DIGTEC/2021-2022

Jesse op den Brouw

DIGTEC

Introductie, poorten, vereenvoudigen

DE HAAGSE HOGESCHOOL

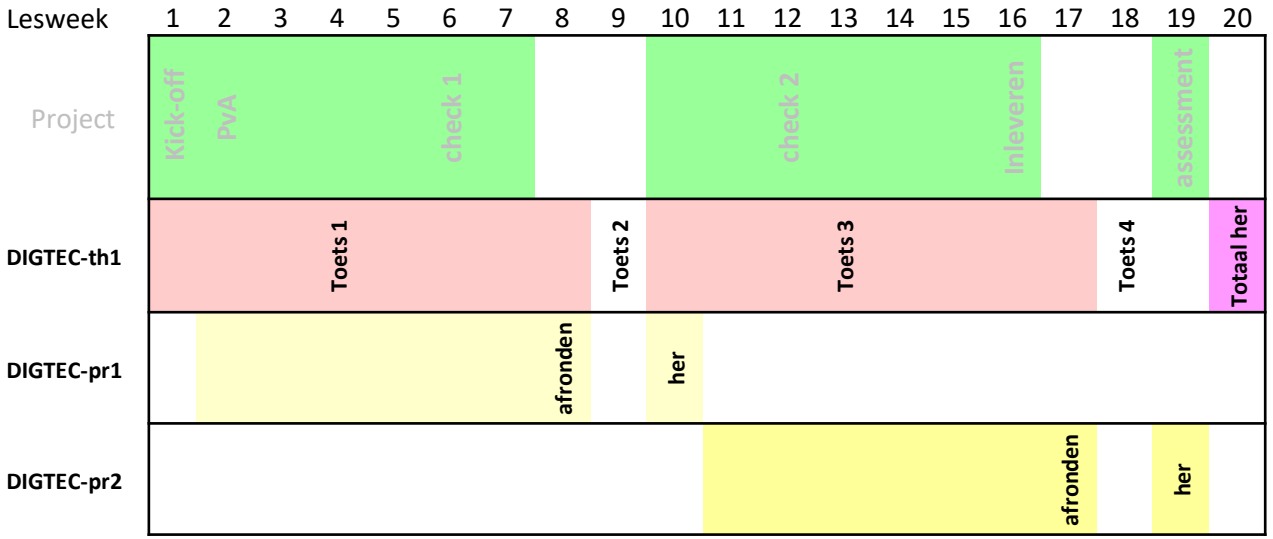
DIGTEC

- DIGTEC = Digitale Techniek
- De digitale techniek in de elektrotechniek, de basis van elke computer
- Ontwerpen van *hardware*
- Het vak duurt 20 weken (semestervak)
- Het vak bestaat uit theorie, practicum en toetsing
- Docenten zijn Jesse op den Brouw, Michiel van der Vlugt (practicum) en Willem-Pieter Zoutendijk (practicum)
- Email: J.E.J.opdenBrouw@hhs.nl

Plaats in het curriculum van DIGTEC

- Voorbereiding voor:
  - Projecten in het vervolg van de opleiding
  - UCPROG (Microcontroller programmeren)
  - OOSPLC (PLC-techniek)
  - Minor Embedded Systems
  - Keuzevak in ECK

Overzicht lesweken



- Projectenlijn is niet van toepassing
- Theorie begint in lesweek 1 en lesweek 10
- Practicum begint in lesweek 2 en lesweek 11

DE HAAGSE HOGESCHOOL

Planning (kan wijzigen)

Block	Week	Date	Monday	Tuesday	Wednesday	Thursday	Friday
3	1	07/02/2022	Start theorie				
3	2	14/02/2022	Start practica				
3	3	21/02/2022					
		28/02/2022	vakantie				
3	4	07/03/2022		CT1			
3	5	14/03/2022					
3	6	21/03/2022					
3	7	28/03/2022					
3	8	04/04/2022	Einde theo/prac				
3	9	11/04/2022				CT2	
3	10	18/04/2022	Paasmaandag	Herkansing practicum pr1			
4	1	25/04/2022	Start theorie		Koningsdag		
4	2	02/05/2022	Start practica			Bevrijdingsdag	
4	3	09/05/2022					CT3
4	4	16/05/2022					
4	5	23/05/2022				Hemelvaart	Hemelvaart
4	6	30/05/2022					
4	7	06/06/2022	Pinksteren	Einde theo/prac			
4	8	13/06/2022				CT4	
4	9	20/06/2022	herkansing practicum pr2				
4	10	27/06/2022				HER	

DE HAAGSE HOGESCHOOL

Werkvormen theorie

- DIGTEC-th1
  - 2 lesuren theorie per week (totaal 16x2 lesuren)
  - Docent is Jesse op den Brouw
- Behandelen van de lesstof a.d.h.v. de voorgeschreven literatuur.
- Beantwoorden van vragen (stel ze!)
- Uitwerken opgaven
- Colleges starten in lesweek 1 en lesweek 10

Werkvormen practicum

- DIGTEC-pr1/pr2
  - 2 lesuren practicum per week (totaal 14x2 lesuren)
  - Docenten zijn Jesse op den Brouw, Michiel van der Vlugt en Willem-Pieter Zoutendijk
- Uitwerken van de opgegeven opdrachten
- Tijdens de practica worden opdrachten afgetekend
- Gebruik een laptop!
- Practica starten in lesweek 2 en lesweek 11
- Aanwezigheid is verplicht!

