hcc Commodore interessegroep

Info Bulletin

Jaargang 30

Juni 2025

Nummer 3

IN DEZE UITGAVE:

Inleiding

In het kader van het 30-jarig bestaan van ons clubblad doe ik af en toe een terugblik. De vorige keer kwam Commie eenmalig terug om onze problemen te bespreken en waar mogelijk op te lossen of een tip te geven. We hadden het op de clubdag van april over de

HCC-dagen mede door het feit dat het telkens zo ontzettend druk is in 't Schuurtje en er geopperd werd om de Jaarbeurs maar weer af te huren. Niet als HCC organisatie maar deze keer met onze Commodore, Amiga, MSX en Atari fanaten. Voor diegene die niet precies weten wat de HCC-dagen inhielden, in de jaren 80 organiseerde HCC jaarlijks de HCC-dagen, een evenement waarbij leden en geïnteresseerden samenkwamen om workshops te volgen, presentaties bij te wonen, onderdelen voor hun computers te kopen en om de nieuwste ontwikkelingen op het gebied van informatietechnologie te ontdekken. De opkomst was bij deze dagen vaak zo massaal dat rond Utrecht speciale verkeersmaatregelen moesten worden genomen om de toestroom te regelen. Ruim 110.000 mensen bezochten dan één van de drie HCC-dagen. De leden kregen gratis entreekaarten die met het Computer!Totaal magazine werden meegestuurd.

Een heel mooi filmpje over de HCC-dagen van het NOS journaal in 1986 kun je hier vinden: https://www.youtube.com/watch?v=DP73zDB0JsE

Dit filmpje van het NOS journaal uit 1986 laat zien hoe populair de HCC-dagen destijds waren. Het was zo druk op het evenement dat het verkeer in Utrecht vast kwam te staan en omgeleid moest worden. Mensen stonden in lange rijen te wachten om naar binnen te mogen en de verschillende stands te bezoeken. Het was een teken van het succes en de groei van HCC, die uitgroeide tot de grootste en meest invloedrijke computerclub in Nederland in die jaren.

De HCC-dagen waren niet alleen een plek om nieuwe technologieën te ontdekken, maar ook om gelijkgestemde mensen te ontmoeten en ervaringen uit te wisselen. Het was een bruisende en inspirerende omgeving waar leden en experts elkaar konden ontmoeten en van elkaar konden leren. De HCC-dagen boden ook een platform voor bedrijven en organisaties om hun nieuwste producten en diensten te presenteren aan een enthousiast en geïnteresseerd publiek. Grappige anekdote is wel dat het meest verkochte Item een goedkoop steekwagentje was om de dozen met aankopen te vervoeren. Meestal haalde je net het station Utrecht er mee……

Ik heb er zelf een heel aantal keren gestaan, we hadden een eigen Commodore/Amiga stand. We hebben dat heel wat jaren volgehouden, oudgedienden die ik me nog kan herinneren en er altijd bij waren zijn Wim Bonke, Jan Klooster, Ruud Baltissen en Hans Kessels. Zelf heb ik foto's kunnen vinden vanaf 2001 tot en met 2006. In een bericht wat ik vond van RTVUtrecht stond dat de HCC-dagen van november 2010 waren afgelast vanwege gebrek aan exposanten. Dus ik neem aan dat de beurs van 2009 de laatste is geweest. Daarna zijn er nog wel wat beurzen geweest in een afgeslankte vorm, daar hebben ook nog Marien, Robert en nog wel enkele leden (weet niet meer precies wie, excuus) van ons aan meegedaan. Nu zijn er nog de HCC!Kennisdagen en de CompUfair, aan die eerste doen we met de HCC!Commodore ook regelmatig mee.

Natuurlijk heb ik er zelf ook weleens wat gekocht op de HCC-dagen, zoals een Amiga 500. Ik kan me nog goed herinneren dat deze in dozen stond opgestapeld tot misschien wel vijf meter hoog. Ze gingen als warme broodjes over de toonbank. Prijzen werden constant aangepast en de concurrerende bedrijven keken om het uur bij elkaar hoe de hardware geprijsd was. In het laatste uur van de drie beursdagen was het helemaal een spektakel. Veel bedrijven hadden dan geen zin om nog veel spul mee terug te nemen dus op dat moment werd er heel veel voor spotprijzen verkocht.

Wat ik me verder nog kan herinneren is dat er mensen bij onze stand voorbij kwamen met een steekwagentje vol met de allernieuwste PC, hardware en software en dan deze spullen aan de kant zetten om bij ons Bouderdash te spelen op een Commodore 64!

Regelmatig zeggen mensen tegen me dat ze het zo jammer vinden dat er geen HCC-dagen meer zijn. Maar met de opkomst van de online verkoop viel één van de belangrijkste argumenten om naar de jaarbeurs te gaan weg, de prijs van de hard- en software! Ik kocht mijn Amiga 500 op de beurs voor 100 gulden minder dan in een winkel, dus alleen daarvoor ging je wel naar de HCC-dagen. Natuurlijk waren de presentaties en de demonstraties ook erg leuk maar de eerlijkheid gebiedt te zeggen dat dat toch niet de hoofdreden was voor de meeste bezoekers. Het was een mooie tijd en wie nu nog naar presentaties of demonstraties wil kan naar een gebruikersgroep of naar bijvoorbeeld de HCC!Kennisdagen. En koopjes vinden we allemaal wel online!

Misschien ben ik niet helemaal volledig geweest maar het was best lastig om info te vinden over de historie. Mocht iemand hier nog een leuk aanvullend verhaal op weten, je bent van harte welkom om daarover een stukje te schrijven voor het volgende Info Bulletin. Ik wens jullie in ieder geval vandaag heel veel plezier op onze eigen clubdag. Dit doe ik vanaf het Graspop Festival, ik zal vandaag aan jullie denken.

Uw voorzitter én redacteur, Ron van Schaik

PS de volgende clubdag is op een afwijkende datum in verband met vakantie van onze locatiebeheerder. Deze zal zijn op zaterdag 30 augustus!

