

Toetsvoorblad

Naam Student: _____

Studentnummer: _____

Locatie: **Delft**

Opleiding: Elektrotechniek	Toetsnaam: DIGSE1 (proeftoets)
Opsteller: J.E.J. op den Brouw Tweede lezer: J.Z.M. Broeders	Datum: 1 januari 1970 Tijd: 0:00 – 1:30 uur
Groep: EP2, EQ2D Module:	Aantal bladzijden: 4 (inclusief voorblad) Aantal vragen: 5 open vragen

Bij deze toets worden verstrekt:

- Gelineeerd papier
- Ruitjes papier
- Kladpapier
- Omslag voor gemaakt tentamen
- Overig: _____
- Bijlage(n): _____
- Opgavenbladen met ruimte om de vragen te beantwoorden
- Antwoordformulier ABCDE
- Antwoordformulier Ja/Nee
- Antwoordformulier Ja/Nee/Vraagteken

Toegestane eigen hulpmiddelen bij het maken van deze toets:

- Eenvoudige rekenmachine
- Grafische rekenmachine
- Computer
- Formuleblad(en): _____
- Tekenbenodigdheden (liniaal, passer)
- Eigen aantekeningen: zie Opmerkingen
- Boeken/dictaten: zie Opmerkingen

Opmerkingen:

Aantekeningen, boeken, afdrucken van de PowerPoint-slides, programmeerbare rekenmachines.

Cesuur (voorlopig):

Eindcijfer = 1 + (aantal behaalde punten / 10)

In te leveren door student bij surveillant:

- Alle documenten voorzien van naam en studentnummer, per document gesorteerd
- Alle documenten voorzien van naam en studentnummer, per student gesorteerd (in omslag)

Belangrijk:

Voor dit tentamen gelden de regels uit de toetsregeling van het Onderwijs- en Examenreglement. Dit document is aanwezig in het toetslokaal;

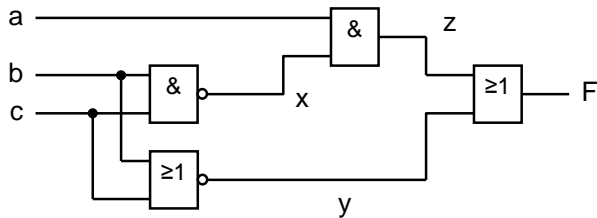
Je dient zelf te controleren of je alle pagina's en vragen van dit tentamen hebt ontvangen;

Dit tentamen is enkelzijdig geprint;

Schrijf je naam en studentnummer op alle documenten.

Opgave 1 (15 pt)

Gegeven onderstaand schema bestaande uit een aantal poorten, ingangen, uitgang en interne signalen. Deze schakeling wordt in een simulator gesimuleerd. De poorten hebben vertraging zonder inertial delay: elke puls, hoe smal ook, wordt doorgegeven. Naast het schema is de bijbehorende VHDL-code gegeven.



```
X <= transport B nand C after 2 ns;  
Y <= transport B nor C after 3 ns;  
Z <= transport A and X after 4 ns;  
F <= transport Z or Y after 5 ns;
```

Figuur 1: Schakeling opgave 1.

Code 1: VHDL-code bij schakeling opgave 1

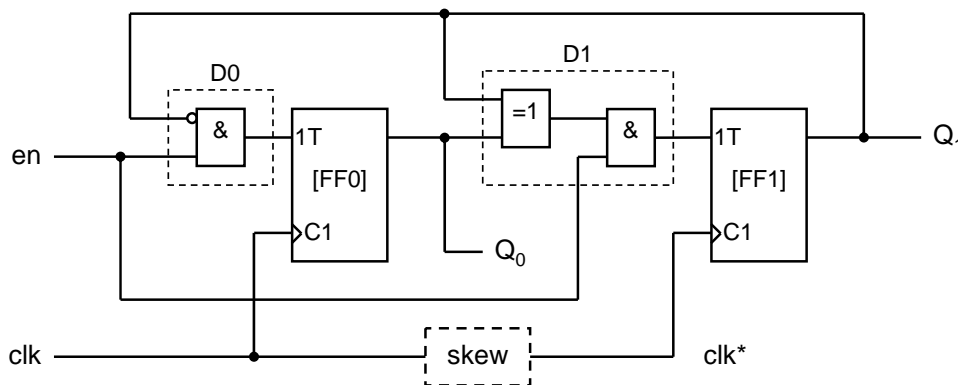
Alle signalen zijn van het type bit en zijn op '0' geïnitieerd. Op tijdstip 0 worden de volgende toekenningen gedaan:

```
a <= '1' after 1 ns;  
c <= '1' after 1 ns;
```

Teken in een tijddiagram het verloop van alle signalen in figuur 1.

Opgave 2 (30 pt)

Gegeven de schakeling in figuur 2. Hierin zijn twee T-flipflops opgenomen (FF0 en FF1), twee stukken combinatoriek (D0, D1) en één klokvertraging (skew). D0 is combinatoriek voor de ingang van FF0 en D1 is combinatoriek voor de ingang van FF1. FF1 krijgt een vertraagd kloksignaal aangeboden (skew).



Figuur 2: Synchrone flipflop-schakeling

Van de bouwstenen zijn de volgende timingparameters gegeven:

T-flipflops: $t_{su}/t_h/t_{P(min)}/t_{P(max)} = 2/1/6/8$ ns

AND: $t_{P(min)}/t_{P(max)} = 2/3$ ns

EXOR: $t_{P(min)}/t_{P(max)} = 3/5$ ns

Skew: $t_P = 3$ ns

Van de skew is maar één vertragingstijd gegeven, er wordt geen onderscheid gemaakt tussen de minimale en maximale vertragingstijd. Het betreft hier namelijk clock skew.

