

Uitwerking gemaakt door E. van de Logt, september 2020. Opmerkingen en correcties: logeh@hr.nl.

Dit document omvat de uitwerkingen van de volgende opgaven van hoofdstuk 7:

Tabel 1: H1 Introduction

Paragraaf	Titel	Opgaven
7.1	Basic Logic Circuit Concepts	P7.1 t/m P7.5
7.2	Representation of Numerical Data in Binary Form	P7.6 t/m P7.21
7.3	Combinatorial Logic Circuits	P7.22 t/m P7.37
7.4	Synthesis of Logic Circuits	P7.38 t/m P7.52 M.u.v. P7.51
7.5	Minimization of Logic Circuits	P7.53 t/m P7.74
7.6	Sequential Logic Circuits	P7.75 t/m P7.88

Errata 1: Opgave P7.11: a) 117.15 moet 117.25 zijn, d) en e) alleen getal voor de komma

Chapter 7 Logic Circuits

7.1 Basic Logic Circuit Concepts

- P7.1 A** Als ruis toegevoegd wordt aan een digitaal signaal, kunnen de logische niveaus nog steeds bepaald worden, als tenminste de amplitude van de ruis niet te groot is. Van een analoog signaal kan ruis meestal niet compleet verwijderd worden.
- B** Componentwaarden in digitale circuits hoeven meestal niet zo precies te zijn als bij analoge circuits.
- C** Hele complexe digitale logische circuits met miljoenen componenten kunnen op een economische manier gemaakt worden. Analoge circuits hebben vaak grote capaciteiten nodig en/of zeer preciese componentwaarden die onmogelijk te maken zijn bij integrated-circuits.
- P7.2 A** Een bit is een binair symbool die de waarden 0 of 1 kan hebben.
- B** Een byte is een binair getal dat uit 8 bits bestaat.
- C** Een nibble is een binair getal dat uit 4 bits bestaat.
- P7.3** Bij positieve logica wordt de logische waarde 1 weergegeven door een hoger spanningsbereik dan bij een logische waarde 0. Bij negatieve logica is dit precies andersom (het hoogste spanningsbereik hoort dan bij een logische 0).
- P7.4** De noise-margins bij een logisch circuit worden gedefinieerd met $NM_L = V_{IL} - V_{OL}$ en $NM_H = V_{OH} - V_{IH}$.
Hierbij is V_{IL} de grootste inputspanning die nog wordt geaccepteerd voor een logische 0.
 V_{IH} is de laagste inputspanning die nog wordt geaccepteerd voor een logische 1.
 V_{OL} is de hoogste outputspanning voor een logische 0.
 V_{OH} is de laagste outputspanning voor een logische 1.
- P7.5** Bij seriële transmissie worden de bits van een woord een voor een verstuurd over een enkel paar draden. Bij parallelle transmissie worden meerdere bits tegelijk verstuurd over een aantal parallelle draden.

7.2 Representation of Numerical Data in Binary Form

P7.6 A 5.625

B 7.75

C 10.25

D 7.875

E 8.3125

F 21.375

P7.7 A $11_{10} = 1011_2 = 00010001_{BCD}$

B $7_{10} = 111_2 = 00000111_{BCD}$

C $1.25_{10} = 0001.01_2 = 0001.00100101_{BCD}$

D $100.75_{10} = 01100100.11_2 = 000100000000.01110101_{BCD}$

E $22.13_{10} = 00100010.00010011_{BCD}$

P7.8 A Voor getallen van 0 tot 100 zijn 7 bits nodig ($2^7 = 128$)

B Voor getallen van 0 tot 1000 zijn 10 bits nodig ($2^{10} = 1024$)

C Voor getallen van 0 tot 10^6 zijn 20 bits nodig ($2^{20} = 1048576$)

P7.9 A $11 + 100.11 = 111.11$

B $100 + 1010.011 = 1110.011$

C $1000\ 0111 + 111.11 = 10001110.11$

P7.10 A $10010011.0101_{BCD} + 00110111.0001_{BCD} = 93.5_{10} + 37.1_{10} = 130.6_{10} = 000100110000.0110_{BCD}$

B $01011000.1000_{BCD} + 10001001.1001_{BCD} = 58.8_{10} + 89.9_{10} = 148.7_{10} = 000101001000.0111_{BCD}$

P7.11 A Let op: 117.25 i.p.v. 117.15: $117.25_{10} = 01110101.01_2 = 75.4_{16} = 165.2_8$

B $193.5_{10} = 11000001.1_2 = C1.8_{16} = 301.4_8$

C $102.5_{10} = 01100110.1_2 = 66.8_{16} = 146.4_8$

D Let op: met 300 rekenen: $300_{10} = 000100101100_2 = 12C_{16} = 454_8$

E Let op: met 411 rekenen: $411_{10} = 000110011011_2 = 19B_{16} = 633_8$

P7.12 A $+19 = 0001\ 0011$ (8-bit signed two's complement)

B $-19: 256 - 19 = 237 = ED_{16} = 1110\ 1101$

C $+75 = 4B_{16} = 0100\ 1011$

D $-87: 256 - 87 = 169 = A9_{16} = 1010\ 1001$

E $-95: 256 - 95 = 161 = A1_{16} = 1010\ 0001$

F $+99 = 63_{16} = 0110\ 0011$

P7.13 A $11.1_{16} = 00010001.1000_2 = 021.4_8 = 17.5_{10}$

B $A2.F_{16} = 10100010.11110010_2 = 242.744_8 = 162.9453125_{10}$

C $B.D_{16} = 1011.1101_2 = 13.64_8 = 11.8125_{10}$

P7.14 A $707.10_8 = 111000111.001000_2 = 1C7.2_{16} = 455.125_{10}$

B $113.5_8 = 001001011.101_2 = 4B.A_{16} = 75.625_{10}$

C $2.135_8 = 0010.001011101_2 = 2.2E8_{16} = 11.181640625_{10}$

P7.15 A $777_{10} + 1 = 778_{10}$

B $777_8 + 1 = 1000_8$

C $777_{16} + 1 = 778_{16}$

P7.16 A 3 bit binair: 0 tot en met 7 decimaal

B 3 cijfers octaal: 0 tot en met $777_8 = 8^3 - 1 = 511$

C 3 cijfers hex: 0 tot en met $FFF_{16} = 16^3 - 1 = 4095$

P7.17 Gray-code behoort niet tot de tentamenstof

P7.18 A $FF.FF_{16} = 255.99609375_{10}$

B $727.128_8 = 471.171875_{10}$

C $3FF.A1B_{16} = 1023.631607055_{10}$

D $110.010111_2 = 6.359375_{10}$

E $7007.17_8 = 3591.234375_{10}$

P7.19 A 0001 1111: 1C = 1110 0000 ; 2C = 1110 0001

B 0001 0001: 1C = 1110 1110 ; 2C = 1110 1111

C 0000 0110: 1C = 1111 1001 ; 2C = 1111 1010

D 0000 1010: 1C = 1111 0101 ; 2C = 1111 0110

E 0000 1111: 1C = 1111 0000 ; 2C = 1111 0001

P7.20 A

+ 44	00101100
+ 45	00101101
— — —	— — — — — — — —
+ 89	01011001

B

− 39	11011001
+ 92	01011100
— — —	— — — — — — — —
+ 53	00110101

C

− 75	10110101
+ 59	00111011
— — —	— — — — — — — —
− 16	11110000

D

− 1	11111111
+ 1	00000001
— — —	— — — — — — — —
0	00000000

E

+ 10	00001010
− 3	11111101
---	-----
+ 7	00000111

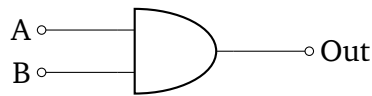
P7.21 Een overflow/underflow ontstaat als twee getallen die bij elkaar opgeteld worden hetzelfde teken hebben en het resultaat een ander teken heeft.