Laatste nieuwtjes

Je kunt een nieuwe video op het YouTube-kanaal van Pixel Playback bekijken. In deze video: De geschiedenis van de Amiga CD32, die CD-ROM technologie, multimediafunctionaliteit en de kracht van de Amiga 1200 computer combineerde in één elegant apparaat: https://youtu.be/ernId0Qw6Lc?si=WG5WNmfUH2OoWIXQ

Protrekkr is een trackerprogramma dat een softwaresynthesizer combineert met een traditionele sampletracker die kan worden gebruikt om elektronische muziek te creëren. De MorphOS-versie is ontwikkeld door Franck Charlet: https://aminet.net/package/mus/edit/Protrekkr_MOS

FAST-40 is een nieuwe cartridge voor de Commodore VIC20 computer, ontwikkeld door Mark Johnson. Deze cartridge modificeert het tekstscherm met 22 kolommen naar 40 kolommen. Het is softwarematig, maar vereist wel een geheugen uitbreiding van 8 kB RAM: https://sleepingelephant.com/ipw-web/bulletin/bb/viewtopic.php?t=11264

The Retro Hour is een professionele Podcast gemaakt door Dan Wood & Ravi Abbott. Het doel van The Retro Hour is de wereld te laten zien wat er gebeurt in de Europese retro spel industrie. In deze aflevering: Een interview met Stuart Maine: https://theretrohour.com/psygnosis-g-police-writing-fps-history-with-stuart-maine-the-retro-hour-ep479/

De Connomore64 is een hardware project met een real-time cyclus-exacte emulatie van de C64 met meerdere parallel geschakelde microcontrollers. De kenmerken zijn: De Commodore IC's (6510, VIA, SID, VIC) zijn vervangen door RP2040/RP2350 microcontrollers, DVI/HDMI video/audio, IEC, gebruikerspoort en joysticks: https://github.com/c1570/Connomore64

Hoe werkt een processor deel 4

Wat moet er allemaal in een processor zitten?

Registers

Een processor heeft als eerste meerdere registers nodig. Een wat modernere processor (6502, 6800, Z80, 8086 en nieuwer) heeft minimaal:
- Accumulator, nodig om berekeningen op uit te voeren. - Program Counter, wijst naar het adres waar (een deel) van de instructie staat. - Stack Pointer, wijst naar een adres in een stuk gereserveerd geheugen waar een stapel (= stack) data staat. - Flag register, bewaart de toestand van belangrijke gebeurtenissen. - Index register, geeft de mogelijkheid om in een tabel data op te zoeken. - Tijdelijk register, nodig voor tussentijdse berekeningen of sprongen. Niet toegankelijk voor de gebruiker.

ALU = Arithmetic Logical Unit

Dit is in feite het rekengedeelte van de processor. Het moet minimaal over de volgende functies beschikken:
- AND
- OR
- EXOR
- NOT
- optellen
- aftrekken
- een mogelijkheid om naar links of rechts te shiften

We hebben een register nodig om de ALU te vertellen wat hij moet gaan doen.
Behalve de uitkomst van een van bovenstaande functies in getalvorm hebben we ook nog minimaal twee resultaten van de ALU nodig:
- De Carry van de bewerking (indien van toepassing). Zeer belangrijk bij het optellen en aftrekken.
- De Zero flag. Deze geeft aan of het resultaat nul is, of niet.

Bij de meeste functies heb je twee invoerwaardes nodig. Er kan er maar één daarvan rechtstreeks uit een register of het geheugen gelezen worden. Dus moet de tweede eerst van te voren in een speciaal register opgeslagen worden. Ook het resultaat moet eerst in een ander register worden opgeslagen want als we het naar een ander register of het geheugen weg willen schrijven moet de data bus worden vrij gemaakt en is de oorspronkelijke ingevoerde waarde weg. Maw., de ALU heeft een paar tijdelijke registers nodig.

De beslissingsnemer

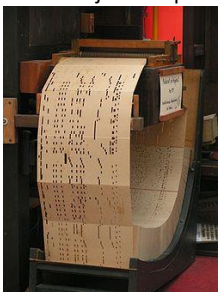
verwachte waarde	gevonden waarde	actie
0	0	1
0	1	0
1	0	0
1	1	1

We kunnen in een computerprogramma beslissingen in bouwen: IF A=10 THEN GOTO 40. Daarvoor is een stukje hardware nodig. Hoe ziet dat er minimaal uit?

Kort gezegd: niets anders dan een EXNOR poort! Hoe kunnen we de "verwachte waarde" en "gevonden waarde' naar elektronica vertalen? De "verwachte waarde" zou een bit van het Flag register kunnen

zijn. De "gevonden waarde" is dan de Zero flag uitgang van de ALU. We kiezen de functie "aftrekken", laden het tweede invoerregister van de ALU met de waarde 10, zetten de inhoud van het geheugen dat de variabele 'A' voorstelt op de eerste ingang en trekken beide waardes van elkaar af. Is het verschil nul, dan wordt de Zero flag uitgang '1'.
Van te voren hadden we het nieuwe adres in het Tijdelijk register gezet en in plaats dat de Program Counter naar de volgende opdracht springt, laadt deze het nieuwe adres.
Klinkt bovenstaande te moeilijk? Kijk dan even terug naar het stukje "Counter".

De eerste processors
Ik mag aannemen dat je wel eens een draaiorgel hebt gezien. Hoe weet een orgel wat voor muziek hij moet spelen? Dat komt door het 'Draaiorgelboek'.



Dit zijn vellen stevig karton met gaatjes er in. Elk gat correspondeert met een bepaald instrument. Komt een pennetje in het orgel een gat in het karton tegen, dat wordt het daar aan verbonden instrument geactiveerd. De lengte van het gat bepaalt de duur.

