

Antwoorden deeltentamen Architectuur/Digitale techniek voor studenten Informatica

Vraag 1 Boole algebra en poorten (12 punten resp. 2, 3, 2, 5 punten)

1.1. Geef de waarheidstabel van een NAND-poort met twee ingangen weer.

v	w	Z
0	0	1
0	1	1
1	0	1
1	1	0

1.2. Stel dat voor een NAND-operator het \otimes -symbool wordt gebruikt. Bewijs m.b.v. een waarheidstabel dat voor deze operator de volgende gelijkheid niet geldt: $p \otimes (q \otimes r) = (p \otimes q) \otimes r$.

p	q	r	$q \otimes r$	$p \otimes (q \otimes r)$	$p \otimes q$	$(p \otimes q) \otimes r$
0	0	0	1	1	1	1
0	0	1	1	1	1	0
0	1	0	1	1	1	1
0	1	1	0	1	1	0
1	0	0	1	0	1	1
1	0	1	1	0	1	0
1	1	0	1	0	0	1
1	1	1	0	1	0	1

Antwoord opgave 1.2

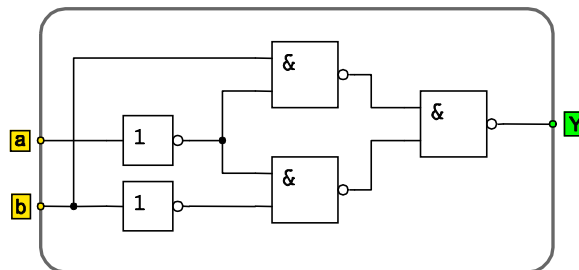
Voor bijv. $p = 0, q = 0$ en $r = 1$ geldt: $p \otimes (q \otimes r) = 0 \otimes (0 \otimes 1) = 0 \otimes 1 = 1$.

Terwijl $(p \otimes q) \otimes r = (0 \otimes 0) \otimes 1 = 1 \otimes 1 = 0$.

- 1.3.** Is in 1.2 bewezen dat:
- a. de associatieve wet,
 - b. de commutatieve wet of
 - c. de distributieve wet voor de NAND-operator niet geldt?

Antwoord is a. De associatieve wet geldt niet.

- 1.4.** Geef van de schakeling weergegeven in figuur 1 de Boole-uitdrukking weer (geef een uitdrukking voor Y als functie van a en b). Vereenvoudig deze uitdrukking zo ver als mogelijk is. m.b.v. de regels van de Boole algebra. Geef duidelijk weer welke wet(ten)/regel(s) je hierbij gebruikt.



Figuur 1

Antwoord: $Y = \overline{\overline{b} \cdot a}$ $\overline{\overline{b} \cdot a} = \overline{\overline{b}} + \overline{a} = b + \overline{a} = \overline{a} \cdot (b + 1) = \overline{a} \cdot 1 = \overline{a}$.

Achtereenvolgens: De Morgan; eerste distributieve wet; rekenregel.

Vraag 2 Rekenschakelingen; two's complementcode (9 punten resp. 2, 2, 5 punten)

2.1. Wat is in het geval van een zes bit computersysteem het grootste en het kleinste getal dat kan worden gerepresenteerd? Geef zowel de two's complement code als de decimale waarde ervan weer.

Antwoord: Grootste getal: 011111 = 31; Kleinste getal: 100000 = -32.

2.2. Wat is een 'full adder'? Geef de waarheidstabel van een Full Adder weer.

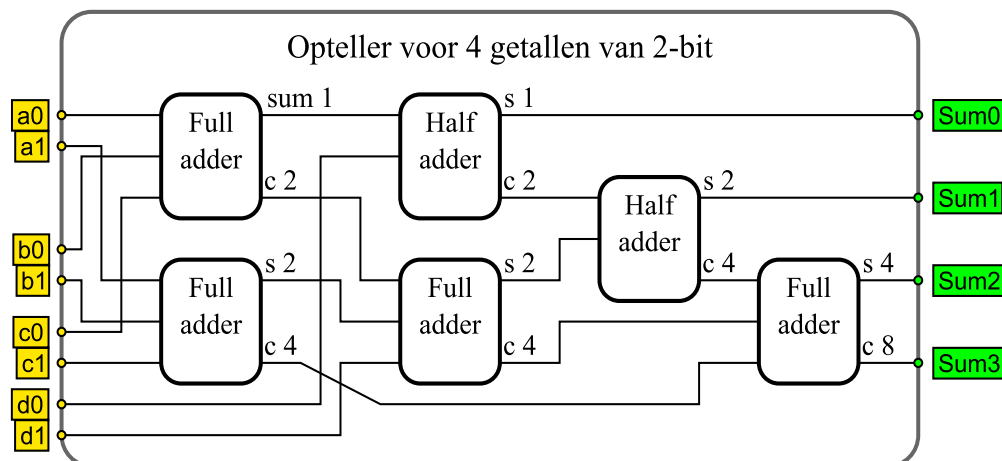
Antwoord: Een 'full adder' is een schakeling waarmee drie 1-bits getallen kunnen worden opgeteld.