7.3 Combinatorial Logic Circuits

P7.22 Een waarheidstabel laat alle mogelijke combinaties van ingangssignalen zien met daarbij de waarde(n) van de output(s).

P7.23 De wetten van de Morgan luiden als volgt: Als de variabelen in een logische formule geïnverteerd worden, de AND operaties vervangen worden door OR operaties en de OR operaties worden vervangen door AND operaties en de gehele formule geïnverteerd wordt, dan blijft de logische formule hetzelfde. In formulevorm:

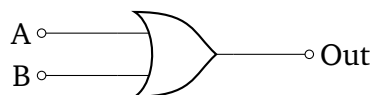
$$A \cdot B \cdot C = \overline{\overline{A} + \overline{B} + \overline{C}}, \text{ maar ook } (A + B + C) = \overline{\overline{A} \cdot \overline{B} \cdot \overline{C}}$$



Schakeling 1: AND poort

Tabel 2: Waarheidstabel AND poort

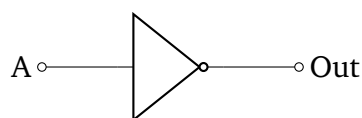
A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1



Schakeling 2: OR poort

Tabel 3: Waarheidstabel OR poort

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1

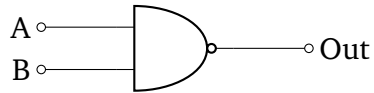


Schakeling 3: Inverter poort

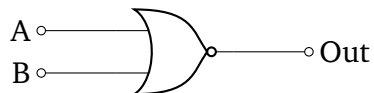
P7.24

Tabel 4: Waarheidstabel Inverter poort

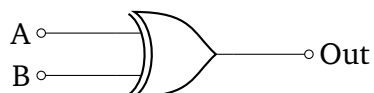
A	Out
0	1
1	0

**Schakeling 4:** NAND poort**Tabel 5:** Waarheidstabel NAND poort

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

**Schakeling 5:** NOR poort**Tabel 6:** Waarheidstabel NOR poort

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

**Schakeling 6:** XOR poort**Tabel 7:** Waarheidstabel XOR poort

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

Tabel 8: Tabel met mogelijkheden 0..7

A	B	0	AND	$A\bar{B}$	A	$\bar{A}B$	B	XOR	OR
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

Tabel 9: Tabel met mogelijkheden 8..15

A	B	NOR	XNOR	$\bar{B}$	$A + \bar{B}$	$\bar{A}$	$\bar{A} + B$	NAND	1
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

P7.25 Een manier om te kijken of de vergelijking links en rechts van het = teken hetzelfde zijn, is door van beide vergelijkingen een waarheidstabel te maken. Daarmee kun je aantonen dat voor alle mogelijke input-combinaties de linker- en rechtervergelijking hetzelfde opleveren.

Tabel 10: Waarheidstabel P7.26 a: $D = A \cdot B \cdot C + \overline{A \cdot C}$

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

P7.26

P7.27 A $F = \bar{A}B + C$

B $F = \overline{\bar{A}B \cdot C} + C = \overline{C(\bar{A}B + 1)} = \bar{C}$

C $F = \overline{\bar{A} + B + \bar{C}} \cdot C = (A + B + C) + \bar{C} = 1$

Tabel 11: Waarheidstabel P7.26 b: $E = \overline{A \cdot B} + C \cdot D + A \cdot C$

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

Tabel 12: Waarheidstabel P7.26 c: $Z = \overline{A \cdot B} + A \cdot B\bar{C}$

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

P7.28 Beide tabellen hebben dezelfde output voor alle combinaties van ingangssignalen, dus $(A + \bar{B})(\bar{A} + C) = AC + \bar{A}\bar{B}$

P7.29 Beide tabellen hebben dezelfde output voor alle combinaties van ingangssignalen, dus $\overline{(A + B)(A + C)} = (\bar{B} + \bar{C})\bar{A}$

Tabel 13: Waarheidstabel P7.26 d: $D = \overline{A \cdot B} + \overline{A \cdot C}$

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

Tabel 14: Waarheidstabel P7.28 links

A	B	C	$A + \bar{B}$	$\bar{A} + C$	$(A + \bar{B})(\bar{A} + C)$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	1	1	1

Tabel 15: Waarheidstabel P7.28 rechts

A	B	C	AC	$\bar{A}\bar{B}$	$AC + \bar{A}\bar{B}$
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	1	0	1
1	1	0	0	0	0
1	1	1	1	0	1

P7.30 De tabellen hebben bij 101 een andere output, dus $\overline{(A + B)(A + C)} \neq (\bar{B} + \bar{C})\bar{A}$

Tabel 16: Waarheidstabel P7.29 links

A	B	C	$A+B$	$A+C$	$\overline{(A+B)(A+C)}$
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	1	1	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	1	0

Tabel 17: Waarheidstabel P7.29 rechts

A	B	C	$\bar{B} + \bar{C}$	$\bar{A}$	$(\bar{B} + \bar{C})\bar{A}$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	0	0	0

Tabel 18: Waarheidstabel P7.30 links

A	B	C	$AB+C$	$\overline{AB+C}$	$A+B$	$\overline{(AB+C)(A+B)}$
0	0	0	0	1	0	1
0	0	1	1	0	0	1
0	1	0	0	1	1	0
0	1	1	1	0	1	1
1	0	0	0	1	1	0
1	0	1	1	0	1	1
1	1	0	1	0	1	1
1	1	1	1	0	1	1

P7.31 Beide tabellen hebben dezelfde output voor alle combinaties van ingangssignalen, dus $ABC + \bar{A}BC + A\bar{B}C = (A+B)C$

Tabel 19: Waarheidstabel P7.30 rechts

A	B	C	AB	$\bar{A}C$	$\bar{A}\bar{B}$	$AB + \bar{A}C + \bar{A}\bar{B}$
0	0	0	0	0	1	1
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	1	0	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	1
1	1	1	1	0	0	1

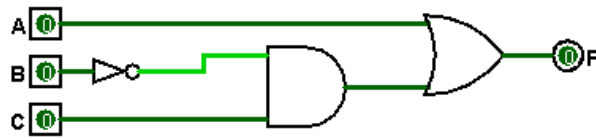
Tabel 20: Waarheidstabel P7.31 links

A	B	C	ABC	$\bar{A}BC$	$A\bar{B}C$	$ABC + \bar{A}BC + A\bar{B}C$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	1
1	0	0	0	0	0	0
1	0	1	0	0	1	1
1	1	0	0	0	0	0
1	1	1	1	0	0	1

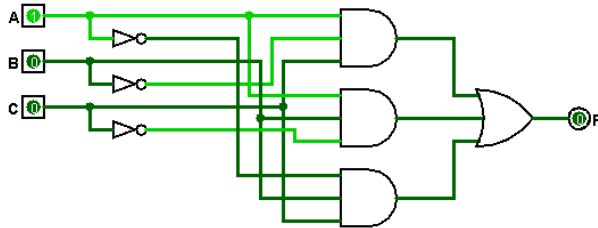
Tabel 21: Waarheidstabel P7.31 links

A	B	C	A+B	$(A+B)C$
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

P7.32 A Zie figuur 1.**B** Zie figuur 2.

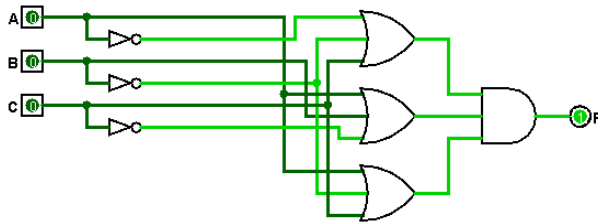


Figuur 1: P7.32 a



Figuur 2: P7.32 b

C Zie figuur 3.



