



Jaargang 30
Februari 2025
Nummer 1

IN DEZE UITGAVE:

INLEIDING

LAATSTE NIEUWTJES

HET NUT VAN RANDAARDE

HOE WERKT EEN PROCESSOR DEEL 2

NIEUWE SOTWARE

MSX & ... DEMO'S!

ATARI NIEUWS

COLOFON EN AGENDA

Inleiding

Inmiddels zitten we alweer in het jaar 2025. Ik denk dat we als club afgelopen jaar wel als het meest succesvolle uit onze historie mogen noemen. Nog nooit hebben zoveel mensen de weg naar 't Schuurtje gevonden en hebben we mogen genieten van allerlei zeer geslaagde activiteiten. We beginnen deze februari clubdag in ieder geval weer prima, met een soldeer workshop. Gelukkig hebben we het onderling erg leuk en zie je zeer veel interactie bij onze bezoekers en clubleden. Dit gaat na de clubdagen vaak verder via social media zoals Facebook en YouTube, maar vooral is er veel actie in onze WhatsApp groepen. Wat zijn dat toch een fijne media als het goed en netjes gebruikt wordt. Helaas is er de laatste jaren wereldwijd een kentering gaande en zie je dat er allerlei vuiligheid gespuid wordt, vooral op X is het af en toe om het schaamrood op je kaken te krijgen. Heel erg jammer, hopelijk komt hier toch



een keer een einde aan en komt er een periode dat we weer op een normale manier met elkaar om zullen gaan.

Gelukkig is dat bij onze clubleden niet het geval en wordt WhatsApp vooral gebruikt om allerlei interessante informatie uit te wisselen. We hebben een C= Koffiecorner en C= Clubdagen groep. Zo is de C= Koffiecorner vooral in

gebruik om te vertellen welke mooie computers je hebt en wat er allemaal mee gedaan wordt, software- maar ook hardware matig. Over het repareren en schoonmaken van de Amiga of Commodore. Over bestellingen die worden gedaan en wat daarvan bevalt of juist niet. Natuurlijk over aanbiedingen die gezien zijn op Marktplaats of Ebay en de onvoorstelbare prijzen die soms worden gevraagd. Maar ook over leuke historische zaken.



Zo was er laatst een foto te zien van een dame in gesponsord Commodore shirt en werd gevraagd wie dat ook alweer was. Bleek te gaan om de tennisster Steffi Graf. In de C= Clubdagen groep zie je vooral discussies over wat men heeft meegemaakt op een clubdag, wat men de volgende keer wil presenteren of demonstreren en wat er de komende clubdag meegenomen wordt. Of vaak nog mooier dat men iets wil meenemen voor een ander.

Mocht je hieraan deel willen nemen laat het dan even aan ons weten, dan kun je worden toegevoegd aan één van de twee groepen. Zag dat dit alweer de 30^e jaargang was van ons onvolprezen clubblad, kan ik best trots op zijn al zeg ik het zelf 😊. Veel plezier vandaag!

Uw voorzitter én redacteur, Ron van Schaik

Laatste nieuwtjes



De HD-64, ontworpen en geproduceerd door SPL, is een moderne modulatorvanger voor de Commodore 64 computer, die een kristalhelder HDMI-beeld creëert, rechtstreeks vanaf een bestaande VIC-II-chip:

<https://www.retro8bitshop.com/product/spl-hd-64/>

Op het YouTube-kanaal Retro Dream kun je een erg leuke video bekijken over de gouden eeuw van de personal computer. De video toont 80 personal computers, van de Apple II in 1977 tot de Amiga 1200 in 1992:

<https://www.youtube.com/watch?v=wQPmPdL5t3Y>



Tynemouth Software schreef een blog over de Mini VIC. Dit project startte in het jaar 2020, maar een van de moeilijkste onderdelen is het vinden van een manier om de VIC-video- (en geluids-)chip te vervangen:

<http://blog.tynemouthsoftware.co.uk/2025/01/mini-vic-revisited-part-1.html>



NEO1541 is een Commodore 1541 emulator voor Linux, ontwikkeld door Domenico Muti. Momenteel kun je PRG-bestanden rechtstreeks vanaf je harde schijf, of vanaf D64-imagebestanden, laden en opslaan, of vanaf D64-imagebestanden. Om het te gebruiken, heb je een

hardware parallele poort en een XM1541-kabel nodig:

<https://github.com/domenicomuti/neo1541>



De Cody Computer is een 8-bits homecomputer die bedoeld is om als doe-het-zelfproject te worden gebouwd. Geïnspireerd door de Commodore homecomputers uit de jaren 80, is het gebouwd rond de 65C02 en 65C22 en de Parallax Propeller

microcontroller. De eigenschappen zijn: gedeeltelijke SID-emulatie, C64 / Plus/4 grafische modi, BASIC-interpreter, seriële communicatie, Atari-stijl joysticks, uitbreidingspoort, een 3D-geprinte behuizing en een toetsenbord:

<https://codycomputer.org/>

Het nut van randaarde

Een logische gedachte is dat randaarde alleen nodig is voor als het echt mis gaat. Het voorkomt dat een metalen kast onder spanning kan komen te



staan als er van binnen in het apparaat iets vreselijk mis gaat. Maar een computer of diskdrive met een kast helemaal gemaakt van plastic waarom heeft die dan een stekker met randaarde, dat is dan toch niet nodig? Nou zeker wel en dat heeft niet zozeer te maken met de

veiligheid maar vooral heel veel met netfilters.

Wat zijn netfilters? Netfilters voorkomen dat stoorsignalen gegenereerd door de voeding van de computer (of de computer zelf) op het elektriciteitsnet terecht komen en andere apparatuur storen. Hoe nuttig deze filters ook zijn, voor de verzamelaar van oude computers zijn ze soms best irritant. Het is niet ongebruikelijk dat een 40 jaar oude machine na een minuutje ingeschakeld te zijn een wat vreemd ruikende rookwolk loslaat, het typische kenmerk van een niet langer functionerend apparaat. De reden hiervoor is de condensatoren in netfilters die defect kunnen raken en dan in vlammen opgaan. Op zich niet gevaarlijk, het is vooral veel rook en stank. In de praktijk wordt zo'n netfilter dan vervangen door een nieuw exemplaar of zelfs in z'n geheel verwijderd. Wie denkt aan het "recappen" van netfilters, vergeet het maar, deze zijn volledig ingeblikt en ingegoten. Maar we dwalen af, dit artikel gaat immers over het nut van randaarde.