Toetsing

- Toetsing theorie
  - In lesweek 4, 9, 13, 18, 20 (her)
  - Toetsing is cumulatief over de stof (10, 20, 30, 40 punten)
  - Hertoets is over de gehele stof (cumulatieve toetsen vervallen)
  - Open boek, open vragen

- Toetsing practicum
  - In lesweek 8, 10 (her), 17, 19 (her)

Leermiddelen

- Boek: Jesse op den Brouw, *Digitale Techniek,* ISBN 9789065624543

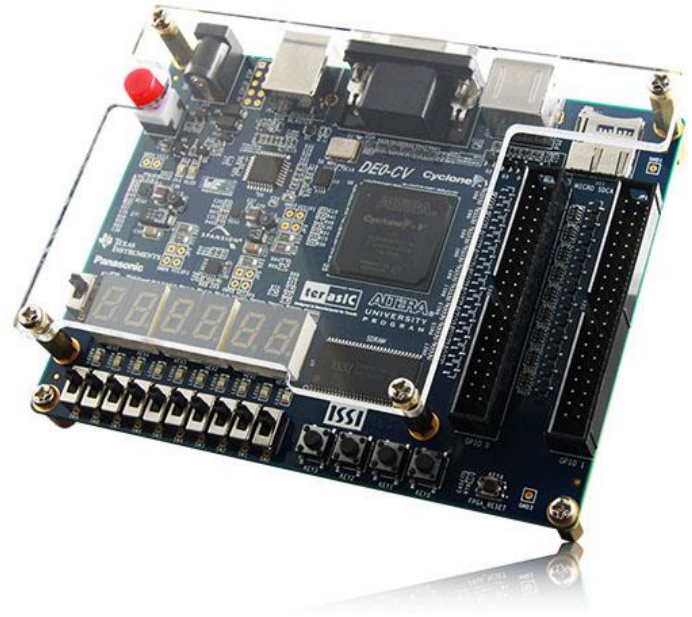
- Slides
- Uitwerkingen van de opgaven in het boek
- Practicumhandleiding
- Practicumopdrachten
  - Via BlackBoard

DE HAAGSE HOGESCHOOL

Leermiddelen

- Gebruik van een ontwikkelomgeving:
  - Quartus Prime Lite 20.1.1 + ModelSim
  - Werkt op Windows en Linux
  - Geen versie voor macOS
  - Kijk op BlackBoard voor een download-link

- In het lab:
  - Experimenteerbordje DE0-CV
  - Projecten zijn beschikbaar, maar niet afgerond



DE HAAGSE HOGESCHOOL

Wat weet je al?

- Wat weet je al van *poorten*?
  - AND, OR, EXOR, NOT
- Wat weet je al van *geheugens*?
  - Latch, flipflop
- Wat weet je al van *binair rekenen*?
  - Optellen, aftrekken, negatieve getallen
- Wat weet je al van *timing*?
  - Maximale klokfrequentie
- Wat weet je al van *spanning en stroom*?
  - Logische niveaus, wat is 0 en 1

Inhoud DIGTEC

- Introductie
- Basale logische schakelingen (*poorten*)
- Opbouw logische schakelingen, waarheidstabel
- Aansluitingen van leds en schakelaars
- Binaire/hexadecimale getallen, karaktercodes, BCD-code
- Schakelalgebra, de wiskunde van logische schakelingen
- Vereenvoudiging van schakelformules, Karnaughdiagrammen
- Synthese schakelingen, multiplexer
- Binair rekenen + schakelingen: optellen, vermenigvuldigen, delen

Inhoud DIGTEC

- Two’s complement-getallen, rekenen met negatieve getallen, aftrekken
- Vergelijken
- Geheugens: latches, flipflops, registers, timing
- Registers en schuifregisters
- Tellers
- Meer over spanning en stroom, opbouw logische schakelingen
- Timing: maximale klokfrequentie, setup- en hold-tijd
- Toestandsmachines

- Niet behandeld: floating-point getallen (erg complex), VHDL

FPGA

- We maken gebruik van een FPGA.
- Deze chip bevat logica, geheugens en programmeerbare verbindingen.
- We ontwerpen een schakeling met poorten en geheugens.
- Daarna “plaatsen” we deze logica in de FPGA.

- De FPGA heeft dus al logica aan boord.
- We moeten alleen de juiste verbindingen maken.
- Dat doet Quartus.

Simuleren

- We kunnen een schakeling simuleren.
- We doen dat met ModelSim (geinstalleerd via Quartus)
- We noemen dit ook wel “droogzwemmen”.

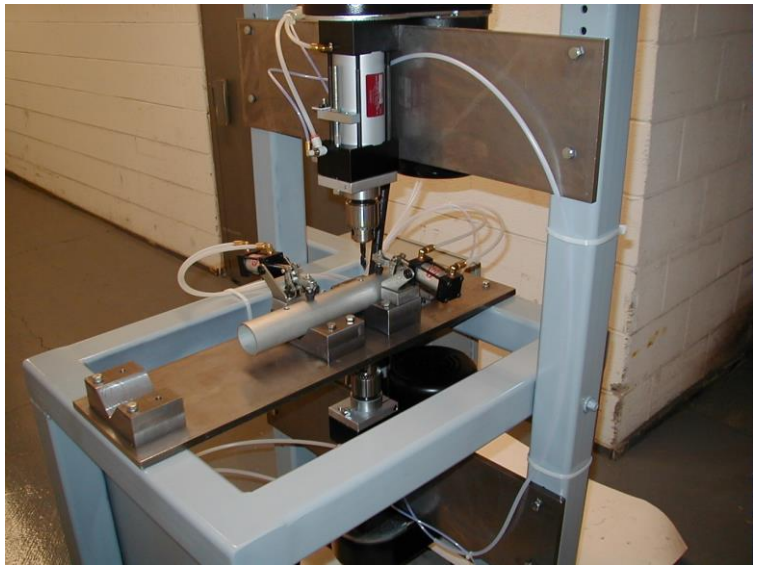
- Bij eenvoudige schakelingen kunnen we direct het experimenteerbordje gebruiken.
- Naar mate de schakelingen complexer worden, moeten we meer gebruik maken van de simulator.