Figuur 3: P7.32 c

P7.33 A $F = AB + (\bar{C} + A)\bar{D} = \overline{(\bar{A} + \bar{B})(C\bar{A} + D)}$

B $F = A(\bar{B} + C) + D = \overline{(\bar{A} + B\bar{C})\bar{D}}$

C $F = A\bar{B}C + A(B + C) = \overline{(\bar{A} + B + \bar{C})(\bar{A} + \bar{B}\bar{C})}$

D $F = (A + B + C)(A + \bar{B} + C)(\bar{A} + B + \bar{C}) = \overline{\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}C}$

E $F = ABC + A\bar{B}C + \bar{A}B\bar{C} = \overline{(\bar{A} + \bar{B} + \bar{C})(\bar{A} + B + \bar{C})(A + \bar{B} + C)}$

P7.34 NAND gates zijn voldoende voor combinatorische logica, omdat iedere booleaanse vergelijking gemaakt kan worden met alleen maar NANDs. Dit geldt ook voor NOR gates, ook deze zijn voldoende voor het kunnen maken van combinatorische logica.

P7.35 $E = CD(A + \bar{B})$.

P7.36 $E = \overline{A(\bar{B} + CD)}$.

P7.37 $C = \overline{\bar{A} + \bar{B}} = AB$. Dit is dus hetzelfde als een AND poort.

Tabel 22: Waarheidstabel P7.35

A	B	CD	E
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Tabel 23: Waarheidstabel P7.36

A	B	CD	E
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

P7.38 $F = \overline{\overline{D}E} = D + E$. Dit is dus hetzelfde als een OR poort.

7.4 Synthesis of Logic Circuits

P7.38 Bij het maken van een logische uitdrukking in de vorm van een som van producten (SOP), worden de regels gemaakt die in de waarheidstabel een 1 opleveren. Een product van logische variabelen komt dan overeen met die regel in de waarheidstabel. Vervolgens worden alle producten bij elkaar gevoegd met een OR.

Bij het maken van een logische uitdrukking in de vorm van een product van sommen (POS) (geen tentamenstof!), worden de regels gemaakt die in de waarheidstabel een 0 opleveren. Een som van logische variabelen komt dan overeen met die regel in de waarheidstabel. Vervolgens worden alle sommen bij elkaar gevoegd met een AND.

P7.39 Een goed voorbeeld van een decoder is een CD4511 BCD to 7-segment decoder. Met een 4-bit BCD woord worden de 7 segments van een 7-segment decoder aangestuurd. Een ander voorbeeld is de 74HCT138 3-naar-8 decoder. Met een 3-bit binair getal wordt een van de 8 outputs laag gemaakt.

P7.40 We gaan uit van alleen de Som-of-Products (SOP).
Dan is $F = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC\bar{C} = \Sigma m(0,2,4,6)$.

P7.41 We gaan uit van alleen de Som-of-Products (SOP).
Dan is $G = \bar{A}BC + ABC = \Sigma m(3,7)$.

P7.42 We gaan uit van alleen de Som-of-Products (SOP).
Dan is $H = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC = \Sigma m(0,1,2,3)$.

P7.43 We gaan uit van alleen de Som-of-Products (SOP).
Dan is $I = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C = \Sigma m(2,3,4,5)$.

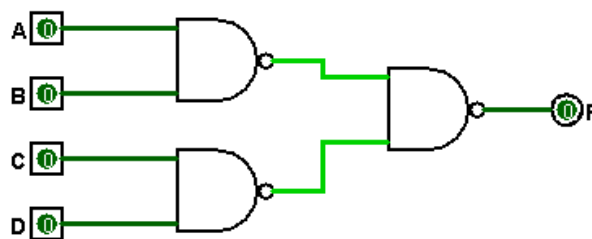
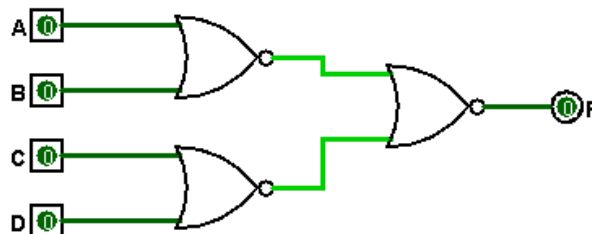
P7.44 We gaan uit van alleen de Som-of-Products (SOP).
Dan is $J = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC = \Sigma m(0,1,2,3,4,5,6,7)$.

P7.45 We gaan uit van alleen de Som-of-Products (SOP).
Dan is $K = \bar{A}B\bar{C} + \bar{A}BC + ABC = \Sigma m(2,3,7)$.

P7.46 $F = AB + CD = \overline{\bar{A}\bar{B}\bar{C}\bar{D}}$. De termen met AB en CD maak je met NANDs, het resultaat gaat ook weer in een NAND.
Zie figuur 4.

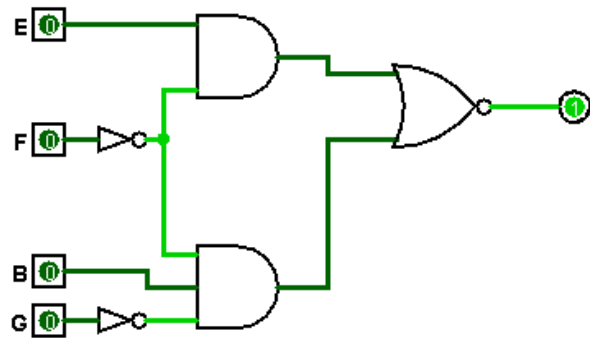
P7.47 $F = (A+B)(C+D) = \overline{\overline{(A+B)(C+D)}}$. De termen met A+B en C+D maak je met NORs, het resultaat gaat ook weer in een NOR.
Zie figuur 5.

P7.48 A

**Figuur 4:** P7.46**Figuur 5:** P7.47**Tabel 24:** Waarheidstabel P7.48 b: $E = \overline{A \cdot B + C \cdot D + A \cdot C}$

B	E	F	G	I
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

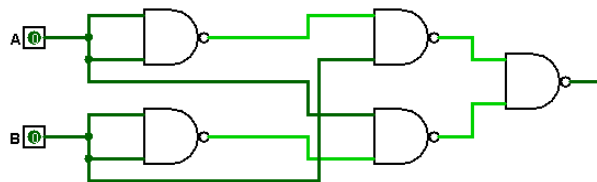
B $I = \Sigma m(4,5,8,12,13)$.**C** Geen tentamenstof**D** $I = E\bar{F} + B\bar{F}\bar{G}$. Zie figuur 6.

**Figuur 6:** P7.48

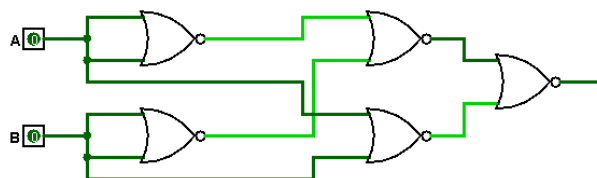
P7.49 Een logische formule voor een exclusive-OR is: $Out = A\bar{B} + \bar{A}B = \overline{\overline{A\bar{B}} \cdot \overline{\bar{A}B}}$. Zie figuur 7.

Tabel 25: P7.49 Waarheidstabel XOR poort

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

**Figuur 7:** P7.49

P7.50 Een logische formule voor een exclusive-OR is: $Out = A\bar{B} + \bar{A}B = \overline{\overline{A\bar{B}} \cdot \overline{\bar{A}B}} = \overline{(\bar{A} + B)(A + \bar{B})} = \overline{\bar{A} + B} + \overline{A + \bar{B}}$. Zie figuur 8.

