

Antwoorden & oplossingen: Tentamen Digitale techniek Voor AI studenten

Vraag 1 Ontwerpen van een schakeling (12,5 punten)

S1	S2	a	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Antwoord vraag 1a

Vraag 1a: Hiernaast is de volledige waarheidstabel van de Logic Unit weergegeven.

Vraag 1b: Een expressie voor Y als functie van S1, S2 en a (in termen van Boole algebra) is:

$$Y = \overline{S1.S2.a} + \overline{S1.S2.a} + \overline{S1.S2.a} + S1.S2.a$$

Vraag 1c: Deze expressie kan als volgt worden vereenvoudigd.

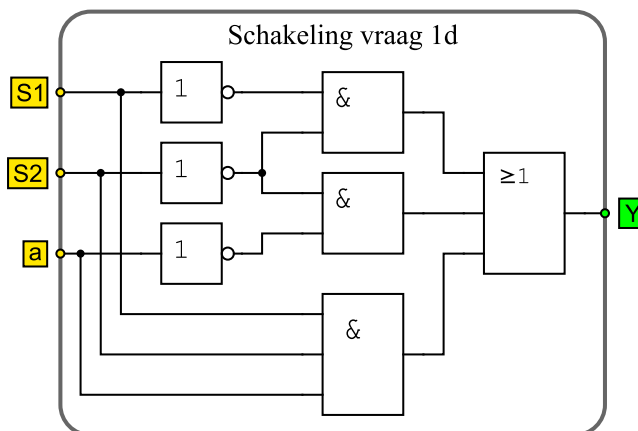
$$Y = \overline{S1.S2.a} + \overline{S1.S2.a} + \overline{S1.S2.a} + S1.S2.a = \overline{S1.S2.(a + a)} + S1.S2.a + S1.S2.a = \overline{S1.S2} + \overline{S1.S2.a} + S1.S2.a = \overline{S2.(S1 + S1.a)} + S1.S2.a = \overline{S2.(S1 + a)} + S1.S2.a = \overline{S2.S1} + \overline{S2.a} + S1.S2.a$$

Achtereenvolgens gebruikt: eerste distributieve wet (buiten haakjes halen); rekenregel voor variabelen; eerste distributieve wet; absorptiewet: $\overline{S1} + S1.a = \overline{S1} + a$.

Vraag 1d: Een implementatie met poorten naar eigen keuze van de vereenvoudigde Boole-expressie is:

Het is mogelijk een oplossing af te leiden met slechts twee poorten, een NAND-poort en een XOR-poort. De afleiding gaat als volgt:

$$Y = \overline{S1.S2} + \overline{S2.a} + S1.S2.a = \overline{S2.(S1 + a)} + S1.S2.a = \overline{S2.(S1.a)} + S1.(S2.a) = \overline{S2} \oplus (S1.a) = \overline{S2} \oplus \overline{S1.a}$$



Vraag 2 ALU, rekenschakelingen (10 punten)

Vraag 2a: Het schema van een schakeling weer waarmee twee getallen van vier-bits kunnen worden opgeteld staat in de syllabus (figuur 2.3 bladz. 22).

Vraag 2b: Zie tabel:

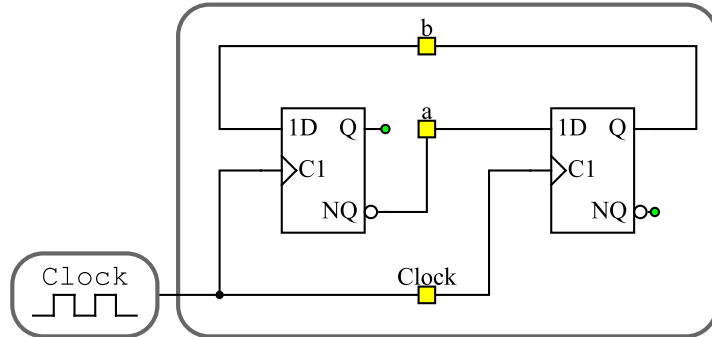
	Kleinste getal	Grootste getal
2b1	0000 _{bin} = 0 _{dec}	1111 _{bin} = 15 _{dec}
2b2	1000 _{two's comp.} = -8 _{dec}	0111 _{two's comp.} = +7 _{dec}
2b3	1000 _{two's comp.} = 8 _{hex}	0111 _{two's comp.} = 7 _{hex}

Hexadecimale code is een verkorte schrijfwijze van de binaire code.

