

# Toets Digitale Systemen 31/05/2007, 08.30 – 10.30 uur

De toets is **open boek** en bestaat uit 10 multiple-choice (MC) vragen en 3 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
- *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!)

U mag het boek "Contemporary Logic Design", het VHDL boek, en eventuele prints van het **college** slides 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.

De toets is niet verplicht maar kan u in sommige gevallen van een 5 naar een 6 helpen bij het eindtentamen. Succes!!

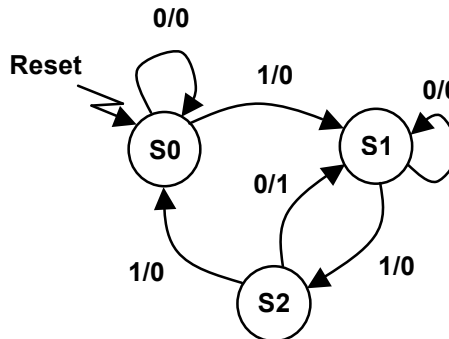
## A. MC-vragen (gewicht: $10 \times 6\% = 60\%$ )

### Vraag 1

Gegeven bijgaande FSD (Mealy machine):

Welke 3-bits reeks wordt door het circuit gedetecteerd ( $\Rightarrow$  uitgang 1)?

- 011
- 101
- 010
- geen van bovenstaande antwoorden.

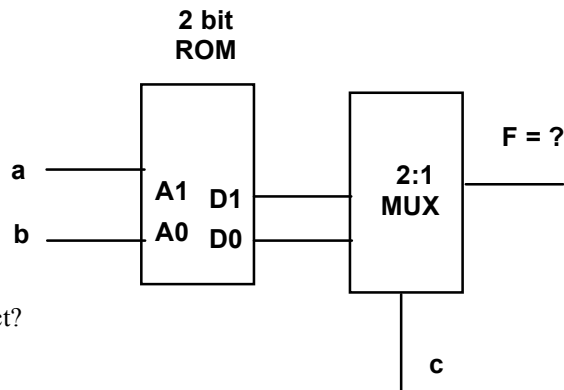


### Vraag 2

Beschouw nevenstaande circuit.

De ROM is als volgt geprogrammeerd:

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



Welke van de volgende expressies voor F is correct?

- $F = (b \oplus c)' + ac$
- $F = c'b' + cb$
- $F = b \oplus c + ac'$
- geen van bovenstaande antwoorden.

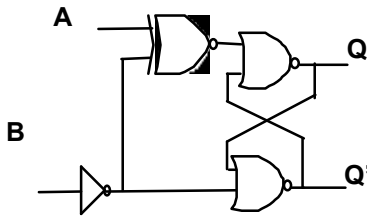
**Vraag 3**

Gegeven  $C = A - B$ , waarbij  $A = 011011$  en  $C = 110000$ , en  $A, B, C$  in 2's-complement notatie. Welke van de volgende uitspraken is correct?

- a.  $B = 11$
- b.  $B = 43$
- c.  $B = -21$
- d. er is overflow en  $B$  kan dus niet bepaald worden.

**Vraag 4**

Gegeven bijgaand circuit:

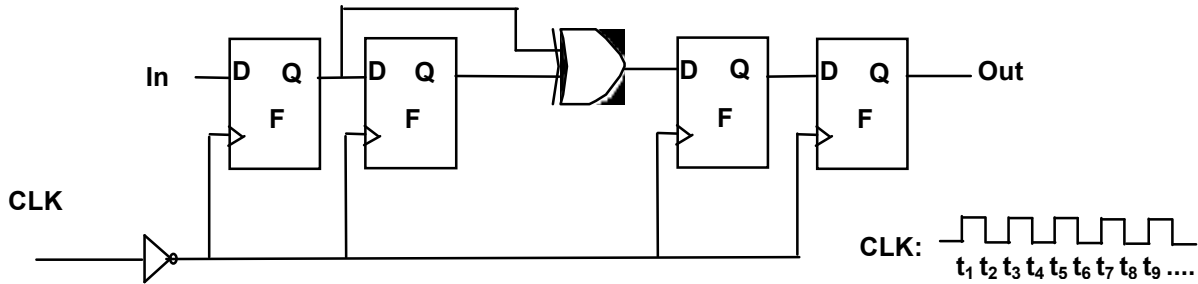


Dit circuit is

- a. een latch met geen verboden ingangscombinatie
- b. een latch;  $AB = 10$  is de verboden ingangscombinatie
- c. geen bruikbaar geheugenelement: de Hold-combinatie ontbreekt
- d. geen van bovenstaande antwoorden.