**Figuur 8:** P7.50

P7.51 Geen tentamenstof

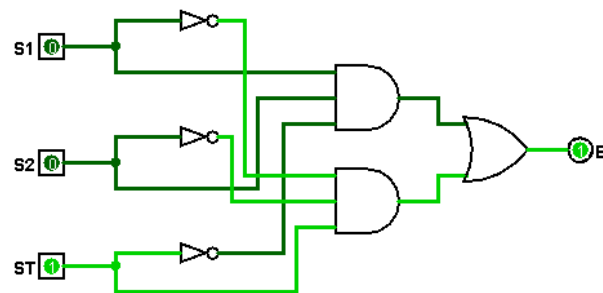
P7.52 A

Tabel 26: Waarheidstabel P7.52

S_1	S_2	S_T	E
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	0	0
1	1	0	1
1	1	1	0

B $E = \Sigma m(1,6) = \bar{S}_1 \bar{S}_2 S_T + S_1 S_2 \bar{S}_T.$

C Zie figuur 9.

**Figuur 9:** P7.52

7.5 Minimization of Logic Circuits

P7.53 A Zie Karnaugh-diagram

		<i>CD</i>			
		00	01	11	10
<i>AB</i>	00	0	0	0	1
	01	1	1	0	1
	11	1	1	0	0
	10	0	0	0	0

B $F = B\bar{C} + \bar{A}C\bar{D}$.

C POS is geen tentamenstof

P7.54 A Zie Karnaugh-diagram

		<i>BC</i>			
		00	01	11	10
<i>A</i>	0	0	1	0	0
	1	0	1	0	1

B $F = \bar{B}C + AB\bar{C}$.

C POS is geen tentamenstof

P7.55 Door als SOP te schrijven wordt dit: $D = \Sigma m(0,2,4,6)$.

A Zie Karnaugh-diagram

		<i>BC</i>			
		00	01	11	10
<i>A</i>	0	1	0	0	1
	1	1	0	0	1

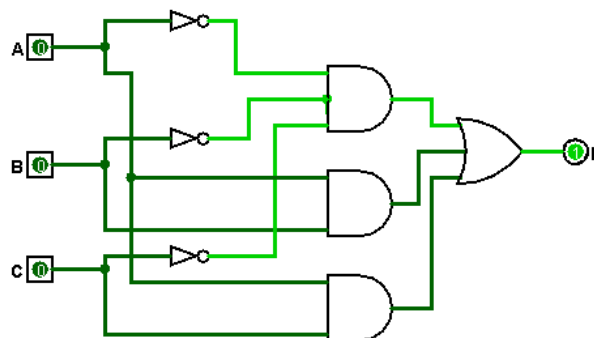
B $F = \bar{C}$.

C POS is geen tentamenstof

P7.56 A Zie Karnaugh-diagram en figuur 10.

		<i>BC</i>			
		00	01	11	10
<i>A</i>	0	1	0	0	0
	1	0	1	1	1

B $F = \bar{A}\bar{B}\bar{C} + AB + AC$.



Figuur 10: P7.56

C POS is geen tentamenstof

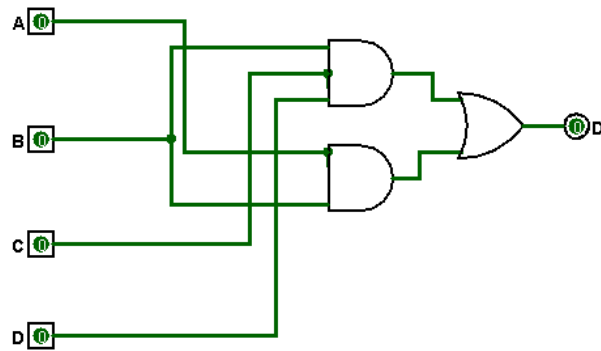
P7.57 A Zie Karnaugh-diagram

		<i>CD</i>			
		00	01	11	10
<i>AB</i>	00	0	0	0	0
	01	0	0	1	0
	11	1	1	1	1
	10	0	0	0	0

B $F = AB + BCD$.

C zie figuur 11

D POS is geen tentamenstof



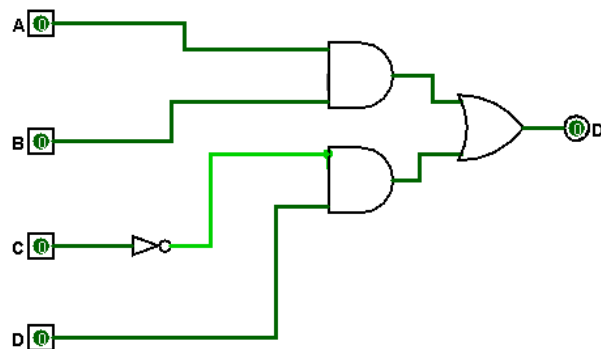
Figuur 11: P7.57

P7.58 A Zie Karnaugh-diagram voor F

		CD			
		00	01	11	10
AB	00	0	1	0	0
	01	0	1	0	0
	11	1	1	1	1
	10	0	1	0	0

B $F = AB + \bar{C}D$.

C zie figuur 12

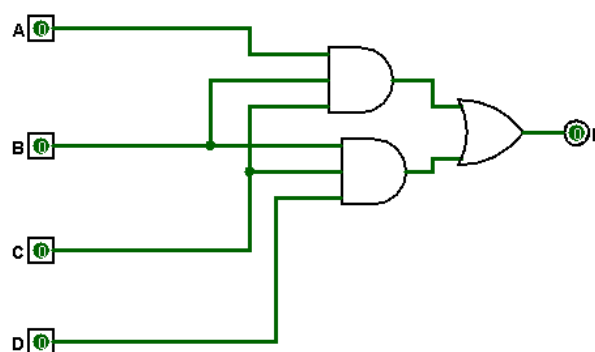


Figuur 12: P7.58

D POS is geen tentamenstof

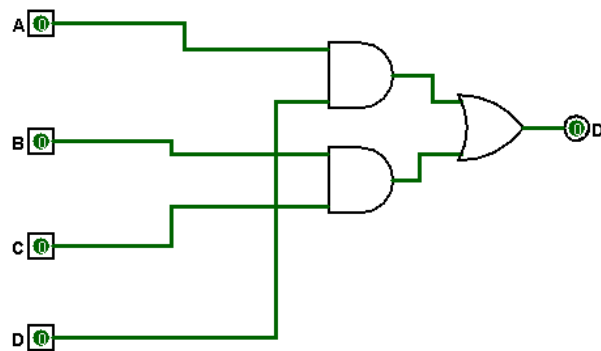
P7.59 A Zie Karnaugh-diagram voor G

		CD			
		00	01	11	10
AB	00	0	0	0	0
	01	0	0	1	0
	11	0	0	1	1
	10	0	0	0	0

B $G = ABC + BCD$.**C** zie figuur 13**Figuur 13:** P7.59**D** POS is geen tentamenstof**P7.60 A** Zie Karnaugh-diagram voor H

		CD			
		00	01	11	10
AB	00	0	0	0	0
	01	0	0	1	1
	11	0	1	1	1
	10	0	1	1	0

B $G = AD + BC$.**C** zie figuur 14



Figuur 14: P7.60

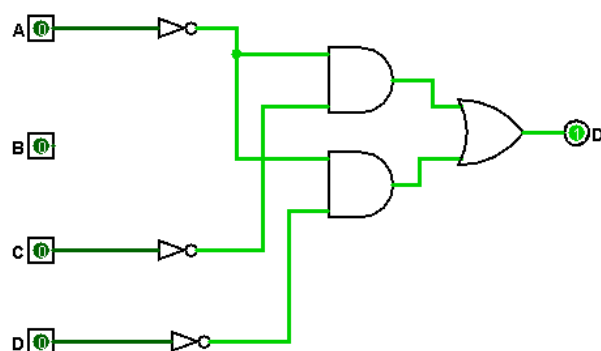
D POS is geen tentamenstof

P7.61 A Zie Karnaugh-diagram voor I

		CD			
		00	01	11	10
AB	00	1	1	0	1
	01	1	1	0	1
	11	0	0	0	0
	10	0	0	0	0

B $G = \bar{A}\bar{C} + \bar{A}\bar{D}$.