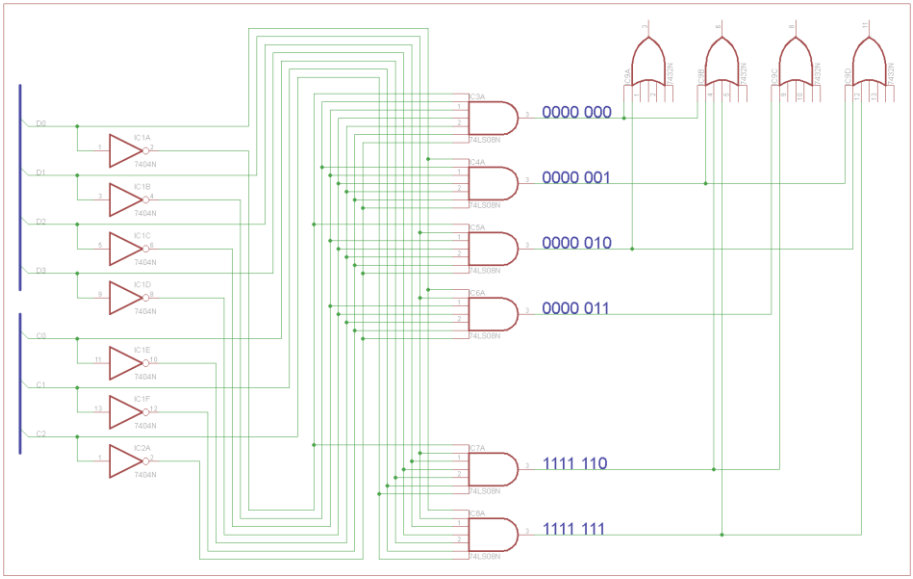
De eerste processors werkten ongeveer ook zo: de processor las eerst een opdracht uit het geheugen, de instructiecode, en sloeg deze op in een tijdelijk register. Nu nemen we voor het gemak aan dat elk van bovengenoemde registers een aansluiting heeft om het te vertellen dat er uit gelezen gaat worden en een aansluiting heeft om te vertellen dat er naar toe geschreven gaat worden. We zouden dus die bitjes van de instructiecode kunnen gebruiken om de genoemde aansluitingen aan te sturen. Maar dan hebben we wel een hele hoop nodig. Gelukkig is er een eenvoudige oplossing om het aantal te reduceren. We kunnen eigenlijk maar uit één register tegelijk lezen en in principe zullen we ook maar naar één register tegelijk schrijven. Met een 4-naar-16 decoder kunnen we zestien registers aansturen. Daar hebben we er zeker twee van nodig anders kunnen we niet uit het ene register lezen en de gevonden waarde naar een ander register schrijven.
. Maar hoeveel bits heb je nu nodig? De Z3 was een 22-bitter maar daar was wel een deel data van. Ik ben zelf ook bezig zo'n 'draaiorgel-processor' te bouwen en die van mij is een 32-bitter: 24 instructiebits en 8 databits.

Het op deze manier gebruiken van de instructiecode had twee nadelen:
- Hoe meer functies, registers of wat dan ook dat je toevoegde, hoe meer bitjes dat je nodig had, hoe breder de databus werd.
- Als je ook maar iets aan de hardware veranderde, ook al veranderde je de breedte van de databus niet, veranderde je ook meteen de instructiecode.
Dat laatste hield in dat, als er tijdens de bouw van een serie er verbeteringen werden aangebracht, software die voor een bepaalde computer was geschreven, niet op die verbeterde versie kon worden gedraaid (en omgekeerd).

Instructie decoder
Het viel een aantal programmeurs op dat bij een heleboel opdrachten elke keer weer steeds dezelfde instructies gebruikten. Dus besloten ze deze groepen instructies hardwarematig te

combineren in mega-instructies en die mega-instructies een eigen zelfverzonnen code te geven. Deze nieuwe codes werden 'opcodes' genoemd.
Aangezien de setjes oude instructiecodes in een vaste volgorde zouden worden uitgevoerd, konden ze ook door een counter worden aangestuurd. Die setjes oude instructiecodes werden in de loop van de tijd micro-code genoemd. Het aansturen van die micro-code gebeurt door de 'Instructie decoder' (ID).

Wat hebben we in eerste instantie nodig op om een Instructie Decoder te maken?



Bovenstaand schema is een deel van de Instructie decoder van een 4-bits processor die binnen elke instructie maximaal acht stappen kon uitvoeren. Zo'n decoder is in drie stukken in te delen:
- Het eerste deel is het linker gedeelte van het schema, links van de blauwe getallen. Dit deel is, in dit geval, niets anders dan een 7-naar-128 decoder. Er zijn maar een paar uitgangen weergegeven en het getal achter de uitgang geeft weer wat er is gedecodeerd.
- Het tweede deel bestaat uit de OR poorten aan de bovenkant. Elke ingang van een register of decoder welke eerst op een bit van de instructiecode moest worden aangesloten, wordt nu op een eigen OR poort aangesloten. In bovenstaand schema staan vier OR poorten, goed dus voor vier van die ingangen.
- Het derde deel is de matrix van verbindingen tussen uitgangen van de AND poorten en de ingangen van de OR poorten. Het is nu een kwestie van uitzoeken welk register wanneer wordt gebruikt binnen een bepaalde opcode en dan de juiste verbinding te leggen.

Opmerking: in bovenstaand schema is niets geoptimaliseerd. In bovenstaand voorbeeld heb ik allemaal OR poorten met zeven ingangen getekend maar dat is eerlijk gezegd puur voor

mijn gemak gedaan. Al die verschillende opcodes duren echt niet allemaal even lang. De laatste actie van elke opcode zou kunnen zijn om die counter te resetten. Met vier bits hebben we 16 opcodes en dan hebben we misschien ook een OR poort met 16 ingangen nodig. Aan de andere kant kan ik mij voorstellen dat een ingang maar een enkele keer wordt aangesproken en is er dus een OR poort met minder ingangen nodig.
Aan de decoder kant kan zeker wat geoptimaliseerd worden. Ik heb net al geschreven dat niet iedere opcode acht stappen nodig heeft. Waarom zou ik dan alle stappen moeten uit decoderen? Minder decodering = minder IC's = goedkoper/kleiner.

Een Instructie decoder is ook ideaal om de beslissingsnemer in te integreren: we hoeven de decoder alleen een extra ingang te geven.

Wat zijn de voordelen van een Instructie decoder?
Het directe gevolg is dat er een smallere databus nodig is en dat heeft weer verdere gevolgen. De allerbelangrijkste daarvan is dat er minder geheugen nodig is, veel minder. De Z3 was een 22-bitter. Stel dat we die met een Instructie decoder terug konden brengen naar een 8-bitter; dat zou voor het geheugen een reductie van meer dan 60 procent betekenen. Maar wacht even, een opcode bestaat uit meerdere regels micro-code. Bij een gemiddelde van vier regels komen we zelfs uit op een besparing van 90 procent!

Het tweede grote voordeel is dat de opcode compleet onafhankelijk is van de hoeveelheid registers/functies binnen in de processor. Tussentijdse verbeteringen en/of toevoegingen hebben nu geen invloed op de opcode meer. Eenmaal geschreven programma's hoeven dus in principe niet meer herschreven te worden. Het is dan ook puur het gebruik van de Instructie decoder die het mogelijk maakt dat programma's geschreven voor de originele IBM PC-XT nog steeds op de nieuwste Pentium 4 kunnen draaien. Mijn eigen P4 start zonder problemen het dertig jaar oude MS-DOS 2.11 op.

