

Informatieoverdracht en -verwerking lesaantekeningen

Fordeyn Tibo

INHOUDSTAFEL

I Informatietheorie	2
I.1 Discrete informatiebronnen	3
I.2 Broncodering	4
I.3 Oefenzitting één	5
I.4 Continue informatiebronnen	6
I.5 Digitaliseren continue informatiebronnen	7
I.6 Oefenzitting twee	8
I.7 Digitale signalen	9
I.8 Digitale signaalverwerking	10
I.9 Oefenzitting drie	11
I.10 Discrete transmissiekanalen	12
I.11 Vierde oefenzitting	13
I.12 Fysische transmissiekanalen	14
I.13 Basisbandtransmissie	15
I.14 Doorlaatbandtransmissie	16
I.15 Oefenzitting B2	17
I.16 Digitaal ontwerp	18
I.17 Combinatorische logica	19
17.1 SOP en POS optimaliseren	19

Deel I

Informatietheorie

I.1 Discrete informatiebronnen

I.2 Broncodering

I.3 Oefenzitting één

I.4 Continue informatiebronnen

I.5 Digitaliseren continue informatiebronnen

I.6 Oefenzitting twee

I.7 Digitale signalen

I.8 Digitale signaalverwerking

I.9 Oefenzitting drie

I.10 Discrete transmissiekanalen

I.11 Vierde oefenzitting

I.12 Fysische transmissiekanalen

I.13 Basisbandtransmission

I.14 Doorlaatbandtransmissie

I.15 Oefenzitting B2

I.16 Digitaal ontwerp

Voorbeeld 16.0.1

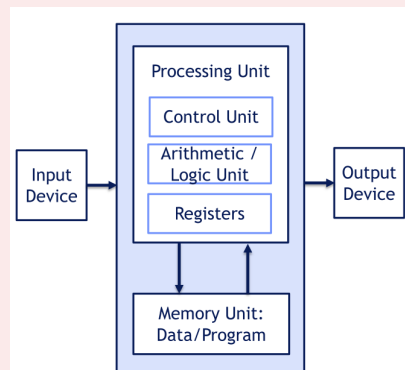
Verschillende niveaus van abstractie in digitale computer

Niveau	Inhoud	
Applicaties	Programma's, code	CW
Besturingssysteem	Drivers, framework, library, ...	
Architectuur	Instructieset	
Micro-architectuur	Datapad, controller	Dit vak en aansluitende ESAT vakken
Digitaal circuit	Geheugen, opteller, MUX, ...	
Logische poort	AND, OR, NOT, ... poorten	
Transistor	N-/P-transistors	Natuurkunde en materiaalkunde
Fysica	Elektronen, kwantum tunneling	

Definitie 16.0.1: Von Neumann architectuur

De kracht is je maakt een PU, die data uit geheugen haalt en kan egschrijven en je kan er oneindig veel programmas op uitvoeren. Een nieuw programma maken en opslaan in het geuehgen, dan kan dezelfde processor opnieuw gebruikt worden.

- Opslag gebaseerd, programma in geheugen
- Herprogrammeren mogelijk
- Basis architectuur hedendaagse computer



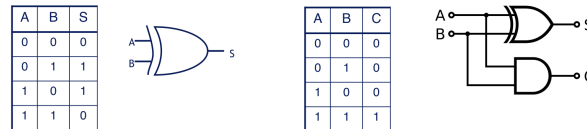
niet alle acties kunnen door één regel code worden uitgevoerd door de processor. We hebben drie types instructies;

- Spring: jump in programmacounter
- Register; doe bepaalde operatie op waarden in register
- Geheugen; van/naar memory van/naar register

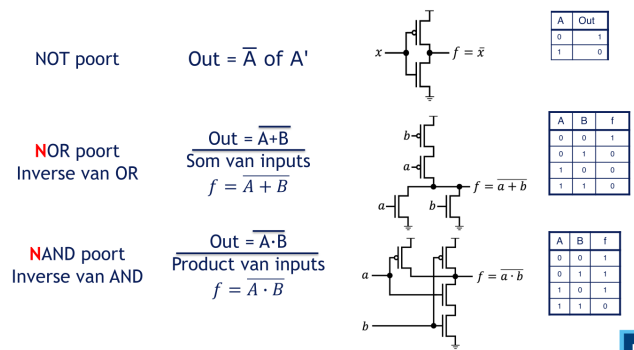
2 complement kun je nemen door va nrechts naar links te gaan tot je de één bit tegenkomt en dan daarna alles inverteren. Me de 2-complement bewerking kun je een ALU en MUX schakeling maken, bekijk nog eens zal niet moeilijk zijn.

I.17 Combinatorische logica

verschil tussen half adder en full adder; full adder zal ook de carr-bit van de vorige operatie meenemen!
de carry bit is de and poort. De sum bit is gelijk aan de XOR poort.



meest rechtse is dus de 1 bit adder.
onthoud hoeveel transistorene er in not nor en nand poorten zitten.



hoe groter de poort ho complexer en hoe groter de stijgtijd en daaltijd van de poort.

Definitie 17.0.1: kritische pad

het kritische pad is het langste pad van ingang naar uitgang.

17.1 SOP en POS optimaliseren

dit is met een stappenplan dat zeer belangrijk is voor examen!!! Karnaugh kaart; manier om gemakkelijk te zien of bepaalde variabelen niet belangrijk zijn in tegenstelling tot waarheidstabel.

Idee of vraag 17.1.1

Snap de karnaugh minimalisatie niet zijn lti systemen belangrijk?