



Inhoudsopgave

11	Digitale techniek	4
11.1	Wat leer je in dit hoofdstuk	4
11.2	Inleiding	4
11.2.1	Een stukje geschiedenis	4
11.2.2	Digitaal	5
11.2.3	Analoog en digitaal	5
11.2.4	Binair, decimaal en hexadecimaal	6
11.2.5	Omzetten van decimaal naar binair en omgekeerd	8
11.2.6	Rekenen met binaire getallen	9
11.3	Logische functies	11
11.3.1	Van 1 en 0 naar tegenstelling	11
11.3.2	Poorten en hun logische functies	12
11.3.3	De NIET- (NOT-)poort en de buffer	12
11.3.4	De EN- en de OF-poort	14
11.3.5	De NIET-EN- (NEN-) en de NIET-OF- (NOF-)poort	17
11.3.6	De Exclusieve OF- of EXOF-poort	19
11.3.7	De Exclusieve NOF (EXNOF, XNOF)-poort	20
11.3.8	De poorten samengevat	21
11.3.9	Transistor-transistor logica (TTL) (tot nu toe geen examenstof)	23
11.4	Booleaanse algebra (schakelalgebra)	24
11.4.1	Inleiding	24
11.4.2	Regels met 1 of 0	24
11.4.3	Regels met één variabele	24
11.4.4	Regels met twee of meer variabelen	25
11.4.5	Poortconversie met De Morgan	29
11.5	Combinatielogica (“combinational logic”)	30
11.5.1	Inleiding	30
11.5.2	Een EN- of OF -poort met n ingangen	30
11.5.3	EN en OF gecombineerd	31
11.5.4	Voorbeeld De Morgan met EN-poorten en OF-poort	31



11.5.5	Meer vereenvoudigingen	32
11.5.6	Meer voorbeelden van combinatielogica; “spelen” met poorten	33
11.5.7	Compacte tekenwijze	36
11.5.8	De optelschakeling	37
11.6	Opgaven.....	40
11.6.1	Opgave 11-1.....	41
11.6.2	Opgave 11-2.....	41
11.6.3	Opgave 11-3.....	41
11.6.4	Opgave 11-4.....	42
11.6.5	Opgave 11-5.....	42
11.6.6	Opgave 11-6.....	42
11.6.7	Opgave 11-7.....	43
11.6.8	Opgave 11-8.....	43
11.7	Sequentiële logica: flipflops.....	44
11.7.1	Inleiding.....	44
11.7.2	De flipflop en zijn verwanten	44
11.7.3	De RS-flipflop	47
11.7.4	De impulsgestuurde (geklokte) RS-flipflop.....	48
11.7.5	Het master-slave systeem.....	49
11.7.6	De flankgestuurde RS-flipflop	50
11.7.7	De D-flipflop.....	51
11.7.8	De JK-flipflop.....	52
11.7.9	De master-slave D-flipflop als frequentiedeler	54
11.7.10	Samengevat	55
11.8	Frequentiedelers, tellers en registers	55
11.8.1	Frequentiedelers en tellers.....	55
11.8.2	Delers en tellers voor andere getallen dan machten van 2	57
11.8.3	Schuifregisters.....	57
11.9	Opgaven.....	60
11.9.1	Opgave 11-9.....	60
11.9.2	Opgave 11-10.....	61
11.9.3	Opgave 11-11.....	61



11.9.4	Opgave 11-12.....	61
11.9.5	Opgave 11-13.....	62
11.10	Antwoorden bij de opgaven	63
11.10.1	Uitwerking van Opgave 11-1.....	63
11.10.2	Uitwerking van Opgave 11-2.....	64
11.10.3	Uitwerking van Opgave 11-3.....	65
11.10.4	Uitwerking van Opgave 11-4.....	66
11.10.5	Uitwerking van Opgave 11-5.....	67
11.10.6	Uitwerking van Opgave 11-6.....	68
11.10.7	Uitwerking van Opgave 11-7.....	69
11.10.8	Uitwerking van Opgave 11-8.....	70
11.10.9	Uitwerking van Opgave 11-9.....	71
11.10.10	Uitwerking van Opgave 11-10.....	72
11.10.11	Uitwerking van Opgave 11-11.....	73
11.10.12	Uitwerking van Opgave 11-12.....	74
11.10.13	Uitwerking van Opgave 11-13.....	75



11 Digitale techniek

11.1 Wat leer je in dit hoofdstuk

Tot nu toe ging het in deze cursus over analoge techniek. Tegenwoordig wordt analoge vaak gezien als het tegendeel van digitaal, maar eigenlijk is digitaal meer een manier om de analoge wereld waarin we leven, bij benadering te beschrijven. Daarom geven we een korte beschouwing over het verschil tussen beide begrippen.

Digitale techniek is gebaseerd op nullen en enen, dat wil zeggen het binaire (tweetallige) stelsel van getallen. Daarmee begint dit hoofdstuk dan ook. In Hoofdstuk 2 van deze cursus hebben we het al kort gehad over de opbouw van getallenstelsels. We gaan daarmee verder in dit hoofdstuk. We maken kennis met het omzetten van getallen uit het tweetallige, het tientallige en het zestientallige stelsel naar één van de andere stelsels in ditzelfde rijtje. Dat wordt gevolgd door een kort overzicht van binair rekenen.

In de daaropvolgende paragrafen duiken we in de logische (binaire) functies en poorten, schakelingen met poorten en hoe je kunt rekenen aan poortschakelingen met de zogenoemde Booleaanse of Boolese algebra die, ook wel *schakelalgebra* heet. We bespreken ook hoe je via verschillende schakelingen verschillende poorten kunt maken en we geven voorbeelden van het inwendige van poortschakelingen.

Dan volgt een aantal voorbeelden die laten zien hoe je verschillende combinaties van poorten hetzelfde kunt laten doen en hoe je ze in elkaar omrekent. Dat heet ook wel *combinatieloga*.

Een toepassing die daarop aansluit is een optelschakeling voor binaire getallen met zogenoemde *half adders* en *full adders*.

Deze poortentheorie of *combinatieloga* wordt gevolgd door een stuk zogenoemde *sequentiële logica*. Daarin wordt behalve van poorten gebruik gemaakt van opslag van informatie. Dat gaat eerst over zogenoemde flipflops. Die worden aaneengeschakeld in *registers*. Daarmee sluiten we dit hoofdstuk af. In volgende hoofdstukken komen toepassingen aan de orde waarin ze logisch passen.

11.2 Inleiding

11.2.1 Een stukje geschiedenis

De grondslag van digitale techniek is logica. Logica is terug te voeren tot Griekse wijsgeren, onder wie Aristoteles (384-322 v. Chr.) die de toenmalige wetenschappelijke kennis in een samenhangend systeem probeerde te vatten. Iets is waar of niet waar. De ene waarheid kan uit de andere volgen.

Het systeem van Aristoteles hield eeuwenlang stand, maar aan het eind van de middeleeuwen kwamen er barsten in. Die ontstonden doordat men beter ging waarnemen, ging experimenteren en daarvoor instrumenten ontwikkelde. Toen werd



langzaam duidelijk dat de natuurkunde van de wereld en het heelal anders in elkaar stak dan Aristoteles had beweerd. Dat betrof vooral zijn aannames. Zijn strikte en systematische denkwijze is daarna alleen maar verder uitgebouwd.

In 1847, zo'n 2200 jaar na Aristoteles, publiceerde de Britse wiskundige George Boole zijn *Mathematical Analysis of Logic*. Dat is de grondslag van de huidige logische analyse, soms *Booleaanse algebra* genoemd.

De grondlegger van de toepassing ervan in signaaltheorie en -verwerking is Claude E. Shannon, een Amerikaans wiskundige en elektronicus die in 1948 zijn "A Mathematical Theory of Communication" publiceerde. Dat was alweer een eeuw verder, maar daarna ging het hard.

11.2.2 Digitaal

Logische schakelingen worden ook "digitale" schakelingen genoemd. *Digitus* is Latijn voor vinger. Wie heeft zich niet de eerste beginselen van de rekenkunst eigen gemaakt door met de vingers te rekenen, ook al zei de juf of meester misschien dat het anders moest? Elke vinger is een eenheid. Drie vingers en vier vingers bijvoorbeeld. Vingers tellen en de uitkomst is 7 vingers. Eigenlijk zijn twee handen met elk vijf vingers samen de eenvoudigste rekenmachine die er is. Je hoeft alleen te kunnen tellen. Ons getallenstelsel voor dagelijks gebruik is niet voor niets gebaseerd op het getal 10. Aan dat stelsel is in hoofdstuk 2 al aandacht besteed.

Het tientallige stelsel is ongeschikt voor logische schakelingen. Een schakeling moet dan tien toestanden, bijvoorbeeld tien spanningen, hebben die foutloos van elkaar te onderscheiden zijn. Als een schakeling miljoenen bewerkingen moet maken, gaat dat gegarandeerd wel eens mis. Het eenvoudigste en veiligste is daarom: twee toestanden. Dus het tweetallige stelsel, 0 of 1.

11.2.3 Analooq en digitaal

De schakelingen die we tot nu toe in deze cursus zijn tegengekomen, vallen onder de term *analoqe schakelingen*. De grootheden daarin kunnen een oneindig aantal toestanden aannemen. Een stroom, hoe klein ook, kan een oneindig aantal verschillende waarden hebben. Een spanning, een weerstand, zelfinductie, capaciteit, noem maar op, ook. Al die grootheden of onderdelen worden *analoog* genoemd. De naam komt uit een bijna vergeten vakgebied in de elektronica: de *analoqe computer* en de *analoqe simulatie*. Dat vakgebied omvat het bouwen van schakelingen die voldoen aan dezelfde vergelijkingen als sommige natuurkundige verschijnselen, zoals warmtestroming, doorbuigen van bouw materiaal of stroming van grondwater. Die elektrische systemen gedroegen zich *analoog* aan de niet-elektrische die ze moesten nabootsen. Vandaar de naam.

Digitale schakelingen kennen geen waarde tussen 0 en 1. Om veel toestanden of getallen te kunnen maken, gebruikt men daarom gelijktijdig een groot aantal schakelingen die elk een 0 of een 1 kunnen maken. Maar ook dan blijft het aantal combinaties eindig, terwijl



analoge schakelingen een oneindig aantal toestanden kennen. Hoeveel je daarvan in werkelijkheid kunt onderscheiden, hangt af van de precisie van de meetapparatuur.

Iemand gaf ooit het verschil tussen analoog en digitaal kernachtig aan met de zin: “Op een analoge klok kun je zien hoe laat het is, op een digitale klok hoe laat het geweest is.”

11.2.4 Binair, decimaal en hexadecimaal

In heel veel apparaten voor allerlei toepassingen worden tegenwoordig digitale schakelingen gebruikt. Denk aan wasmachines, gereedschap, audio, video, auto's en dus niet alleen de PC. Die werken met binair gecodeerde informatie om de bediening te vereenvoudigen of mogelijk te maken.

Ook aan de zendhobby komt steeds meer digitale techniek te pas. Vandaar dat elementen van die techniek onder de exameneisen zijn gaan vallen. Bijvoorbeeld de binaire getallen, die de basis vormen van de vorm van te verwerken informatie. Hoewel die getallen in hoofdstuk 2 al zijn behandeld, geven we hier een kleine herhaling. Daarin passeren binair en hexadecimaal de revue, naast de vertrouwde decimale getallen,

Eén binair cijfer, 0 of 1, wordt vaak aangegeven met de term *bit*. Dat is een samentrekking van *binary digit*, binair cijfer. We kennen ook de *byte*. Daarmee worden tegenwoordig vrijwel altijd 8 bij elkaar horende bits bedoeld. Met 8 bits kun je $2^8=256$ verschillende combinaties maken. Die staan voor de getallen 0 tot en met 255. We zullen dat niet helemaal uitschrijven, want dat wordt een heel grote tabel. We doen het voor 4 bits en het eerste daaropvolgende getal van 5 bits. Met 4 bits kunnen we $2^4=16$ combinaties maken. Dat zijn de getallen 0 tot en met 15. Tabel 11.2-1 geeft het resultaat. De tabel laat zien dat de positie van een bit een waarde vertegenwoordigt. Of het bit zelf 1 of 0 is, bepaalt of de bijbehorende waarde wel of niet in het getal voorkomt.

Op de op één na onderste regel zijn alle bits 1. De uitkomst is dan 15. Voor hogere getallen moet er links een bit bij. Dat zien we in de onderste regel met het getal 16. Daarna herhaalt zich alles tot en met de waarde 31. Die staat niet meer in de tabel. Bij 32 worden alle vijf bits 0 en moet er links weer een nieuwe kolom bij voor de 1 die de waarde 32 voorstelt. Bij 64 ($=2^6$) herhaalt zich dat. De volgende “halte” is $2^7 = 128$ en zo gaat dat verder via 256, 512, enz.



Tabel 11.2-1. Binaire getallen van 4 bits en één van 5 bits

2 ⁴	2 ³ (8)	2 ² (4)	2 ¹ (2)	2 ⁰ (1)	Getalswaarde per bit en de som
					Getal
	0	0	0	0	0+0+0+0 = 0
	0	0	0	1	0+0+0+1 = 1
	0	0	1	0	0+0+2+0 = 2
	0	0	1	1	0+0+2+1 = 3
	0	1	0	0	0+4+0+0 = 4
	0	1	0	1	0+4+0+1 = 5
	0	1	1	0	0+4+2+0 = 6
	0	1	1	1	0+4+2+1 = 7
	1	0	0	0	8+0+0+0 = 8
	1	0	0	1	8+0+0+1 = 9
	1	0	1	0	8+0+2+0 = 10
	1	0	1	1	8+0+2+1 = 11
	1	1	0	0	8+4+0+0 = 12
	1	1	0	1	8+4+0+1 = 13
	1	1	1	0	8+4+2+0 = 14
	1	1	1	1	8+4+2+1 = 15
1	0	0	0	0	16+0+0+0+0 = 16

Dit laat ook zien dat voor n kolommen er 2^n combinaties mogelijk zijn. Daarin is het hoogste getal 2^n-1 en niet 2^n , omdat we bij 0 en niet bij 1 beginnen met tellen. In 4 kolommen passen 2^4 combinaties, in 5 kolommen 2^5 combinaties en in n kolommen 2^n combinaties. Voorbeeld: $2^{10} = 1024$. Voor 0 t/m 1023 heb je daarom 10 bits nodig. Bezwaar van binaire getallen is dat ze erg lang worden. Voor ons mensen is daarom het tientallige stelsel veel geschikter, maar een apparaat breng je met die lange binaire getallen niet in de war.

Merk op dat bij even getallen het meest rechtse bit 0 is en bij oneven getallen 1. Met andere woorden: delen door twee geeft bij even getallen een rest van 0 en bij oneven getallen een rest van 1.

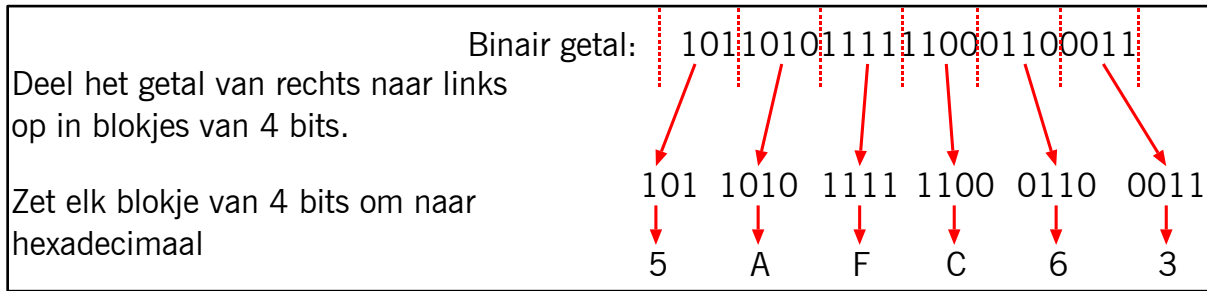
Die kennis is van belang als je een decimaal getal moet omzetten in een binair getal. In 11.2.5 zullen we dat toepassen.

Een manier om binaire getallen voor de mens beter hanteerbaar te maken, is het hexadecimale stelsel. Het basisgetal is daarbij 16. Het kent de cijfers 0-9, aangevuld met de letters A-F. Ze staan voor de waarden 0 t/m 15. Dat zijn 16 cijfers. Omdat 16 een gehele macht van 2 is (2^4), worden in één hexadecimaal cijfer de 4 bits volledig gebruikt (Tabel 11.2-2).

Tabel 11.2-2. De getallen 0 t/m 17 in decimaal, binair en hexadecimaal.

Decim.	0	1	2	3	4	5	6	7	8
Binair	0000	0001	0010	0011	0100	0101	0110	0111	1000
Hex	0	1	2	3	4	5	6	7	8
Decim.	9	10	11	12	13	14	15	16	17
Binair	1001	1010	1011	1100	1101	1110	1111	1 0000	1 0001
Hex	9	A	B	C	D	E	F	10	11

Voorbeeld: het binaire getal 1011010111110001100011 wordt omgezet naar hexadecimaal volgens Figuur 11.2-1 hieronder.



Figuur 11.2-1. Omzetting van het binaire getal 10110101111110001100011 naar hexadecimaal. Bewerkingen zijn in **rood** aangegeven.

Figuur 11.2-1 laat zien dat het binaire getal 10110101111110001100011 in hexadecimaal of kortweg *hex* 5AFC63 is. Hexadecimale notatie wordt bijvoorbeeld gebruikt als de inhoud van een digitaal geheugen wordt uitgelezen.

11.2.5 Omzetten van decimaal naar binair en omgekeerd

Beide omzettingen zijn beknopt aan de orde geweest in Hoofdstuk 2. We doen het hier wat uitvoeriger. De werkwijze voor omzetting van decimaal naar binair gaat voor elk willekeurig decimaal getal als volgt.

- Stap 1. Is het meest rechtse decimale cijfer oneven of even? Is het oneven, dan is het meest rechtse binaire cijfer een 1, zoals in Tabel 11.2-1 te zien is. Is het even, dan is het meest rechtse binaire cijfer een 0. Dat binaire cijfer wordt wel aangeduid met de afkorting “LSB”. Dat staat voor “Least Significant Bit”, het minst betekende bit.
- Stap 2. Trek het gevonden binaire cijfer (dus 0 of 1) af van het oorspronkelijke decimale. Deel de uitkomst door 2. Is het meest rechtse cijfer van het gevonden decimale getal oneven of even? Anders gezegd: levert delen door 2 een rest van 1 of van 0? Is de rest 1, dan komt links van het laatst gevonden binaire cijfer een 1; is het even, dan komt daar een 0.
- Stap 3. Herhaal stap 2 tot de hele uitkomst 0 is, dus zonder rest. Het laatst ingevulde binaire cijfer wordt wel aangeduid met “MSB”, “Most Significant Bit”.

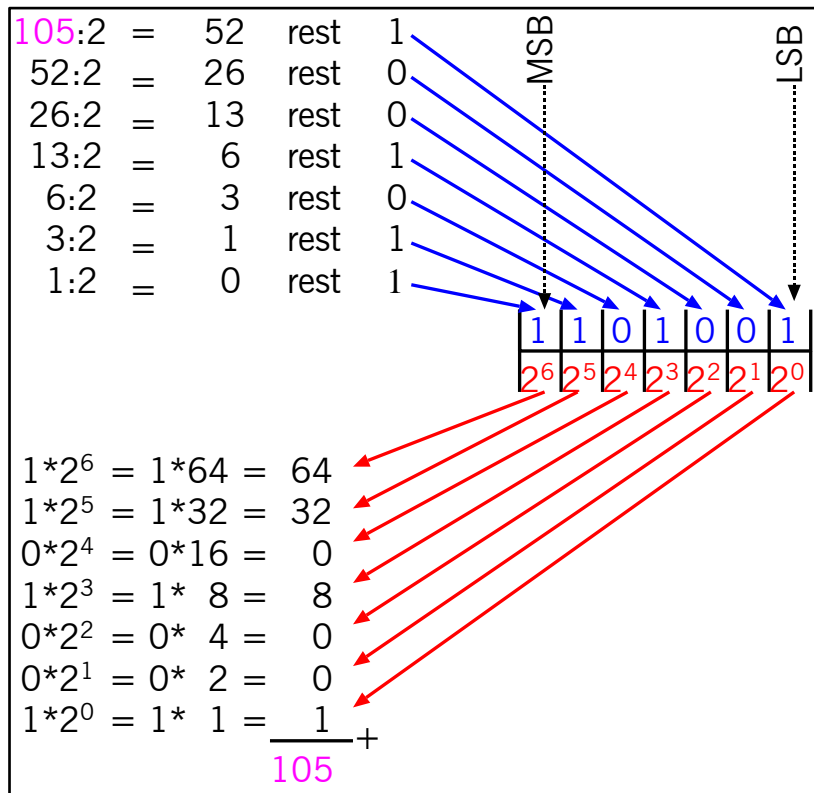
LSB en MSB worden ook wel geschreven als LSD en MSD, waarbij de D staat voor “Digit”.

Een klein voorbeeld: het getal 5. Je werkt altijd van rechts (LSB) naar links (MSB)

- 5 is oneven, dus we beginnen met 1. 1 eraf is 4. Delen door 2. $4:2 = 2$.
- 2 is even, dus er komt links een 0 bij: 01 dus. 0 eraf, blijft 2. Delen door 2. $2:2 = 1$.
- Oneven, dus er komt links een 1 bij: 101. 1 van de uitkomst af is 0, klaar.
- 5 decimaal is dus 101 binair.

Nu de omgekeerde bewerking. 101 binair is $1*2^2 + 0*2^1 + 1*2^0 = 4+0+1=5$. Onze decimale 5 is weer terug.

Een voorbeeld met het grotere getal 105 is uitgewerkt in Figuur 11.2-2.

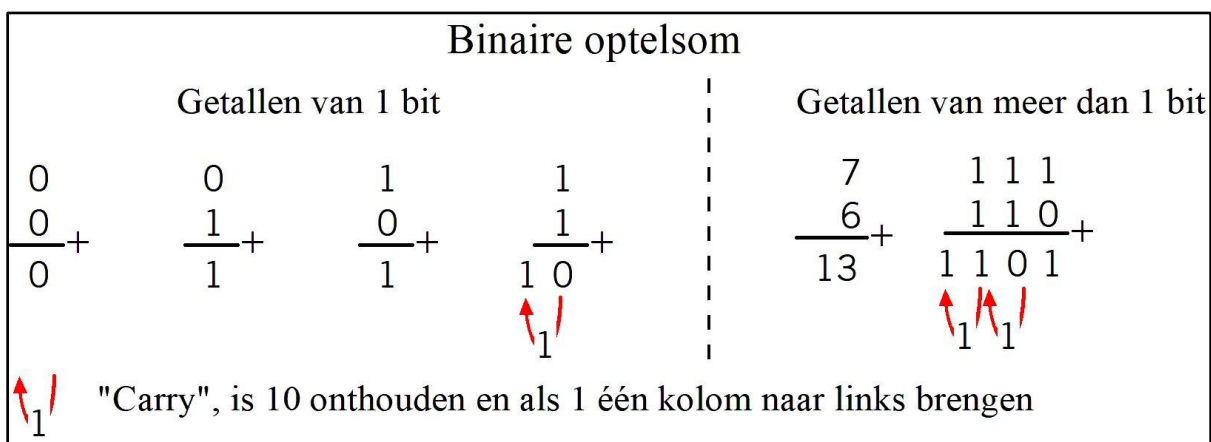


Figuur 11.2-2. Omzetting van het decimale getal 105 naar het binaire getal 1101001 met dezelfde waarde en daarna weer terug. MSB staat voor "Most Significant Bit"; LSB voor "Least Significant Bit". Blauwe pijlen: decimaal naar binair; rode pijlen: binair naar decimaal.

11.2.6 Rekenen met binaire getallen

De rekenregels voor binaire getallen zijn dezelfde als die voor decimale. Bij binaire getallen is er alleen door de lengte van de getallen meer schuifwerk. We zien hieronder voorbeelden voor optellen, aftrekken, vermenigvuldigen en delen.

Bij binair optellen is $0+0 = 0$, $0+1 = 1$, $1+0 = 1$ en $1+1 = 10$ (Figuur 11.2-3).



Figuur 11.2-3. Binair optellen. Links de basisbewerkingen met enkelvoudige bits; rechts met meerdere bits.

Bij binair aftrekken is $1-0 = 1$, $1-1 = 0$ en $10-1 = 1$

Binaire aftreksom								
Getallen van 1 bit					Getallen van meer dan 1 bit			
0	1	1	1		12	1	1	1
0	0	1	1		5	1	1	0
0	1	0	1		7	0	1	1
$\begin{array}{r} 1 \\ \downarrow \\ 10 \end{array}$					$\begin{array}{r} 1 & 1 & 1 \\ \downarrow & \downarrow & \downarrow \\ 1 & 1 & 0 & 0 \end{array}$			
$\begin{array}{r} 1 \\ \downarrow \\ 1 \end{array}$					"Borrow": 1 lenen en als 10 één kolom naar rechts brengen.			

Figuur 11.2-4. Binair aftrekken. Links de basisbewerkingen met enkelvoudige bits; rechts met meerdere bits.

Binair vermenigvuldigen is schuiven naar links. Om verwarring met de letter "x" te vermijden, gebruiken we in de tekst een ster * als vermenigvuldigteken. *1 is: er gebeurt niets. *10 is: 1 kolom naar links. *100 is: 2 kolommen naar links, enz. In Figuur 11.2-5 gebruiken we wel een x omdat daar geen misverstand mogelijk is.

Binair delen is schuiven naar rechts. Bij delen door 1 gebeurt niets; bij delen door 10 (decimaal is dat 2):1 positie naar rechts. Bij delen door 100 (decimaal is dat 4) → 2 posities naar rechts, enz.

