

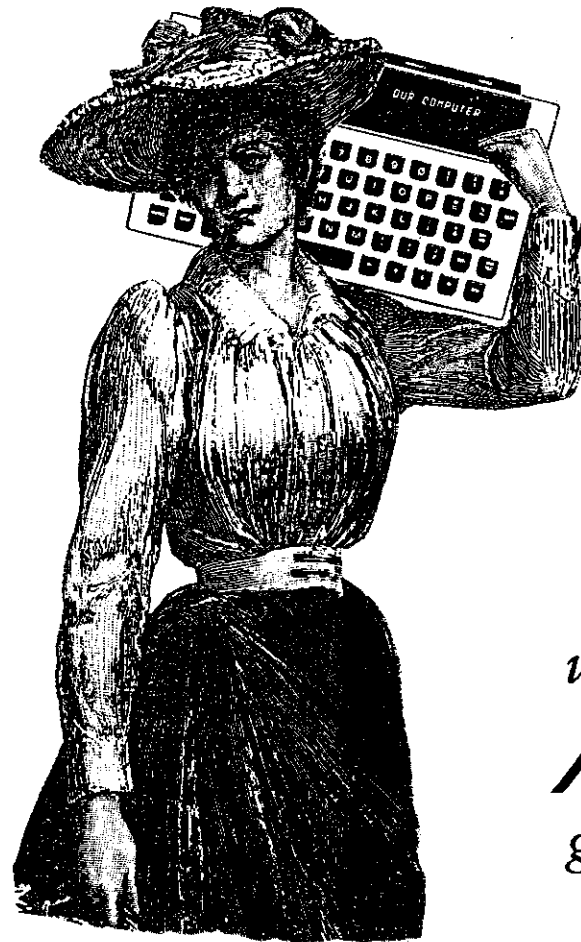


welcome to the  
wonderful world of

**B+**

*NewBrain-*  
gebruikersgroep

*NewBrain*  
**on-line**



**28**

oktober 2000

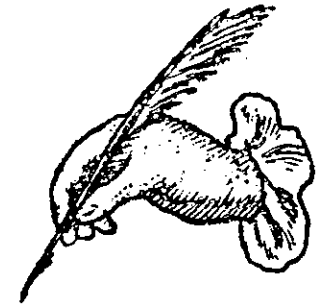
uitgave van de  
*NewBrain-*  
gebruikersgroep

# New Brain on-line

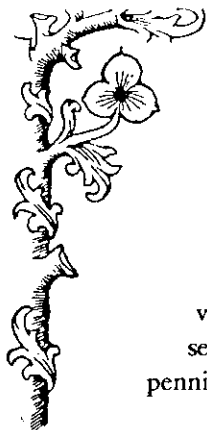
## ten geleide

deze achtentwintigste uitgave van newbrain on-line is een novum: goed gevuld met gevarieerde artikelen, die echter allemaal, ja u leest het goed, *allemaal* afkomstig zijn van dezelfde auteur: dré jansen. nu zijn wij hem daar heel dankbaar voor, maar wij zijn met hem van mening, dat het nog beter zou zijn, als ook anderen bereid zijn om hun ervaringen en gezichtspunten te delen met hun clubgenoten

dré jansen legt in een vierdelig artikel de principes uit van de rekenen zoals dat door de processor van de computer gedaan wordt. maar voordat u daaraan toe bent, hebt u gelezen hoe de duimwielchakelaar nou eigenlijk werkt en hoe je met fet's het toerental van kleine motoren regelt, en hebt u ontdekt dat deze newbrain on-line het definitieve naslagwerk is voor weerstanden



menno stevens



**New-Brain-**  
**gebruikersgroep**  
**postbus 94494**  
**1090 GL amsterdam**



voorzitter: jan wubben, (010) 4557698  
secretaris: maarten floor, (020) 4964374  
penningmeester: menno stevens, (020) 6924137  
dré jansen, (0174) 414199  
albert stuurman, (030) 2280163

postgiro 2505800 tnv hcc newbrain-gebruikersgroep

de newbrain-gebruikersgroep is een onderdeel van de  
hcc hobby computer club  
de molen 24, 3994 DB houten  
inschrijvingsnummer kvk utrecht 82311

website <http://www.hobby.nl/~newbrain-gg/>

landelijke newbraindagen 21 april en 20 oktober 2001  
in het clubhuis van de afdeling gouda  
nonnenwater 8, 2801 VA gouda

#### newbrain on-line

redactie: menno stevens, (020) 6924137  
kopij voor nummer 29 graag naar het adres van de gebruikersgroep  
of per e-mail aan [mennostevens@hetnet.nl](mailto:mennostevens@hetnet.nl)  
geplaatste artikelen mogen alleen voor niet-commerciële doeleinden  
en onder bronvermelding worden overgenomen

# duimwiel

## uitlezing van een duimwielschakelaar

Men heeft twee numerieke duimwielschakelaars, met elk 10 standen (er bestaan namelijk ook hexadecimale duimwielschakelaars met 16 standen). De meest gebruikte controllers kennen een 8-bits structuur, waarmee 255 mogelijkheden kunnen worden onderscheiden. De toegepaste spanning is gewoonlijk 5 V, waardoor er stapjes van 20 mV te onderscheiden zijn. Kleinere stapjes kunnen niet worden onderscheiden!

Zaak is wel dat je dan gebruik maakt van een zeer stabiele voeding. Ik ga uit van twee decimale duimwielschakelaars, dus hoeft ik maar 100 stapjes te onderscheiden..... *Nee!* Ik moet 100 stappen onderscheiden, maar de controller moet wel 255 stappen onderscheiden. Er zijn namelijk een heleboel stappen die niet worden gebruikt.

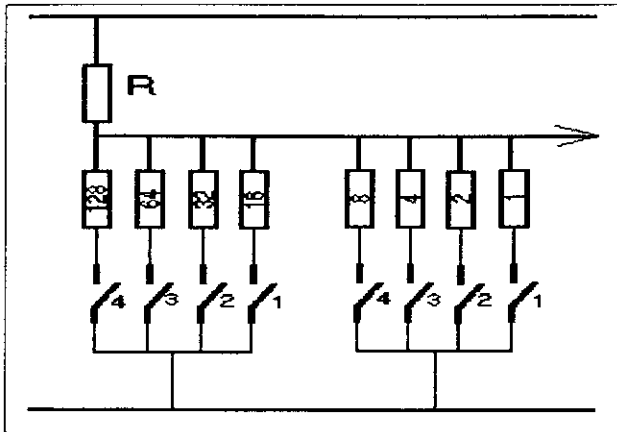
Nu kan je op twee manieren te werk gaan. Per poort decodeer je één schakelaar, of je werkt nauwkeurig en werkt met twee schakelaars per poort.

De persoon waarmee ik sprak en vervolgens na aansporing door een ander tot het schrijven van dit stukje kwam, had slechts 42 stappen nodig. Dan praat je over anderhalve duimwielschakelaar, ofwel 6 bits. Wat neer komt op 64 stapjes. Er is een niet gebruikt deel, namelijk van 0A tot en met 0F. De stapjes zijn nu geen 20 mV maar 80 mV. Deze zijn veel makkelijker te onderscheiden. Bovendien stelt dit minder strenge eisen aan de voeding.

Nu eerst wat vertellen over duimwielschakelaars, want niet iedereen weet

wat daarmee bedoeld wordt. Welnu, u heeft vast wel eens een schakelaar bediend, die meerdere standen kent. De keuzeschakelaar van uw universeelmeeter is een voorbeeld hiervan. Deze keuzeschakelaar is geen duimwielenschakelaar, maar het geeft wel een idee van de functie. Een klein wieltje dat met de duim bediend wordt, of een klein schakelaartje waarbij je een klein wieltje draait, waarmee middels cijfers de stand kan worden uitgelezen.

Het verschil is, dat nu niet om beurten het moedercontact met de afgaande contacten verbonden wordt, maar middels de binaire tellingsmethode wordt verbonden. Nadere uitleg over deze telling is te vinden in een andere uitgave van het *Robotijte*, NewBrain on-line, *Bitje Randstad*, en nog vele andere bladen.



- Stand 0 geeft geen verbinding
- Stand 1 sluit alleen contact 1
- Stand 2 sluit contact 2
- Stand 3 sluit contacten 1 en 2, enzovoorts

Wanneer je nu weerstanden aansluit die zich met elkaar verhouden als de waarden die op de tekening in de weerstanden staan vermeld, dan is de telling correct. Twee digits geven de waarde aan. Het LSB (minst belangrij-

ke bit) is de rechter schakelaar, en het MSB is de linker schakelaar (meest belangrijke bit).

Nu moet nog de spanning worden bepaald. Dat verloopt volgens de *wet van Ohm*. De spanning over de weerstanden, R en het samenspel van 8 weerstanden aan de schakelaars, geeft een spanning op de pijlpunt.

Wanneer weerstand 1 wordt kortgesloten, dan is de laagste waarde 0 V. Stand 0 geeft de hoogste waarde, namelijk 5 V. Wanneer we uitsluitend de waarden 1 tot 99 willen herkennen, dan moeten we softwarematig de waarden 0 V en 5 V als illegaal opgeven. Softwarematig geen probleem. Daarbij moeten we dus weerstand 1 een waarde geven, bijvoorbeeld 1 K. Weerstand 2 wordt dan 2 K, weerstand 4 krijgt de waarde 4 K, enzovoort. Natuurlijk zijn die waarden niet allemaal exact te vinden, dan moet u de waarden kiezen die er het dichtst bij liggen, of middels twee weerstanden zelf benaderen.

Wanneer u geen 100 stappen nodig heeft, moet u minder stappen kiezen. De reden is, dat er dan minder nauwkeurig gewerkt hoeft te worden. Er hoeven softwarematig minder waarden te worden vergeleken, dus uw programma werkt sneller.

De maximale stroom vloeit bij de kleinste weerstandswaarde. Dus wanneer alle weerstanden 'in bedrijf' zijn. Gaan we uit van de kleinste waarde die gelijk is aan 1 K, dan is de weerstand 500  $\Omega$ , wanneer alle weerstanden ingeschakeld zijn. Wanneer we een stroombron als voeding gebruiken, dan kunnen we van de grootst mogelijke spanningsval gebruik maken.