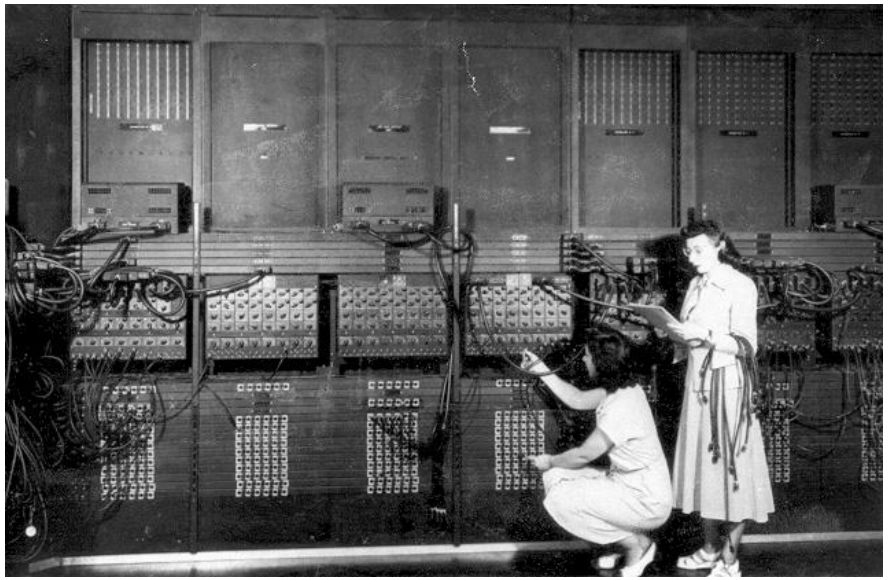
Waarom digitaal?

- Sommige gegevens zijn van nature digitaal:
  - Tijd in uren, minuten en seconden
  - Datum
  - Aantal fietsers per uur over een weg
  - Word-document
- Andere gegevens zijn digitaal te maken:
  - Spanning tussen twee punten in een netwerk (spanningsmeter)
  - Lichtintensiteit
  - Audio
  - Video

Oorsprong

- Automatisering productieprocessen
- Uitvoeren van complexe berekeningen





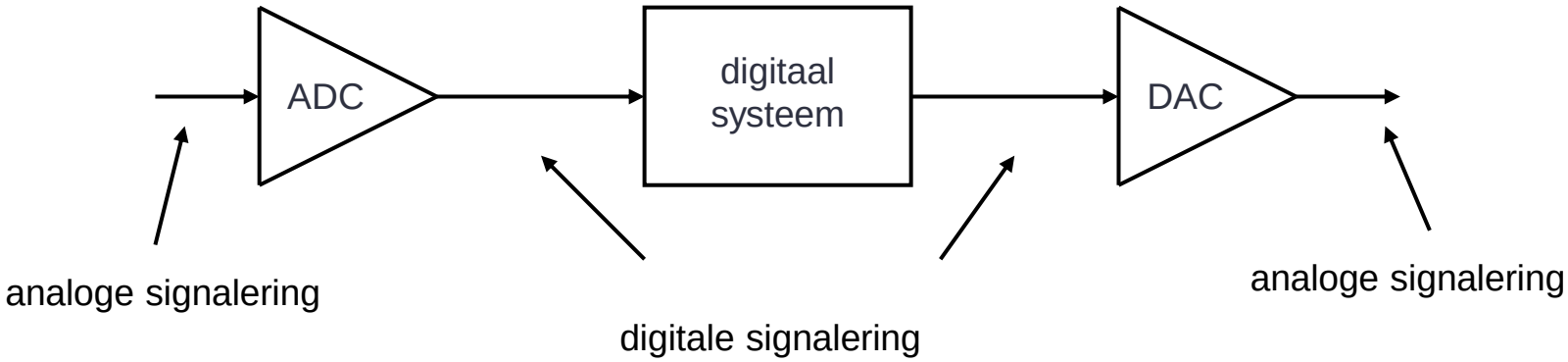
DE HAAGSE HOGESCHOOL

Digitalisering

- Bekende fysische grootheden (stroom, spanning, druk, licht, hoek, etc.) worden omgezet naar getallen.
- Het is eigenlijk een transformatie van de fysische grootheden naar getallen die in principe niet binair (tweewaardig) hoeven te zijn.
- De digitale informatie kan met behulp van elektronica (computers) bewerkt worden en weer naar de fysische grootheden terug getransformeerd worden.
- Dus: fysiek $\xrightarrow[\text{omzetting}]{}$ getallen $\xrightarrow[\text{omzetting}]{}$ fysiek

Digitalisering

- Transformatie gebeurt met behulp van Analog-Digital Converter (ADC) en Digital-Analog Converter (DAC).



DE HAAGSE HOGESCHOOL

Digitalisering

- Waarom digitaliseren we informatie?

- Verwerking
  - informatie is gemakkelijk te bewerken (versleutelen, comprimeren, foutdetectie, foutcorrectie, complexe signaalbewerkingen);
  - gemakkelijk informatie op te slaan voor latere bewerking;
  - informatie kan langere tijd bewaard worden met behoud van kwaliteit (reproduceerbaarheid van informatie);
  - betrouwbaarheid m.b.t. storende signalen;
  - snelheid;
  - parallel verwerken van informatie.