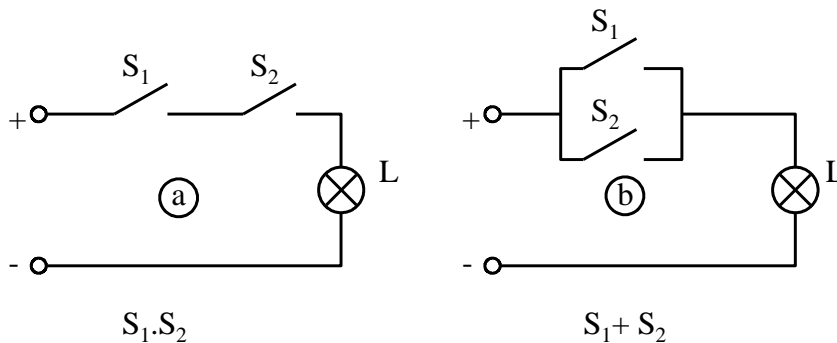
Binair vermenigvuldigen								
Vermenigvuldigen met 0, 2 ⁰ , 2 ¹ en 2 ²					Met iets ingewikkelder getallen			
1 1	1 1	1 1	1 1		5	1 0 1	1 0 1	1 0 1
0	1	1 0	1 0 0		3	1 1	1 0 1	1 0 1 0
0	1 1	1 1 0	1 1 0 0		15	1 0 1	1 1 1 1	1 1 1 1
$\begin{array}{r} 1 & 1 \\ \times & 1 \\ \hline 0 & 1 & 1 \end{array}$					$\begin{array}{r} 5 \\ \times 3 \\ \hline 15 \end{array}$			
$\begin{array}{r} 1 & 1 \\ \times 1 & 0 \\ \hline 1 & 1 & 0 \end{array}$					$\begin{array}{r} 1 & 0 & 1 \\ \times 1 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{array}$			
$\begin{array}{r} 1 & 1 & 0 \\ \times 1 & 0 \\ \hline 1 & 1 & 0 \end{array}$					Binair delen			
1 1	1 1 0	1 1 0 0	1 1 0 0		$11 / 1111 \setminus 101$			
1	1 0	1 0 0	1 0 0		$\begin{array}{r} 1 & 1 \\ \downarrow & \downarrow \\ 0 & 1 & 1 \\ \hline & 1 & 1 \\ & \hline & 0 \end{array}$			
1	1 1	1 1 0 0	1 1 0 0		$\begin{array}{r} 1 & 1 \\ \times 1 & 1 \\ \hline 1 & 1 \end{array}$			
1	1 1	1 1 0 0	1 1 0 0		$\begin{array}{r} 1 & 1 \\ \times 1 & 1 \\ \hline 1 & 1 \end{array}$			

Figuur 11.2-5. Binair vermenigvuldigen en delen. Vermenigvuldigen is schuiven naar links, delen naar rechts. Zelfs een klassieke staartdeling werkt.

11.3 Logische functies

11.3.1 Van 1 en 0 naar tegenstelling

We laten de getalsbetekenis van 0 en 1 voorlopig achter ons. We vervangen hem door tegenstellingen. Waar of niet waar; aan of uit; hoog of laag; ja of nee. Alles kan, als het maar een tegenstelling is. 0 en 1 (ver)worden van cijfers tot aanduidingen die vaak met *waar of niet waar* worden aangeduid. Het één of het ander. Er zit niets tussen. We beginnen met een vertrouwd onderdeel: de aan/uit schakelaar. Van daaruit bouwen we verder naar poorten en andere logische schakelingen. Bekijk Figuur 11.3-1



Figuur 11.3-1. a. EN (AND)-schakeling met schakelaars, b. OF (OR)-schakeling met schakelaars.

De gesloten schakelaar krijgt de waarde 1. Dan is de open schakelaar 0. Voor het lampje geldt: aan is 1, uit is 0. In schakeling a moeten S₁ **en** S₂ gesloten zijn, wil het lampje L branden. In logische termen is schakeling a) een EN- of in het Engels AND-schakeling. Het symbool voor EN is \wedge of de vermenigvuldig-punt. Dus voor schakeling a) geldt:

$$L = S_1 \wedge S_2 \equiv L = S_1 \cdot S_2$$

Spreek uit: *L* is *S*₁ **en** *S*₂. In schakeling b) is dat S₁ **of** S₂. In logische termen is schakeling b) een OF- of OR-schakeling. Het symbool voor OF is \vee of + (plus). Het symbool \equiv betekent: "is hetzelfde als". Voor schakeling b) geldt:

$$L = S_1 \vee S_2 \equiv L = S_1 + S_2$$

Spreek uit: *L* is *S*₁ **of** *S*₂. De werking van schakelingen a) en b) kunnen we samenvatten in zogenoemde *waarheidstabellen*. Tabel 11.3-1 laat ze voor beide schakelingen zien.

Tabel 11.3-1 Waarheidstabellen voor de schakelingen a en b in Figuur 11.3-1. 0 is "open", 1 is "dicht".

Schakeling a			Schakeling b		
$L = S_1 \wedge S_2 = S_1 \cdot S_2$			$L = S_1 \vee S_2 = S_1 + S_2$		
S ₁	S ₂	L	S ₁	S ₂	L
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	1

11.3.2 Poorten en hun logische functies

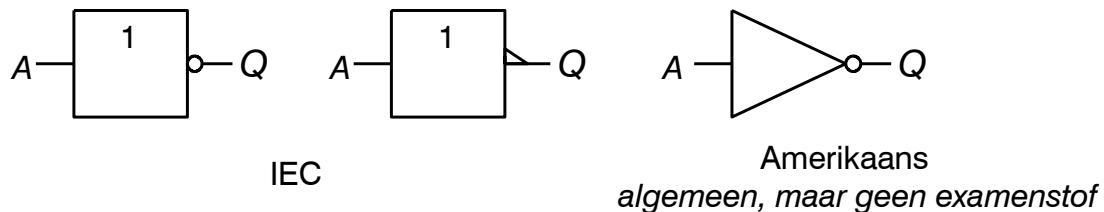
De logische functies zoals we die zojuist hebben gezien, zijn niet de enige. Ook worden ze in de praktijk niet met schakelaars uitgevoerd, maar met meer complexe geïntegreerde schakelingen (“chips”) op basis van transistoren of (MOS)FET’s. Een “1” is dan een spanning tussen de halve en de hele positieve voedingsspanning en een “0” een spanning dicht bij 0 V, in de praktijk minder dan ongeveer 0,8 V. Dat is niet meer dan een afspraak, maar wel één waaraan praktisch iedereen zich houdt.

Zulke schakelingen heten *poortschakelingen* of kortweg *poorten*. Ze hebben elk hun eigen schemasymbool. Helaas zijn er verschillende soorten symbolen in omloop. Er zijn Engelse, Duitse en Amerikaanse in omloop geweest. Bij een groot bedrijf als Philips waren ooit op verschillende afdelingen verschillende symbolen in gebruik(!).

Europa is tegenwoordig gestandaardiseerd op de zogenoemde IEC-symbolen. IEC staat voor International Electrotechnical Commission. De Nederlandse normen (NEN5152) zijn erop gebaseerd. In een groot deel van de wereld worden nog de Amerikaanse symbolen gebruikt. Men spreekt daarover wel eens als “schoenzolen en strijkijzers”. We laten ze naast de IEC-symbolen zien. Dan wordt de oorsprong van die termen wel duidelijk. We lopen de verschillende vormen één voor één langs.

11.3.3 De NIET- (NOT-)poort en de buffer

De NIET-poort (Engels: NOT) zet 1 om in 0 en 0 in 1. 1 erin, 0 eruit en omgekeerd. Hij heet ook *negator*, ontkenner. In de Amerikaanse terminologie heet hij meestal *inverter*. *Inverter* is een ook in Nederland veel gebruikte term. Inverteren is omkeren.



Figuur 11.3-2. Schemasymbolen voor de NIET-poort. Het rondje op de uitgang symboliseert in beide gevallen de omkeersfunctie. Het driehoekje op de uitgang van het tweede IEC-symbool betekent hetzelfde.

Het rondje en het driehoekje op de uitgangen van de twee IEC-symbolen betekenen hetzelfde: NIET. Beide zijn toegestaan. De schrijfwijze is $Q = \bar{A}$. Spreek uit: “Q is niet A” of “Q is A-niet”. Het streepje boven de A betekent “niet”. Het heet ook *negatiestreep*. We zullen hem veel tegenkomen. De waarheidstabel is de kleinst mogelijke (Tabel 11.3-2).

Tabel 11.3-2. Waarheidstabel voor de NIET-poort (Figuur 11.3-2)

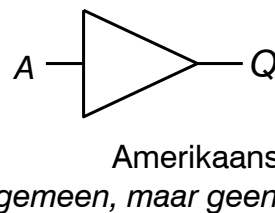
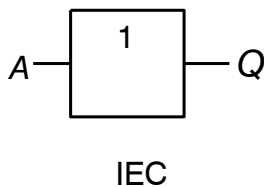
$Q = \bar{A}$	
Ingang (A)	Uitgang (Q)
1	0
0	1

Een poort met één ingang en één uitgang die niet invertteert, bestaat ook. Die heet *buffer*. Je komt hem weinig tegen. Hij is vooral bedoeld voor toepassingen waarbij een uitgang meer poortingangen moeten aansturen dan waarvoor hij is gemaakt. Wat erin gaat, komt eruit, maar met meer vermogen. De waarheidstabel (Tabel 11.3-3) is dan ook de simpelste van allemaal.

Tabel 11.3-3. Waarheidstabel voor de buffer

Q = A	
Ingang (A)	Uitgang (Q)
1	1
0	0

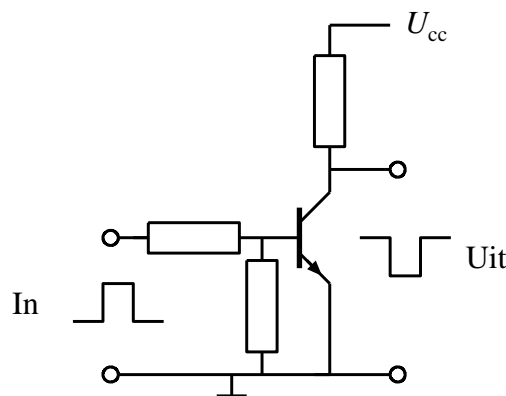
Het schemasymbool staat in Figuur 11.3-3.



Figuur 11.3-3. Schemasymbolen voor de buffer. Geen rondje of driehoekje op de uitgang, want er wordt niets geïnverteerd.

Het binnenwerk van een NIET-poort

Het inwendige van een NIET-poort kan eenvoudig zijn: een transistor in gemeenschappelijke emitterschakeling (GES, Figuur 11.3-4). In hoofdstuk 8 hebben we gezien dat die schakeling een faseverandering van 180° veroorzaakt. Simpel gezegd: het signaal gaat op de kop. Bij een NIET-poort is dat de bedoeling.



Figuur 11.3-4. Eenvoudige NIET-schakeling. De afkorting U_{cc} wordt algemeen gebruikt om de voedingsspanning voor het collectorcircuit aan te geven.

Het zal de oplettende lezer opvallen dat in Figuur 11.3-4 de emitterweerstand ontbreekt. Dat is precies de bedoeling. De transistor moet, zoals dat heet, *in verzadiging* komen. Verzadiging ontstaat als de basisstroom zo groot wordt dat deze niet meer leidt tot een

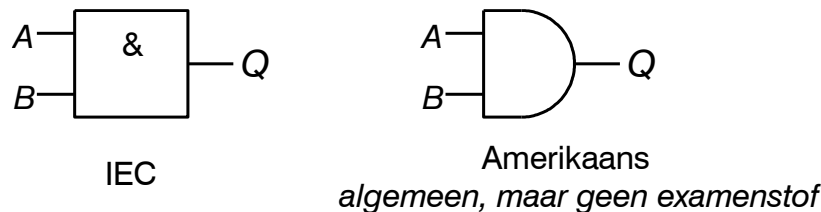
evenredige toename van de collectorstroom. Dat is in de buurt van de *kniespanning* (hoofdstuk 8) van de transistor. De spanning over een Si-transistor kan daarbij teruglopen tot bijvoorbeeld 0,4 V. Dat houdt in dat zowel de overgang van emitter naar basis als die van basis naar collector geleidend zijn. Dat op zich is geen examenstof, maar soms kom je in examens tegen dat een transistor “niet verzadigd” is. Vooral daarom gebruiken we de term. Lineair versterken is bij digitale schakelingen niet aan de orde. Het gaat om 1 of 0, meer niet.

Dat de spanning over een verzadigde transistor zo laag is, maakt gelijkstroomkoppeling van poortschakelingen mogelijk. Dat laatste is nodig, want het kan zijn dat eenzelfde signaalniveau gedurende lange tijd moet worden doorgegeven. Om in- en uitgangsspanningen op elkaar afgestemd te houden, worden waar nodig dioden in de schakeling opgenomen. In geïntegreerde schakelingen (*Integrated Circuits, IC's*) met veel transistoren, dioden, enz. op één chip zijn zulke zaken “eenvoudig” realiseerbaar.

11.3.4 De EN- en de OF-poort

De EN-poort

De EN-poort (Engels: *AND-gate*) hebben we in de vorm van schakelaars al gezien in Figuur 11.3-1. Het schemasymbool staat in Figuur 11.3-5 hieronder.



Figuur 11.3-5. Schemasymbolen voor de EN-poort.

De schrijfwijze is $Q = A \cdot B$ of $Q = A \wedge B$. Spreek uit: “Q is A èn B”. In de figuur is de EN-poort getekend met twee ingangen. Dat mogen er (onbeperkt veel) meer zijn. De waarheidstabel voor de symbolen in Figuur 11.3-5 zien we in Tabel 11.3-4.

Tabel 11.3-4. Waarheidstabel voor de EN-poort van Figuur 11.3-5

$Q = A \cdot B \equiv A \wedge B$		
A	B	Q
0	0	0
1	0	0
0	1	0
1	1	1

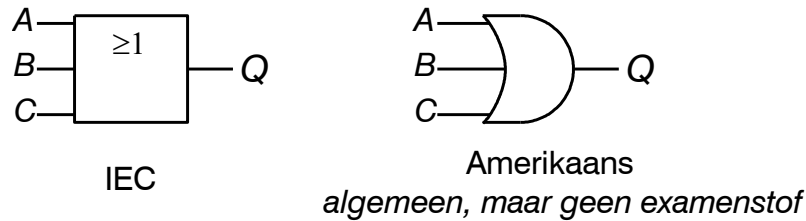
Uit Tabel 11.3-4 kunnen we nog een bijzonderheid van de EN-functie aflezen:

$$A \cdot \bar{A} = 0$$

Als $A = 1$, dan is $\bar{A} = 0$. Er is dan altijd een ingang 1 en ingang 0. De tweede en de derde regel in Tabel 11.3-4 geven die situatie weer. De uitkomst Q is voor beide 0.

De OF-poort

De OF-poort (Engels: *OR gate*) kennen we net als de EN-poort in de vorm van de parallelle schakelaars in Figuur 11.3-1. Het schemasymbool vinden we in Figuur 11.3-6.



Figuur 11.3-6. Schemasymbolen voor de OF-poort.

De schrijfwijze is $Q = A + B + C$ of $Q = A \vee B \vee C$. Spreek uit: “Q is A of B of C”.

Voor de verandering zijn er drie ingangen weergegeven om te laten zien dat twee ingangen bij deze en de vorige poortschakeling niet verplicht is. Alles met twee of meer ingangen kan, net als bij de EN-poort. Tabel 11.3-5 is de waarheidstabel.

Tabel 11.3-5. Waarheidstabel voor de OF-poort van Figuur 11.3-6

$Q = A + B + C \equiv A \vee B \vee C$			
<i>A</i>	<i>B</i>	<i>C</i>	<i>Q</i>
0	0	0	0
1	0	0	1
0	1	0	1
1	1	0	1
0	0	1	1
1	0	1	1
0	1	1	1
1	1	1	1

Zodra *A* of *B* of *C* de waarde 1 heeft, is *Q* ook 1. Alleen als *A*, *B* en *C* alle drie 0 zijn, is *Q* ook 0.

Voor de liefhebbers, géén examenstof! Negatieve logica.

In Tabel 11.3-5 kun je zien dat de bijbehorende OF-poort ook is te beschouwen als een EN-poort voor nullen. Op dezelfde manier kun je een EN-poort voor enen zien als een OF-poort voor nullen.

Deze omkering staat bekend onder de naam *negatieve logica* (*negative logic*). Wiskundig gezien is dat volkomen in orde, zolang er maar consequent mee wordt omgegaan. Negatieve logica wordt in bijzondere gevallen toegepast en is geen examenstof.

Tabel 11.3-5 is twee keer zo lang als Tabel 11.3-4. Dat komt doordat Tabel 11.3-5 gaat over een poortschakeling met 3 ingangen in plaats van 2 in Tabel 11.3-4. De waarheidstabel van een poort wordt twee keer zo lang bij elke toegevoegde ingang. Dat zit hem in het tweetallige getallenstelsel, waarvan deze logica gebruik maakt.

Net als de EN-poort heeft de OF-poort een eigenaardigheid. Bij de EN-poort gold dat

$$A \cdot \bar{A} = 0 \equiv A \wedge \bar{A} = 0$$

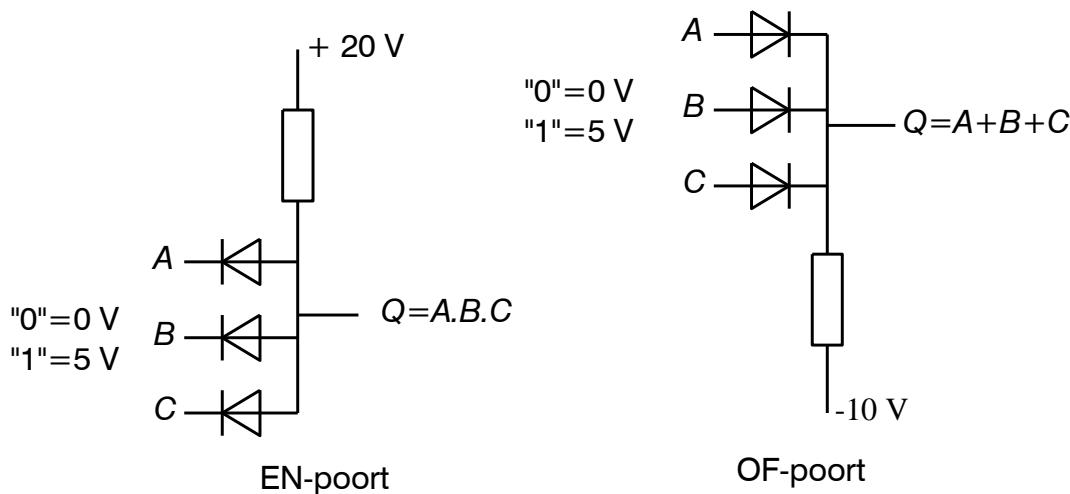
Voor de OF-poort geldt

$$A + \bar{A} = 1 \equiv A \vee \bar{A} = 1$$

Als maar één van de ingangen van een OF-poort 1 is, is de uitgang ook 1. Bij $A + \bar{A}$ is dat altijd het geval, want als de één 0 is, is de ander $\bar{0} = 1$.

Het binnenwerk van een EN- of OF-poort

Het is mogelijk, met alleen dioden een EN- of een OF-poort te maken. Dat gebeurde al in de oertijd van de computer, toen die nog voor een belangrijk deel met buizen werkte. Halfgeleiderdioden bespaarden ruimte en niet te vergeten warmteontwikkeling. In Figuur 11.3-7 zien we van zowel de EN als de OF een voorbeeld. De techniek stond bekend als DDL (Diode-Diode Logic).



Figuur 11.3-7. Een EN- en een OF-poort, uitgevoerd met dioden.

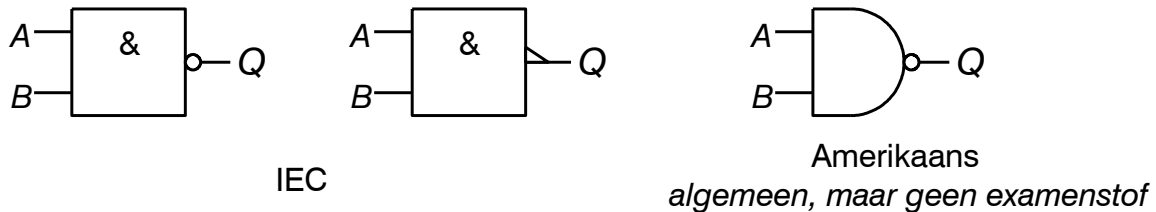
We bekijken de EN-poort (links). Een logische "1" is 5 V, een "0" is 0 V. Stel $A=0$. Dan is ook $Q = 0$, want een "1" op B of C stuit op een gesperde diode. Als $A = 1$, dan wordt die "1" alleen doorgegeven naar Q als de andere twee poorten ook 1 zijn. Dat is EN-gedrag. De weerstand voorkomt eventuele kortsluitschade. De spanning aan de "bovenkant" van de weerstand moet groter zijn dan 5 V.

Bij de OF-poort rechts liggen de zaken omgekeerd. Als één van de ingangen "1" is, staat op de uitgang al 5 V. De andere dioden sperren als op de betreffende ingang 0 V staat. Dat betekent OF-gedrag.

11.3.5 De NIET-EN- (NEN-) en de NIET-OF- (NOF-)poort

De NEN-poort

Een NIET-EN- of NEN-poort (Engels: *NAND gate*) is een EN-poort met een NIET-uitgang. In schema's heeft hij hetzelfde rondje op de uitgang dat we kennen van de NIET-poort. Bij de IEC-symbolen zie je ook wel eens het driehoekje. Figuur 11.3-8 laat ze zien.



Figuur 11.3-8. Schemasympolen voor de NEN-poort. Het rondje op de uitgangen van het linker IEC-symbool en het Amerikaanse symbool betekenen ook hier NIET, net als het driehoekje op de uitgang van het rechter IEC-symbool.

De uitgang met rondje of driehoekje heet ook wel *genegeerde uitgang*.. De schrijfwijze is $Q = \overline{A \cdot B}$ of $Q = \overline{A \wedge B}$. Spreek uit: "Q is niet A en B". De waarheidstabel is Tabel 11.3-6.

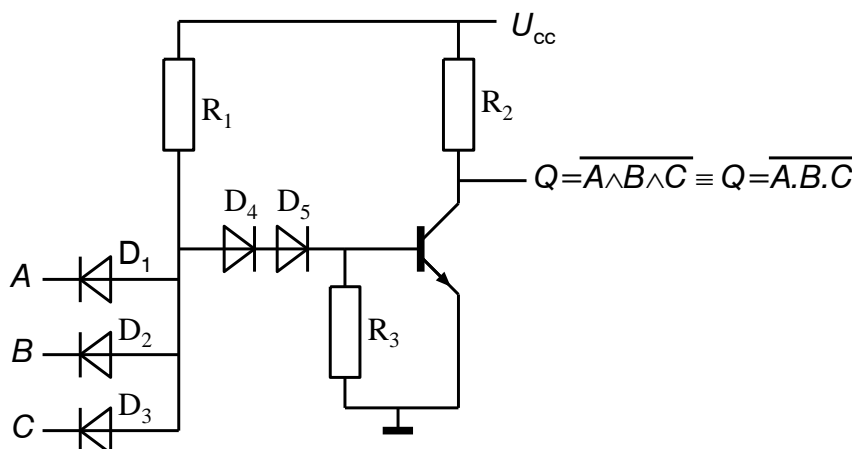
Tabel 11.3-6. Waarheidstabel voor de NEN-poort van Figuur 11.3-8.

$Q = \overline{A \cdot B} \equiv \overline{A \wedge B}$		
A	B	Q
0	0	1
1	0	1
0	1	1
1	1	0

Dit is het tegengestelde van Tabel 11.3-4 (en anders dan de tabel van een OF-poort).

Het binnenwerk van een NEN-poort

Dat kan er bijvoorbeeld uitzien als



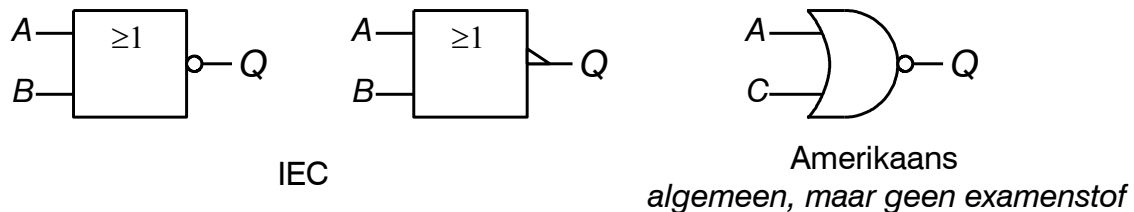
Figuur 11.3-9. Een voorbeeld van een NEN-poort in DTL (Diode Transistor Logic). Het symbool \equiv betekent "is hetzelfde als". De omkerende transistor zagen we eerder in Figuur 11.3-4.

De poort is opgebouwd in DTL, Diode Transistor Logic. Ook dat is een vorm van logische schakelingen die je tegenwoordig vrijwel niet meer tegenkomt, maar geschikt is om de werking van een poort te laten zien als je DDL kent (subparagraaf 11.3.4). De signalen A , B en C komen links binnen via dioden D_1 , D_2 en D_3 . Als A of B of C de waarde 0 heeft, heeft het knooppunt van R_1 en de dioden een lage spanning, een 0 dus. Die wordt pas 1 als $A \cdot B \cdot C = 1$. Dat hoort bij een EN-poort. Het signaal moet nog wel de transistor door. Daarin gaat het op de kop, zodat we aan de uitgang niet $A \cdot B \cdot C$, maar $\overline{A \cdot B \cdot C}$ vinden.

De dioden D_4 en D_5 moeten de basisspanning van de transistor voldoende laag houden om de transistor te sperren als er een 0 op het ingangsknooppunt staat.

DE NIET-OF- of NOF-poort

De NOF-poort (Engels: *NOR gate*) is een OF-poort met een NIET-uitgang. Net als bij zijn NEN-collega wordt het NIET-karakter aangegeven met een rondje of driehoekje aan de uitgang (Figuur 11.3-10)



Figuur 11.3-10. Schemasymbolen voor de NOF-poort. Het driehoekje en de beide tondjes symboliseren het NIET-karakter.