De condensatoren in het netfilter (samen met nog wat andere onderdelen) voeren de stoorsignalen af en zijn verbonden met de randaarde. Echter wanneer je de stekker in een stopcontact ZONDER randaarde hebt gestoken, dan kan het zo maar zo zijn dat er via deze condensatoren de halve netspanning op het chassis van je apparaat komt te staan. Dit komt omdat de filtercondensatoren in het netfilter een spanningsdeler vormen die niet correct meer aangesloten is, immers er is geen randaarde aangesloten, dus de spanning op de netfilter wordt niet langer afgevoerd. Dit klinkt heel dramatisch, maar vrees niet, je voelt hier (afhankelijk van de situatie) soms

bijna niets van wanneer je met je vingers de metalen delen van je printerpoort of video connector aanraakt. Maar met een fatsoenlijk multimeter meet je vaak wel degelijk iets. En hier schuilt dan ook het gevaar want als je meerdere apparaten op elkaar gaat aansluiten, zoals een je diskdrive aan de computer en nog een monitortje en printertje en je doet dat als deze apparaten al aan staan, dan zal het spanningsverschil zich bij het aansluiten van de connectoren gaan vereffenen en kunnen er kortstondig toch echt wel stroompjes gaan lopen waardoor er vitale delen stuk zouden kunnen gaan. Tip 1: sluit nooit niets op elkaar aan als de spanning al is ingeschakeld. Tip2: als je persé op een gewoon stopcontact wilt werken, doe dat dan i.c.m. een verdeeldoos met randaarde, op deze manier wordt de randaarde van de diverse apparaten toch goed met elkaar verbonden en lopen de "enge" stroompjes niet over de signaalkabels maar via de randaarde aansluiting van het net snoer via de verdeeldoos. Oké, het netfilter werkt dan nog steeds niet zoals bedoeld maar voor je apparatuur is het toch net ietsje veiliger dan alles zomaar te laten zweven. Het is dan ook goed om te zien dat alle verdeeldozen op de clubdagen in het HCC krat allemaal randaarde hebben. Dit probleem moet men niet verwarren met statische elektriciteit maar de overeenkomsten zijn groot. Of de spanning nu komt van een niet op randaarde aangesloten netfilter of van een computergebruiker met een fleece trui aan op een koude droge winterdag, als er spanningsverschillen zijn dan zullen zich die vereffenen zodra er een elektrisch contact wordt gemaakt.



Een lang verhaal waarvan we eigenlijk maar 1 ding moeten onthouden: stop dingen met een randaarde stekker in een stopcontact met randaarde en als je dat niet kunt en toch meerdere apparaten met elkaar moet verbinden, gebruik dan een verdeeldoos met randaarde en sluit daar alle apparaten op aan. Op deze manier zullen via deze verdeeldoos alle randaarde aansluitingen met elkaar verbonden zijn en zullen via deze weg alle chassis van de aangesloten apparaten hetzelfde potentiaal krijgen.

Artikel geschreven door Jan Derogee

Hoe werkt een processor deel 2

Het binaire stelsel

Op het allerlaagste niveau werken computers puur binair: er loopt stroom door een schakeling of niet. Dit is te vertalen als 1, wel stroom, of 0, geen stroom. Binair dus Wij mensen hanteren het decimale oftewel tientallig stelsel waarbij we de getallen '0' tot en met '9' gebruiken. Hoe werkt het tientallig stelsel? We nemen als voorbeeld het getal 37652:

$$\begin{array}{r} 4 \\ 37652 = 3 \text{ keer } 10^4 = 3 \text{ keer } 10000 \\ 3 \\ 7 \text{ keer } 10^3 = 7 \text{ keer } 1000 \quad + \\ 2 \\ 6 \text{ keer } 10^2 = 6 \text{ keer } 100 \quad + \\ 1 \\ 5 \text{ keer } 10^1 = 5 \text{ keer } 10 \quad + \\ 0 \\ 2 \text{ keer } 10^0 = 2 \text{ keer } 1 \quad + \end{array}$$

De plaats van een getal in het geheel bepaalt dus de toegekende waarde aan een getal. Wat meer wiskundig gezien: de exponent van het basisgetal. Het binaire stelsel mag dan maar twee getallen kennen, de '0' en de '1', maar dat belet ons niet om er getallen mee te kunnen maken. Neem nu het binaire getal 11011:

$$\begin{array}{r} 4 \\ 11011 = 1 \text{ keer } 2^4 = 1 \text{ keer } 16 \\ 3 \\ 1 \text{ keer } 2^3 = 1 \text{ keer } 8 \quad + \\ 2 \\ 0 \text{ keer } 2^2 = 0 \text{ keer } 4 \quad + \\ 1 \\ 1 \text{ keer } 2^1 = 1 \text{ keer } 2 \quad + \\ 0 \\ 1 \text{ keer } 2^0 = 1 \text{ keer } 1 \quad + \\ \text{----} + \\ 27 \end{array}$$

Ook hier bepaalt de plaats van een getal in het geheel de exponent van het basisgetal. Het basisgetal is in dit geval 2. Dus 11011 is de binaire versie van 27. Wat is de binaire versie van 37652? Dat is: 1001001100010100. Eerlijk gezegd, dat heb ik met de calculator op mijn PC uitgerekend. Dit is ook met de hand uit te rekenen maar die specifieke kennis is voor de rest van het verhaal niet nodig.