Digitalisering

- Waarom digitaliseren we informatie?

- Ontwerpen
  - snel ontwerpen en testen m.b.v. tools (beschrijvingstaal, simulatie);
  - eenvoudig te leren, geen uitgebreide wiskundige kennis nodig;
  - uitbreidbaarheid (vooral in programmeerbare logica);
  - hoge integratiedichtheid (veel functionaliteit op een klein oppervlak);
  - goede testvoorzieningen aan te brengen.
  - gebruik van standaard blokken (IP-cores: Ethernet, USB, processoren, DSP, RAM).

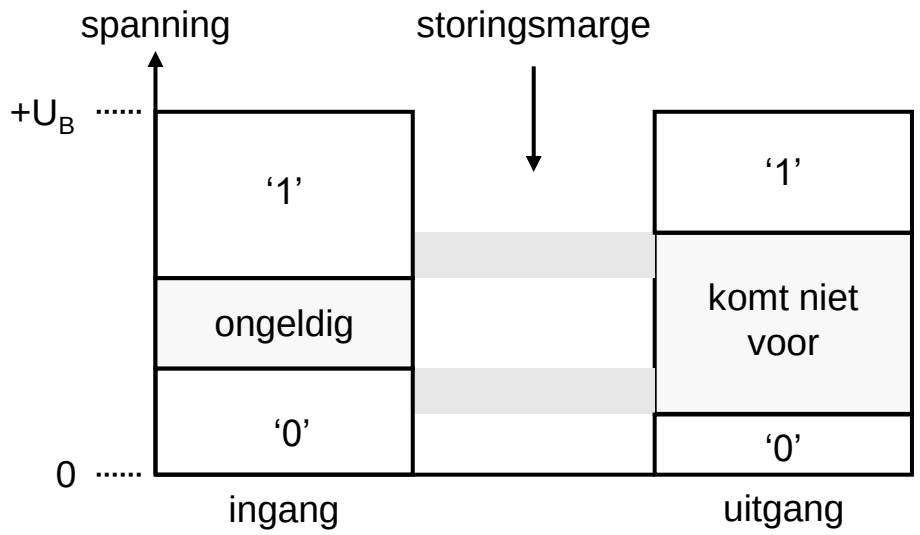
Logische waarden

- In de digitale techniek werken we met twee waarden: 0 en 1
  - Schakelingen werken met deze 0 en 1
  - Soms geven we aan deze twee waarden een waarheidsgehalte:
  - 0 = niet waar (false)
  - 1 = waar (true)
  - We noemen dit positieve logica

- Met 0 en 1 kunnen we ook getallen representeren
  - Binaire getallen (bv in een computer)
  - Codes (bv karakters)

Logische spanningen

- Een digitale schakeling werkt met spanningen en stromen. Voor het aangeven van een logische waarde wordt een spanningsgebied gebruikt. Een storingsmarge tussen ingang en uitgang is noodzakelijk om variaties in voedings- en signaalspanningen op te vangen.



Ub = 5 V; 3,3 V; 2.5 V: 1,8 V

Arduino Uno: 5 V

STM32: 3,3 V

Cyclone V: (o.a.) 2,5 V

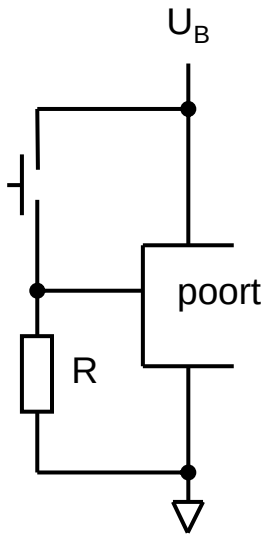
DE HAAGSE HOGESCHOOL

Logische stromen

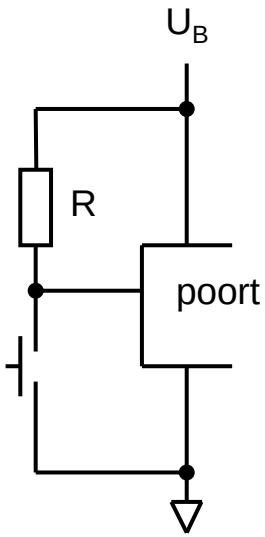
- Afhankelijk van de schakeling:
- Ouderwetse TTL : input: min 40 µA, output: max 16 mA,
- Moderne CMOS: input 1 µA – 10 µA, output 1 mA – 20 mA
- Veel schakelingen hebben een stroombegrenzing
  - Arduino Uno: 20 mA
  - STM32: 8 mA
  - Cyclone V: 12 mA

Actief hoog en laag

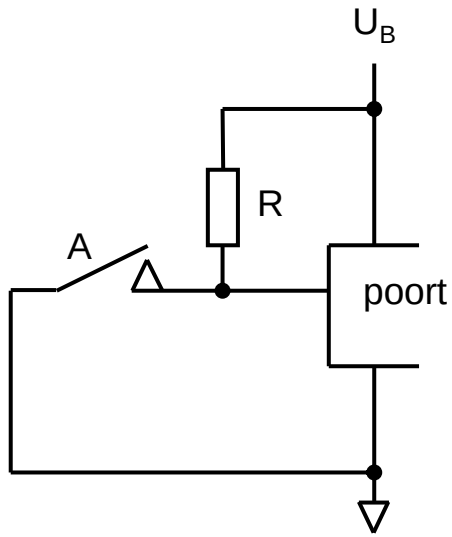
- Schakelaars kunnen actief hoog en laag worden aangesloten.