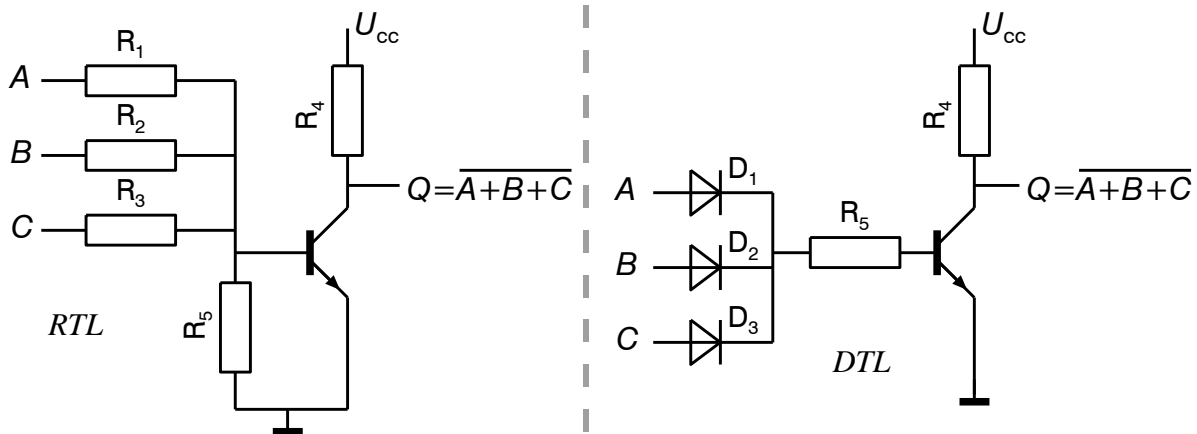
De schrijfwijze is $Q = \overline{A + B}$ of soms $Q = \overline{A \vee B}$. Spreek uit: “Q is niet A of B”. De bijbehorende waarheidstabel vinden we in Tabel 11.3-7 hieronder.

Tabel 11.3-7. Waarheidstabel voor de NOF-poort van Figuur 11.3-10.

$Q = \overline{A + B} = \overline{A \vee B}$		
A	B	Q
0	0	1
1	0	0
0	1	0
1	1	0

Het binnenwerk van een NEN- en NOF-poort

Twee voorbeelden van het binnenste van een NOF-poort vinden we in Figuur 11.3-11.



Figuur 11.3-11. Links: NOF-poort in RTL (Resistor Transistor Logic). Rechts: NOF-poort in DTL (Diode Transistor Logic).

De linker poort in de figuur is een voorbeeld van RTL (Resistor Transistor Logic). Deze vorm van logica is nog wat ouder dan DTL in het rechter voorbeeld. In beide gevallen is een logische 1 op één van de ingangen genoeg om de transistor in geleiding te sturen. Dat is het gedrag van een OF-poort. De transistor zet het signaal op de kop, zodat er geen OF-, maar een NOF-poort ontstaat. Merk op dat in de DTL-schakeling de dioden omgekeerd staan in vergelijking met de NEN-poort in Figuur 11.3-9.

11.3.6 De Exclusieve OF- of EXOF-poort

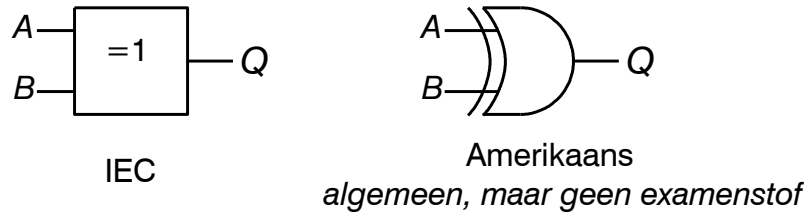
De uitgang is 1 als beide ingangen tegengesteld zijn en 0 als ze gelijk zijn. Dat lukt niet met een enkele poort. Je hebt er drie nodig (eigenlijk vijf, als je de NIETen meetelt). De EXOF-poort (Engels: *EXOR of XOR*) is dus eigenlijk een combinatiepoort. Als je zo'n poortschakeling aanschaft (kost een bedragje van niks), zitten ze wel allemaal, meestal met z'n vieren, in één geïntegreerde schakeling (IC) ofwel één blokje.

De schrijfwijze: $A \cdot \bar{B} + \bar{A} \cdot B$., maar meestal $A \oplus B$. Dat laatste is korter. We geven eerst de waarheidstabel (Tabel 11.3-8), daarna de schemasymbolen (Figuur 11.3-12).

Tabel 11.3-8. Waarheidstabel voor de EXOF-poort van Figuur 11.3-12

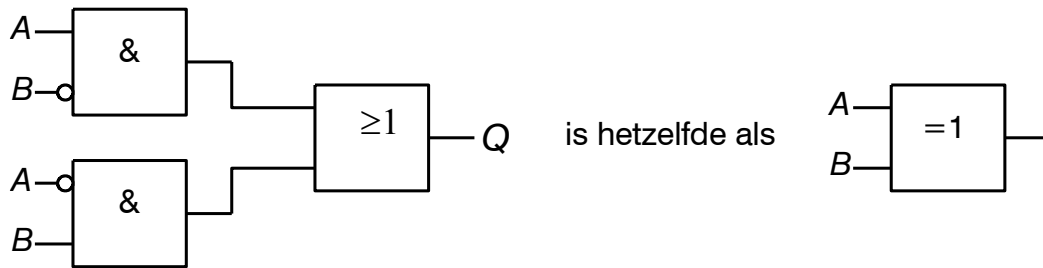
$Q = A \cdot \bar{B} + \bar{A} \cdot B = A \oplus B$		
A	B	Q
0	0	0
1	0	1
0	1	1
1	1	0

$Q = 0$ als de ingangen gelijk zijn en $Q = 1$ als ze ongelijk zijn.



Figuur 11.3-12. Schemasymbolen voor de EXOF-poort.

Reken zelf na aan de hand van de schakeling die hoort bij de vergelijking in Tabel 11.3-8.

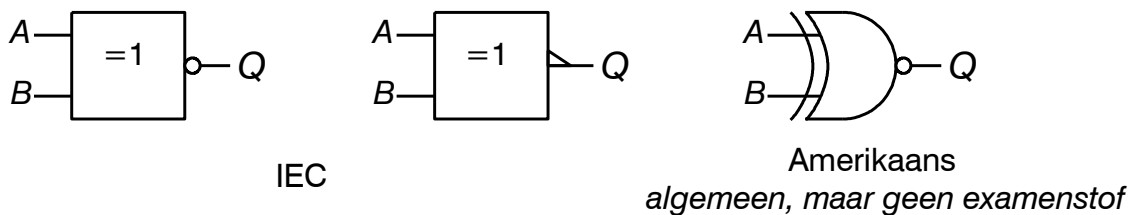


Figuur 11.3-13. EXOF-schakeling (links) en als enkele poort getekend (rechts).

De twee ingangen met rondjes (mogen ook driehoekjes zijn) zijn zogenoemde *genegeerde* ingangen. Dat zijn ingangen met een “ingebouwde” NIET-poort. Zo ontstaat een iets overzichtelijker beeld dan met een aparte NIET-poort (inverter) vóór die ingangen. De waarde A op een ingang met een rondje wordt bij binnenkomst in de schakeling eerst omgezet naar \bar{A} . Of dat binnen deze poort gebeurt of in een voorgeschakelde inverter, doet er voor het resultaat niet toe. Hetzelfde geldt voor B .

11.3.7 De Exclusieve NOF (EXNOF, XNOF)-poort

Ook de EXOF-poort heeft zijn NIET-variant, de EXNOF (*Engels: Ex-NOR of XNOR*). De schemasymbolen zullen geen verrassing meer zijn: een rondje op de uitgang of bij het IEC-symbool eventueel een driehoekje.



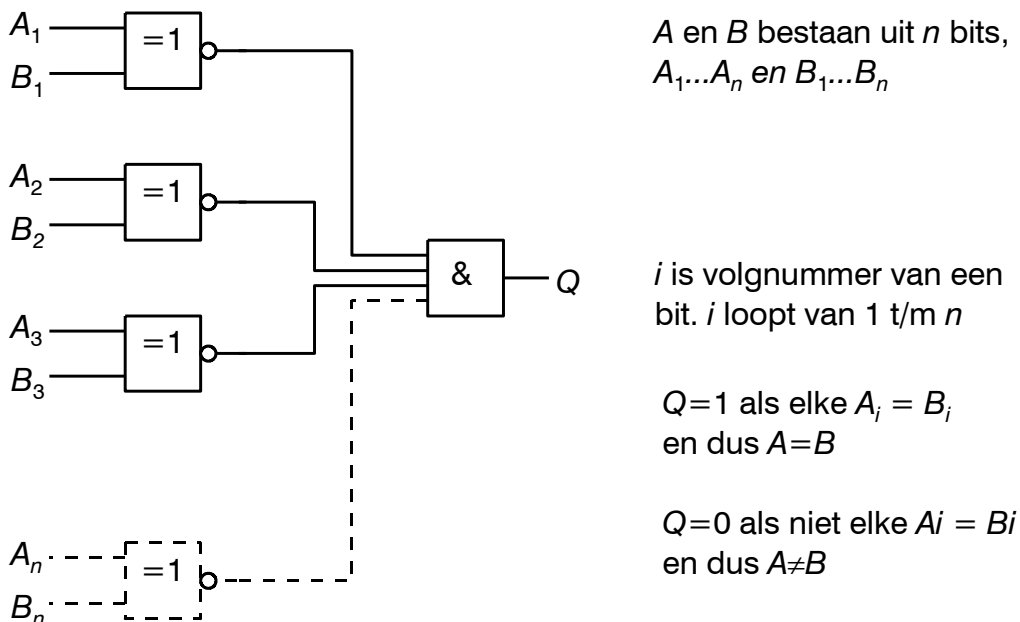
Figuur 11.3-14. Schemasymbolen voor de EXNOF-poort.

Tabel 11.3-9 toont de waarheidstabel.

Tabel 11.3-9. Waarheidstabel voor de EXNOF-poort van Figuur 11.3-14

$Q = \overline{A \cdot \overline{B}} + \overline{\overline{A} \cdot B} = \overline{A \oplus B}$		
A	B	Q
0	0	1
1	0	0
0	1	0
1	1	1

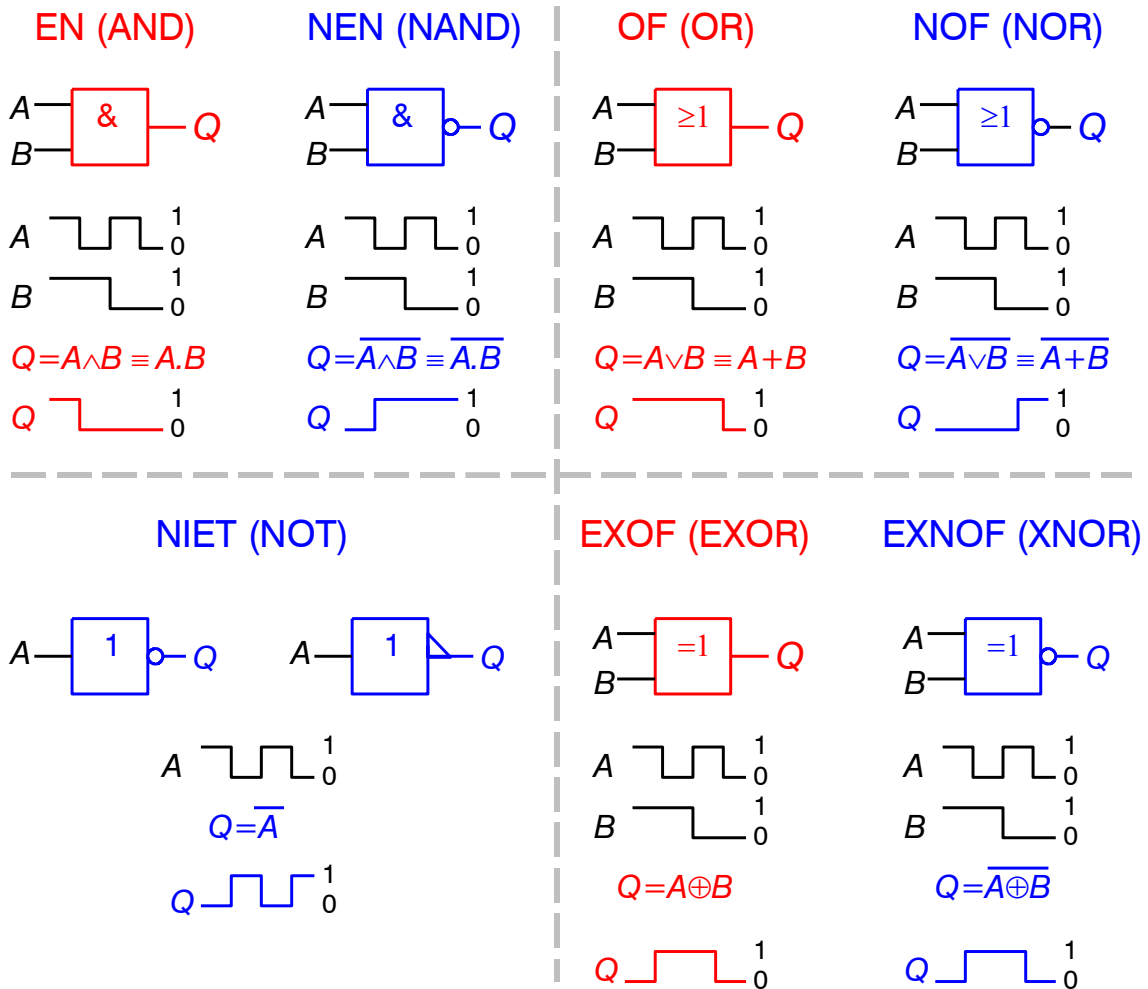
Met EXNOF (of een EXOF) kun je bijvoorbeeld vaststellen of twee binaire getallen even groot zijn. Dat gaat bit voor bit. Figuur 11.3-15 laat zien hoe dat gaat met EXNOF-poorten en twee getallen A en B van n bits. De EN-poort moet n ingangen hebben.



Figuur 11.3-15. Vergelijken van twee binaire getallen die elk bestaan uit n bits, $A_1 \dots A_n$ en $B_1 \dots B_n$. De EN-poort moet n ingangen hebben. Die kan eventueel uit meer dan één EN-poort worden samengesteld.

11.3.8 De poorten samengevat

Figuur 11.3-16 toont alle behandelde poorten in één overzicht. In- en output is weergegeven met blok- en pulssignalen in plaats van een waarheidstabel. Hoog is 1, laag is 0. Wat de poorten ingaat, is zwart weergegeven. Wat eruit komt is **blauw** als er een NIET aan te pas komt, **rood** als dat niet het geval is. De schrijfwijze is in de bijbehorende kleur. Bij EN- en NEN-poorten laten we twee geldige schrijfwijzen zien. Die zijn gescheiden met een “ \equiv ”-teken (“*is hetzelfde als*”).

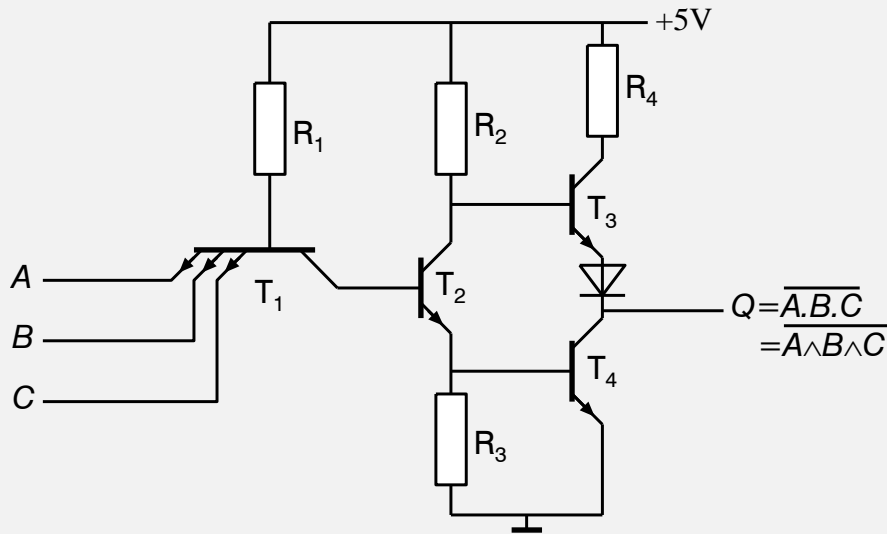


Figuur 11.3-16. Alle behandelde poorten en hun werking in 1 overzicht. Zwart: alles wat gemeenschappelijk is. Rood: poorten zonder negatie (zonder NIET). Blauw: poorten met negatie (met NIET).

11.3.9 Transistor-transistor logica (TTL) (tot nu toe geen examenstof)

Voor de liefhebbers. Nog nooit gevraagd op een examen...

TTL is gebaseerd op transistoren. Dioden nemen een beperkte plek in. De ingangen zijn emitters van één multi-emittertransistor. De techniek dateert uit de zestiger jaren van de 20^e eeuw. Wie er meer over wil lezen, kan bijvoorbeeld terecht op de helaas Engelstalige webpagina https://en.wikipedia.org/wiki/Transistor%E2%80%93transistor_logic.



Figuur 11.3-17. NEN-poort in TTL met multi-emitteringang en "Totem pole" uitgang.

Links transistor T_1 met drie emitteringangen en de ingangssignalen A , B en C . De basis van T_1 krijgt stroom via R_1 . Als maar één van de drie ingangen 0 is, geleidt T_1 . Daardoor spert T_2 . Dan is de collectorspanning van T_2 1 en de emitterspanning 0 en geleidt T_3 , spert T_4 en is de uitgang 1. Zijn alle drie de ingangen 1, dan spert T_1 , geleidt T_2 , spert T_3 en geleidt T_4 . Dan is de uitgang 0.

Dit is het gedrag van een NEN-poort. De uitgangsschakeling is een zogenoemde Totem Pole. In het schema zien we een rijtje van een weerstand (R_4), transistor (T_3), diode en weer een transistor (T_4) boven elkaar.

Doordat T_3 en T_4 worden aangestuurd vanuit de collector, resp. de emitter van T_2 die in tegenfase zijn, geleidt óf T_3 en spert T_4 , óf T_3 spert en T_4 geleidt. De uitgangsimpedantie blijft op deze manier ongeveer even hoog, ongeacht of het om een 1 of een 0 gaat. Omdat de emittervolger T_3 zonder maatregelen niet helemaal dichtgaat, zorgt de diode voor een extra spanningsdrempel.

Dit soort uitgang met twee transistoren of FETs in serie, waarbij de één dicht en de ander open is, komen we tegenwoordig bij de meeste logische IC's tegen.

11.4 Booleaanse algebra (schakelalgebra)

11.4.1 Inleiding

De eerdergenoemde heer George Boole heeft een vorm van wiskunde ontwikkeld, waarmee de logische uitdrukkingen waarmee we hebben kennis gemaakt, in onderling verband kunnen worden bewerkt. We gaan daar niet heel diep op in; voor het zendexamen is een korte kennismaking voldoende. Het nut van deze techniek die ook wel *Booleaanse algebra* wordt genoemd, is vooral het vereenvoudigen van logische schakelingen. In deze paragraaf zetten we de regels op een rij. Het is niet de bedoeling dat je dit allemaal uit het hoofd leert, wel dat je het snapt. Het meeste doe je op gevoel al vanzelf goed. Als je iets wel grondig moet kennen, wordt dat vermeld. Probeer uitdrukkingen in een waarheidstabel of schema te vatten.

11.4.2 Regels met 1 of 0

NIET 1 kan alleen 0 zijn. Meer smaken zijn er niet. Dus: $\bar{1} = 0$. NIET NIET 1 is weer 1, dus $\bar{\bar{1}} = 1$. Twee NIETen (negatiestrepen) vallen tegen elkaar weg, net als twee opeenvolgende mintekens in de rekenkunde. Min min is plus, NIET NIET is WEL. Met een oneven aantal NIETen houd je één NIET over.

NIET 0 kan alleen maar 1 zijn. Opnieuw: meer smaken zijn er niet. Dus $\bar{0} = 1$, maar net als in de vorige alinea voor de 1, geldt $\bar{\bar{0}} = 0$.

11.4.3 Regels met één variabele

$$\bar{\bar{A}} = A \quad (11.4-1)$$

NIET NIET A is A. Een variabele staat voor 0 of 1, dus geldt daarvoor hetzelfde als voor 0 en 1.

$$A \cdot A = A \quad (11.4-2)$$

Als $A=1$, dan is de uitkomst $1 \cdot 1 = 1$; is $A=0$, dan is de uitkomst $0 \cdot 0 = 0$. Ook geldt:

$$A \wedge 1 = A \quad (11.4-3)$$

Voor de verandering gebruikten we in (11.4-3) het andere EN-symbool.

Als één van de ingangen van een EN-poort met twee ingangen 1 is, volgt de uitgang de waarde van de andere ingang, want $0 \cdot 1 = 0$ en $1 \cdot 1 = 1$. Heeft een EN-poort meer ingangen dan nodig, dan worden de overtollige ingangen (via een weerstand) verbonden met de voedingsspanning. Die wordt gezien als 1.

$$A \wedge 0 = 0 \quad (11.4-4)$$

De volgende kennen we al van subparagraaf 11.3.4

$$A \cdot \bar{A} = 0 \quad (11.4-5)$$



En deze ook:

$$A + A = A \quad (11.4-6)$$

Als $A = 1$, dan is de uitgang van de OF-poort 1; is $A = 0$, dan is de uitgang 0. In beide gevallen komt er dus A uit. Maar nu komen de verschillen met de EN-poort.

$$A + 0 = A \quad (11.4-7)$$

Bij een OF-poort hoeft maar één van de ingangen 1 te zijn, dan is de uitgang is ook 1. De toestand van de andere ingang(en) doet er dan niet meer toe. Daarom geldt:

$$A + 1 = 1 \quad (11.4-8)$$

Bij een OF-poort hoeft maar één ingang 1 te zijn om een 1 op de uitgang te krijgen. De toestand van de andere ingang(en) doet er dan niet meer toe. Heeft een OF-poort meer ingangen dan nodig, verbind dan de overtollige ingangen met 0 V en alles werkt zoals bedoeld.

$$\bar{A} + A = 1 \quad (11.4-9)$$

Als twee variabelen elkaars tegengestelde zijn is er altijd ééntje 1 en ééntje 0. Bij een OF-poort levert dat altijd 1 op en bij een EN-poort altijd 0, zoals in (11.4-5).

11.4.4 Regels met twee of meer variabelen

Commutatieve wetten

De term *commutatief* is afgeleid van *commutare*, onderling verwisselen.

$$A \cdot B = B \cdot A \quad (11.4-10)$$

Bij een EN-, OF-, NEN-, NOF of EXOF-poort maakt het niet uit, welke ingang welk signaal ontvangt. Ze zijn dus verwisselbaar. Dus geldt bijvoorbeeld ook:

$$A + B = B + A \quad (11.4-11)$$

Kort samengevat: bij een EN-, een OF- of een EXOF-constructie maakt de volgorde van de ingangsvariabelen niets uit.

Associatieve en distributieve wetten

Vertaald: *verbindingswetten* en *verdelingswetten*. Die hebben betrekking op het gebruik van haakjes. Verbindingswetten hebben betrekking op haakjes in vergelijkingen met één soort bewerking, verdelingswetten op vergelijkingen met meer dan één soort. De betekenis van haakjes is vergelijkbaar met die in de gewone rekenkunde. Er is één belangrijk verschil: er is geen speciale voorrang voor bepaalde bewerkingen zoals in de rekenkunde, waar bijvoorbeeld vermenigvuldigen voorrang heeft op optellen.

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C = A \cdot B \cdot C \quad (11.4-12)$$



Van twee EN-poorten met twee ingangen, maak je effectief één EN-poort met drie ingangen. $B.C$ is één poort en de uitkomst gaat met A de volgende poort in of $A.B$ is er één en de uitkomst gaat met C de volgende poort in. Dat zullen we in de volgende paragraaf over combinatieloga uitvoeriger behandelen.

$$A + (B + C) = (A + B) + C = A + B + C \quad (11.4-13)$$

Eigenlijk hetzelfde verhaal als bij (11.4-12): van twee OF-poorten met twee ingangen maak je er effectief één met drie ingangen.

Nu twee verschillende soorten bewerkingen: EN en OF.

$$A.(B + C) = A.B + A.C \quad (11.4-14)$$

(11.4-14) gaat net als bij de rekenkunde. Maar de volgende wijkt af doordat OF en EN gelijkwaardig zijn.

$$A + (B.C) = (A + B).(A + C) \quad (11.4-15)$$

Deze werkt net als (11.4-14), al lijkt dat op het eerste gezicht niet zo. Dat komt doordat EN en OF in het Booleaans gelijkwaardig zijn. Ten opzichte van(11.4-14) is in (11.4-15) alle EN vervangen door OF en alle OF door EN. **Eén om te onthouden!**

Absorptiewetten

Dit zijn wetten waarbij de één of andere vorm van vereenvoudiging optreedt. Ze zijn vermoedelijk met een beetje hulp voor de beginner redelijk goed te begrijpen Die hulp staat er dan ook bij. Maak eventueel voor jezelf een waarheidstabel.

$$A + A.B = A \quad (11.4-16)$$

Deze is snel te snappen. Als $A = 0$, komt er 0 uit en als $A = 1$, komt er 1 uit. B heeft geen invloed op de uitkomst en is overbodig.

$$A.(A + B) = A \quad (11.4-17)$$

Dit is een uitbreiding van (11.4-16). $A.A = A$. Dan krijgen we $A + A.B$ en zijn we terug bij vergelijking (11.4-16).

$$A.(\bar{A} + B) = A.B \quad (11.4-18)$$

$A.\bar{A} = 0$. Dan houden we $0 + A.B = A.B$ over, want een 0 op één ingang van een OF-poort met twee ingangen betekent dat de toestand op de andere ingang die van de uitgang bepaalt.

$$A + \bar{A}.B = A + B \quad (11.4-19)$$

Als $A = 0$, dan is de uitkomst B . Als $A = 1$, dan is de uitkomst A . De uitkomst is daarom $A + B$. **Deze niet verwarren met (11.4-16)!**

De stellingen van De Morgan

Als laatsten krijgen we de twee stellingen van De Morgan. Die zijn de belangrijkste van allemaal. Het is goed, deze twee te kennen, want achter een examentafel bedenk je dit niet. Je kunt met De Morgan van een NEN-poort een OF-poort maken en van een NOF een EN. De oplettende lezer zal zich misschien het kadertje over negatieve logica bij de behandeling van de OF-poort in 11.3.2 herinneren. Eigenlijk is dat ook een De Morgan-gedachte. Augustus De Morgan was een Engels wiskundige uit de 19^e eeuw.

