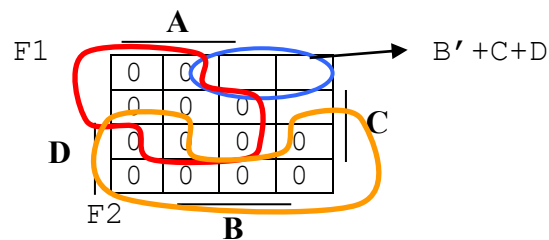


Antwoorden en Uitwerkingen

Opgave		Opgave		Opgave		Opgave	
1	B	6	A	11	B	16	D
2	C	7	C	12	C	17	D
3	B	8	C	13	A	18	B
4	A	9	B	14	D	19	D
5	B	10	D	15	B	20	D

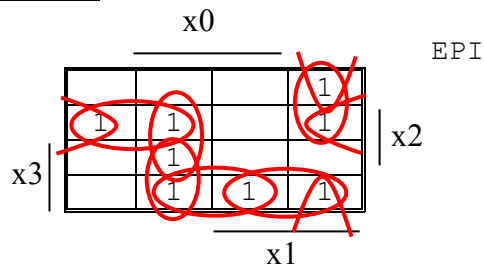
Opgave 1: (b)

Let op: de enable ingangen van de tri-state buffers zijn geïnverteerd. Dus is het van belang om te weten wanneer functies F1 en F2 nul zijn. De functie F1 en F2 staan al in product van somtermen notatie en kunnen dus makkelijk in een Karnaugh-map worden geplaatst:



De antwoorden kunnen als maxtermen (de nullen) worden gelezen en deze kunnen eenvoudig worden geplaatst binnen de getoonde Karnaugh-map. Hieruit kan worden gelezen dat antwoord $B' + C + D$ het enige antwoord is waarbij de nullen niet samenvallen met die van F1 en F2.

Opgave 2: (c)



Hieruit is af te lezen dat er 8 PI's zijn en dat er geen enkele essentieel is omdat elke 1 door minstens twee PI worden gecovered.

Opgave 3: (b)

x3	x2	x1	x0	A=x1+x0	B=x3+x2	C=A ⊕ B	D=x1.x0'	C + D	j
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	1	0	1	1
0	0	1	0	1	0	1	1	1	2
0	0	1	1	1	0	1	0	1	3
0	1	0	0	0	1	1	0	1	4
0	1	0	1	1	1	0	0	0	5
0	1	1	0	1	1	0	1	1	6
0	1	1	1	1	1	0	0	0	7
1	0	0	0	0	1	1	0	1	8
1	0	0	1	1	1	0	0	0	9
1	0	1	0	1	1	0	1	1	10
1	0	1	1	1	1	0	0	0	11
1	1	0	0	0	1	1	0	1	12
1	1	0	1	1	1	0	0	0	13
1	1	1	0	1	1	0	1	1	14
1	1	1	1	1	1	0	0	0	15

Opgave 4: (a)

CYCLE					(NOR)	UITGANGEN		
	D	C	B	A	z	EXOR	AND	EXOR
0	0	0	1	1	1	0	0	1
1	1	0	0	1	0	1	0	1
2	1	1	0	0	0	0	0	0
3	0	1	1	0	0	1	1	1
4	1	0	1	1	0	1	1	0
5	0	1	0	1	0	1	0	1
6	1	0	1	0	0	1	1	1
7	1	1	0	1	0	0	0	1
8	1	1	1	0	0	0	0	0
9	0	1	1	1	0	1	1	0
10	0	0	1	1	1			

Opgave 5: (b)

Langste pad is duidelijk door de EXOR & AND & EXOR poorten met daarbij opgeteld de omslagtijd en setuptijd van de shiftregisters:

$$10 + 8 + 10 + 4 + 10 = 42 \text{ nsec} \rightarrow 23.81 \text{ MHz} \rightarrow \text{maximaal } 23 \text{ MHz}$$

Opgave 6: (a)

Eerst beschouwen we de uitgangen om de 1-equivalentie te bepalen. Er blijken 2 klassen van states te zijn waarbij de uitgang hetzelfde reageert op de ingangswaarde. We kijken vervolgens voor elke state tot welke klasse de volgende state behoort, afhankelijk van de ingangswaarde.

	1						2		1-equiv
	(A	B	D	E	G	H)	(C	F)	
X=0	1	1	1	2	1	1	1	1	
X=1	1	2	1	1	1	1	1	2	

Hieruit blijkt dat klasse (A B D E G H) zich splitst in 3 aparte klassen (A D G H), (B) en (E) die dus niet meer 2-equivalent zijn.

