

## 2de Excerpt reactie van fabrikanten 23 januari 1963

Telefunken: “.... we believe that our TR 4 computer will meet your specificatione in an outstanding way.” Nederlandse vertegenwoordiger heeft een persoonlijk onderhoud aangevraagd; ik heb gesuggereerd. dit persoonlijk onderhoud uit te stellen, totdat wij over wat meer documentatie over de TR 4 beschikken (die ik alleen maar uit roddelverhalen ken). Hij zou gegevens sturen, maar excuseerde zich bij voorbaat “omdat hij niet zo veel had”.

Siemens: Verstandige brief van de Nederlandse vertegenwoordiger, die beseft, dat we niet in een Siemens 2002 geïnteresseerd zijn. “Nieuwe ontwikkelingen, waarvan de aankondiging binnen afzienbare tijd verwacht mag worden, beantwoorden stellig aan het beeld, dat U zich van de computer van 1965 gevormd hebt. Wij zouden het daarom zeer op prijs stellen, U een dezer dagen persoonlijk op de hoogte te mogen stellen van de plannen van Siemens voor de naaste toekomst. In afwachting van Uw gewaardeerde berichten.....”. Ik zal proberen een afspraak met deze man te maken.

NV Electrologica: Explicatie van het feit, dat nog geen gedetailleerde offerte gedaan is “Naar U bekend zal zijn zal onze aanbieding zich richten op apparatuur, welke door onze tot dusverre nog niet word afgeleverd, zodat een voorafgaand intern beraad noodzakelijk is.”

Honeywell: De brief, gericht naar het Amerikaanse adres is op 19 december vanuit London beantwoord “This is to advise you that we will forward our summary answers prior to 1st Februari, 1963, in accordance with your instructions.”

Een maand later komt een brief met een excerpt van de eigenschappen van hun bestaande producten, geen gedrukte documentatie over deze machines, en een uitnodiging om toch vooral naar London te komen. Hij biedt aan: “We are now able to supply three computers both in Great Britain and on the Continent and these are namely the Honeywell 400, the Honeywell 800 and Honeywell 1800. The last named machine, the Honeywell 1800, is available in two forms. The first form is a computer which has the same order code as the Honeywell 800 and can consequently use all its software without any changes whatsoever. The second version of the Honeywell-1800 is a wholly scientific machine which is still not out of the design stage. However, by mid-1963 the full specifications of this machine will be available.” en onder het hoofdje Price: “We note that \$750000 is the upper limit. This rather restricts the choice as far as Honeywell is concerned and it would seem the only computers available from our range are the Honeywell 400 and the Honeywell 800. The price of the Honeywell 400 starts at approximately \$240000 and the price of the Honeywell 800 commences at approximately \$1,000,000. However, whether it would be possible to come to an arrangement about this with respect to either rental or educational discount, is a matter for discussion should you be interested in any one of our products.” Aangezien verder over de prijs van de 1800 niets gezegd wordt, in geen van beide versies, leid ik af, dat de prijs daarvan helemaal prohibitief is. De Honeywell 400 is geen reële propositie: maximum omvang van het kerngeheugen 4096 woorden. “On the Honeywell 400 it is possible to print in parallel with computing and also possible to overlap search time of random access discs with computing. Furthermore, overlapping of other operations on peripheral devices is possible but the Honeywell 400 is strictly speaking a non-parallel machine.” De machine heeft —als alle Honeywell machines— een drieadrescode, voor de 400 wordt opgegeven “add A to B and place result in C takes 120 microseconds. There is no float point hardware on this computer.” Fatsoenshalve zal ik vragen om een Reference manual van de Honeywell 800. Mijn indruk van hun apparatuur is robuust, betrouwbaar en lomp. Hun magnetisch-band-eenheden zijn aanleiding voor oprechte trots: ik heb deze dingen feilloos zien werken in een ruimte, waar nog zo veel aan geknutseld werd, dat de vensterbanken met een laagje rood stof bedekt waren. Wat ik van de parallel processing capabilities van de Honeywell 800 begrepen heb, is iets heel loms, wat dan wel een explicatie voor de prijs is. De volgende opmerking is wel wat naïef “Our computers have both a binary and a B.C.D. (dwz: binary coded decimal) system, and orders are available for treating any number either as a purely binary number or a configuration of bits representing a decimal number.....All our computers with the exception of the second version of the 1800 have been designed for both commercial and scientific usage and we have found this duplication of instructions to be extremely valuable to

all users. It should be made clear that the addressing system is in binary. However, we have such a variety of compilers and other languages that we think this point is comparatively of minor importance.” Behalve dan, dat de klant de B.C.D.-faciliteiten wel moet betalen, ook al ziet hij helemaal geen kans om ze te gebruiken. En wat ze dan aanbieden aan software is “our AUTOMATH compiler which is compatible with Fortran 2”. Ze moeten maar eens met gegevens komen, vind ik.