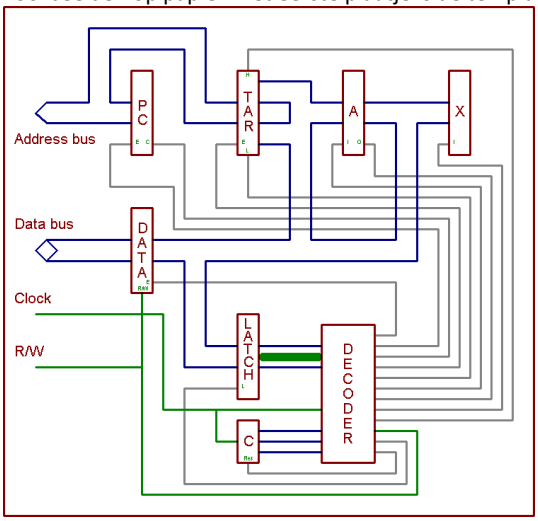
Wat zijn de nadelen van een Instructie decoder?
De Instructie decoder heeft eigenlijk maar één nadeel: hij is groot. Hoewel de theorie al sinds de jaren veertig bekend was, werd hij pas sporadisch toegepast bij computers die mbv. transistors werden gebouwd. Let wel, losse transistors! Pas bij de toepassing van geïntegreerde circuits werd het gebruik van de Instructie decoder gemeengoed.
Een praktijkvoorbeeld van een Instructie decoder?
Er zijn mensen die het spannend vinden om zelf een processor te bouwen, waaronder ondergetekende. Ik heb diverse ontwerpen gezien en steevast gebruikte men EPROMs of FlashRAMs als Instructie decoder. Een prachtig voorbeeld is de Nibbler:

De Nibbler is een 4 bitter en de hele computer is opgebouwd uit 17 IC's. De twee goed zichtbare AT20C16's zijn FlashRAMs en vormen samen de Instructie decoder. Rechts van het LCD-schermpje zit ook een FlashRAM en die dient als ROM en bevat het te draaien programma. Onder het LCD-schermpje zit de ALU, in dit geval een compleet IC, de 74181. Het geheugen is het ICtje rechtsboven van de rechter AT20C16, een 4K 4-bits RAMmetje.

De Nibbler is ook in een ander opzicht een prachtig voorbeeld. Welk deel van de schakeling is processor en wat is de rest van de computer? Ik heb de hele tijd verteld dat de processor over registers beschikt. Die heeft de processor van de Nibbler ook alleen zitten die in het werkgeheugen. De allereerste computers werkten ook zo. Het hebben van aparte registers betekende namelijk extra hardware. En in de tijd van relais en buizen werd elk relais of buis minder in zeer grote dank aanvaard!

Voorbeelden

Ik zal nu proberen de werking van de processor te demonstreren aan de hand van een paar voorbeelden op papier. Het eerste plaatje is de template van alle verdere plaatjes.



De bruine rechthoeken zijn de diverse registers, tellers, etc. De blauwe lijnen geven de adres- en databus aan. Grijze lijnen geven signalen aan die niet actief zijn op dat moment, de groene geven juist de actieve signalen aan. Let wel: het plaatje laat maar een klein, vereenvoudigd (en niet helemaal correct) beeld van de werking van de 6502 of 6800 zien

Gebruikte afkortingen:

PC - Program Counter, een 16 bits pre-loadable up-counter. Wordt gebruikt om de externe adresbus aan te sturen. Mbv. de 'E'-ingang (van Enable) kunnen we uitgangen van de teller (de)activeren. 'C' (voor Clock) dient als ingang om de teller te verhogen.

TAR - Tijdelijk Adres Register, ook een pre-loadable up-counter. Onder andere gebruikt om uit/naar het geheugen of I/O te lezen/schrijven. Van de tellerfunctie geef ik geen voorbeeld dus vind je de bijbehorende 'C' ingang, die hij in echt wel heeft, ook niet terug in de tekening. 'L' is om een byte te laden in het onderste (Low) deel van de teller, 'H' voor het bovenste (H) deel.

DATA - Een bi-directionele buffer om de interne en externe databus te scheiden.

A - Register A, een 8 bits register. In de 6502 is dit het hoofd register. 'I' (input) is om een byte te laden, 'O' (output) om de uitgang te activeren.

X - Register X, ook een 8 bits register.

LATCH - Een 8 bits register om de opcode in op te slaan. 'L' is om het byte te laden.

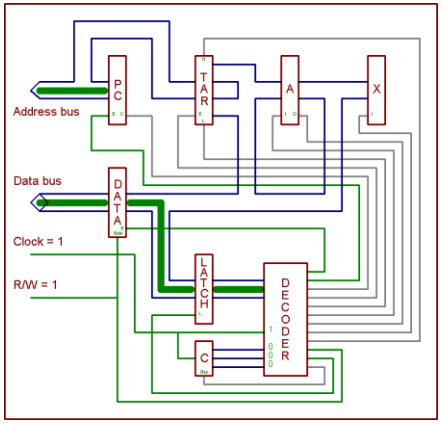
C - De 3-bits teller voor de Instructie decoder counter. Deze wordt aangestuurd door de externe klok. 'RES' is om de teller te resetten.

Clock - Het inkomende klok signaal.

R/W - Read/Write = Lees/Schrijf signaal. Vertelt de buitenwereld of er een lees- of schrijfactie op handen is. 0 = schrijven, 1 is lezen.

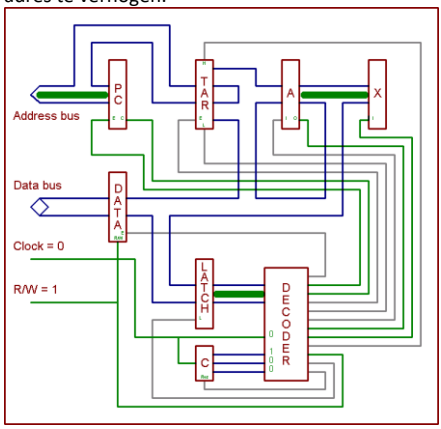
Opcode laden

De eerste stap is voor elke instructie hetzelfde: het vastleggen van de opcode. De PC heeft een adres op de adresbus gezet, R/W = 1 en de DATA buffer is geactiveerd zodat de processor de opcode vanuit de buitenwereld kan lezen. De 'L' ingang van de LATCH is geactiveerd zodat deze de code kan opslaan.