Hexadecimale getallen, nibbles, bytes, words en double-words

Tot nu toe hebben we het maar met een paar nulletjes en eentjes te maken gehad. Maar de Pentium is een 64-bits processor en dat betekent dat hij getallen hanteert die uit 64 nulletjes en/of eentjes bestaat. Hoe goed hij of zij ook was, zelfs de beste techneuten konden hier weinig mee aanvangen en daarom werd al snel besloten ze in groepjes van vier nullen en/of enen te combineren tot hexadecimale getallen (na een hele korte overstap naar het octallige stelsel bestaande uit groepjes van drie bits). Dit zijn getallen die op het zestientallig stelsel gebaseerd. Om die zestien getallen te kunnen weergeven heeft men ons eigen tientallig systeem aangevuld met de letters A t/m F.

| | | |
|------|---|---|
| 0000 | - | 0 |
| 0001 | - | 1 |
| 0010 | - | 2 |
| 0011 | - | 3 |
| 0100 | - | 4 |
| 0101 | - | 5 |
| 0110 | - | 6 |
| 0111 | - | 7 |
| 1000 | - | 8 |
| 1001 | - | 9 |
| 1010 | - | A |
| 1011 | - | B |
| 1100 | - | C |
| 1101 | - | D |
| 1110 | - | E |
| 1111 | - | F |

Om het verschil te zien tussen hexadecimale en decimale getallen bestaan er (helaas) meerdere methodes. De drie meest gebruikte zijn:

- het getal begint met een '\$'(6502, Motorola, Turbo Pascal, Free Pascal)
- het getal eindigt op een 'H' of 'h' maar begint wel altijd met een getal, desnoods een '0' (Intel, Zilog)
- het getal begint met 0x (C, Unix, Linux)

$\$92 = 92H = 92h = 0x92 = 146$

$\$A3 = 0A3H = 0A3h = 0xA3 = 163$

In bepaalde gevallen is het gebruik van binaire getallen wel wenselijk. Deze worden door een '%' vooraf gegaan:

$\$A3 = 0A3h = 0xA3 = \%10100011 = 163$

Twee nibbles, samen goed voor acht bits, vormen samen een byte. De meest bekende processoren uit de jaren '70 waren 8-bitters: 6800, 6502 en Z80. Om die reden is de C64 dus ook een zogenaamde '8 bitter' (de 6510 is een afgeleide van de 6502). Twee bytes vormen een word (dus 16 bits). In het begin van de jaren '80 kwamen de 16-bits processoren uit waaronder de Intel 8086 en Motorola 68000. Bekende namen die deze processoren gebruikten: IBM PC, Amiga en Apple McIntosh. Twee words vormen weer double-word (32 bits). In het midden van de jaren '80' kwamen de 32-bits processoren uit: Motorola 68020, Intel 80386, Zilog Z8000. Tegenwoordig kennen we ook al de 64-bitters waarvan de Intel Pentium (vanaf versie 2) de meest bekende is.

Een praktisch gebruik van hexadecimale getallen: een stukje programma

Er is nu wel één ding duidelijk: de computer, lees: processor, werkt met getallen en alleen maar getallen. Maar ook al hebben we al die nullen en eentjes naar hexadecimale getallen omgezet, eigenlijk kunnen we er als mensen nog weinig mee.

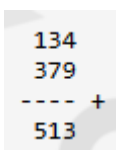
| adres | instructie | opcode | operand | betekenis |
|-------|------------|--------|-----------|-----------------------|
| E00F | BD 0000 | mov | bp,0 | |
| E012 | FA | cli | | ; Disable interrupts |
| E013 | B4 D5 | mov | ah,0D5h | |
| E015 | 9E | sahf | | ; Store ah into flags |
| E016 | 73 50 | jnc | loc_6 | ; Jump if carry=0 |
| E018 | 75 4E | jnz | loc_6 | ; Jump if not zero |
| E01A | 7B 4C | jnp | loc_6 | ; Jump if not parity |
| E01C | 79 4A | jns | loc_6 | ; Jump if not sign |
| | | | | |
| FFF0 | EA 0FE00F0 | jmp | F000:E00F | |

Bovenstaande code is afkomstig uit de BIOS van een PC uit de jaren '80. De eerste kolom geeft het adres weer met daar achter de instructie. En zoals je ziet bestaat die alleen uit getallen. En omdat de betekenis van die getallen niet gemakkelijk te onthouden was, sterker, bij de eerste computers veranderde die getallen wel eens, is men die codewoorden in de derde en vierde kolom gaan gebruiken. In het Engels werd dit de "assembly language" genoemd. Vaak gebruikt men ook alleen het woord "assembly". In het Nederlands gebruikt men ook vaak de term "machinetaal" (Engel: machine language) maar strikt genomen zijn de getallen uit de tweede kolom machine taal. Aangezien er praktisch niemand is die rechtstreeks met de getallen werkt, mogen we rustig aannemen dat als iemand het over machinetaal heeft, hij assembly bedoelt. Op Engelstalige forums is men zich iets beter bewust van het verschil maar gebruikt men toch vaak de afkorting ML van "Machine Language" ipv. assembly om de doodeenvoudige reden dat ML minder schrijfwerk is.

Terugkomend op de adresbus in bovenstaand stukje programma: op adres 0FFF0h staat een sprongopdracht naar 0E00Fh. Maar vanaf dat adres wordt het adres elke keer met één verhoogd. Dit is misschien niet meteen zichtbaar omdat een regel meerdere bytes kan bevatten welke elk op een eigen adres staan. Dit is vanaf adres E012 beter te zien.

Optellen

We weten dat computers en calculators allemaal kunnen rekenen. Wij mensen kunnen dat ook maar, hoe doen wij dat eigenlijk?



