

Eindtentamen Digitale Systemen 07/07/2006, 9.00 – 12.00 uur

Het tentamen is **open boek** en bestaat uit 18 multiple choice (MC) vragen en 2 open vragen. De MC-vragen dienen beantwoord te worden op het uitgereikte **MC-formulier**. Enkele aanwijzingen bij het invullen van de MC-formulieren:

- slechts 1 antwoord is het correcte antwoord (NB: a,b,c,d staan door elkaar)
- vul de gekozen vakjes *helemaal* in (liefst met ballpoint, of met potlood)
- vul het formulier pas aan het einde in om fouten te voorkomen
- breng *geen* veranderingen aan op het formulier: haal dan een nieuw formulier
- het onbeantwoord laten van een vraag werkt altijd in uw nadeel
- vergeet niet uw *studienummer* in te vullen (cijfers *en* vakjes!)
- vergeet niet uw handtekening te zetten op het antwoordformulier

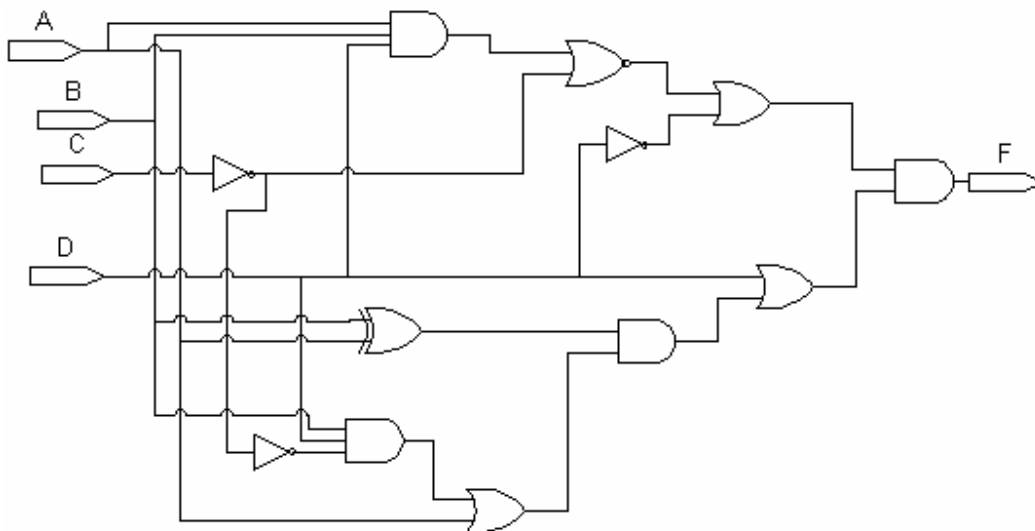
U mag het boek "Contemporary Logic Design", het VHDL boek, en eventuele prints van de **collegeslides** bij u hebben. Verder dus niets! Wij benadrukken dat u tijdens toetsen het tentamen dus **GEEN** gebruik mag maken van oude examens en toetsen. Gebruikt u deze toch dan zijn de tentamen fraude regels van toepassing.

Als een 5 is behaald, wordt nader onderzocht of er op grond van toetsresultaten aanleiding is om het cijfer om te zetten in een 6.

Succes!