We hadden bedacht dat er zo weinig mogelijk hardware aanwezig mocht zijn. Dan kiezen we voor weerstand R een waarde van 10 K. In het ergste geval loopt er een stroom van nog geen halve mA. De hoogste spanning is dan 4,64 V, de laagste spanning 0,25 V. De 'verboden' waarden van 0 V en 5 V zijn duidelijk lager en hoger.

Een ander probleempje doet zich voor. Wanneer je de maximale spanningsval op de maximale stappen deelt, dan blijkt elk stapje 17,3 mV groot te zijn. Dat is te klein, want aan het begin hadden we besloten dat we mini-

maal stappen van 20 mV moeten maken. Welnu, hiermee is aangetoond dat deze methode wel leuk is, maar niet op dergelijke schaal kan worden toegepast. Voor degene die 42 stappen nodig had, is het wel een goede methode, want dan zijn er maar 64 verschillende stappen van 80 mV te onderscheiden.

Op deze manier is een eenvoudige keuzeschakelaar te bouwen: twee duimwielchakelaars en een handje weerstanden zijn voldoende.

## software

De software is een verhaal apart. Nemen we een controller met een ingebouwde A/D-converter, dan voeren we de spanning in, en bepalen bij welke uiterste waarden naar welke routine gesprongen moet worden. Nooit één waarde kiezen, want bij de minste spanningsvariatie kom je niet meer op die waarde en dan spring je niet naar de gewenste routine.

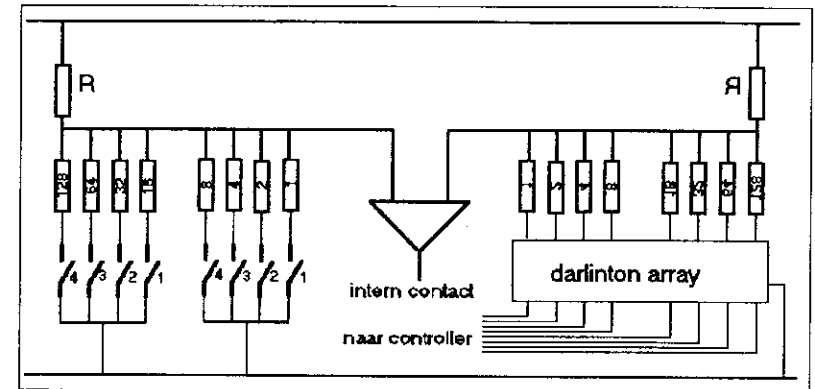
Met een universeelmeter meet je de waarden, vervolgens kies je een bereik met waarden dat 40 mV hoger ligt, en 40 mV lager. Dan heb je het bereik van 80 mV. Elke waarde tussen deze gekozen twee uitersten is dan goed (ik ga hier uit van de mogelijkheid waarbij 64 stappen worden gebruikt). Het *pollen* van de keuzemogelijkheden is op meerdere manieren mogelijk: stap naar het midden, kijk of het hoger of lager is, stap vervolgens weer naar het midden van de gekozen helft. Controleer of het hoger dan wel lager is. Na een aantal stappen ben je op de gekozen waarde uitgekomen. De jump naar de gewenste routine kan worden gemaakt.

Een controller met slechts een enkele comparator. Hier moet je wat extra weerstanden toepassen, namelijk dezelfde als die je in de schakelaar gebruikte. Vervolgens sluit je de uitgangsspanning van de duimwielchakelaar op de ene poot, de uitgang van het andere stel weerstanden op de andere poot, en je gaat bitjes hoog maken vanaf het MSB.

Wanneer de comparator omklapt, wordt het bitje weer laag gezet, en het op een na hoogste bitje wordt geset. Klapt de comparator nu niet om, dan blijft het bitje hoog. Vervolgens het derde bitje. Zo behandel je alle bitjes,

tot aan het LSB.

De fout die je nu gemaakt hebt is hooguit 1 bit, immers er zit altijd verschil tussen de weerstanden. Het verschil met de vorige schakeling is dat je nu een binair getal hebt, waarvan de afwijking niet groter is dan + of - 1 bit. Op deze manier zijn 128 stappen mogelijk, aanzienlijk meer dan bij de eerste methode.



Toch kan het nog nauwkeuriger, namelijk wanneer je meerdere bits gebruikt bij tellen. Een comparator is een analoog element, dat vele waarden kent.

Een *Atmel* is intern voorzien van een comparator, en de darlington array is niet nodig, omdat de poorten van een *Atmel* sterk genoeg zijn om rechtstreeks de weerstanden aan te sturen. Op deze manier is de *Atmel* ook te gebruiken als voltmeter, omdat middels deze methode elke spanning is te meten. Door gebruik te maken van meerdere weerstanden, is een grotere nauwkeurigheid haalbaar.

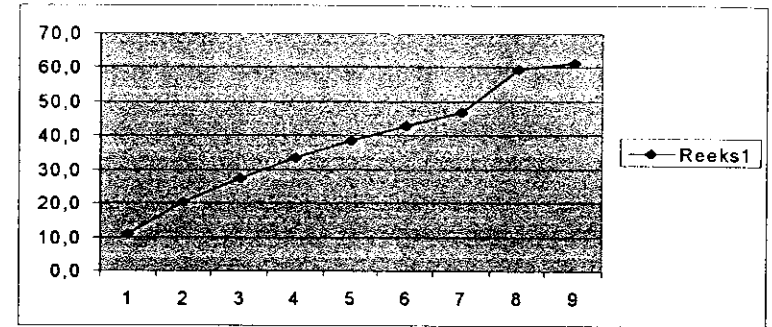
Bovenstaand verhaal heb ik geschreven in *Robobit* van de *Robotica-gg*. Daarbij heb ik naderhand pas een duimwielchakelaar gemaakt om zelf te ervaren, dat het verhaal veel mooier is dan de daadwerkelijke duimwielchakelaar. Natuurlijk, ik kan het verhaal geheel overnieuw schrijven, maar dat is mij te veel moeite.

Een van die Roboticanen, de persoon die mij oorspronkelijk aanzette tot het schrijven van voorgaand stukje, had een spreadsheetprogramma gemaakt waarmee hij de weerstanden kon bepalen. Dat liet hij mij dan ook zien en warempel, het voorbeeld dat hij gaf werkte perfect. In een spreadsheet maakte hij de formules, en vervolgens berekende hij de spanningsverhouding waarmee het selectieproces mee aan de gang moet.

Ik, ach arme ik, ben totaal onbekend met het gebruik van spreadsheets, sterker nog, ik heb er nooit mee gewerkt. Dat gaf een extra handicap, maar ik heb van de nood een deugd gemaakt. Net noodlot heeft mij de ziekte wet in gedreven en dat gaf mij de gelegenheid om een cursus *Excel* te volgen. Vervolgens gedaan wat gedaan moest worden, en de tabel aangepast.

A	B	C	D	E	F	G
<i>weerstanden</i>	<i>E24 reeks</i>			<i>stand 0</i>	<i>waarde</i>	<i>spanning</i>
680	10	22	47	a5 1	8200	10,9
2000	11	24	51	a4 2	3900	20.4
3900	12	27	56	a5 a4 3	2643	27.5
8200	13	30	62	a3 4	2000	33.3
1000	15	33	68	a5 a3 5	1608	38.3
	16	36	75	a4 a3 6	1322	43.1
	18	39	82	a5 a4 a3 7	1138	46.8
	20	43	91	a2 8	680	59.5
				a5 a2 9	628	61.4

In kolom E zie je de weerstanden die betrokken zijn met betreffende stand van de duimwielchakelaar. Deze standen zie je in de horizontale as van de grafiek. De verticale grafiekas geeft de procentuele waarde van de spanning over de serieweerstand (R), waarover de spanning wordt afgenomen. De formules achter de cellen zijn niet zichtbaar, maar in eerdere verhalen over serieschakelen en parallelschakelen van weerstanden is daar uitvoerig op ingegaan. Mocht u het niet meer precies weten, sla dan oude NewBrain on-lines erop na. Kijken op onze website kan natuurlijk ook geen kwaad. Daar is ook veel tekst te vinden over het wel en wee van de elektronica.



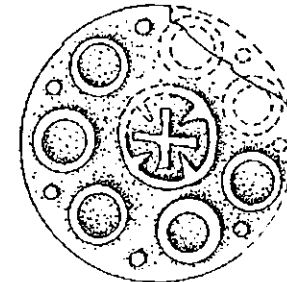
De uiterste waarden van 0 % en 100 % zijn natuurlijk niet haalbaar, omdat er altijd een weerstand tussen zit, aan de 100% zijde zitten de weerstanden van de duimwielchakelaar en aan de 0% zijde zit de serieweerstand (of net andersom, als je de aansluitingen verwisselt).

Dat heb ik in de spreadsheet namelijk gedaan. Een en ander is namelijk niet gelijktijdig geschreven. De koppeling is later weer gemaakt.

Wanneer je een duimwielchakelaar tegen komt die voorzien is van meerdere contacten, één moedercontact en 10 afgaande contacten, dan kan er veel nauwkeuriger gewerkt worden. Dan is voor elke waarde slechts één weerstand verantwoordelijk. Dat geeft dan wel weer meteen meer benodigde onderdelen.

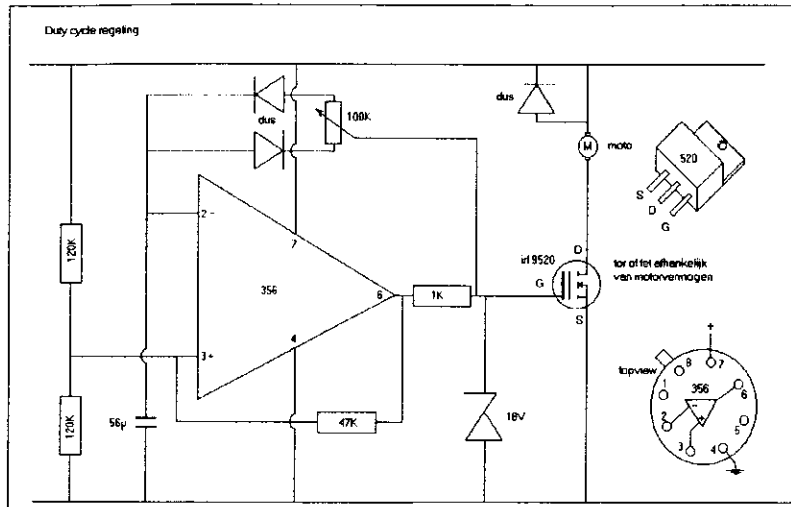
Zoals in het begin is vermeld, het gaat hier om een zo eenvoudig mogelijke schakeling. Dat het met wat decoders veel mooier en uitgebreider te maken is, weet ik. Maar daarvoor is hier juist niet gekozen.

