

Toetsvoorblad

Naam Student: _____

Studentnummer: _____

Locatie: **Delft**

Opleiding: Elektrotechniek	Toetsnaam: DIGSE2 (proeftoets)
Opsteller: J.E.J. op den Brouw Tweede lezer: W. Muhammad	Datum: 1 januari 1970 Tijd: 0:00 – 1:30 uur
Groep: EP2, EQ2D Module:	Aantal bladzijden: 4 (inclusief voorblad) Aantal vragen: 5

Bij deze toets worden verstrekt:

- | | |
|---|--|
| <input checked="" type="checkbox"/> Gelinieerd papier | <input type="checkbox"/> Opgavenbladen met ruimte om de vragen te beantwoorden |
| <input checked="" type="checkbox"/> Ruitjes papier | <input type="checkbox"/> Antwoordformulier ABCDE |
| <input type="checkbox"/> Kladpapier | <input type="checkbox"/> Antwoordformulier Ja/Nee |
| <input type="checkbox"/> Omslag voor gemaakt tentamen | <input type="checkbox"/> Antwoordformulier Ja/Nee/Vraagteken |
| <input type="checkbox"/> Overig: _____ | |
| <input type="checkbox"/> Bijlage(n): _____ | |

Toegestane eigen hulpmiddelen bij het maken van deze toets:

- | | |
|---|--|
| <input checked="" type="checkbox"/> Eenvoudige rekenmachine | <input type="checkbox"/> Tekenbenodigdheden (liniaal, passer) |
| <input checked="" type="checkbox"/> Grafische rekenmachine | <input checked="" type="checkbox"/> Eigen aantekeningen: zie Opmerkingen |
| <input type="checkbox"/> Computer | <input checked="" type="checkbox"/> Boeken/dictaten: zie Opmerkingen |
| <input type="checkbox"/> Formuleblad(en): _____ | |

Opmerkingen:

Toegestane hulpmiddelen: Aantekeningen, boeken, dictaat, afdrukken van de PowerPoint-slides, programmeerbare rekenmachine

Cesuur (voorlopig):

Maximaal 90 punten te behalen. 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.

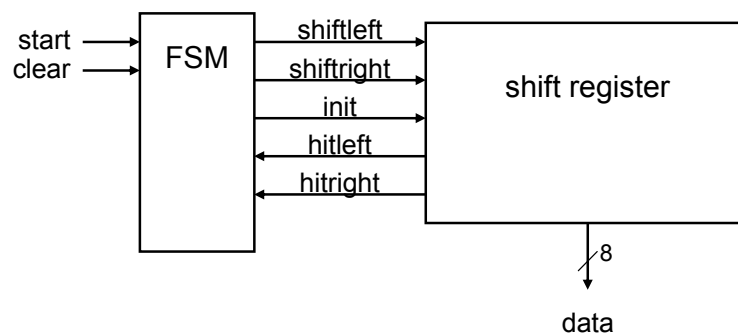
Opgave 1 (20 pt)

Een synchrone machine heeft één ingang X en één uitgang Z . De machine moet doorlopend, zonder overlap, het patroon 0011 herkennen in een op X aangeboden willekeurige bitreeks, beginnend met de nullen. Is het patroon herkend, dan moet uitgang Z één klokperiode logisch 1 zijn, anders is Z logisch 0.

- Ontwerp een toestandsdiagram voor deze machine. Dit moet een Mealy-machine zijn. Er zijn niet meer dan vijf toestanden nodig, minder kan wel. Zorg voor een minimaal aantal toestanden. (10 pt)
- Stel voor het toestandsdiagram uit [a](#) de waarheidstabel op. Maak gebruik van D flipflops. Gebruik de binaire telcode als codering. N.B: alle ongebruikte toestanden moeten als don't care gespecificeerd worden. (10 pt)

Opgave 2 (30 pt)

Gegeven het onderstaande blokschema (zie figuur 1) van een systeem bestaande uit een besturing (FSM) en een 8-bits schuifregister. Beide zijn synchroon geklokt en hebben een asynchrone reset (niet getekend).



Figuur 1: Blokschema machine.

