

Bouw en gebruik van rekenapparaten bij de Mathematische Afdeling van het Centrale laboratorium der PTT

HT de Beer
H.T.de.Beer@gmail.com
<http://heerdebeer.org>

Amsterdam, 26 februari 2008

Inhoudsopgave

1	De Mathematische Afdeling en het onderzoek naar elektronische rekenapparaten	1
1.1	Rekenwerk aan de Mathematische Afdeling	1
1.2	De behoefte aan een rekenmachine	3
1.3	Van ARCO tot TESTUDO	4
2	PTERA	6
2.1	Ontwikkeling en bouw van de PTERA	6
2.2	De machinecode van de PTERA	7
2.3	De programmatuur van de PTERA	8
2.4	Het gebruik van de PTERA	10
3	ZERO	12
4	ZEBRA	15
4.1	Het ontstaan van de Stantec ZEBRA	15
4.2	Het logische ontwerp van de ZEBRA	18
4.3	Programmatuur voor de ZEBRA: de Simple Code	20
4.4	Het gebruik van de ZEBRA: de ZEBRA-club, onderwaterprogrammeren en trucologie	21
4.4.1	Het gebruik van de ZEBRA	21
4.4.2	de ZEBRA-club: meester-trucologen	23

1 De Mathematische Afdeling en het onderzoek naar elektronische rekenapparaten

1.1 Rekenwerk aan de Mathematische Afdeling

Op 1 juli 1947 richt het Centraal Laboratorium van de PTT de Mathematische Afdeling op om het laboratorium te ondersteunen bij wiskundige problemen

en het uitvoeren van rekenwerk. L. Kosten kreeg de leiding over deze afdeling dat verder bestond uit de wiskundige A.L. Steiner, een aantal rekenaars en een aantal technici.¹ De Mathematische Afdeling hield zich vooral bezig met problemen uit de telecommunicatie, dat waren problemen die betrekking hadden op het telefoon-, telegraaf- en radioverkeer en problemen uit de communicatie theorie.

Net als bij andere rekencentra was het werk verdeeld in het opstellen van rekenschema's voor wiskundige problemen door de wiskundigen en het daarop uitvoeren van deze rekenschema's met behulp van tafelrekenmachines, tabellenboeken, pen en papier door de rekenaars. Kosten beschreef de werkzaamheden van de rekenaar in meer detail in 1953: 'De rekenaar slaat enige getallen aan in het toetsenbord en zet dan de machine in werking. Na enkele seconden is de machine met de bewerking klaar. De rekenaar moet dan:

1. de resultaten uit de machine aflezen en bv. op papier registreren;
2. in zijn rekenschema opzoeken welke volgende bewerking verricht moet worden;
3. de voor deze bewerking benodigde gegevens opzoeken en in de machine invoeren; dit kunnen beginggegevens of ook tijdelijk geregistreerde tussenresultaten zijn.

Hierna kan de machine weer aan het werken gezet worden.'²

Kosten kwam verder tot de conclusie dat in het rekenproces niet de rekenmachine de beperkende factor was maar de rekenaar. Om het rekenproces te versnellen was het dan ook nodig om de handelingen die de rekenaar uit moest voeren te mechaniseren. Voor het mechaniseren van het verwerken van grote hoeveelheden vergelijkbare gegevens, bijvoorbeeld in de statistiek of in de administratie, kon het gebruik van een ponskaartenmachine uitkomst bieden. Voor geavanceerd technisch-wetenschappelijk rekenwerk was de ponskaartenmachine echter niet flexibel genoeg: 'De handrekenaar kan echter zo nodig zijn gegevens dán weer van de éne plaats in zijn registreermedium (...) halen, dán weer van de andere plaats. Hij kan dus in dit medium "springen".'³

Deze flexibiliteit werd bereikt tijdens en na de Tweede Wereldoorlog door de bouw van een grote elektromechanische rekenmachine: de ENIAC. Over deze ENIAC en elektronische rekenmachines in het algemeen schreef Kosten in 1950 het artikel *Electronische rekenmachines*. Na een algemene en historische inleiding over "cijfermachines" behandelde hij een voor een de vijf standaardcomponenten van een elektronische rekenmachine: invoer, uitvoer, geheugen, besturing en rekenkundig orgaan.⁴ Hij besprak mogelijkheden, keuzen en oplossingen voor de verschillende organen, zoals het getalstelsel, de representatie van negatieve getallen en verschillende geheugenelementen. Deze volledig elektronische rekenmachines konden 'beschouwd worden als volledige mechanisering van het systeem: "rekenaar + rekenmachine + rekenschema". Met name is volledig "springen" in de getallenvoorraad mogelijk, terwijl ook de werksnelheid

¹D. van de Nieuwe Giessen, *Onderzoek en ontwikkeling bij KPN. Een geschiedenis van de eerste honderd jaar* (Leidschendam: KPN Research 1996), 161; W.L. van der Poel, 'The Early History of Computing at PTT', *Delft Progress Report 5* (1980), 174-184, aldaar 174

²L. Kosten, 'PTERA I. Wordingsgeschiedenis van PTERA; doel en mogelijkheden', *Het P T T-bedrijf* 5:4 (december 1953), 116-123, aldaar 116