Dré Jansen  
Djansen.2@hccnet.nl



# boormachine

## toerenregeling voor kleine motoren



Het is niet altijd nodig om een microcontroller toe te passen, wanneer er een klein motortje in toeren geregeld moet worden. Jaren geleden, lang voordat er computers waren, heb ik dit schakelingetje gebouwd. Ik had weinig ruimte ter beschikking en de *dikke fet's* waren net uitgevonden. Onbetaalbaar voor de gewone man, maar ik kende iemand die als *application engineer* werkte bij een groot bedrijf. Deze man kreeg van de fabrikant enkele van deze fet's en moest daar toepassingen voor bedenken. Een van de toepassingen is bovenstaande.

Tot op de dag van vandaag werkt het schakelingetje nog perfect. Mogelijk zijn de fet en de *opamp* niet meer verkrijgbaar, maar in principe is elke

opamp geschikt en de fet- of transistorkeuze is natuurlijk afhankelijk van het motorvermogen.

Een kleine boormachine trekt op 12 V al snel 10 A, wanneer hij vastloopt. Dat is veel stroom, die wel verwerkt moet kunnen worden. De voeding hoeft niet zo kritisch te zijn. Wanneer je twijfelt, dan kun je eenvoudig de elektronica middels een diode scheiden van de motor. Achter de diode sluit je de regelektronica aan, en voor de diode de motor. Natuurlijk moet je de blusdiode dan ook aan de andere kant van de voedingsdiode aansluiten. Achter de voedingsdiode kan je een elco zetten waarmee je de voeding voor de elektronica afvlakt.

## de werking

De opamp probeert beide ingangen op gelijk niveau te houden. De condensator op de min- ingang wordt geladen via de uitgang van de opamp, die hoog is. Deze uitgang is hoog, omdat de positieve ingang hoger is dan de negatieve ingang.

Bekijk je de twee weerstanden van 120 kΩ, dan moet de spanning op het midden de halve voedingsspanning zijn. Bij 10 V is dat 5 V. *Nee*, deze spanning is géén 5 V maar hoger, immers de uitgang is hoog, waardoor de bovenste weerstand lager is dan 120 kΩ, namelijk  $120 // 47 = 38 \text{ k}\Omega$ . De spanning op de positieve ingang is  $120 \text{ k}\Omega \times 10 \text{ V} / (38 \text{ k}\Omega + 120 \text{ k}\Omega) = 7,5 \text{ V}$ .

De condensator wordt nog steeds opgeladen, via 1 kΩ, de bovenste tak van de potmeter en de bovenste diode (1 kΩ is ter bescherming van de fet). Dat proces gaat door, totdat de negatieve ingang eveneens 7,5 V is geworden. Daarna krijgt de negatieve ingang een grotere waarde dan de positieve, waardoor de uitgang van uiterst hoog, naar uiterst laag omklapt. Het gevolg daarvan is, dat de weerstand van 47 kΩ nu plots parallel komt te staan met de onderste van de twee weerstanden van 120 kΩ. Nu is de spanning op de positieve ingang 2,5 V (mag je zelf uitrekenen of het klopt). De condensator zal zich gaan ontladen, immers de uitgang van de opamp is laag, terwijl op de condensator een spanning van 7,5 V staat. Dat

ontladen gaat door, totdat de condensatorspanning onder de 2,5 V komt, dan wordt de uitgang weer hoog, enzovoort enzovoort enzovoort.

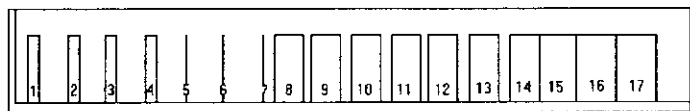
De weerstand waarmee wordt ontladen, en de weerstand waarmee wordt opgeladen, is totaal gelijk, immers dat is de potmeter. Daarom is de *frequentie* van deze oscillator constant.

Wat wel veranderd is de *duty cycle*. De tijd van opladen en de tijd van ontladen is afhankelijk van de stand van de potmeter! Kijk, nu zijn we waar we wezen willen. Wanneer de uitgang van de opamp hoog is, dan is de fet geopend, en vloeit er stroom door de motor. Is de uitgang laag, dan is de fet gesloten en is er geen stroom door de motor.

Een motor kan in toeren geregeld worden door een kleinere spanning, maar dan is ook het koppel kleiner. Dat koppel, ofwel motorkracht, is van groot belang. Trekt een motor in normaal bedrijf een à twee ampère, een geblokkeerde motor trekt een veelvoud daarvan. Zoals in het begin al verteld, dit kan fout gaan. Niet alleen de fet, maar ook de motor zelf moet deze stroom kunnen verwerken. Dat, beste mensen, dat kan de gemiddelde motor niet en zal gaan stinken!

## de grafiek

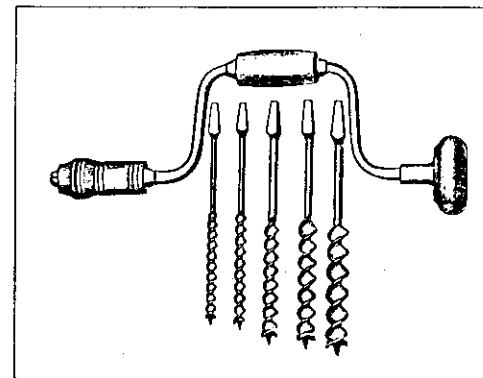
Deze duty cycle regeling geeft de motor steeds de spanning waarvoor hij ontworpen is. Alleen, niet continu, maar afhankelijk van de stand van de potmeter meer of minder tijd. De hoogte van de blokjes geeft de spanning aan, en de breedte de tijd dat de spanning aanwezig is.



De tijdsblokjes staan op regelmatige afstanden van elkaar, zoals je aan de nummering kunt zien (de frequentie is constant). De nummers 1 tot en met 4 is een rustig toerental, terwijl bij de nummers 5, 6 en 7 de motor geheel stil staat. Of zeer langzaam draait.

De nummers 8 tot en met 13 geeft een veel hoger toerental aan, en nummer 14 mag dan wel korter zijn dan de overige nummers, maar die zijn wel aansluitend op nummer 14. Vanaf tijdblok 14 draait de motor op volle toeren.

De komende bijeenkomstdag neem ik de boormachine mee, zodat jullie kunnen zien hoe een en ander in elkaar zit, en kunnen voelen dat de motor bij laag toerental nog een redelijke kracht kan ontwikkelen.



Voor meer info over dit soort zaken zie de website van de gebruikersgroep:  
<http://www.hobby.nl/~newbrain-gg/>

Met een kleine aanpassing is deze schakeling te gebruiken voor het aansturen van servomotoren uit de modelbouw. De frequentie moet dan 50 Hz zijn, immers elke 20 ms moet er een puls komen. De puls mag niet langer zijn dan 2 ms. Daartoe moet er aan een der zijden van de potmeter een weerstand in serie worden geschakeld. De oplaadtijd is dan anders dan de ontlaattijd, en niet door verdraaien andersom te krijgen. Het gehele bereik is dan beschikbaar voor de servo.

Er bestaan elektronische potmeters die eenvoudig middels een binair getal of pulsjes kunnen worden aangestuurd. Voilà, de microprocessorgestuurde servo! De schakeling is gelijk aan deze, maar frequentie en duty cycle zijn aan bepaalde voorschriften gebonden. Dus andere weerstanden.

Dré Jansen





# weerstand

Wanneer we weerstanden berekend hebben, dan willen we natuurlijk ook een weerstand met die berekende waarde hebben. Vaak blijkt dat juist die weerstand niet bestaat. Dan moet je een handelswaarde kiezen die in de buurt komt. Hier een lijstje met handels waarden

Er zijn verschillende reeksen: E6, E12, E24 en E96.

De E6 reeks kent 6 verschillende waarden binnen een honderdtal, namelijk 10, 15, 22, 33, 47 en 68. De daar op volgende waarde is dan 100, en vervolgens 150, 220 enzovoort. Dit begint bij 10  $\Omega$  en loopt door tot ver in het megaohmgebied.

De meest gebruikte reeks is de E12-reeks, die de 12 verschillende waarden kent. Dat deze waarden dicht bij elkaar liggen, moge duidelijk zijn. De handelswaarde die u moet aanschaffen wijkt minder af van de door u berekende waarde. Eerst maar eens een tabel van de handelswaarden:

E6	E12	E24	E96	E6	E12	E24	E96
10	10	10	10,0	33	33	33	31,6
			10,2				32,4
			10,5				33,2
			10,7				34,0
	11	11	11	11,0	36	36	34,8
				11,3			35,7

E6	E12	E24	E96	E6	E12	E24	E96			
	12	12	11,5	47	47	47	36,5			
			11,8				37,4			
			12,1				38,3			
			12,4				39,2			
			12,7				40,2			
			13,0				41,2			
	13	13	13				13,3	51	51	42,2
							13,7			43,2
							14,0			44,2
							14,3			45,3
							14,7			46,4
							15,0			47,5
15	15	15	15,4	56	56	56	48,7			
			15,8				49,9			
			16,2				51,1			
			16,5				52,3			
			16,9				53,6			
			17,4				54,9			
	18	18	18				17,9	62	62	56,2
							18,2			57,6
							18,7			59,0
							19,1			60,4
							19,6			61,9

E6	E12	E24	E96	E6	E12	E24	E96
		20	20,0				63,4
			20,5				64,9
			21,0				66,5
			21,5	68	68	68	68,1
22	22	22	22,1				69,8
			22,6				71,5
			23,2				73,2
			23,7			75	75,0
		24	24,3				76,8
			24,9				78,7
			25,5				80,6
			26,1		82	82	82,5
			26,7				84,5
	27	27	27,4				86,6
			28,0				88,7
			28,7			91	90,9
			29,4				93,1
		30	30,1				95,3
			30,9				97,6

Zo, dat is een heel typewerk geweest om deze getalletjes neer te zetten. Ik moet eerlijk bekennen, dat ik er geen logische volgorde in herken. Mocht iemand die wel zien, dan hoor ik dat graag (djansen.2@hccnet.nl). Hoe en waarom men juist deze waarden heeft gekozen, is mij niet bekend. De meeste elektronikaboeren hebben de E24-reeks wel in huis, maar voor de

E96-reeks moet u toch naar de betere elektronica-onderdelenhandel. Vaak is met het parallel schakelen van weerstanden nog wat verdere nauwkeurigheid te bereiken.