2.3. Ontwerp een schakeling waarmee vier getallen van twee bits kunnen worden opgeteld. Gebruik als bouwstenen 'full adders'. Geef hierbij het 'gewicht' van iedere bit aan. Optimaliseer je ontwerp op propagatie tijd. Wat is het grootste binaire getal dat op de uitgangen van de schakeling kan voorkomen?

a	b	c	Carry	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

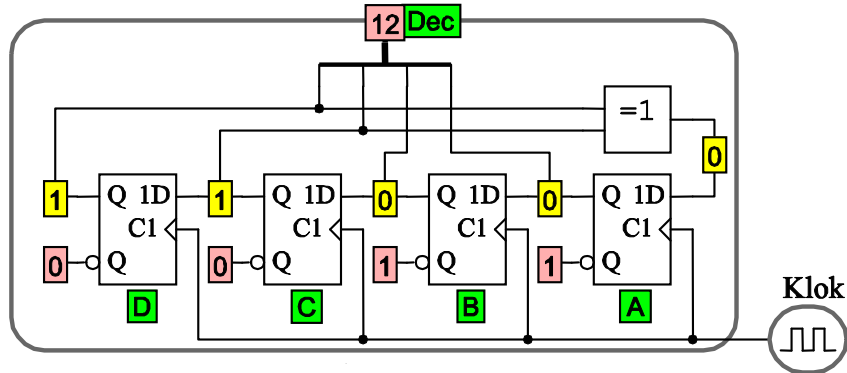
Waarheidstabel full adder

Antwoord: Het grootste getal = 1100 (12). Het signaal, dat de langste weg aflegt, moet door 4 poorten. Toepassen van half adders verkort de propagatietijd. Dit geldt alleen voor de meest rechts half adder.



Vraag 3: Sequentiële schakelingen (9 punten resp 3, 2, 2, 2 punten) Linear Feedback Shift Register

Een Linear Feedback Shift Register (afgekort **LFSR**) is een schuifregister dat als belangrijkste kenmerk heeft dat bepaalde uitgangen via een poort worden teruggekoppeld naar de ingang van het schuifregister. Het schuifregister is op deze manier in staat een reeks van 'pseudo random' getallen te genereren. De lengte van de reeks hangt af van het aantal geheugenelementen en van welke uitgangen teruggekoppeld worden. Elk getal uit de reeks komt maar één keer voor. Als de reeks is afgelopen wordt deze herhaald in dezelfde volgorde, vandaar de naam "pseudo random". Figuur 2 geeft een voorbeeld van een pseudo random getallen generator weer. De LFSR genereert hier het decimale getal 12.



Figuur 2

- 3.1. Welke getallen worden er, uitgaande van de toestand in figuur 2, na 1, na 2 en na 3 klokpulsen gegenereerd? *Antwoord zie tabel 1.*

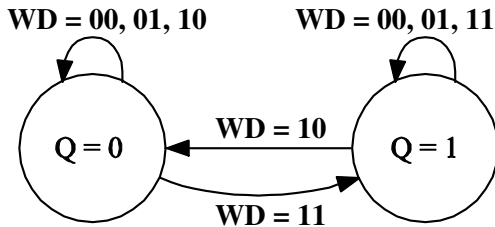
na pulsnummer	Q _D	Q _C	Q _B	Q _A	1D flipflop A	Decimaal
0	1	1	0	0	0	12
1	1	0	0	0	1	8
2	0	0	0	1	0	1
3	0	0	1	0	0	2
4	0	1	0	0	1	4
5	1	0	0	1	1	9
6	0	0	1	1	0	3