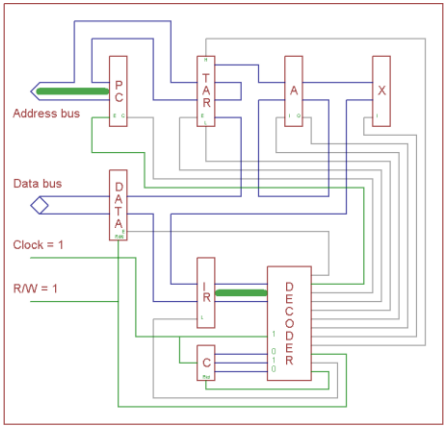


TAX

Als eerste voorbeeld nemen we de instructie TAX. TAX betekent "kopieer (Transfer) de inhoud van register A naar register X". TAX is een 1-byte instructie. In deze fase zijn zowel de uitgang van register A als de ingang van register X geactiveerd. De Program Counter krijgt tevens een puls om het adres te verhogen.

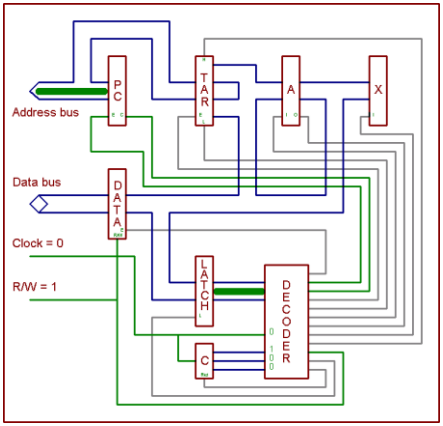


De laatste stap is het resetten van de teller van de ID. Aangezien dit in een oogwenk gebeurt, zijn we meteen terug bij de aller eerste stap, het laden van de opcode.

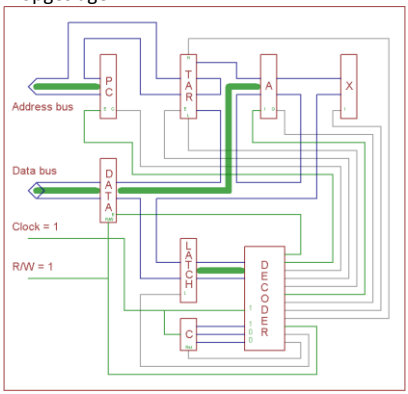


LDA #5

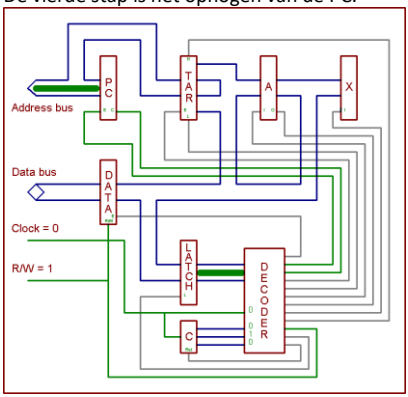
"LDA #5" betekent: "LaaD register A met de waarde 5" en dit is een 2-bytes instructie. Na het laden van de opcode is het enige wat tijdens de tweede stap nog moet gebeuren het ophogen van de PC.



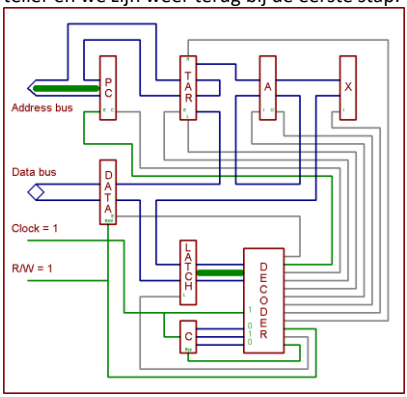
Tijdens de derde stap wordt de data via de DATA buffer naar binnen gelezen en meteen in register A opgeslagen.



De vierde stap is het ophogen van de PC.

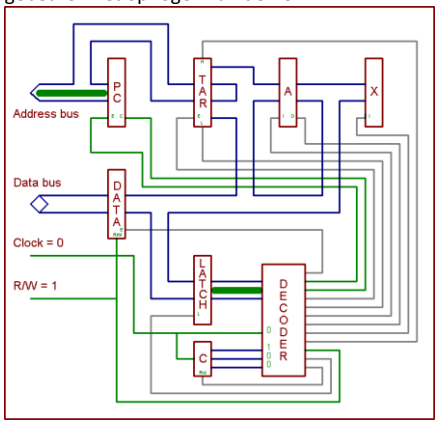


De laatste stap is weer het resetten van de ID teller en we zijn weer terug bij de eerste stap.

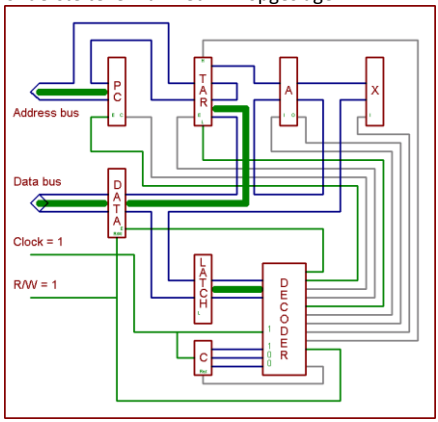


STA $3174

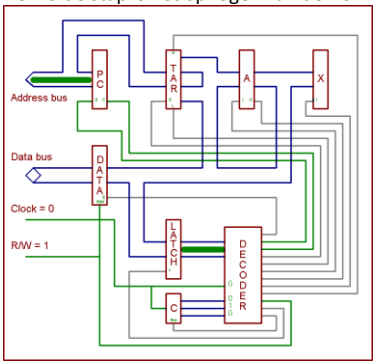
"STA $3174" betekent: "sla (STore) de inhoud van register A op in het geheugen of register op adres $3174". Dit is een 3-bytes instructie. Ook hier is het enige wat tijdens de tweede stap nog moet gebeuren het ophogen van de PC.