De twee stellingen van De Morgan leggen een verband tussen EN, OF en NIET. Hier komt de eerste.

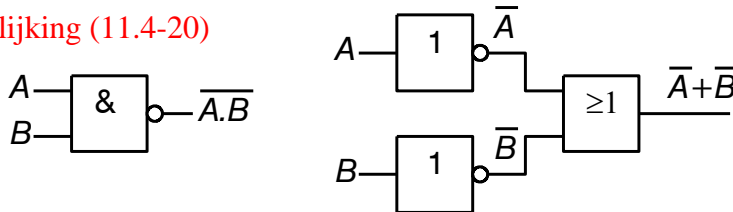
$$\overline{A \cdot B} = \bar{A} + \bar{B} \quad (11.4-20)$$

Een NEN-poort voor A en B is dus een OF-poort voor \bar{A} en \bar{B} . Datzelfde trucje geldt voor een NOF-poort, maar nu komt er een EN-poort voor \bar{A} en \bar{B} uit.

$$\overline{A + B} = \bar{A} \cdot \bar{B} \quad (11.4-21)$$

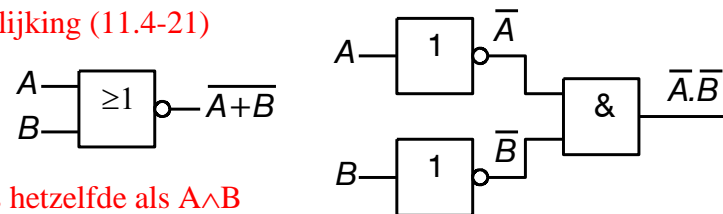
De vergelijkingen van De Morgan zijn in Figuur 11.4-1 met poorten en waarheidstabellen in beeld gebracht.

Vergelijking (11.4-20)



A	B	$\bar{A} \cdot \bar{B}$	$\bar{A} + \bar{B}$
0	0	1	1
1	0	1	1
0	1	1	1
1	1	0	0

Vergelijking (11.4-21)

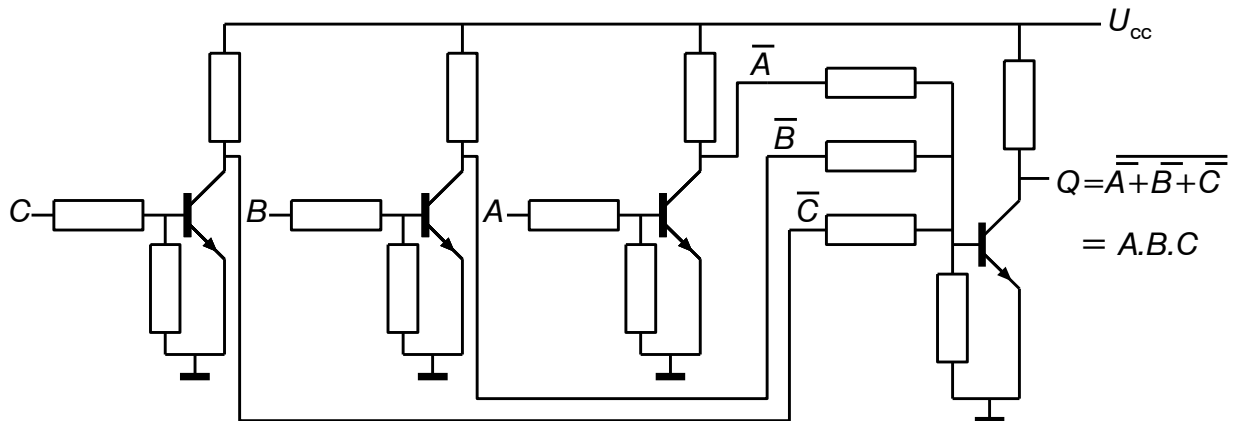


A	B	$\overline{A + B}$	$\bar{A} \cdot \bar{B}$
0	0	1	1
1	0	0	0
0	1	0	0
1	1	0	0

$A \cdot B$ is hetzelfde als $A \wedge B$

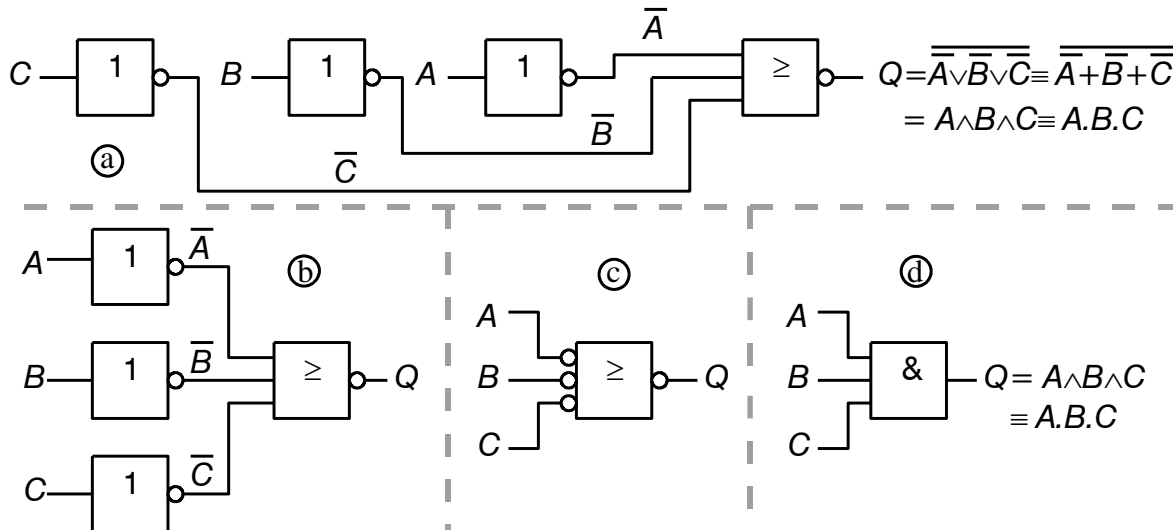
Figuur 11.4-1. Vergelijkingen van De Morgan (11.4-20) en (11.4-21) in beeld met waarheidstabellen.

In woorden: Een NEN wordt een OF met een ontkenningstreep boven elk van de variabelen. Een NOF wordt een EN met een ontkenningstreep boven elk van de variabelen. Een voorbeeld van dit laatste zien we in het volgende schema (Figuur 11.4-2).



Figuur 11.4-2. Een NOF-poort wordt een EN-poort (met dank aan De Morgan).

De drie transistoren inverteren de ingangsvariabelen A , B en C , zodat \bar{A} , \bar{B} en \bar{C} ontstaan. Die gaan de NOF-poort van de eindtransistor in en komen er als $\overline{\bar{A} + \bar{B} + \bar{C}}$ weer uit. De OF-tekens worden vervangen door EN, de ontkenningstreep over het geheel wordt dubbel (NIET NIET) en kan dus weg. Hetzelfde geldt voor de ontkenningstreep boven de afzonderlijke variabelen. Dan houden we $A \wedge B \wedge C \equiv A.B.C$ over. Het maakt niets uit of er twee of drie variabelen binnenkomen. Sterker: het mogen er ook vier of meer zijn. Hetzelfde diagram, maar nu in poorten, is weergegeven in Figuur 11.4-3.



Figuur 11.4-3. Poortendiagram van Figuur 11.4-2; a: zelfde schikking als in Figuur 11.4-2; b: zelfde schema, maar wat overzichtelijker; c: NOT-poorten vervangen door genegeerde ingangen; d: alles vervangen door één enkele EN-poort.

Om te onthouden: beide stellingen van De Morgan kunnen we omzetten naar één recept:

- Vervang elke EN door OF;
- Vervang elke OF door EN;
- Waar een NIET stond, verdwijnt deze;
- Waar geen NIET stond, komt er één.

Terug naar Figuur 11.4-3. De eerste uitkomst was $Q = \overline{\overline{A + B + C}}$. Elke “OF”, getekend als “+” of als “ \vee ”, wordt EN, naar keuze getekend als “.” of “ \wedge ”. Alle bestaande ontkenningstrepen gaan weg. Plaatsen zonder ontkenningstreep waren er niet. Dan komen er geen nieuwe ontkenningstrepen. Resultaat: $Q = A.B.C \equiv Q = A \wedge B \wedge C$.

Let op het **beginnersfoutje** dat $\overline{A + B}$ en $\overline{A + \overline{B}}$ hetzelfde zouden betekenen. **Dat is niet het geval.** Hetzelfde geldt voor $\overline{A}.\overline{B}$. Dat is **niet** hetzelfde als $\overline{A.B}$. De waarheidstabellen (Tabel 11.4-1) laten het zien.

Tabel 11.4-1. Verschil in uitkomsten bij ontkenningstreep boven afzonderlijke variabelen en boven de hele uitdrukking

A	B	$Q = \overline{A + B}$	$Q = \overline{A + \overline{B}}$	$Q = \overline{A}.\overline{B}$	$Q = \overline{A.B}$
0	0	1	1	1	1
1	0	1	0	0	1
0	1	1	0	0	1
1	1	0	0	0	0

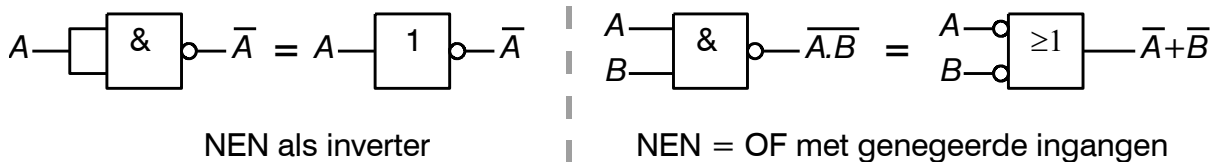
De tabel laat ook zien dat de stellingen van De Morgan kloppen, want

$\overline{A + B} = \overline{A}.\overline{B}$ en $\overline{\overline{A + B}} = \overline{\overline{A}.\overline{B}}$. Controleer met het recept **in rood** van zonet.

11.4.5 Poortconversie met De Morgan

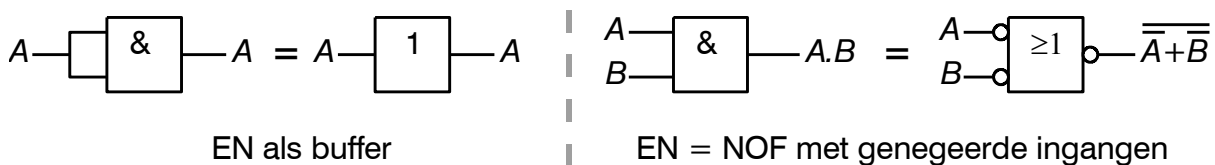
In Tabel 11.4-1 hebben we gezien dat poorten uit de EN-familie dienst kunnen doen als poorten uit de OF-familie en omgekeerd. Dit is niet alleen belangrijk bij het ontwerp van logische schakelingen; het is ook een terugkerend onderwerp op het zendexamen.

Vaak staat de NEN-poort centraal. Je kunt hem gebruiken als inverter door de ingangen kort te sluiten (of de niet gebruikte ingangen 1 te maken) en als OF-poort met genegeerde ingangen (Figuur 11.4-4). Genegeerde ingangen kennen we van subparagraaf 11.3.6.



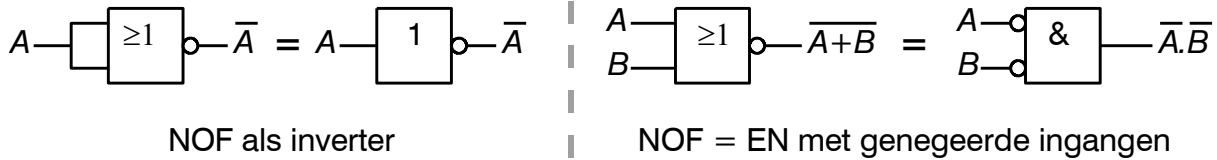
Figuur 11.4-4. NEN als inverter (links) en als OF-poort met genegeerde ingangen (rechts).

Met meer dan 2 ingangen werkt het ook. Een EN, dus zonder ontkenning, maakt er een ontkenningstreep bij (Figuur 11.4-5).



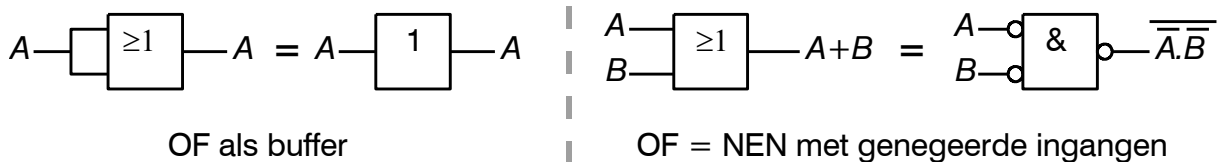
Figuur 11.4-5. EN als buffer (links) en als NOF met genegeerde ingangen (rechts).

Met een NOF kun je behalve een inverter ook een EN-poort met genegeerde ingangen maken (Figuur 11.4-6).



Figuur 11.4-6. NOF als inverter (links) en als EN-poort met genegeerde ingangen.

En met een OF lijkt het sprekend op een EN-conversie (Figuur 11.4-5), maar met alle (N)EN en (N)OF verwisseld.



Figuur 11.4-7. OF als buffer en als NEN-poort met genegeerde ingangen.

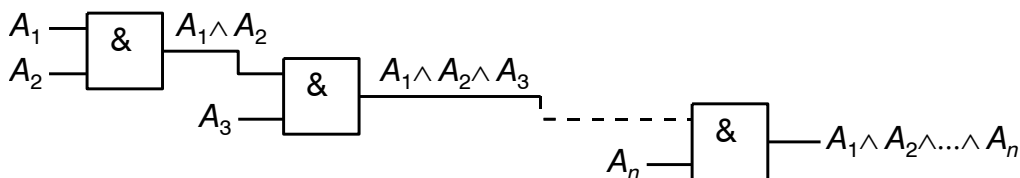
11.5 Combinatieloga (“combinational logic”)

11.5.1 Inleiding

Combinatieloga is logica waarin verschillende poorten worden gecombineerd tot een schakeling. Booleaanse algebra en vooral bewerkingen volgens De Morgan kunnen een belangrijk hulpmiddel zijn om schakelingen te ontwerpen, te vereenvoudigen of fouten te vinden en op te lossen. De vergelijkingen beginnen bij de poortingang. Wat de ingang(en) ingaat, staat aan één kant van het “=” teken. Aan de andere kant ervan staat wat er op de uitgang komt. Bij het examen komt dit soort schakelingen aan de orde. De opgaven blijven meestal beperkt tot het aangeven of er 0 of 1 uitkomt, maar de exameneisen zijn minder scherp gedefinieerd dan dat. Wees dus bedacht op ingewikkelder vragen.

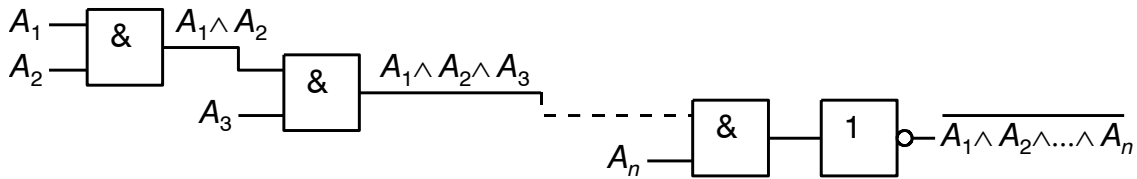
11.5.2 Een EN- of OF -poort met n ingangen

Een EN-poort met heel veel ingangen is te maken door genoeg EN-poorten achter elkaar te zetten op de manier van Figuur 11.5-1. Bij gebruik van poorten met 3 of 4 ingangen gaat het met minder poorten.



Figuur 11.5-1. Een EN-poort met n ingangen maak je op deze manier.

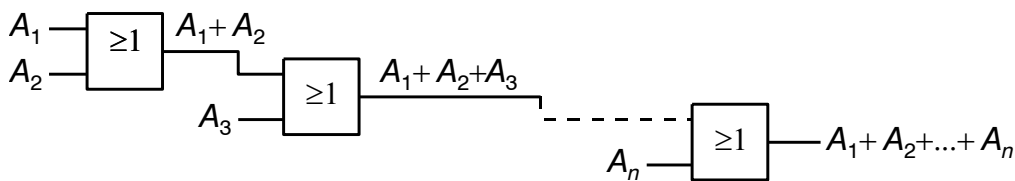
Moet het een NEN-poort worden? Gebruik dezelfde schakeling met een inverter aan het eind van de rij (Figuur 11.5-2).



Figuur 11.5-2. De NEN-versie van de schakeling van Figuur 11.5-1.

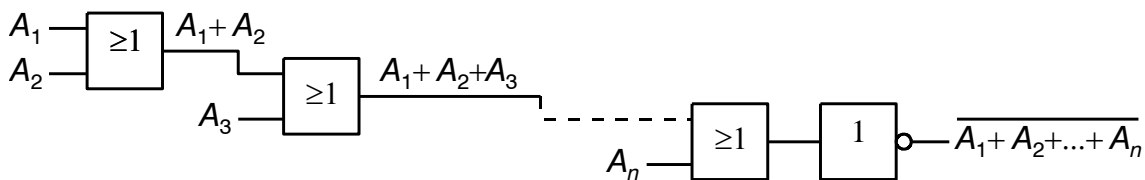
Maak niet de fout, de hele schakeling op te zetten met NEN-poorten. Bedenk zelf, waarom.

Voor een OF-poort met n ingangen gaat het precies zo (Figuur 11.5-3).



Figuur 11.5-3. Een OF-poort met n ingangen maak je op dezelfde manier als de EN in Figuur 11.5-1.

Ook hier geldt: moet het een NOF-poort worden, zet dan een inverter achter de uitgang van de meest rechtse poort in het schema.

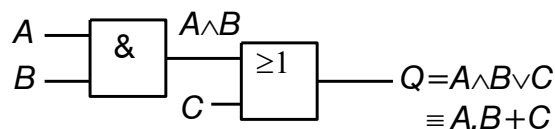


Figuur 11.5-4. De NOF-versie van Figuur 11.5-3.

Ook hier berust het opzetten van zo'n schakeling met alleen NOF-poorten op een misverstand.

11.5.3 EN en OF gecombineerd

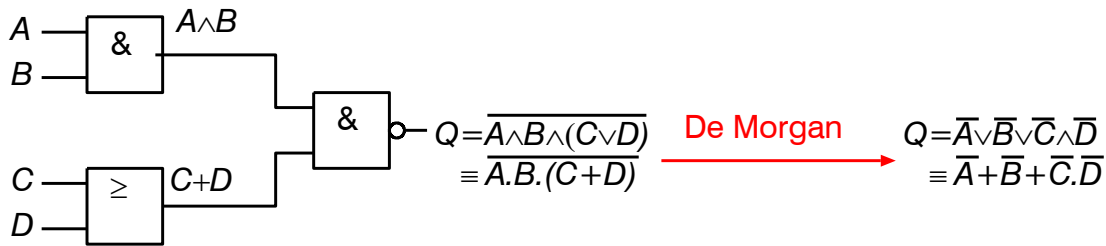
Figuur 11.5-5 toont een combinatie van een EN- en een OF-poort. $A \cdot B$ wordt via een OF-poort gecombineerd met C , waardoor de vergelijking $Q = A \cdot B + C$ ontstaat.



Figuur 11.5-5. Een EN- en een OF-poort gecombineerd.

11.5.4 Voorbeeld De Morgan met EN-poorten en OF-poort

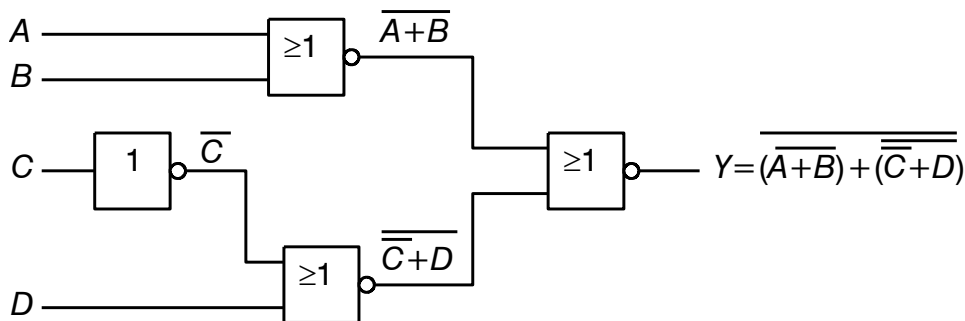
Figuur 11.5-6 is een toepassing van het rode recept van subparagraaf 11.4.4.



Figuur 11.5-6. EN- en OF-poort, samen gevolgd door EN-poort met toepassing van vergelijking (11.4-20) (een van de twee stellingen van De Morgan).

11.5.5 Meer vereenvoudigingen

We beginnen met de schakeling in Figuur 11.5-7.



Figuur 11.5-7. Schema met een inverter en drie NOF-poorten. Uitwerking in de tekst.

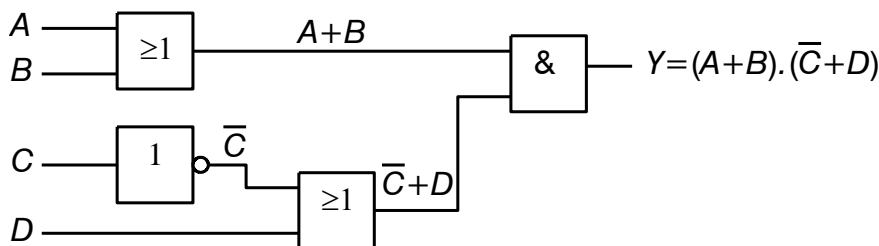
We werken de figuur uit in de tekst. De vergelijking in Figuur 11.5-7 is

$$Y = \overline{\overline{A + B} + \overline{\overline{C} + D}}$$

We passen De Morgan toe, maar laten wat binnen de haakjes staat, ongemoeid. Dan verdwijnt de lange ontkenningsstreep en de twee kortere ontkenningsstrepen boven de termen tussen de haakjes. De “+” tussen $(A + B)$ en $(\overline{C} + D)$ wordt een “.” of “ \wedge ”. Dan krijgen we naar keuze

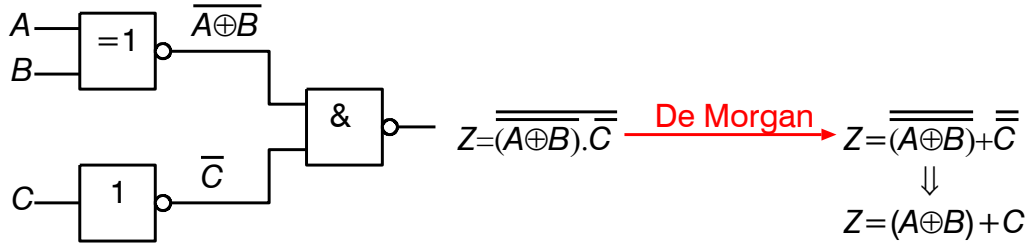
$$Y = (A + B) \cdot (\overline{C} + D) \quad \text{of} \quad Y = (A \vee B) \wedge (\overline{C} \vee D)$$

Figuur 11.5-8 toont het vervangende schema.



Figuur 11.5-8. Het omgezette schema van Figuur 11.5-7

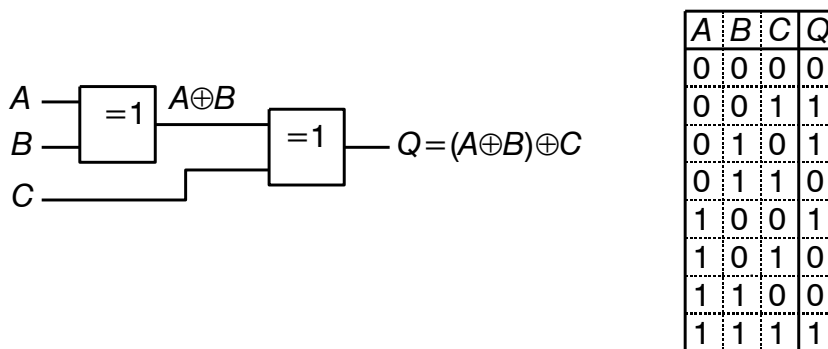
Een voorbeeld met een EXNOF-poort, vinden we in Figuur 11.5-9. Een EXOF of EXNOF behandel je vaak het gemakkelijkst als een enkele variabele, zoals u of x .



Figuur 11.5-9. Zo behandel je een EXNOF via de stellingen van De Morgan. De EXOF-uitdrukking wordt in zijn geheel behandeld als was het een enkele variabele. De NIET-streep wordt wel meegenomen.

Het is altijd toegestaan, de EXOF uit te schrijven als $A \oplus B = A \cdot \bar{B} + \bar{A} \cdot B$, maar dat is zelden handig.

EXOF-poorten kunnen ook in serie geschakeld worden zoals in Figuur 11.5-10. Dat geeft een bijzonder effect. Een waarheidstabel is daarom toegevoegd.

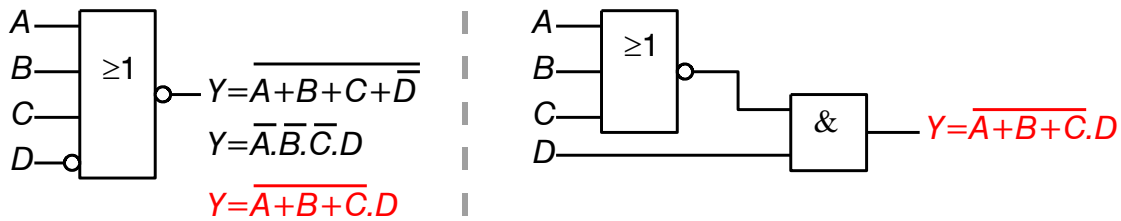


Figuur 11.5-10. Twee in serie geschakelde EXOF-poorten met waarheidstabel. Een oneven aantal keren 1 op de ingangen leidt tot 1 op de uitgang, een even aantal tot 0. Bij vier ingangen is het net andersom.