³Ibidem

⁴L. Kosten, 'Electronische rekenmachines', *Het P T T-bedrijf* 3:1 (mei 1950), 1-16, aldaar 4

alsmede de hoeveelheid der gelijktijdig onthouden getallen nog toegenomen is.’⁵

1.2 De behoefte aan een rekenmachine

Deze karakterisaties van rekenaar, rekenproces en automatische rekenmachine schreef Kosten in 1953 in het artikel *Wordingsgeschiedenis van PTERA; doel en mogelijkheden* waarin hij verklaarde waarom bij het Centraal Laboratorium een behoefte bestond aan een grote elektronische rekenmachine en waarom de leiding van het Laboratorium besloot er zelf een te gaan bouwen. De Mathematische Afdeling hield zich voornamelijk bezig met problemen uit de telecommunicatie voor andere afdelingen van het laboratorium. Bij elke vergroting van de reken-capaciteit verdween die extra capaciteit weer snel door de groei van het aantal rekenopdrachten en de grootte van de rekenopdrachten. Er was dus sprake van een latente vraag naar rekencapaciteit.⁶

In 1947 bezocht de hoofddirecteur van de PTT, Maitland, de Verenigde Staten en hij was ook bij Howard Aiken in Harvard langs geweest die de bouw van de MARK rekenmachines begeleidde. Na zijn bezoek besloot de directie om te onderzoeken of de PTT zo’n machine zou kunnen gebruiken, of er op dit gebied met anderen samengewerkt kon worden en begon het laboratorium alvast met wat voorbereidend technisch onderzoek naar rekenmachines en onderdelen ervan. De conclusie van deze onderzoeken was dat de PTT zo’n elektronische rekenmachine goed zou kunnen gebruiken. Omdat samenwerking niet mogelijk bleek, besloot de PTT in 1949 zelf een computer te gaan bouwen.

Een van de belangrijkste problemen bij het Centraal Laboratorium waren de zogenaamde verkeersproblemen. Bij het ontwerpen van telefooncentrales was het van belang om de beste verhouding te vinden tussen het aantal verbindingen in de centrale en de kans dat de centrale volledig bezet zou zijn. Om deze verhouding te bepalen, berekende men blokkerings- en wachtkansen. Maar een klein deel van dergelijke problemen kon mathematisch worden bepaald. Het grootste deel van deze problemen lostte men op met behulp van simulaties van het verkeer in zo’n centrale.

Kosten promoveerde in 1942 op deze problemen en een deel van zijn proefschrift handelde over het meten aan ‘kunstmatig samengesteld verkeer’ en het creëren van dergelijk kunstmatig samengesteld verkeer. Een verkeerssimulatie kon met de hand gemaakt worden, maar ook automatisch met een elektro-mechanische machine. Bij het laboratorium van de PTT in Den Haag was zo’n verkeersmachine gebouwd en ook gebruikt.⁷

Zodra de mogelijkheden van elektronische rekenmachines bekend werden na de Tweede Wereldoorlog, werd geprobeerd deze technieken toe te passen in verkeersmachines. Ook het Centraal Laboratorium deed onderzoek naar elektronische verkeersmachines. Het bouwen van een verkeersmachine was een van de belangrijkste taken van Kostens Mathematische Afdeling.

Het voorbereidende werk met betrekking tot geheugens en rekenelementen was eind jaren ’40 dan ook allereerst gericht op de constructie van analoge verkeersmachines.⁸ De geheugens waarmee Kosten en zijn team experimenteerden,

⁵Kosten, ‘PTERA I. Wordingsgeschiedenis van PTERA; doel en mogelijkheden’, 117

⁶Ibidem

⁷L. Kosten, ‘Over blokkeerings- en wachtproblemen’, Proefschrift, Technische Hogeschool Delft (Delft 1942), 70–91

⁸W.L. van der Poel, *Een leven met computers* (TUDelft 1988), 11

waren een Williams buisgeheugen en magnetische trommelgeheugens.⁹ In 1950 had de Mathematische Afdeling een buisgeheugen ontwikkeld dat een rooster van 32 bij 32 bits een uur lang foutloos op een scherm kon vasthouden. Volgens Van der Poel heeft het nadien nooit meer gewerkt, dit geheugen bleek te onbetrouwbaar zodat het laboratorium stopte met de ontwikkeling ervan.¹⁰ Meer succes had de Mathematische Afdeling met het magnetische trommelgeheugen.

Het grootste probleem bij deze verkeersmachines was het maken van een betrouwbaar zuiver lootorgaan om de willekeurigheid in telefooncentrales te simuleren. Een lootorgaan creëerde pseudo-willekeurige getallen, bijvoorbeeld door gebruik te maken van stapschakelaars en een Geigerteller, een automatische roulette of met behulp van ruis. In 1950 diende Kosten in een patentaanvraag in waarin hij een verbetering van een binair elektrisch lootorgaan beschreef.¹¹ In 1951 bouwde de PTT samen met Philips Telecommunicatie Industrie in Hilversum een elektronische verkeersmachine.¹²

Ondertussen werd duidelijk dat verkeerssimulaties ook op een digitale rekenmachine uitgevoerd konden worden. Dergelijke rekenmachines waren veel flexibeler omdat ze naast verkeerssimulaties ook allerhande andere berekeningen konden uitvoeren. Bovenal was belangrijk dat met behulp van digitale rekenmachines het veel eenvoudiger was om betrouwbare pseudo-willekeurige getallen te genereren dan met welk lootorgaan dan ook.¹³ In 1949 besloot Kosten daarom besloten om zo'n digitale automatische rekenmachine bij de PTT te gaan bouwen. Niettemin bleef Kosten geïnteresseerd in analoge verkeersmachines, hij maakte samen met E.J. Gröneveld nog een ambitieus ontwerp voor een verkeersmachine. Meer dan een ontwerp zou het niet worden, het kwam nooit tot een realisatie.¹⁴

