

Tentamen Digitale Systemen (EE1410) 6 juli 2012, 9.00 – 12.00 uur

Dit tentamen is een **open boek** tentamen en bestaat uit 18 multiple choice (MC) vragen (63%) en 5 open vragen (37%). 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 op het antwoordvel)
- vul de gekozen vakjes *helemaal* in (liefst met ballpoint, of met potlood)
- vul het formulier pas aan het einde in om fouten te voorkomen
- *geen* veranderingen aanbrengen: 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!) en uw handtekening te plaatsen.

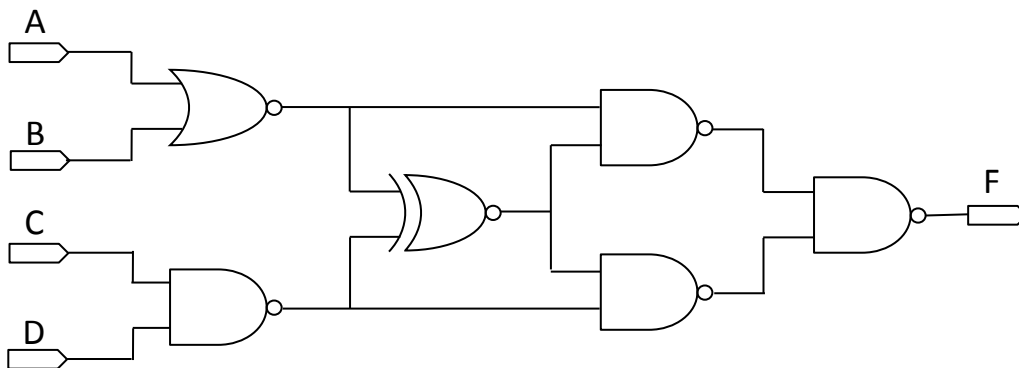
U mag het boek "Fundamentals of Digital Logic with VHDL Design", het VHDL boek (of alternatieve boeken hiervoor), het document RTL_Delta1, prints van het **college** slides en eventueel een niet-programmeerbare rekenmachine bij u hebben. Verder dus niets! Wij benadrukken dat u tijdens toetsen het tentamen dus **GEEN** gebruik mag maken van oude tentamens en toetsen (uitgezonderd tentamen en toets vragen die op de college slides staan). Gebruikt u deze toch dan zijn de tentamen fraude regels van toepassing.

Op grond van toetsresultaten kan het tentamencijfer met maximaal 1 punt verhoogd worden.

Succes!

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

Vraag 1



Voor bovenstaand circuit gelden de volgende vertragingstijden: NAND = 10 ps, NOR = 12 ps, XNOR = 20 ps.

Oorspronkelijk geldt $B = 0$ en $A = C = D = 1$.

Dan wordt $A = 0$. Welke van de volgende uitspraken is correct?

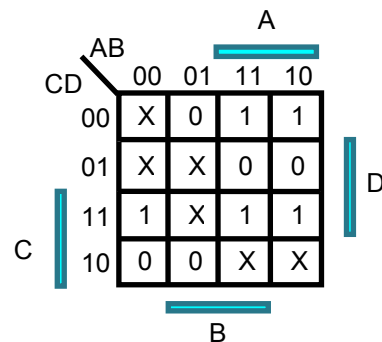
- Er treedt een 0-1-0 hazard op mbt. F
- Er treedt een 1-0-1 hazard op mbt. F
- F is bij aanvang 0, wordt dan 1 en verandert niet meer
- Geen van bovenstaande antwoorden

Vraag 2

Gegeven de nevenstaande K-map (X betekent don't care).

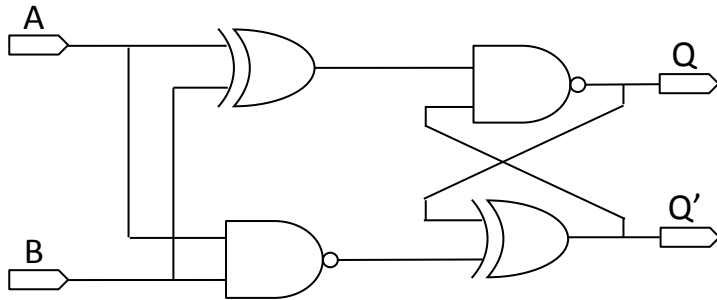
Wat is de bijbehorende minimale som van producten ?

- $A.C'.D' + C.D$
- $A.D' + A'.D$
- $A.D' + C.D$
- $A.D' + A'.D + C.D$



Vraag 3

Gegeven bijgaand circuit:



Welke van de volgende uitspraken is correct?

- a. Dit circuit is geen bruikbaar geheugenelement: de Set-combinatie ontbreekt.
- b. Dit circuit is geen bruikbaar geheugenelement: de Reset-combinatie ontbreekt
- c. Dit circuit is bruikbaar als een latch; $AB = 11$ is de verboden ingangscombinatie
- d. Dit circuit is bruikbaar als een latch: $A=1$ en $B=0$ is de ingangscombinatie voor onthouden

Vraag 4

Gegeven de berekening $C = A - B$ in 2's complement notatie. Er geldt $A = 010011$ en $B = 110100$.
Wat is de waarde van C ?