Een oneven aantal keren 1 op de drie ingangen leidt tot $Q = 1$; een even aantal keren tot $Q = 0$. Zo wordt ook de laatste regel in de waarheidstabel met drie stuks 1 op de ingangen Q geen 0, maar 1. Met vier ingangen, dus drie EXOF-poorten, is het net andersom. Probeer maar als je wilt.

11.5.6 Meer voorbeelden van combinatielogica; “spelen” met poorten

In de volgende tekst met schema's komt een aantal poortschakelingen aan de orde. Ze worden via Booleaans rekenwerk omgezet in een andere schakeling die hetzelfde doet. Er zijn massa's andere voorbeelden te bedenken. **Leer daarom de schema's niet uit het hoofd.** Probeer te begrijpen wat er gebeurt en houd het daarbij. Het is nuttige hersengymnastiek in de voorbereiding op het examen, meer niet. In elk voorbeeld staat het beginschema links en het eindschema rechts in de figuur. De uitkomst is bij beide steeds in rood aangegeven

Voorbeeld 1.


Figuur 11.5-11. Twee keer omzetten mer de stelling van De Morgan leidt tot een gelijkwaardige poortschakeling. Links van de grijze streeplijn de uitgangsschakeling, rechts het resultaat.

We passen het De Morgan-recept weer toe:

$$Y = \overline{A + B + C + \bar{D}} = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D$$

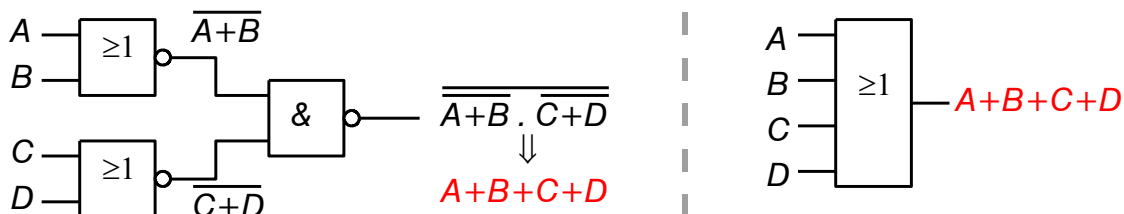
En dan passen we het recept nog een keer toe op het deel $\bar{A} \cdot \bar{B} \cdot \bar{C}$:

$$Y = \overline{\overline{A + B + C}} \cdot D$$

Dat is een nette schakeling met twee poorten die beter duidelijk maakt wat hier de bedoeling is dan het oorspronkelijke ding links in de figuur.

Voorbeeld 2.

We bespreken Figuur 11.5-12.

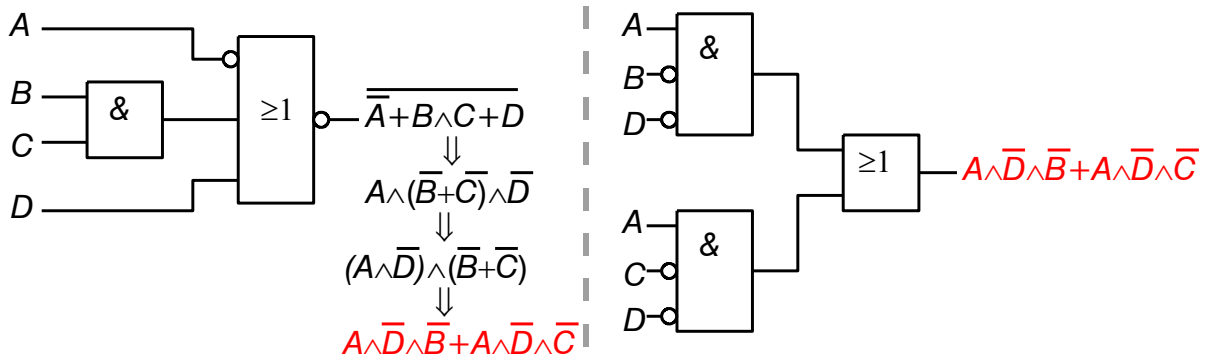


Figuur 11.5-12. De output van elke NOF-poort wordt hier als eenheid behandeld. Punt van uitgang is de schakeling links van de grijze streeplijn; de uitkomst van de berekening staat er rechts van.

De outputs van de twee NOF-poorten links worden als eenheid behandeld (dat is niet verplicht, maar in dit geval handig), net als eerder de output van een EXNOF-poort in sub-paragraaf 11.5.5. Vervang $A + B$ door x en $C + D$ door y . Dan komt er op de uitgang van de NEN-poort $\bar{x} \cdot \bar{y}$ te staan. Pas nu het rode De Morgan-recept van subparagraaf 11.4.4 toe. Laat de NIET-strepen vervallen, vervang het EN-teken door een OF en je krijgt $x + y$. Vervang x weer door $A + B$ en y door $C + D$ en ziedaar het resultaat dat in rood in Figuur 11.5-12 staat.

Voorbeeld 3.

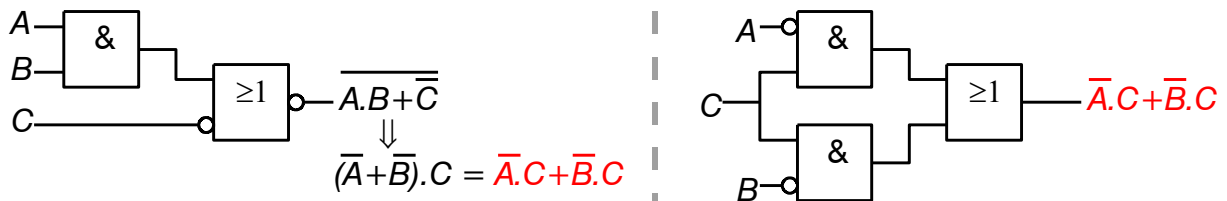
Bekijk Figuur 11.5-13.



Figuur 11.5-13. De variabelen gaan links verschillende soorten ingangen in (EN en NOF), rechts alleen maar EN. Het kan nog iets eenvoudiger (zie tekst)

De variabelen gaan nogal ingewikkeld de schakeling in (links). Let op dat A door de genegeerde ingang \bar{A} wordt. De tussenoplossing $(A \wedge \bar{D}) \wedge (\bar{B} + \bar{C})$ ziet er eenvoudiger uit dan wat er in rood staat. Bedenk voor de tussenoplossing zelf een poortenschema.

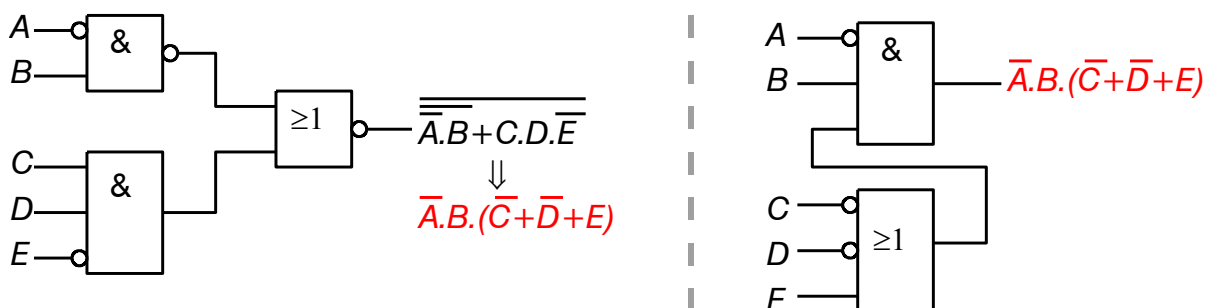
Voorbeeld 4.



Figuur 11.5-14 Links: beginsituatie; rechts: resultaat van de uitwerking. Booleaanse uitkomst in rood.

Ook deze uitwerking levert geen besparing van poorten op, maar is wel een De Morgan-toepassing. Ook zien we dat $(x + y).z = x.z + y.z$, waar x staat voor \bar{A} , y voor \bar{B} en z voor C . Net als bij vermenigvuldigen.

Voorbeeld 5.



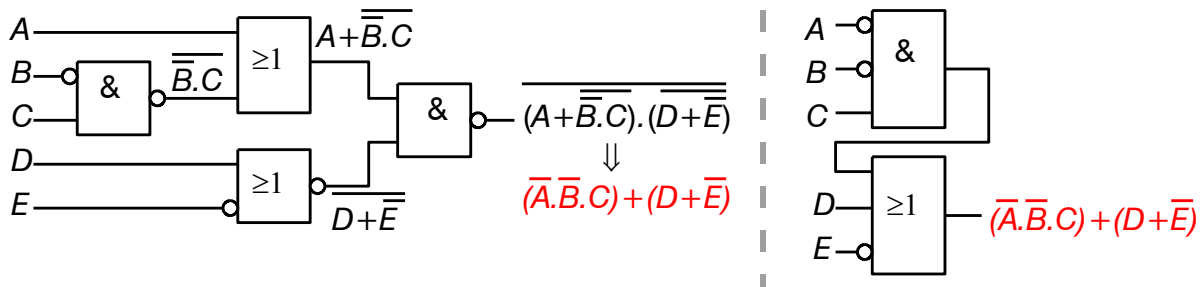
Figuur 11.5-15. Links: beginsituatie; rechts: resultaat van de uitwerking. Booleaanse uitkomst in rood.

Dit voorbeeld leidt tot een kleinere schakeling. In de praktijk kom je genegeerde poorten binnen een poort-IC niet zoveel tegen. Die worden dan gerealiseerd met een aparte inverter, die overigens meestal met z'n zessen in één IC zitten. Als er in een te realiseren schakeling inverters over zijn, kan de rechter schakeling van Figuur 11.5-15 een besparing

op soldeerwerk en onderdelen opleveren. $\bar{A} \cdot B \cdot (\bar{C} + \bar{D} + E)$ kan verder worden uitgewerkt tot $\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{D} + \bar{A}BE$. Omdat $\bar{A} \cdot B$ dan drie keer voorkomt, leidt dit tot een ingewikkelder schakeling.

Voorbeeld 6

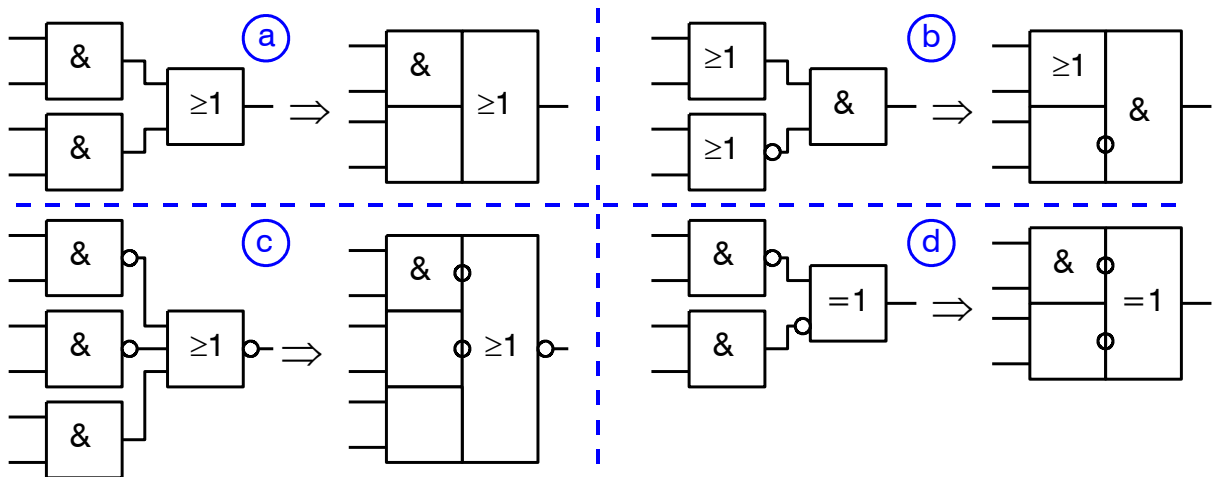
Deze omzetting begint nogal ingewikkeld en eindigt tamelijk eenvoudig.



Figuur 11.5-16. Links: beginsituatie; rechts: resultaat van de uitwerking. Booleaanse uitkomst in rood.

11.5.7 Compacte tekenwijze

We hebben een aantal combinaties van poorten gezien. De schema's kunnen met de IEC-blokken gemakkelijk worden vereenvoudigd door ze tegen elkaar aan te tekenen (probeer dat maar eens met de Amerikaanse symbolen). Dan vervalt in schema's een aantal lijnen die verbindingen tussen poorten voorstellen. Na misschien wat gewenning leidt dit tot overzichtelijker schema's. Figuur 11.5-17 laat wat voorbeelden zien.



Figuur 11.5-17. Omzetting van "normale" naar compact getekende schema's. Links van elke pijl het "normale" schema, rechts ervan de compacte variant.

Tekening a) is rechttoe-rechtaan. Twee EN-poorten die een OF-poort aansturen. De aansturende poorten mogen er ook drie of 17 of wat voor getal ook zijn (tekening c). In tekening b) zien we een OF- en een NOF-poort. Het NIET-rondje (negatierondje) komt op de grens van de OF- en de EN-kant. Het maakt in het schema niet uit of het rondje op de uitgang van de OF-poort zit of op een genegeerde ingang van de EN. Het effect is



precies hetzelfde (tekening d met een EXOF-poort rechts). In tekening c) staan twee genegeerde en één niet-genegeerde uitgang.

Aan het eind van dit hoofdstuk staat een toepassing in een wat ingewikkelder schakeling (Figuur 11.8-5). Die kom je vanzelf tegen na het doorwerken van paragrafen 11.7 en 11.8.

11.5.8 De optelschakeling

Een bijzonder geval van combinatoriële logica is de *optelschakeling*, in het Engels *full adder* genoemd. Dat is geen gisflang die pas heeft gegeten, hoewel het Engelse woord *adder* wel degelijk hetzelfde kan betekenen als het Nederlandse. Maar het Engelse *to add* betekent optellen en daarvan komt het woord *adder* voor optelschakeling. *Opteller*, zouden wij zeggen. Een *full adder* is een volledige (binaire) optelschakeling. De EXOF-poort is daar het hart van. Daarom herhalen we de waarheidstabel van de EXOF (Tabel 11.5-1).

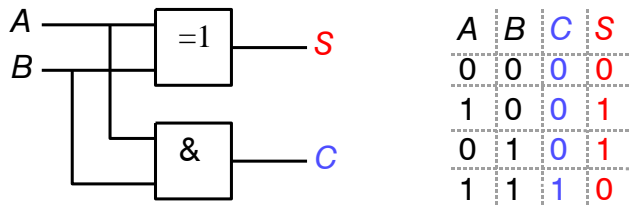
Tabel 11.5-1. Waarheidstabel voor de EXOF-poort van Figuur 11.3-12

$Q = A \cdot \bar{B} + \bar{A} \cdot B = A \oplus B$		
A	B	Q
0	0	0
1	0	1
0	1	1
1	1	0

In de kop van de tabel staat de schakeling in het Booleaans. A en B zijn de ingangen, Q de uitgang. Als we A en B willen optellen, gaat dat prima tot en met de derde regel in de tabel. Rekenkundig: $0 + 0 = 0$; $0 + 1 = 1$; $1 + 0 = 1$. Maar op de vierde regel, bij $1 + 1$, stukt het. Dan krijgen we weer een 0 als uitkomst, maar er moet “1 bewaren” bij. Daarin voorziet de EXOF niet. Decimaal gebeurt hetzelfde bij $9+1$. Daar komt 10 (decimaal) uit. Dus 0 en 1 bewaren. Van één naar twee cijfers. De 0 zal duidelijk zijn, maar waar moet in zo'n schakeling de 1 naar toe? Een plaats naar links, net als bij de decimale optelling $9 + 1 = 10$. “0 opschrijven en 1 bewaren”. Binair: $1 + 1 = 10$. Zelfde verhaal, maar tweetallig. Meestal heet die “1” *carry*, afgeleid van het werkwoord *to carry*, dragen, overbrengen. *Transportbit* en *overdrachtsbit* zijn termen voor hetzelfde. Het overbrengen moet 1 plek naar links in de optelling, want in het getal 10, of het nu decimaal of binair is, staat de 1 links van de 0. We beginnen met de schakeling die de carry vindt. Die heet *half adder*. De naam suggereert dat er iets aan ontbreekt. Dat is ook zo, maar dat komt straks.

De half-adder

Een half-adder is een schakeling die ook de carry detecteert. Dat kan met een EN-poort, want er is pas een carry als beide ingangen van de EXOF 1 zijn. Schema en waarheidstabel staan in Figuur 11.5-18.



Figuur 11.5-18. Half-adder met waarheidstabel voor de som S en de carry C .

De EXOR-schakeling levert de som S en de EN-poort eronder geeft de carry C . Die laatste heeft de waarde 1 als èn A èn B 1 zijn. Met twee bits kom je niet verder dan 3. Schakelingen moeten vrijwel altijd langere getallen aankunnen. Dat los je op met meerdere schakelingen op rij.

Dat kan niet de schakeling van Figuur 11.5-18 zijn, want die is alleen maar in staat, een carry te produceren. Hij kan er niet één van een voorgaande schakeling binnenhalen en verwerken. Vandaar de “half” van *half-adder*. Een volledige optelschakeling moet dat wel kunnen. Dat is de *full-adder*. Omdat alles zijn prijs heeft, is die ingewikkelder.

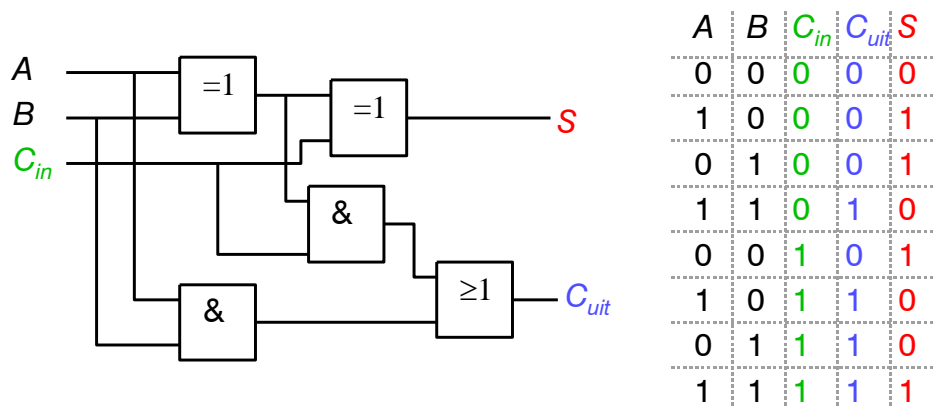
De full adder

Een full adder (Figuur 11.5-19 rechts) heeft drie inputs: twee getalinputs A en B en één voor een carry bit, C_{in} . Net als bij een half adder (Figuur 11.5-19 links) zijn er twee uitgangen, som S en een carry bit C_{uit} .



Figuur 11.5-19. Half adder en full adder in blokschema.

Het poortenschema met waarheidstabel van de full adder staat in Figuur 11.5-20.



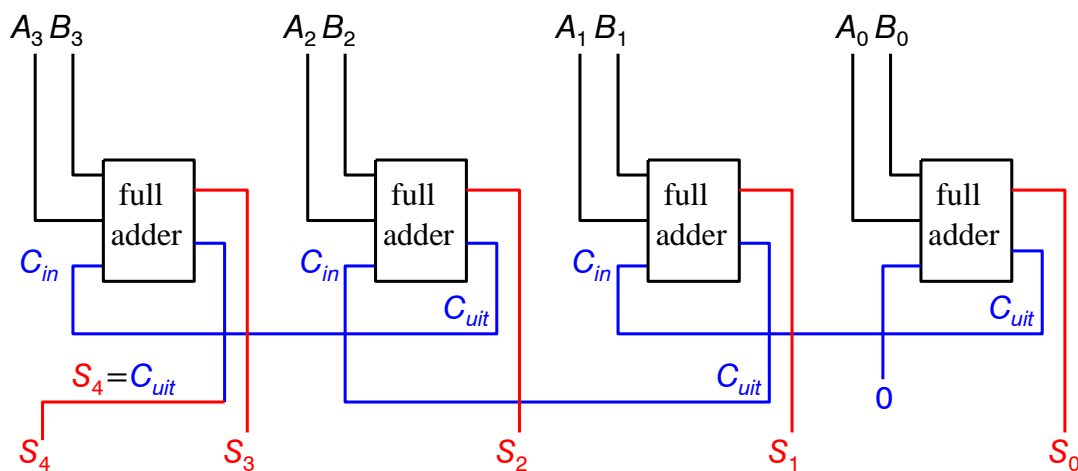
Figuur 11.5-20. Full adder met waarheidstabel. A_1 en A_2 zijn signaalinputen; C_{in} (groen) is de invoer voor de carry (transportbit), C_{uit} (blauw) de uitvoer van de carry (transportbit) en S (rood) is de uitvoer van de som.

Het schema bevat twee EXOR-poorten, twee EN-poorten en een OF-poort. Bekijken we de waarheidstabel, dan zien we voor

1. $C_{in} = 0$:
 - 1.1. 0 op de A - en B -ingang leidt tot $S = 0$ en $C_{uit} = 0$, want $0 + 0 = 0$.
 - 1.2. 0 op de A - en 1 op de B -ingang of andersom leidt tot $S = 1$ en $C_{uit} = 0$, want $1 + 0 = 1$ en $0 + 1 = 1$
 - 1.3. Een 1 op zowel A - als B -ingang leidt tot $S = 0$ en “1 bewaren”, dus $C_{uit} = 1$.
2. $C_{in} = 1$:
 - 2.1. 0 op de A - en B -ingang leidt tot $S = 1$ en $C_{uit} = 0$, want de 1 van C_{in} moet bij S worden opgeteld. Verder komt er niets bij, want $0 + 0 = 0$
 - 2.2. 0 op de A - en 1 op de B -ingang of andersom leidt tot $S = 0$ en $C_{uit} = 1$, want we hebben twee keer 1 op te tellen, net als bij punt c onder A. Of een 1 via een A - of B -ingang binnenkomt of via C_{in} is niet van belang.
 - 2.3. Een 1 op beide A - ingangen leidt tot “1 bewaren”, dus $C_{uit} = 1$. De 1 via C_{in} is de derde 1, zodat nu ook $S = 1$.

Zo worden twee getallen van 1 bit opgeteld, al of niet met carry bit. We behandelen nu een schakeling om twee getallen van 4 bits op te tellen. Daarbij moeten we bedenken dat er een verschil in richting is tussen tekst en schema's aan de ene kant en optellingen aan de andere. We zijn gewend, tekst en schema's van links naar rechts te lezen. Optellingen van twee getallen, of ze nu binair of decimaal zijn, gaan van rechts naar links.

Het minst betekenisvolle bit (Least Significant Bit, LSB) zit rechts. Daar beginnen we mee. Vervolgens werken we de berekening af van rechts naar links, waar het meest betekenisvolle bit (Most significant bit, MSB) staat. Bij het lezen van een schema van een optelschakeling moet je dit in gedachten houden. Het spel begint rechts en verplaatst zich bit voor bit naar links. In Figuur 11.5-21 worden twee getallen A en B van elk 4 bits, genummerd $A_0 \dots A_3$ en $B_0 \dots B_3$, opgeteld.

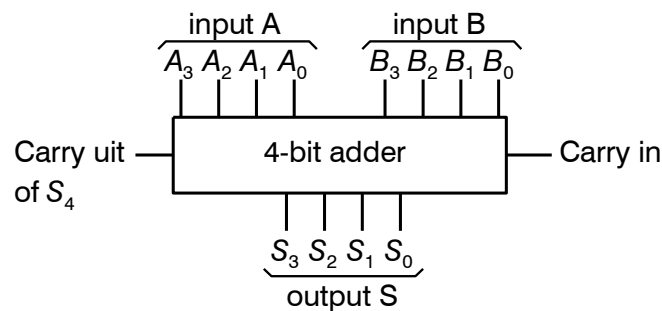


Figuur 11.5-21. Optelschakeling voor twee 4-bits getallen A en B . Sommen S in rood, carry-bits C_{in} en C_{uit} in blauw.

Bij A_0 en B_0 (rechts in het schema) is er geen input-carry. Daarom zou daar een half adder mogen staan, maar een full adder met carry input 0 is zinvoller voor het geval de schakeling in een rij van 4-bits adders wordt opgenomen. De meest rechtse adder levert de som S_0 . Het laatste carry bit van de meest linkse full adder wordt niet opgenomen door een volgende adder (die is niet ingetekend) maar wordt het vijfde som-bit (S_4 dus). Hieronder zien we een decimaal voorbeeld met twee getallen van 4 cijfers, 8792 en 1918, waaruit door optellen een getal van 5 cijfers ($S_0 \dots S_4$) ontstaat.

$$\begin{array}{r}
 \\
 \\
 + \\
 \hline
 1 \\
 S_4
 \end{array}$$

Optellers in geïntegreerde vorm (IC's) zijn vaak uitgevoerd als 4-bits opteller om in schakelingen te worden gebruikt voor (veel) grotere getallen. Zo'n schakeling kun je samenvatten in één schematekening zoals in Figuur 11.5-22.



Figuur 11.5-22. Schematische weergave van een 4-bits adder

Vanzelfsprekend mag de nummering van de bits ook de andere kant op lopen, zolang maar duidelijk is, wat wat is.

11.6 Opgaven

Op dit punt zijn we aangekomen aan het eind van de combinatielogica, een vorm van logica die niet tijdsafhankelijk is. De tegenpool is de sequentiële logica. Daarin veranderen variabelen van waarde onder invloed van een zogenoemde klokfrequentie. Voor we daaraan in de volgende paragraaf beginnen, volgen eerst opgaven over wat we in dit hoofdstuk tot nu toe hebben gehad. De layout van de tekst is zo, dat een opgave nooit over twee bladzijden is verdeeld. De uitwerkingen staan in paragraaf 11.10. Daar staat nooit meer dan één uitwerking op een bladzijde. Onder elke opgave staat een link die je naar de uitwerking brengt. Na de uitwerking kun je via een link terug naar de opgave of via een tweede link naar de volgende opgave, als die er is.