	1				2	3	4	5	2-equiv
	(A	D	G	H)	(B)	(E)	(C)	(F)	
X=0	2	1	2	1					
X=1	3	1	3	1					

Vervolgens blijkt dat klasse (A D G H) zich splitst in:

	1		2		3	4	5	6	3-equiv
	(A	G)	(D	H)	(B)	(E)	(C)	(F)	

Hieruit kan worden afgeleid dat alleen toestanden D en G wel 2-equivalent en niet 3-equivalent zijn.

Opgave 7: (c)

aantal klokcycli	mod-16 counter				mod-4 counter			
	S3	S2	S1	S0	TC	S1	S0	TC = y
13	0	0	1	1	0	0	1	0
	...				0	0	1	0
	1	1	1	1	1	0	1	0
9	0	1	1	1	0	1	0	0
	...				0	1	0	0
	1	1	1	1	1	1	0	0
5	1	0	1	1	0	1	1	1
	...				0	1	1	1
	1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	0	0

tijd

Opgave 8: (c)

Uit de eerste mux kan worden afgelezen dat de F moet zijn:
 $A.B.C'D' + (A'+B').C.D' + (A'+B).C'D + A.B'.C.D$. Dit resulteert in de volgende Karnaughmap:

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

in2 op de tweede mux betekent dat C=1 en A=0 → dus de omkaderde waardes. Hieruit kan eenvoudigweg worden afgelezen dat de expressie D' is. Het invullen van C=1 en A=0 in de logische expressie van F resulteert in dezelfde antwoord.

Opgave 9: (b)

x3	x2	x1	x0	z	j
0	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	0	2
0	0	1	1	0	3
0	1	0	0	1	4
0	1	0	1	1	5
0	1	1	0	0	6
0	1	1	1	0	7
1	0	0	0	1	8
1	0	0	1	1	9
1	0	1	0	1	10
1	0	1	1	0	11
1	1	0	0	1	12
1	1	0	1	1	13
1	1	1	0	1	14
1	1	1	1	1	15

Opgave 10: (d)

De volgende Karnaughmap kan worden opgesteld als we we voor B een D flipflop gebruiken.

B(new) :

	A			
	1	1	1	1
C	0	0	1	0
	B(old)			

Dit kan worden vertaald naar de volgende J,K Karnaughmaps gebruik makende van de excitatie functies:

J(new) :

	A			
	1	1	-	-
C	0	0	-	-
	B			

K(new) :

	A			
	-	-	0	0
C	-	-	0	1
	B			

Dus: $J = C'$ en $K = A'C$.

Opgave 11: (b)

De laatste NAND kan worden geconverteerd naar een OR met geïnverteerde ingangen en deze heffen de inverse van de ervoor liggende NAND poorten - resulterende in normale AND poorten.

$$\begin{aligned} F &= (A \oplus B) \cdot (C \oplus D) + (B \oplus C) \cdot (A \oplus D) \\ &= \dots = A'BC'D + A'BCD' + AB'C'D + AB'CD' + \\ &\quad A'B'CD + AB'CD' + A'BC'D + ABC'D' \end{aligned}$$

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

Dit resulteert dus in de volgende product van somtermen:

$$\begin{aligned} &(A+B+C) \cdot (A+B+D) \cdot (A'+B'+C') \cdot (A'+B'+D') \cdot \\ &(A+C+D) \cdot (B+C+D) \cdot (A'+C'+D') \cdot (B'+C'+D') \end{aligned}$$

Opgave 12: (c)

De eerste sectie vanaf links zal een logische 0 geven wanneer en alleen wanneer $(A + B) \cdot C$ is 1 (pad naar aarde via de nmos transistors en geen pad naar vdd via de pmos transistors). Het resultaat wordt geïnverteerd door de 2^e sectie, dus $F = 1$ wanneer en alleen wanneer $(A + B) \cdot C$ is 1.

Opgave 13: (a)

Opgave 14: (d)

Alle getallen boven 512 moeten met 4 cijfers in het 8-tallig stelsel worden geschreven. Van 512 tot en met 999 kunnen met 3 decimale cijfers worden geschreven $\rightarrow 1000 - 512 = 488$ cijfers.

Opgave 15: (b)

Code 1 lijkt op eerste instantie op de code voor een asynchrone reset, maar let op dat het reset-sigitaal is vergeten in de sensitivity-lijst. Dus dit process wordt alleen uitgevoerd als de het clk-sigitaal verandert, dus ook de test op het reset-sigitaal is synchroon. (let op: bij synthese zal wel een asynchrone reset worden geïmplementeerd, omdat bij synthese de sensitivity-lijst opnieuw wordt opgebouwd en dus anders zal zijn zoals nu is geschreven.