A. MC-vragen (gewicht: $18 \times 3.5 \% = 63 \%$)

Vraag 1



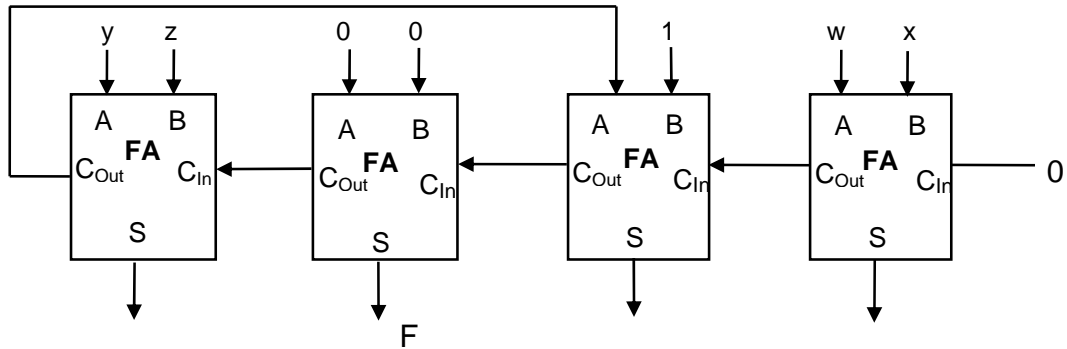
Voor bijgaand circuit gelden de volgende vertragingstijden: INV = 1ns, NAND = NOR = 3ns, AND = OR = 4ns, XOR = 5ns. Oorspronkelijk geldt $A = B = C = D = 0$. Dan wordt $D = 1$.

Welke van de volgende uitspraken is correct?

- a) F wordt 1 na 8 ns en verandert niet meer
- b) Er treedt geen hazard op mbt. F
- c) Er treedt een 1-0-1 hazard op mbt. F
- d) Er treedt een 0-1-0 hazard op mbt. F

Vraag 2

Gegeven bijgaand circuit:



Welke van de volgende vereenvoudigingen is correct?

- a) $F = (w+x)(y+z)$
- b) $F = wx + yz$
- c) $F = x \oplus w + y \oplus z$
- d) geen van bovenstaande antwoorden.

Vraag 3

Welke expressie representeert een hazard-free netwerk?
(we nemen aan dat slechts één ingang van waarde verandert)

- a) $F = BC' + AC'D + A'BC + BD'$
- b) $F = BD' + C'D + A'BD$
- c) $F = BC' + BD' + C'D + A'B$
- d) geen van bovenstaande antwoorden.

		A			
		00	01	11	10
C	00	0	X	1	0
	01	X	X	X	1
	11	0	1	0	0
	10	0	X	1	0
		B			

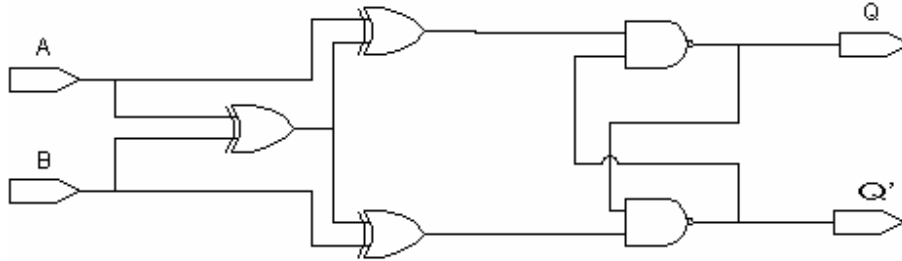
Vraag 4

Gegeven de berekening $C = A - B$ in 2's complement notatie. Er geldt $A = 011010$ en $B = 110101$.
Wat is de waarde van C ?

- a) er is overflow en C kan dus niet bepaald worden.
- b) $C = -27$
- c) $C = 37$
- d) geen van bovenstaande antwoorden.

Vraag 5

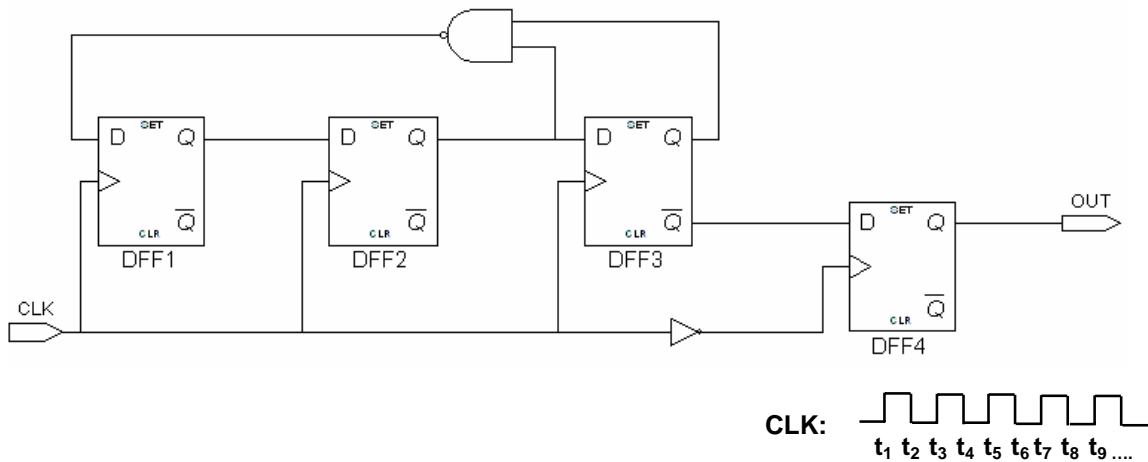
Gegeven bijgaand circuit:



Welke van de volgende uitspraken is correct?

- a) Dit circuit is een latch; $AB = 00$ is de verboden ingangscombinatie
- b) Dit circuit is een latch zonder verboden ingangscombinatie
- c) Dit circuit is geen bruikbaar geheuelement: de Set-combinatie ontbreekt.
- d) $AB = 01$ is de Set-combinatie.

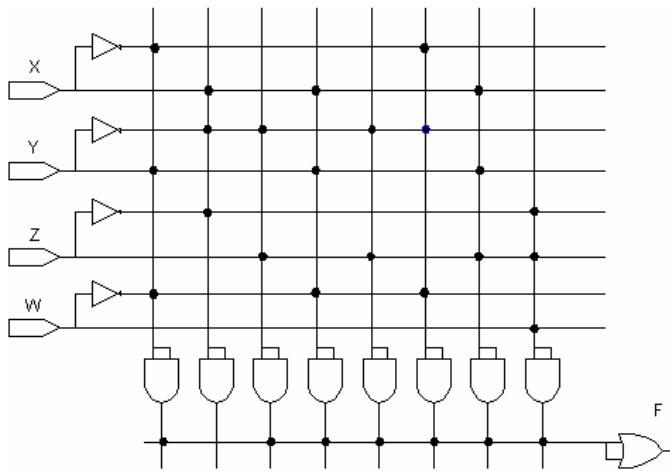
Vraag 6 Gegeven het volgende circuit met positive edge-triggered D filpflops:



Voor bijgaand circuit gelden de volgende vertragingstijden: $INV = 1ns$, $NAND = 2ns$ en CLK periode is $16ns$. Indien op tijdstip t_1 de Q_s van DFF1, DFF2 en DFF3 '1' zijn en Q van DFF4 '0' is, welke van de volgende uitspraken is correct?

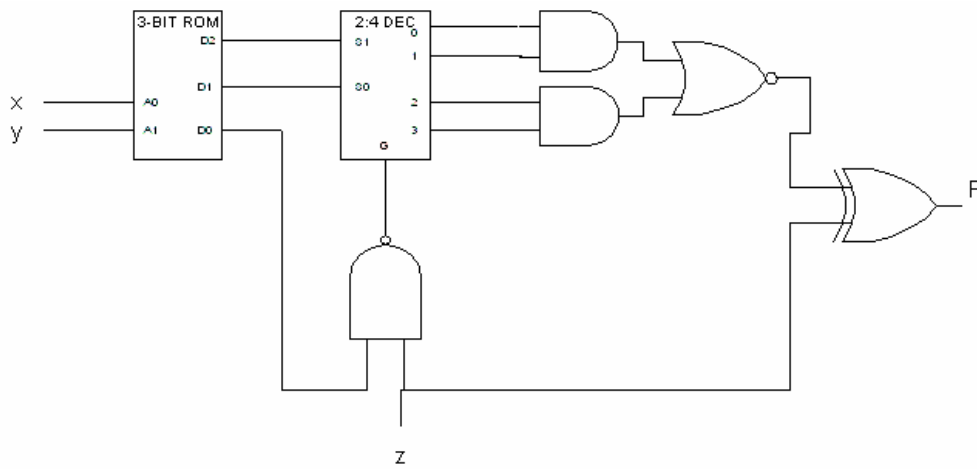
- a) OUT is een periodiek signaal met een periode van 5 clock cycli en wordt voor het eerst 1 na t_7 .
- b) OUT is een periodiek signaal met een periode van 4 clock cycli en wordt voor het eerst 1 na t_5 .
- c) OUT is een periodiek signaal met een periode van 5 clock cycli en wordt voor het eerst 1 na t_6 .
- d) OUT is een periodiek signaal met een periode van 6 clock cycli en wordt voor het eerst 1 na t_7 .