1.3 Van ARCO tot TESTUDO

Zoals eerder gezegd, besloot de Mathematische Afdeling in 1949 om zelf een computer te gaan bouwen. Voor dit project nam Kosten in 1950 de net afgestudeerde natuurkundestudent W.L. van der Poel in dienst. Van der Poel was namelijk bekend met het ontwerp en de bouw van grotere rekenmachines. Net na de Tweede Wereldoorlog schreef de Technische Hogeschool in Delft een prijsvraag uit waaraan Van der Poel meedeed met het ontwerp van een praktisch onuitvoerbare elektromechanische rekenmachine onder het motto: '1 + 1 = 10'¹⁵; In 1947 werd Van der Poel aangesteld als spuurwerkassistent¹⁶ bij de

⁹P.J. van Donselaar, 'De ontwikkeling van elektronische rekenmachines in Nederland (Een historisch overzicht van Nederlandse computers)', Technisch rapport (Amsterdam: Stichting Het Nederlands Studiecentrum voor Administratieve Automatisering en Bestuurlijke Informatieverwerking juli 1967), 12

¹⁰Van der Poel, 'The Early History of Computing at PTT', 175. Overigens meldde Van der Poel dat elders dit geheugen wel met succes was toegepast, bijvoorbeeld in de Feranti van Shell.

¹¹L. Kosten, 'Octrooi No. 75090: tweewaardig elektronisch lootorgaan' (1955), (URL:<http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=GB742305&F=0&QPN=GB742305>)

¹²L. Kosten, 'IV. Beschrijving van een machine voor kunstmatig verkeer op elektronische basis', *De Ingenieur* 44 (Electrotechniek 7) (1951), 62-66, aldaar 64

¹³Van der Poel, 'The Early History of Computing at PTT', 177

¹⁴Ibidem

¹⁵Interview met W.L. van der Poel gehouden op 17 juli 2007 door G. Alberts en H.T. de' (2007)

¹⁶Spuurwerkassistenten werden betaald uit het spuurwerkfonds dat in 1942 was ingesteld als onderdeel van het Delftse Hogeschoolfonds om studenten ervaring op te laten doen met

Bruyn om zijn ontwerp verder uit te werken.¹⁷

Ondertussen raakte professor A.C.S. van Heel geïnteresseerd in deze ontwikkeling. Van Heel hield zich bezig met de ontwikkeling van lenzenstelsels en daarbij kwam erg veel rekenwerk kijken. Een automatische rekenmachine speciaal bedoeld voor dit werk zou een enorme vooruitgang betekenen. Van der Poel was in 1949 klaar met het ontwerp van de Automatische Relais Calculator voor Optische Berekeningen (ARCO) en begon daarna met de bouw van deze machine.¹⁸ In dezelfde periode leerde Van der Poel Kosten kennen. Tijdens een bezoek van Kosten aan de Hogeschool vroeg Van der Poel hem of hij niet een partij relais over had bij de PTT. Vlak na de oorlog was dergelijk materiaal schaars en zonder eerst toestemming te krijgen van de leiding van de PTT schonk Kosten ongeveer 600 relais aan de Hogeschool.¹⁹

Van der Poel studeerde in 1950 af bij Van Heel en bij zijn vertrek werkte de besturing van de ARCO.²⁰ De rest van de machine bouwden C.J.A. Tepas, G.J. Beernink en F. ten Burg. In 1952 kon TNO de ARCO in gebruik nemen.²¹ TNO zette de rekenmachine in voor de berekening van lenzen die de NV Van Leers Optische Industrie in Delft produceerde.²²

Het geheugen van de ARCO was verdeeld in een levend deel en een dood deel. Een klein aantal werkregisters vormden het levende geheugen waarin de ARCO tussenberekeningen uitvoerde. Het dode geheugen bestond uit een pluggenbord en stapshakelaars waarmee de gebruiker de computer instrueerde. Omdat de ARCO instructies ook vanuit het levende geheugen uit kon voeren, kon de ARCO ook beschouwd worden als een stored-program computer.²³

Optellen en aftrekken duurden 30 seconden, de andere rekenkundige operaties, inclusief worteltrekken en de berekening van $(1 - x^2)$, duurden 45 seconden en de machine was daarmee langzamer dan een rekenaar op een Madas rekenmachine. Door deze snelheid, of het gebrek daaraan, werd de ARCO al snel TESTUDO genoemd, testudo betekende schildpad in het Latijn. Niettemin was de TESTUDO een verbetering ten opzichte van de eerdere situatie met rekenaars en Madas tafelrekenmachines: de TESTUDO mocht dan langzaam zijn, hij kon uren, zelfs nachtenlang doorwerken en versloeg daarmee de menselijke rekenaar.²⁴ De bruikbaarheid van de TESTUDO werd eens te meer bewezen door zijn lange dienstbare leven van 12 jaren.