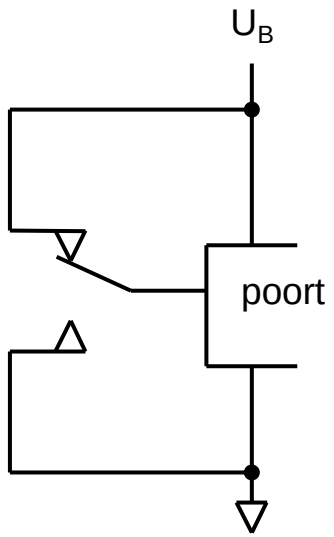
actief hoog



actief laag



actief laag

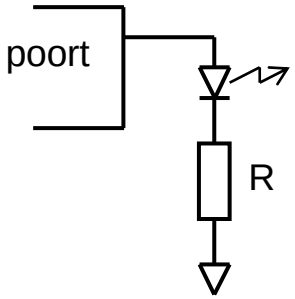


twee actieve standen

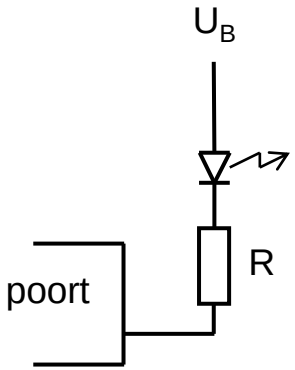
DE HAAGSE HOGESCHOOL

Actief hoog en laag

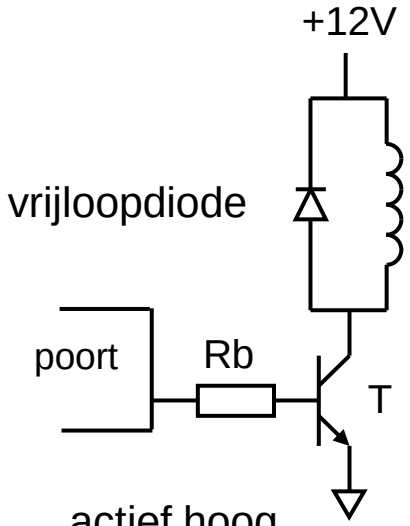
- Uitgangen (leds, spoelen) kunnen actief hoog en laag worden aangesloten.



actief hoog



actief laag



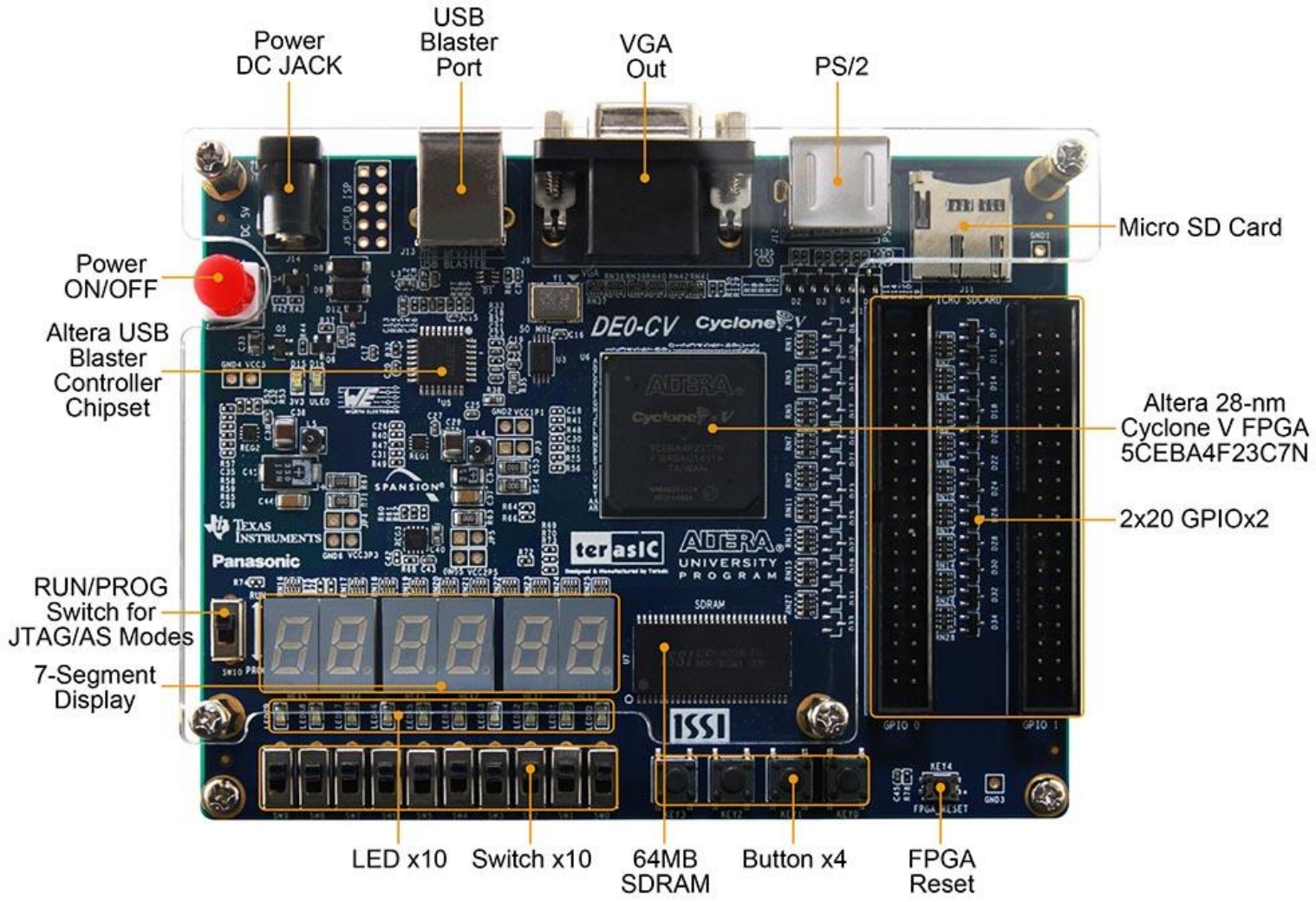
actief hoog

spoel (motor)

transistor (stroomversterker)