Het schuifregister werkt als volgt:

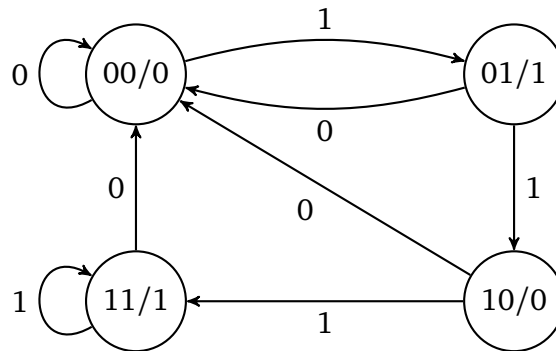
- Het kan geladen worden met 00000001 door signaal `init` logisch 1 te maken.
- Het kan naar links schuiven door signaal `shiftright` logisch 1 te maken.
- Het kan naar rechts schuiven door signaal `shiftright` logisch 1 te maken.
- Als de 1 in het schuifregister helemaal links staat is `hitright` logisch 1.
- Als de 1 in het schuifregister helemaal rechts staat is `hitright` logisch 1.

De machine moet met een opdracht (een druk op start) gestart worden. Als doel heeft de machine om de enige 1 in het schuifregister heen en weer te schuiven, van de meest rechtse positie naar de meest linkse positie en dan weer terug. Dit moet de machine blijven doen totdat signaal `clear` wordt geactiveerd. Let erop dat voordat er geschoven kan worden, het schuifregister geïntialieerd moet worden (zie werking schuifregister).

- Ontwerp een toestandsdiagram voor een Mealy-machine voor de besturing van het bovenstaande gespecificeerde systeem. Er zijn niet meer dan vijf toestanden nodig, minder kan wel. Zorg voor een minimaal aantal toestanden. (20 pt)
- Leidt voor het in [a](#) gevonden toestandsdiagram de *next state equations* af voor een one-hot codering. (10 pt)

Opgave 3 (15 pt)

Gegeven het toestandsdiagram van een Moore-machine (zie figuur 2).



Figuur 2: Toestandsdiagram machine.

- Bepaal het aantal data-ingangen en uitgangen en het minimum aantal flipflops van de machine. (5 pt)
- De machine wordt gereset en staat in toestand 00. De machine krijgt de volgende data op de ingang te verwerken: 0011101101, beginnend met de voorste 0. Bepaal de doorlopen toestanden en de bijbehorende uitgangswaarde(n). (10 pt)

Opgave 4 (15 pt)

Gegeven de VHDL-beschrijving van een toestandsmachine in listing 1.

- Dit is een Mealy-, Moore- of Medvedev-machine? Motiveer het antwoord. (5 pt)
- Teken het toestandsdiagram van de machine in listing 1. Het gebruik van de VHDL-constructie **others** bij de transities is niet toegestaan. (10 pt)

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

entity opgave4 is
    port (clk      : in std_logic;
          areset   : in std_logic;
          x        : in std_logic;
          y        : in std_logic;
          z        : out std_logic);
end entity opgave4;

architecture fsm_2proc of opgave4 is
    type state_type is (s0, s1, s2);
    signal state : state_type;

begin

    combreg: process (clk, areset) is
    begin
        if areset = '1' then
            state <= s0;
        elsif rising_edge(clk) then
```

```

case state is
  when s0 => if x = '1' then
              state <= s1;
            elsif y = '1' then
              state <= s2;
            else
              state <= s0;
            end if;
  when s1 => if x = '0' and y = '0' then
              state <= s0;
            elsif x = '0' and y = '1' then
              state <= s2;
            else
              state <= s1;
            end if;
  when s2 => if x = '1' or y = '1' then
              state <= s2;
            else
              state <= s0;
            end if;
  when others => null;
end case;
end if;
end process;

ol: process (state) is
begin
  z <= '0';
  if state = s0 or state = s1 then
    z <= '1';
  end if;
end process;

end architecture fsm_2proc;

```

Listing 1: VHDL-beschrijving van een toestandsmachine

Opgave 5 (10 pt)

Gegeven de in de colleges besproken microprocessor. Op deze processor wordt de onderstaande code uitgevoerd.

```

ld    R0,#0      ; Load R0 with constant 0
add   R0,R0,#127 ; Add 127 to R0
add   R0,R0,#255 ; Add 255 to R0

```

Listing 2: Instructies voor de microprocessor

- Wat is de inhoud van register *R0* na het uitvoeren van deze instructies? (5 pt)
- Wat zijn de waarden van de *z*-flag en de *c*-flag na het uitvoeren van deze instructies? (5 pt)