C zie figuur 15



Figuur 15: P7.61

D POS is geen tentamenstof

P7.62 $X = \Sigma m(0,2,4,6)$. Uit Karnaugh-diagram volgt dat $X = \bar{A}\bar{D}$.

Tabel 27: Waarheidstabel bij P7.62

B_8	B_4	B_2	B_1	X
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Karnaugh-diagram voor X:

		CD			
		00	01	11	10
AB	00	1	0	0	1
	01	1	0	0	1
	11	0	0	0	0
	10	0	0	0	0

P7.63 Een BCD code kent alleen de getallen 0 tot en met 9. Ieder getal groter dan 9 moet dan ook een 1 opleveren voor X, oftewel $X = \Sigma m(10,11,12,13,14,15)$. Uit Karnaugh-

diagram volgt dat $X = AB + AC$.

		CD			
		00	01	11	10
AB	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	0	0	1	1

P7.64 $X = \Sigma m(4,6,12,14)$. Uit Karnaugh-diagram volgt dat $X = B\bar{D}$.

		CD			
		00	01	11	10
AB	00	0	0	0	0
	01	1	0	0	1
	11	1	0	0	1
	10	0	0	0	0

P7.65 Eerst de waarheidstabel opstellen:

Tabel 28: Waarheidstabel P7.65

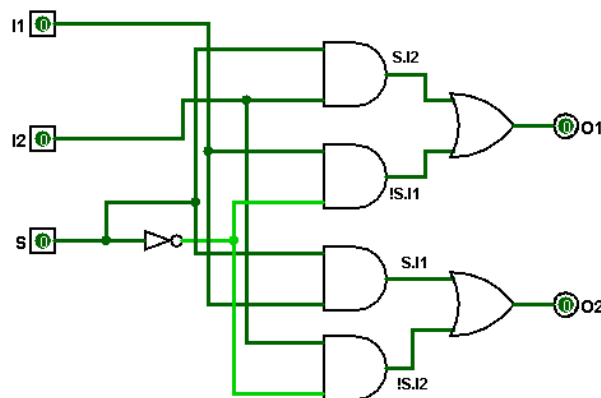
S	I_1	I_2	O_1	O_2
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

		$I_1 I_2$			
		00	01	11	10
S	0	0	0	1	1
	1	0	1	1	0

Karnaugh-diagram voor O_1 : $O_1 = SI_2 + \bar{S}I_1$.

		$I_1 I_2$			
		00	01	11	10
S	0	0	1	1	0
	1	0	0	1	1

Karnaugh-diagram voor O_2 : $O_2 = SI_1 + \bar{S}I_2$.



Figuur 16: P7.65

P7.66 De Karnaugh-map voor een raad van 3 leden is als volgt:

		BC			
		00	01	11	10
A	0	0	0	1	0
	1	0	1	1	1

Er zijn dus 3 groepen van 2 nodig, hetgeen je ook zou verwachten. De vergelijking wordt nu: $X = AB + AC + BC$.

Voor een raad van 5 personen zijn er dus minimaal 3 nodig voor een meerderheid. In een Karnaugh-diagram ziet dat er als volgt uit:

		<i>DE</i>			
		00	01	11	10
<i>BC</i>	00	0	0	0	0
	01	0	0	1	0
	11	0	1	1	1
	10	0	0	1	0
		<i>A = 0</i>			

		<i>DE</i>			
		00	01	11	10
<i>BC</i>	00	0	0	1	0
	01	0	1	1	1
	11	1	1	1	1
	10	0	1	1	1
		<i>A = 1</i>			

Je kunt deze 2 Karnaugh diagrammen boven elkaar denken. En hiermee kunnen we iedere keer groepjes van 4 maken. Om de linker Karnaugh af te dekken, hebben we 4 groepen van 4 nodig. Deze zijn getekend. Om in de rechter Karnaugh alles af te dekken met groepjes van 4, hebben we nog 6 groepen van 4 nodig. De totale vergelijking wordt dan: $X = ABC + ABD + ABE + ACD + ACE + ADE + BCD + BCE + BDE + CDE$.

P7.67 Met de 'vote is tied' wordt bedoeld dat de stemmen staken. Dat is het geval indien 2 van de 4 leden voor stemmen en de andere 2 tegen. In een Karnaugh diagram ziet dat er als volgt uit:

		<i>CD</i>			
		00	01	11	10
<i>AB</i>	00	0	0	1	0
	01	0	1	0	1
	11	1	0	0	0
	10	0	1	0	1

Er zijn geen groepjes te maken, dus de uitdrukking wordt: $X = \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + AB\bar{C}\bar{D}$.

P7.68 A Eerst de waarheidstabel opstellen: hieruit volgt dat het totaal aantal enen in een rij altijd even is.

Tabel 29: Waarheidstabel bij P7.68

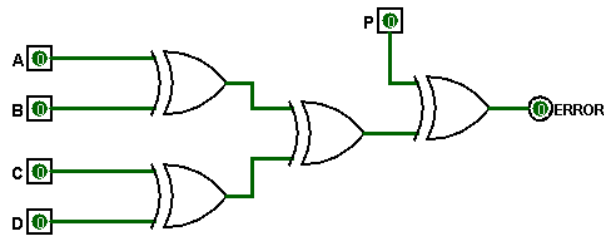
A	B	C	D	E
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

B Het Karnaugh diagram hiervoor ziet er als volgt uit:

		<i>CD</i>			
		00	01	11	10
<i>AB</i>	00	0	1	0	1
	01	1	0	1	0
	11	0	1	0	1
	10	1	0	1	0

Ook met dit Karnaugh diagram zijn geen groepen te maken, de uitdrukking wordt dan: $P = \Sigma m(1,2,4,7,8,11,13,14)$.

C Door de output van het huidige circuit te controleren met het P bit kan op fouten gecontroleerd worden. Anders gezegd: als het P-bit ongelijk is aan de output van het huidige circuit dan moet de output 1 worden. Ook dit is weer een XOR functie (bij een XOR wordt de output 1 als de ingangen ongelijk aan elkaar zijn), zie figuur 17.



Figuur 17: P7.68

P7.69 Van binaire code ABC naar Gray code XYZ gaat als volgt:

A Het Karnaugh diagram voor Gray code X ziet er als volgt uit:

		BC			
		00	01	11	10
A	0	0	0	0	0
	1	1	1	1	1

Er geldt dus dat: $X = A$.

B Het Karnaugh diagram voor Gray code Y ziet er als volgt uit:

		BC			
		00	01	11	10
A	0	0	0	1	1
	1	1	1	0	0

Er geldt dus dat: $X = A\bar{B} + \bar{A}B = A \oplus B$.

C Het Karnaugh diagram voor Gray code Z ziet er als volgt uit:

		BC			
		00	01	11	10
A	0	0	1	0	1
	1	0	1	0	1

Er geldt dus dat: $X = B\bar{C} + \bar{B}C = B \oplus C$.

P7.70 Van Gray code XYZ naar binaire code ABC gaat als volgt:

A Het Karnaugh diagram voor binaire code A ziet er als volgt uit:

		YZ			
		00	01	11	10
X	0	0	0	0	0
	1	1	1	1	1

Er geldt dus dat: $A = X$.

B Het Karnaugh diagram voor binaire code B ziet er als volgt uit:

		YZ			
		00	01	11	10
X	0	0	0	1	1
	1	1	1	0	0

Er geldt dus dat: $B = X\bar{Y} + \bar{X}Y = X \oplus Y$.