De basisregels voor decimaal optellen zijn bekend: 9 plus 4 is 13. 3 opschrijven, 1 onthouden. 7 plus 3 is 10. Plus de onthouden 1 maakt 11. Etc., etc. In ons 10-talig stelsel hebben wij twintig mogelijkheden om een getal op te tellen. Maar een computer is van huis uit binair en kent daarom maar vier mogelijkheden:

| | | | |
|-----|---|-----|----|
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| --- | + | --- | + |
| 0 | 1 | 1 | 10 |

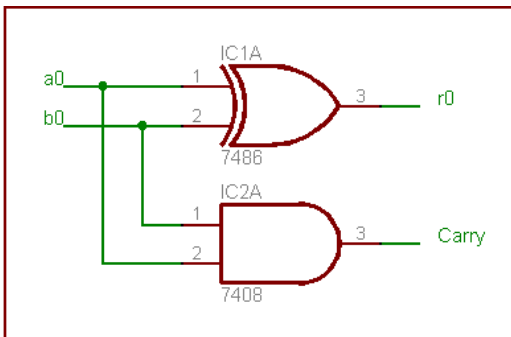
Bij binair optellen is "1 plus 1 is 10", dus 0 opschrijven en 1 onthouden. Die "1 onthouden" heet in computertermen "Carry". Bekijken we het binair optellen eens vanuit een iets andere hoek:

| | | | | |
|------------|-----|---|-----|---|
| Bit 1: | 0 | 1 | 0 | 1 |
| Bit 2: | 0 | 0 | 1 | 1 |
| Resultaat: | --- | + | --- | + |
| Carry: | 0 | 0 | 0 | 1 |

En nu zetten we dit om in tabellen:

| Bit 1 | Bit 2 | Resultaat | Bit 1 | Bit 2 | Carry |
|-------|-------|-----------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |

Komen deze tabellen niet bekend voor? Het ziet er naar uit dat optellen niets anders is dan een EXOR operatie voor het resultaat en een AND operatie voor de Carry. Met het volgende **logische** circuit kunnen we dus twee bitjes optellen:



Ik heb het woord 'logische' bewust vet gemaakt om bewust een beetje afstand te nemen van de hardware. Ik heb al eerder bewezen dat een poort zowel met behulp van transistors als relais is te bouwen. Maar dat hadden ook buizen kunnen zijn of mechanische schakelaars (is echt geprobeerd!). Van nu af aan is de achterliggende elektronica helemaal niet spannend meer.

We gaan een stap verder en we maken een telmachine die getallen van twee bits kan optellen. Maar nu moeten we na het optellen van alle losse bitjes nog wel wat doen met de resultaten en Carry's:

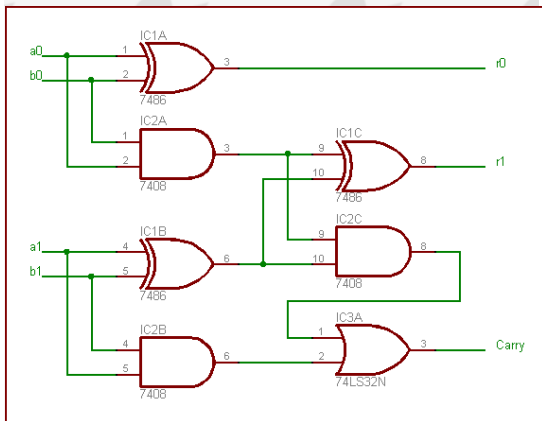
| Carry 1 | Result 2 | Carry 2 | Result 2b | Carry 2b |
|---------|----------|---------|-----------|------------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | Onmogelijk |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | Onmogelijk |

Toelichting: het optellen van twee bits kan er nooit toe leiden dat zowel het resultaat als de Carry 1 zijn. Vandaar de twee regels met de tekst 'Onmogelijk'.

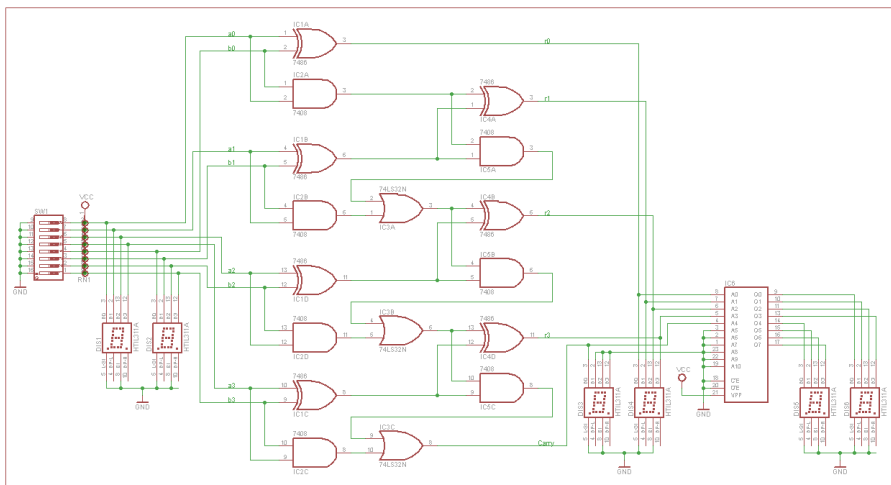
Bovenstaande tabel is weer te vertalen in:

$$\text{Result 2b} = \text{Carry 1 EXOR Result 2}$$

$$\text{Carry 2b} = (\text{Carry 1 AND Result 2}) \text{ OR Carry 2}$$



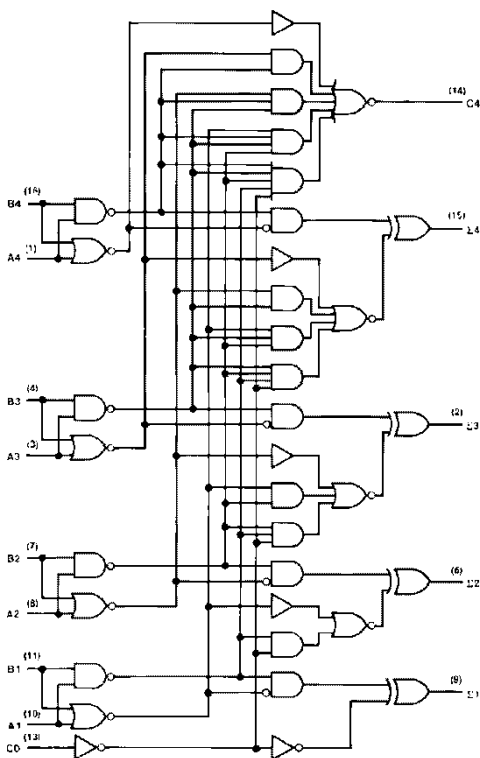
Voor een multi-bit opteller hoeven we dus alleen maar genoeg van deze "optellers" achter elkaar te schakelen. Bij deze het schema van een praktijkvoorbeeld:



Kunnen we nu een 64-bits opteller bouwen zoals gebruikt in een Pentium 4? Ja en nee. Bovenstaand circuit is zeker gebruikt toen de eerste computers nog met relais (Zuse, jaren '30), buizen (jaren '40 en '50) en losse transistors (jaren '60) gebouwd werden. Zelfs bij de eerste 8-bits processoren werd dit circuit nog gebruikt.