Vraag 2c: Een 'carry flag' treedt op als het bereik van een unsigned integer (getal zonder teken, binaire code) wordt overschreden en 'overflow flag' treedt op als het bereik van een integer (getal met teken, two's complement code) wordt overschreden.

Vraag 3 Sequentiële schakelingen (12 punten)

Vraag 3a: Bij sequentiële schakelingen wordt de toestand van de uitgang bepaald door de logische niveaus van de ingangen en door de voorgeschiedenis d.w.z. de voorafgaande logische toestanden. Dit laatste betekent dat sequentiële schakelingen geheugenwerking bezitten.



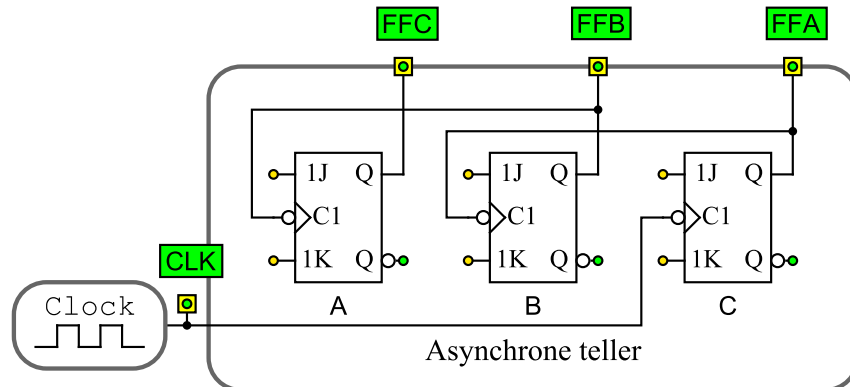
Figuur 1

Vraag 3b: Bij een D-latch is de cel toegankelijk gedurende het '1'-zijn van de E-ingang. Bij een D-flipflop is de cel toegankelijk bij een (positieve of negatieve) flank van de klokingang.

Vraag 3c: Het schema van een drie bits asynchrone teller samengesteld uit JK-flipflops is hieronder weergegeven.

De schakeling werkt als volgt: stel dat de begintoestand van de uitgangen van de drie flipflops '0' is. De JK-flipflops staan in de 'toggle mode' (De J- en K-ingangen zijn niet aangesloten dus $J = K = '1'$). Door een negatieve klokflank wordt flipflop A '1'.

Door een tweede negatieve klokflank wordt flipflop A weer '0' en daardoor Flipflop B '1'. Bij de derde klokflank wordt A weer '1'. Bij de vierde wordt A '0' daardoor B '0' en daardoor flipflop C '1' etc.. De stand van de teller is dus na vier klokpulsen $100_{\text{bin}} = 4$.



Vraag 3d: Een nadeel van bovenstaande teller is dat de tijd tussen een negatieve flank van de klok en het moment dat flipflop C 'omgaat', **de som is van de propagatietijden** van de drie flipflops.

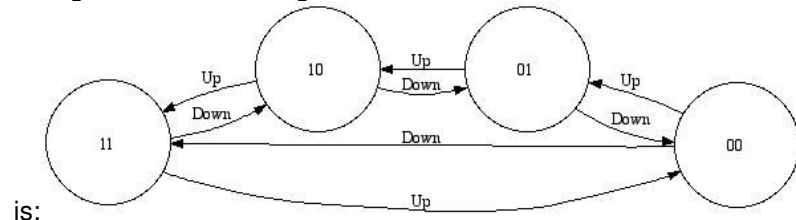
Vraag 4 Finite State System (16 punten)

Een twee-bits teller is voorzien van een Up/Down – ingang. Als $Up/Down = '1'$ dan wordt de teller door iedere positieve klokflank met 1 verhoogd. De telcyclus is dus: 00, 01, 10, 11, 00 etc.