Remington Rand. Na enig misverstand over de rol, die collega van der Sluis in het contact met Remington Rand zou moeten spelen, is de communicatie nu goed op gang geraakt en hebben we vrij veel gegevens over Univac 1107. Op het eerste gezicht krijg je de indruk, dat ze niet precies wisten, wat ze Moesten bouwen en dat ze daarom maar “alles” gebouwd hebben. De prijs is er dan ook naar. (Er wordt ons 50 procent korting aangeboden, de nu volgende bedragen moeten dus gehalveerd worden.) Een kast met ponsbandlezer (400 characters per seconde) en een ponsler (110 characters per seconde) kost f140,000. (Hogere snelheden —lezen 1000 en ponsen 300— zijn "optional"). Dat er empooi zou zijn voor een installatie met niet evenveel lezers als ponsers komt niet tot uiting. Om deze ponsbandapparatuur te kunnen gebruiken, moet je echter nog een “Paper Tape Control and Synchronizer” hebben, die f.244.000 kost. Dit is dan voldoende voor drie van zulke kastjes, die dan allemaal simultaan kunnen werken, waarbij “werken van een kastje” betekent “of ponsen of lezen” maar niet allebei tegelijkertijd. Dit voor f221,000 per kastje.

Het kerngeheugen is niet excessief duur: f50 voor een woord van 36 bits; dit is een prijs van augustus 1962. De “Univac Central Computer 1107” kost daarentegen (nog zonder kerngeheugen) f3.550.000. Het is de machine met het uitgebreidst instructierepertoire, dat ik ooit gezien heb.

Het bevat (gelukkig) geen opdrachten voor decimaal rekenen. Wel zitten er opdrachten voor floating point in, edoch slechts 8 bits voor exponent (met teken) en 28 bits voor mantisse (met teken); precisie van de mantisse met iets meer dan negen decimalen is niet overdreven groot, maar redelijk; maximum waarde +127 voor de binaire exponent is beslist heel erg aan de krappe kant.

De geheugenadressering is indrukwekkend wat betreft de mogelijkheden, om gedeelten van woorden te adresseren: het geheugenwoord kan je verdelen in tweeën, in drieën of in zessen. Elk aangewezen part (dwz. het hele woord, of een van de  $2 + 3 + 6 = 11$  gedeelten) kunnen tegen de minstsignificante kant van een “register” aangeschoven getransporteerd worden. De bits, die een part aanwijzen, zijn helaas niet betrokken in de B-correctie. (Dat heeft de ATLAS verstandiger, waar de woordlengte 48 bits is, en rechts van de adresbits in de opdracht nog drie extra bits zitten om een syllabe aan te wijzen.)

Het kerngeheugen van deze machine is een naar rommeltje: je kunt maximaal 65536 woorden kerngeheugen krijgen. Als je minder hebt, zeg 32768, dan krijgje 0–16383 en 32768–49151. De helft begint te tellen bij 0, de andere helft begint bij 32768. Dit heeft een technische achtergrond: de twee helften van het gehele geheugen zitten per definitie in twee verschillende “banks”: als opdracht en operand uit de verschillende banks komen, dan duurt het merendeel der opdrachten de helft van de tijd, die nodig zou zijn, als ze beide uit dezelfde bank moeten komen. Dit verschil van een factor twee is zo groot, dat je het niet kunt laten lopen; de hele programmaorganisatie vertoont dan ook de sporen daarvan.

De programmeur wordt, zelfs in de meer geavanceerde systemen “under executive control” beleefd verzocht zijn geheugenbehoefte “banksgewijs” te specificeren. De motivering, om bij niet volledig geheugen toch de tweede helft te gaan tellen bij 32768 is zwak, voor zover ik zie “In this manner, the system lends itself tot future expansion.” Op andere manieren ook wel; het offer, dat je je adressen niet meer consecutief genummerd hebt voor het geval, dat je dat zou willen, is dunkt mij wat lichtvaardig gebracht.