Vraag 7 Welke van de volgende vereenvoudigingen is correct?



- a) $F = (y + x'y')(w' + z)$
- b) $F = (x' + y)w' + (x + y')z$
- c) $F = (y' + xy)(w + z')$
- d) geen van bovenstaande antwoorden.

Vraag 8 Gegeven bijgaand circuit:



De ROM heeft de volgende inhoud:

A1	A0	D2	D1	D0
0	0	0	1	1
0	1	1	0	1
1	0	1	1	1
1	1	1	1	0

Welke van de volgende vereenvoudigingen is correct?

- a) $F = z'$
- b) $F = z(x' + y')$
- c) $F = y(x \oplus z)$
- d) geen van bovenstaande antwoorden.

Vraag 9

Gegeven de volgende Delta I assembly code:

```
                clr    c
                ld     10100101b
                and    01011010b
                bz     test2
test1           add    10000000b
                jp     test3
test2          xor    01111111b
                add    10101111b
                bc     test1
test3          add    00010000b
```

Wat gebeurt met A, Z en C na de executie van de laatste instructie?

- a) $A = BF_H$, $C = 1$, $Z = 0$
- b) $A = BE_H$, $C = 1$, $Z = 1$
- c) $A = BE_H$, $C = 0$, $Z = 0$
- d) $A = BF_H$, $C = 0$, $Z = 1$

Vraag 10

Gegeven de volgende ASM.

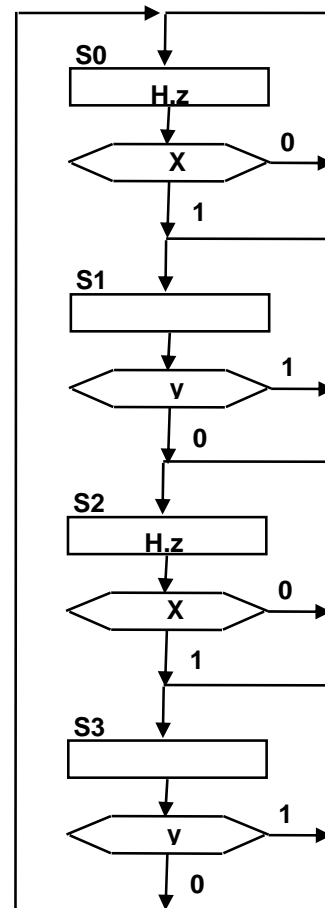
Stel dat we de toestand S_i gecodeerd hebben met 2 variabelen Q_0 , Q_1 volgens:

Q_1	Q_0	state
0	0	S0
0	1	S1
1	1	S2
1	0	S3

Stel dat de ASM met D flip-flops en 2-input NAND gates moet worden gerealiseerd.

Hoeveel NAND gates zijn *minimaal* benodigd voor een netwerk?

- a) 9 NAND
- b) 10 NAND
- c) 11 NAND
- d) geen van de bovenstaande antwoorden geeft de minimale oplossing weer.



Vraag 11

Gegeven zijn de volgende processen:

```
proc1: process (x,y,z) is -- Process 1
    variable var_s1 : std_logic;
    begin
        L1: var_s1 := x or y;
        L2: res <= var_s1 nand z;
    end process;
proc2: process (x,y,z) is -- Process 2
    begin
        L3: sig_s1 <= x or y;
        L4: res <= sig_s1 nor z;
    end process;
```

De waarden van de signalen voordat de processen starten zijn als volgt: x = '1', y = '1', z = '0', sig_s1 = '0'. Wat is de waarde van 'res' na executie van beide processen?