C Het Karnaugh diagram voor binaire code C ziet er als volgt uit:

		YZ			
		00	01	11	10
X	0	0	1	0	1
	1	1	0	1	0

Er kan geen optimalisatie gemaakt worden, dus $C = X\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + ABC + \bar{A}\bar{B}\bar{C}$.

P7.71 Omzetten BCD code naar 4221 code

A Het Karnaugh diagram voor 4221 code F ziet er als volgt uit:

		CD			
		00	01	11	10
AB	00	0	0	0	0
	01	1	0	1	1
	11	X	X	X	X
	10	1	1	X	X

Er geldt dus dat: $F = A + BC + B\bar{D} + BC$.

B Het Karnaugh diagram voor 4221 code G ziet er als volgt uit:

		CD			
		00	01	11	10
AB	00	0	0	0	0
	01	0	1	1	1
	11	X	X	X	X
	10	1	1	X	X

Er geldt dus dat: $G = A + BC + BD$.

C Het Karnaugh diagram voor 4221 code H ziet er als volgt uit:

		CD			
		00	01	11	10
AB	00	0	0	1	1
	01	0	1	0	0
	11	X	X	X	X
	10	1	1	X	X

Er geldt dus dat: $H = A + \bar{B}C + B\bar{C}D$.

D Het Karnaugh diagram voor 4221 code I ziet er als volgt uit:

		<i>CD</i>			
		00	01	11	10
<i>AB</i>	00	0	1	1	0
	01	0	1	1	0
	11	X	X	X	X
	10	0	1	X	X

Er geldt dus dat: $H = D$.

P7.72 Omzetten van 4221 code naar BCD code

A Het Karnaugh diagram voor BCD code A ziet er als volgt uit:

		<i>HI</i>			
		00	01	11	10
<i>FG</i>	00	0	0	0	0
	01	X	X	0	X
	11	0	0	1	1
	10	0	X	X	X

Er geldt dus dat: $A = FH$.

B Het Karnaugh diagram voor BCD code B ziet er als volgt uit:

		<i>HI</i>			
		00	01	11	10
<i>FG</i>	00	0	0	0	0
	01	X	X	1	X
	11	1	1	0	0
	10	1	X	X	X

Er geldt dus dat: $B = \bar{F}G + F\bar{H}$.

C Het Karnaugh diagram voor BCD code C ziet er als volgt uit:

		HI			
		00	01	11	10
FG	00	0	0	1	1
	01	X	X	0	X
	11	1	1	0	0
	10	0	X	X	X

Er geldt dus dat: $C = G\bar{H} + \bar{G}H = G \oplus H$.

D Het Karnaugh diagram voor BCD code D ziet er als volgt uit:

		HI			
		00	01	11	10
FG	00	0	1	1	0
	01	X	X	1	X
	11	0	1	1	0
	10	0	X	X	X

Er geldt dus dat: $D = I$.

P7.73 Omzetten van BCD code naar Excess-3 code

A Het Karnaugh diagram voor Excess-3 code W ziet er als volgt uit:

		CD			
		00	01	11	10
AB	00	0	0	0	0
	01	0	1	1	1
	11	X	X	X	X
	10	1	1	X	X

Er geldt dus dat: $W = A + BC + BD$.

B Het Karnaugh diagram voor Excess-3 code X ziet er als volgt uit:

		CD			
		00	01	11	10
AB	00	0	1	1	1
	01	1	0	0	0
	11	X	X	X	X
	10	0	1	X	X

Er geldt dus dat: $X = \bar{B}C + \bar{B}D + B\bar{C}\bar{D}$.

C Het Karnaugh diagram voor Excess-3 code Y ziet er als volgt uit:

		CD			
		00	01	11	10
AB	00	1	0	1	0
	01	1	0	1	0
	11	X	X	X	X
	10	1	0	X	X

Er geldt dus dat: $Y = CD + \bar{C}\bar{D} = \bar{C} \oplus \bar{D}$.

D Het Karnaugh diagram voor Excess-3 code Z ziet er als volgt uit:

		CD			
		00	01	11	10
AB	00	1	0	0	1
	01	1	0	0	1
	11	X	X	X	X
	10	1	0	X	X

Er geldt dus dat: $Z = \bar{D}$.

P7.74 Omzetten van Excess-3 code naar BCD code

A Het Karnaugh diagram voor BCD code A ziet er als volgt uit:

		YZ			
		00	01	11	10
WX	00	X	X	0	X
	01	0	0	0	0
	11	1	X	X	X
	10	0	0	1	0

Er geldt dus dat: $A = WX + WZ$.

B Het Karnaugh diagram voor BCD code B ziet er als volgt uit:

		YZ			
		00	01	11	10
WX	00	X	X	0	X
	01	0	0	1	0
	11	0	X	X	X
	10	1	1	0	1

Er geldt dus dat: $B = \bar{X}\bar{Y} + XYZ + WY\bar{Z}$.

C Het Karnaugh diagram voor BCD code C ziet er als volgt uit:

		YZ			
		00	01	11	10
WX	00	X	X	0	X
	01	0	1	0	1
	11	0	X	X	X
	10	0	1	0	1

Er geldt dus dat: $C = \bar{Y}Z + Y\bar{Z} = X \oplus Y$.

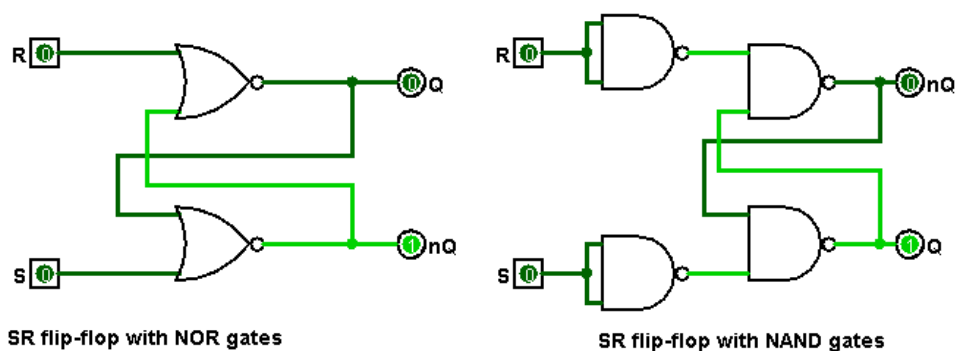
D Het Karnaugh diagram voor BCD code D ziet er als volgt uit:

		YZ			
		00	01	11	10
WX	00	X	X	0	X
	01	1	0	0	1
	11	1	X	X	X
	10	1	0	0	1

Er geldt dus dat: $D = \bar{Z}$.

7.6 Sequential Logic Circuits

P7.75 Een SR flip-flop met NOR gates en met NAND gates ziet er uit zoals weergegeven in figuur 18.

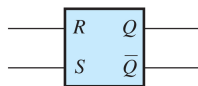


Figuur 18: P7.75 SR flip-flop met NOR en NAND gates

P7.76 Dit is figuur 7.39 in het boek, zie figuur 19.

R	S	Q_n
0	0	Q_{n-1}
0	1	1
1	0	0
1	1	Not allowed

(a) Truth table

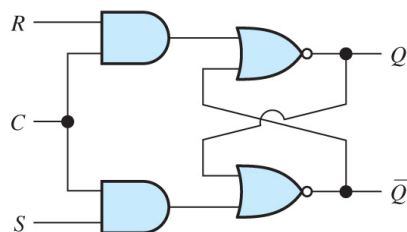


(b) Circuit symbol

Copyright © 2008 Pearson Prentice Hall, Inc.

Figuur 19: P7.76 Figuur 7.39 Hambley

P7.77 Dit is figuur 7.44 in het boek, zie figuur 20.

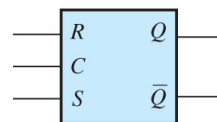


(a) Circuit diagram

R	S	C	Q_n
0	0	$\times$	Q_{n-1}
0	1	1	1
1	0	1	0
1	1	1	Not allowed
$\times$	$\times$	0	Q_{n-1}

(b) Truth table

Copyright © 2008 Pearson Prentice Hall, Inc.