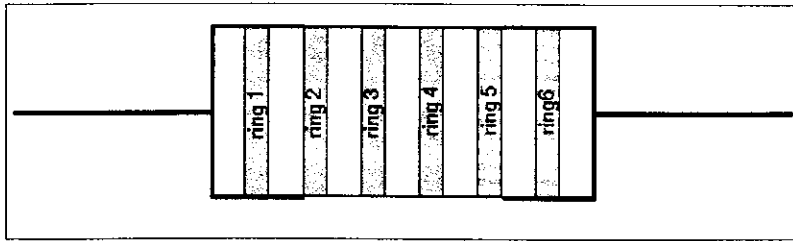
Weerstanden zijn in diverse toleranties verkrijgbaar:

B	0,1 %	paars
C	0,25 %	blauw
D	0,5 %	groen
F	1 %	bruin
G	2 %	rood
J	5 %	goud
K	10 %	zilver
M	20 %	geen kleuraanduiding

Ook hier is het mij geheel onduidelijk, waarom de letters A, E, H, I en L zijn overgeslagen. Voor de letters I en L kan ik me nog wel wat voorstellen, want de hoofdletter I en de kleine letter l lijken er veel op het cijfer 1, maar waarom die andere niet gebruikt zijn, is mij een raadsel. Ook hier geldt: wie het weet mag het zeggen. Dit zijn de codes die in de handel gebruikt worden.

Weerstanden worden steeds kleiner, en lang geleden ontdekte men al dat een cijfertje moeilijk leesbaar is, en makkelijk beschadigt, waardoor het onleesbaar wordt. Kleuren zijn beter herkenbaar en vooral goedkoper aan te brengen.

De kleuraanduiding voor de toleranties zijn al gegeven. Elke kleur krijgt een *cijferwaarde*, en vervolgens dient men nog de richting van lezen en de betekenis van de positie van de kleurring aan te geven.



De meesten onder ons kennen die kleurcode wel, maar voor de volledigheid van dit verhaal schrijf ik het er toch even bij. Bovendien is het weer een aardige opvulling van ons blad.

De eerste ring is de meest linkse, en gewoonlijk bevat een weerstand 4 of 5 ringen. Daarmee is de linker ring goed herkenbaar, want die zit het dichtst bij het aansluitdraadje.

De eerste drie ringen geven de drie cijfers aan waarmee de waarde van de weerstand wordt weergegeven. Dus rood-wit-blauw betekent: 396.

× 0,01	zilver
× 0,1	goud
× 1	zwart
× 10	bruin
× 100	rood
× 1 k	oranje
× 10 k	geel
× 100 k	groen
× 1 M	blauw
× 10 M	paars

0	zwart
1	bruin
2	rood
3	oranje
4	geel
5	groen
6	blauw
7	paars
8	grijs

Dan is er de vierde ring, die de vermenigvuldigingsfactor aangeeft.

Dus die zelfde weerstand van daarnet, met oranje wimpel, zou een waarde vertegenwoordigen van 396 kΩ, immers oranje is: '× 1000'.

Nu wilt u deze weerstand kopen, maar helaas, hij bestaat niet, zelfs in de E96-reeks komt hij niet voor, jammer!

De eerste vier banden komen op elke weer-

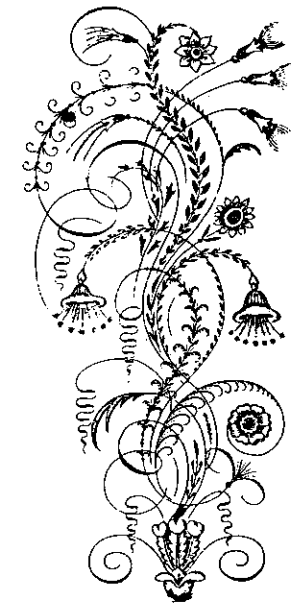
stand voor, maar de vijfde band is voor de tolerantie. Weerstanden van 20 % hebben geen tolerantieband. De meeste weerstanden zijn 10 % of 5 %, dus hebben een zilveren of gouden band. Dat is dan de laatste band. Duidelijk herkenbaar, vergissen uitgesloten. Goud en zilver zijn geen kleuren waarmee de waarde wordt aangegeven.

oranje	15 ppm
geel	25 ppm
rood	50 ppm
bruin	100 ppm

Soms wordt er nog een zesde band vermeld. Die is zeldzaam, maar komt wel voor. Deze zesde band wordt gebruikt om de temperatuurgevoeligheid van de weerstand mee aan te geven. U snapt zeker wel, dat de meest stabiele weerstanden niet de goedkoopste zijn. Voor het gewone elektronica-werk, dat wij verrichten, is dit niet nodig. Toch maar even vermelden, al is het alleen maar voor de bladvulling.

Zo, als jullie nou ook eens wat schrijven, dan kan er mogelijk wat vaker een NewBrain on-line verschijnen.

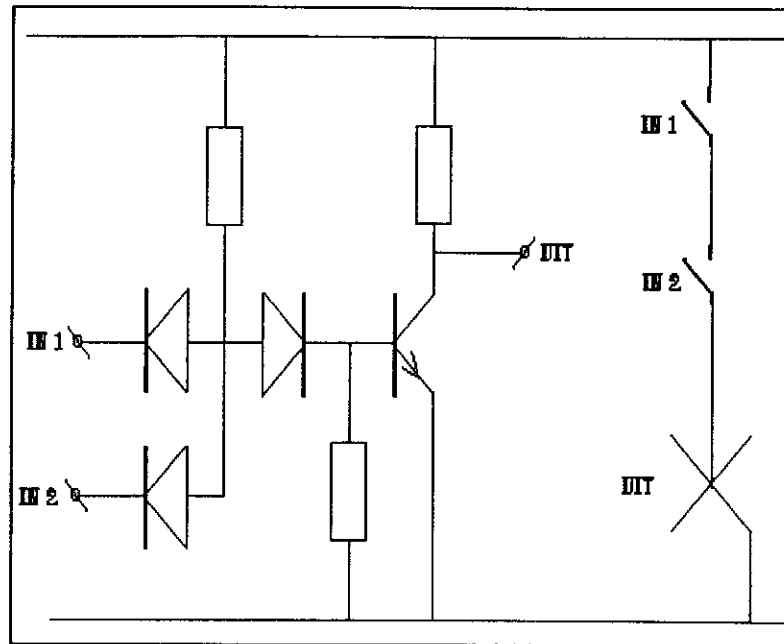
Dré Jansen  
djansen.2@hccnet.nl



# digitaal

## deel 1

Na al het analoge geweld wordt het tijd om digitaal te gaan. In tegenstelling tot analoog, kent de digitale techniek slechts twee standen, *aan* en *uit*. Dat is natuurlijk al in vele verhaaltjes verteld, dus hier vertel ik niets nieuws. Digitale schakelingen zijn opgebouwd uit analoge zaken zoals transistoren.



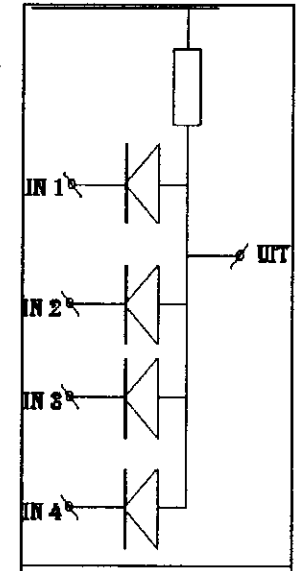
Aan de linker zijde bevinden zich de twee ingangen en aan de rechterzijde zit de uitgang. Wanneer er niets is aangesloten, dan zijn beide ingangen hoog, en is de uitgang laag. Wanneer een of beide ingangen laag zijn, wordt de uitgang hoog. Dat is een NAND-functie; daaraan zijn in andere artikelen al veel woorden vuil gemaakt. Ik zal het hier dunnetjes overdoen, voor de mensen die deze woorden mogelijk nog niet gelezen hebben. De schakeling is vergelijkbaar met een serieschakeling van twee contacten die een lamp bedienen. Beide schakelaars moeten in staan, alvorens de lamp zal branden.

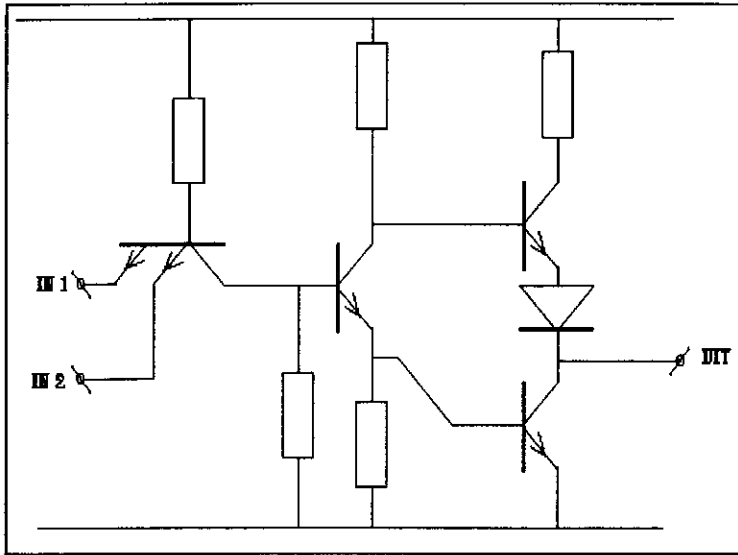
De weerstand aan het verbindingpunt van de drie dioden, zorgt dat dit punt hoog is. De transistor wordt daardoor opengestuurd, waardoor de spanning op de collector laag is. Mocht u dit niet begrijpen, dan kunt u de antwoorden vinden op de website van de gebruikersgroep: <http://www.hobby.nl/~newbrain-gg/>

De diode die voor de basis van de transistor zit, is ervoor om het spanningsverlies te compenseren, dat ontstaat wanneer u beide ingangen laag maakt. Zonder deze diode zou de transistor niet geheel sluiten. Deze schakeling is veel sneller dan een vergelijkbare schakeling met relais, maar het kan nog sneller door de uitgang actief te maken. De uitgang wordt actief hoog en actief laag gemaakt.