- | | proc1 | proc2 |
|----|-------|-------|
| a. | 0 | 0 |
| b. | 0 | 1 |
| c. | 1 | 0 |
| d. | 1 | 1 |

Vraag 12

We doen de volgende uitspraken over 'functions' en 'procedures' binnen VHDL:

- I. 'Functions' kunnen 'wait' statements bevatten en mogen daarom niet binnen processen met een sensitivity list worden gebruikt.
- II. Dezelfde 'procedure' wordt sequential of concurrent genoemd afhankelijk van zijn plaatsing binnen een architecture body of binnen een process body.

Welke uitspraken zijn juist/onjuist?

- | | | |
|----|------------------------|-------------------------|
| a. | Uitspraak I is onjuist | Uitspraak II is onjuist |
| b. | Uitspraak I is onjuist | Uitspraak II is juist |
| c. | Uitspraak I is juist | Uitspraak II is onjuist |
| d. | Uitspraak I is juist | Uitspraak II is juist |

Vraag 13

Gegeven zijn de volgende twee processen:

```
proc1: process (x, y, z) is
    variable var_s1, var_s2: std_logic;
    begin
        var_s1 := var_s2 and y;
        var_s2 := var_s1 xor z;
        res1 <= var_s1 nand var_s2; -- res is een port
    end process;

proc2: process (x, y, z, sel) is
    begin
        if (sel = "-0") then -- het symbol "-" representeert een don't care
            w <= x xnor y; -- voor std_logic types
        endif;
        v <= (x and y) and z;
    end process;
```

Met betrekking tot synthese en inference van latches, welk van de volgende uitspraken is correct?

- a. Bij proc1 wordt **niet** een latch geinferred en bij proc2 wordt **niet** een latch geinferred.
- b. Bij proc1 wordt **niet** een latch geinferred en bij proc2 wordt **wel** een latch geinferred.
- c. Bij proc1 wordt **wel** een latch geinferred en bij proc2 wordt **niet** een latch geinferred.
- d. Bij proc1 wordt **wel** een latch geinferred en bij proc2 wordt **wel** een latch geinferred.

Gegeven zijn de volgende architectures (vragen 14 t/m 17 hebben hierop betrekking):

```
entity probleem is
  port (a, b, c : in bit; d : out bit);
end entity probleem;
```

```
architecture een of probleem is
begin
  process (a,b) is
  begin
    if (b='0') then d <= a;
    end if;
  end process;
end architecture een;
```

```
architecture twee of probleem is
begin
  process (a,b) is
  begin
    if (a='0') then d <= '1';
    else d <= not(b);
    end if;
  end process;
end architecture twee;
```

```
architecture drie of probleem is
begin
  process (b) is
  begin
    if (b='0') then d <= a;
    end if;
  end process;
end architecture drie;
```

```
architecture vier of probleem is
begin
  process (a,b) is
  begin
    if (a='1') then d <= b;
    else d <= '0';
    end if;
  end process;
end architecture vier;
```

```
architecture vijf of probleem is
begin
  process (a,b,c) is
  begin
    if (b='0') then d <= '0';
    elsif (c'event and c='0') then d <= a;
    end if;
  end process;
end architecture vijf;
```

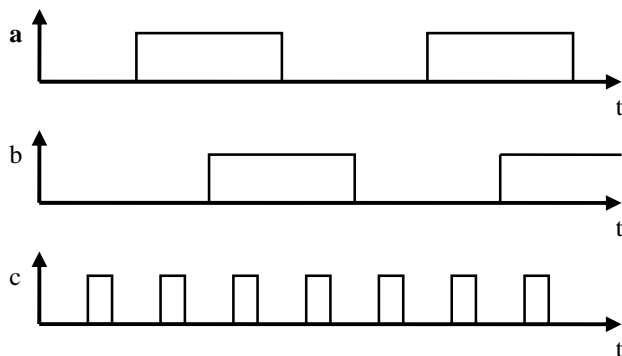
```
architecture zes of probleem is
begin
  process (b,c) is
  begin
    if (c'event and c='1') then d <= a;
    else d <= '0';
    end if;
  end process;
end architecture zes;
```

Vraag 14

Welk van de onderstaande beweringen is juist?