In de late jaren '40 groeide Van der Poel uit tot een van de experts in Nederland op het gebied van computers, voornamelijk door zijn werk aan de ARCO. In 1947 volgde hij een aantal lezingen aan het net opgerichte Mathematisch Cen-

toegepast-wetenschappelijk onderzoek. Een geschiedenis van het Delftse Hogeschoolfonds is te vinden in: Fons Alkemade, 'OD95. 1925–2000: van Delftsch Hoogeschoolfonds naar UFD', *Delft Integraal* 5 (2000), (URL:<http://www.delftintegraal.tudelft.nl/info/index7acb.html?hoofdstuk=Artikel&ArtID=1973>)

¹⁷Van der Poel, *Een leven met computers*, 8–9

¹⁸W.L. van der Poel, 'Curriculum Vitae of Willem Louis van der Poel' (2003), (URL:<http://sepc.twi.tudelft.nl/~poel/packages/curricul.44q>)

¹⁹Van der Poel, 'The Early History of Computing at PTT', 175

²⁰Van der Poel, 'Curriculum Vitae of Willem Louis van der Poel'

²¹Van der Poel, *Een leven met computers*, 9–10

²²R. van der Veen, 'Een schildpad als rekentuig. De Testudo, voorloper van de huidige p.c.', *Delft Integraal* 1 (2005), (URL:<http://www.tudelft.nl/live/pagina.jsp?id=339fb018-21db-4bcc-a5fe-a105ce2c8c14&lang=nl&binary=/doc/DI051-32-34.pdf>), 32–34, aldaar 32

²³Van der Poel, 'Curriculum Vitae of Willem Louis van der Poel'

²⁴Van der Poel, *Een leven met computers*, 9–10

trum in Amsterdam waarin Hartree sprak over moderne rekenapparaten en de ENIAC in het bijzonder.²⁵ Daar leerde Van der Poel een andere vooraanstaande figuur op het gebied van computers in Nederland kennen: Van Wijngaarden. In mei 1949 hield Van der Poel een lezing op het Symposium Moderne Rekenmachines met de titel *Schakelingen van automatische cijfermachines*²⁶ waarin hij voornamelijk schakelingen van optellers behandelde. Een jaar later ging hij, net als Van Wijngaarden, naar Engeland waar onder leiding van Wilkes de eerste programmeercursus plaats had.²⁷

2 PTERA

2.1 Ontwikkeling en bouw van de PTERA

De conclusie van het voorbereidende onderzoek naar elementen van rekenmachines dat de Mathematische Afdeling tussen 1947 en 1949 uitvoerde, was dat ‘een machine met een betrekkelijk langzaam rekenkundig orgaan met volledige parallelbehandeling van het vermenigvuldigen en delen, en uitgerust met een magnetische trommel als registreerorgaan der getallen, een aantrekkelijke oplossing bood, welke qua kosten niet exorbitant zou zijn.’²⁸ Tevens werden prototypes gemaakt van de elementen van een rekenkundig orgaan.

Mede gebaseerd op deze resultaten besloot de PTT in 1949 tot de bouw van een computer, de PTT Electronische Rekenautomaat (PTERA). In 1950 haalde Kosten Van der Poel binnen bij de Mathematische Afdeling om deze PTERA te ontwikkelen. Omdat er grote behoefte was aan deze rekenmachine om verkeersberekeningen uit te kunnen voeren werd de PTERA onder grote druk ‘in elkaar geflanst’²⁹. Samen met de technici G.J. de Zwart en C.F.W. van Weezel en de Werkplaats bouwde Van de Poel de PTERA in krap drie jaar tijd.³⁰ Op 14 september 1953 nam directeur-generaal Neher de PTERA officieel in gebruik.³¹ Het Centraal Laboratorium zou later naar hem genoemd worden.

De PTERA bestond, zoals gebruikelijk, uit de vijf welbekende elementen: invoer, uitvoer, besturing, geheugen en rekenkundig orgaan. De invoer verliep via een zelfgemaakte foto-elektrische telex-ponsbandlezer. Eerder was geprobeerd om de invoer via telex-over-radioapparatuur van de PTT te laten verlopen, maar dat bleek te langzaam. De uitvoer verliep via een gewone typemachine waarbij de PTERA via magneetjes onder bepaalde toetsen die toetsen kon aanslaan. Een andere mogelijkheid was het ponsen via een telex-ponsmachine waardoor ‘geleverde resultaten te eniger tijd weer als “voer” voor PTERA gebruikt [konden] worden’³².

²⁵Van der Poel, ‘Curriculum Vitae of Willem Louis van der Poel’ Van deze lezingen is een rapport bij het Mathematisch Centrum verschenen: D.R. Hartree, ‘Lectures on modern calculating machines’, Technisch rapport CR-7 (Amsterdam: Mathematisch Centrum 1947)

²⁶W.L. van der Poel, ‘Schakelingen van automatische cijfermachines’, *Nederlands Tijdschrift voor Natuurkunde* 15 (1949), 13–22

²⁷Van der Poel, ‘Curriculum Vitae of Willem Louis van der Poel’

²⁸Kosten, ‘PTERA I. Wordingsgeschiedenis van PTERA; doel en mogelijkheden’, 11

²⁹Interview met W.L. van der Poel’

³⁰‘De wonderen zijn de wereld nog niet uit. Indienststelling van PTERA’, *PTT-bedrijfsbanden* 10 (oktober 1953), 295–297, aldaar 295

³¹Ibidem

³²Kosten, ‘PTERA I. Wordingsgeschiedenis van PTERA; doel en mogelijkheden’, 119–120

Het rekenkundige orgaan was in staat om op te tellen, af te trekken, te vermenigvuldigen en te delen en gebruikte daarbij drie registers: A, B, en R. De accumulator A en register B vormden samen het dubbele lengte getal na vermenigvuldiging. Het register R was als enige register uitgevoerd met relais, de rest was elektronisch. De PTERA gebruikte register R bij vermenigvuldigen en delen en bevatte respectievelijk het vermenigvuldiggetal of de deler.