Ingangen worden hoogohmiger gemaakt door de ingangsdioden meteen op het basissubstraat van de ingangstransistor te etsen. Hierdoor krijg je een transistor met meerdere emitters.

Er zijn twee ingangen, maar daar kan wat aan gedaan worden. Zijn er meerdere ingangen nodig, dan kan met dioden het aantal ingangen worden uitgebreid. Verder is zelfs zonder actieve componenten een AND te vormen. De uitgang is hoog, wanneer geen der ingangen aan massa komt te liggen. Zodra een in-

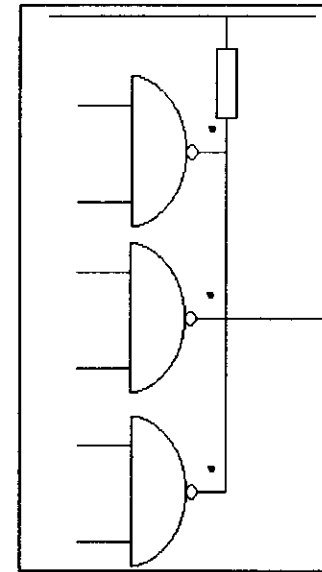




gang laag wordt, dan is de uitgang eveneens laag. Wanneer deze schakeling wordt aangesloten op een NAND, dan heeft die NAND een heleboel ingangen. Natuurlijk zijn er NAND's in de handel met een heleboel ingangen, maar je zal zo'n IC maar net niet in huis hebben liggen.

Uitgangen zijn natuurlijk ook te koppelen, maar dan kan het gebeuren dat de een poort 'hoog' is en de andere 'laag'. Dus van de ene poort staat de bovenste transistor in geleiding, en van de andere de onderste. Dat er dan een ontoelaatbaar hoge stroom gaat vloeien moge duidelijk zijn.

Om toch uitgangen te kunnen koppelen, moet er een uitvoering worden gekozen waarbij de pull-up of collector weerstand ontbreekt. In het type en bij het symbool wordt dit meestal aangegeven met een sterretje (\*). Daarbij moet natuurlijk wel de mogelijkheid bestaan om de 'hoog'-situatie weer te geven. Dus je moet zelf een uitwendige weerstand aanbrenge. De uitwendige weerstand moet natuurlijk zo hoog zijn dat de maximale stroom die een poort kan verdragen niet te groot wordt.



Bij hoge snelheden, en hoge impedanties, kunnen er fouten optreden als gevolg van stoerpulsen. Ook daar kan een externe weerstand hulp bieden. Door een (extra) weerstand te plaatsen wordt de uitgang laagohmiger. Hoe lager de uitgangsweerstand des te ongevoeliger is de schakeling voor stoerpulsen. Het nadeel is wel dat de schakeling meer vermogen opneemt.

Belastbaarheid van de poorten is ook al zoiets: je kunt niet oneindig veel ingangen aan een uitgang hangen. De ingangen mogen dan wel hoogohmig zijn, maar nemen wel iets vermogen op. Uitgangstrappen zijn gewoonlijk geen vermogenstrappen, dus ook hier is het beschikbare vermogen aan grenzen gebonden. Gewoonlijk kunnen er 10 ingangen op een uitgang

worden aangesloten. Dit wordt *fan out* genoemd. Een fan out van 6 betekent, dat er maximaal 6 ingangen op deze uitgang kunnen worden aangesloten.

Er zijn speciale buffer poorten met een fan out van 50, dus voor elk probleem is er een panklare oplossing. Omdat digitale poorten zijn opgebouwd uit analoge elementen, is het zelfs mogelijk om zelf een uitgang te bouwen. Wanneer je open collector uitgangen neemt, dan is zelfs binnen bepaalde grenzen een grotere spanning op de uitgang toe te laten. Ook hier zijn chips, die spanningen tot 50 V aan de uitgang toelaten.

Het kan nog veel sneller. Er zijn namelijk speciale series die gebouwd zijn op snelheid en voor een laag energiegebruik. LS- en de HC-series zijn daar voorbeelden van. Deze hoge snelheden en lage vermogens worden bereikt door een hogere impedantie. Zoals gezegd, een hogere impedantie geeft weer grotere storingsgevoeligheid. Die storingsgevoeligheid kan weer worden afgeschermd door speciale ingangsfilters op het voedingsapparaat en afscherming.

Magnetische afscherming door ijzer, een ijzeren buisje of kastje doet wonderen, vooral wanneer de schakeling staat opgesteld bij motoren en magneetschakelaars. Wanneer er sprake is van hoogfrequent instraling, dan is een koperen omhulling de oplossing. Door de goede elektrische eigenschappen van koper ontstaat een kooi van Faraday die de HF-stoorsignalen middels wervelstromen afschermt.

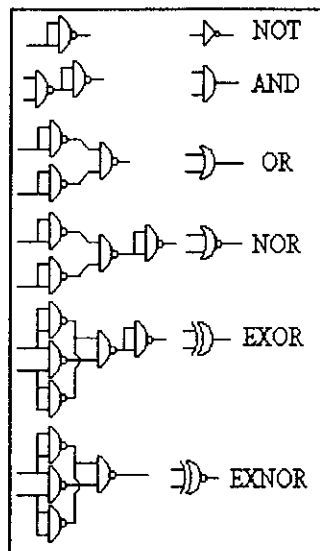
Dus, mochten hoge snelheden en lage vermogensopname nodig zijn, dan bestaan er voor storingsproblemen wel degelijk oplossingen. Vergeet vooral ook niet de bedrading af te schermen, want een mooiere antenne, gevormd door lange niet afgeschermde draden is nauwelijks denkbaar.

Totnogtoe heb ik het uitsluitend gehad over de NAND. U begint zich af te vragen of er nog andere zijn? Welke andere? Er zijn geen andere! Toch hebt u de termen AND, NOT, OR, EX-OR, NOR en EX-NOR gehoord. Welnu, die worden gevormd uit schakelingen gemaakt met NAND's.

Een NOT is niets meer of minder dan een schakeling die de polariteit van het signaal omkeert, dus 1 wordt 0 en 0 wordt 1. Een AND is een NAND

waarachter door een NOT het signaal wordt omgekeerd. Wanneer je de waarheidstabel van een AND bekijkt, dan zie je dat de nullen zich gedragen als een OR, welnu, omgekeerde enen zijn toch nullen? Of niet soms. Een AND voorafgegaan door NOT's geeft OR. Een NOR is een OR gevolgd door NOT. Een EX-OR, ofwel exclusief OR is een OR die bij twee gelijke ingangen 0 geeft en bij ongelijke ingangen een 1. De EX-NOR is een geïnverteerde EX-OR.

De kreet *waarheidstabel* is gevallen. Een waarheidstabel is niets meer dan een tabel waarin vermeld staat, wat de uitgang zal zijn bij de diverse mogelijkheden die de ingangen kunnen aannemen. Vier ingangen geven 16 mogelijkheden, dus zo'n tabel



zal 16 regels lang zijn.

Hier gaat het slechts om 2 ingangen, dus de waarheidstabel kent 4 regels.

NAND			AND		
in	in	uit	in	in	uit
0	0	1	0	0	0
1	0	1	1	0	0
0	1	1	0	1	0
1	1	0	1	1	1

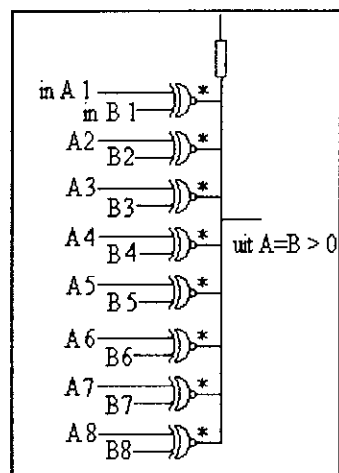
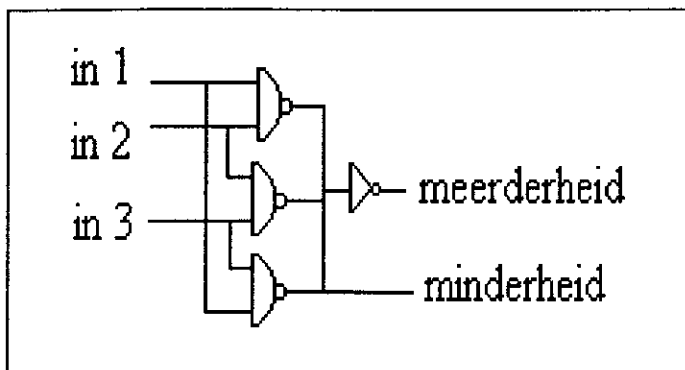
OR			NOR			EX-OR			EX-NOR		
in	in	uit	in	in	uit	in	in	uit	in	in	uit
0	0	0	0	0	1	0	0	0	0	0	1
1	0	1	1	0	0	1	0	1	1	0	0
0	1	1	0	1	0	0	1	1	0	1	0
1	1	1	1	1	0	1	1	0	1	1	1

Zoals gezegd, wanneer je de nullen van de AND vergelijkt met de enen van de OR dan merk je op dat deze overeenkomen. Telkens waar bij een AND een 0 staat vindt je bij de OR een 1. Dit 'nullenverhaal' wordt ook wel *negatieve logica* genoemd. Verder niet veel aandacht aan besteden, maar onthoud dat je hier gebruik van kan maken.

In een schakeling kom je een OR te kort. Je hebt wel verdeeld over diverse chips een aantal NAND-poorten over. Welnu, bespaar de ruimte en vermogensopname van een OR-chip door de niet gebruikte NAND-poorten te benutten. Zo kun je van een EX-OR of EX-NOR die over is, een *inverter* maken; sterker nog, je kunt wel of niet inverteren al naar gelang de polariteit van het andere pootje.