**Vraag 5**

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



Indien voor tijdstip  $t_1$  alle  $Q$ s zijn 0 en op 'In' het constante signaal 1 wordt gezet, wanneer wordt 1 op 'Out' waargenomen? Op of direct na:

- a.  $t_4$
- b.  $t_5$
- c.  $t_6$
- d.  $t_7$  of later.

**Vraag 6** Gegeven de volgende Delta I assembly code:

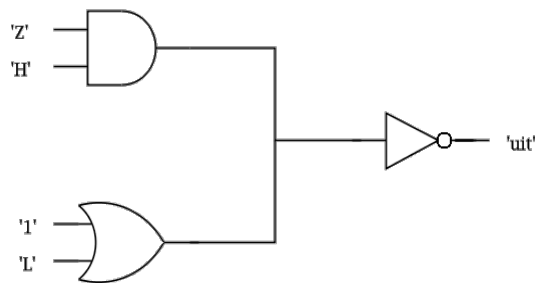
```
set c
ld 11110000b
and 11100001b
add 11011011b
st R1
xor 11111111b
add R1
```

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

- a. A = 0<sub>H</sub> C = 1 Z = 0
- b. A = FF<sub>H</sub> C = 0 Z = 0
- c. A = 0<sub>H</sub> C = 1 Z = 1
- d. A = FF<sub>H</sub> C = 1 Z = 1

**Vraag 7**

Gegeven is de volgende schakeling met de aangegeven 'std\_logic' input-signalen. De overige signalen zijn ook van het type 'std\_logic'.



Wat is de waarde van het uitgangssignaal? ( Maak gebruik van de bijlage "std\_logic\_1164", blz. 9).

- a. 'U'
- b. 'X'
- c. 'W'
- d. '1'

Hieronder is een entity gegeven met daarbij zes verschillende architectures:

```
ENTITY functie IS
    PORT(in1,in2,in3: in bit; out1: out bit);
END functie;

ARCHITECTURE werkingA OF functie IS
BEGIN
    PROCESS(in1,in2,in3)
    BEGIN
        IF(in2 = '0') THEN out1 <= '0';
        ELSIF(in3'event AND in3 = '0') THEN out1 <= 0;
        END IF;
    END PROCESS;
END werkingA
```

```

ARCHITECTURE werkingB OF functie IS
BEGIN

    PROCESS(in1,in2)
    BEGIN
        IF (in2='1' AND in2'event) THEN
            out1 <= in1;
        END IF;
    END PROCESS;
END werkingB

ARCHITECTURE werkingC OF functie IS
BEGIN
    PROCESS(in1,in2)
    BEGIN
        out1 = '0';
        IF(in1 = '1') THEN out1 = in2;
        ELSIF(in2 = '0') out1 = '1';
        END IF;
    END PROCESS;
END werkingC

ARCHITECTURE werkingD OF functie IS
BEGIN
    PROCESS(in1,in2)
    BEGIN
        IF(in1 = '0') THEN out1 <= '1';
        ELSE out1 <= NOT in2;
        END IF;
    END PROCESS;
END werkingD

ARCHITECTURE werkingE OF functie IS
BEGIN
    PROCESS(in1,in2,in3)
    BEGIN
        IF(in1 = '1') THEN out1 = in2;
        END IF;
        out1 = in3;
    END PROCESS;
END werkingE

ARCHITECTURE werkingF OF functie IS
BEGIN
    PROCESS(in2,in3)
    BEGIN
        IF(in2 = '0') THEN out1 <= '0';
        ELSIF(in3 = '1') THEN out1 <= '1';
        END IF;
    END PROCESS;
END werkingF

```

**Vraag 8** Welke uitspraak is waar?

- a. architecture 'werkingF' beschrijft de combinatorische functie in2 and in3
- b. architecture 'werkingB' beschrijft een simpele latch
- c. architecture 'werkingC' beschrijft de combinatorische functie in1 xnor in2
- d. geen van de drie genoemde mogelijkheden is goed

**Vraag 9** Welke architectuur beschrijft een 2-input multiplexer?

- a. werkingA
- b. werkingD
- c. werkingE
- d. geen van de bovengenoemde architecturen