- a. -6
- b. 29
- c. 30
- d. Geen van bovenstaande antwoorden.

Vraag 5

Van nevenstaand schema zijn voor de poorten de volgende tijden gegeven:

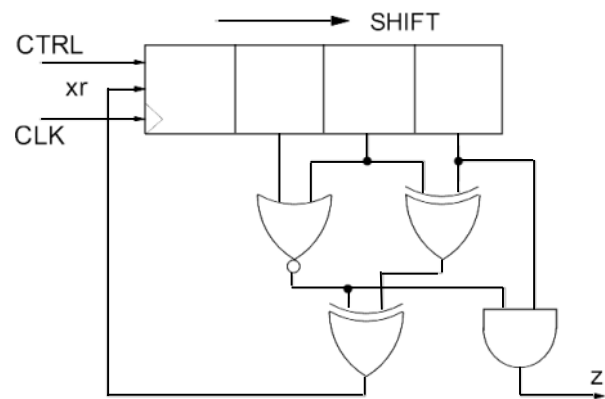
delay XOR: 11 ps
delay NOR: 15 ps
delay AND: 9 ps

en voor het schuifregister (bestaande uit D flip-flops):

setup tijd: 4 ps
hold tijd: 2 ps
omslagtijd: 10 ps

Wat is de minimale klokperidotijd waarbij het systeem nog correct kan werken?

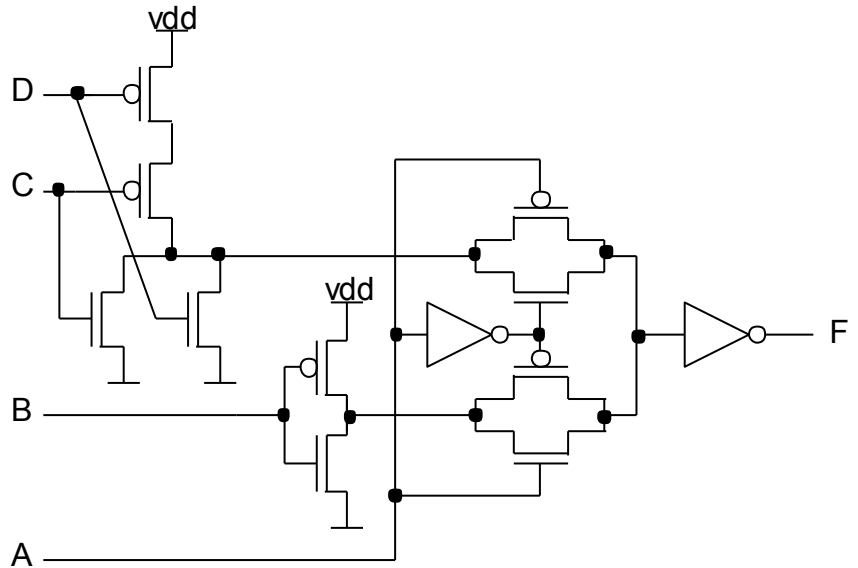
- a. 36 ps
- b. 38 ps
- c. 40 ps
- d. 42 ps



Vraag 6

Gegeven nevenstaande MOS schakeling.
Wat is een uitdrukking voor F ?

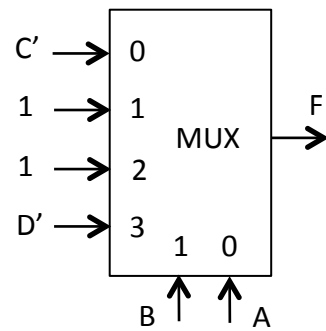
- a. $A.B + A'.C.D$
- b. $A'.B' + A.(C + D)'$
- c. $A.B + A'.(C + D)$
- d. $A'.B + A.C.D$



Vraag 7

Gegeven is nevenstaand schema.
Bepaal de functie $F(A,B,C,D)$ van de schakeling en schrijf F als een product van sommen.

- a. $(A + B + C').(A' + B' + D')$
- b. $(A' + B + C').(A + B' + D')$
- c. $(A + B' + C').(A' + B + D')$
- d. $(A' + B' + C').(A + B + D')$



Vraag 8

Gegeven de volgende Delta I assembly code:

```

set    c
ld     00001100b
add    00000011b
st     R3
bc     labela
xor    11110000b
labela: add 00100000b
bc     labelb
add    R3
labelb: add R3
    
```

Wat is de waarde van A na de executie van de laatste instructie?

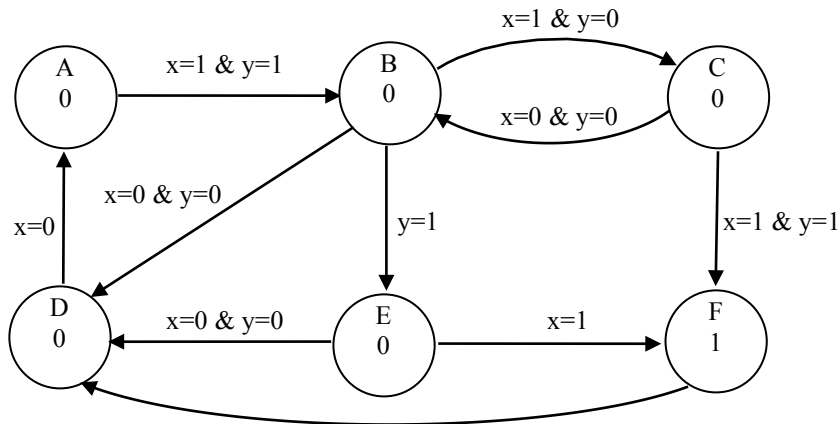
- a. $A = 2F_H$
- b. $A = 30_H$
- c. $A = 21_H$
- d. $A = 11_H$

Vraag 9.