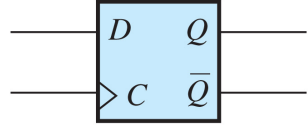
(b) Circuit symbol

Figuur 20: P7.77 Figuur 7.44 Hambley

P7.78 Bij een asynchrone flip-flop worden de ingangssignalen direct verwerkt, dus zijn veranderingen in uitgangssignalen onafhankelijk van een clock-sigitaal. Bij een synchrone flip-flop worden de ingangssignalen alleen verwerkt wanneer het clock-sigitaal hoog is.

P7.79 Bij edge-triggering worden signalen alleen verwerkt wanneer het clock-sigitaal van laag naar hoog (positive edge triggering) of van hoog naar laag (negative edge-triggering) verandert.

P7.80 Dit is figuur 7.47 in het boek, zie figuur 21.

		C	D	Q_n
		0	×	Q_{n-1}
		1	×	Q_{n-1}
		↑	0	0
		↑	1	1

(a) Circuit symbol

(b) Truth table
↑ indicates a transition from low to high

Copyright © 2008 Pearson Prentice Hall, Inc.

Figuur 21: P7.80 Figuur 7.47 Hambley

P7.81 De exclusive-or geeft een 1 op D_0 als Q_1 of Q_2 een 1 bevat. De waarheidstabel ziet er dan als volgt uit:

Dus na 7 clock-pulsen komt de begin-toestand weer terug.

Tabel 30: Waarheidstabel P7.81

Q_0	Q_1	Q_2
1	0	0
0	1	0
1	0	1
1	1	0
1	1	1
0	1	1
0	0	1
1	0	0

P7.82 A Wanneer de XOR gate vervangen wordt door een OR gate, ziet de waarheidstabel er als volgt uit:

Dus na 5 clock-pulsen wordt de toestand 111 bereikt en hier blijft de teller ook op staan. De begintoestand komt dus niet meer terug.

B Wanneer de XOR gate vervangen wordt door een AND gate, ziet de waarheidstabel er als volgt uit:

Tabel 31: Waarheidstabel P7.82a

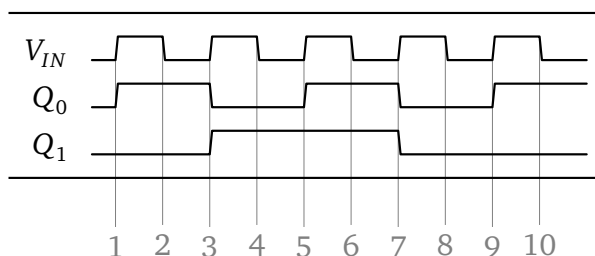
Q_0	Q_1	Q_2
1	0	0
0	1	0
1	0	1
1	1	0
1	1	1
1	1	1

Tabel 32: Waarheidstabel P7.82b

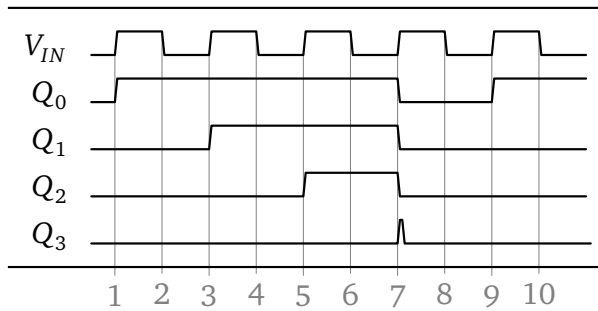
Q_0	Q_1	Q_2
1	0	0
0	1	0
0	0	1
0	0	0
0	0	0

Dus na 4 clock-pulsen wordt de toestand 000 bereikt en hier blijft de teller ook op staan. De begintoestand komt dus niet meer terug.

P7.83 De clock-ingang van flip-flop 1 wordt geclocked door de $\overline{Q_0}$ uitgang van flip-flop 0. In het timingsdiagram is dat terug te zien als een neergaande flank bij Q_0 (tijdstippen 3 en 7).



P7.84 De vier flip-flops hebben een asynchrone clear ingang, d.w.z. dat de uitgangen direct gereset (0) worden, dus onafhankelijk van de clock-ingang. Bij deze schakeling gebeurt dat wanneer Q_3 hoog wordt. Bij Q_3 zie je dus een korte puls, omdat na het hoog worden, ook deze uitgang weer direct gereset wordt.



P7.85 Om een JK flip-flop te maken van een (negative edge-triggered) D flip-flop, moet de D ingang een waarde krijgen, afhankelijk van de waarheidstabel van een JK flip-flop. Bij $J = K = 0$ (memory), moet D gelijk worden aan de Q uitgang (die waarde wordt dan bewaard). Bij $J=1$ en $K=0$ (set) moet de uitgang D een 1 worden, bij $J=0$ en $K=1$ (reset) moet de uitgang D een 0 worden. En bij $J=K=1$ moet de uitgang veranderd worden (toggle). In een waarheidstabel ziet dat er als volgt uit:

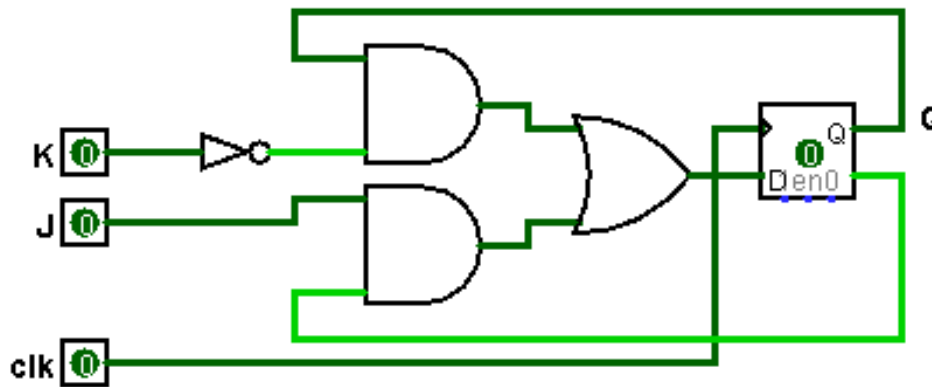
Tabel 33: Waarheidstabel P7.85

<i>Uitleg</i>	Q	J	K	D
memory	0	0	0	0
reset	0	0	1	0
set	0	1	0	1
toggle	0	1	1	1
memory	1	0	0	1
reset	1	0	1	0
set	1	1	0	1
toggle	1	1	1	0

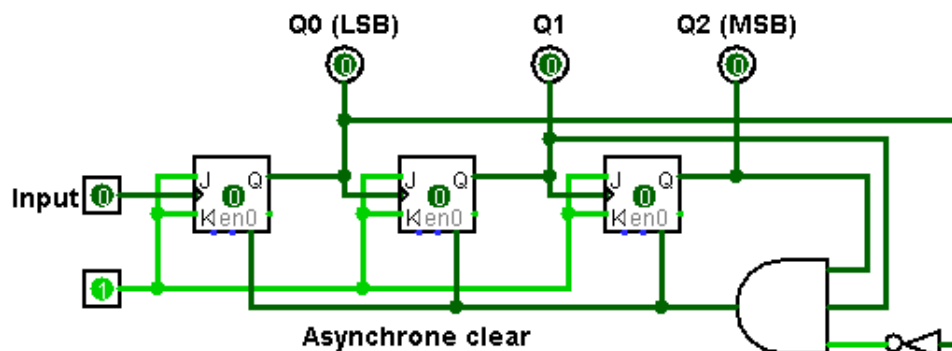
		<i>JK</i>			
		00	01	11	10
<i>Q</i>	0	0	0	1	1
	1	1	0	0	1

Er geldt dus dat: $D = \bar{Q}J + Q\bar{K}$. De schakeling hiervan is weergegeven in figuur 22.

