

EE1410: Digitale Systemen

BSc. EE, 1e jaar, 2012-2013, vragencollege 2

Arjan van Genderen, Stephan Wong, Computer Engineering
7-6-2013

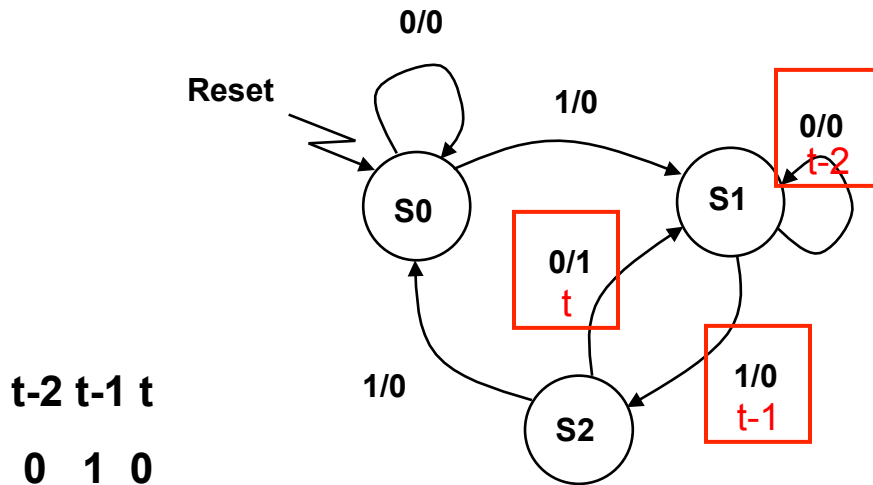
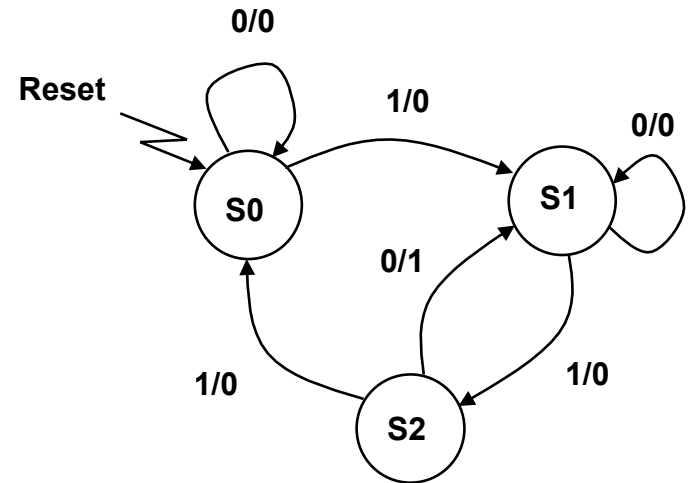
Vragencollege

- Tentamen dinsdag 25 juni 9.00 – 12.00
Hertentamen dinsdag 13 aug. 9.00 – 12.00
- Vandaag: Bespreking toets ET1405 31-5-2007
- Naast toetsen en tentamen van EE1410 van verleden jaar kunnen ook oude tentamens ET1405 als oefenmateriaal worden gebruikt. Verschillen zijn:
 - Niet meer tot de stof behoort: ASM schema's en het voorkomen van hazards m.b.v. karnaugh maps.
 - Bij EE1410 meer aandacht voor technologie aspecten (CMOS schakelingen)
- VHDL opdracht 2 uiterlijk vandaag klaar !

Vraag 1

Gegeven bijgaand FSD (Mealy machine):
Welke 3-bits reeks kan door het circuit worden gedetecteerd (=> uitgang 1), net perse vertrekkende vanuit de reset state?
Het eerst binnenkomende bit staat links.

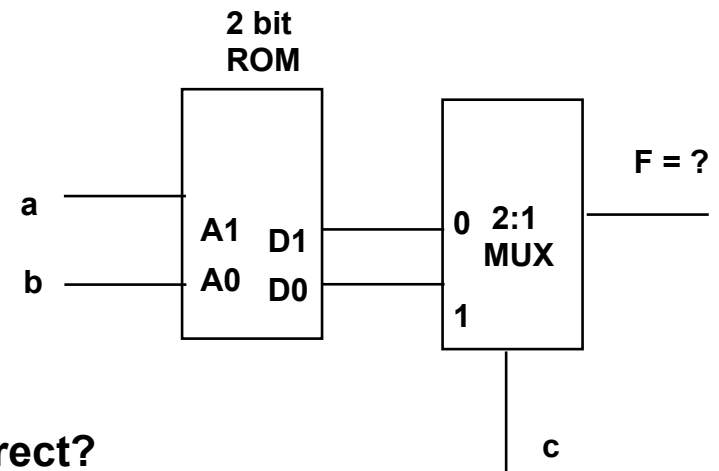
- a. 011
- b. 101
- ✓ c. 010
- d. geen van bovenstaande antwoorden.