Tabel 1

- 3.2. Geef het bijbehorende tijdvolgordediagram weer. Klok, 1D_{FFA}, Q_A, Q_B, Q_C en Q_D als functie van de tijd (= klokpulsnummer)
- 3.3. Welke typen 1-bits geheugenelementen bestaan er?
Antwoord: Set-Reset latch, D-latch, D-flipflop en JK-flipflop.
- 3.4. Zijn de geheugenelement in figuur 2 flank(edge) of niveau(level) gestuurd en waarom?
Antwoord: Edge-triggered. Als de geheugenelementen level-triggered waren dan trad er "racing" op zodra de klok en dus alle C1-ingangen 'hoog' zijn. D.w.z. dat de signalen op de uitgangen van de flipflops via de XOR-poort 'rondracen' zolang de klok 'hoog' is met een frequentie die afhangt van de propagatietijd van de geheugenelementen. Ook aan het symbool is te zien dat het (edge-triggered) D-flipflops zijn.

Vraag 4: Eindige toestandsautomaat (17 punten resp 2, 3, 2, 3, 2, 2 punten)

Registers worden o.a. gebruikt om (tussen)resultaten van berekeningen op te slaan. Registers zijn opgebouwd uit D-flipflops en poorten. We beschouwen een 1-bits register. Naast een klokingang heeft deze twee ingangen, Data en Write, en één uitgang Q.



4.2 State Diagram

Write	Data	Clock	Q _{next}	Functie
0	0	↑	Q _{current}	Memory
0	1	↑	Q _{current}	Memory
1	0	↑	0	Load 0
1	1	↑	1	Load 1

Tabel 2

De relatie tussen Write, Data, Clock en Q is vastgelegd in tabel 2.

- 4.1. Breid tabel 2 uit met een kolom Q_{current}
- 4.2. Geef het State Diagram van het 1-bits register weer.
- 4.3. Geef een expressie in termen van Boole-algebra voor Q_{next}.

$$Q_{next} = W.D.\overline{Q_c} + \overline{W}.D.Q_c + \overline{W}.D.Q_c + W.D.Q_c.$$

- 4.4. Vereenvoudig deze expressie zover als mogelijk is. Geef hierbij duidelijk aan welke regel(s)/wet(ten) uit de Boole-algebra je hierbij gebruikt.

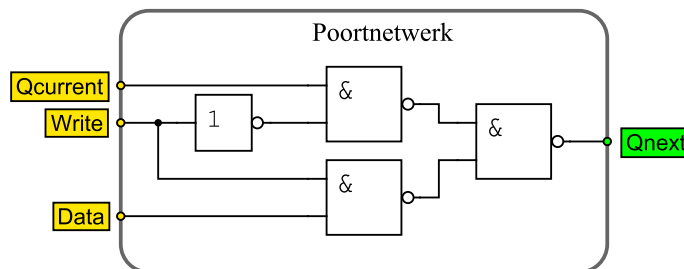
$$Q_{next} = W.D.(\overline{Q_c} + Q_c) + \overline{W}.Q_c.(D + \overline{D}) = W.D + \overline{W}.Q_c$$

- 4.5. Ontwerp de poortschakeling. Hierbij mag alleen gebruik worden gemaakt van NAND-poorten

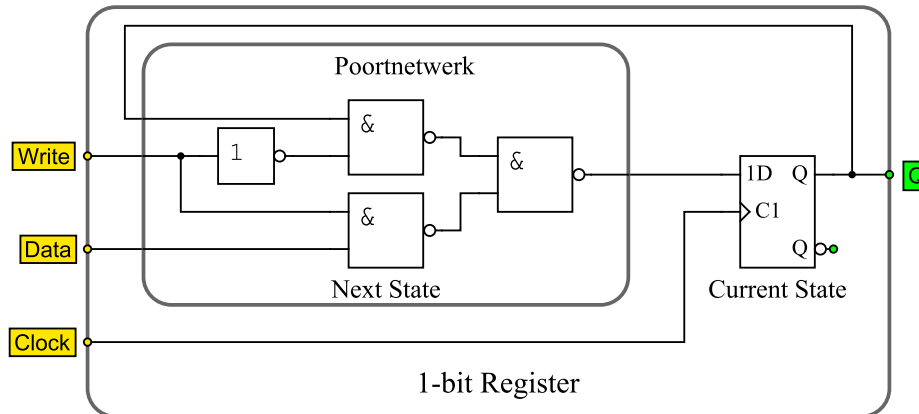
$$\text{met twee ingangen. } Q_{next} = W.D + \overline{W}.Q_c = \overline{\overline{W.D} \cdot \overline{\overline{W}.Q_c}}$$

Write	Data	Q _{current}	Q _{next}	Functie
0	0	0	0	Memory
0	1	0	0	Memory
1	0	0	0	Load 0
1	1	0	1	Load 1
0	0	1	1	Memory
0	1	1	1	Memory
1	0	1	0	Load 0
1	1	1	1	Load 1

Tabel 2a



- 4.6. Geef het schema van het 1-bits register bestaande uit opgave 4.5 en de "Current state" van de automaat weer.