De besturing gebruikte het vierde register, C, om er de volgende uit te voeren instructie in te bewaren. Het ophalen van een instructie was zelf ook een instructie, die door de PTERA zelf toevoegde, de programmeur hoefde daar geen rekening mee te houden. Tijdens het ophalen van een instructie onthield de besturing het adres waaruit deze instructie kwam in het register B en vermeerderde dat adres met een.³³

Het geheugen van de PTERA was een magnetische trommel met een capaciteit van 2048 woorden van 31 bits. Alhoewel er voor elk van de 64 sporen een lees en schrijfkop aanwezig was, kon maar een kop tegelijkertijd een actie uitvoeren; het geheugen werkte dus serieel. Op een spoor pasten 32 woorden die echter niet achter elkaar werden geschreven, maar bitsgewijs, dus eerst alle eerste bits, dan alle tweede bits, enzovoorts.³⁴ Door deze indeling van een spoor in het geheugen was de machine twee keer minder snel, maar daardoor was het mogelijk de rest van de machine eenvoudiger te maken.³⁵ De trommel wekte ook het kloksignaal op. Aan het uiteinde van de ronddraaiende trommel was een gegraveerde ijzeren schijf met 992 tandjes gemonteerd die 992 klokimpulsen per omwenteling genereerden.³⁶

2.2 De machinecode van de PTERA

Volgens Van der Poel bestond het programmeren uit een aantal stappen. Als eerste werd een stroomdiagram opgesteld waarin de programmeur de handelingen die het programma moest verrichten aangaf in ‘normale woordentaal of wiskundige taal’.³⁷ Hierna moest de programmeur de verschillende blokken in het stroomdiagram uitwerken in reeksen instructies. Voor veel voorkomende functies en subprogramma’s maakte de programmeur subroutines die steeds opnieuw gebruikt konden worden, eventueel aangeroepen met andere parameters. Dergelijke subroutines heetten standaard subroutines en samen vormden ze een subroutinebibliotheek. Deze bibliotheek bepaalde voor een groot deel de mogelijkheden van de machine.³⁸

De instructies uit de uitgewerkte blokken van het stroomdiagram vormden samen een programma dat de PTERA, eenmaal in het geheugen gebracht, uitvoerde. Bij het invoeren van het programma gaf de programmeur naast deze instructies ook invoeraanwijzingen mee die de computer vertelden hoe en waar delen van het programma in het geheugen te plaatsen. Het inzetprogramma bepaalde dan ook voornamelijk de code waarin een programma werd geschreven

³³W.L. van der Poel, ‘PTERA II. De werking van PTERA’, *Het P T T-bedrijf* 5:4 (december 1953), 124–134, aldaar 126–127

³⁴Interleaving

³⁵Ibidem, 124

³⁶Ibidem, 131

³⁷W.L. van der Poel, ‘PTERA III. Het programmeren voor PTERA’, *Het P T T-bedrijf* 5:4 (december 1953), 135–147, aldaar 135

³⁸Ibidem

00	Stop en optellen
01	Ponsband lezen
10	Opbergen
11	Typen en deconverteren
20	Sprong
21	Afronden
30	Test en optellen
31	Conjunctie
40	Optellen
50	Vorbereiden en optellen
60	Vermenigvuldigen
70	Delen

Figuur 1: De twaalf hoofdgroepen van instructies van de PTERA. Uit: Van der Poel, ‘De PTERA’, 92–93.

om ingevoerd te kunnen worden. Het inzetprogramma had de taak om programma’s die geschreven waren in die code om te zetten in machinecode en op de juiste plaats in het geheugen te zetten.

De laatste stap van het programmeren was het ponsen van het programma op band waarbij de programmeur het programma als het ware assembleerde: hij kopiëerde subroutines van de standaardbandjes uit de “programmotheek”. Hierna kon de gebruiker de band aanbieden aan de machine via het inzetprogramma.³⁹

De PTERA was een één-adrescodemachine waarbij elke instructie uit een operatie en een adres bestond. Overigens was het ook mogelijk dat een woord een getal bevatte, een breuk tussen -1 en 1. Een woord was 31 bits lang waarvan het eerste bit het teken aangaf. Negatieve getallen werden aangegeven met het complement van het positieve getal.

De PTERA kende 54 verschillende instructies die over twaalf hoofdgroepen waren verdeeld (Zie Figuur 1). Elk van deze hoofdgroepen had een aantal varianten, groepen 40 tot en met 70 hadden acht varianten en de andere groepen elk vier. Deze varianten werden met een tweede getal aangegeven zodat een opdracht door middel van twee getallen van drie bits werden aangegeven: hoofdgroep en variant.⁴⁰ De hoofdgroep bepaalde dus het type van de opdracht, zoals optellen, invoer, uitvoer, springen, enzovoorts.