De formule $F = A.B'.C' + A.B.D + A.C'.D' + B.C'.D + A'.B.C.D$ is te vereenvoudigen tot:

- a. $A + B.D$
- b. $A.C' + B.D$
- c. $A.D + B.C.D$
- d. Het juiste antwoord staat er niet bij.

Vraag 10.

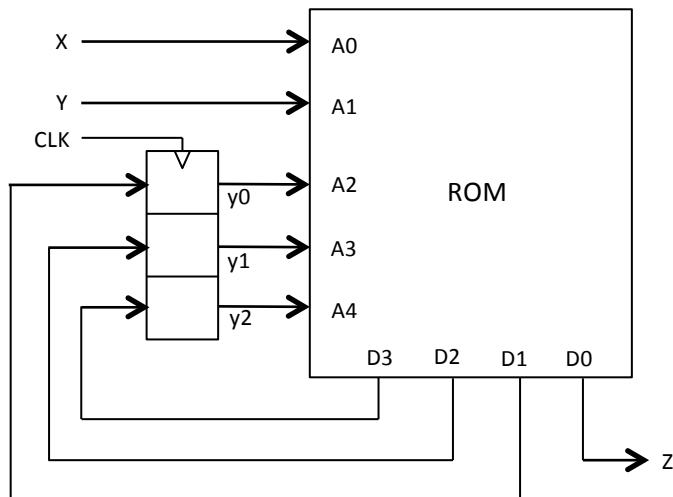


Gegeven bovenstaand toestandsdiagram van een Moore machine met ingangen x en y, en een uitgang z. (Indien voor een bepaalde toestand niet voor alle ingangscombinaties een overgang beschreven is, betekent dit dat het systeem voor de overblijvende ingangscombinaties in de huidige toestand blijft.)

Vertrekkende vanuit een willekeurige toestand: welke ingangstreeks geeft een 1 aan de uitgang ?

- a. xy: 00 → 01 → 01 → 11
- b. xy: 10 → 01 → 01 → 10
- c. xy: 00 → 01 → 00 → 10
- d. xy: 10 → 01 → 00 → 11

Vraag 11.



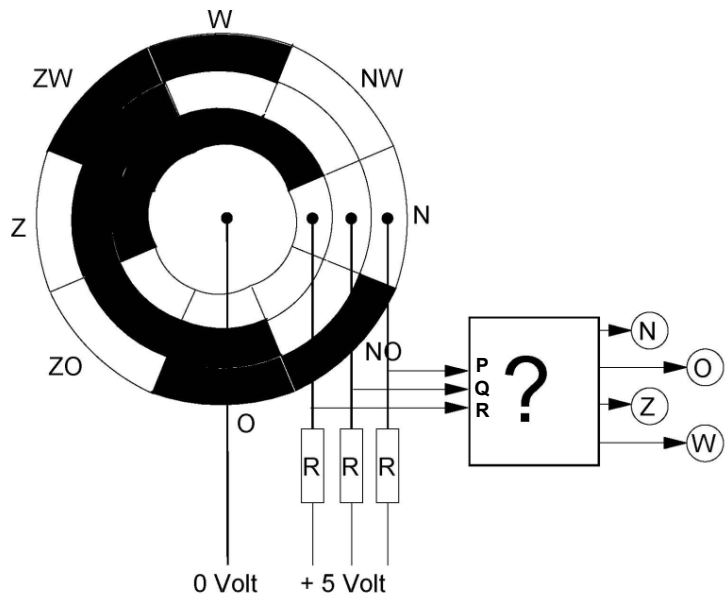
	y2	y1	y0
A	0	0	0
B	0	0	1
C	0	1	0
D	0	1	1
E	1	0	0
F	1	0	1

Als de toestands codering uit de tabel rechts geldt voor het toestandsdiagram van vraag 10, en een realisatie m.b.v. een ROM wordt uitgevoerd zoals hierboven links is aangegeven, wat is dan de inhoud van de ROM op adres 10001 ?

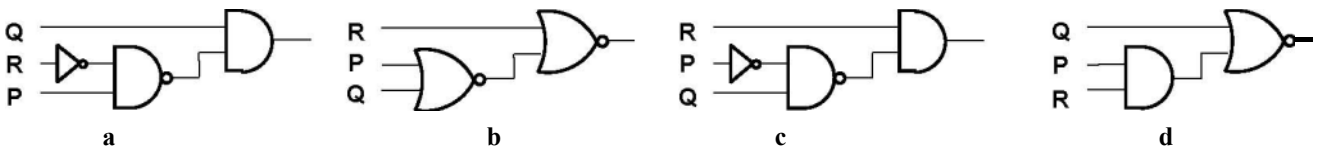
- a. 0101
- b. 0110
- c. 1000
- d. Geen van bovenstaande antwoorden.