Hieronder wordt een beschrijving gegeven van een 4-bits comparator. De binaire signalen a en b worden vergeleken. Het resultaat hiervan wordt de boolean c. We geven 3 entities en een architecture:

```
--mogelijkheid 1:
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
USE IEEE.std_logic_arith.all;
ENTITY comparator IS
    PORT (a: in unsigned(3 DOWNTO 0);
          b: in unsigned(3 DOWNTO 0);
          c: out boolean);
END comparator;

--mogelijkheid 2:
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
USE IEEE.std_logic_arith.all;
ENTITY comparator IS
    PORT (a: in signed(3 DOWNTO 0);
          b: in signed(3 DOWNTO 0);
          c: out std_logic);
END comparator;

--mogelijkheid 3:
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
USE IEEE.std_logic_arith.all;
ENTITY comparator IS
    PORT (a: in unsigned(3 DOWNTO 0);
          b: in std_logic_vector(3 DOWNTO 0);
          c: out boolean);
END comparator;

ARCHITECTURE werking OF comparator IS
BEGIN
    c <= (a < b);
END werking;
```

**Vraag 10** Welke van de drie entities wordt door de VHDL compiler niet geaccepteerd?

- a. mogelijkheden 1 en 2 worden niet geaccepteerd.
- b. mogelijkheden 2 en 3 worden niet geaccepteerd.
- c. mogelijkheden 1 en 3 worden niet geaccepteerd.
- d. alledrie worden niet geaccepteerd.

## B. Open vragen (gewicht: 40%)

U wordt gevraagd om een regelaar te ontwerpen die een automatische schuifdeur bestuurt. De schuifdeur is uitgevoerd met aan weerszijden een sensor om personen te detecteren. Deze sensoren genereren respectievelijk signalen A en B, waarbij geldt:

A	B	betekenis
1	1	niemand aan weerszijden
1	0	iemand aan de B-kant
0	1	iemand aan de A-kant
0	0	iemand aan weerszijden

De deur wordt bestuurd mbv. het signaal D, waarbij geldt:

D	betekenis
1	deur gaat dicht en/of blijft dicht
0	deur gaat open en/of blijft open

Uw regelaar moet aan onderstaande specificatie voldoen:

- als iemand bij A of B dan deur open
- als niemand bij A of B dan deur dicht, maar als de deur open is, pas *nadat* een zekere vertragingstijd  $\Delta$  is gepasseerd (na een timeout).

Naast bovengenoemde signalen A, B, en D heeft u de beschikking over een clocksignaal CLK. Om de gewenste vertragingstijd te implementeren kunt u gebruik maken van een reeds ontworpen timer met ingangsignaal S (start) en uitgangsignaal T (timeout). Het gedrag van de timer kunt u bepalen aan de hand van de bijbehorende VHDL specificatie:

```
ENTITY timer IS
PORT (
    S: IN bit; -- reset
    T: OUT bit; -- timeout
    CLK: IN bit -- clock
);
END timer;

ARCHITECTURE timer_arch OF timer IS
BEGIN
    PROCESS (CLK)
        VARIABLE count: INTEGER := 0;
    BEGIN
        IF (CLK'event AND CLK = '1') THEN
            IF (S = '1') THEN
                count := 0; -- reset
            ELSIF (count < 10000000) THEN
                count := count + 1; -- wait / timeout
            END IF;
            IF (count < 10000000) THEN -- map count to T
                T <= '0';
            ELSE
                T <= '1';
            END IF;
        END IF;
    END PROCESS;
END timer_arch;
```

### Vraag 11 (15%)

Specificeer de gewenste besturing van de regelaar mbv. een FSD of ASM (u mag zelf kiezen) met gebruikmaking van de eerder beschreven signalen A, B, D, S, T. Deze FSD/ASM vormt tevens de basis voor de navolgende vraag. (Ook als uw FSD/ASM fout is, wordt uw FSD/ASM als uitgangspunt gehanteerd bij de beoordeling.) Kies voor de FSM een Moore machine.

### Vraag 12 (10%)

Ontwerp nu een hardware-oplossing voor de besturing aan de hand van uw FSD/ASM.

- Schrijf de waarheidstabel op die aangeeft hoe de nieuwe state (S0/1/2) en uitgangsignalen (D, S) afhangen van de huidige state en ingangsignalen (A, B, T).
- Kies de volgende state assignment die de states  $S_i$  afbeeldt op de state-variabelen  $Q_j$ :

State	Q1	Q0
S0	0	0
S1	0	1
S2	1	1

Teken de K-maps en leidt de vereenvoudigde expressies af voor de  $Q_j$ 's, D en S.