Stel je nu eens voor dat het circuit is opgebouwd uit relais. Als we de schakeling nu vragen de waarden 1 en \$FFFF op te tellen, dan leveren de eerste twee bits een Carry op. Die Carry plus de '1' van het tweede paar bits leveren op hu beurt weer een Carry op voor het volgende paar. Zo door redenerend weten we dat dus door de hele schakeling de Carry moet worden geactiveerd. Het omklappen van een relais kost een beetje tijd en die tijd moet het tweede stel bits wachten. Het derde stel moet op zijn beurt op het tweede stel wachten. Zo door redenerend moet het 64e stel in een 64-bits opteller dus wel "heel lang" wachten. Als gebruiker zie je bij het gebruik van relais het hele gebeuren gewoon als een treintje door de schakeling heen racen. Nu is een transistor uiteraard veel sneller dan een relais maar dat verandert niets aan het feit dat de hele schakeling, relatief gezien, lang moet wachten. Toch heeft de 64-bits Pentium er weinig last van. Hoe is dat nu mogelijk?

Zoals wij tabellen hebben gebruikt om de 1-bits en 2 bits versie te "berekenen", is dat ook



mogelijk voor de 4-bits versie. Maar dit wordt een tabel met acht ingangen en dus 256 mogelijkheden. Dat is niet echt meer met de hand te berekenen, dat moet je aan een computer overlaten. Texas Instruments was zo vriendelijk om in hun databook het schema van hun "7483 4-bit full adder" af te drukken. Deze schakeling heeft veel en veel meer poorten nodig als die van ons maar heeft het voordeel dat deze van input naar output maar vier poorten diep is. Een 8-bits uitvoering heeft nog meer poorten nodig maar zou evengoed maar vier poorten diep zijn.

Opmerking: Als je goed kijkt zie je dat de 7483 een Carry ingang heeft. Dit maakt het mogelijk om meerdere 7483s achter elkaar te plaatsen. De Carry ingang van de eerste zou je gewoon aan de '0' kunnen hangen. Maar we zullen zien dat deze ingang ook nog een tweede functie kan hebben.

Aftrekken, vermenigvuldigen, delen et cetera.

We hebben nu gezien dat we met gewone logica kunnen optellen en dat dus beslist geen abracadabra is. Ik kan nu een lang verhaal hoe we dit met aftrekken, vermenigvuldigen en delen kunnen realiseren maar ik denk dat jullie mij gewoon op mijn woord moeten geloven.

Aftrekken

De eerste gedachte is om een aftrekcircuit te ontwerpen zoals dat met de opteller hebben gedaan. De volgende bewerking zouden we dan kunnen doen we met behulp van deze 16-bit "aftrekker":

```
0000110110100110      3494
0000100010110110      2230
----- -
0000010011110000      1264
```

Nu doen we het nog een keer, maar op een iets andere manier:

```
0000110110100110      3494
1111011101001010      - 2230
----- +
0000010011110000      1264
```

In plaats van het ene getal van het andere af te trekken, tellen we een positief en negatief getal bij elkaar op.

Hoe ziet een negatief getal er uit? De afspraak is dat dan het meest linkse bit, (Engels: MSB, Most Significant Bit) altijd 1 is. Jammer genoeg is het niet alleen een kwestie van dat hoogste bitje 1 maken. Bekijk het volgende voorbeeld:

```
00000001      1
10000001      - 1
----- +
10000010      0
```

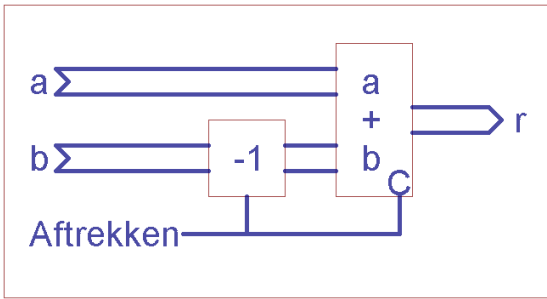
Ehhh, wat %10000010 ook mag zijn, het is beslist geen nul. Dan maar meteen een voorbeeld wat wel correct is:

```
00000010      2
11111110      - 2
----- +
(1)00000000      0
```

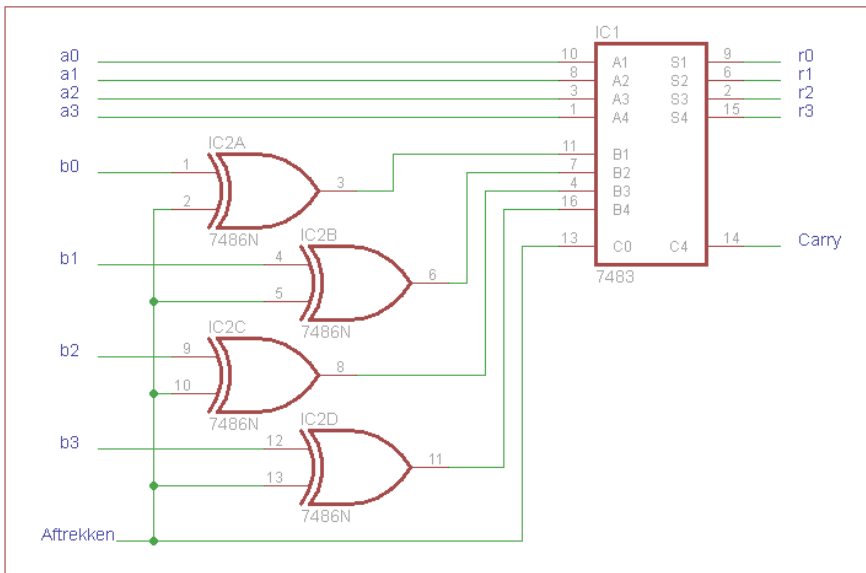
Bij bovenstaand voorbeeld wordt gebruik gemaakt van het zogenaamde "2-complement" van een getal. Dit 2-compliment verkrijg je door alle bitjes van een getal te inverteren en vervolgens 1 bij het resultaat op te tellen.

```
00000010      2
* -1
-----
11111101
  1
----- +
11111110      -2
```