11.6.1 Opgave 11-1.

Het binaire getal 1011 wordt decimaal geschreven als

- A. 111
- B. 5
- C. 11
- D. 17

Antwoord gevonden? Naar de uitwerking

**11.6.2 Opgave 11-2.**

Het decimale getal 129 wordt binair geschreven als

- A. 1000 0001
- B. 1001 0000
- C. 1111 1111
- D. 0100 0001

Antwoord gevonden? Naar de uitwerking

**11.6.3 Opgave 11-3.**

In	In	Uit
0	0	1
1	0	0
0	1	0
1	1	0

De waarheidstabel is van een

- A. OF-poort
- B. NEN-poort
- C. EN-poort
- D. NOF-poort

Antwoord gevonden? Naar de uitwerking



11.6.4 Opgave 11-4.

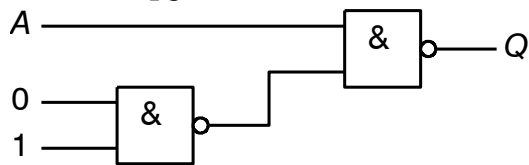
Bij de uitdrukking $\overline{A + B + C}$ hoort de waarheidstabel van een

- A. NOF-poort
- B. NEN-poort
- C. OF-poort
- D. EN-poort

Antwoord gevonden? Naar de uitwerking



11.6.5 Opgave 11-5.



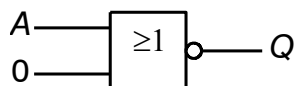
De waarde van Q is

- A. Altijd 0
- B. 1 als $A = 0$
- C. Altijd 1
- D. Onafhankelijk van A

Antwoord gevonden? Naar de uitwerking



11.6.6 Opgave 11-6.

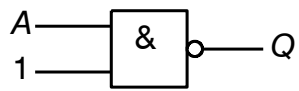


Als de 0 in de afbeelding 1 wordt en A blijft onveranderd,

- A. Blijft Q onveranderd
- B. Is $Q = 1$
- C. Is $Q = 0$
- D. Is $Q = \bar{A}$

Antwoord gevonden? Naar de uitwerking

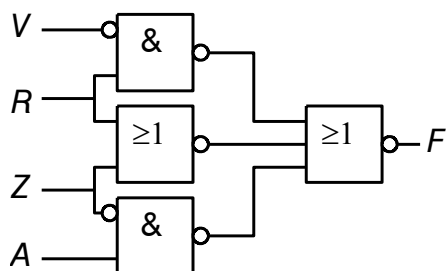


11.6.7 Opgave 11-7.

Als de 1 in de afbeelding 0 wordt en A blijft onveranderd,

- A. Blijft Q onveranderd
- B. Is $Q = 1$
- C. Is $Q = 0$
- D. Is $Q = \bar{A}$

Antwoord gevonden? Naar de uitwerking

**11.6.8 Opgave 11-8.**

Maak een waarheidstabel voor deze schakeling.

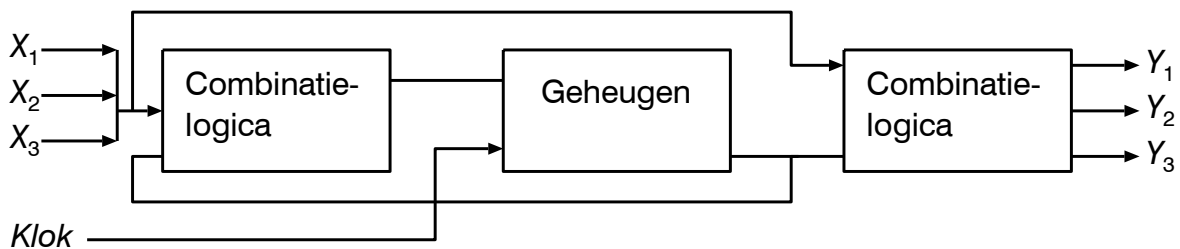
Antwoord gevonden? Naar de uitwerking



11.7 Sequentiële logica: flipflops

11.7.1 Inleiding

Bij combinatielogica wordt de toestand van de uitgangen rechtstreeks bepaald door die van de ingangen. Bij sequentiële logica wordt de toestand van de uitgangen mede bepaald door vorige toestanden aan de ingangen. Dat betekent dat in een sequentiële schakeling een geheugen moet zitten. Daarnaast komt in de schakeling meestal combinatielogica voor. Zo'n schakeling kun je in de vorm van een blokdiagram ongeveer voorstellen als dat van Figuur 11.7-1.



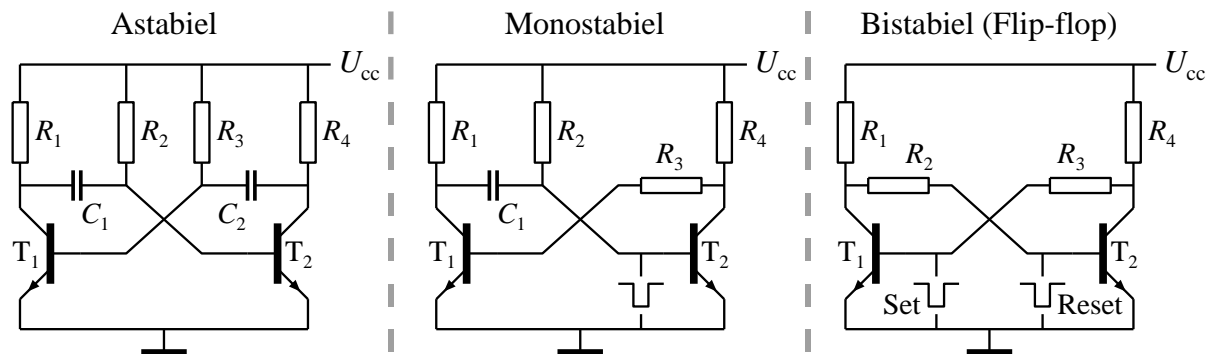
Figuur 11.7-1. Een mogelijke opzet van een schakeling met sequentiële logica. De klokpuls laat het geheugen een nieuwe toestand opnemen en/of een oude afgeven.

De gegevens $X_1 \dots X_3$ komen links de schakeling binnen. Rechts komen de resultaten $Y_1 \dots Y_3$ er weer uit. Er loopt een data-omweg om de geheugenschakeling heen. Die kan bijvoorbeeld nodig zijn als twee opeenvolgende toestanden aan de ingang moeten worden vergeleken. De klok, die een blok- of pulsvormige spanning levert, zorgt dat gegevens door het geheugen worden opgenomen en/of vrijgegeven. Hierbij is de eerste vraag, hoe zo'n geheugen dan wel in elkaar zit. Het antwoord is dat het bestaat uit flip-flops. Dat zijn schakelingen die twee stabiele toestanden hebben. We beginnen dan ook met flipflops.

11.7.2 De flipflop en zijn verwanten

Flipflops horen tot een familie van schakelingen die wordt aangeduid met de term *multivibrator* of korter *multivib*. De familie telt drie leden, elk met andere eigenschappen. Je kunt ze maken met transistoren en met poorten. We beginnen met transistoren. Het is niet nodig, de schakelingen zelf in detail te onthouden. Begrijpen moet wel. We kennen

- De *astabiele* vorm. Dit is een oscillator die blok- of pulsspanningen produceert.
- De *monostabiele* vorm heeft één stabiele en één onstabiele toestand. Wordt de schakeling in de onstabiele toestand gebracht, dan valt hij na zekere tijd terug in de stabiele toestand. Je zou de schakeling kunnen zien als een oscillator die als hij een "duwtje" krijgt, één halve periode van een blok golf, dus één puls, produceert. De schakeling heet ook wel *one shot* of *single shot*, vanwege de enkelvoudige puls.
- De *bistabiele* vorm. Die heeft twee stabiele toestanden. Alleen een puls van buitenaf kan hem van de ene in de andere toestand brengen. **Dat is de flipflop.** Noem de ene toestand 1 en de andere 0 en je hebt een binair geheugenelement.



Figuur 11.7-2. De drie soorten multivibrators naast elkaar.

Figuur 11.7-2 toont van alle drie vormen een voorbeeldschakeling. We bekijken ze stuk voor stuk.

De astabiele schakeling (links in Figuur 11.7-2).

1. Bij inschakelen komen beide transistoren nooit exact gelijktijdig in geleiding. Laten we ervan uitgaan dat T_1 eerst is. Dan zakt zijn collectorspanning tot bijna 0.
2. Via C_1 gaat de basisspanning van T_2 mee omlaag. Daardoor spert T_2 .
3. Van C_1 lekt lading weg via R_2 . De basisspanning van T_2 stijgt en T_2 gaat geleiden.
4. De collectorspanning van T_2 gaat dan snel omlaag. Via C_2 valt de basisspanning van T_1 mee terug en spert T_1 .
5. De collectorspanning van T_1 wordt hoog.
6. De lading van C_2 lekt weg via R_3 , de basisspanning van T_1 stijgt, T_1 komt weer in geleiding. Dan begint het spelletje vanaf punt 1 opnieuw. De schakeling oscilleert met het signaal op de collector van T_1 in tegenfase met dat op de collector van T_2 .

De monostabiele vorm (Figuur 11.7-2, midden)

De linkerhelft is een kopie van de linkerhelft van de astabiele vorm. In de rechterhelft ontbreekt de condensator en is de condensatorkoppeling van de collector van T_2 met de basis van T_1 vervangen door een weerstand (R_3). T_2 geleidt, doordat de basis stroom krijgt via R_2 . Dan is de collectorspanning van T_2 laag. Via R_3 is de basisspanning van T_1 ook laag en spert T_1 .

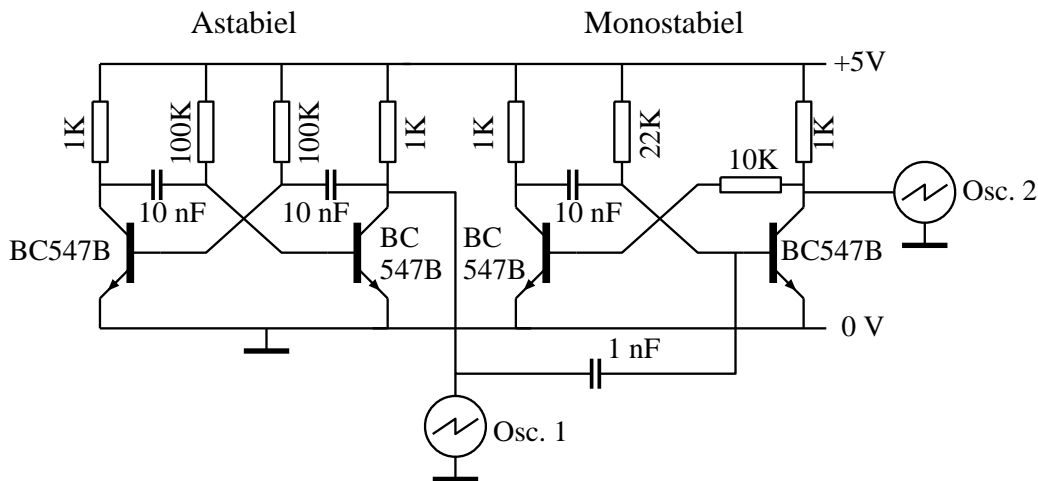
Wordt nu de basisspanning van T_2 door een (negatieve) puls tijdelijk (bijna) 0 V, dan spert T_2 en gaat de collectorspanning van T_2 omhoog. T_1 komt in geleiding, zijn collectorspanning gaat omlaag. Via C_1 wordt de spanning op de basis van T_2 laag gehouden tot de lading van C_1 voldoende is weggelekt om T_2 weer in geleiding te laten komen. Dan herstelt zich de stabiele toestand van vóór de puls.

De stabiele vorm (rechts in Figuur 11.7-2).

Als T_1 spert, is zijn collectorspanning hoog, loopt er stroom via R_2 naar de basis van T_2 en geleidt T_2 . Als T_2 spert, is zijn collectorspanning hoog, loopt er stroom via R_3 naar de basis van T_1 en geleidt T_1 . Beide situaties zijn stabiel. Komt er een negatieve puls van voldoende grootte op de basis van de geleidende transistor terecht, dan spert die en komt de andere

transistor in geleiding. In Figuur 11.7-2 zijn ze voor beide transistoren ingetekend als “Set” en “Reset”. “Set” betekent dat de betreffende schakeling “1” wordt en “Reset” dat die “0” wordt. Er is altijd een “0” en een “1” in de schakeling. Bij welke van de twee de toestand “1” of “0” wordt genoemd, is willekeurig.

Nu naar een werkende schakeling. Die zien we in Figuur 11.7-3 hieronder.



Figuur 11.7-3. De astabiele en monostabiele multivib van Figuur 11.7-2 gekoppeld en voorzien van realistische componentwaarden. De koppeling van beide schakelingen loopt via de condensator van 1 nF. Oscilloscoop aansluitingen geven aan op welke punten de scoop van Foto 11.7-1 is aangesloten.

Het astabiele deel van de schakeling (links in het schema) produceert een blokgolfachtige spanning op oscilloscoop punt 1 (Foto 11.7-1, gele grafiek).

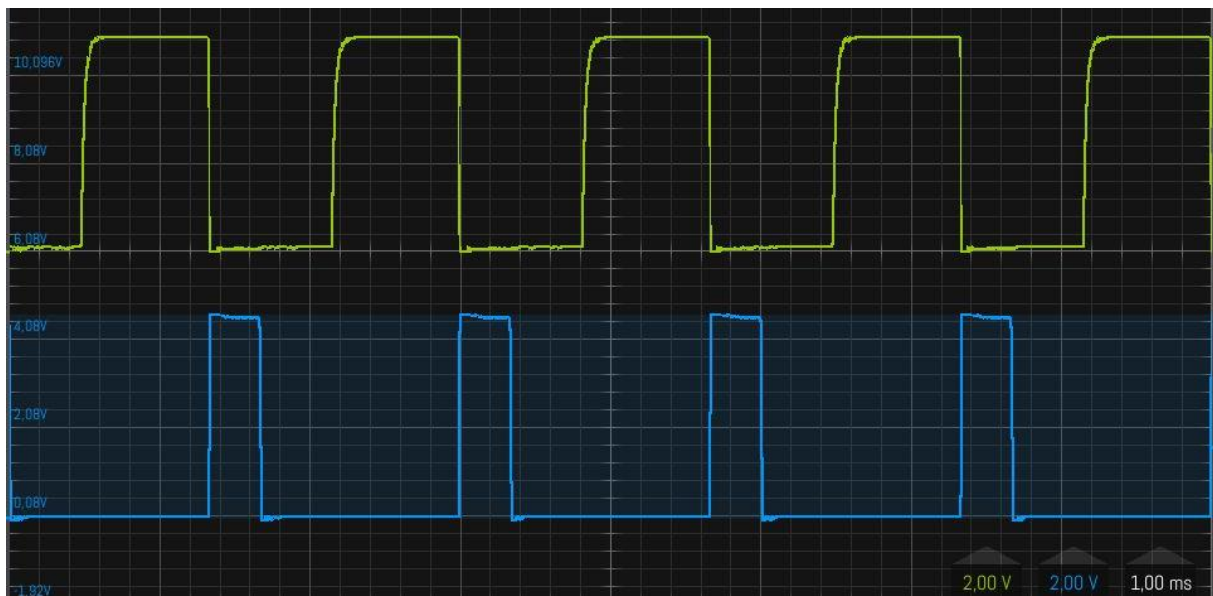


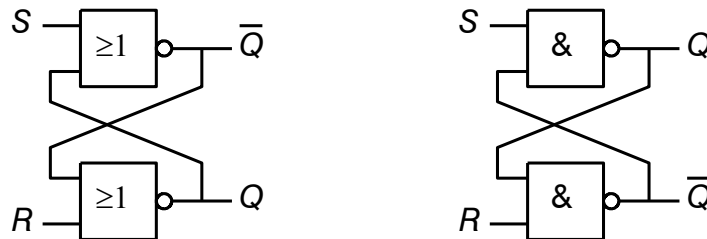
Foto 11.7-1. Geel (boven): spanning op de uitvoer van het astabiele deel en invoer van het monostabiele deel (Osc. 1 in Figuur 11.7-3). Blauw (onder): uitvoer van het monostabiele deel (Osc. 2 in Figuur 11.7-3).

Op de neergaande flank wordt het monostabiele deel (rechterhelft van het schema) getriggerd. Doordat de RC -tijd, gevormd door de weerstand van 22 K en de condensator van 10 nF kleiner is dan die in het astabiele deel (beide helften 100 K en 10 nF), duurt de puls op oscilloscoop punt 2 korter dan een halve periode van de blokgolf van het astabiele deel van de schakeling.

Combineren van het astabiele deel met het stabiele deel heeft in dit stadium niet veel zin. De linkerhelft van het astabiele deel stuurt op de negatieve flank de stabiele schakeling in de ene stand ("Set") en de rechterhelft doet hetzelfde, zodat we vrijwel een kopie van de gele grafiek op Foto 11.7-1 krijgen. Maar zo simpel blijft het niet. We gaan verschillende soorten flipflops bekijken.

11.7.3 De RS-flipflop

Eigenlijk kennen we de RS-flipflop al in de vorm van de bistabiele schakeling van Figuur 11.7-2. Hij kan ook met poorten worden gemaakt en ziet er dan op schema veel eenvoudiger uit. Dat is gezichtsbedrog, want zo'n schema toont het "binnenwerk" van de poorten niet. In Figuur 11.7-2 zag je dat wel. De S staat voor "set" en de R voor "reset". Een RS-flipflop is te maken van twee NEN- of twee NOF-poorten (Figuur 11.7-4). De waarheidstabellen staan eronder in Tabel 11.7-1.



Figuur 11.7-4. RS-flipflops. Links: gebaseerd op NOF-poorten. Rechts: gebaseerd op NEN-poorten. Waarheidstabellen in Tabel 11.7-1.

Tabel 11.7-1. Waarheidstabellen van de flipflops van Figuur 11.7-4. Links: waarheidstabel voor de RS-flipflop van NOF-poorten. Rechts: idem voor die van NEN-poorten.

R	S	Q	Niet Q	Toestand
0	0	Q	\bar{Q}	onthoud
1	0	0	1	reset
0	1	1	0	set
1	1	0	0	"verboden"

R	S	Q	Niet Q	Toestand
1	1	Q	\bar{Q}	onthoud
0	1	0	1	reset
1	0	1	0	set
0	0	1	1	"verboden"

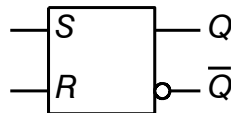
"Set" betekent dat de uitgang Q op 1 wordt gezet, bij "Reset" is dat 0. Beide flipflops hebben de kruislingse terugkoppeling die we al kennen van Figuur 11.7-2. Bij de NOF-constructie is van belang dat de uitgang van een OF- en een NOF-poort ongevoelig zijn voor veranderingen aan één ingang, zolang de andere 1 is. Bij een EN- of NEN-poort geldt hetzelfde als die andere ingang 0 is. Q en \bar{Q} zijn behalve uitgangs- ook ingangssituatie.

Daarom is bij de NEN-opbouw alles precies andersom als bij de NOF-opbouw. Vervang bij de NOF-opbouw elke 0 door 1 en elke 1 door 0 en je hebt het juiste verhaal voor de NEN-opbouw. We lopen nu de NOF-constructie langs.

Zijn R en S beide 0, dan zijn de waarden van beide uitgangen elkaars tegendeel. Ze houden hun waarde vast en blijven dat doen tot er aan de ingangen iets verandert. Het maakt niets uit of $Q = 1$ of $Q = 0$, \bar{Q} is altijd het tegendeel van Q . Als $R = 1$ en $S = 0$, dan dwingt dat tot $Q = 0$ en $\bar{Q} = 1$. (reset). Als $R = 0$ en $S = 1$, dwingt dat tot $Q = 1$ en $\bar{Q} = 0$ (set). De combinatie $R = 1$ en $S = 1$ zou moeten leiden tot $Q = 0$ en $\bar{Q} = 0$. Daarmee zijn ze niet gelijktijdig elkaars tegendeel. Die toestand is dan ook “verboden” omdat de schakeling in zijn eigen uitgangspunten verstrikt zou raken met onvoorspelbaar resultaat.

Zoals gezegd: maak van elke 0 een 1 en van elke 1 een 0, en je hebt het verhaal voor de RS-flipflop, opgebouwd uit twee NEN-poorten.

Een verkort schemasymbool voor de RS-flipflop is weergegeven in Figuur 11.7-5.



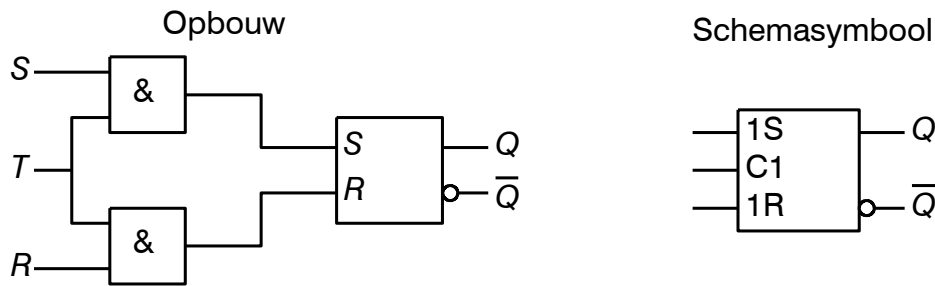
Figuur 11.7-5. Schemasympool voor een RS-flipflop.

Een toepassingsvoorbeeld is de stopknop in een lijnbus. Een passagier drukt op de stopknop als hij op de volgende halte wil uitstappen. Dat is de “Set”. De rode lamp bij de chauffeur blijft branden tot de flipflop na de haltestop een “Reset” krijgt en de rode lamp weer uitgaat. In liften vind je iets vergelijkbaars.

11.7.4 De impulsgestuurde (geklokte) RS-flipflop

Als een RS-flipflop in de “onthoud”-stand staat met $R = S = 0$ voor de NOF-gebaseerde versie of $R = S = 1$ voor een NEN-gebaseerd exemplaar, dan is het kleinste stoorsignaal op de R - of S -ingang genoeg om de flipflop in een andere stand te brengen. Met “klein” bedoelen we iets in de orde van een fractie van een milliseconde.

Om de kans daarop te verkleinen, kan men de impulsgestuurde of “geklokte” RS-flipflop toepassen. Alleen als de klokpuls, laten we hem T noemen, 1 is, kan de waarde van R en S de toestand van Q beïnvloeden. Bij aanwezigheid van stoorsignalen kunnen die de toestand van de flipflop alleen beïnvloeden als $T = 1$ of $T = 0$, al naar gelang de inwendige opbouw van de flipflop. De opbouw is weergegeven in Figuur 11.7-6, samen met het schemasymbool.



Figuur 11.7-6. Links: impulsgestuurde RS-flipflop die alleen kan schakelen als $T=1$. Rechts: schemasymbool.

De waarheidstabel bij Figuur 11.7-6 is Tabel 11.7-2

Tabel 11.7-2. Waarheidstabel bij Figuur 11.7-6.voor een geklokte RS-flipflop, opgebouwd uit NOF-poorten.

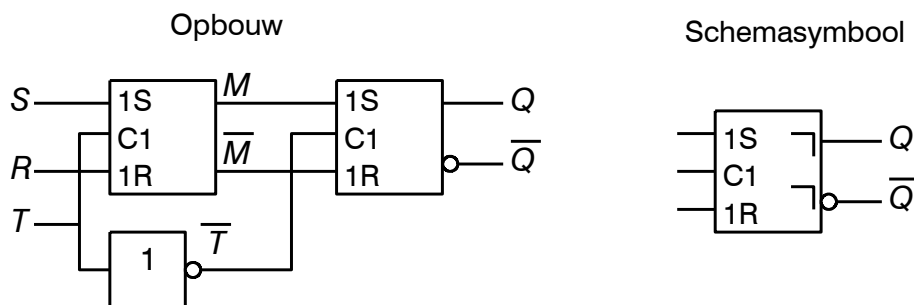
R	S	T	Q	Niet Q	Toestand
x	x	0	Q	\bar{Q}	onthoud
0	0	1	Q	\bar{Q}	onthoud
1	0	1	0	1	reset
0	1	1	1	0	set
1	1	1	0	0	“verboden”

Als $T = 0$, dan is de schakeling ongevoelig voor ongeacht welke waarde van R en S . Vandaar de twee keer “x” in de tabel, betekenis: “maakt niet uit”.

De klokpuls T kan bijvoorbeeld worden geleverd via een monostabiele schakeling die een heel smalle puls produceert waardoor de flipflop zo kort mogelijk gevoelig is voor stoorpulsen. Op Foto 11.7-1 is te zien dat de puls uit de monostabiele schakeling smaller is dan de blokgolf waardoor hij getriggerd wordt. Met een goede keuze van weerstand en condensator kan dat verschil nog veel groter worden gemaakt.

11.7.5 Het master-slave systeem

Een meester-slaaf flipflop bestaat uit twee flipflops, waarbij de tweede (de slaaf) het signaal van de eerste (de meester) overneemt. Het prinsipschema staat in Figuur 11.7-7, samen met het schemasymbool voor de hele schakeling.



Figuur 11.7-7. Master-slave RS-flipflop. Links: opbouw, rechts: schemasymbool.

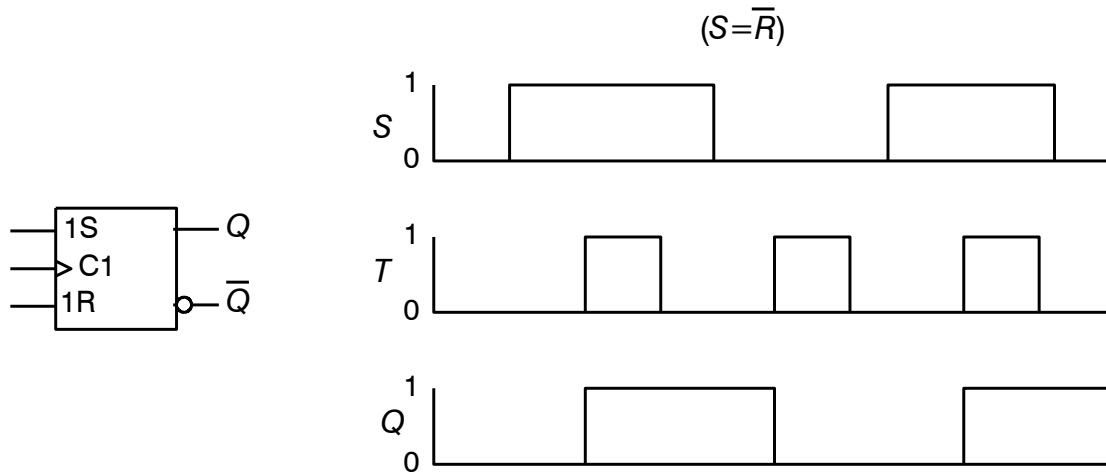
De eerste flipflop (master) neemt de toestand van R en S over als $T = 1$. De tweede flipflop (de slave) die via de NIET-poort \bar{T} ontvangt, zal de informatie op de uitgangen M en \bar{M} overnemen, zodra $T = 0$, want dan is $\bar{T} = 1$.

