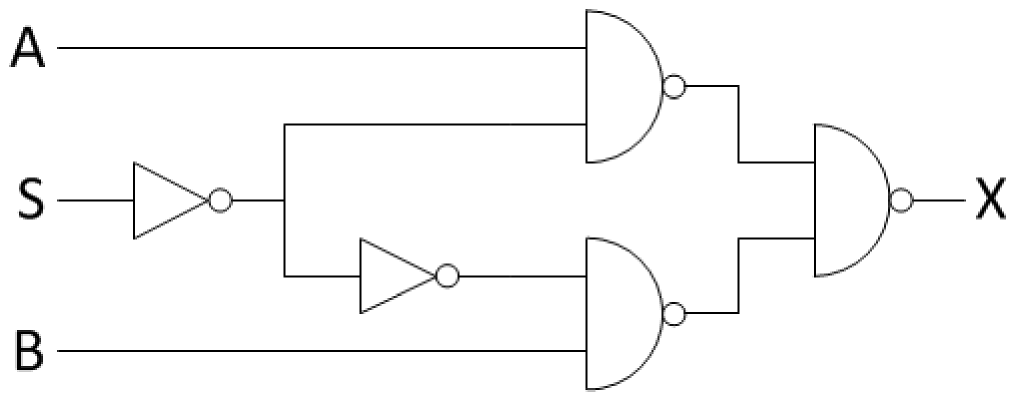




- $X = A \oplus B = \bar{A} \,.\, B + A \,.\, \bar{B}$

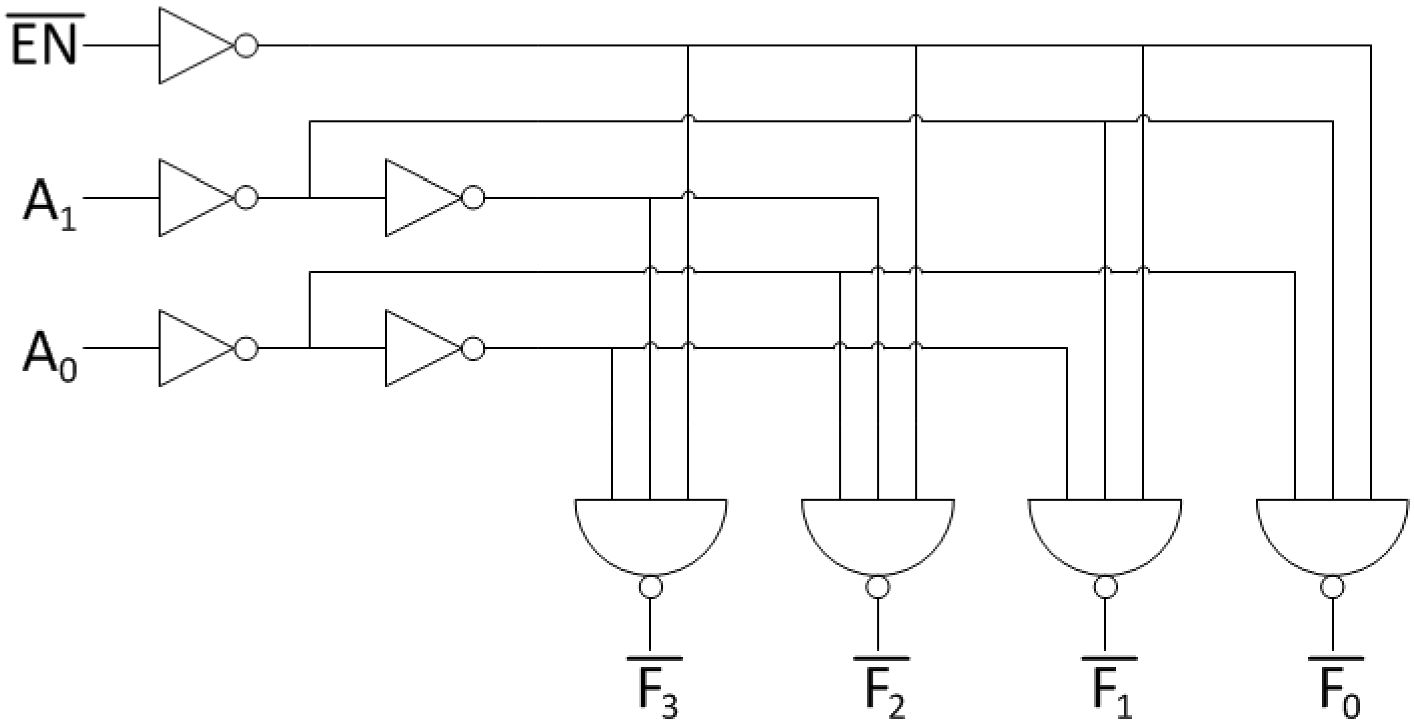
Multiplexer

- Een andere veel gebruikte component is de multiplexer:



- $X = \bar{S} \,.\, A + S \,.\, B$
- Multiplexers bestaan in vele smaken (4 naar 1, 8 naar 1, etc.).