Als $Up/Down = '0'$ dan wordt de teller door iedere positieve klokflank met 1 verlaagd. De telcyclus is dan: 00, 11, 10, 01, 00 etc.

U	H	L	nH	nL
0	0	0	1	1
0	0	1	0	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

Vraag 4a: Het State Diagram van de teller



Vraag 4b: Hierboven is een tabel weergegeven met de volgende toestand van de teller als functie van de huidige toestand en van de toestand van de $\overline{\text{Up/Down}}$ -ingang.

Vraag 4c: Hieronder zijn de expressies voor beide bitjes van de next state weergegeven:

$$nH = \overline{U.H.L} + \overline{U.H.L} + U.\overline{H.L} + U.H.\overline{L}$$

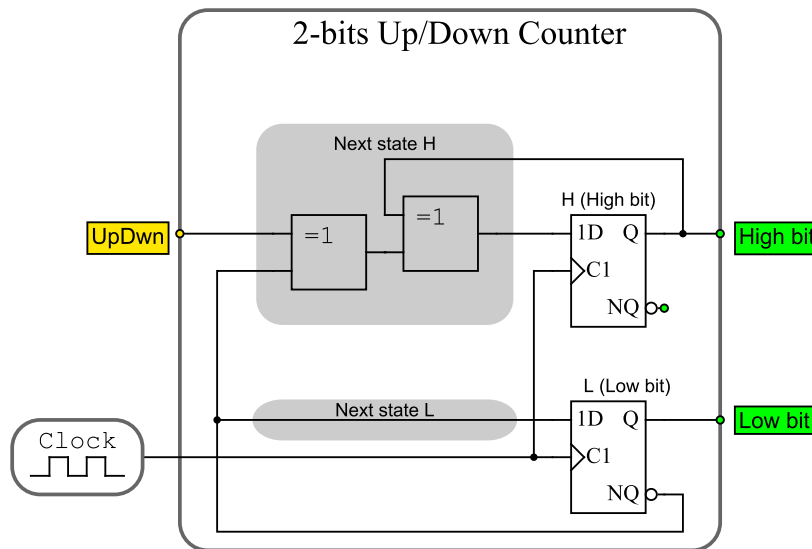
$$nL = \overline{U.H.L} + \overline{U.H.L} + U.\overline{H.L} + U.H.\overline{L}$$

Vraag 4d: Deze expressies kunnen als volgt worden vereenvoudigd en in "XOR-vorm" worden

herschreven:
$$nH = \overline{U.H.L} + \overline{U.H.L} + U.\overline{H.L} + U.H.\overline{L} = \overline{U} \cdot (\overline{H.L} + H.L) + U \cdot (\overline{H.L} + H.L) = \overline{U.H} \otimes L + U \cdot (H \oplus L) = \overline{U \oplus H} \otimes L = U \oplus H \oplus L$$

$$nL = \overline{U.H.L} + \overline{U.H.L} + U.\overline{H.L} + U.H.\overline{L} = \overline{U.L} \cdot (\overline{H} + H) + U.L \cdot (\overline{H} + H) = \overline{U.L} + U.L = \overline{L} \cdot (\overline{U} + U) = \overline{L}$$

Vraag 4e: Een implementatie van de twee-bits $\overline{\text{Up/Down}}$ -counter volgens het model van Moore, bestaande uit XOR-poorten en D-flipflops, is hieronder weergegeven:



Vraag 5 Architectuur (19 punten)

Vraag 5a: De belangrijkste principes van het "stored-program" concept zijn:

1. Instructies worden gerepresenteerd als getallen
2. Programma kunnen worden opgeslagen in het geheugen om gelezen en geschreven te worden als getallen.

Vraag 5b: Een Harvard machine is uit de volgende hoofdcomponenten opgebouwd:

1. PC Program Counter

De PC houdt het adres bij van de instructie die wordt uitgevoerd.

2. Instruction Memory

In het instructiegeheugen is het programma opgeslagen. Ieder geheugenadres bevat één instructie.

3. De register file. Een register file is een 'blok registers', meestal 16 of 32. In de registers worden variabelen van een programma opgeslagen.