Code 2 is de standaard-code voor een synchrone reset. Alhoewel het reset-sigitaal in de sensitivity-lijst staat blijft het een synchrone reset omdat deze pas wordt geëvalueerd binnen de if-statement met het clk-sigitaal in de conditie.

Opgave 16: (d)

Na invullen van nullen als mintermen - gezien de inverse aan het eind van de expressie - kan de volgende Karnaughmap worden bepaald:

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

De expressie die hierbij hoort is: $F = A'B' + CD'$

Opgave 17: (d)

Op tijdstip $(10 \text{ ns} + \Delta)$ neemt sigitaal a de waarde van b over. Omdat dit niet overeenkomt met de klokflank moet dit process wachten op de volgende klokflank voordat de nieuwe waarde in a kan worden overgenomen door sigitaal c. Dit gebeurt op $(20\text{ns} + \Delta)$.

Opgave 18: (b)

j	y3	y2	y1	y0	Y3	Y2	Y1	Y0	TOGGLE?
3	0	0	1	1	0	1	0	0	1
4	0	1	0	0	0	1	0	1	0
5	0	1	0	1	0	1	1	0	0
6	0	1	1	0	0	1	1	1	0
7	0	1	1	1	1	0	0	0	1
8	1	0	0	0	1	0	0	1	0
9	1	0	0	1	1	0	1	0	0
10	1	0	1	0	1	0	1	1	0
11	1	0	1	1	1	1	0	0	1
12	1	1	0	0	0	0	1	1	1

Deze waarden in een Karnaughmap zetten levert het volgende op:

T(Y2)

		y0		
	-	-	1	-
y2	0	0	1	0
y2	1	-	-	-
	0	0	1	0
			y1	
			y3	

De logische expressie is dan : $y_0y_1 + y_2y_3$

Opgave 19 : (d)

In welke gevallen verandert z_5 van waarde ten opzichte van x_5 ?
Dat zijn de volgende gevallen:

	x_5	x_4	x_3	x_2	x_1	x_0	s_0	→	z_5	z_4	z_3	z_2	z_1	z_0
-	0	1	1	1	1	0	1	(+2)	1	0	0	0	0	0
-	0	1	1	1	1	1	0	(+1)	1	0	0	0	0	0
-	0	1	1	1	1	1	1	(+2)	1	0	0	0	0	1
-	1	1	1	1	1	0	1	(+2)	0	0	0	0	0	0
-	1	1	1	1	1	1	0	(+1)	0	0	0	0	0	0
-	1	1	1	1	1	1	1	(+2)	0	0	0	0	0	1

Hieruit kan worden gelezen dat $z_5 = x_5 \oplus x_4 \cdot x_3 \cdot x_2 \cdot x_1 \cdot (s_0 + x_0)$

Opgave 20: (d)

De excess-3 code heeft de eigenschap dat het binair de volgende waarde voorstelt "binair(X+3)". Als dus twee getallen worden opgeteld, dan zal $v_3v_2v_1v_0$ een binaire waarde "binair(X+Y+6)" voorstellen, dus kunnen we de volgende tabel samenstellen.

Bv. om van een 6 een 3 te maken moeten we daar 13 (1101) bij optellen en om van een 0 een 3 te maken moeten we daar 3 (0011) bij optellen.

Getal	Cu	$v_3v_2v_1v_0$	$s_3s_2s_1s_0$	cuit	$k_3k_2k_1k_0$
0	0	0110 (6)	0011 (3)	0	1101
1	0	0111 (7)	0100 (4)	0	1101
2	0	1000 (8)	0101 (5)	0	1101
3	0	1001 (9)	0110 (6)	0	1101
4	0	1010 (10)	0111 (7)	0	1101
5	0	1011 (11)	1000 (8)	0	1101
6	0	1100 (12)	1001 (9)	0	1101
7	0	1101 (13)	1010 (10)	0	1101
8	0	1110 (14)	1011 (11)	0	1101
9	0	1111 (15)	1100 (12)	0	1101
10	1	0000 (0+16)	0011 (3)	1	0011
11	1	0001 (1+16)	0100 (4)	1	0011
12	1	0010 (2+16)	0101 (5)	1	0011
13	1	0011 (3+16)	0110 (6)	1	0011
14	1	0100 (4+16)	0111 (7)	1	0011
15	1	0101 (5+16)	1000 (8)	1	0011

16	1	0110 (6+16)	1001 (9)	1	0011
17	1	0111 (7+16)	1010 (10)	1	0011
18	1	1000 (8+16)	1011 (11)	1	0011

Hieruit is makkelijk te lezen dat $k_3 = \text{cuit}'$, $k_2 = \text{cuit}'$, $k_1 = \text{cuit}$, en dat $k_0 = 1$ zijn.