Dus om het ene getal van een ander af te trekken hebben we een handje vol inverters nodig en nog een manier om de waarde '1' bij het resultaat op te tellen. Maar is het niet mogelijk om het zo aan te pakken zodat we voor zowel het aftrekken als optellen maar één circuit nodig hebben? Ja, dat kan:



Het idee is dat het blokje waar "-1" in staat al dan niet de vier aangeboden bits inverteert. Als er al geïnverteerd moet worden, moet er '1' bij het resultaat opgeteld worden en dat wordt heel simpel gerealiseerd door de Carry ingang '1' te maken. De praktijk:



Voor de zekerheid volgt nog eens de waarheidstabel voor de EXOR poort maar dan toegespitst op deze specifieke situatie:

| Inverteren | Bin | Bout |
|------------|-----|------|
| nee = 0 | 0 | 0 |
| nee = 0 | 1 | 1 |
| ja = 1 | 0 | 1 |
| ja = 1 | 1 | 0 |

EXOR

Ga voor je zelf na of de EXOR poort doet wat ik heb geschreven.

Opmerking: als het hoogste bit van een getal '1' is wil dat niet automatisch zeggen dat dit getal dus negatief is. In de programmeertaal Pascal kan ik een variabele definiëren als zijnde van het type 'word' of als zijn de van het type 'integer'. In het eerste geval kan de variabele de waarde 0 t/m 65535 (of \$0000 t/m \$FFFF) hebben, in het tweede geval de waarde -32768 t/m 32767.

```
var
  i1 : integer;
  w1 : word;

begin
  w1 := $1000;
  i1 := w1;
  writeln('w1 = ', w1, '      i1 = ', i1);

  w1 := $F000;
  i1 := w1;
  writeln('w1 = ', w1, '      i1 = ', i1);
end.
```

Zou je dit Pascal programma uitvoeren dan staat het volgende op het scherm:

```
w1 = 4096      i1 = 4096
w1 = 61440     i1 = -4096
```

Maar hoe weet de processor nu van deze afspraak af? Niet dus, de afspraak ligt puur op software niveau. Op de geheugenplaats die voor i1 is gereserveerd staat ook de waarde \$F000. Pas op het moment dat i1 zichtbaar moet worden gemaakt, dan gaat Pascal kijken of er een bepaalde afspraak over i1 is gemaakt. Op de keper beschouwd doet Pascal dat eigenlijk ook met w1 want zoals je boven kunt zien laat Pascal de decimale waarde zien en niet de hexadecimale waarde.

Het is ook de software die achteraf moet controleren of de optelling correct is verlopen. Als ik twee negatieve getallen optel en het resultaat is positief, zal ik softwarematig een en ander moeten corrigeren.

Vermenigvuldigen

```
   111      getal A      111
   111      getal B      101
----- *
   111
  1110
 11100
----- +
110001

   111      [1]
   111
  0000
 11100
----- +
100011
```

Bovenstaande voorbeelden laten zien dat er eigenlijk geen verschil is tussen binair en decimaal vermenigvuldigen. OK, de rij met [1] gemerkt zouden we normalerwijze niet

opschrijven maar die hebben we nodig voor de latere uitleg.

Net als bij decimaal vermenigvuldigen, moeten we de rij elke keer één positie naar links schuiven (Engels: shift). Ook dit kan elektronisch, waarover later meer.

De eerste processoren die konden vermenigvuldigen gingen als volgt te werk:

| | | | |
|--------|-----------------|------------|--|
| 111 | getal A | | |
| 111 | getal B | | |
| ----- | * | | |
| 111 | register 1 | | |
| 1110 | register 2 | | |
| ----- | + | | |
| 10101 | register 3 -> 1 | register 3 | |
| 11100 | register 2 | register 4 | |
| ----- | + | | |
| 110001 | register 3 | register 5 | |

Eerst wordt de vermenigvuldiging opgedeeld in losse optellingen. De eerste twee waardes worden naar register 1 en 2 gekopieerd waarna de waarde in register 2 één keer wordt geshift. Register 1 en 2 worden nu opgeteld en het resultaat komt nu in register 3. De inhoud van register 3 wordt naar register 1 gekopieerd, register 2 wordt weer gevuld maar nu wordt er twee keer geshift, en de boel wordt weer opgeteld. Dit kopiëren en shiften gebeurt zo vaak als er bitjes zijn (minus 1). Dit is de manier zoals het in de 6809 en 8088/8086 gebeurt. Er zijn processoren die van huis uit niet kunnen vermenigvuldigen, zoals de 6502 en Z80. Maar dat wil niet zeggen dat er dus helemaal niet vermenigvuldigd kan worden. De vermenigvuldiging geschiedt dan door het uitvoeren van losse assembler opdrachten en in plaats van registers werd er geheugen van de computer als (tijdelijke) opslag gebruikt.

Bovenstaande kost wel tijd. De volgende generatie processoren plaatste gewoon een hele batterij aan registers achter elkaar (zie de kolom met de registers 3, 4 en 5). Het shiften is niet eens nodig; de waarde wordt meteen met de juiste hoeveelheid nulletjes er achter in een register gekopieerd. Dan is het een kwestie van alles optellen. Dit gebeurt in de 80386, 80486 en nieuwer.

Er is nog een derde, theoretische, manier en dat is door de uitkomst dmv. tabellen te berekenen. Softwarematig wordt dit wel vaker gedaan. Ik weet dat dit ook in hardware mogelijk is, bijvoorbeeld in FPGA's. Maar of dit ook in processoren gebeurt, ik weet het niet. Maar zie verderop. Kan het misschien nog anders? Ook hier weer: ik weet het niet. Maar denk dan wel aan het bekende gezegde: er leiden veel wegen naar Rome.