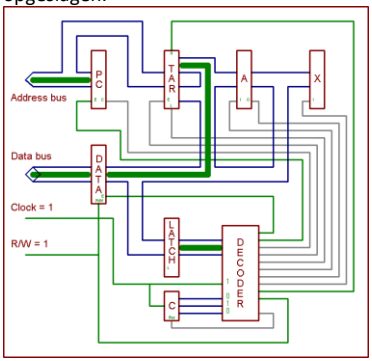
Tijdens de derde stap wordt de data via de DATA buffer naar binnen gelezen en meteen in de onderste teller van het TAR opgeslagen.



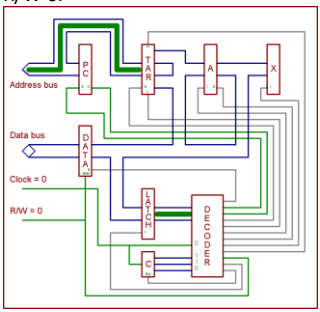
De vierde stap is het ophogen van de PC.



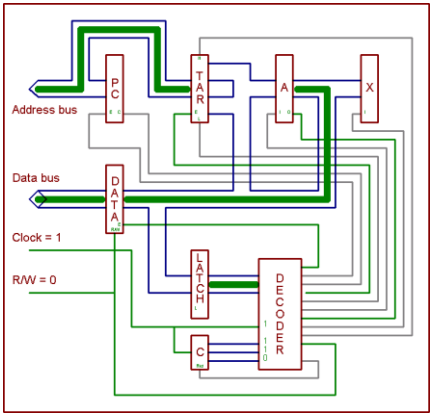
Tijdens de vijfde stap wordt weer de data via de DATA buffer naar binnen gelezen en maar nu meteen in de bovenste teller van de TAR opgeslagen.



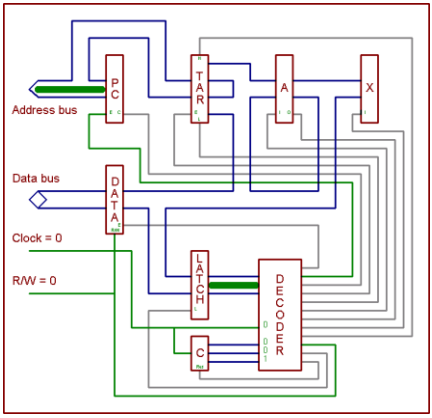
In de zesde stap wordt weer de PC opgehoogd. Maar aangezien we bij de volgende stap de TAR nodig hebben, worden de uitgangen van de PC gedeactiveerd en die van de TC juist geactiveerd. En omdat het om een schrijfactie gaat, wordt R/W 0.



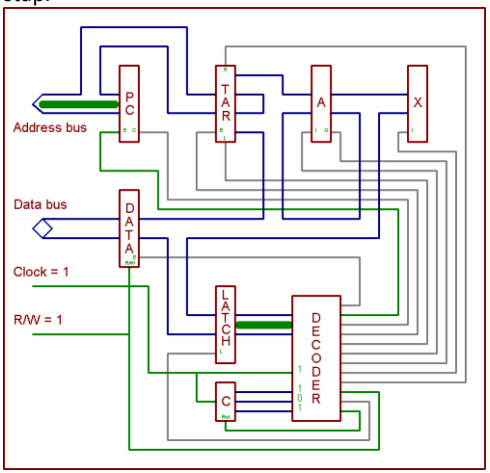
Tijdens de zevende stap wordt de uitgang van register A geactiveerd en de data gaat via de DATA buffer naar de buitenwereld.



Tijdens de achtste stap krijgt de PC weer de controle over de adresbus. Merk op dat de teller niet wordt verhoogd omdat dat al eerder is gebeurd. Die ophoging had ook nu kunnen gebeuren maar dit is nu net een van de voorbeelden waar we niet weten hoe de 6502 precies werkt. Sterker, de oorspronkelijke ontwerper, Bill Mensch, weet het ook niet meer :) Maar het is ook meteen een voorbeeld van de flexibiliteit van de Instructie decoder: voor de werking van de processor maakt het namelijk niets uit.



De laatste stap is wederom het resetten van de ID teller en we zijn weer terug bij de eerste stap.



Speciale ingangen
De meeste processoren hebben een of meer speciale ingangen. Een reset ingang hebben ze allemaal en de meeste processoren hebben vaak een of meerdere zogenaamde interrupt ingangen. De aanwezigheid van een signaal op zo'n ingang moet ook aan de Instructie Decoder verteld worden en deze heeft dan ook minimaal een extra ingang om kenbaar te maken dat een of meerdere van die speciale ingangen zijn geactiveerd.

Reset
De meest bekende ingang is de Reset. Elke bekende processor heeft een start signaal nodig om vanaf een nulpunt te kunnen beginnen. Hoe de processor reageert op een reset is verschillend per type. De computers die ik ken die een "draaiorgel" processor hebben starten allemaal van adres nul. Logisch, want dat was gewoon een kwestie van de Program Counter resetten. De Z80, bekend van CP/M en MSX computers start ook op vanaf $0000.
De Intel processoren die in PC's gebruikt worden zoeken naar een opcode op adres $FFFF0. Op deze plek staat in de regel een lange sprong naar een ander deel van de BIOS (zie programma voorbeeld helemaal boven).
De 6502 zoekt naar het eigenlijke startadres op de plaatsen $FFFC en $FFFD.

Interrupt / Non Maskable Interrupt
Stel je voor je bent een boek aan het lezen en de telefoon gaat. Jij bent in principe vrij om die te negeren maar in de regel neem je de telefoon op. Maar voordat je dat doet, zorg je er eerst voor dat je weet waar je in je boek was gebleven door bijvoorbeeld een blaadje tussen de bladzijden te stoppen. Na het telefoontje kun je weer verder gaan met lezen vanaf waar je gebleven was dankzij dit blaadje.

Maar als er een brandalarm afgaat, moet je reageren. Is de brand bedwongen en je boek heeft het overleefd, kun je, ook weer dankzij het blaadje, weer vanaf de oorspronkelijke plek verder lezen.

