

OPGAVE 2:**5 + 5 PUNTEN**

Gegeven is de formule voor Z (de som van mintermen zijn gegeven met de don't cares (d))

$$F(A,B,C,D) = \sum m(0, 3, 8, 11, 15), d(A,B,C,D) = \sum m(7, 10, 12)$$

- a) Geef een minimale logische formule voor Z als sum-of-products (SOP).

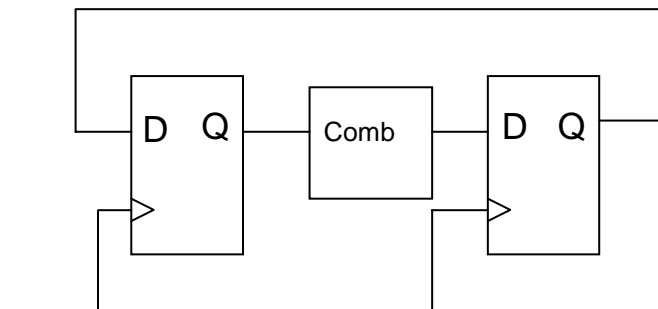
SOP

$$Z = !B.!C.!D + C.D$$

- b) Geef een minimale logische formule voor Z als products-of-sums (POS).

POS

$$Z = (A+!B) \cdot (C+!D) \cdot (!C+D)$$

OPGAVE 3**5 + 5 PUNTEN**

In bovenstaand schema komen twee identieke D-flipflops voor met de volgende relevante parameters: T_{co} (de vertraging tussen het aanpassen van de uitgang Q nadat er een opgaande flank van de klok is geweest), T_{su} (setup tijd) en T_h (hold tijd).

De combinatoriek (Comb) heeft een minimale vertraging van T_{min} ($T_{min} > 0$ seconden) en de maximale vertraging is T_{max} . De clock skew is T_{skew} . De overige vertraging in de bedrading wordt verder niet meegenomen.

- a) Gevraagd wordt de maximale klokfrequentie voor dit systeem (uitgedrukt in bovenstaande parameters).

$$f = 1 / (T_{co} + T_{max} + T_{su} + T_{skew})$$

- b) Gevraagd wordt de maximaal toelaatbare clock skew (uitgedrukt in bovenstaande parameters).

$$T_{skew} < T_{co} - T_h$$

Naam:

OPGAVE 4**4 + 3 + 4 PUNTEN**

- a) Converteer het niet gehele decimale getal 611,625 naar octaal en hexadecimaal.

Octaal	1143,5
Hexadecimaal	263,A

- b) Wat is voor een 8-bits processor het bitpatroon van het decimale getal -20 in achtereenvolgens Signed Magnitude notatie, 1-complement notatie en 2-complement notatie?

Signed Magnitude	10010100
1-complement	11101011
2-complement	11101100

- c) Bij 8-bits 2-complement notatie: wat is het kleinste positieve decimale getal dat bij het decimale getal -20 opgeteld moet worden zodat er voor het eerst een “Carry-Out” optreedt?

20

Naam:

OPGAVE 5

7 PUNTEN

Voor de communicatie wordt aan de data i een pariteitsbit toegevoegd. Er wordt gebruik gemaakt van even pariteit, d.w.z. het totaal aantal enen in de data i en pariteitsbit is even.

Gevraagd wordt de VHDL beschrijving voor de combinatorische schakeling waarmee het dit pariteitsbit o bepaald kan worden. De entity beschrijving is al gegeven.

```

     $n \leq 2^p - 1$ 
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
ENTITY even_parity IS
    GENERIC (width : integer := 3);
    PORT (i : IN std_logic_vector(width-1 DOWNT0 0); --data
          o : OUT std_logic); -- pariteitsbit
END even_parity;

ARCHITECTURE behaviour OF even_parity IS
BEGIN
    PROCESS(i)
        VARIABLE tmp : std_logic;
    BEGIN
        tmp := '0';
        FOR lp IN i'RANGE LOOP
            tmp:=tmp XOR i(lp);
        END LOOP;
        o <= tmp;
    END PROCESS;
END behaviour;
```

OPGAVE 6**3 + 3 + 6 PUNTEN**

Voor de testbaarheid van een systeem wordt naar de *controllability* en *observability* gekeken. Geef een omschrijving van deze termen.

Omschrijving controllability

geeft aan hoe in hoeverre nodes in het systeem
'1' of '0' gemaakt kunnen worden

Omschrijving observability

geeft aan in hoeverre de waarde van elke interne node zichtbaar
te maken is aan de uitgang

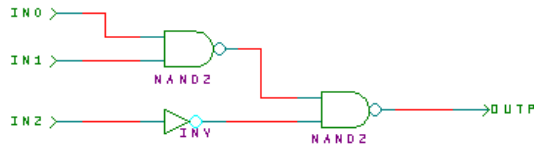
Om een ontwerp testbaar te maken wordt bij het ontwerpen ervan rekening gehouden met de testbaarheid. Dit kan o.a. door het toevoegen van scan-paden. Geef drie nadelen van het toevoegen van scanpaden:

1. meer I/O pinnen (scan-input, mode FF)
2. area overhead (scanFF)
3. extra delay tgv mux nodig voor scanlogica F

Naam:

OPGAVE 7

5 PUNTEN



```
ENTITY greater IS
    PORT (in2,in1,in0 : in bit;
          outp : out bit
          );
END greater;

ARCHITECTURE gates OF greater IS
BEGIN
    outp <= in2 OR (in1 AND in0);
END gates;
```

Gegeven is de VHDL beschrijving greater. Is het gegeven schema een correcte realisatie van deze VHDL beschrijving. Motiveer uw antwoord.