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?

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

ROM: $D0 = A1 + A0 = a + b$ en $D1 = A0' = b'$

MUX: $F = c' D1 + c D0$

$$F = c'b' + c(a + b) = c'b' + ca + cb$$

$$\begin{aligned} \text{Antwoord van a: } (b \oplus c)' + ac &= (bc' + b'c)' + ac \\ &= bc + b'c' + ac \end{aligned}$$

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.

$$A = 011011 = 16 + 8 + 2 + 1 = 27$$

$$C = 110000 \Rightarrow -C = 001111 + 1 = 010000 = 16$$

$$\text{Dus uit } C = A - B \text{ volgt: } -16 = 27 - B$$

$$\Rightarrow B = 43$$

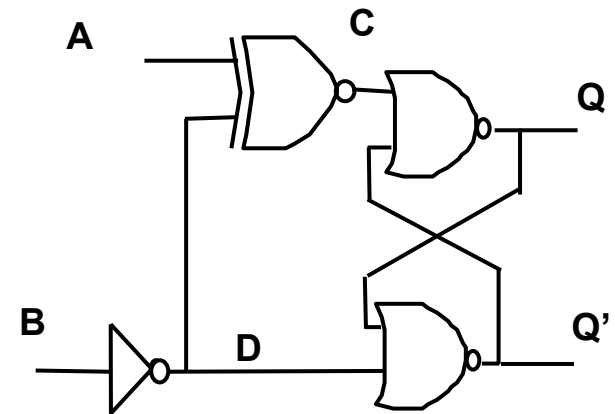
Dit past niet want het maximum getal is 011111

$$= 16 + 8 + 4 + 2 + 1 = 31$$

Vraag 4

Gegeven bijgaand circuit:
Dit circuit is

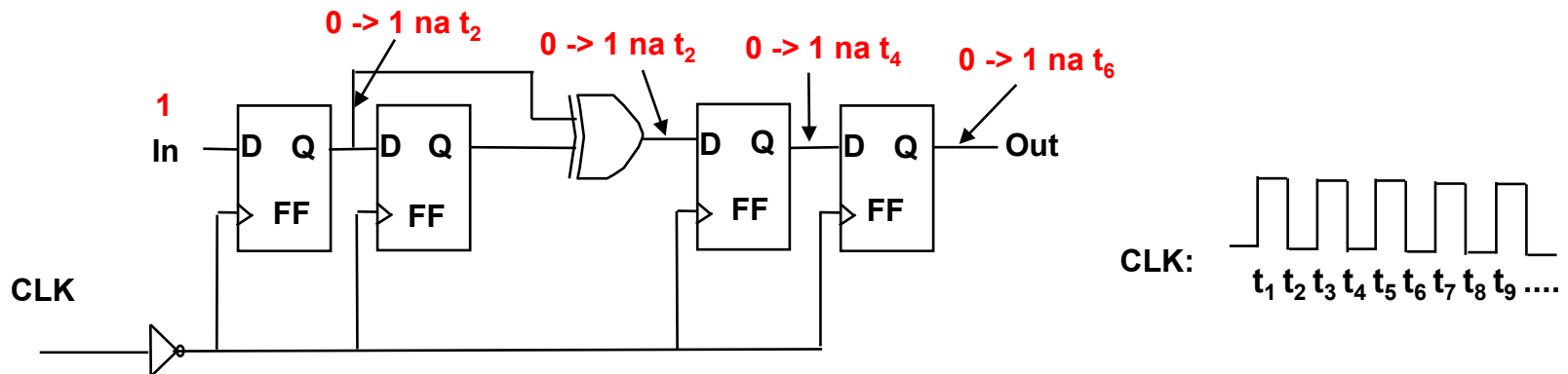
- a. een latch met geen verboden ingangscombinatie
- ✓ b. een latch; $AB = 10$ is de verboden ingangscombinatie
- c. geen bruikbaar geheugenelement: de Hold-combinatie ontbreekt
- d. geen van bovenstaande antwoorden