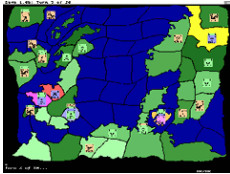
Delen

Wetende dat vermenigvuldigen mogelijk is, is het niet meer dan logisch om te veronderstellen dat delen ook tot de mogelijkheden hoort. Maar de behandeling van dit onderwerp op digitaal niveau kost naar verhouding te veel tijd.

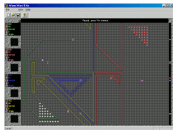
Een opmerking is wel op zijn plaats: een processor kan niet met breuken omgaan. Ik weet dat de Intel 8086 eerst het getal voor de komma en dan alleen de rest als resultaat weergeeft. 35 / 10 levert dus niet 3,5 maar "3 rest 5" op.

Artikel geschreven door Ruud Baltissen, volgende keer deel 3

Nieuwe software



Saga is een conversie van het TSR-bordspel "Saga: Age of Heroes" uit het jaar 1981, voor maximaal 6 spelers. Spelers kunnen worden bestuurd door een mens of de Amiga. In Saga moet je kwaadaardige monsters verslaan, fantastische schatten winnen en over grote koninkrijken heersen: <https://aminet.net/package/game/board/Saga>

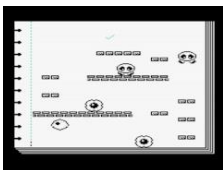


WormWars is een arcadespel voor de Amiga-computer ontwikkeld door James Jacobs. De eigenschappen zijn: 37 objecten, 37 wezens, 13 bonussen, 4 wormen en een level editor: <https://aminet.net/package/game/actio/WormWars>

Microsonix 001 is een demo/muziekdoos voor de Commodore Plus/4 computer. De muziekdoos is ontwikkeld door Mermaid en is een muziekcollectie met acht SID-melodieën die een aantal suggestieve soundscapes produceren:



https://plus4world.powweb.com/software/Microsonix_001



Doodlebug is een nieuw spel voor de Commodore VIC20 computer. Het originele spel is ontwikkeld door Dave Hughes, en de VIC20 versie is gemaakt door Team Moritz met behulp van MPAGD (Multi Platform Arcade Game Designer). Het spel vereist 35 kB geheugen:

<https://kamaleon70.itch.io/doodlebug-vic-20>



Berefth is een nieuw avonturenspel voor de Commodore 64 computer, ontwikkeld door ComSha (code & pixels), Razy (muziek), Logiker (hulp) en Wil (hulp). Je bent een onderzoeker, maar iemand is ingebroken in je huis en heeft al het bewijsmateriaal gestolen van een zaak waar je aan werkt. Het is jouw verantwoordelijkheid om het bewijsmateriaal terug te halen:

<https://comsha.itch.io/berefth>

MSX & ... demo's!

De MSX *demoscene* was – zeker in mijn 'venstertje' van de tijd – erg klein. Zoals eigenlijk alle scenes ontstond deze vanuit copiers/hackers, met cracktro's en intro's. Namen die ik me daarvan herinner zijn The Boss (Nederlands), SMA International (Frans), Martos (Spaans) en Starcracks (weer Nederlands).

Op <https://www.pouet.net> staan zo'n 200 demo's voor de diverse MSX-modellen. Vóór '95 echter maar 30 – en dat is inclusief enkele previews en cracktro's! Twee ervan zijn voor MSX2+, twee voor Turbo-R; de rest MSX2. <https://www.generation-msx.nl> toont 266 *kind* "demo" matches, echter komt veel daarvan niet uit de demoscene; aan die selectie hebben we niet zoveel.

De effecten in de vroege MSX demo's zijn bekende, zoals sinus tekst scrollers (*heel veel* tekst scrollers!), colorbars, wobblers en (bescheiden) 3D vector graphics. De meeste demo's, zeker megademo's, gebruikten de MSX2 of hoger. De MSX2 videochip ondersteunt palettes en horizontal line interrupts, waarmee veel effecten te maken zijn. Pas na 'mijn' tijd, eind jaren '90, is de MSX1 herontdekt voor het maken van demo's. Die revival kwam met name vanuit een onverwachte hoek: Finland. Effecten daarin zijn o.a. tunnels, plasma's & character-based animaties.

Een onverwachte afwijking van andere scenes is dat de meeste demo's voor de MSX *verkoch*t werden. Zo herinner ik me nog Dawn of Time van Station gekocht te hebben. Wat een demo!

Een andere afwijking, die ik me eigenlijk nu pas besef, is dat bij veel demo's de graphics en/of muziek uit games geript werd. Daarnaast waren er dan wel veel veel zogeheten "*music disks*" & "*picture disks*" te koop; de artiesten waren er wel, alleen niet altijd verenigd in een demogroup.

Metal Limit begint met hercreatie van Metal Gear, waarin je elke keer minutenlang moet spelen om de demo parts te bereiken. Creatief, echter wellicht niet wat je verwacht als je een demo start...

Het aantal megademo's (sommige meerdere 720k floppy's groot) is zeer bescheiden; veel producties waren of bevatten 'homebrew' game previews (zoals Metal Limit hierboven), BBS promo's en andere kleine releases. Ook vond je veel klein werk op disk magazines. Die trokken niet echt mijn aandacht; veel daarvan is aan mij voorbij gegaan.