Het getal van de variant werd gecodeerd met functionele bits. Het eerste bit bepaalde het schoonmaken van de accumulator. Het tweede bit gaf aan of het rekenorgaan een operatie positief dan wel negatief uitvoerde. Het laatste bit was bedoeld om aan te geven of register R veranderde of niet. Dit laatste bit had enkel invloed bij de hoofdgroepen 40 tot en met 70.⁴¹

Het programmeren van de PTERA, of beter gezegd, het leren programmeren van de PTERA zag Van der Poel niet als een probleem. ‘Theoretisch heeft men aan de tabel, welke de werking van de operaties verklaart voldoende om ze te kunnen gebruiken. In de praktijk echter zal men meer hebben aan enkele representatieve voorbeelden.’⁴² De aandacht ging voornamelijk uit naar de

³⁹Ibidem, 135–136

⁴⁰W.L. van der Poel, ‘De PTERA’, in: *Nederlands rekenmachine genootschap 1959 – 1969. Colloquium Moderne Rekenmachine I* (Amsterdam 1969), 88–94, aldaar 92–93

⁴¹Van der Poel, ‘PTERA III. Het programmeren voor PTERA’, 136–137

⁴²Ibidem, 137

machine zelf: ‘De machine was omvangrijk en besloeg een hele kamer maar de opdrachtcode stond op een klein kaartje.’⁴³

2.3 De programmatuur van de PTERA

Voor de programmeur was de PTERA echter meer dan de machine alleen. Om de machine te kunnen gebruiken, of om het gebruik ervan makkelijker te maken voor de programmeur, waren er een aantal systeemprogramma’s⁴⁴ geschreven. Het meest belangrijkste was wel het inzetprogramma dat al even aan bod is gekomen. De PTERA kende echter ook nog een aantal andere systeemprogramma’s, zoals het adresboekprogramma, een programma om drijvende adressering mogelijk te maken en Van der Poel schreef een artikel over zogenaamde do-de programma’s⁴⁵. Verder waren er programma’s voor het snel invoeren van breuken of instructies.⁴⁶

Het inzetprogramma was altijd in de PTERA aanwezig en zorgde voor het invoeren van instructies en getallen waarbij het programma de conversie van het decimale getalstelsel naar het binaire uitvoerde. Verder verzorgde het inzetprogramma voor relatieve adressering van subroutines. Daarnaast moest volgens Van der Poel een inzetprogramma voldoen aan een vijftal eisen. Allereerst moest het inzetprogramma zelfstartend zijn, als de registers schoon waren bij het starten van de machine, dan startte het inzetprogramma automatisch. Daarnaast maakte het programma het invoeren van niet-significante cijfers overbodig. Het inzetten moest op elke willekeurige plaats in het geheugen kunnen beginnen, aangegeven met een inzetinstructie. De vierde eis was dat getallen en instructies door elkaar heen ingevoerd konden worden en tenslotte moest het mogelijk zijn het inzetprogramma tijdelijk te verlaten om de machine andere handelingen te laten verrichten waarna het inzetprogramma in dezelfde toestand weer verder moest kunnen gaan.⁴⁷

Het inzetprogramma bracht programma’s van de band in het geheugen van de PTERA, maar zichzelf inbrengen ging natuurlijk niet. Om het inzetprogramma in het geheugen te krijgen gebruikte de operator pre-inzetprogramma’s. Het allereerste pre-inzetprogramma moest met de hand ingebracht worden via schakelaars. Dit programma kon dan het volgende pre-inzetprogramma lezen die in een speciale code op band stond, enzovoorts, totdat het echte invoerprogramma ingevoerd was.⁴⁸

Bij het programmeren schreef de programmeur een hoofdprogramma dat een aantal subroutines aanriep die op hun beurt ook weer subroutines aan konden roepen; een programma bestond dus uit een aanroepiërarchie van subroutines. Als een programma of subroutine een andere subroutine aan wilde roepen, dan moest de plaats van die aan te roepen subroutine bekend zijn. De programmeur

⁴³W.L. van der Poel, ‘De ontwikkeling van de computer in Nederland.’, *Histechnicon* 15:1 (april 1989), 2–9, aldaar 3. Er was overigens wel een handleiding voor het programmeren voor de PTERA: W.L. van der Poel, ‘Het programmeren voor de PTERA’, Technisch rapport 12 MA (Centraal Laboratorium PTT 1952).

⁴⁴Van der Poel gaf geen naam aan dit type programmatuur.

⁴⁵W.L. van der Poel, ‘Dead programmes for a magnetic drum automatic computer’, *Applied Scientific Research B* 3 (1954), 190–198

⁴⁶W.L. van der Poel, ‘The logical principles of some simple computers’, Proefschrift, Universiteit van Amsterdam (1956), 43–48

⁴⁷Van der Poel, ‘PTERA III. Het programmeren voor PTERA’, 141–144; Van der Poel, ‘The logical principles of some simple computers’, 28–33

⁴⁸Ibidem, 35. Dit proces werd later bootstrapping genoemd.

moest precies weten en bijhouden waar welke subroutine in het geheugen te vinden was. Om de programmeur te ontlasten van deze administratieve rompslomp was er een adresboekprogramma ontwikkeld voor de PTERA dat net als het inzetprogramma altijd in het geheugen aanwezig was.

Het adresboekprogramma werkte als volgt: programma's die lager in de aanroephiërarchie stonden moesten eerder ingelezen worden dan programma's die hoger in de hiërarchie stonden. Het hoofdprogramma moest dus altijd als laatste ingelezen worden. De programmeur moest elk subprogramma afsluiten met inzetaanwijzing "2/". Het adresboekprogramma wist dan waar het volgende subprogramma zou beginnen in het geheugen, namelijk het adres dat volgde op het adres waar de laatste instructie voor "2/" in het geheugen was geschreven. Dit beginadres werd dan in het adresboek opgenomen en kreeg een volgnummer. Met behulp van dat volgnummer kon dan het beginadres van het subprogramma opgevraagd en gebruikt worden. Om meerdere hoofdprogramma's tegelijkertijd in het geheugen te hebben, was er een superadresboekprogramma dat alle beginadressen van de hoofdprogramma's bevatte.⁴⁹