Veel processoren beschikken over een 'Interrupt ReQuest' (IRQ) en een 'Non Maskable Interrupt request' (NMI) ingang. De eerste kan genegeerd worden (Engels: maskable), op de tweede moet de processor dus reageren. Hoe gaat de een processor met een Interrupt om? Dit verschilt per processor en in dit geval spits ik de uitleg een beetje toe op de meest bekende processoren, die van Intel en AMD. Een interrupt wordt pas in behandeling genomen als een lopende instructie volledig is afgewerkt. Het eerste wat de Interrupt dan doet is het momentele adres van de Program Counter op de Stack opslaan. De Stack is een

van te voren door het programma gereserveerd gedeelte van het geheugen. Als tweede wordt het Flag Register opgeslagen. Vervolgens zoekt de processor in een tabel naar vier bytes die samen het adres vormen waar de eigenlijke Interrupt routine staat en voert die routine dan uit.

Dit uitvoeren van die Interrupt routine kan niet zo maar klakkeloos gebeuren. Deze routine wil natuurlijk van een aantal registers gebruik maken maar er is een goede kans dat die al in gebruik zijn door het programma wat eerst liep. Deze registers moeten dus eerst ergens opgeslagen worden en dat gebeurt in de regel ook weer op de Stack. Na het beëindigen van de routine worden de opgeslagen waardes weer van de Stack gehaald en weer in de oorspronkelijk registers teruggeschreven. Als laatste stap wordt ook het Flag register en Program Counter hersteld met de waardes op de Stack. Dit laatste houdt meteen in dat de processor automatisch verder gaat met het eerder onderbroken programma.

Moderne processoren
Wat zijn nu de belangrijkste verschillen tussen de 8088 en een Pentium 4 waardoor de laatste stukken sneller is dan de eerste:
- Nieuwe productie technieken maken het mogelijk op een hogere clock te draaien.
- De P4 heeft een bredere databus; 64 bits tegen 8 bits
- De opcodes van de P4 zijn sneller gemaakt. Denk aan het voorbeeld van het vermenigvuldigen.
- De P4 kent 'pipelining'. Een opcode kent meerdere stappen. Tijdens de stappen dat er niet van de databus gebruik hoeft te worden gemaakt begint de P4 de volgende instructie al te lezen. Maar dit werkt niet altijd. Stel dat de eerste opdracht een voorwaardelijke sprong is en er moet inderdaad gesprongen worden, dan is de leesactie van de opdracht na de sprong voor niets geweest. Maar al met al is het niet langzamer dan als er geen pipelining was geweest. - De P4 heeft meerdere cores. Dit is hetzelfde alsof er meerdere processoren in de computer zitten met elk een eigen stuk geheugen. Bij een P4 kan één core zich bezig houden met het geluid en een tweede core met de video. Let wel, het betreffende spel moet wel speciaal geschreven zijn om met meerdere cores te kunnen werken. Zou je een oud spel van voor het P4 tijdperk spelen, dan zal zich ook maar één core zich met het spel bezig houden en de anderen in principe werkloos toekijken.

Zelf een processor bouwen
Ik heb al gezegd dat er mensen zijn die zelf hun eigen processor hebben gebouwd. Bij deze diverse interessante sites:
- De allereerste bekende computers werkten met relais. Hier een voorbeeld van een relais computer: https://web.cecs.pdx.edu/~harry/Relay/index.html
- Een werkende Nibbler: https://www.bigmessowires.com/nibbler/
- Een werkende 8-bits processor: https://www.bigmessowires.com/bmow1/
- Een werkende 16-bits processor: https://www.timefracture.org/D16.html

Artikel geschreven door Ruud Baltissen

Nieuwe software

Mugwump Plus is een nieuw spel voor de Commodore 64 computer, ontwikkeld door Logiker (code & pixels), Pararaum (code) and Will (muziek & test). In het spel moet je de Mugwumps vinden in een raster van 10 bij 10: https://csdb.dk/release/?id=252279

Novaload transfer is een programma voor de Commodore 64 computer, ontwikkeld door Fungus. Het programma maakt het mogelijk om een hele cassette of meerdere cassettes in één keer over te zetten met behulp van een REU. De broncode en een korte handleiding zijn inbegrepen: https://csdb.dk/release/?id=252190

Matthias Kramm heeft een demo gemaakt voor de Commodore 64, met video en audio. Dat lijkt niet zo bijzonder, maar hij gebruikt de geluidschip voor de video en de videochip voor het geluid. Wissel de video- en audiosignalen op je monitor om en start de demo: https://www.quiss.org/signal_carnival/

Kye is een nieuw spel voor de Plus/4, ontwikkeld door Patai Gergely. In het spel moet je de groene cirkel verplaatsen en alle diamanten verzamelen. Maar zorg dat je niet vast komt te zitten of opgegeten wordt door de monsters: https://plus4world.powweb.com/software/Kye

Donutsman is een nieuw spel voor de Commodore 64, ontwikkeld door Youki Games. Je speelt een beroemde videospelheld, maar zijn grootste passie zijn donuts. 's Nachts ontsnapt hij uit zijn spel om de spookachtige donutmagazijnen te verkennen: https://youki06.itch.io/donutsman-c64

MSXdev24 - everybody's Kung Fu Fighting!

De 20e editie van deze bijna-jaarlijkse competitie heeft weer mooie spellen opgeleverd! Op https://www.file-hunter.com/MSXdev/ zijn inmiddels meer dan 300 inzendingen vanaf 2003 online speelbaar.

De compo was, net als vorig jaar, 'freestyle': alle MSX platformen en uitbreidingen zijn toegestaan. Ook dit jaar weer zijn veel inzendingen geschikt voor MSX1, de meeste in ROM formaat. Echter wel een daling – vorig jaar bijna 80% MSX1, nu is de verhouding 18 MSX1 : 12 MSX2 : 1 Turbo-R. Meer spellen gebruiken MSX-MUSIC of SCC (6 in totaal; was 2). Eén kan zelfs de voortgang opslaan op de Pana Amusement Cartridge.

Op 31 januari (pas) sloot de negen maanden durende inzendtermijn. De vier juryleden publiceerden op 28 maart 2025 hun oordeel over de maar liefst 31 inzendingen voor MSXdev24 – vier maanden later ten opzichte van de uitslag van MSXdev23 (waarvan de resultaten al op 29 november 2023 bekend werden). MSXdev25 is nog niet gestart – als ze nog verder het volgende jaar in blijven schuiven, gaan we ooit ergens een jaar moeten overslaan ;)

De verzamelde prijzenpot bedroeg bijna €900,- plus 11 gesponsorde items, beschikbaar gesteld door onder andere Supersoniqs, BrewOtaku en de bij ons welbekende Jeroen Taverne! De prijzenpot wordt verdeeld onder de top-4 scoorders; de items onder de top-11.