## deel 2

Ik heb drie signalen, die allemaal hetzelfde doen. Echter het kan zijn dat een van deze signalen fout is. Daarom wordt in vliegtuigen bijvoorbeeld een besluit pas genomen, wanneer de meerderheid in overeenstemming is. De meeste stemmen gelden, dus wanneer er twee hoog zijn, dan is de uitgang hoog, zijn er twee laag, dan is de uitgang laag.



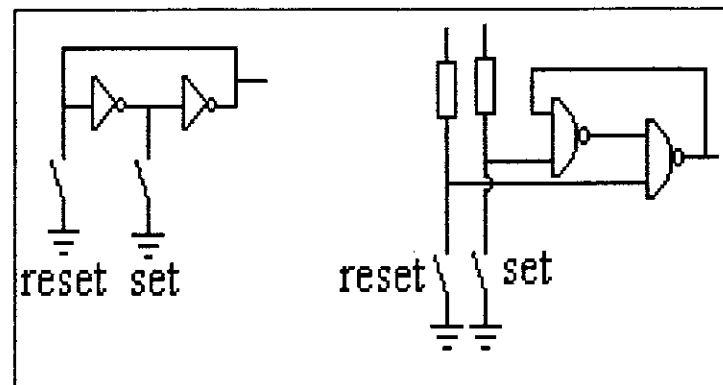
Een andere zaak is het vergelijken van twee waarden. In de analoge techniek komt het zeer vaak voor, dat er twee waarden met elkaar vergeleken worden. Zo ook in de digitale wereld. Wanneer beide waarden gelijk zijn aan elkaar, dan is er geen uitgang, zijn ze ongelijk aan elkaar dan is er wel signaal.

Hiervoor zijn complete chips, maar een aantal gekoppelde EX-OR of EX-NOR-poorten doet het ook. Kies natuurlijk wel opencollectorexemplaren.

Een EX-NOR geeft een 0 bij ongelijke ingangen. Is de waarde A niet gelijk aan

de waarde B, dan zal de uitgang 0 zijn. Deze chip komt ook voor met EX-OR en met een inverterende en rechte uitgang.

Een *RS flipflop* wordt gevormd door twee inverters achter elkaar geschakeld. De stand die wordt aangenomen is onbepaald. Door extra ingangen kan de stand worden geset of gereset.



Met een flipflop kan een mechanische schakelaar worden *ontdenderd*. Wanneer je een schakelaar omzet, dan gaat dat niet in een keer, maar dendert de schakelaar vele malen. De flipflop schakelt al bij de eerste keer om, en zal de waarde behouden ondanks het gedender van de schakelaar. In plaats van twee enkelpolige schakelaars kan natuurlijk ook een wisselschakelaar worden gebruikt.

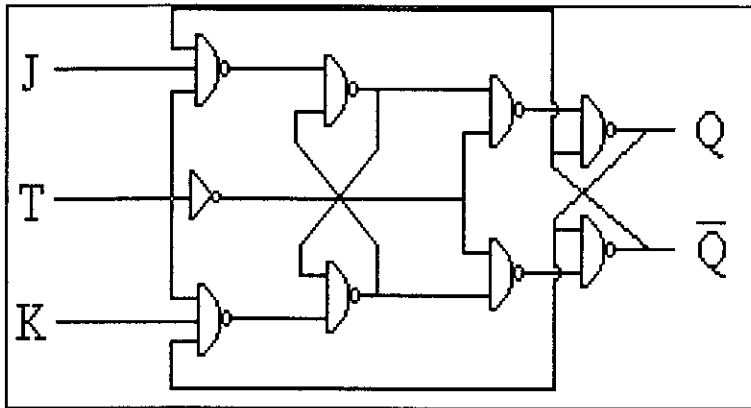
Dan kent de flip-flop nog een verboden stand: beide ingangen 0. Wanneer dat zo is, dan is de uitgang ongedefinieerd.

Een bijzondere flipflop is de *J-K flipflop*. Hier wordt de stand van de ingang aan de uitgang doorgegeven op het moment, dat de trigger ingang een opgaande flank ervaart. Dan heeft de ingang geen invloed meer, en zal de uitgang de waarde van de ingang, die op dat moment aanwezig was, overnemen.

De ingangscombinatie 0 - 0 geeft na een triggerpuls geen wijziging aan de

uitgang.

De stand  $J = 1$  en  $K = 0$  geeft aan de uitgang een overeenkomstige stand.  
De stand  $J = 0$  en  $K = 1$  geeft aan de uitgang eveneens een zelfde stand.  
De stand  $J = 1$  en  $K = 1$  is een heel bijzondere: dan wisselt de uitgang bij elke triggerpuls. Was  $Q = 1$ , dan wordt hij  $0$ ; was  $Q = 0$  dan zal hij na triggeren  $1$  zijn.  
Dat maakt de J-K flipflop geschikt als frequentiedeler en teller.

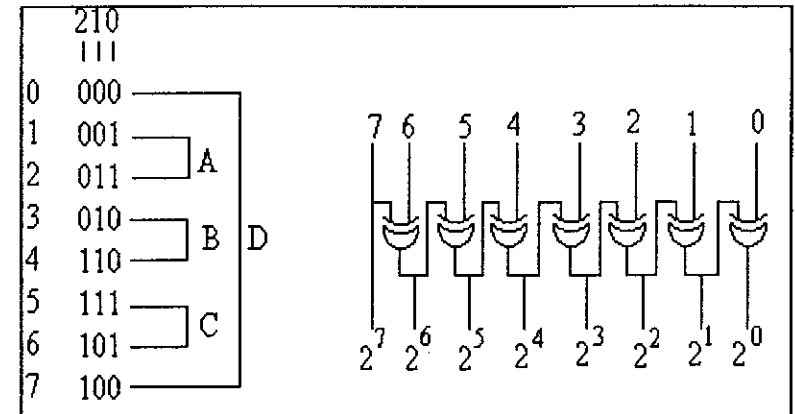


Verderop meer over deze digitale bouwsteen.

Wanneer we een windwijzer bouwen, een verplaatsingsopnemer of noem maar wat op, waarbij de stand aan de hand van een digitale opnemer moet worden bepaald, dan wordt daar de *Gray-code* voor gebruikt.

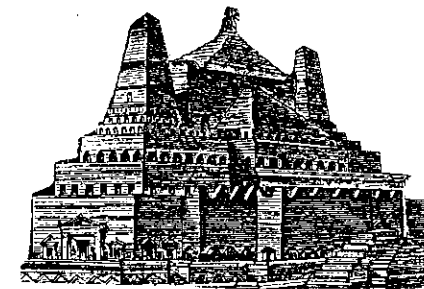
Het grote voordeel van de gray code is, dat er tussen twee opeenvolgende positieaanduidingen slechts een bitje verandert. Stel, ik heb 16 posities aan te geven, waarbij ik een normale digitale liniaal gebruik. Positie 7 wordt aangegeven met 0111 en de eerstvolgende positie (8) met 1000. Daarvoor gebruik ik 4 sleepcontacten waarvan door slijtage of onnauwkeurige constructie, de buitenste twee contacten wat eerder omschakelen dan de middelste twee contacten. Dan, vlak voordat positie 7 overgaat in positie 8, wordt er heel even positie 14 doorgegeven ( $14 = 1110$ ). Immers de buitenste twee 0..1 gaan eerst 1..0 en de middelste blijven nog heel even .11.

Om dit te voorkomen is er de gray-code bedacht. Daarbij zijn verschillende mogelijkheden, zolang er maar telkens een bit verandert. Kiezen we een code van 3 bits

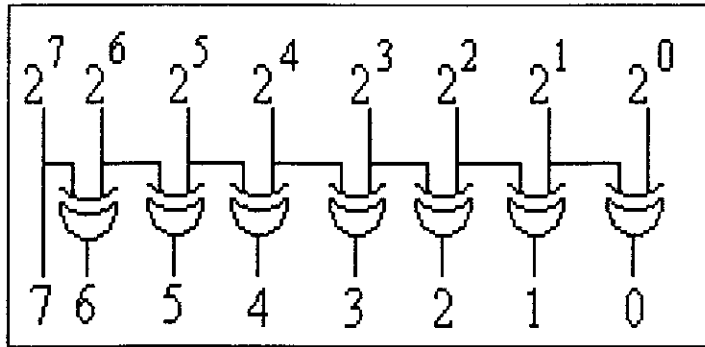


Een zuivere gray-code is herkenbaar aan de mogelijkheid dat twee opeenvolgende codes kunnen worden weggelaten waarbij aan de voorwaarde van het slechts een bit verandert bij overgang wordt voldaan. Haal combinatie A weg, het blijft een gray code, zo ook bij B, C en D. Het is natuurlijk onzin om A en C weg te halen, dan houd je vier standen over.

Het LSB komt te vervallen (blijft 0), waardoor er een twee bits code over is. Vervolgens moet de code worden omgezet in de gebruikelijke binaire code, immers dat is de code waarmee we rekenen. Daartoe is de naastliggende schakeling bedacht, een 8 bits gray code wordt omgezet in een 8 bits binaire code. Een binaire code kan ook worden omgezet in een gray-code.







### deel 3

Het woord *computer* betekent *rekenaar* naar het Engelse werkwoord *to compute*. Daar computers digitale apparaten zijn, moet het dus mogelijk zijn om met digitale waarden te rekenen. Hoera! Hier worden een stelletje open deuren ingetrapt. Natuurlijk wist je dat al.

Rekenen, dat kennen we al van de lagere school, dus hoef ik weinig uit te leggen. Toch zijn er wat aanpassingen, zoals hoe geef je een negatief getal aan? Vooraf moet je natuurlijk afspreken of we dit wel of niet doen. In de decimale rekenkunde wordt dit met het plusteken en het minteken aangegeven. Dat had even goed met de letters *P* en *N* gekund; het is een afspraak. In de digitale techniek worden daar de 1 en 0 voor gebruikt, en wel op de eerste positie, de positie die gewoonlijk door het MSB wordt ingenomen. Een 8-bits code wordt dan 7-bits, want het achtste bit is nu het tekenbit. Een 0 geeft een positief getal aan, een 1 duidt op een negatief getal. +35 is 0100011 en -35 is 1100011. Zou de afspraak niet gemaakt zijn, dan leest men 99 in plaats van -35. Stel je eens voor, dat deze fout op de giro gemaakt wordt.

optellen:  $23 + 12 = 35$      $00010111 + 00001100 = 00100011$   
 aftrekken:  $23 - 12 = 11$      $00010111 - 00001100 = 00001011$

Dat is dus niet veel anders dan in de decimale wereld. Een computer kan helaas niet aftrekken... Dus aftrekken op de bovenstaande manier is dan ook onmogelijk. Daarvoor wordt de *2-complementmethode* toegepast.