Het gebruik van drijvende adressen (floating addresses) was gebaseerd op de observatie dat adressen van de meeste instructies niet van belang zijn omdat er nooit naar verwezen werd met een sprongopdracht. Het idee was om enkel deze belangrijke adressen, de drijvende adressen, in een lijst te bewaren waarna naar deze adressen verwezen kan worden door het volgnummer ervan in de lijst aan te geven. Het voordeel van drijvende adressen was dat in veel gevallen een programma aangepast kon worden zonder het hele programma te hernummeren. Het nadeel was echter dat het debuggen van programma's een stuk moeilijker was: het eigenlijke adres waarop iets fout ging in relatie tot het programma was onbekend.⁵⁰

Veelgebruikte programma's werden vaak permanent in het geheugen opgeborgen. Omdat andere programma's het gehele geheugen konden lezen en beschrijven, was het mogelijk dat door een fout in een programma dat programma geheugenplaatsen herschreef die door standaardprogramma's al in gebruik waren. Om dit probleem op te lossen werd het zogenaamde dode geheugen geïntroduceerd: een stuk van het geheugen waarop niet geschreven kon worden.

De subroutines in het dode geheugen schreven zelf ook naar het geheugen, het terugkeeradres moest bij aanroep achteraan de subroutine geschreven worden. In een dood geheugen kon dat niet meer. De oplossing was het afhandelen van begin en einde van elke subroutine te standaardiseren en uit te laten voeren door een speciaal subprogramma. Bij aanroep van een subroutine werd dit programma aangeroepen en het terugkeeradres opgeslagen. Bij het verlaten van de subroutine werd het programma weer aangeroepen om het terugkeeradres op te halen. Dit programma kon voor het grootste deel in het dode geheugen worden gehuisvest. Voor een klein deel van dit programma, bijvoorbeeld voor het opslaan van de terugkeeradressen, kon een stuk van het levende geheugen worden gebruikt.⁵¹

Deze programmatuur stond de programmeur terzijde en nam een aantal ver-

⁴⁹Van der Poel, 'PTERA III. Het programmeren voor PTERA', 144-145; Van der Poel, 'The logical principles of some simple computers', 33-35

⁵⁰W.L. van der Poel, 'Floating addresses in an automatic digital computer', in: *Nederlands rekenmachine genootschap 1959 - 1969. Colloquium Moderne Rekenmachine II* (Amsterdam 1969), 19-27; Van der Poel, 'The logical principles of some simple computers', 40-43

⁵¹Van der Poel, 'Dead programmes for a magnetic drum automatic computer'

velende taken over of beschermd de programmeur tegen fouten. Van der Poel merkte aan het einde van zijn artikel over het programmeren van de PTERA op dat nog meer taken overgenomen konden worden. Bij het rekenen met complexe getallen of dubbele lengte getallen moest de programmeur voor elke handeling met zo'n complex getal of dubbel lengte getal verschillende machinecode instructies opgeven. Het was mogelijk om een interpretatief programma te maken dat deze machinecode instructies automatisch uitvoerde bij het lezen van instructies in een nieuwe code die wel direct met complexe getallen of dubbel lengte getallen kon werken.⁵²

2.4 Het gebruik van de PTERA

Over het gebruik van de PTERA is niet veel bekend. Kosten schreef in de reeks artikelen over de PTERA in *Het P T T-bedrijf* een artikel over een rekenopdracht, "A7"⁵³. Verder schreven Kosten en Van der Poel hier en daar wat over de prestaties van de PTERA en waarvoor ze de machine in grote lijnen gebruikten. Kosten schreef dat de PTERA in de eerste plaats bedoeld was als rekenmachine en niet-rekenkundige opgaven zoals het sorteren van gegevens en eenvoudig werk in boekhouding en statistiek konden beter met een ponskaartsysteem of tafelrekenmachine gedaan worden. De PTERA was ook niet geschikt voor "real-time" problemen. Over de PTERA zei hij dat het 'zeer flexibel in gebruik was'.⁵⁴ Zo werden er pseudo-willekeurige getallen mee gegenereerd volgens de methode *Razende Random* dat op het Mathematisch Centrum in Amsterdam ontwikkeld was.⁵⁵

De PTERA werd in elk geval ingezet voor rekenopdrachten van andere afdelingen van het Centraal Laboratorium, zoals filterberekeningen voor draaggolftelefonie of het berekenen van een kabelbreuk in onderzeese kabels. Maar soms werden ook opdrachten uitgevoerd voor derden, zoals voor Philips waarvoor aan ionosfeerproblemen werd gerekend.⁵⁶

Het artikel over "A7" liet zien hoe bij de Mathematische Afdeling een wiskundig probleem werd opgelost met behulp van de PTERA. Het probleem werd eerst wiskundig geanalyseerd. Stap voor stap paste de programmeur de wiskundige formulering van het probleem aan om het beter geschikt te maken voor berekening door een automatische rekenmachine. Belangrijk hierbij was het schalen van getallen omdat in de PTERA getallen werden voorgesteld als breuken tussen -1 en 1 .