Dit viel me op:

- Ook dit jaar weer een tekstgebaseerd spel, dat *ook dit jaar weer* een zeer lage score krijgt – de laagste zelfs. Wizsoul is een wat warrig text adventure in Engels en Spaans. Het lijkt geen aanrader om tekstgebaseerde spellen in te sturen naar MSXdev…

- Gamecast en joesg stuurden meerdere inzendingen in, respectievelijk vier en twee games. Met een hoogst behaalde 12e plaats grijpen ze daarmee *net* naast de pot met sponsorprijzen.

- The Curse Of Lies draait op MSX2 met MSXDOS2 maar gebruikt vreemd genoeg geen capaciteiten boven die van MSX1.

Wat de genres betreft is er voor elk wat wils: roguelike, text adventure, action adventure, puzzle, arcade/action, platformer, shooter, maze, ...

Er zijn enkele classic remakes van Minas, Sokoban, Q*bert, Abu Simbel en Pyjamarama.

De uitslag
Alle games zijn gescoord op categorieën Algeheel (3x100 punten), Gameplay (100), Graphics (100) en Sound (100). De einduitslag is de optelling daarvan.

De top vijf:

1. **Maiden's Extreme Fist - BUN-GA-RUA**, Retromixis (513/600, 85%) - tevens beste muziek (SCC!)
2. Room 5, MO5.com (484/600, 80%)
3. Doomlings, Totta (480/600, 80%) - tevens beste gameplay
4. Puzzle Pals, Furcifer Studios (474/600, 79%) - tevens beste graphics
5. Princess Paloma's Rescue, InfiniteMSX (464/600, 77%)

Een eervolle vermelding gaat wat mij betreft naar deze Nederlandse inzending:

Super Sokoban, 2NICE (458/600, 76%)

Super Sokoban - MSXdev / 2NICE

Naast een opmerking over onhandige controls, is er verder veel lovend commentaar van de jury over de winnaar *"Maiden's Extreme Fist – BUN-GA-RUA"*:

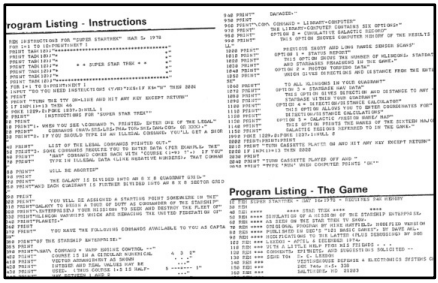
- *"basic aesthetics of Yie-Ar-Kung-Fu 2 brought to MSX2 with great mastery"*
- *"Yie-Ar-Kung-Fu on MSX2 like Konami could have released"*
- *"supercharged Yie-Ar-Kung-Fu 3 with much more exploration"*

Artikel geschreven door Hedzer Westra

Atari nieuws (door Dennis Roos)

Uit de regio

Vandaag is alweer de 9e editie van de jaarlijkse ‘Atari Invasion’. Dit jaar is het thema “Atari in de BeNeLux”. We hebben dan ook voornamelijk spullen mee uit deze landen en in deze landen ontwikkelde software. Binnen onze eigen gelederen zijn er diverse demo’s en spellen gemaakt in de loop der jaren die zeker de revue zullen passeren. Achterin het schuurtje kun je interactief veel dingen uitproberen en bekijken. Op de info-beamer schermen komen onder meer een aantal Atari BeNeLux reclames voorbij en zie je een groot deel van de commercieel uitgebrachte hard- en software langs komen.

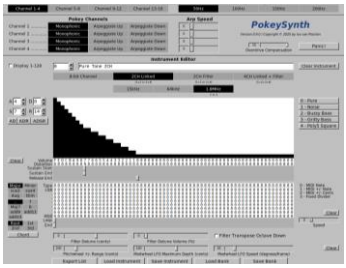


Modern klassiek

De BASIC listing uit 1978(!) van ‘Super Star Trek’ is door ‘Electric Dreams Games’ aangepast voor GFA Basic (Atari ST(e)). STDoom is eigenlijk alleen speelbaar op een ST met een 68020 processor, Doom8088 is wel speelbaar op een gewone ST, maar is helaas monochroom. Atari heeft ook weer een aantal spelcartridges klaar staan: Voor de 2600+ is dat Avalanche en voor de 7800+ zijn dat Super Circus Atari, Counter Measure 2, Dark Chambers en Tiger Heli. Interessante nieuwe homebrews voor de 7800 zijn ‘Quest for la Hermanita’ en Downland. Interessante homebrews voor de 2600 zijn: Galactic Intrusion en Seaquential. Op 8-bit vlak zijn er diverse BASIC 10liner spelen verschenen, waarvan de multiplayer ‘FujiNet Battleship’ er uit springt. Meer informatie en links naar nieuwe titels vind je in de lijsten van Zero Page Homebrew op het Atari Age forum: https://tinyurl.com/2025ZPH2600.

Niet altijd fysiek

Atari Age gebruiker Woj heeft de Atari 8-bit MiSTer core onder handen genomen en flink wat bugs opgelost. Ivo_P heeft een (paar) alternatieven bedacht voor de steeds moeilijker te verkrijgen kristallen https://tinyurl.com/PAL35468. POKEY muziek componeren zonder een Atari, ook dat kan tegenwoordig met Ivo_P’s LV2 Pokey virtual instrument https://tinyurl.com/PKYSYNTH. Reifsnyderb ontwikkelt nog altijd verder aan diverse Atari 1090 compatibele uitbreidingskaarten. De laatste variant is een seriele en parallele kaart, binnenkort verkrijgbaar via Tindie https://tinyurl.com/reifsneiderb



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

27 t/m 29 juni 2025 Posadas Party
2025 Posadas, Spanje
https://posadasparty.com/

18 t/m 20 juli 2025 Arok Party
Ajka, Hongarije
https://arok.intro.hu/2025/

24 t/m 27 juli Boom Part 2025
Tuchola, Polen
https://www.boom-party.c64.fun/

1 t/m 3 augustus 2025
Pågadata 2025 Teckomatorp, Zweden
https://xn--pgadata-exa.se/2025/

30(!) augustus 2025 10:00 – 16:00 uur
Clubdag in 't Schuurtje, Maarssen

18 oktober 2025 10:00 – 16:00 uur
Clubdag in 't Schuurtje, Maarssen