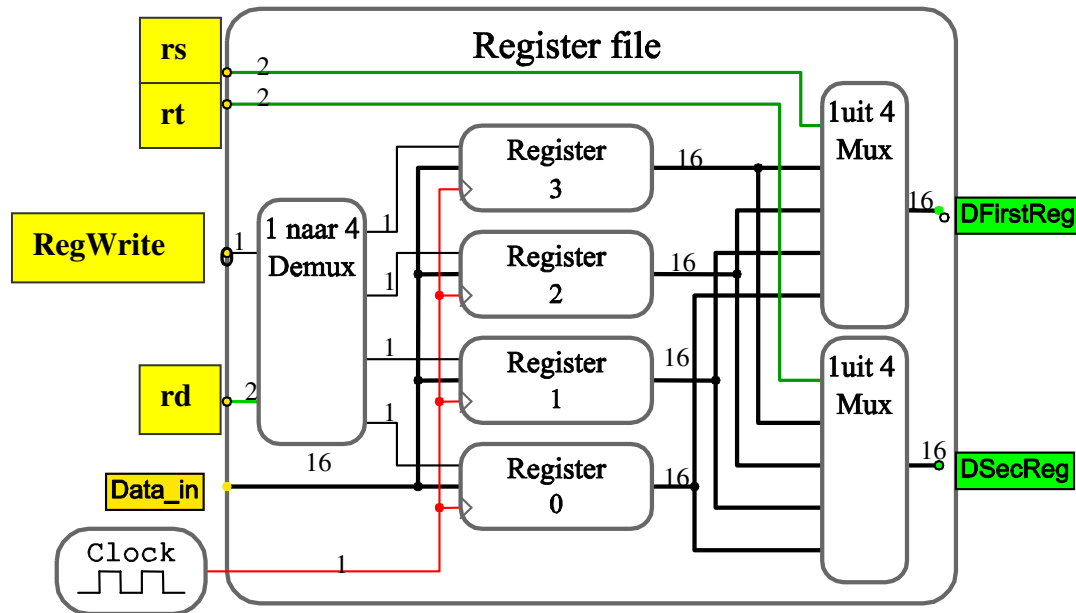


4.7. Wat is de "set-up time" van de ingang Write? Waarvoor is de set-up time van belang?
 De "set-up time" van de ingang Write is 0,3 ns, als we aannemen dat de propagatietijd van een poort 0,1 ns is. De waarde op de write-ingang moet minstens de set-up time (hier 0,3 ns) aanwezig zijn voordat de klokflank de klokingang bereikt.

Vraag 5 Registerfile (9 punten resp 2, 2, 5 punten)

De opdracht is de registerfile zoals weergegeven bij de Harvard architectuur in figuur 4 (volgende opgave) te ontwerpen. De registerfile bestaat uit 16 registers waarvan er 4 zijn weergegeven. Ieder van deze registers kan 16 bits aan gegevens bevatten. Figuur 3 geeft het blokschema van de registerfile weer. Enkele componenten zijn weergegeven door een letter.

- 5.1. Component A zorgt er voor dat het adres dat bij ieder register hoort wordt doorgegeven. Alleen de Write-ingang van het betreffende register wordt dan 1. Wat is een betekenisvolle naam voor component A? *Antwoord: 1 naar 4 Demux.*
- 5.2. Componenten B en C kiezen welk register met de uitgangen DFirstReg resp. DSecReg wordt doorverbonden. Wat is een betekenisvolle naam voor de componenten B en C? *Antwoord: 1 naar 4 Mux.*
- 5.3. Geef alle benodigde verbindingen aan tussen de ingangen, de weergegeven componenten en de uitgangen. Geef van iedere verbinding de bitbreedte aan.