4. ALU. Dit is het rekenorgaan van de computer.

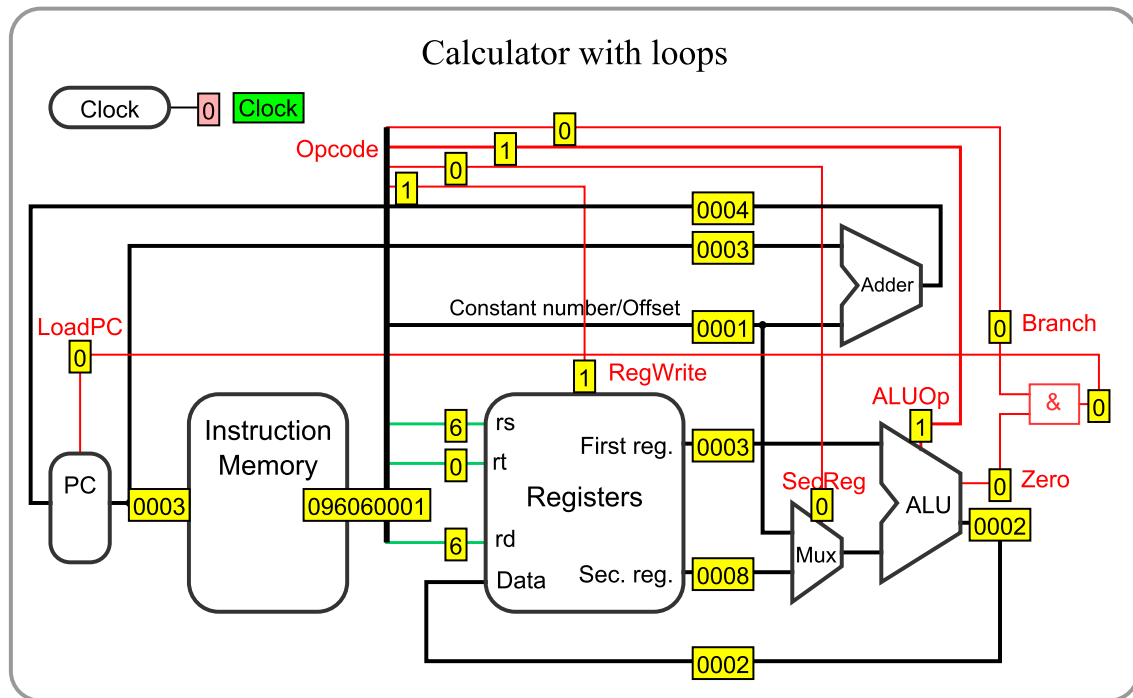
5. Data Memory. Hierin worden de gegevens opgeslagen.

Vraag 5c: Het instructieformaat van de machine afgebeeld in figuur 2 is:

Instructieformaat Loop-machine (totaal 33 bits)									
Instructie	Opcode					First Register rs	Second Register rt	Destination Register rd	Constant getal/ offset
	Bra	Reg Wr.	Mux SecReg	ALU					
				S ₁	S ₀				
Aantal bits	1	1	1	1	1	4	4	4	16

Vraag 5d: Er wordt de instructie: **SUBI \$6, \$6, 1** uitgevoerd.

Toelichting: SecReg = 0 dus constante 0001_{hex} gaat naar ALU. Ook het getal '0003' uit register \$6 gaat naar ALU. ALU-resultaat is 0002 dus ALU trekt af. RegWrite = '1', dus ALU-resultaat wordt naar register \$6 geschreven.



Figuur 2

Vraag 5e: Voor een 'loopje' is een Branch-instructie nodig. Bij een dergelijke instructie wordt het Branch-bit '1'. Als ook aan een tweede voorwaarde, het '1'-worden van het zero-bit van de ALU wordt voldaan, wordt de LoadPC-ingang van de Program Counter '1'. De PC springt naar een nieuwe stand en dus een nieuw adres van het instructiegeheugen. Dit nieuwe adres is de som van het huidige adres (is huidige waarde van de PC) en de "offset" die in het instructiegeheugen is opgeslagen.