DE HAAGSE HOGESCHOOL

DE0-CV board



HAAGSE HOGESCHOOL

Poorten

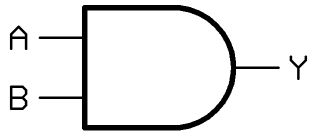
- Digitale schakelingen worden verreweg het meest ontworpen met elektronische componenten.
- Andere mogelijkheden:
  - pneumatische componenten (lucht)
  - hydraulische componenten (vloeistof)
  - fotonische componenten (licht)
- Digitale schakelingen worden opgebouwd met behulp van *poorten*.
- Eerst worden de elementaire poorten besproken.

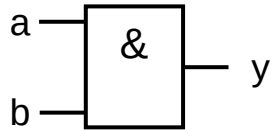
Poorten

- Elke digitale schakeling kan worden opgebouwd met behulp van drie typen poorten:
- AND (en)
- OR (of)
- NOT (niet, inverter)
- Andere veel voorkomende poorten:
  - Buffer, EXOR, NAND, NOR, EXNOR

AND

Uitgang y geeft alleen een 1 af als beide ingangen een 1 zijn, anders geeft uitgang y een 0 af.





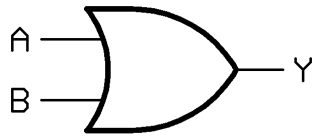
a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

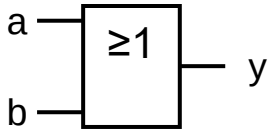
Functie: $y = a \cdot b = ab$

Uitspraak: “y is gelijk aan a and b”

OR

Uitgang y geeft een 1 af als één of meer ingangen 1 zijn, anders geeft uitgang y een 0 af.





a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

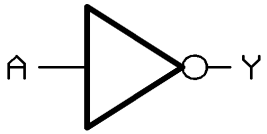
Functie: $y = a + b$

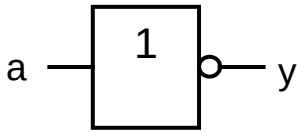
Uitspraak: “y is gelijk aan a or b”

DE HAAGSE HOGESCHOOL

NOT

Uitgang y geeft een 0 af ingang a een 1 is, anders geeft uitgang y een 1 af.





a	y
0	1
1	0

Functie: $y = \overline{a}$

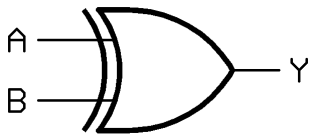
Uitspraak: “y is gelijk aan not a”

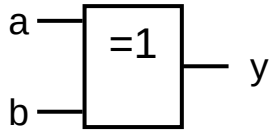
Poorten

- Hiervoor zijn de drie basispoorten besproken. Hiermee kan elke digitale schakeling worden ontworpen en gebouwd.
- In de praktijk worden nog andere poorten gebruikt.
- Deze poorten zijn niet strikt noodzakelijk, wel heel handig. Ze worden gebruikt bij veel voorkomende bewerkingen.
- Al deze poorten kunnen worden opgebouwd uit AND, OR en NOT.

EXOR

Uitgang y geeft een 1 af als precies één ingang 1 is, anders geeft uitgang y een 0 af.





a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

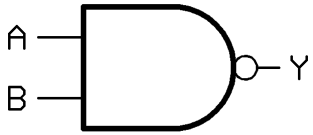
Functie: $y = a \oplus b$

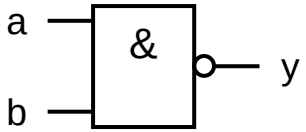
Uitspraak: “y is gelijk aan a exor b”

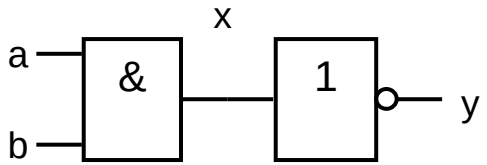
Waarom: komt voor bij rekenschakelingen

NAND (AND-NOT)

Inverse van de AND







a	b	x	y
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

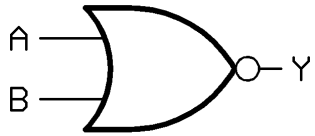
Functie: $y = \overline{a \cdot b}$

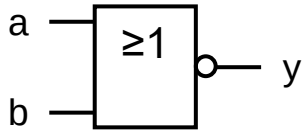
Uitspraak: “y is gelijk aan a nand b”

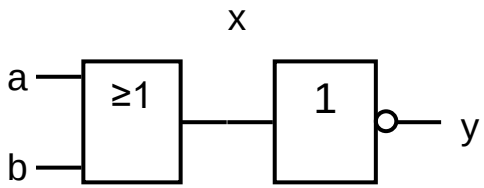
Waarom: makkelijk te maken op een chip

NOR (OR-NOT)

Inverse van de OR







a	b	x	y
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Functie: $y = \overline{a + b}$

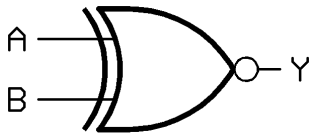
Uitspraak: “y is gelijk aan a nor b”