De eerste 128 plaatsen van de eerste bank bestaan maar half. Je kunt ze alleen via hocus pocus van de console vullen met een of ander geschikt “pril begin”, ze zijn alleen via de opdrachtselectie bereikbaar. Via de getalselectie refereer je, als het adres  $< 128$  is, niet naar de kernwoorden, maar naar een van de 128 woorden thin film magnetic memory. Uit film geheugen kan je dus per definitie geen opdrachten gehoorzamen. Lees en schrijfcyclus van film geheugen zijn elk 300 nanosec (tegen 2 microsec. voor de kernen, het is dus zowat 7 maal zo snel).

Het is niet zo eenvoudig, om die 128 plaatsen verstandig te gebruiken. De ontwerpers hebben ons een handje geholpen, door 32 van deze woorden een speciale rol te geven, nl 16 “input Access Control Words” en 16 “output A.C.W.’s voor de 16 inputen de 16 outputkanalen. (Dat is het maximum dat er aan kan; die kunnen allemaal tegelijkertijd lopen, de koppeling tussen kanaal en periphere apparatuur is voor elke installatie een constante: als je er iets aan wilt hebben, omdat je wilt kunnen, als dat nodig is, dan blokkeert die aanwezigheid de bijbehorende —dwz: het er aan toegekende kanalen permanent.Hoi.)

Dan is er verder een aantal registers op het filmgeheugen (om precies te zijn 47) die een speciale functie hebben; sommige zijn heel specifiek zelfs, zoals de “real time clock”), eentje fungeert als “locale variabele van de herhaalopdrachten”, eentje fungeert als parameter voor de herhaalopdrachten en eentje fungeert als parameter van de “maak-instructions”. Dan hebben we voorts 16 accumulatoren en 16 indexregisters, wat totaal maar 28 adressen is, want vier hiervan zijn zowel als A-als als 8-register benoembaar. ( Dat viertal is dus op drie verschillende manieren benoembaar, want het zijn ook nog —via de getalselectie bereikbare— normale adressen.)

Het lijkt mij moeilijk, om al deze plaatsen te gebruiken. Voor de constructie van een of ander individueel programma, waarbij je je van niemand of niets iets hoeft aan te trekken, zijn er geen principiële moeilijkheden. (Ik voorzie ernstige praktische moeilijkheden, als de programmeur ten offer valt aan een ongebreidelde neiging tot optimalisatie: met 12 accumulatoren, 12 indexregisters en noq vier, die dubbel bruikbaar zijn, heb je nog eventjes. Ter verhoging van de feestvreugde zijn deze 12-tallen essentieel lineair gaordend: je hebt operaties, die werken op twee successieve accumulatoren.) Wat mij veel meer interesseert is, hoe je op deze accumulatoren en indexregisters een organisatie denkt op te bouwen. Als je “op je eigen houtje” programmeert, dan gebruik je deze registers als “locale variabelen” van je proces; wil ik nu een subroutine maken, die als waarachtig nieuw “primitivum” fungeert, dan eis je dus, dat deze subroutine, die om te kunnen werken, deze speciale registers wel gebruiken moet, dat hij de spaciale registers, die hij gebruikt voor terugkomst in het aanroepende programma “in oude staat hersteld” presenteert. Dit impliceert een redden en herstellen, dat, als het op gezag van de subroutine gebeurt, allerlei overbodig werk met zich mee brengt, als de subroutine aangeroepen wordt in een context, waar deze speciale registers toevallig geen zinvolle inhoud hebben. De meest voor de handliggende oplossing is, om maar een klein gedeelte van deze 28 registers te gebruiken en die altijd te redden en te herstellen. Bij overschakeling van het ene programma op het andere ten gevolge van interruptie spreekt dit allemaal nog eens zo sterk. Ik zou wel eens willen weten, wat de firma Remington Rand voor ideeën heeft aangaande een politiek, om deze registers te gebruiken. Ik ben bang van geen.