Is het schema een correcte realisatie: Ja/NEE: ja ...
Motivatie: a=inverteer de statement tweemaal, dan de morgan toepassen
Dit resultaat in het schema (2 NANDS en inverter)

Gevraagd wordt de toestandsmachine van een seriële optelschakeling. Er zijn 2 data-ingangen voor de twee operanden A en B. De beide operanden hebben willekeurige lengte (de lengte van de beide operanden is gelijk). De data wordt van 'least significant bit' naar 'most significant bit' aangeboden. Op elke actieve flank van de klok worden de volgende twee bits van beide operanden aangeboden. Als het n^e ingangsbite van beide operanden aanwezig is op de ingang staat op de uitgang het n^e bit van het resultaat. Naast beide data-ingangen is er nog een R (reset) ingang. Indien R is '1' wordt de schakeling in de begintoestand gebracht. Als R is '1' is het uitgangssignaal '0'. Als R is '0' wordt de optelbewerking zoals hiervoor is beschreven uitgevoerd.

1) Waarom is dit een Mealy toestandsmachine

De uitgang hangt direct af van de ingang

2) Geef een toestandsmachine met minimum aantal toestanden

state	input			next state	Output
	A	B	R		
S0	-	-	1	S0	0
S0	0	0	0	S0	0
S0	0	1	0	S0	1
S0	1	0	0	S0	1
S0	1	1	0	S1	0
S1	-	-	1	S0	0
S1	0	0	0	S0	1
S1	0	1	0	S1	0
S1	1	0	0	S1	0
S1	1	1	0	S1	1

Naam:

OPGAVE 9

5 + 5 + 5 PUNTEN

Onderstaande toestandstabel beschrijft een toestandsmachine van een synchroon systeem. De codering van de vier toestanden is achter de toestand (=state) tussen haakjes gegeven. Voor flipflop Q2 wordt een T-flipflop gebruikt en voor flipflop Q1 wordt een JK-flipflop gebruikt. Verder heeft de schakeling naast de klok ingang nog de ingangen A en B en uitgang Y.

State (Q ₂ Q ₁)	next state				Y
	AB=00	AB=01	AB=10	AB=11	
S0 (00)	S0	S0	S2	S2	0
S1 (01)	S1	S1	S3	S3	0
S2 (10)	S3	S3	S1	S1	0
S3 (11)	S3	S2	S1	S0	1

- a) Geef een minimale logische formule in SOP vorm voor de T ingang van de T-flipflop Q2.

T= A

- b) Geef een minimale logische formule in SOP vorm voor de K ingang van de JK-flipflop Q1.

K= Q₂• B

- c) Na de ontwerpfase blijkt de T-flipflop niet meer geleverd te kunnen worden maar is er nog wel een JK-flipflop beschikbaar. Kan dit systeem ook gerealiseerd worden met twee JK-flipflops, motiveer uw antwoord

Kan het systeem met 2 JK-flipflops worden gerealiseerd? ja/nee

Motivatie:

J en K met elkaar verbinden dan heb je een T-FF, of elke FF kan altijd

OPGAVE 10

8 PUNTEN

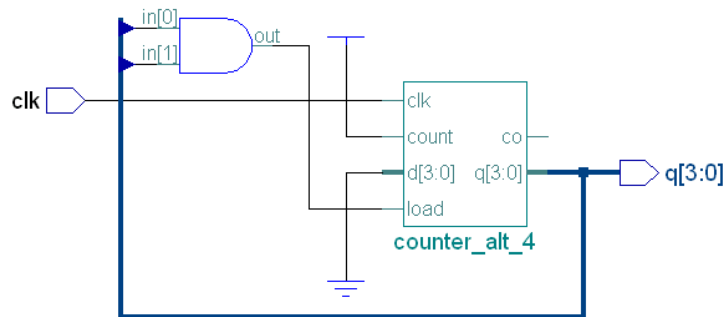
```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
ENTITY counter_alt IS
    GENERIC (w : integer := 8);
    PORT (clk, load, count : IN std_logic;
          d : IN unsigned(w-1 downto 0);
          q : OUT unsigned(w-1 downto 0);
          co : OUT std_logic);
END counter_alt;

ARCHITECTURE bhv OF counter_alt IS
    SIGNAL qi : unsigned(w downto 0);
BEGIN
    PROCESS(clk)
    BEGIN
        IF rising_edge(clk) THEN
            IF load='1'
            THEN qi <= '0' & d;
            ELSE qi <= qi + (" " & count);
            END IF;
        END IF;
    END PROCESS;
    q <= qi(w-1 downto 0);
    co <= qi(w);
END bhv;

```

Bovenstaand ontwerp wordt gebruikt om een BCD teller mee te ontwerpen. Het ontwerp is in onderstaand schema gegeven. (de count ingang is met '1' verbonden en de d ingang is met '0' verbonden).



De entity beschrijving van dit ontwerp is gegeven. Gevraagd wordt de architecture beschrijving waarbij gebruik wordt gemaakt van een instantiatie van de entity *counter_alt* waarbij de generic *w* de waarde 4 moet krijgen.

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
ENTITY bcd_impl IS
    PORT (clk : IN std_logic;
          q : OUT unsigned(3 downto 0));
END bcd_impl;

ARCHITECTURE structure OF bcd_impl IS
    SIGNAL qi : unsigned(3 DOWNTO 0);
    SIGNAL load : std_logic;

BEGIN
    cnt : ENTITY work.counter_alt(bhv)
        GENERIC MAP (w=>4)
        PORT MAP (clk=>clk, load=>load,
                  count=>'1', d=>"0000", q=>qi, co=>OPEN);

    load <= qi(0) AND qi(3);
    q <= qi;

```

Naam:

END structure;