Vraag 12.

Op een as van een windwijzer is een elektrisch geleidend codeschijfje bevestigd. Als een geleidende sleepveer op het witte, geleidende deel van het schijfje rust, zal er een logische "0" worden geproduceerd, maar als de veer op het isolerende zwarte deel rust, zal er een logische "1" worden afgegeven. Op deze manier worden 3 signalen P, Q en R gemaakt. De stand van de as, die één der 8 windrichtingen aangeeft, moet worden zichtbaar gemaakt met behulp van de lampjes N(oord), O(ost), Z(uid) en W(est). Met behulp van een code-omzetter worden uit de signalen P, Q en R de stuursignalen voor de 4 lampjes gemaakt (aan = 1 en gedoofd = 0).



Welk van onderstaande schakelingen kan worden gebruikt voor het aansturen van het lampje N ?



Vraag 13.

Gegeven de volgende VHDL code:

```
library ieee;
use ieee.std_logic_1164.all;

entity mux_network is
  port (
    x0 : in  std_logic;
    x1 : in  std_logic;
    x2 : in  std_logic;
    x3 : in  std_logic;
    z  : out std_logic);
end mux_network;

architecture structural of mux_network is
  component mux
    port (
      x0 : in  std_logic; -- de 0-de ingang van de multiplexer
      x1 : in  std_logic; -- de 1-de ingang van de multiplexer
      s  : in  std_logic; -- selector-ingang van de multiplexer
      o  : out std_logic); -- de uitgang van de multiplexer
    end component;
  signal o1, o2 : std_logic;
begin
  mux_1 : mux port map (x0 => '0', x1 => x3, s => x2, o => o1);
  mux_2 : mux port map (x0 => '0', x1 => o1, s => x0, o => o2);
  mux_3 : mux port map (x0 => o2, x1 => x3, s => x1, o => z);
end structural;
```

Welke functie wordt hiermee geïmplementeerd?

- $f(x_3, x_2, x_1, x_0) = \sum m(6, 7, 13, 14, 15)$
- $f(x_3, x_2, x_1, x_0) = \sum m(10, 11, 13, 14, 15)$
- $f(x_3, x_2, x_1, x_0) = \sum m(2, 3, 5, 6, 7)$
- $f(x_3, x_2, x_1, x_0) = \sum m(4, 12, 13, 14, 15)$

Vraag 14.

Gegeven zijn de volgende processen:

```
proc1:      process (x,y,z) is -- Process 1
             variable var_s1 : std_logic;
             begin
                 L1: var_s1 := x and y;
                 L2: res1 := var_s1 nor z;
             end process;

proc2:      process (x,y,z) is -- Process 2
             begin
                 L3: sig_s1 <= x and y;
                 L4: res2 <= 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?

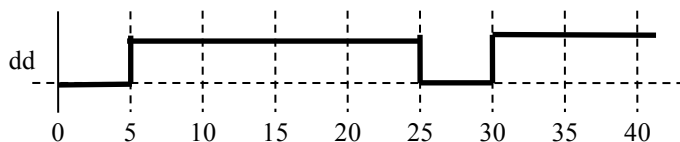
	res1	res2
a.	0	0
b.	1	0
c.	0	1
d.	1	1

Vraag 15.

```
entity boolean_test is
end boolean_test;

architecture gedrag of boolean_test is
    component boolean_functie is
        port (a, b, c : in bit; uit : out bit);
    end component;
    signal aa, bb, cc, dd ; bit;
begin
    N1: boolean_functie port map (aa, bb, cc, dd);
    aa <= '1', '0' after 5 ns, '1' after 15 ns, '1' after 30 ns, '1' after 35 ns, '0'
        after 40 ns;
    bb <= '0', '1' after 10 ns, '0' after 20 ns, '1' after 35 ns;
    cc <= ... - deze regel moet nog worden ingevuld
end gedrag;
```

De component boolean_functie is niet gegeven in VHDL, maar wel gegeven als een product van maxtermen $F(a,b,c) = \Pi M(1,2,5,7)$. Verder is het de bedoeling dat de component de volgende uitgang realiseert:



Welke van de vier onderstaande assignments moet worden ingevuld als waveform van signaal cc om de gevraagde signaal dd te genereren?

- $cc \leq '0', '1' \text{ after } 15 \text{ ns}, '0' \text{ after } 20 \text{ ns}, '1' \text{ after } 25 \text{ ns}, '0' \text{ after } 30 \text{ ns}, '1' \text{ after } 35 \text{ ns};$
- $cc \leq '1', '0' \text{ after } 10 \text{ ns}, '1' \text{ after } 15 \text{ ns}, '0' \text{ after } 25 \text{ ns}, '1' \text{ after } 30 \text{ ns}, '0' \text{ after } 40 \text{ ns};$
- $cc \leq '1', '0' \text{ after } 5 \text{ ns}, '1' \text{ after } 10 \text{ ns}, '0' \text{ after } 15 \text{ ns}, '1' \text{ after } 25 \text{ ns}, '0' \text{ after } 30 \text{ ns}, '1' \text{ after } 40 \text{ ns};$
- geen van de bovenstaande antwoorden.

De volgende informatie heeft betrekking op opgaven 16-17. Hieronder is een entity gegeven met daarbij drie verschillende architectures:

```

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

architecture one of probleem is
begin
L1:  process (a, b)
      begin
          if (a = '0') then d <= '1';
          else d <= not b;
          end if;
      end process;
end one;

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

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

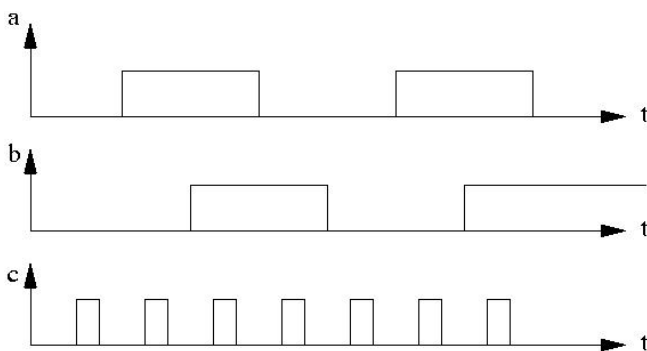
```

Vraag 16.

Welke van de volgende uitspraken is correct?

- architecture two beschrijft een latch.
- architecture one beschrijft de combinatorische functie a nand b.
- architecture three beschrijft de combinatorische functie a and b.
- geen van de drie genoemde mogelijkheden is goed.

In onderstaande Figure 1 zijn de binaire signalen a, b en c gegeven als functie van de tijd.



Vraag 17.

Gevraagd wordt, als signalen a, b en c verlopen zoals in Figure 1, hoe vaak in architecture one de statement “ d<= not b ” wordt gescheduled en hoe vaak “ d<= '0' ” in architecture two wordt gescheduled.

- a. d<= not b in architecture one: 4 keer;
d<= '0' in architecture two: 3 keer.
- b. d<= not b in architecture one: 4 keer;
d<= '0' in architecture two: 4 keer.
- c. d<= not b in architecture one: 3 keer;
d<= '0' in architecture two: 4 keer.
- d. geen van de drie genoemde mogelijkheden is goed.

Vraag 18.

Gegeven het volgende VHDL process:

```
process
begin
    out1 <= s1 and s4 after 4 ns;
    out2 <= not(s3) after 10 ns;
    wait for 20 ns;
end process;
```

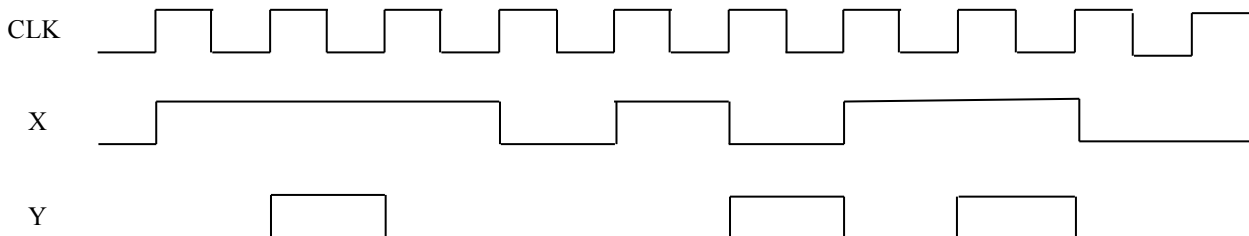
Alle signalen zijn van het type **std_logic**. Neem aan dat alle signalen de waarde '1' hebben. Als het process op 0 (nul) ns eenmaal is gestart, op welk tijdstip zal dit process opnieuw worden gestart (uitgedrukt in simulatietijd)?

- a. geen tijd (0 ns)
- b. een delta delay (Δ ns)
- c. de delay van de wait statement (20 ns)
- d. de som van alle delay statements inclusief wait statement (34 ns)

B. Open vragen (gewicht: 37%)

Vraag 19 (7 %)

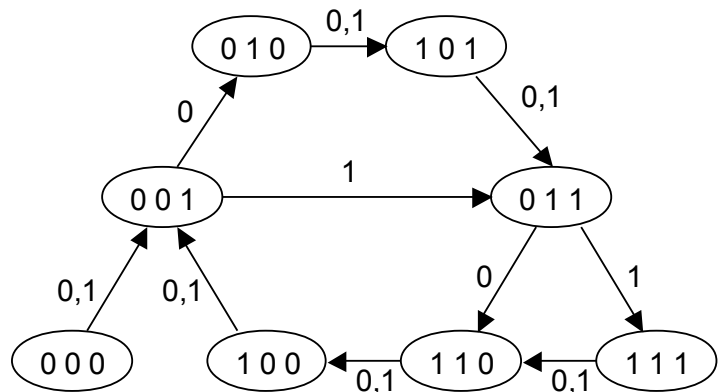
Gegeven eeningangssignaal X dat gesynchroniseerd is met de klok. D.w.z. het signaal verandert alleen van waarde direct na de actieve klokflank. Het signaal kan 1 klokperiode lang zijn waarde behouden, maar kan ook meerdere klokperiodes achter elkaar dezelfde waarde behouden. U wordt gevraagd om d.m.v. een Finite State Diagram (FSD) een schakeling te ontwerpen met een uitgang Y welke *gedurende 1 klokperiode* een waarde 1 afgeeft wanneer hetingangssignaal X van 0 naar 1 is gegaan. Zie onderstaand voorbeeld. Het FSD moet een Finite-State Machine van het type Moore beschrijven, en zo weinig mogelijk toestanden gebruiken.



Vraag 20 (10 %)

Voor een schakeling met eeningangssignaal S is het gecodeerde toestands-diagram hiernaast weergegeven. De schakeling wordt opgebouwd met 3 T flipflops C, B en A (de uitgangen van de T flipflops corresponderen in deze volgorde met de getoonde toestand bits).

- Geef een Karnaugh map voor de nieuwe waarde van de uitgang van T flipflop A na de volgende actieve klokflank.
- Bepaal een logische expressie (minimale som van producten) voor de aansturing van de ingang van T flipflop A.



Vraag 21 (4 %)

Schrijf in de assembly taal van de Delta 1 processor een programma dat springt naar label 'R1neg' indien de inhoud van register R1 negatief is (2's complement representatie). Het kan in 4 instructies.

Vraag 22 (6 punten)

Schrijf de VHDL code op van een D-flipflop met gecombineerd een asynchrone reset-sigitaal en een synchrone set-sigitaal. De entity is gedefinieerd als:

```
entity Dff is
  port (ingang, reset, set, clk: in bit;
        uitgang: out bit);
end Dff;
```

Vraag 23 (10 punten)

Maak gebruik van de in Vraag 22 geïntroduceerde entity Dff en bouw met dit element een 3-bit rotator (bit3_rotator) rotator (d.w.z. een schuifregister waarbij de beide uiteinden met elkaar zijn verbonden – dus bij naar links schuiven wordt de hoogste bit in de laagste positie ingeschoven en bij het naar rechts schuiven de laagste bit naar de hoogste positie ingeschoven) die ook de mogelijkheid heeft om een nieuwe 3-bit waarde in te laden als een load-sigitaal hoog is. Hou tevens rekening met het volgende:

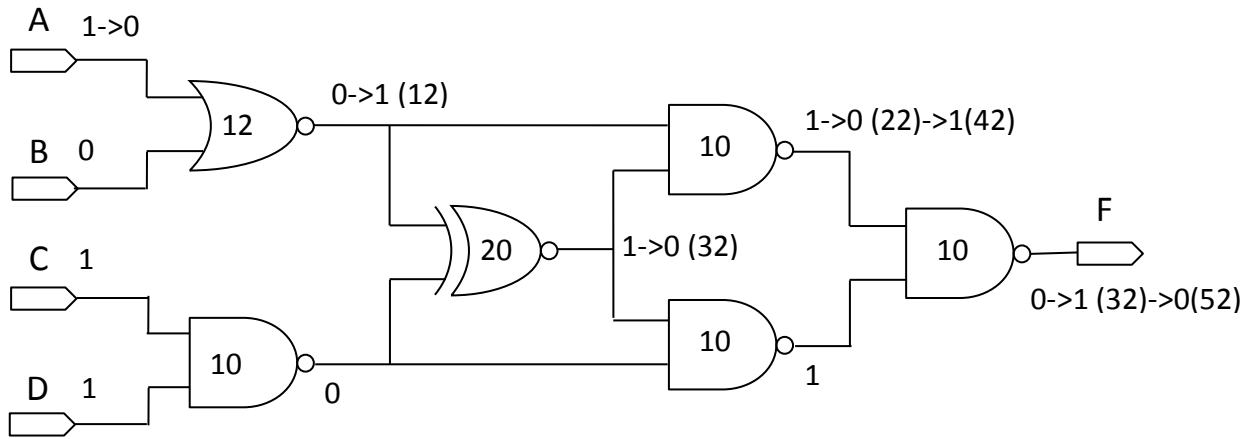
- vergeet niet de uitgang van de D flip-flops ook zichtbaar te maken als een port van de bit3_rotator entity.
- geef duidelijk aan welke ingangssignalen je nodig hebt en hoe je deze hebt benoemd in je VHDL code.
- vergeet de component declaratie van Dff entity niet.
- het is niet toegestaan om processen te gebruiken.
- HINT: denk aan de multiplexer en hoe deze kan worden gebouwd met een combinatorisch netwerk.
- je moet gebruik maken van de volgende entity beschrijving:

```
entity bit3_rotator is
  port (links, niets, rechts, load, reset, clk: in bit;
        -- van de signalen links, niets, rechts, load worden
        -- aangenomen dat nooit twee tegelijk '1' kunnen zijn.
        ingang : in bit_vector (2 downto 0);
        uitgang: out bit_vector (2 downto 0));
end bit3_rotator;
```

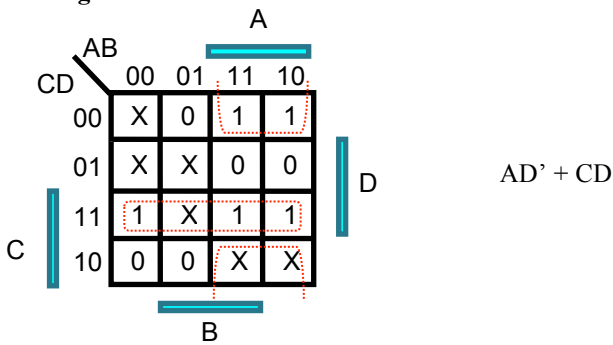
Uitwerkingen Tentamen Digitale Systemen (EE1410) 6 juli 2012

MC-vragen:

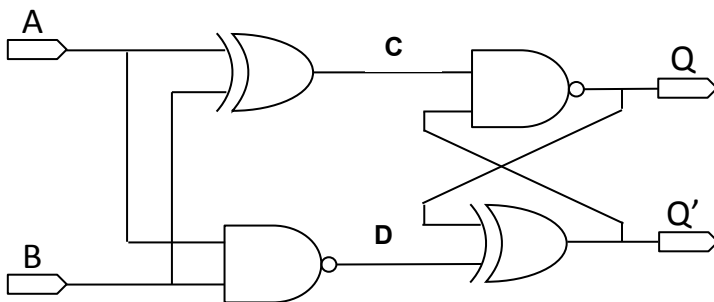
Vraag 1: a



Vraag 2: c



Vraag 3: b



A	B	C	D	Q	Q'	
0	0	0	1	1	0	set
0	1	1	1	Q	Q'	hold
1	0	1	1	Q	Q'	hold
1	1	0	0	1	1	verboden

Vraag 4: d

$A = 010011 = 19$ en $B = 110100 = -12$ dus $C = 19 - (-12) = 31$

Vraag 5: c

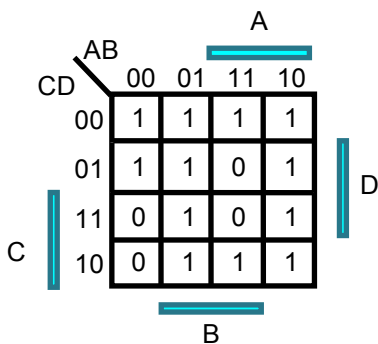
Maximale delay van het combinatorische gedeelte = $t(\text{NOR})+t(\text{XOR}) = 15 + 11 = 26 \text{ ps}$
 De minimale klokperiode is dan: $t_{\text{OMSLAG}} + t_{\text{COMB}} + t_{\text{SU}} = 10 + 26 + 4 = 40 \text{ ps}$

Vraag 6: c

De rechter helft van de schakeling vormt met transmission gates een selector, waarbij wanneer $A=1$ de linker onderschakeling wordt geselecteerd (een invertor voor signaal B) en wanneer $A=0$ de linker boven schakeling (een NOR poort voor C en D). Aangezien de uitgang van de selector geïnverteerd wordt doorgegeven, wordt dan respectievelijk B of $(C + D)$ doorgegeven.
 Dus $F = A \cdot B + A' \cdot (C + D)$

Vraag 7: a

Men kan door gebruik te maken van de MUX-expressie de volgende logische expressie herleiden:
 $F = C' \cdot B' \cdot A' + 1 \cdot B' \cdot A + 1 \cdot B \cdot A' + D' \cdot B \cdot A$
 Vul dit in in een Karnaugh-map om de maxtermen te vinden:

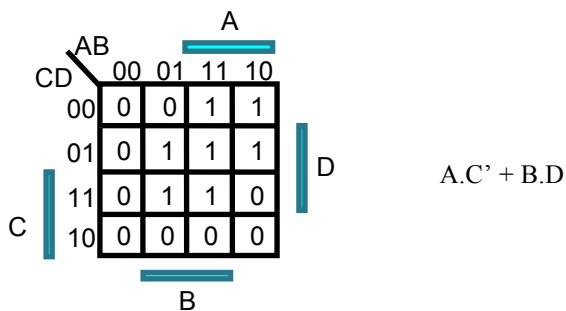


$F = (A + B + C')(A' + B' + D') \Rightarrow$ antwoord a

Vraag 8: d

		A	C
	set c	?	1
	ld 00001100b	00001100b	1
	add 00000011b	00010000b	0
	st R3	00010000b	0
	bc labela	00010000b	0
	xor 11110000b	11100000b	0
labela:	add 00100000b	00000000b	1
	bc labelb	00000000b	1
labelb:	add R3	00010001b	0

Vraag 9: b



Vraag 10: a

Kijk, beginnende bij toestand F (uitgang is 1), hoe je daar terecht kan komen.
 Antwoord a beschrijft een mogelijkheid hoe dat kan, namelijk via toestanden C B E E F.

Vraag 11: d

Adres 10001 wil zeggen: huidige toestand is E (100). ingangswaarden y en x zijn 0 en 1. Uit het toestandsdiagram volgt dan: de volgende toestand is F (101) en de uitgangswaarde z is 1, oftewel D3,D2,D1,D0 = 1011.

Vraag 12: d

RQP		NANDa a	NORb b	NANDc c	ANDd d
000	\overline{N}	1	0	1	0
100	\overline{NW}	1	0	1	1
101	\overline{W}	1	0	0	1
111	ZW	1	1	0	1
110	Z	1	1	0	0
010	ZO	1	1	0	0
011	O	0	0	0	1
001	\overline{NO}	0	0	0	1

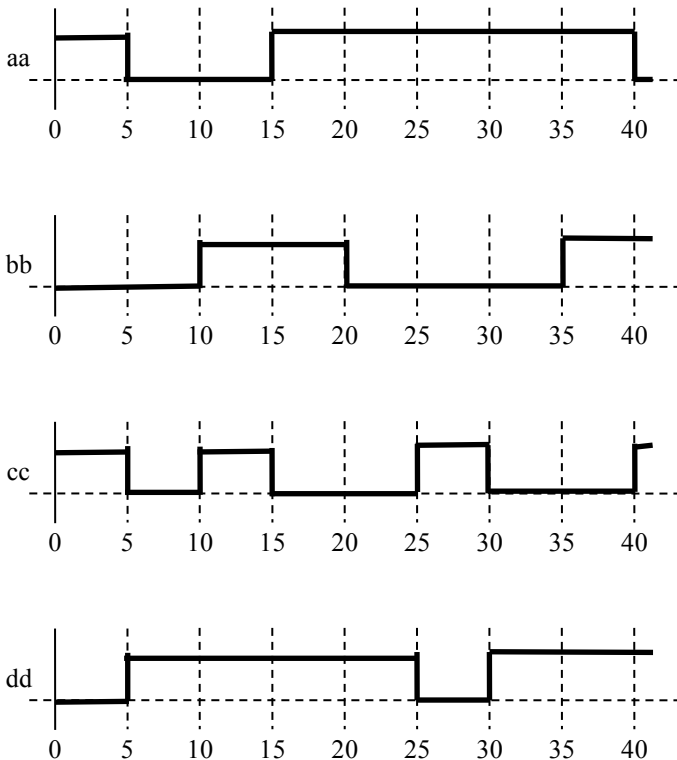
Vraag 13: b

De functie die wordt geïmplementeerd met de schakeling is: $f(x_3, x_2, x_1, x_0) = z = x_3(x_1 + x_2x_0)$ en dit komt overeen met: $\Sigma m(10,11,13,14,15)$.

Vraag 14: c

`var_s1 := '1' and '1' := '1' → res1 := '1' nor '0' := '0'`
`res2 <= '0' nor '0' (= '1')`

Vraag 15: c



De waarheidstabel van de functie F is:

j	a	b	c	d
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

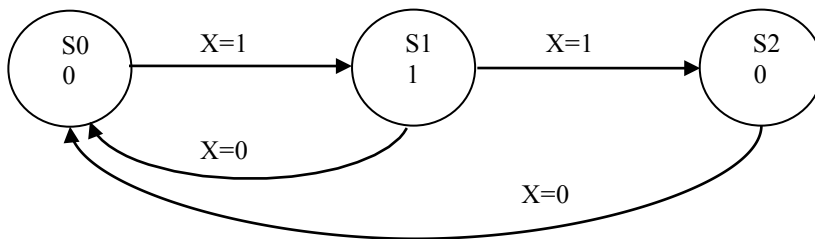
Vraag 16: b

Vraag 17: a

Vraag 18: c

Open vragen

Vraag 19



Vraag 20

K-maps :

A :

SC		AB			
		00	01	11	10
S	00	1	1	0	0
	01	1	0	0	1
	11	1	0	0	1
	10	1	1	1	1

T_A :

SC		AB			
		00	01	11	10
S	00	1	1	1	1
	01	1	0	1	0
	11	1	0	1	0
	10	1	1	0	0

$$T_A = A'B' + A'C' + C'S' + ABC$$

Vraag 21