In het schemasymbool rechts in Figuur 11.7-7 is het master-slave principe te herkennen aan de hoekjes bij de uitgangen. Ze stellen de achterkant van de klokpuls voor. Het master-slave systeem waarborgt een nog betere storingsvrijheid dan de geklokte enkele flipflop, omdat storingen gedurende de tijd dat $T=1$ geen invloed op de toestand van de uitgangen Q en \bar{Q} hebben. De uitgangen nemen de toestand van R en S over van het ogenblik dat onmiddellijk voorafgaat aan de overgang van $T = 1$ naar $T = 0$, mits $R = \bar{S}$.

11.7.6 De flankgestuurde RS-flipflop

Zelfs bij de allerkortste klokpuls kan nog storing optreden, waarbij de flipflop niet in de gewenste toestand komt. Per klokpuls is die kans minimaal, maar als een flipflop in een systeem per minuut zo'n honderd miljoen keer van inhoud moet veranderen, kan er toch nog wel eens iets misgaan.

De flankgestuurde flipflop reageert op de opgaande of neergaande flank van een puls. Dan blijft er voor eventuele stoorsignalen bijna geen tijd meer over om een ongewenste invloed te hebben op de werking van de flipflop. Het schemasymbool staat links in Figuur 11.7-8. De flanksturing is aangegeven met het driehoekje op de klokingang.

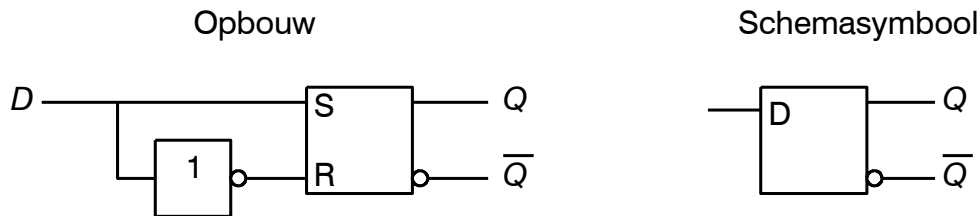


Figuur 11.7-8. Flankgestuurde RS-flipflop. Links: schemasymbool. Rechts: een voorbeeld van de gang van zaken. Een hoge positie is 1, een lage 0. Verondersteld is dat $S = \bar{R}$.

Rechts in de figuur de voorstelling van een mogelijke gang van zaken. In de figuur is verondersteld dat $S = \bar{R}$. De uitgang Q neemt de 1 van S over op de opgaande flank van de klokpuls T en blijft die houden, ook al is S veranderd van 1 naar 0. Op de opgaande flank van een nieuwe klokpuls T neemt Q de 0 van S over. Die 0 blijft staan, ook al is S alweer 1. Pas op de opgaande flank van de volgende klokpuls T wordt die 1 overgenomen door Q .

11.7.7 De D-flipflop

Als tussen de R en de S van een RS-flipflop een NIET-poort (inverter) wordt geschakeld (Figuur 11.7-9), dan is de “verboden” toestand op de ingangen onmogelijk geworden. Dat heet een D -flipflop.



Figuur 11.7-9. D -flipflop.

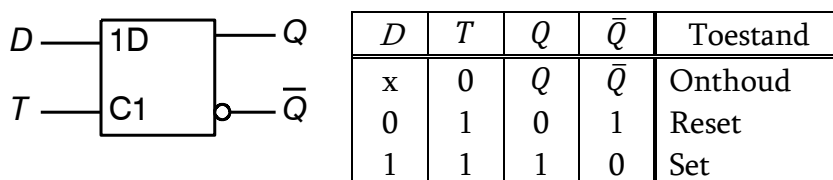
Tabel 11.7-3 toont de waarheidstabel van de D -flipflop van Figuur 11.7-9.

Tabel 11.7-3. Waarheidstabel van de D -flipflop in Figuur 11.7-9.

D	Q	\bar{Q}	Toestand
0	0	1	Reset
1	1	0	Set

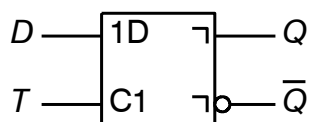
Deze tabel is weliswaar eenvoudig, maar laat ook zien dat de D -flipflop ongeschikt is om de toestand van D te onthouden. Het nut is zacht gezegd beperkt: als D verandert, verandert Q mee.

Met een geklokte D -flipflop is dat anders. Dan kan Q zich alleen aan D aanpassen tijdens een klokpuls en is er wél sprake van een geheugenfunctie (Figuur 11.7-10).



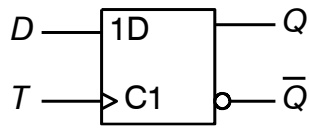
Figuur 11.7-10. Geklokte D -flipflop. Links: schemasympool. Rechts: waarheidstabel.

Een master-slave opbouw kan ook. De waarheidstabel is dezelfde als in Figuur 11.7-10. Het schemasympool is voorzien van de inmiddels bekende twee hoekjes (Figuur 11.7-11)



Figuur 11.7-11. Master-slave D -flipflop.

Het kan ook met een flankgestuurde D -flipflop (Figuur 11.7-12).



Figuur 11.7-12. Flankgestuurde D-flipflop. Waarheidstabel: Tabel 11.7-4 hieronder.

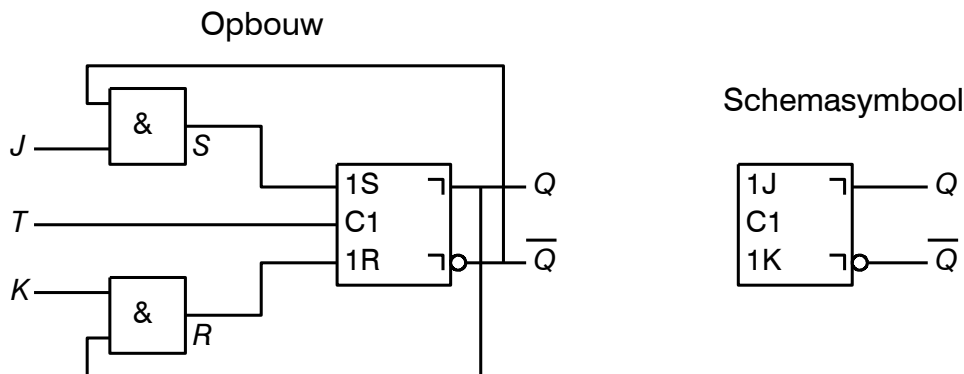
Tabel 11.7-4 Waarheidstabel bij Figuur 11.7-12

D	T	Q	\bar{Q}	Toestand
x	0	Q	\bar{Q}	Onthoud
x	1	Q	\bar{Q}	Onthoud
0	0→1	0	1	Reset
1	0→1	1	0	Set

Een geklokte D-flipflop staat bekend onder de Engelse naam *bistable latch*. De term *bistable* is bekend. *Latch* betekent slot of grendel. Die term behoeft geen nadere toelichting. Een bit gaat achter slot. Het kan worden gelezen, maar veranderen kan pas als het slot opengaat.

11.7.8 De JK-flipflop

Ook de master-slave JK-flipflop is een flipflopschakeling zonder verboden toestand. De opbouw en het schemasymbool zijn weergegeven in Figuur 11.7-13.



Figuur 11.7-13. De JK-flipflop. Links de opbouw, rechts het schemasymbool.

De JK-flipflop bestaat uit een geklokte RS-flipflop, waarin de R - en de S -ingang worden voorafgegaan door een EN-poort. Eén van de twee ingangen van de EN op de S -ingang is verbonden met de \bar{Q} -uitgang en één van de twee ingangen van de EN op de R -ingang met de Q -uitgang. De andere ingang van elke EN-poort heet de J -ingang aan de S -kant en de K -ingang aan de R -kant. Dan is er nog de klokkingang. Anders is het geen geklokte flipflop.

De uitgangen zijn dus rechtstreeks gekoppeld met de ingangen. Dat bepaalt mede de manier waarop de schakeling reageert op de toestand van de J - en de K -ingang. In de set-toestand is $Q = 1$ en $\bar{Q} = 0$, zoals bij de RS-flipflop. De toestand op de J -ingang heeft nu geen invloed, want de EN-poort op de S -ingang blokkeert. Om te resetten is behalve $Q = 1$ ook $K = 1$ nodig en een klokpuls die op dat moment 1 is.

Na de reset is $Q = 0$ en $\bar{Q} = 1$. Nu heeft de toestand op de K -ingangen geen invloed. Een set kan nu alleen worden veroorzaakt door $J = 1$ en een klokpuls (\bar{Q} is al 1). J en K kunnen zo alleen om beurten de flipflop setten, resp. resetten.

Als tegelijk $J = 1$ en $K = 1$, dan zal op het eind van elke klokpuls de toestand op de uitgangen veranderen. De uitgang die 1 was, wordt 0 en de uitgang die 0 was, wordt 1.

Samengevat zien we dat terug in de waarheidstabel (Tabel 11.7-5).

Tabel 11.7-5. Waarheidstabel van een J-K flipflop.

J	K	T	Q	Niet Q	Toestand
0	0	010	Q	\bar{Q}	Onthoud
0	1	010	0	1	Reset
1	0	010	1	0	Set
1	1	010	$Q \rightarrow \bar{Q}$	$\bar{Q} \rightarrow Q$	Omslag

Het oscillogram van Foto 11.7-2 brengt de omslag in beeld. Daar zien we dat dit leidt tot een halvering van de frequentie van de klokpuls. We hebben daarmee dubbele winst geboekt: geen “verboden” toestanden meer en een frequentiedeler!

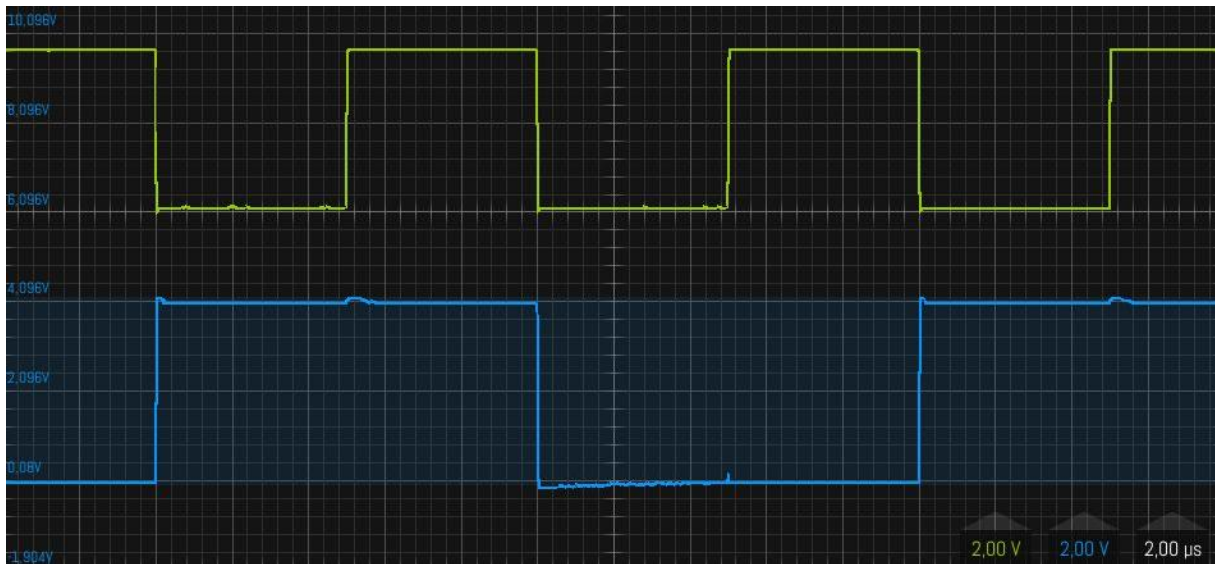
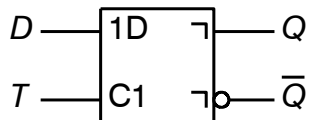


Foto 11.7-2. Frequentiedeling met een JK-flipflop bij $J=1$ en $K=1$. Geel: klok. Blauw: Spanning op uitgang Q .

Foto 11.7-2 laat ook zien dat de steile flank van de klokpuls op de Q -uitgang een klein piekje veroorzaakt, maar dat is lang niet genoeg om door een vervolgschakeling onverhoeds voor een 1 te worden aangezien.

11.7.9 De master-slave D-flipflop als frequentiedeler

Ook de master-slave D-flipflop kan als frequentiedeler dienen als de \bar{Q} -uitgang wordt verbonden met de D-ingang. We nemen in Figuur 11.7-14 het schema van de master-slave D-flipflop van Figuur 11.7-11 erbij met bijbehorende waarheidstabel.

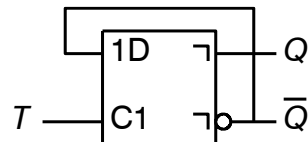


Figuur 11.7-14. De master-slave D-flipflop van Figuur 11.7-11. De waarheidstabel is Tabel 11.7-6 hieronder.

Tabel 11.7-6 Waarheidstabel bij de flipflop van Figuur 11.7-14.

D	T	Q	\bar{Q}	Toestand
x	0	Q	\bar{Q}	Onthoud
0	1	0	1	Reset
1	1	1	0	Set

En nu de als frequentiedeler geschakelde M-S D-flipflop (Figuur 11.7-15):



Figuur 11.7-15. Als frequentiedeler geschakelde Master-Slave D-flipflop.

In de figuur zien we de verbinding tussen de D-ingang en de \bar{Q} -uitgang. Dan is $D = \bar{Q}$. Op basis hiervan en de waarheidstabel in Figuur 11.7-14 maken we een tabel voor T en \bar{Q} (Tabel 11.7-7).

Tabel 11.7-7. Tabel voor de waarden van T en \bar{Q} van Figuur 11.7-15.

T	0	1	0	1	0	1	0	1	0	1	0	en zo verder
\bar{Q}	0	1	1	0	0	1	1	0	0	1	1	en zo verder

De tabel laat zien dat de frequentie van \bar{Q} en dus ook die van Q inderdaad de helft is van die van T .

11.7.10 Samengevat

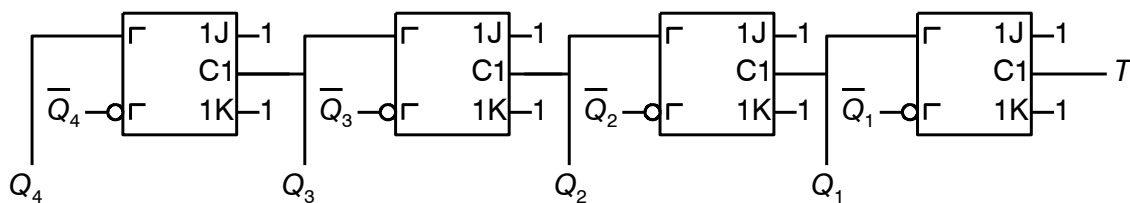
Op al deze flipflopschakelingen zijn nog talloze andere variaties te bedenken. We zullen dat niet doen, maar een samenvatting geven van de besproken flipflopvormen.

1. De RS-flipflop in zijn eenvoudigste vorm. Voordeel: eenvoud. Nadeel: storingsgevoeligheid en geen mogelijkheid om een verboden toestand op de ingangen te vermijden.
2. De geklokte RS-flipflop. Ten opzichte van de gewone RS is deze vorm minder gevoelig voor storingen, maar het bezwaar van de verboden ingangstoestand blijft.
3. De master-slave RS-flipflop heeft een verder verminderde gevoeligheid voor storingen, maar nog steeds met het nadeel van een verboden ingangstoestand.
4. De flankgestuurde RS-flipflop met heel geringe gevoeligheid voor storingen, maar alweer met verboden ingangstoestand.
5. De D-flipflop met één ingang in diverse vormen: geklokt als *bistable latch* zeer geschikt als geheugenelement en werkend volgens het master-slave beginsel ook geschikt als frequentiedeler. De D-flipflop kent géén verboden ingangstoestand, want er is maar één ingang.
6. De JK-flipflop volgens het master-slave beginsel: 2 ingangen zonder verboden toestand, geschikt als geheugenelement en als frequentiedeler. De meest veelzijdige, maar ook de meest ingewikkelde.

11.8 Frequentiedelers, tellers en registers

11.8.1 Frequentiedelers en tellers

We hebben gezien dat het mogelijk is, een master-slave JK-flipflop te gebruiken als frequentiedeler. Die deelt de frequentie door 2. Het is een koud kunstje om een reeks van zulke flipflops achter elkaar te zetten en de door 2 gedeelde frequentie aan te bieden aan de volgende flipflop. Die deelt de frequentie weer door 2, enzovoort. Zo levert de schakeling de oorspronkelijke frequentie f_T van het kloksignaal T gedeeld door $2^1, 2^2, 2^3, 2^4$, enz. Dat leidt tot de frequenties $f_T/2, f_T/4, f_T/8, f_T/16$, enz. Figuur 11.8-1 toont een 4-bits frequentiedeler met 4 JK-flipflops in serie.



Figuur 11.8-1. Asynchrone 4-bits frequentiedeler, bestaand uit 4 JK-flipflops.

De deler is *asynchroon*. Dat wil zeggen dat de flipflops, als ze van toestand veranderen, dat niet gelijktijdig doen. Elke flipflop veroorzaakt een kleine vertraging in het doorgeven van het signaal. Het effect van een klokpuls op de T -ingang verplaatst zich dan ook met een eindige snelheid door de rij flipflops. Een Engelse benaming is *ripple through*.

Foto 11.8-1 geeft van de schakeling in Figuur 11.8-1 het kloksignaal en dat van Q_1 t/m Q_4 weer. De foto toont het scherm van een zogenoemde *logic analyzer*. Dat is een apparaat waarmee verschillende toestanden in een digitale schakeling gelijktijdig zichtbaar kunnen worden gemaakt.

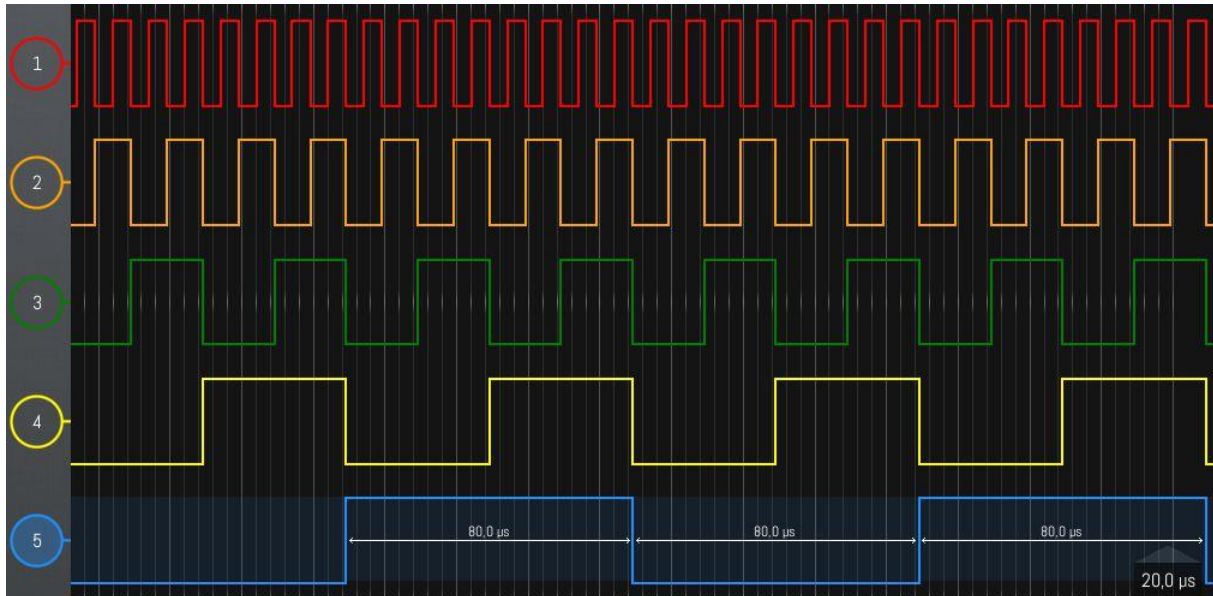


Foto 11.8-1. Golfvormen van de schakeling van Figuur 11.8-1. Spoor 1 (rood) is de klokpuls op ingang T. Spoor 2 (oranje) is Q_1 ; spoor 3 (groen) is Q_2 ; spoor 4 (geel) is Q_3 en spoor 5 (blauw) is Q_4 .

Spoor 1 bovenin (rood) is het kloksignaal op ingang T. Daaronder volgen Q_1 (spoor 2, oranje), Q_2 (spoor 3, groen), Q_3 (spoor 4, geel) en Q_4 (spoor 5, blauw). Op spoor 5 is de tijdsduur aangegeven: één periode duurt tweemaal $80\ \mu\text{s} = 160\ \mu\text{s}$. De tijdschaal is rechtsonder nog een keer aangegeven: $20,0\ \mu\text{s}$ per schaaldeel. Het blauwe signaal beslaat inderdaad $160/20 = 8$ schaaldelen per periode. De periodeduur van T is dan $1/16$ van $160\ \mu\text{s}$ is $10\ \mu\text{s}$, overeenkomend met $100\ \text{kHz}$.

Bij een synchrone deler wordt de klokpuls op ingang T via een aanvullende schakeling ook gebruikt om de flipflops gelijktijdig van inhoud te laten veranderen als de toestand van hun ingangen dat nodig maakt.

De schakeling van Figuur 11.8-1 is ook een teller. De toestand (of inhoud) van Q_1 t/m Q_4 vormt een binair getal waarvan de laagste waarde 0 is en de hoogste waarde 15. Dat verklaart ook de volgorde van de flipflops in Figuur 11.8-1: van rechts naar links, net als de manier waarop onze getallen in elkaar zitten. Merk op dat de schakeling optelt: hij begint bij 0, loopt op tot 15 (ongeveer midden in het beeld) en valt dan terug op 0 en begint opnieuw. Voor een terugteller gebruik je de \bar{Q} - uitgangen van de flipflops.

Met 5 flipflops tel je van 0 tot en met 31, met 6 van 0 t/m 63, enz. Dit soort tellers vind je allang niet meer in de vorm van losse flipflops, laat staan met losse transistoren. Het zijn praktisch altijd schakelingen, geïntegreerd op een Si-chip.

11.8.2 Delers en tellers voor andere getallen dan machten van 2

Voor ons mensen zijn binaire getallen lastige dingen om mee te werken. Vaak wordt dan de BCD-code gebruikt. BCD is een afkorting van *binary coded decimal*, tweetallig gecodeerd tientallig. Dat kan door de teller zichzelf na 9 terug te laten zetten op 0 en een volgende teller een pulsje bij te laten tellen. Dan krijg je het decimale getal 10 (of 20 of 30, enz.). Ze zitten bijvoorbeeld in digitale klokken. Daarin vind je ook zestellers, omdat er nu een maal 60 seconden in een minuut en 60 minuten in een uur zitten.

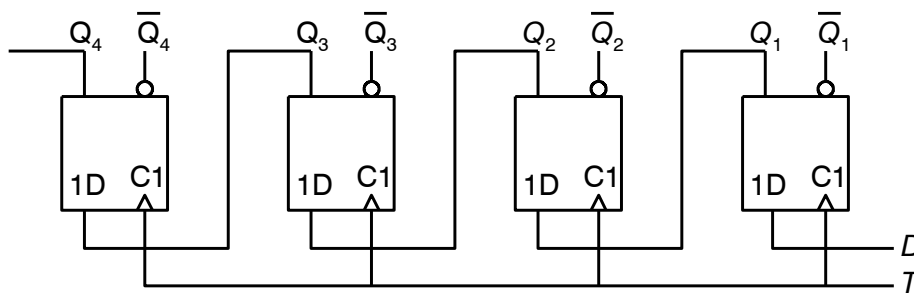
11.8.3 Schuifregisters

Serie-invoer

Tellers hebben bij elk aantal pulsen een daarbij horende combinatie van uitgangstoestanden. Schuifregisters zijn naaste familie van tellers, maar hun uitgangstoestand is niet alleen gebaseerd op een aantal pulsen, maar ook op de gegevens (data) die ze op een data-ingang worden aangeboden.

Ze schuiven bits naar elkaar door. Je zou ze kunnen vergelijken met een emmertjesbrigade waarin emmertjes met binaire inhoud, 0 of 1, van de ene flipflop naar de andere worden doorgegeven. Je kunt je daar een beeld uit vroeger tijden bij voorstellen van een rij mensen die emmertjes water aan elkaar doorgeven om een brand te blussen.

Een register is net als een teller een groep flipflops in serie. Het verschil met een teller is dat in een register de inhoud (data) via de Q -uitgang wordt doorgegeven naar de D -ingang van de volgende flipflop. De datadoorgifte vraagt om een flankgestuurde flipflop. Bij een pulsgestuurd systeem zou de data-inhoud van de eerste flipflop voldoende tijd hebben om in één keer door te rollen naar de laatste. Maar bij een register is het juist de bedoeling dat per klokpuls een bit één flipflop opschuift. Dat lukt met flanksturing. Figuur 11.8-2 toont de opzet van een 4-bits schuifregister



Figuur 11.8-2. De opbouw van een 4-bits schuifregister met D-flipflops.