Vraag 6 Harvard machine (16 punten; resp. 2, 2, 2, 3, 2, 5 punten)

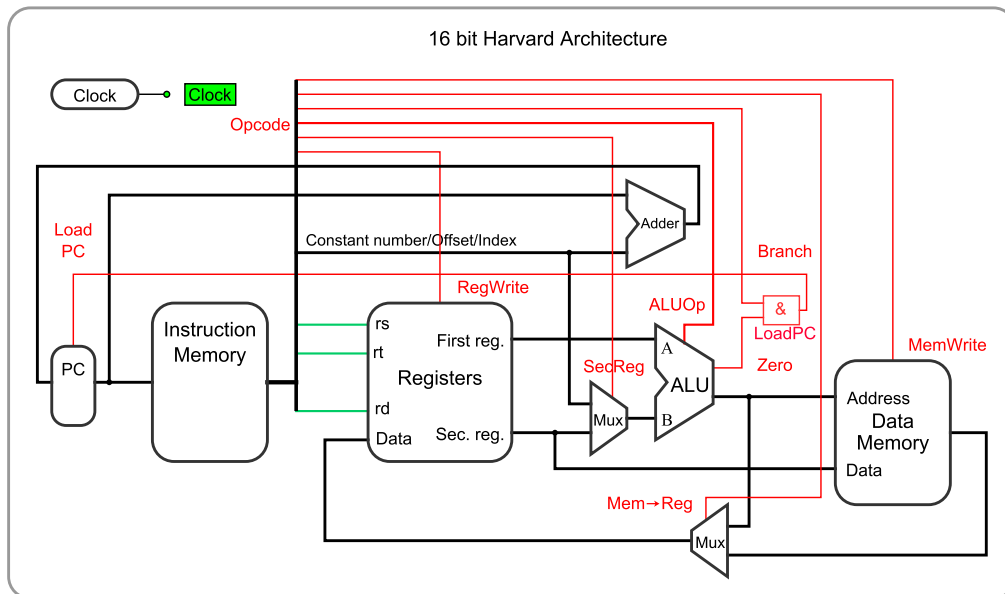
Zie voor de volgende vragen figuur 4. De getallen in figuur 4 zijn in hexadecimale code weergegeven.

- 6.1. Wat is een assembler en wat is machinetaal? *Een assembler is een vertaalprogramma dat een assemblerprogramma omzet naar machinetaal. Machinetaal bestaat uit instructies die de processor direct kan uitvoeren. Iedere instructie wordt als een reeks enen en nullen in het computergeheugen opgeslagen.*
- 6.2. Wat is de functie van de ALU bij instructies van het type Memory?
De ALU berekent het geheugenadres.
- 6.3. Wat is de functie van de ALU bij Branch-instructies?
De ALU bepaalt de waarde van de zero-flag. Deze waarde geeft aan of er een Branch wordt uitgevoerd.
- 6.4. Uit welke fases bestaat het executeren van een ADDI-instructie?
Instruction fetch; instruction decode; execution en write back.
- 6.5. Een ALU heeft slechts één vlag 'Zero' en kan slechts de operaties optellen, aftrekken en bitwise AND uitvoeren. Leg uit hoe je hiermee van twee gegeven gehele getallen het grootste kan bepalen.
Trek het getal B af van het getal A. Als $A > B$ dan is het msb (most significant bit) 0. Als $A < B$ dan is het verschil negatief dus het msb 1. Voer de operatie & (bitwise AND) uit op het verschil en het masker 1000000000000000 (ingeval van een 16 bit machine). De Zero-vlag zal dan alleen 1 zijn als $A < B$.

6.6. De relatie tussen de afkortingen in tabel 3 en de teksten in figuur 4 is: MW = MemWrite; M2R = Mem→Reg; B = Branch; RW = RegWrite; SecReg = SecReg. De ALU kent de volgende operaties: ADD 00: SUB 01; AND 10 en B-doorlaten 11.

Instructieformaat 16 bit Harvardmachine (totaal 35 bits)										
Instructie	Opcode						First Register rs	Second Register rt	Destination Register rd	getal/ offset/ index
	M W	M2 R	B	RW	Sec Reg	ALU				
Aant. bits	1	1	1	1	1	2	4	4	4	16
ADD	0	0	0	1	1	00	rs	rt	rd	x
LOADI	0	0	0	1	0	11	x	x	rd	getal
BZ	0	0	1	0	1	11	x	rt	x	offset
LW	0	1	0	1	0	00	rs	x	rd	index

Tabel 3: Instructieformaat: Harvard-machine



Figuur 4