Eerst even in de decimale wereld laten zien wat hier wordt bedoeld. De 12 moet worden omgebouwd. Elk cijfer moet van de 9 worden afgetrokken. De 1 wordt dan  $9-1 = 8$  en zo verandert de 2 in 7. De 9-complementsweergave van 12 is 87. Tellen we op, jawel want optellen kan wel, dan komt de waarde  $23+87 = 110$ . Schrappen we het eerste digit 1, dan blijft er 10 (decimaal) over.

Wat blijkt, de uitkomst is 1 te laag want  $23-12=11$  en geen 10. Daarom moet er 1 bij worden opgeteld.  $10+1=11$  Dit noemt men dan de 9-complement + 1 methode ook wel 10-complement methode genoemd. Toch moet er van 9 worden afgetrokken en niet van 10. De uiteindelijke waarde moet met 1 vermeerderd worden. Verwarrend? Ja, jammer maar helaas!

Precies zo moet het bij de digitale rekensom. Hier moet elk cijfer van de 1 worden afgetrokken. Dat komt er op neer dat elk cijfer geïnverteerd wordt, inclusief tekenbit. Dus 00001011 wordt 11110100. Het eerste bit geeft een negatief getal. Wanneer nu wordt opgeteld dan kom je ook 1 te kort:  $23 + (-12) = 00010111 + 11110011 = 100001010$  9 bit's! Nee, het is geen fout, want net als in de decimale situatie moet ook hier het eerste bit worden weggesneden. De uitkomst is 00001010, wat overeenkomt met 10. Weer in overeenkomst met de decimale rekenarij moet ook hier 1 worden bijgeteld. Dus  $00001010 + 00000001 = 00001011$ . Dat laatste is dan 11(decimaal). Dit is 1-complement+1 methode, en die wordt 2-complement methode genoemd.

Hoe verging het u in de lagere school? Na optellen en aftrekken kwamen de breuken, of was het vermenigvuldigen? Ik weet het niet meer, maar ik behandel breuken.

Kijken we vanaf de komma, dan is het cijfer direct links ervan  $2^0$  het daarop volgende bit  $2^1$ , vervolgens  $2^2, 2^3, 2^4, 2^5, 2^6, 2^7$  enzovoort. De andere kant op wordt wat moeilijker, want dat zijn breuken. In het decimale stel-

sel is het eerste bit rechts van de komma  $10^{-1}$  het volgende cijfer  $10^{-2}$  enzovoort. In de digitale getallenreeks wordt dat  $2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}$  enzovoort.

Wanneer je dat in decimale getallen gaat weergeven, wordt het:

binair	macht	decimaal
0,1	$2^{-1}$	0,5
0,01	$2^{-2}$	0,25
0,001	$2^{-3}$	0,125
0,0001	$2^{-4}$	0,0625
0,00001	$2^{-5}$	0,03125
0,000001	$2^{-6}$	0,015625

Decimaal:  $5 : 4 = 1,25$ . Dat is:  $1 \times 10^0 + 2 \times 10^{-1} + 5 \times 10^{-2}$ . Delen in het decimale stelsel gaat het gemakkelijkst als de deler een veelvoud van het grondtal 10 is. Bij delen door 10 schuift de komma een positie naar links. Delen door 100 (het kwadraat van het grondtal) doet de komma twee plaatsen opschuiven.  $2345 : 100$  wordt 23,45

$\begin{aligned} \text{binair } 5 : 4 &= 1,25 \\ 0101 : 0100 &= 0001,01 \end{aligned}$
--

De deler is 4. Dat is de tweede macht van het grondtal 2, dus de komma verschuift 2 plaatsen.

Vermeld dient nog te worden de notatie van negatieve breuken:  $-1,25$  wordt  $-2 + 0,75$ . Bij breuken wordt het deel achter de komma altijd positief weergegeven. Dus  $0001,01$  in negatieve vorm wordt:  $1101,11$ .

Vermenigvuldigen. Decimaal: in de decimale rekenwereld maakt men bij het vermenigvuldigen gebruik van tafels. Het grondtal 10 is er de oorzaak

van dat er  $10 \times 10 = 100$  verschillende mogelijkheden zijn. Een look-up tabel wat kommaschuifwerk en optellen.

$\begin{aligned} 346 \times 54 &= 4 \times 61 + 4 \times 4 \times 10 + 4 \times 3 \times 100 + \\ &10 \times 5 \times 6 \times 1 + 10 \times 5 \times 4 \times 10 + 10 \times 5 \times 3 \times 100 \end{aligned}$
--

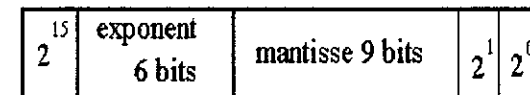
Deze getallen zijn middels de look-up tabel te vinden en de factoren 1, 10 en 100 is een kwestie van de komma verschuiven.

Digitaal gaat dat een stuk makkelijker, jawel want het grondtal 2 maakt dat er maar 4 mogelijkheden bestaan.

$0 + 0 = 0$	$0 - 0 = 0$	$0 \times 0 = 0$	$0 : 1 = 0$
$0 + 1 = 1$	$0 - 1 = 1 \text{ \& borrow}$	$0 \times 1 = 0$	$1 : 1 = 1$
$1 + 0 = 1$	$1 - 0 = 1$	$1 \times 0 = 0$	delen door 0 is onmogelijk
$1 + 1 = 1 \text{ \& carry}$	$1 - 1 = 0$	$1 \times 1 = 1$	

Vermenigvuldigen kan het eenvoudigst met *shift* en *add*. Dus vermenigvuldigen met 5 door twee posities te schuiven en het te vermenigvuldigen getal er eenmaal bij op te tellen.

Zwevende komma (*floating point*). De boekhouder kent veel grotere getallen dan tot nog toe zijn beschreven. Ik ga voor het gemak uit van een 16-bits register. Dat betekent dat het grootste getal dat kan worden weergegeven is:  $2^{15} = 32768$  positief en negatief. Het hoogste bitje is gereserveerd voor de polariteit van het getal. Delen we de ruimte wat slimmer in dan kan er een veel ruimer getalsbereik worden gerealiseerd.



tekenbit

Het MSB is nog steeds gereserveerd voor het tekenbit.. Dan zijn er 6 bits voor de exponent, die in 2-complementsnotatie wordt geregistreerd. Tot slot 9 bits voor de mantisse van het getal. Het getal, dat altijd kleiner dan 1 is en tot de macht van het eerder genoemde exponent moet worden verheven. Kijk naar de wetenschappelijke notatie, die je op je zakjapanner tegenkomt. Hier natuurlijk digitaal. Daar wordt het getal ook als 0, . . . weergegeven met een tweecijferig getal er achter (al dan niet negatief).

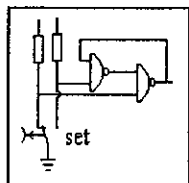
Het grootste getal dat kan worden weergegeven is:  
0 011111 111111111

De eerste 0 is het tekenbit; de tweede 0 geeft aan, dat het om een positieve exponent gaat ( $2^{31}$ ), de mantisse 0,998; dus  $0,998 \times 2^{31} = 2143\ 000\ 000$  dit zowel positief als negatief. Wanneer uw boekhouder moeite heeft om dit salaris op uw rekening te storten, dan mag u hem toestemming geven het in twee keer te doen. Kortom, hier wordt de ruimte efficiënter benut.

Rest mij nog (op bladzijde 35) een tabel te geven waarin de machten van 2 staan:

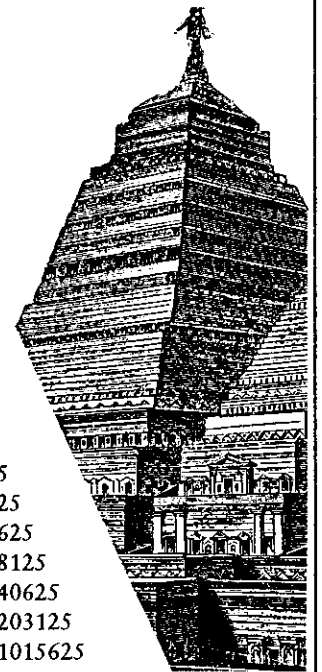
#### deel 4

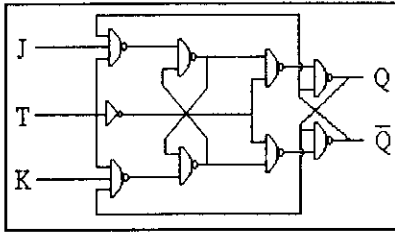
In deel 2 heb ik het al gehad over de RS flip-flop. Deze hier is dan met een als terugverende wisselschakelaar uitgevoerd. Standaard is de boel in de *reset*-stand, maar wanneer de knop wordt ingedrukt, dan is de FF geset. Dit is een manier om schakelaars te ontzenderen.



In verdere schakelingen zijn hier klokpulsen mee te maken. Klokpulsen zijn impulsen die tellers het commando geven om een stand verder te gaan. Dat dit niet handmatig gaat, moge duidelijk zijn, maar in testopstellingen moet het soms met de hand. Dan wil je er zeker van zijn, dat je op bepaalde momenten één puls hebt, en niet door het denderen van een of andere schakelaar, een onbekende hoeveelheid pulsen achter elkaar.