- architecture *drie* beschrijft een latch.
- architecture *vier* beschrijft de combinatorische functie 'a nand b'.
- architecture *twee* beschrijft de combinatorische functie 'a nand b'.
- geen van de drie genoemde mogelijkheden is goed.

FIGUUR 1:



Vraag 15

Gevraagd wordt, als de signalen a, b, en c verlopen zoals in figuur 1, hoe vaak in architecture *een* de statement ' $d \leftarrow a$ ' wordt gescheduled en hoe vaak ' $d \leftarrow b$ ' in architecture *vier* wordt gescheduled: (de initialisatie van de processen op $t=0$ wordt niet meegerekend)

- $d \leftarrow a$ in architecture *een*: 3 keer en $d \leftarrow b$ in architecture *vier*: 4 keer.
- $d \leftarrow a$ in architecture *een*: 4 keer en $d \leftarrow b$ in architecture *vier*: 3 keer.
- $d \leftarrow a$ in architecture *een*: 3 keer en $d \leftarrow b$ in architecture *vier*: 3 keer.
- Geen van de drie genoemde mogelijkheden is goed.

Vraag 16

Welke van de onderstaande beweringen is juist:

- Architecture *zes* beschrijft een flipflop, met signaal c als klok, die werkt op de neergaande flank en een synchrone reset heeft.
- Architecture *vijf* beschrijft een flipflop, met signaal c als klok, die werkt op de neergaande flank en een asynchrone reset heeft.
- Bij de in figuur 1 gegeven signalen wordt $d \leftarrow '0'$ in architecture *zes*, 9 keer gescheduled.
- Geen de drie genoemde beweringen zijn juist.

Vraag 17

Als de signalen a, b, en c verlopen zoals in figuur 1, hoe vaak wordt dan $d \leftarrow a$ gescheduled in architecture *vijf*: (de initialisatie van de processen op $t=0$ wordt niet meegerekend)

- 3 keer
- 4 keer
- 6 keer
- Geen van de drie genoemde mogelijkheden is juist.

Vraag 18

De volgende twee beweringen worden gedaan:

Bewering 1: De volgende twee processen zullen na synthese dezelfde hardware opleveren ondanks het feit dat de simulaties van beide processen verschillende resultaten opleveren:

```
proc1: process (x,y,z) is -- Process 1
  variable var_s1, var_s2 : std_logic;
  begin
    L1: var_s1 := x and y;
    L2: var_s2 := var_s1 xor z;
    L3: res <= var_s1 and var_s2;
  end process;
```

```
proc2: process (x,y,z) is -- Process 2
  begin
    L1: sig_s1 <= x and y;
    L2: sig_s2 <= sig_s1 xor z;
    L3: res <= sig_s1 and sig_s2;
  end process;
```

Bewering 2: De delay en wait statements in VHDL worden door de synthese-compiler gebruikt om de juiste klok signaal met de overeenkomstige klokfrequentie(s) te genereren voor de hardware na synthese.

Welke bewering is juist/onjuist?

- | | | |
|----|-----------------------|-----------------------|
| a. | Bewering 1 is onjuist | Bewering 2 is onjuist |
| b. | Bewering 1 is onjuist | Bewering 2 is juist |
| c. | Bewering 1 is juist | Bewering 2 is onjuist |
| d. | Bewering 1 is juist | Bewering 2 is juist |

B. Open vragen (gewicht: 37%)

Vraag 19 (gewicht 20%)

Ontwikkel een besturingssysteem voor een automatische deur die deel uitmaakt van een deursluis. Het gewenste gedrag van de deur kan als volgt worden omschreven:

- Normaliter (de default situatie) is de deur gesloten.
- Als iemand vlakbij de deur staat dient deze open te gaan, maar alleen als de andere deur van de sluis dicht is.
- Als de deur open staat dient deze open te blijven zolang er iemand in de buurt is. Als er niemand meer in de buurt is, dient de deur dicht te gaan.