P7.86 De teller gereset moet worden wanneer de tellerstand gelijk wordt aan 6, dit is binair 110. De schakeling hiervan is weergegeven in figuur 23.

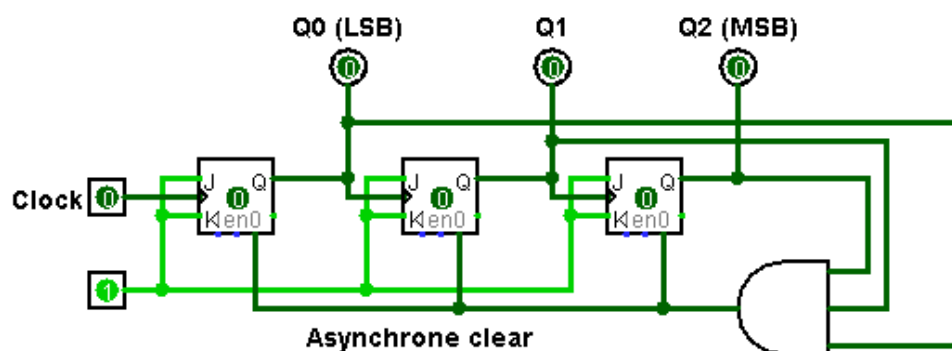


Figuur 22: P7.85 schakeling



Figuur 23: P7.86 teller schakeling

P7.87 A De schakeling voor deze dobbelsteen is een variant van de vorige opgave (P7.86), alleen moet het cijfer 6 nu ook nog getoond worden. De asynchrone reset moet nu dus pas gebeuren bij een 7 (111 binair). De schakeling hiervan is weergegeven in figuur 24.



Figuur 24: P7.87 teller schakeling

B De toestanden 0 en 7 geven voor alle LEDs een don't care (X), omdat deze toestanden niet kunnen voorkomen in de teller. De waarheidstabel voor de LEDs in de dobbelsteen ziet er als volgt uit:

De Karnaugh-diagrammen voor D_1 en D_7 zijn aan elkaar gelijk en zien er als volgt uit:

Tabel 34: Waarheidstabel P7.87

Q_3	Q_2	Q_1	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	X	X	X	X	X	X	X
0	0	1	0	0	0	1	0	0	0
0	1	0	1	0	0	0	0	0	1
0	1	1	1	0	0	1	0	0	1
1	0	0	1	1	0	0	0	1	1
1	0	1	1	1	0	1	0	1	1
1	1	0	1	1	1	0	1	1	1
1	1	1	X	X	X	X	X	X	X

		Q_2Q_1			
		00	01	11	10
Q_3	0	X	0	1	1
	1	1	1	X	1

Er geldt dus dat: $D_1 = D_7 = Q_2 + Q_3$.

De Karnaugh-diagrammen voor D_2 en voor D_6 zijn ook aan elkaar gelijk en zien er als volgt uit:

		Q_2Q_1			
		00	01	11	10
Q_3	0	X	0	0	0
	1	1	1	X	1

Er geldt dus dat: $D_2 = D_6 = Q_3$.

De Karnaugh-diagrammen voor D_3 en voor D_5 zijn ook aan elkaar gelijk en zien er als volgt uit:

		Q_2Q_1			
		00	01	11	10
Q_3	0	X	0	0	0
	1	0	0	X	1

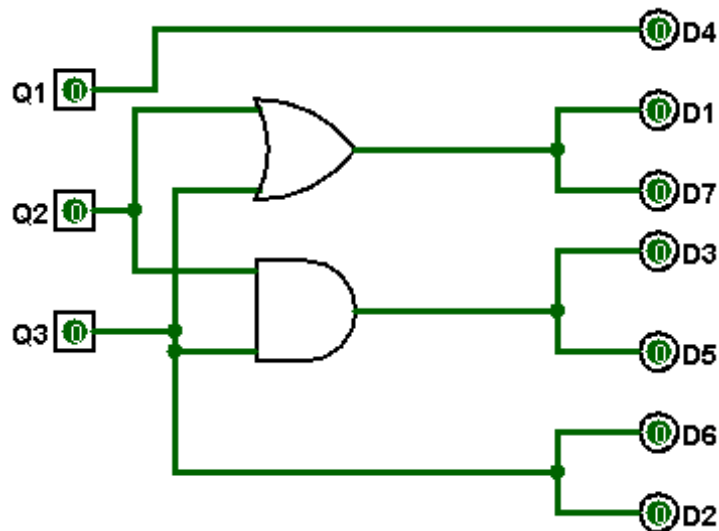
Er geldt dus dat: $D_3 = D_5 = Q_3Q_2$.

Het Karnaugh-diagram voor D_4 ziet er als volgt uit:

		Q_2Q_1			
		00	01	11	10
Q_3	0	X	1	1	0
	1	0	1	X	0

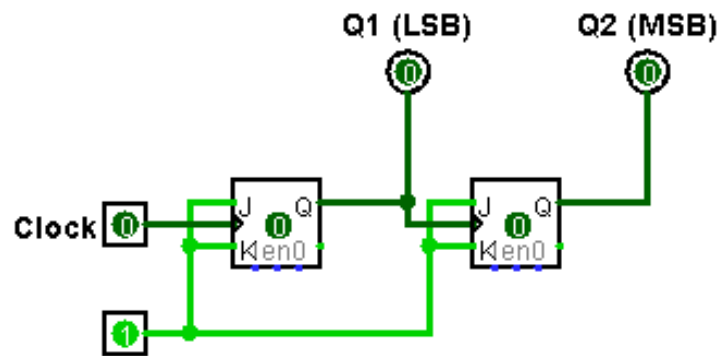
Er geldt dus dat: $D_4 = Q_1$.

De schakeling hiervan is weergegeven in figuur 25.



Figuur 25: P7.87 LED schakeling

- P7.88 A** Een complete omwenteling duurt 2 seconden. Dat betekent dat iedere LED 0,5 s aan moet staan. De clockfrequentie moet dus 2 Hz zijn.
- B** Een geschikte teller hiervoor is weergegeven in figuur 26.
- C** De waarheidstabel voor deze schakeling ziet er als volgt uit:
Het Karnaugh-diagram voor D_1 ziet er als volgt uit:

**Figuur 26:** P7.88 modulo-4 teller**Tabel 35:** Waarheidstabel P7.88

S	Q_2	Q_1	D_1	D_2	D_3	D_4
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	0	1
1	0	0	1	0	0	0
1	0	1	0	0	0	1
1	1	0	0	0	1	0
1	1	1	0	1	0	0

		Q_2Q_1			
		00	01	11	10
S	0	1	0	0	0
	1	1	0	0	0

Er geldt dus dat: $D_1 = \overline{Q_2Q_1}$.

Het Karnaugh-diagram voor D_2 ziet er als volgt uit:

		Q_2Q_1			
		00	01	11	10
S	0	0	1	0	0
	1	0	0	1	0

Geen vereenvoudiging mogelijk, dus: $D_2 = \overline{S}Q_2Q_1 + SQ_2Q_1$.

Het Karnaugh-diagram voor D_3 ziet er als volgt uit:

		Q_2Q_1			
		00	01	11	10
S	0	0	0	0	1
	1	0	0	0	1

Er geldt dus dat: $D_3 = Q_2\overline{Q_1}$.

Het Karnaugh-diagram voor D_4 ziet er als volgt uit:

		Q_2Q_1			
		00	01	11	10
S	0	0	0	1	0
	1	0	1	0	0

Ook hier weer geen vereenvoudiging mogelijk, dus: $D_4 = S\overline{Q_2}Q_1 + \overline{S}Q_2Q_1$.