A	B	C	D
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Vraag 5

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



Indien voor tijdstip t_1 alle Qs zijn 0 en op 'In' het constante signaal 1 wordt gezet, wanneer wordt 1 op 'Out' waargenomen?
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:

	ACCUM.	C	Z
set c	?	1	?
ld 11110000b	11110000	1	?
and 11100001b	11100001		
	11100000	1	0
add 11011011b	11011011		
	10111100	1	0
st R1			
	10111100	1	0
xor 11111111b	11111111		
	01000011	1	0
add R1	10111100		
	00000000	1	0

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

- ✓ a. A = 0_H C = 1 Z = 0
- b. A = FF_H C = 0 Z = 0
- c. A = 0_H C = 1 Z = 1
- d. A = FF_H C = 1 Z = 1

Vraag 7

Gegeven is het volgende process statement:

```
proc1: process (a, b, c, d)
  begin
    ...
  end process;
  \ \ stippellijn.
```

Op de stippellijn kunnen de volgende drie codes met sequential signal assignment statements worden uitgevoerd:

```
code1: x <= a and b;
        y <= c xor d;
        z <= x or y;
```

```
code2: y <= c xor d;
        x <= a and b;
        z <= x or y;
```

```
code3: z <= x or y;
        x <= a and b;
        y <= c xor d;
```

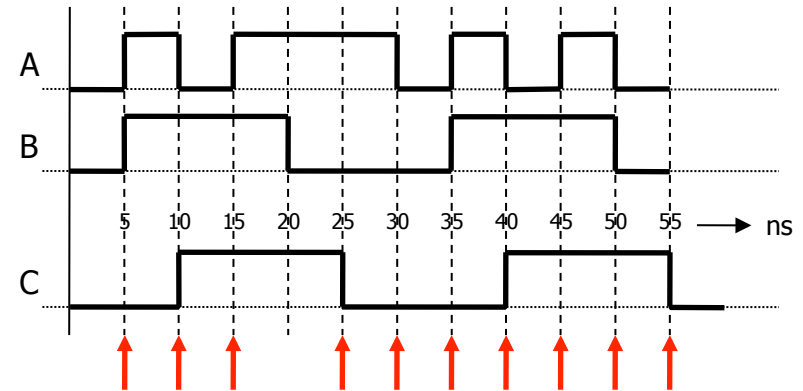
- a. Alleen code 1 is gelijk aan code 2.
- b. Alleen code 2 is gelijk aan code 3.
- c. Alleen code 1 is gelijk aan code 3.
- ✓ d. Alle drie de codes zijn gelijk aan elkaar.

Vraag 8

De volgende twee processen staan in dezelfde architecture body:

```
L1: process (B)
  begin
    C <= B after 5 ns;
  end process;

L2: process (A, C)
  begin
    D <= (A nand C) after 5 ns;
  end process;
```



Bij het starten van de discrete event simulator bestaat de volgende event list al:

A: (0,0ns), (1,5ns), (0,10ns), (1,15ns), (0,30ns), (1,35ns), (0,40ns), (1,45ns), (0,50ns)

B: (0,0ns), (1,5ns), (0,20ns), (1,35ns), (0,50ns)

Hoe vaak wordt de signal assignment "D <= ..." in process met label L2 uitgevoerd zonder dat de initialisatie wordt meegerekend?

- a. 4 keer.
- b. 9 keer.
- ✓ c. 10 keer.
- d. 11 keer.

Vraag 9

Hieronder is een (onvolledige) VHDL beschrijving van een entity met bijbehorende architecture gegeven:

```
entity schakeling is
    port (K, L : in bit;
          M    : out bit);
end entity schakeling;

architecture werking of schakeling is
    component circuit is
        port (A, B : in bit;
              C    : out bit);
    end component circuit;
    signal tussen : bit;
begin
    Lx:    ...
          M <= tussen;
end werking;
```

Welke mogelijkheden zijn geldige en correcte VHDL statements?

- ✓ a. L1 en L3.
- b. L2 en L3.
- c. L1 en L2.
- d. Alleen L1.

Er zijn drie mogelijkheden gegeven voor label Lx gegeven, namelijk:


```
L1: circuit port map (A=>K, B=>L, C=>tussen);
L2: circuit port map (K=>A, L=>B, tussen=>C);
L3: circuit port map (K, L, tussen);
```

Vraag 10

Gegeven is een hardware component met de volgende entity en architecture:

```
entity onbekend is
    port (a, b, c: in bit; d: out bit);
end onbekend;
architecture gedrag of onbekend is
    signal e: bit;
begin
    P1: process (b, c) is
        begin
            if (b='1') then e<='0';
            elsif (c'event) then
                if (a='1') then e<=not(e); end if;
            end if;
        end process;
        d<=e;
    end gedrag;
```

Welk van de onderstaande beweringen is juist?