Het beschreven gedrag moet u erg bekend voorkomen. Denk maar aan de achterdeur van het ET-gebouw tijdens stormachtig weer!

Om te detecteren dat er personen aanwezig zijn, zit er een sensor op de deur die een signaal **P** genereert dat **1** is zolang er iemand in de buurt is en anders **0**. Bovendien zijn er twee sensoren die de positie van de deur als volgt aangeven:

DO	DC	betekenis
0	0	de deur bevindt zich in een tussenstand (gaat van open naar dicht)
0	1	de deur is dicht
1	0	de deur is open
1	1	de deur bevindt zich in een tussenstand (gaat van dicht naar open)

De status van de andere deur van de sluis is beschikbaar middels het invoersignaal **A**:

- als **A 1** is, dan is de andere deur open of bevindt zich in een tussenstand;
- als **A 0** is, dan is de andere deur dicht.

Ons systeem moet eveneens zijn eigen status doorgeven aan de andere deur door middel van een **Busy**-signaal die dient te worden gegenereerd. De waarden van het **Busy**-signaal moeten voldoen aan dezelfde conventie als het zonet beschreven **A**-signaal.

De regelaar die de deur beweegt, wordt bestuurd door de twee volgende signalen:

Open	Close	betekenis
0	0	de deur blijft in zijn huidige positie
0	1	sluit de deur
1	0	open de deur
1	1	niet toegestaan

- a) (14 p) Kies voor de FSM van de deur besturing een Moore machine. Specificeer het gewenste gedrag mbv. een ASM (dit *moet* een **ASM** zijn!) met gebruikmaking van de eerder beschreven signalen **P**, **DO**, **DC**, **A**, **Busy**, **Close** en **Open**. Deze ASM vormt tevens de basis voor Vraag 19 b). (Ook als uw ASM fout is, wordt uw ASM als uitgangspunt gehanteerd bij de beoordeling).
- b) (6 p) Kies een state assignment die de states S_i afbeeldt op de state-variabelen Q_j . De state-opvolging hoeft niet progressief te zijn. Hoeveel ingangsvariabelen hebben de K-maps van Q_j^+ , **Busy**, **Close** en **Open**? Teken de K-maps *alleen* voor **Busy**, **Close** en **Open**, en leidt de expressies af voor **Busy**, **Close** en **Open**.

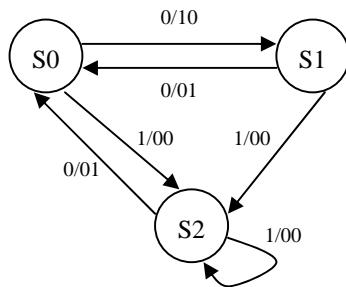
Vraag 20 (gewicht 7%)

Schrijf in VHDL (geef de entity en de architecture) het model van een negative edge-triggered JK flipflop. Een JK flipflop heeft de volgende transitie functie:

Vorige toestand = Q(t)	J(t)K(t)			
	00	01	10	11
0	0	0	1	1
1	1	0	1	0
Volgende toestand = Q(t+1)				

Vraag 21 (gewicht 10%)

Gegeven is de volgende toestandsdiagram van een controller voor verkeerslichten.



Schrijf in VHDL (geef de entity en architecture) het model van deze controller. LET OP: De controller moet zijn geklokt en er moet een reset gegeven kunnen worden waardoor de controller weer naar state S0 teruggaat.

U kunt gebruik maken van de volgende template:

```
library IEEE;
use IEEE.std_logic_1164.all;

entity controller is
port ();
end entity controller;

architecture behavior of controller is
type statetype is (S0, S1, S2); -- declaratie van een nieuwe type
signal state : statetype;
... -- U kunt hier meer signalen/variabelen declareren
begin
label: ... -- architecture body is weggelaten
end architecture behaviour;
```

In deze template wordt gebruik gemaakt van het feit dat binnen VHDL nieuwe types kunnen worden gedeclareerd. Dit betekent dus dat signal *state* de waarden S0, S1, en S2 kan aannemen.