Een en ander geldt nog sterker voor 12 zg.R-registers en de 53 woorden “unassigned”. De 12 R-registers zijn woorden van het filmgeheugen; er zijn echter twee opdrachten (load en store), waarin deze plaatsen als register in het functiegedeelte aangegeven worden, wat je de mogelijkheid geeft, om naar deze 12 plaatsen willekeurige transporten uit of naar geheugen te spelen. Je kunt daar wel grappige dingen mee doen (het ene R-register in het andere copieren gaat in 1 opdracht en je mag kiezen of je dat met een load of een store wilt doen.) Maar zijn deze faciliteiten voldoende belangrijk, om de rompslomp van conventies, die hun gebruik met zich meebrengt? De 53 woorden “unassigned” zijn nog dubieuzer. Het zijn maar halfslachtige geheugenplaatsen, omdat er geen opdrachten uit geselecteerd kunnen worden. Er staan dus getallen en bij gebruik staan de opdrachten dus op kerngeheugen. Opdrachten duren dus minstens 4 microseconde, dwz. precies even lang als wanneer de getallen in de andere “bank” hadden gestaan. Het enige nut, dat ik van deze plaatsen kan bedenken is, als twee programma’s in verschillende “banks” intensief op dezelfde getallen zouden moeten opereren. Dan loopt een van de beide programma's twee keer zo snel als wanneer dit gemeenschappelijk getallenmateriaal in een van de Banks was ondergebracht. Deze hypothetische configuratie lijkt mij nauwelijks een rechtvaardiging voor hun aanwezigheid. Je kan dus wel eens een beetje teveel moeten betalen. (Je kunt het verdedigen met de opmerking, dat er nu tenminste voldoende filmgeheugen is om het aantal kanalen uit te breiden, maar dat is, gezien de kosten van een kanaal, ook nauwelijks aantrekkelijk.) Ik zou van R.R. wel eens willen horen, hoe ze deze 53 extra woorden filmgeheugen denken te gebruiken.

De synchronisatiefaciliteiten maken een redelijke indruk. Ik begrijp het niet allemaal, maar het lijkt zo op het eerste (en zelfs op het tweede) gezicht een minder chaotische indruk dan bv. Bendix of IBM 7040. Ik ben wel

bang, dat de beperking van 16 input en 16 output kanalen essentieel is, wat alleen maar een gerechtvaardigde conclusie is, als de “rate of information flow” over een aantal van deze kanalen hoog is.

De trommels worden ook als communicatieapparaat gespeeld; ik heb er nu verder geen studie van gemaakt, Remington Rand is vanouds beroemd om de kwaliteit en de capaciteit van zijn trommels. Totzover Univac 1107.

Telefunken: De Nederlandse vertegenwoordiger heeft de documentatie ter beschikking gesteld, het is zojuist per post gearriveerd. Hij heeft er goed aangedaan, zich bij voorbaat te excuseren. Het is niet meer dan twee brellende vlugschriften van 16 kantjes elk (inclusief kaft), met kleuren maar zonder informatie. “Die TR4-Datenautomation bringt Ihrem Unternehmen neue Moglichkeiten fur optimale Losungen schwieriger Aufgabenstellungen.” etc.etc. Schandal ig!

PHILCO: Over de reactie’s van Philco kan ik kort zijn. Ze bieden aan de Philco 4000, wat een versnelde versie van de Philco 1000 is. In menig opzicht kan je deze machine beschouwen als een verbeterde 1620. Hij is uitgerust met een geheugen opgebouwd uit onafhankelijk adresseerbare syllabes van zes bits, die in serie afgewerkt worden, zodat het geheel nogal langzaam is. Hij is beter dan de 1620 in zoverre hij binair werkt en niet de lengte van operanden met behulp van flagbits aangeeft. Verder is het ontwerp wat in- en output betreft waanzinnig. Het desideratum, dat de machine tijdens inen output niet tot ledigheid gedoemd is, hebben ze vertaald in “rekenen en communiceren met simultaan plaats kunnen vinden.” Dit laatste is gerealiseerd, door twee central processors te koppelen en samen in de zelfde kamer te zetten! Als de een rekt, kan de andere communiceren. Heel eenvoudig. Verder is bijkans elke poging om de machine dynamisch tot betere performance te laten komen, tot mislukking gedoemd. Ik citeer+“All input-output transmissions cause the Central Processor to wait at the input-output instruction until the transmission portion of the instruction is completed. Thus, input-output operations are a function of the speed of the device being used. The computer will go on to to the next instruction as soon as the transmission is completed. (De machine rekt dan verder tijdens de “non-transmission portion” van de communicatie-opdracht, het afvoeren ven de ponskaart of zo.) Note that any attempt to check the status of the device being used will cause the computer to wait .... ..until all functions designated by a prior input-output instruction are completed.” Dit citaat is uit een manual, gedateerd oktober 1962.

De oogst is dus tot nog toe niet zo bemoedigend.

E.W.Dijkstra

---

transcribed by Bart Vreugdenhil  
revised 27-Dec-2011