- a. De component is een D-flipflop met c als klok die werkt op alleen de neergaande flank en heeft tevens een asynchrone reset.
- b. De component is een D-flipflop met c als klok die werkt op beide flanken en heeft tevens een asynchrone reset.
- c. De component is een T-flipflop met c als klok die werkt op beide flanken en heeft tevens een synchrone reset.
-  d. De component is een T-flipflop met c als klok die werkt op beide flanken en heeft tevens een asynchrone reset.

Deel B: Vraag 11-13

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 D is gepasseerd (na een timeout).

Vraag 11-13 vervolg

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 metingangsignaal S (reset) 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
                    T <= '0';           -- map count to T
                ELSE
                    T <= '1';
                END IF;
            END IF;
        END PROCESS;
    END timer_arch;
```

Vraag 11

Specificeer de gewenste besturing van de regelaar mbv. een FSD met gebruikmaking van de eerder beschreven signalen A, B, D, S, T.

Dit FSD vormt tevens de basis voor de navolgende vraag. (Ook als uw FSD fout is, wordt uw FSD als uitgangspunt gehanteerd bij de beoordeling.)

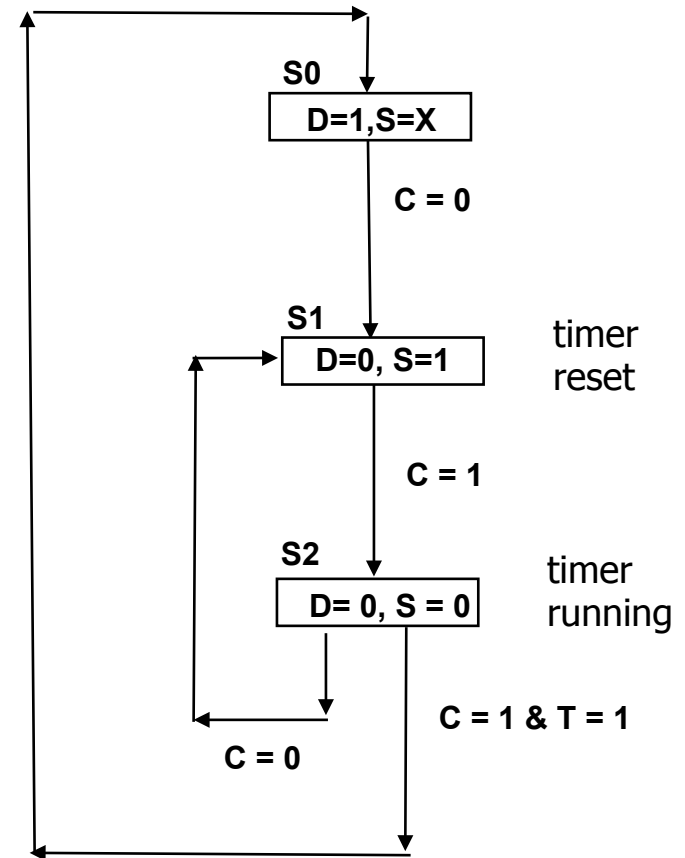
Kies voor de FSM een Moore machine.

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

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

Voor het gemak hebben we een signaal C geïntroduceerd volgens $C = A \text{ and } B$.

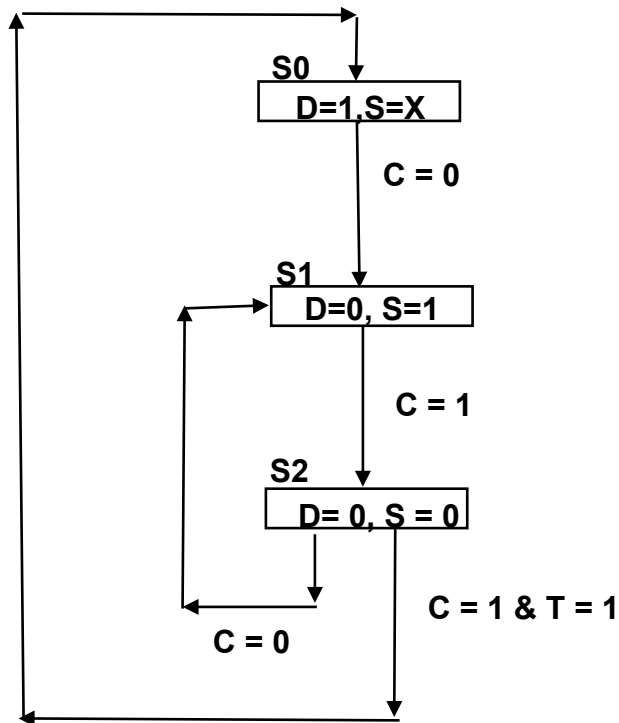
Dus $C = 0$ als iemand aan de deur



Vraag 12a

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

a. Schrijf de waarheidstabel op die aangeeft hoe de nieuwe state ($S_0/1/2$) en uitgangsignalen (D, S) afhangen van de huidige state en ingangsignalen (A, B, T).



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

Vraag 12b

b. Kies de volgende state assignment die de states S_i afbeeldt op de state-variabelen Q_j .
Teken de K-maps en leidt de vereenvoudigde expressies af voor de Q_j 's, D en S.

State	Q_1	Q_0
S0	0	0
S1	0	1
S2	1	1

Huidig	Volgend				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

CT	$Q_1 Q_0$				Q_0^{t+1}
	00	01	11	10	
00	1	1	1	X	
01	1	1	1	X	
11	0	1	0	X	
10	0	1	1	X	

CT	$Q_1 Q_0$				Q_1^{t+1}
	00	01	11	10	
00	0	0	0	X	
01	0	0	0	X	
11	0	1	0	X	
10	0	1	1	X	

Huidig	Volgend				Uitgang
	CT				
	00	01	10	11	DS
00	01	01	00	00	1x
01	01	01	11	11	01
11	01	01	11	00	00

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

Q_0	Q_1		D
	0	1	
0	1	X	
1	0	0	

$$D = Q_0'$$

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

Q_0	Q_1		S
	0	1	
0	X	X	
1	1	0	

$$S = Q_1'$$

Vraag 13

Nu gaan we gebruik maken van de synthesesmogelijkheden van VHDL. Ontwerp een VHDL specificatie van de besturing. U mag zelf weten welk nivo van abstractie u kiest voor uw beschrijving. De specificatie mag bv. 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 timeout
    CLK: IN bit         -- clock
);
END besturing;

ARCHITECTURE your_architecture OF besturing IS

    ....
    ....

END your_architecture;
```