```
ld R1 : laad R1 in A
xor 10000000b : bij R1 < 0 wordt A[7] nu 0.
and 10000000b : test op A[7]: A[7] = 0 => A = 0 => z = 1
bz R1neg : z = 1 => spring naar R1neg
```

Vraag 22:

```
entity Dff is
  port (ingang, reset, set, clk: in bit;
        uitgang: out bit);
end Dff;

architecture behaviour of Dff is
begin
  lbl1:
  process(reset, clk)
  begin
    if (reset = '1') then
      uitgang <= '0';
    elsif (clk'event and clk = '0') then
      if (set = '1') then
        uitgang <= '1';
      else
        uitgang <= ingang;
      end if;
    end if;
  end process clk_process;
end architecture behaviour;
```

Vraag 23:

```
entity bit3_rotator is
  port (links, niets, rechts, load, reset, clk: in bit;
        -- van de signalen links, niets, rechts, load wordt
        -- aangenomen dat nooit twee tegelijk '1' kunnen zijn.
        ingang : in bit_vector (2 downto 0);
        uitgang: out bit_vector (2 downto 0));
end bit3_rotator;

architecture struct of bit3_rotator is
component Dff
  port (ingang, reset, set, clk: in bit;
        uitgang: out bit);
end component;
signal din0, din1, din2 : bit;
signal duit0, duit1, duit2 : bit;
begin
  inst0: Dff port map (din0, reset, '0', clk, duit0);
  inst1: Dff port map (din1, reset, '0', clk, duit1);
  inst2: Dff port map (din2, reset, '0', clk, duit2);
  din0 <= (links and duit2) or (rechts and duit1)
         or (niets and duit0) or (load and ingang(0));
  din1 <= (links and duit0) or (rechts and duit2)
         or (niets and duit1) or (load and ingang(1));
  din2 <= (links and duit1) or (rechts and duit0)
         or (niets and duit2) or (load and ingang(2));
  uitgang(0) <= duit0;
  uitgang(1) <= duit1;
  uitgang(2) <= duit2;
end struct;
```