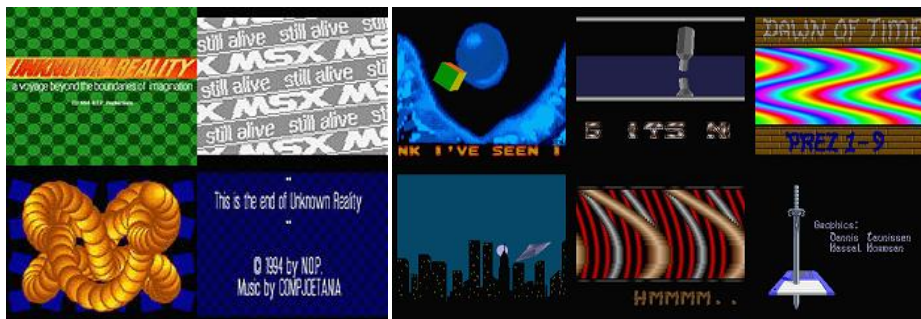
Tijdens onderzoek voor dit artikel las ik tot mijn verbazing dat veel demo's X-BASIC gebruikten. Dat is een compiler voor MSX-BASIC, waarmee je in de buurt kan komen van snelheden van software die in (Z80) assembly is geschreven. Hmm, was de MSX-demoscene toch minder *hard core* dan ik destijds voor ogen had!

Picture disks hebben me nooit zo bekoord; music disks dan weer wel. Die had ik behoorlijk veel. Mijn Philips Music Module was behoorlijk geschikt voor het ten gehore brengen van de toen startende elektronische muziekstijl, waaronder natuurlijk Eurodance – ook veel samples & melodieën van top-10 hits werden geript!

De MSX was bovenmatig populair in Nederland, niet verwonderlijk dus dat veel demogroups daar vandaan kwamen: ANMA, Compjoetania, FAC (ik zal de afkorting uit respect voor onze gastheer niet voluit schrijven), Fony, Fuzzy Logic, Impact, IOD, N.O.P., Station – allemaal Nederlandse demogroups.

Deze vijf demo's voor MSX2 zijn sowieso een aanrader om eens te kijken:

- Unknown Reality, NOP
- Metal Limit, IOD
- Source of Power, ANMA (onderdeel van hun Amusement Disk, jawel, game previews!)
- Dawn of Time, Station
- Calculus, Compjoetania TNG (V9990)



Unknown Reality - N.O.P. – 1994

Dawn of Time - Station - 1992

MSX & ... – mijn herinneringen van de (MSX-)wereld van eind '80 tot halverwege '90. Ik hoor graag hoe jij je die wereld herinnert! Vertel het me op een clubdag.

Artikel geschreven door Hedzer Westra

Atari nieuws (door Dennis Roos)



Pluspunt

De 7800+ is inmiddels overal beschikbaar. De meegeleverde cx-78+ draadloze controller werkt uitstekend (en is bruikbaar op bijna alles dankzij de bijgesloten DB9 en USB dongel). Je kunt op de 7800+, net als met de oorspronkelijke console, Atari 2600 spellen spelen. En als je in het bezit bent van de Dragonfly multicart kun je bijvoorbeeld een spel als Rikki & Vikki ook op de 7800+ draaien, aldus een

betrouwbare bron. Het is dan wel zaak dat je console met een recente firmware draait. Er gonsen geruchten over een nog uit te brengen arcade spel voor de 7800+, maar details zijn nog niet bekendgemaakt. We houden het voor jullie in de gaten.

Gevonden voorwerpen

In 2025 zou je toch verwachten dat alles aan Atari spellen wel gevonden is, maar toch duikt er zo nu en dan weer iets op. Zo vond Dutchman2000 op een backup tape van een van Atari's VAXen het spelletje fish. Meer informatie vind je terug op de atariprotos.com website



(<https://tinyurl.com/2600fish>). Voor de Atari 8-bit home computer is Bubble Bobble eindelijk af en voor fans van het Rogue genre is Rogül een leuke uitdaging. Een demo van Block'Em Sock'Em is verkrijgbaar voor zowel de home computer als voor de 5200 console. Ook voor beide systemen is een Mario-achtige platformer die zomaar uit de lucht kwam vallen eind 2024, nu nog een 'tech demo' maar wel een met heel veel potentie. Het Lynx spel, Grub-O-Grotto dat was geschreven voor LynxJam2024 is voltooid in de loop van vorig jaar en is nu te downloaden, net als een level editor. Deze en andere titels vind je grotendeels ook terug in de lijsten van Zero Page Homebrew op het Atari Age forum: <https://tinyurl.com/2025ZPH2600>



Emulatie

De emulator A8DS voor de Nintendo 3DS is begin dit jaar bijgewerkt en de laatste release staat op github

(<https://tinyurl.com/WMA8DS>). De Pokey vervanger (alleen audio) genaamd Hokey is ook weer leverbaar. De advanced versie laat nog even op zich wachten. Niet helemaal Atari gerelateerd, maar zeker het vermelden

waard is de MiSTer Pi. Een MiSTer compatible HAT voor de Raspberry Pi (<https://tinyurl.com/MiSTerPis>). Helaas heeft de campagne voor een nieuwe 8-bit behuizing niet voldoende opgebracht, een nieuwe campagne volgt nog.

Colofon

Bestuur:

Ron van Schaik, voorzitter
Marien Kaptein, penningmeester
Robert Sprokholt, secretaris

Kernleden:

Hans Kessels
Ruud Baltissen
Antony Mo
Erwin van Betten
Jeroen Vlasveld

Kidscorner:

Rinus van de Reep
Youri Wagenaar

Redactie:

Ron van Schaik

Website:

<https://www.commodoreclub.nl/>
<http://twitter.com/commodoreig>
<http://www.youtube.com/commodoregg>
<http://www.facebook.com/commodoreclubnederland>
<https://www.instagram.com/commodoreclubnederland>
Email: bestuur@commodore.hcc.nl

Dit magazine verschijnt twee
maandelijks op de derde zaterdag van
de even maand.

Agenda

21 – 23 februari 2025 BCC Party #19
Demo Party, 8 bit Party
Rungestraße 20, Berlijn, Duitsland
<https://csdb.dk/event/?id=3466>

14 – 16 maart 2025 Forever 2025
Multi Platform 8-bit Computer Party
Trnava, Slowakije
<https://forever.zeroteam.sk/>

21 - 23 maart 2025 Fioniadata 2025
Multi Platform 8-bit Computer Party
Langeskov, Denemarken
<http://fioniadata.dk/>

19 april 2025 10:00 – 16:00 uur
Clubdag in 't Schuurtje, Maarsse

21 juni 2025 10:00 – 16:00 uur
Clubdag in 't Schuurtje, Maarsse