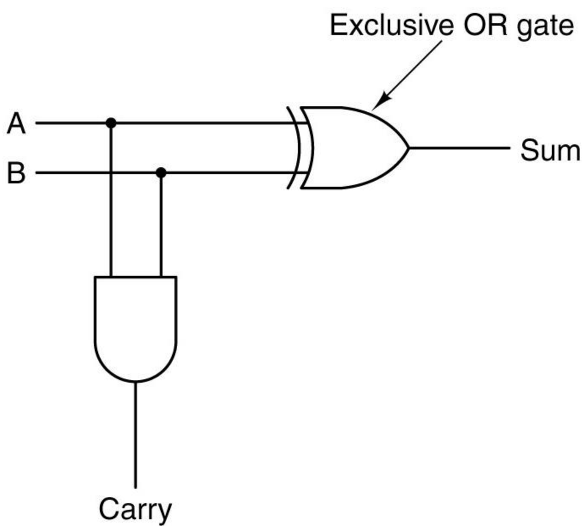
Decoder



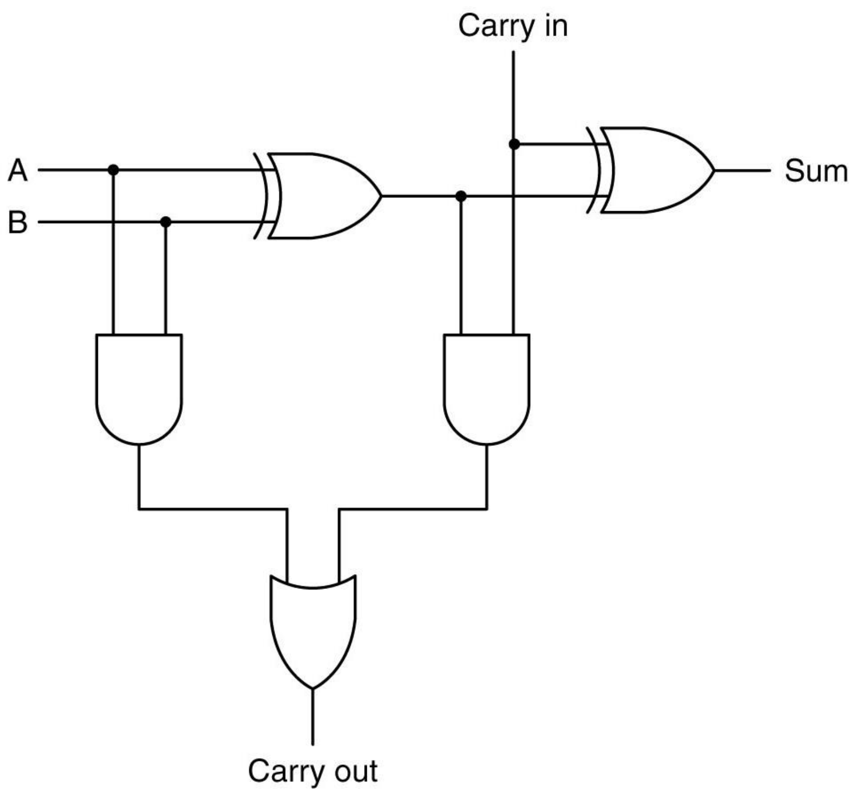
$\overline{EN}$
A_1
A_0
$\overline{F_3}$
$\overline{F_2}$
$\overline{F_1}$
$\overline{F_0}$

Adders

Half adder



Full adder



Samenvatting

- Booleaanse algebra
  - B = {0,1}, ., +, $\overline{}$
  - elke Booleaanse functie f: $B^n \rightarrow B^m$
    kan met ., +, $\overline{}$ geschreven worden
  - disjunctieve normaalvorm als standaardvorm
- Gates
  - grafische presentatie van Booleaanse functies
  - schakelingen voor NAND, NOR, NOT, AND, OR