### Vraag 13 (15%)

Nu gaan we gebruik maken van de synthesesmogelijkheden van VHDL. Ontwerp een VHDL specificatie van de besturing. U mag zelf weten welk niveau van abstractie u kiest voor uw beschrijving. De specificatie mag bijv. lijken op de hardware-oplossing. De interface van de besturing is reeds gegeven. Wat vult u in op de gestippelde plek?

```
ENTITY besturing IS
PORT (
  A:   IN bit;      -- sensor A
  B:   IN bit;      -- sensor B
  D:   OUT bit;     -- door control
  S:   OUT bit;     -- timer start
  T:   IN bit;      -- timer time-out
  CLK: IN bit       -- clock
);
END besturing;

ARCHITECTURE your_architecture OF besturing IS
  ....
  ....
END your_architecture;
```

Uittreksel van "std\_logic\_1164"

```

-----
-- tables for logical operations
-----
-- truth table for "and" function
CONSTANT and_table : stdlogic_table := (
-----
--   | U   X   0   1   Z   W   L   H   -   |   |
-----
    ( 'U', 'U', '0', 'U', 'U', 'U', '0', 'U', 'U' ), -- | U |
    ( 'U', 'X', '0', 'X', 'X', 'X', '0', 'X', 'X' ), -- | X |
    ( '0', '0', '0', '0', '0', '0', '0', '0', '0' ), -- | 0 |
    ( 'U', 'X', '0', '1', 'X', 'X', '0', '1', 'X' ), -- | 1 |
    ( 'U', 'X', '0', 'X', 'X', 'X', '0', 'X', 'X' ), -- | Z |
    ( 'U', 'X', '0', 'X', 'X', 'X', '0', 'X', 'X' ), -- | W |
    ( '0', '0', '0', '0', '0', '0', '0', '0', '0' ), -- | L |
    ( 'U', 'X', '0', '1', 'X', 'X', '0', '1', 'X' ), -- | H |
    ( 'U', 'X', '0', 'X', 'X', 'X', '0', 'X', 'X' ) -- | - |
);
-- truth table for "or" function
CONSTANT or_table : stdlogic_table := (
-----
--   | U   X   0   1   Z   W   L   H   -   |   |
-----
    ( 'U', 'U', 'U', '1', 'U', 'U', 'U', '1', 'U' ), -- | U |
    ( 'U', 'X', 'X', '1', 'X', 'X', 'X', '1', 'X' ), -- | X |
    ( 'U', 'X', '0', '1', 'X', 'X', '0', '1', 'X' ), -- | 0 |
    ( '1', '1', '1', '1', '1', '1', '1', '1', '1' ), -- | 1 |
    ( 'U', 'X', 'X', '1', 'X', 'X', 'X', '1', 'X' ), -- | Z |
    ( 'U', 'X', 'X', '1', 'X', 'X', 'X', '1', 'X' ), -- | W |
    ( 'U', 'X', '0', '1', 'X', 'X', '0', '1', 'X' ), -- | L |
    ( '1', '1', '1', '1', '1', '1', '1', '1', '1' ), -- | H |
    ( 'U', 'X', 'X', '1', 'X', 'X', 'X', '1', 'X' ) -- | - |
);
-- truth table for "not" function
CONSTANT not_table : stdlogic_1d :=
-----
--   | U   X   0   1   Z   W   L   H   -   |
-----
    ( 'U', 'X', '1', '0', 'X', 'X', '1', '0', 'X' );
-----

-- overloaded logical operators ( with optimizing hints )
-----

FUNCTION "and" ( l : std_ulogic; r : std_ulogic ) RETURN UX01 IS
BEGIN
    RETURN (and_table(l, r));
END "and";

FUNCTION "nand" ( l : std_ulogic; r : std_ulogic ) RETURN UX01 IS
BEGIN
    RETURN (not_table ( and_table(l, r)));
END "nand";

FUNCTION "or" ( l : std_ulogic; r : std_ulogic ) RETURN UX01 IS
BEGIN
    RETURN (or_table(l, r));
END "or";

FUNCTION "nor" ( l : std_ulogic; r : std_ulogic ) RETURN UX01 IS
BEGIN
    RETURN (not_table ( or_table( l, r )));
END "nor";

FUNCTION "not" ( l : std_ulogic ) RETURN UX01 IS
BEGIN
    RETURN (not_table(l));
END "not";