2 tot de macht n	n	2 tot de macht -n
1	0	1
2	1	0,5
4	2	0,25
8	3	0,125
16	4	0,0625
32	5	0,03125
64	6	0,015625
128	7	0,0078125
256	8	0,00390625
512	9	0,001953125
1024	10	0,0009765625
2048	11	0,00048828125
4090	12	0,000244140625
8192	13	0,0001220703125
16384	14	0,00006103515625
32708	15	0,000030517578125
65536	16	0,0000152587890625
131072	17	0,00000762939453125
262144	18	0,000003814697265625
524288	19	0,0000019073486328125
1048570	20	0,00000095367431640625
2097152	21	0,000000476837158203125
4194304	22	0,0000002384185791015625
8388608	23	0,00000011920928955078125
16777216	24	0,000000059604644775390625
33554432	25	0,0000000298023223876953125
67108864	26	0,00000001490116119384765625
134217728	27	0,000000007450580596923828125
268435456	28	0,0000000037252902984619140625
536870912	29	0,00000000186264514923095703125
1073741824	30	0,000000000931322574615478515625
2147483648	31	0,0000000004656612873077392578125
4294967296	32	0,00000000023283064365386962890625
8589934592	33	0,000000000116415321826934814453125
17179869184	34	0,0000000000582076609134674072265625
34359738368	35	0,00000000002910283045673370261328125
68719476736	36	0,000000000014551915228366851806640625

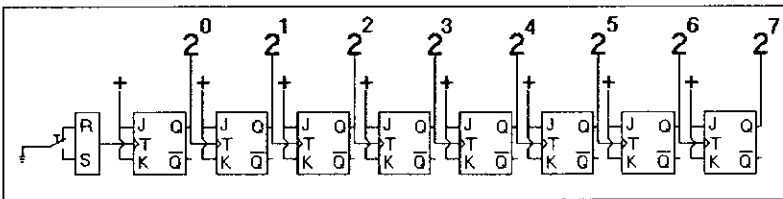




De J-K flip-flop is ook al eerder ter sprake geweest. Hier kunnen tellers van gemaakt worden. Door een reeks van deze FF's achter elkaar te hangen, waarbij Q aan J en K aan Q zit.

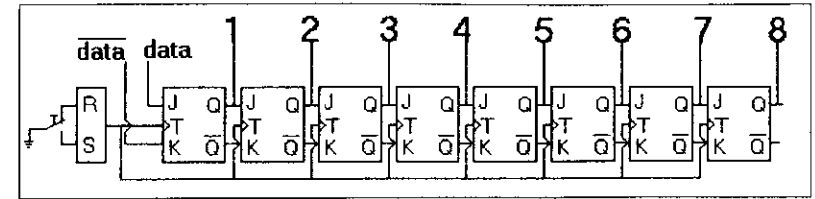
Wanneer we nu de bitjes bekijken vanaf de eerste Q van de JK waarvan we de klokpulsen toevoeren, dan zien we een keurige teller, waarbij de eerste Q het LSB is, en de laatste Q het MSB. JK FF's komen voor met twee FF's in een behuizing. D-flipflops zijn niet te gebruiken, omdat die geen J en K ingang hebben. Een D-FF heeft alleen een J-ingang, die nu ineens D-ingang heet. De K-ingang wordt intern middels een inverter uit de D-ingang gedestilleerd.

De D-FF's worden gebruikt als geheugenelement en als signaalgever, waarbij het signaal zowel hoog- als laagohmig wordt verzonden. Een 1-sigitaal is veel hoogohmiger dan een 0 signaal. Door nu een signaal zowel hoog- als laagohmig over te zenden (twee draden) is de storingsgevoeligheid veel minder. Het hoog dient samen te gaan met het bijbehorende laag.



Teller: telkens wanneer de schakelaar de RS FF set, dan wisselt de polariteit van de Q-uitgang van de betreffende JK-FF.  $2^0$  geeft de eenheden weer, en  $2^1$  de tweetallen, enzovoort. Het MSB wordt door  $2^7$  aangegeven. Een teller die telkens als de RS FF twee keer geset wordt de teller met 1 verhoogt.

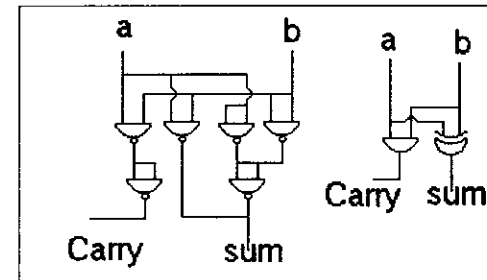
Schuifregister gebouwd uit JK FF's



Telkens wanneer de knop van de RS FF wordt ingedrukt, dan verschuiven de data een positie naar rechts. Wanneer je 7 van deze strengen parallel hebt lopen, dan kun je 7-segments displays aansturen. Als je erg veel JK FF's hebt liggen, en ook nog veel soldeertin, dan kun je alfanumerieke  $5 \times 7$  segmenten aansturen. Daar zijn wel efficiëntere methodes voor denkbaar, maar toch, het zou kunnen.

Op deze manier is een vertragsingslijn of een lichtkrant te bedenken. Met een beetje extra poorten is het mogelijk om een bidirectioneel schuifregister te maken. Kortom, de fantasie is de begrenzenende factor.

Natuurlijk zijn dit soort zaken ook met D-FF's te bouwen. Deze bouwstenen komen vaak in grotere aantallen per chip voor. Dus is compactere bouw mogelijk. Tegenwoordig kosten microcontrollers niet veel meer, dus wat complexere schuifregisters zijn met microcontrollers voordeliger te bouwen.



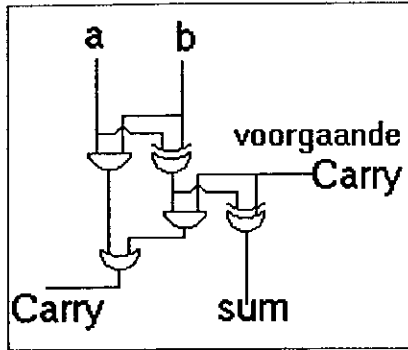
Hiernaast staat een *half adder*. Dat is geen halve slang, maar een basiselement van een teller. Getekend met NAND's, en met AND en EX-OR.

Optellen: ook hier zijn microcontrollers de juiste keuze, maar om te experimenteren en voor de begripsvorming van het optellen kan dit verhaal toch nog interessant zijn. Zoals je ziet is een tekening met AND en EX-OR duidelijk, maar in deel 1 zei ik, dat ik uitsluitend met NAND's zou werken.

Tel a en b op, dan zie je:  
 $a + b = \text{sum} (+ \text{carry})$

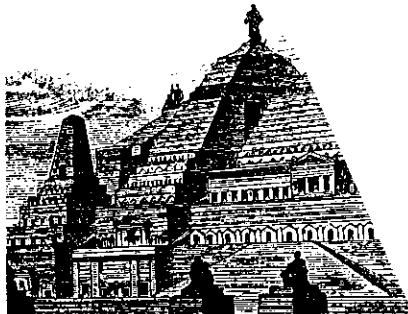
$1 + 0 = 1$
$0 + 1 = 1$
$1 + 1 = 0 \ \& \ \text{carry}$

De naam *half adder* doet al vermoeden dat er ook een *full adder* moet zijn. Welnu, dat is zo, dat zijn twee *half adders* achter elkaar. Hiermee is het mogelijk om meerdere van deze optelelementen achter elkaar te schakelen. Dit apparaat heeft drie ingangen, a, b en c, de carry van de voorgaande optelling. Er zijn twee uitgangen, de som en de carry.

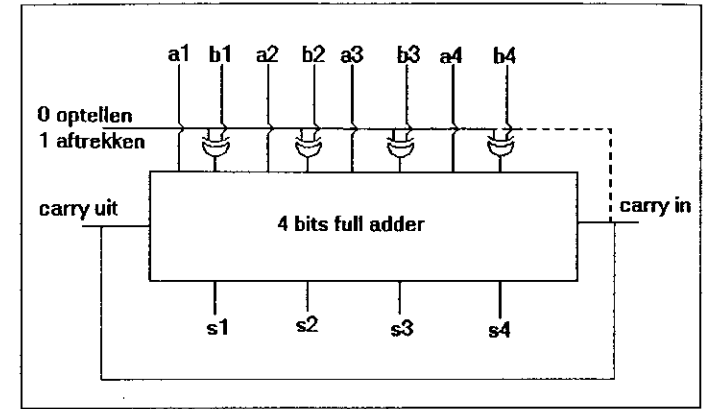


Dit is een bijzonder bouwstelement, want van elke bouwsteen mag je de ingangen uitwisselen, a-b-c, mag je door elkaar gebruiken.

de waarheidstabel
$a + b + c_i = S + C_u$
$0 + 0 + 0 = 0 + 0$
$1 + 0 + 0 = 1 + 0$
$0 + 1 + 0 = 1 + 0$
$1 + 1 + 0 = 0 + 1$
$0 + 0 + 1 = 1 + 0$
$1 + 0 + 1 = 0 + 1$
$0 + 1 + 1 = 0 + 1$
$1 + 1 + 1 = 1 + 1$



Een aantal van deze *adders* bij elkaar vormt een eenheid waarmee kan worden afgetrokken en worden opgeteld. Zelfs de 2-complementsmethode is mogelijk. Het optellen van de 'extra' 1 bij het aftrekken wordt gedaan door de inputcarry 1 te maken. Verder is het toegestaan om ingangen per



poort naar believen te wisselen, dit komt het leggen van printsporen weer ten goede.

Een vierbits optel- en aftrekschakeling. De EXOR invertteert het signaal van het b-kanal, waardoor er wordt afgetrokken. De wat lichtere verbinding tussen de beide carry's is nodig voor de 1-complementscodering. De gestippelde verbinding moet worden gelegd, wanneer de codering 2-complements is.

Tot zo ver voorlopig het digitale geweld. Mocht iemand anders ook eens een verhaaltje willen schrijven, dan is dat natuurlijk niet verboden, sterker nog, het is zeer gewenst als meerdere mensen schrijven. Dat mag best over het zelfde gaan als waarover ik geschreven heb. Een ander zal schrijven vanuit zijn eigen visie en belevingswereld. Dat geeft zondermeer een ander verhaal dat zeer wel mogelijk een beter verhaal zal blijken te zijn. De een snapt het zoals ik het beschrijf, maar voor een ander is een andere uitlegtrant beter. Da's persoonlijk.

Dré Jansen  
 djansen.2@hccnet.nl



# inhoud

## on-line 28



- 3 duimwiel  
uitlezing van een duimwielschakelaar /dré jansen/
- 10 boormachine  
plusbreedteregeling  
toerenregeling voor kleine motoren /dré jansen/
- 14 weerstandenreeksen /dré jansen/
- 20 digitaal verhaal  
deel 1 tot en met 4 /dré jansen/