Merk op dat er in dit systeem **drie** paden zijn tussen de flipflops waarlangs data wordt getransporteerd.

- a) Geef aan welke paden er tussen de flipflops zijn waarlangs data wordt getransporteerd. Geef hierbij duidelijk aan welke stukken combinatorische logica worden gepasseerd. (5 pt)

Van het systeem moet de maximale frequentie worden berekend. Na onderzoek blijkt het pad FF1 → D1 → FF1 de grootste minimale periodetijd op te leveren.

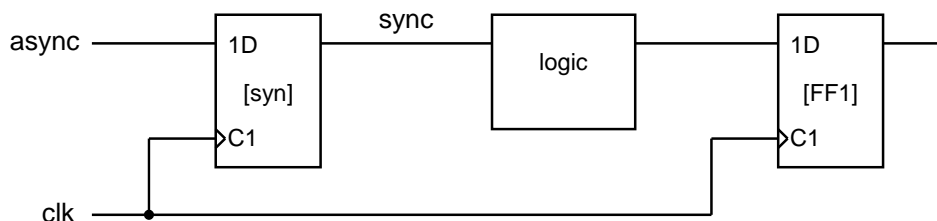
- b) Teken een tijddiagram met alleen relevante timingparameters waarmee de maximale frequentie berekend moet worden. (10 pt)
- c) Bereken de maximale frequentie waarop dit systeem nog betrouwbaar werkt. (5 pt)

Het signaal en (enable) wordt vertraagd aangeboden aan de ingang van FF1. Dat heeft invloed op de setup- en holdtijden van signaal en t.o.v. het kloksignaal clk.

- d) Teken een tijddiagram met alleen relevante timingparameters waarmee de setuptijd van signaal en t.o.v. van het kloksignaal clk berekend moet worden. (5 pt)
- e) Bereken de setuptijd van signaal en t.o.v. van het kloksignaal clk. (5 pt)

Opgave 3 (15 pt)

Gegeven de schakeling in figuur 3.



Figuur 3: Synchronische schakeling met synchronizer

Van flipflop FF1 is gegeven: $t_{su}/t_h/t_{P(min)}/t_{P(max)} = 5/2/4/10$ ns

Van de synchronizer is gegeven: $\tau = 0,15$ ns, $T_0 = 9,8 \cdot 10^6$ s

Van logica is gegeven: $t_{P(min)}/t_{P(max)} = 7/10$ ns

De systeemfrequentie is 40 MHz. Een ontwerper wil dat bij dit systeem de gemiddelde tijd tussen twee fouten 1000 jaar (1 jaar = 365 dagen) bedraagt.

- a) Bereken de maximale tijd die de synchronizer heeft om uit de metastabiele toestand te komen. Stel hiervoor een formule op. (10 pt)
- b) Bereken de maximaal toegestane frequentie van het asynchroon binnenkomende signaal om aan de eis van de ontwerper te kunnen voldoen. (5 pt)

Opgave 4 (15 pt)

De schakeling in figuur 2 (pag. 2) is een teller. De werking van deze teller moet onderzocht worden. De ingang en (enable) wordt op logisch '1' gezet. De beide flipflops hebben de (begin-)stand 0 ($Q_1Q_0 = 00$). Hierna worden klokpulsen aan de klokingang aangeboden. De teller zal hierdoor van stand veranderen en doorloopt een bepaalde telcyclus.

- a) Bepaal de doorlopen telcyclus van deze teller. (5 pt)

Een student wil deze teller gebruiken voor een practicumopdracht. Daarvoor is een entity-beschrijving nodig.

- b) Geef de entity-beschrijving van de teller in figuur 2. (10 pt)

Opgave 5 (15 pt)

Ontwerp een BCD-teller in VHDL. De teller moet aan de volgende eisen voldoen:

- De teller heeft een asynchrone reset, actief laag.
- De teller heeft een enable, actief hoog, waarmee het tellen kan worden geactiveerd.
- De teller telt cyclisch omhoog (van 0 t/m 9) als het signaal updown hoog is.
- De teller telt cyclisch omlaag (van 9 t/m 0) als het signaal updown laag is.
- De telstand is als BCD-getal beschikbaar.

In de onderstaande code is de entity met port-beschrijving al ingevuld.

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

entity bcdupdowncounter is
  port (clk      : in std_logic;           -- de klok
        areset   : in std_logic;         -- asynchrone reset
        enable   : in std_logic;         -- aan of uit
        updown   : in std_logic;         -- omhoog of omlaag tellen
        count    : out unsigned(3 downto 0) -- de telstand
  );
end bcdupdowncounter;

architecture rtl of bcdupdowncounter is

  -- Deze code moet jij ontwikkelen

end architecture rtl;
```

Code 2: VHDL-code bij opgave 5

- a) Geef de VHDL-code voor de werking van de teller. (10 pt)

De teller moet uitgebreid worden met een **terminal count** die aangeeft dat de teller in een uiterste stand staat. Daarvoor wordt een extra uitgang aangemaakt (tc). Merk op dat de teller **twee** uiterste standen heeft, afhankelijk van de telrichting. Bij omhoog tellen is de uiterste telstand 9 en bij omlaag tellen is de uiterste telstand 0.

- b) Geef alleen de VHDL-code voor de werking van de uitgang tc. De entity hoeft niet opnieuw gegeven te worden. (5 pt)