Een andere en gebruikelijke manier om dat realiseren, is toepassing van master-slave flipflops. In dat geval wordt bij elke flipflop de toestand op de ingang, zoals die was bij het optreden van de voorflank van de klokpuls, pas bij het optreden van de achterflank doorgegeven naar de uitgang van de flipflop. Zo kun je een binair getal cijfer voor cijfer (bit voor bit) in een schuifregister invoeren. De flipflops nemen op de flank van elke klokpuls de toestand van de voorganger over. Zo kan op de ingang van de eerste flipflop

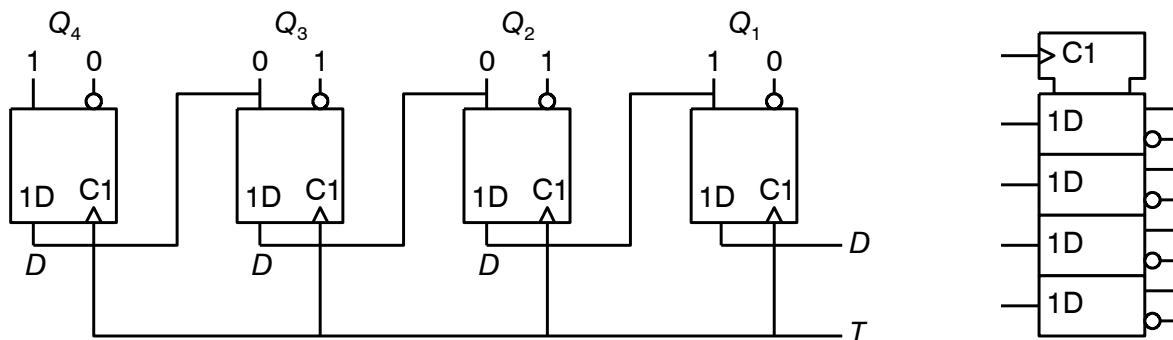
een nieuw cijfer (bit) worden ingevoerd zonder dat de informatie van vorige cijfers verloren gaat, want die is alleen maar 1 flipflop opgeschoven.

Serie-uitvoer

In dezelfde volgorde als bij de invoer komen de bits aan de achterkant van de rij flipflops weer tevoorschijn. Zo gezien lijkt een schuifregister op een etende rups, waarin het opgegeten materiaal na verloop van tijd aan de achterkant weer naar buiten komt. Omdat de schuifregisterrups niet aan spijsvertering doet, komt aan de achterkant alles er precies zo uit als het er aan de voorkant is ingegaan.

Wanneer een binair getal bit voor bit wordt verwerkt, heet dat *in serie*.

Als het schuifregister achtereenvolgens per klokpuls een 1, een 0, nog een 0 en weer een 1 binnenkrijgt, dan staat na vier klokpulsen het binaire getal 1001 in het register (Figuur 11.8-3).



Figuur 11.8-3. Links: 4-bits register met D-flipflops en binaire inhoud 1001. De niet-genegeerde Q-uitgangen zijn verbonden met de D-ingang van de volgende flipflop. In werkelijkheid zal de Q-uitgang dan via een poort of buffer naar buiten zijn uitgevoerd. Rechts: Het register in compacte tekenvorm. De klokpuls komt binnen op C1 van het zogenoemde common control block dat wordt gekenmerkt door de hamervorm.

Het eerst binnengekomen bit, in dit geval 1, staat dan op de uitgang. Na de volgende klokpuls staat er een 0, bij de daaropvolgende puls weer een 0 en vervolgens een 1. Nog een klokpuls en de inhoud van het register is volledig vernieuwd. De oorspronkelijke inhoud, het binaire getal 1001 dus, kan dan verloren zijn gegaan of elders opgeslagen. Dat hangt af van de verdere gang van zaken in de schakeling waarin het register is opgenomen en die in Figuur 11.8-3 niet ook is weergegeven.

Er bestaan ook schuifregisters waarin, afhankelijk van de status (0 of 1) van een controle-ingang de inhoud van links naar rechts of van rechts naar links kan schuiven.

Het opschuiven in een register is ook een vorm van vermenigvuldigen. Neem bijvoorbeeld het binaire getal 101; decimaal is dat 5. Schuift dat getal 1 flipflop naar links op en komt er een 0 achter, dan krijgen we 1010, decimaal 10. Vermenigvuldigd met 2 dus. Komt er in

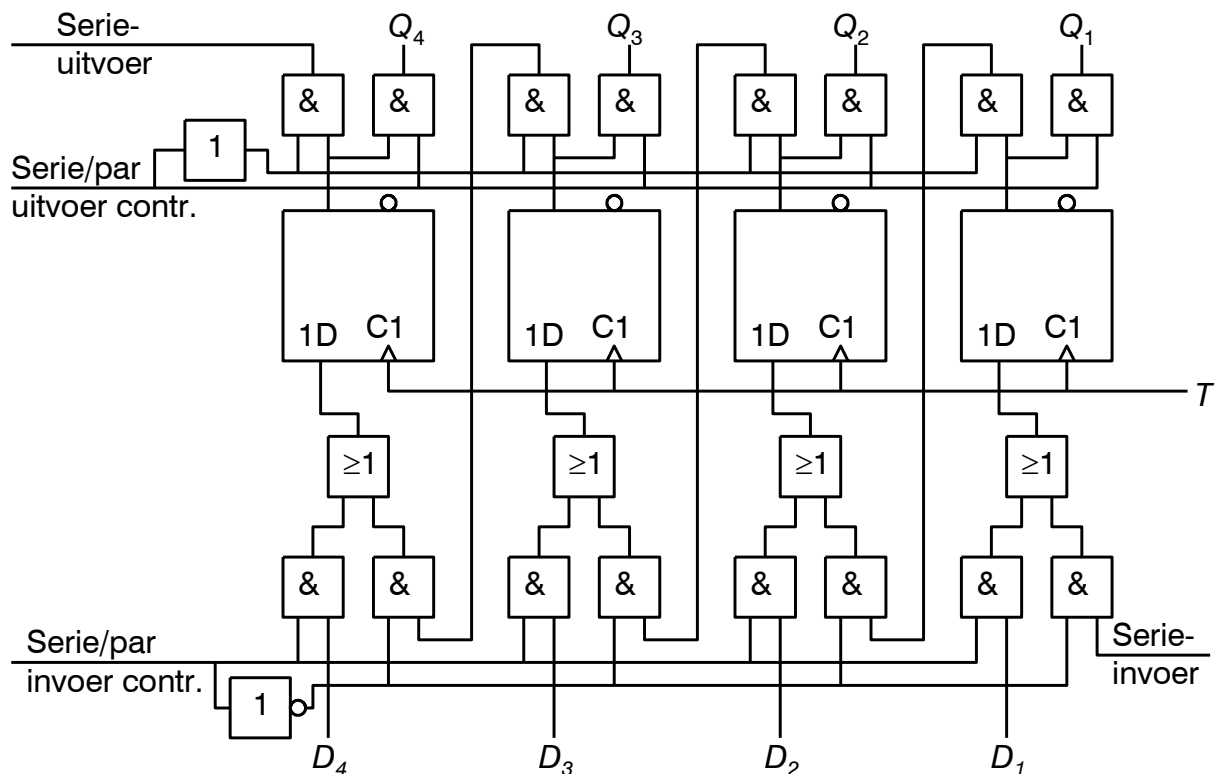
plaats van een 0 een 1 bij, dan krijgen we 11; vermenigvuldigd met 2 en de pas binnengekomen 1 bij de uitkomst opgeteld.

Schuift het register de andere kant op, dan verandert vermenigvuldigen in delen. Bedenk wel dat een eventuele laatste 1 in het binaire getal dan verloren gaat. Zonder maatregelen wordt 5:2 dus 2. Die maatregelen bestaan, maar het bespreken ervan gaat (veel) te ver voor een zendcursus.

Parallele in- en uitvoer

We kunnen met behulp van één klokpuls ook alle flipflops van een schuifregister tegelijk van een nieuwe inhoud voorzien. Dan moeten de data-ingangen (D-ingangen in Figuur 11.8-3) van buitenaf toegankelijk zijn. Daar zijn extra poorten voor nodig.

Het is ook mogelijk, de inhoud van een register in één keer naar buiten uit te voeren. Daarvoor zijn evenveel data-uitgangen nodig als er bits in het register kunnen. Ook hiervoor is een hoeveelheid extra poorten nodig. Gelijktijdige in- of uitvoer van bits in een register heet *parallel*, in tegenstelling tot de *seriële in- en -uitvoer* die we hiervoor zagen. Bij die laatste worden gegevens bit voor bit in- of uitgevoerd. Een voorbeeld van een schakeling voor serie-parallel en parallel-serieomzetting staat hieronder in Figuur 11.8-4.

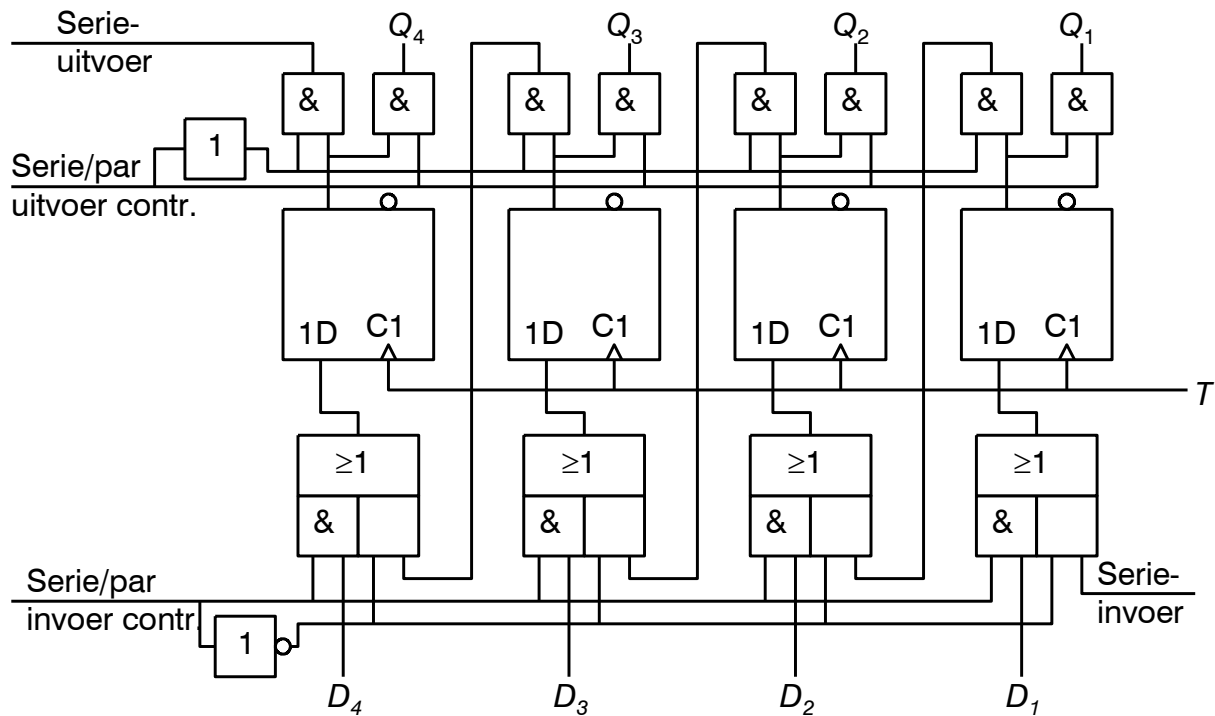


Figuur 11.8-4. 4-bits serie-parallel- en parallel-serie-omzetter.

Probeer niet, dit schema in je hoofd te proppen. Dat is nergens voor nodig.

Het is bedoeld om te laten zien dat bij omzetting tussen serie en parallel het register wordt omringd door poorten. Probeer te snappen wat er gebeurt. Dat is een nuttige oefening.

Figuur 11.8-5 geeft hetzelfde schema als Figuur 11.8-4, maar met deels de compacte tekenwijze. Ook de flipflops hadden tegen elkaar aan mogen worden getekend, maar dan hadden de drie verticale verbindingen tussen de flipflops om dat hele blok heen moeten. Dat zou aan de overzichtelijkheid juist niet hebben bijgedragen.



Figuur 11.8-5. Hetzelfde register als in Figuur 11.8-4, maar nu met het onderste poortenstelsel getekend in compacte vorm.

Het is misschien goed, je te realiseren dat alles wat je aan handelingen met getallen in je hoofd kunt bedenken, met meer of minder moeite met logische schakelingen kan worden gemaakt.

11.9 Opgaven

11.9.1 Opgave 11-9.

Om een 14-deler te kunnen maken is het minimaal benodigde aantal tweedelaars:

- A. 2
- B. 3
- C. 4
- D. 7

Antwoord gevonden? Naar de uitwerking

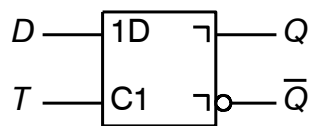


11.9.2 Opgave 11-10.

Als een 3-, een 4- en een 5-deler beschikbaar zijn, is de hoogst bereikbare deelfactor

- A. 12
- B. 60
- C. 120
- D. 0,01667

Antwoord gevonden? Naar de uitwerking

**11.9.3 Opgave 11-11.**

Dit is een

- A. Master-slave D-flipflop
- B. Flankgestuurde D-flipflop
- C. RS-flipflop
- D. Poortschakeling met genegerde en niet-genegerde uitgang.

Antwoord gevonden? Naar de uitwerking


**11.9.4 Opgave 11-12.**

Het getal 18 moet in binaire vorm in een register worden opgeslagen. Voor seriële invoer zijn a klokpulsen nodig en voor parallelinvoer b klokpulsen. Wat is juist?

- A. $a = 4, b = 1$
- B. $a = 1, b = 5$
- C. $a = 1, b = 4$




D. $a = 5, b = 1$

Antwoord gevonden? Naar de uitwerking 

11.9.5 Opgave 11-13.

In een serie-parallelregister van 8 flipflops wordt het getal 7 ingevoerd. Bedenk een zeer eenvoudige manier om dat getal na invoer met 2 te vermenigvuldigen.

Antwoord gevonden? Naar de uitwerking 



11.10 Antwoorden bij de opgaven

11.10.1 Uitwerking van Opgave 11-1.

Het binaire getal 1011 wordt decimaal geschreven als

- A. 111
- B. 5
- C. 11
- D. 17

Uitwerking

1011 binair is $1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 2 + 1 = 11$. Antwoord C dus.



Terug naar de opgave

Naar de volgende opgave



11.10.2 Uitwerking van Opgave 11-2.

Het decimale getal 129 wordt binair geschreven als

- A. 1000 0001
- B. 1001 0000
- C. 1111 1111
- D. 0100 0001

Uitwerking

Volg het recept van Figuur 11.2-2. We beginnen met het decimale getal 129

Eerst $129:2 = 64$, rest 1. Dat geeft 1 als binair cijfer rechts. Daarmee vervalt antwoord B.

Dan $64:2 = 32$, rest 0. Dat geeft 0 als tweede cijfer van rechts

Dan $32:2 = 16$, rest 0. Dat geeft 0 als derde cijfer van rechts

Dan $16:2 = 8$, rest 0. Dat geeft 0 als vierde cijfer van rechts

Dan $8:2 = 4$, rest 0. Dat geeft 0 als vijfde cijfer van rechts

Dan $4:2 = 2$, rest 0. Dat geeft 0 als zesde cijfer van rechts

Dan $2:2 = 1$, rest 0. Dat geeft 0 als zevende cijfer van rechts

Dan $1:2 = 0$, rest 1. Dat geeft 1 als achtste cijfer van rechts

Uitkomst: acht cijfers met 1 als eerste en laatste cijfer en de rest nullen: 1000 0001 en dat is antwoord A.

Opmerking

Je kunt de vier antwoorden ook stuk voor stuk proberen. Omdat je bij A meteen al “raak schiet” is dat een kortere weg, maar dat weet je niet van te voren.



Terug naar de opgave

Naar de volgende opgave



11.10.3 Uitwerking van Opgave 11-3.

In	In	Uit
0	0	1
1	0	0
0	1	0
1	1	0

De waarheidstabel is van een

- A. OF-poort
- B. NEN-poort
- C. EN-poort
- D. **NOF-poort**

Uitwerking

Als een 0 op beide ingangen een 1 op de uitgang geeft, gaat het in elk geval om een poort met NIET. Daarmee houden we alleen antwoorden A en C over. De waarheidstabel geeft aan dat het niet uitmaakt of er één of twee keer een 1 op de ingangen staat. Dat kan alleen bij een OF en een NOF-poort. De OF was al afgevallen, dus blijft de NOF over. Antwoord D is daarom het enige goede.



Terug naar de opgave

Naar de volgende opgave





11.10.4 Uitwerking van Opgave 11-4.

Bij de uitdrukking $\overline{A + B + C}$ hoort de waarheidstabel van een

- A. NOF-poort
- B. NEN-poort
- C. OF-poort
- D. EN-poort

Uitwerking

Vereenvoudiging volgens De Morgan houdt in: OF wisselt met EN en omgekeerd; waar een NIET-streep staat verdwijnt deze; waar geen NIET-streep staat, komt er één. Voor dit laatste is geen “klandizie”, dus alle NIET-strepen verdwijnen. Dat leidt tot $A.B.C \equiv A \wedge B \wedge C$. Dat is een EN-poort, antwoord D.

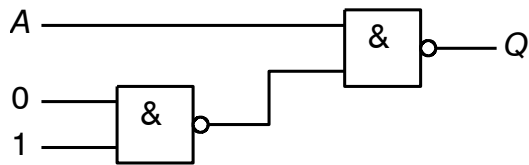


Terug naar de opgave

Naar de volgende opgave



11.10.5 Uitwerking van Opgave 11-5.



De waarde van Q is

- A. Altijd 0
- B. 1 als $A = 0$**
- C. Altijd 1
- D. Onafhankelijk van A

Uitwerking

De figuur toont twee NEN-poorten. De onderste heeft twee ingangen, één is 0, de ander 1. Bij een EN-poort levert dat een 0 op de uitgang. Bij een NEN wordt die 0 een 1.

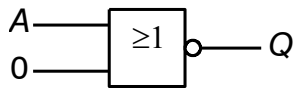
Bij de bovenste NEN-poort staat op de onderste ingang diezelfde 1. Q is daarmee afhankelijk van A . Als $A=1$, dan geldt $Q=0$, want ook hier inverteert de uitgang de waarde. Als $A=0$, dan is om dezelfde reden $Q=1$. Dat leidt tot antwoord B.



Terug naar de opgave

Naar de volgende opgave



11.10.6 Uitwerking van Opgave 11-6.

Als de 0 in de afbeelding 1 wordt en A blijft onveranderd,

- A. Blijft Q onveranderd
- B. Is $Q = 1$
- C. Is $Q = 0$
- D. Is $Q = \bar{A}$

Uitwerking

Bij een NOF-poort is de uitgang 0, zodra 1 of meer ingangen 1 zijn (**goed om dit te onthouden**). De waarde op andere ingangen doet er dan niet meer toe. Antwoord C is daarom goed.

Als A 1 was, was Q al 0, maar dat wordt niet gevraagd. Als de 0 niet verandert, is inderdaad $Q = \bar{A}$, maar de 0 werd nu juist wèl 1. Datzelfde geldt voor antwoord D dat onder deze omstandigheden ook fout is.

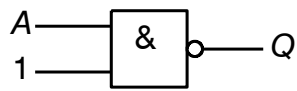


Terug naar de opgave

Naar de volgende opgave



11.10.7 Uitwerking van Opgave 11-7.



Als de 1 in de afbeelding 0 wordt en A blijft onveranderd,

- A. Blijft Q onveranderd
- B. Is $Q = 1$**
- C. Is $Q = 0$
- D. Is $Q = \bar{A}$

Uitwerking

Bij een NEN-poort is de uitgang 1, zodra 1 of meer ingangen 0 zijn (**goed om dit te onthouden**). De waarde op andere ingangen doet er dan niet meer toe. Als de ingang met 1 in de afbeelding 0 wordt, is $Q = 1$, ongeacht wat er op de andere ingang staat. Antwoord B is dan ook het enige goede.

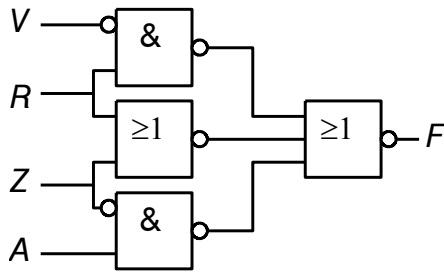


Terug naar de opgave

Naar de volgende opgave



11.10.8 Uitwerking van Opgave 11-8.



Maak een waarheidstabel voor deze schakeling.

Uitwerking

Met vier binaire variabelen heb je $2^4 = 16$ mogelijke combinaties. De tabel is met de kop mee $16+1$ is 17 regels lang Vul eerst de waarheidstabel met alle mogelijke combinaties van invoerwaarden zoals weergegeven. Bedenk dat een 0 op één ingang van een NEN-poort altijd een 1 op de uitgang veroorzaakt en een 1 op één ingang van een NOF-poort altijd een 0 op de uitgang.

$R = 0$ leidt tot op de uitgang van de bovenste NEN-poort tot een 1. De NOF-poort rechts geeft dan $F = 0$, ongeacht waar er op de andere twee ingangen staat. Dus $R = 0 \rightarrow F = 0$ Dat is 8 keer 0 voor F (zwarte cijfers).

$A = 0$ maakt de uitgang van de onderste NEN 1: $A = 0 \rightarrow F = 0$. Vier **rode** cijfers.

$V = 1$ maakt de uitgang van de bovenste NEN 1: $V = 1 \rightarrow F = 0$. **Blauwe** cijfers.

Na al deze nullen zijn er nog twee rijen over. Om $F = 1$ te krijgen, moeten alle drie ingangen van de rechtse NOF-poort 0 zijn. We beginnen met de bovenste ongevulde rij. Die heeft $V = Z = 0$ en $R = A = 1$. Dat leidt tot 0 op de uitgangen van alle drie poorten links en dus $F = 1$. De eerste 1 (**groen**). Voor $V = 0$, rest=1 is de uitgang van de onderste NEN gelijk aan 1, dus dat wordt weer $F = 0$. Ook een **groen** cijfer.

V	R	Z	A	F	V	R	Z	A	F
0	0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	1	0
0	0	1	0	0	1	0	1	0	0
0	0	1	1	0	1	0	1	1	0
0	1	0	0	0	1	1	0	0	0
0	1	0	1	1	1	1	0	1	0
0	1	1	0	0	1	1	1	0	0
0	1	1	1	0	1	1	1	1	0

Dit is dus een poortsysteem dat maar bij één combinatie een 1 geeft en bij de rest een 0.



Terug naar de opgave



11.10.9 Uitwerking van Opgave 11-9.

Om een 14-deler te kunnen maken is het minimaal benodigde aantal tweedelers:

- A. 2
- B. 3
- C. 4
- D. 7

Uitwerking

Het deeltal bepaalt het benodigde aantal delers. Dat aantal is gelijk aan de exponent van de eerste gehele macht van 2 die groter of gelijk is aan het deeltal. Met drie flipflops kun je hoogstens door $2^3 = 8$ delen, met vier stuks hoogstens door $2^4 = 16$. Dat betekent in dit geval een schakeling van 4 flipflops, voorzien van een poortschakeling die ervoor zorgt dat na stand 13 de teller weer op 0 komt. Antwoord C is dus goed. Met 7 flipflops heb je er meer dan **minimaal** nodig is.



Terug naar de opgave

Naar de volgende opgave





11.10.10 Uitwerking van Opgave 11-10.

Als een 3-, een 4- en een 5-deler beschikbaar zijn, is de hoogst bereikbare deelfactor

- A. 12
- B. 60**
- C. 120
- D. 0,01667

Uitwerking

Met een 3-, een 4- en een 5-deler kun je maximaal delen door $3 * 4 * 5 = 60$. Dat betekent antwoord B.

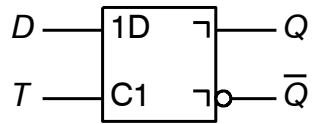


Terug naar de opgave

Naar de volgende opgave



11.10.11 Uitwerking van Opgave 11-11.



Dit is een

- A. Master-slave D-flipflop
- B. Flankgestuurde D-flipflop
- C. RS-flipflop
- D. Poortschakeling met genegeerde en niet-genegeerde uitgang.

Uitwerking

De ingang 1D verraadt dat dit een D-flipflop is. De ingang C1 geeft aan dat het om een geklokt exemplaar gaat. Omdat bij de C1-ingang het driehoekje ontbreekt, kan het geen flankgestuurde flipflop zijn. Dan blijft antwoord A over. Dat wordt bevestigd door de hoekjes bij de uitgangen die een master-slave flipflop aangeven. Dubbel bewijs voor antwoord A.



Terug naar de opgave

Naar de volgende opgave



**11.10.12 Uitwerking van Opgave 11-12.**

Het getal 18 moet in binaire vorm in een register worden opgeslagen. Voor seriële invoer zijn a klokpulsen nodig en voor parallelinvoer b klokpulsen. Wat is juist?

- A. $a = 4, b = 1$
- B. $a = 1, b = 5$
- C. $a = 1, b = 4$
- D. $a = 5, b = 1$

Uitwerking

18 ligt tussen $2^4 = 16$ en $2^5 = 32$. Dat betekent dat 4 flipflops niet genoeg zijn en er 5 nodig zijn om het getal op te slaan. Dan zijn voor seriële invoer 5 pulsen nodig. Dan geldt dat $a = 5$. Dat alleen al betekent dat het antwoord D moet zijn. Een slimme examenkandidaat controleert toch nog even of $b = 1$. Je kunt tenslotte een denkfout hebben gemaakt. Maar voor parallelinvoer heb je inderdaad maar 1 puls nodig, dus $b = 1$ en antwoord D is met dubbel bewijs goed.



Terug naar de opgave

Naar de volgende opgave





11.10.13 Uitwerking van Opgave 11-13.

In een serie-parallelregister van 8 flipflops wordt het getal 7 ingevoerd. Bedenk een zeer eenvoudige manier om dat getal na invoer met 2 te vermenigvuldigen

Uitwerking

De simpelste methode is het getal 1 plaats naar voren te schuiven. Daarvoor is 1 klokpuls voldoende.



Terug naar de opgave