```



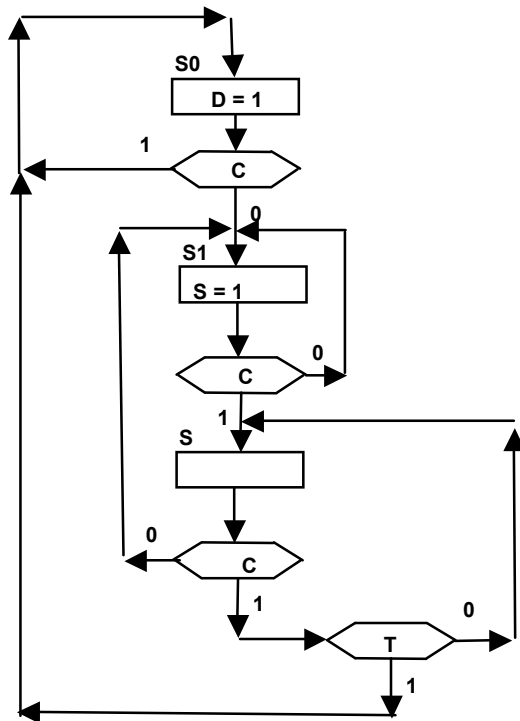
# Uitwerkingen Toets Digitale Systemen 31/05/2007, 08.30 – 10.30 uur

MC-vragen:

- |      |       |
|------|-------|
| 1. c | 6. a  |
| 2. a | 7. b  |
| 3. d | 8. c  |
| 4. b | 9. d  |
| 5. c | 10. b |

## Vraag 11

In de vorm van een ASM ziet de FSM-specificatie er als volgt uit (voor het gemak hebben we een signaal C geïntroduceerd volgens  $C = A \text{ and } B$ , dus  $C = 0$  als iemand aan de deur):



## Vraag 12

a.

Hdg. state	Vlgd. state				Uitgang
	CT				
	00	01	10	11	DS
S0	S1	S1	S0	S0	1x
S1	S1	S1	S2	S2	01
S2	S1	S1	S2	S0	00

b.

		$Q_1 Q_0$			
		00	01	11	10
C T	00	1	1	1	X
	01	1	1	1	X
	11	0	1	0	X
	10	0	1	1	X

$$Q_0^{t+1} = C' + Q_1' Q_0 + T' Q_0$$

		$Q_1$	
		0	1
$Q_0$	0	1	X
	1	0	0

$$D = Q_0'$$

		$Q_1 Q_0$			
		00	01	11	10
C T	00	0	0	0	X
	01	0	0	0	X
	11	0	1	0	X
	10	0	1	1	X

$$Q_1^{t+1} = C T' Q_0 + C Q_1' Q_0$$

		$Q_1$	
		0	1
$Q_0$	0	X	X
	1	1	0

$$S = Q_1'$$

### Vraag 13.

Een behavioral VHDL architecture kan er als volgt uitzien (let op de correspondentie met de ASM):

```
ARCHITECTURE your_architecture OF besturing IS
  TYPE door_state IS (DICHT, OPEN_1, OPEN_2);
  SIGNAL STATE, NEW_STATE: door_state;
  SIGNAL C: bit;
BEGIN
  C <= A nand B;

  plb1:
  PROCESS(CLK)
  BEGIN
    if (CLK'EVENT and CLK='1') THEN
      STATE <= NEW_STATE;
    END IF;
  END PROCESS;

  plb2:
  PROCESS(C, T, STATE)
  BEGIN
    CASE STATE IS
      WHEN DICHT =>
        D <= '1';
        S <= '0';
        IF (C = '1') THEN
          NEW_STATE <= OPEN_1;
        ELSE
          NEW_STATE <= DICHT;
        END IF;
      WHEN OPEN_1 =>
        D <= '0';
        S <= '1';
        IF (C = '0') THEN
          NEW_STATE <= OPEN_2;
        ELSE
          NEW_STATE <= OPEN_1;
        END IF;
      WHEN OPEN_2 =>
        D <= '1';
        S <= '0';
        IF (T = '1') THEN
          NEW_STATE <= DICHT;
        ELSIF (C = '1') THEN
          NEW_STATE <= OPEN_1;
        ELSE
          NEW_STATE <= OPEN_2;
        END IF;
    END CASE;
  END PROCESS;
END your_architecture;
```