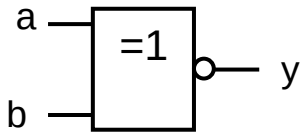
Waarom: makkelijk te maken op een chip

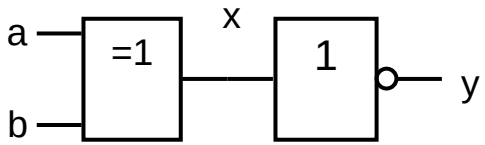
DE HAAGSE HOGESCHOOL

EXNOR (EXOR-NOT)

Inverse van de EXOR







a	b	x	y
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Functie: $y = \overline{a \oplus b}$

Uitspraak: “y is gelijk aan a exnor b”

Waarom: wordt ook wel de “gelijkheidsschakeling” genoemd

DE HAAGSE HOGESCHOOL

Voorbeeld schakeling

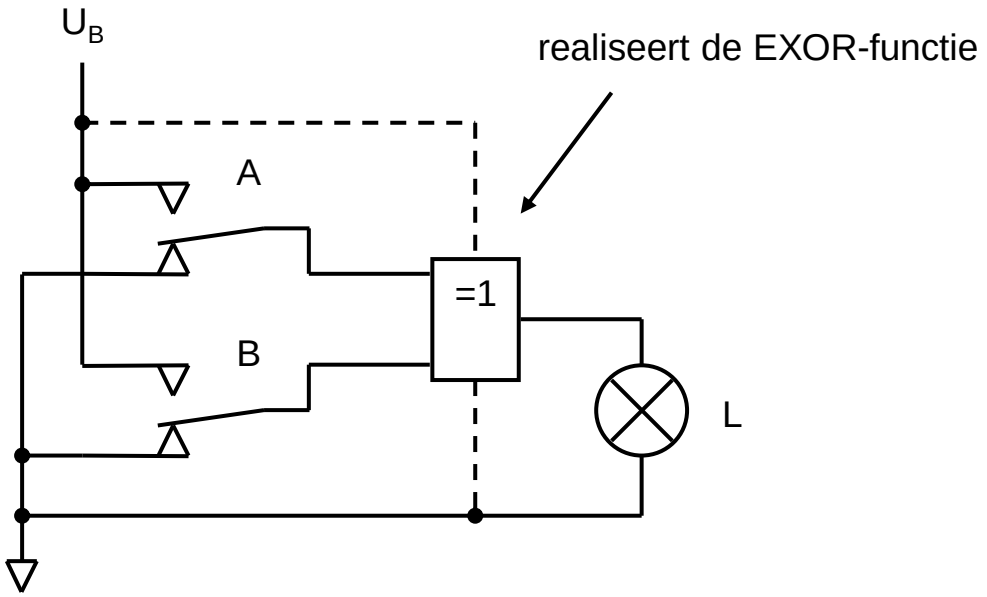
Deze schakeling realiseert de EXOR-functie met behulp van een poort.

De lamp brandt als beide schakelaars verschillende stand hebben (ongelijk zijn)

Vertaling spanningen voor ingangen en uitgang:

‘0’ = 0 V

‘1’ = U_B V



DE HAAGSE HOGESCHOOL

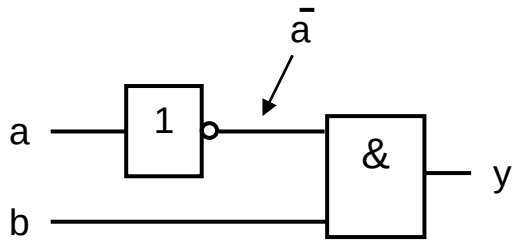
Vereenvoudigen

- Met behulp van de vorige slides kunnen we digitale poortschakelingen vereenvoudigen.
- De eenvoudigste vorm is de vorm die het minst aantal poorten bevat en met het minst aantal ingangen per poort.
- Het vinden van de eenvoudigste vorm heet *minimaliseren*.
- Er zijn drie manieren om te vereenvoudigen:
  - Wiskunde (schakelalgebra)
  - Karnaughdiagrammen (grafisch)
  - Computerprogramma (computer-algoritme: Quine-McCluskey, Espresso)

Vereenvoudigen

Gegeven onderstaande waarheidstabel. Wat is hiervan de functie en het schema?

Functie: $y = \bar{a} \cdot b$



a	b	y
0	0	0
0	1	1
1	0	0
1	1	0

Uitgang y = 1 als a = 0 en b = 1, anders y = 0;

DE HAAGSE HOGESCHOOL

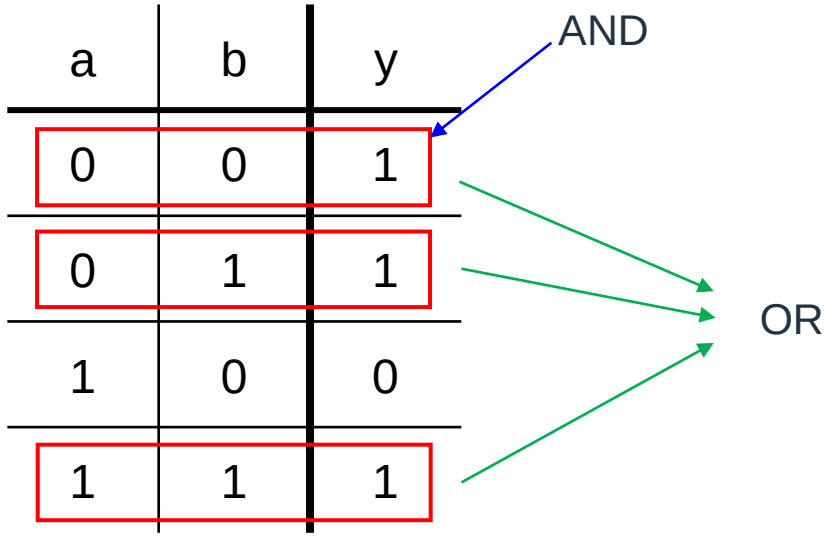
Vereenvoudigen

Gegeven onderstaande waarheidstabel. Wat is hiervan de functie en het schema?