Vraag 13 vervolg

```

ARCHITECTURE your_architecture OF besturing IS
  TYPE door_state is (S0, S1, S0);
  SIGNAL STATE, NEW_STATE: door_state;
  SIGNAL C: bit;
BEGIN
  C <= A and B;

  plb11:
  PROCESS (CLK)
  BEGIN
    if (CLK'EVENT and CLK='1') THEN
      STATE <= NEW_STATE;
    END IF;
  END PROCESS;

  plb12:
  PROCESS (C, T, STATE)
  BEGIN
    CASE STATE IS
      WHEN S0 =>
        D <= '1';
        S <= '0';
        IF (C = '0') THEN
          NEW_STATE <= S1;
        ELSE
          NEW_STATE <= S0;
        END IF;
    END CASE;
  END PROCESS;
END your_architecture;

```

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

```

  WHEN S1 =>
    D <= '0';
    S <= '1';
    IF (C = '0') THEN
      NEW_STATE <= S1;
    ELSE
      NEW_STATE <= S2;
    END IF;
  WHEN S2 =>
    D <= '0';
    S <= '0';
    IF (C = '0') THEN
      NEW_STATE <= S1;
    ELSIF (T = '1') THEN
      NEW_STATE <= S0;
    ELSE
      NEW_STATE <= S2;
    END IF;
  END CASE;
END PROCESS;
END your_architecture;