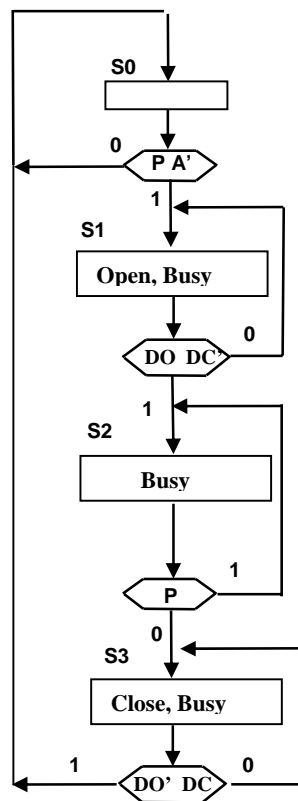
Uitwerkingen Eindtentamen Computersystemen 07/07/2006

MC-vragen:

- | | | | | |
|------|------|-------|-------|-------|
| 1. d | 5. a | 9. d | 13. c | 17. a |
| 2. b | 6. c | 10. c | 14. c | 18. c |
| 3. c | 7. b | 11. d | 15. a | |
| 4. a | 8. a | 12. b | 16. b | |

Vraag 19 a

De ASM ziet er als volgt uit:



Vraag 19 b

State assignment:

state	Q_1	Q_0
S0	0	0
S1	0	1
S2	1	0
S3	1	1

De K-maps van Q_0^+ , Q_1^+ , Open, Close en Busy hebben respectievelijk 6, 6, 2, 2 en 2. De K-maps van Open, Close en Busy (gegeven de eerdere state assignment):

Open: Q_1 :

	0	1
Q_0 :	0	0
	1	1

$\Rightarrow \text{Open} = Q_1' Q_0$

Close: Q_1 :
 0 1
 Q_0 : -----
 0 | 0 | 0 |
 1 | 0 | 1 | \Rightarrow **Close** = $Q_1 Q_0$

Busy: Q_1 :
 0 1
 Q_0 : -----
 0 | 0 | 1 |
 1 | 1 | 1 | \Rightarrow **Busy** = $Q_1 + Q_0$

Vraag 20

```

library IEEE;
use IEEE.std_logic_1164.all;

entity JKflipflop is
port ( in1, in2, clock : in std_logic;
       output : inout std_logic);
end entity controller;

architecture behavior of JKflipflop is
begin
proc: process (clock) is
begin
    if (clock'event and clock = '0') then
        if (in1 = '0' and in2 = '0') then
            output <= output;
        elsif (in1 = '1' and in2 = '0') then
            output <= '1';
        elsif (in1 = '0' and in2 = '1') then
            output <= '0';
        elsif (in1 = '1' and in2 = '1') then
            output <= not(output);
        end if;
    end if;
end process proc;
end architecture behaviour;

```

Vraag 21

```
library IEEE;
use IEEE.std_logic_1164.all;

entity controller is
port ( reset, clk, x : in std_logic;
      z : out std_logic_vector (1 downto 0));
end entity controller;

architecture behavior of controller is
type statetype is (S0, S1, S2); -- declareren van een nieuwe type
signal state, next_state : statetype;
begin
comb_process:
  process (state, x) is
  begin
  case state is
    when S0 =>
      if x = '0' then
        next_state <= S1;
        z <= '10';
      else
        next_state <= S2;
        z <= '00';
      end if;
    when S1 =>
      if x = '0' then
        next_state <= S0;
        z <= '01';
      else
        next_state <= S2;
        z <= '00';
      end if;
    when S2 =>
      if x = '0' then
        next_state <= S0;
        z <= '01';
      else
        next_state <= S2;
        z <= '00';
      end if;
  end case;
  end process comb_process;

clk_process:
  process
  begin
  wait until (rising_edge(clk));
  if reset = '1' then
    state <= S0;
  else
    state <= next_state;
  end if;
  end process clk_process;
end architecture behaviour;
```

-0-0-0-0-0-0-0-