Functie: $y = \bar{a} \cdot \bar{b} + \bar{a} \cdot b + a \cdot b$

Uitgang y = 1 als a = 0 en b = 0
of a = 0 en b = 1
of a = 1 en b = 1
anders y = 0

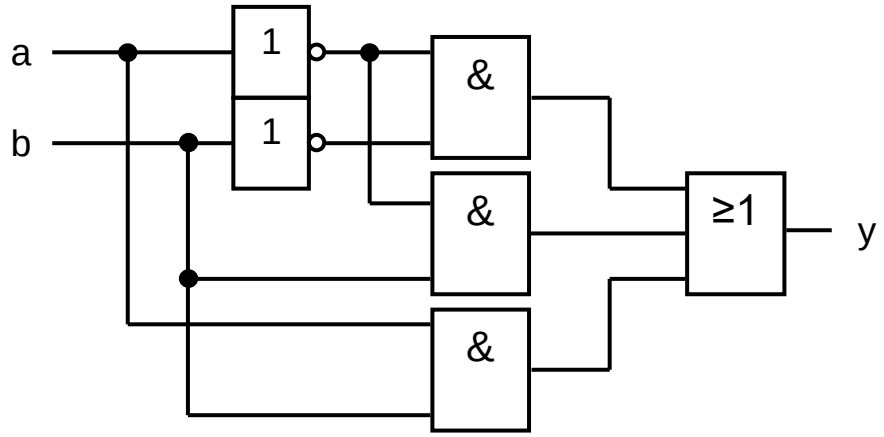
a	b	y
0	0	1
0	1	1
1	0	0
1	1	1



DE HAAGSE HOGESCHOOL

Vereenvoudigen

- De schakeling kan als volgt gebouwd worden:



- Kosten: 2x NOT, 3x AND, 1x OR3
- Ingangen: 11

- Opmerking: ingangen/uitgangen is bedrading (wires) op een IC.

Vereenvoudigen

Eenvoudig is te zien dat de functie een 1 levert als a 0 is.

Dus: $\bar{a} \cdot \bar{b} + \bar{a} \cdot b = \bar{a}$

Ook is snel te zien dat de uitgang een 1 levert als b 1 is.

a	b	y
0	0	1
0	1	1
1	0	0
1	1	1

Dus: $\bar{a} \cdot b + a \cdot b = b$

Opvallend: de term $\bar{a} \cdot b$ wordt twee keer gebruikt.

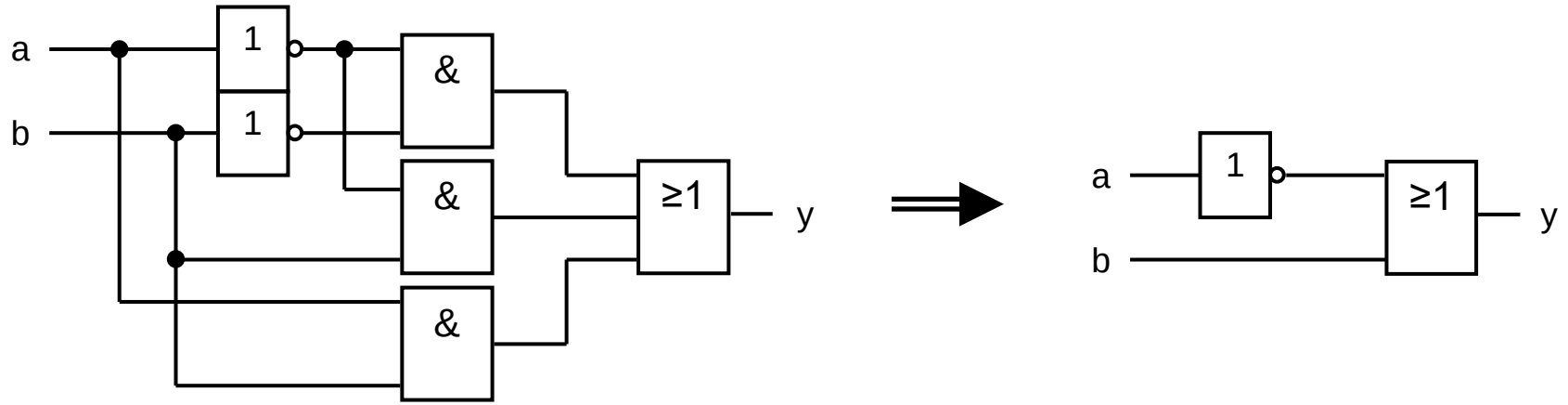
DE HAAGSE HOGESCHOOL

Vereenvoudigen

De functie is dus als volgt:

$$y = \bar{a} \cdot \bar{b} + \bar{a} \cdot b + a \cdot b$$
$$= \bar{a} + b$$

Het bijbehorend schema:



DE HAAGSE HOGESCHOOL

Opgaven

- Ontwerp een omschakelbare buffer/inverter
  - Eén signaal dient als stuursignaal (geeft aan wat er moet gebeuren)
  - Eén signaal dient als data (moet gebufferd of geinverteerd worden)

- Ontwerp een schakeling die twee 4-bits getallen test op gelijkheid
  - We gaan hier ervan uit dat je bekend bent met binaire getallen

let’s change