```

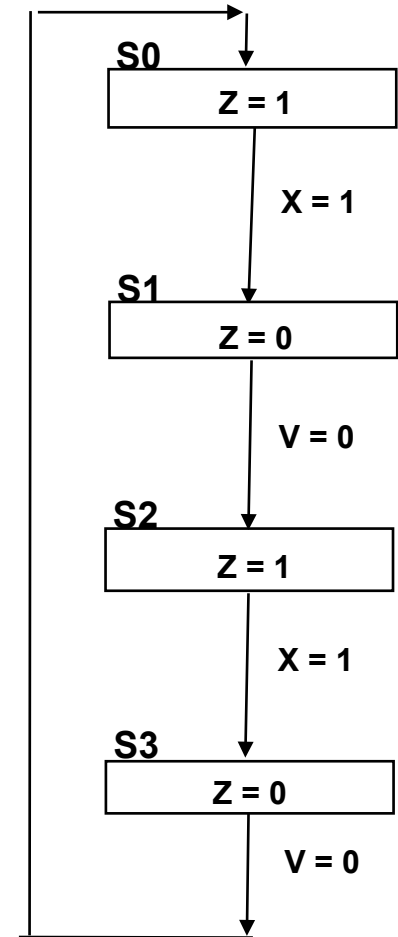
Tentamen 7 juli 2006 - vraag 10

Gegeven het volgende FSD. Stel dat we de toestand S gecodeerd hebben met 2 variabelen Q0, Q1 volgens:

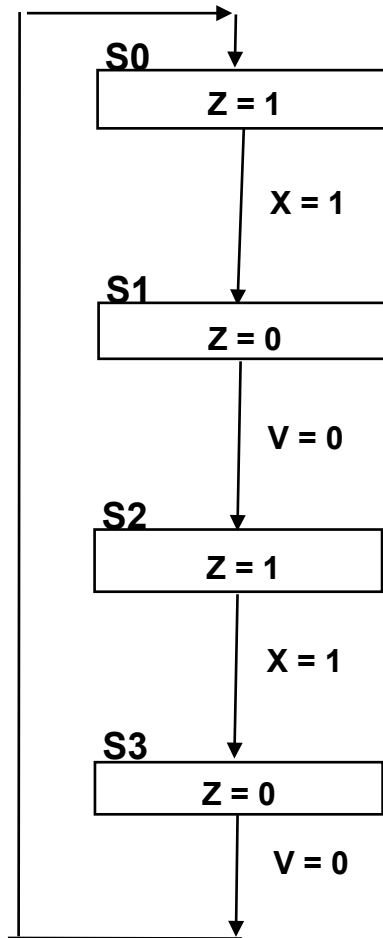
Q1	Q0	state
0	0	S0
0	1	S1
1	1	S2
1	0	S3

Stel dat het FSD met D flip-flops en 2-input NANDs moet worden gerealiseerd.
Hoeveel NANDs zijn *minimaal* benodigd?

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



Vraag 10 vervolg



De toestandstabel is:

x	v	Q ₁	Q ₀	(S ₀)	Q ₁ ⁺	Q ₀ ⁺	(S ₀)	z
0	-	0	0	(S ₀)	0	0	(S ₀)	1
1	-	0	0	(S ₀)	0	1	(S ₁)	1
-	1	0	1	(S ₁)	0	1	(S ₁)	0
-	0	0	1	(S ₁)	1	1	(S ₂)	0
0	-	1	1	(S ₂)	1	1	(S ₂)	1
1	-	1	1	(S ₂)	1	0	(S ₃)	1
-	1	1	0	(S ₃)	1	0	(S ₃)	0
-	0	1	0	(S ₃)	0	0	(S ₀)	0

		Q ₁ Q ₀		Q ₁	
		00	01	11	10
z	X V	1	0	1	0
	00	1	0	1	0
X	01	1	0	1	0
	11	1	0	1	0
	10	1	0	1	0
	00	1	0	1	0

$$z = Q_1' Q_0' + Q_1 Q_0 = ((Q_1' Q_0')'(Q_1 Q_0)')' \Rightarrow 3 \text{ NANDs}$$

		Q ₁ Q ₀		Q ₁	
		00	01	11	10
Q ₀ ⁺	X V	0	1	1	0
	00	0	1	1	0
X	01	0	1	1	0
	11	1	1	0	0
	10	1	1	0	0
	00	1	1	0	0

$$Q_0^+ = Q_0 x' + Q_1' x = ((Q_0 x')'(Q_1' x)')' \Rightarrow 4 \text{ NANDs}$$

		Q ₁ Q ₀		Q ₁	
		00	01	11	10
Q ₁ ⁺	X V	0	1	1	0
	00	0	0	1	1
X	01	0	0	1	1
	11	0	0	1	1
	10	0	1	1	0
	00	0	1	1	0

$$Q_1^+ = Q_0 v' + Q_1 v = ((Q_0 v')'(Q_1 v)')' \Rightarrow 4 \text{ NANDs}$$

Dus hebben we 4 + 4 + 3 = 11 NAND gates nodig
 $\Rightarrow c$