Nadat de wiskundige analyse klaar was, maakte de programmeur een stroomdiagram die dat diagram vervolgens uitwerkte tot een programma. Dit omzetten was 'geen moeilijk werk, doch moet wel zeer zorgvuldig gebeuren',⁵⁷ om fouten te voorkomen. De programmeur annotateerde het programma in de kantlijn met opmerkingen en symbolen om het werken met het programma, het lezen ervan of het zoeken van fouten eenvoudiger te maken. Door middel van nummers in de blokken van het stroomdiagram te vermelden in de annotatie van de code die zo'n blok implementeerde maakte de programmeur de connectie tussen de

⁵²Van der Poel, 'PTERA III. Het programmeren voor PTERA', 147

⁵³L. Kosten, 'PTERA IV. "A 7"', *Het P T T-bedrijf* 5:4 (december 1953), 148-163

⁵⁴Kosten, 'PTERA I. Wordingsgeschiedenis van PTERA; doel en mogelijkheden', 122

⁵⁵Ibidem

⁵⁶Interview met W.L. van der Poel

⁵⁷Kosten, 'PTERA IV. "A 7"', 152

twee duidelijk.

Als het programma klaar was, bevatte het vaak nog fouten. Ponsfouten konden opgespoord worden door twee banden te ponsen en deze met elkaar te vergelijken. Daar waar ze van elkaar afweken was dan een fout gemaakt in een van de twee banden. Deze controle voerde de PTERA zelf uit: er was een speciaal programma dat de ponsband instructie voor instructie uit kon typen. Programmeerfouten konden alleen gevonden worden door het programma uit te laten voeren. Deze fouten waren moeilijker op te sporen. Om de programmeur te helpen bij het debuggen was er een geheugen-uittypprogramma gemaakt dat de inhoud van de registers uit kon typen. Zodra het programma foutvrij was, kon het gebruikt worden.⁵⁸

Tijdens het gebruik van de PTERA bleken sommige programma's vaker gebruikt dan andere. Bij elk gebruik van zo'n programma moest het hele programma ingelezen worden en de relatieve adressering berekend. Omdat dit relatief langzaam ging, was het beter om deze programma's nadat ze eenmaal ingelezen waren weer uit te ponsen. Deze door de machine geponste programma's konden door een speciaal inzetprogramma extra snel ingelezen worden.⁵⁹

Over de prestatie van de PTERA schreef Kosten het volgende: 'Sinds de indienststelling in Sept. 1953 is PTERA vrijwel onafgebroken in bedrijf geweest. Sinds 1 Jan. 1954 is de bedrijfstijd zelfs verlengd tot 14½ uur per werkdag. Gehoopt was, dat een tijdsrendement van 80 % gehaald zou kunnen worden, welk getal gunstig ligt vergeleken bij andere electronische automaten (gemiddeld 60 %). Gebleken is, dat dit cijfer zeker tussen de 80 en 90 % ligt.'⁶⁰ Of dit rooskleurige beeld van de PTERA volkomen naar waarheid was, valt te betwijfelen. Volgens Van der Poel was 'de PTERA (...) dan ook echt een zorgenkindje. De machine heeft 50% van zijn tijd nuttig gedraaid en de andere 50% was hij in reparatie.'⁶¹

3 ZERO

Tijdens de bouw van de PTERA werden ideeën ontwikkeld voor een volgende machine. De onderdelen van de PTERA die al gereed waren, zoals het magnetische trommelgeheugen en een deel van het rekenorgaan, gebruikt Van der Poel om zijn ideeën uit te werken in een prototype, de ZERO, dat in 1951 gereed kwam.⁶² De bouw en het gebruik van de ZERO was echter voornamelijk bedoeld om ideeën voor de volgende machine uit te werken.⁶³ Zodra de onderdelen van de ZERO nodig waren voor de bouw van de PTERA demonteerde Van der Poel de ZERO.⁶⁴ Van der Poels 'gaafste creatie'⁶⁵ heeft maar veertien dagen bestaan.⁶⁶

De ZERO was allereerst een oefening in vereenvoudiging. Zo bevatte het rekenkundige orgaan geen vermenigvuldiger of deler, die operaties moesten uit-

⁵⁸Ibidem

⁵⁹Van der Poel, 'PTERA III. Het programmeren voor PTERA', 147

⁶⁰Kosten, 'PTERA I. Wordingsgeschiedenis van PTERA; doel en mogelijkheden', 122

⁶¹Van der Poel, 'De ontwikkeling van de computer in Nederland.', 3

⁶²Van der Poel, *Een leven met computers*, 17-18

⁶³'Interview met W.L. van der Poel'

⁶⁴Van der Poel, *Een leven met computers*, 18

⁶⁵Ibidem

⁶⁶Van de Nieuwe Giessen, *Onderzoek en ontwikkeling bij KPN*, 163

instructie	X	R	S	I	uitleg
Xn	0	0	0	0	Haal een instructie op uit register n .
Pn	1	0	0	0	Tel de inhoud van n op bij een schoongemaakte accumulator.
Qn	1	0	0	1	Trek de inhoud van n af van een schoongemaakte accumulator.
On	1	0	1	0	Tel de inhoud van n op bij de inhoud van de accumulator.
An	1	0	1	1	Trek de inhoud van n af van de inhoud van de accumulator.
Sn	1	1	0	0	Berg het getal in de accumulator op in register n en maak de accumulator schoon.
Rn	1	1	1	0	Berg het getal in de accumulator op in register n .

De cijfers X, R, S, en I hadden de volgende betekenis:

X: een X-opdracht of een niet-X-opdracht;

R: een lees (read) of schrijf (write) opdracht;

S: schoonmaken;

I: inverse operatie, dus aftrekken in plaats van optellen.

Figuur 2: De zeven opdrachten van de ZERO en hun functionele bit codering, uit: Van der Poel, ‘A simple electronic digital computer’.

gevoerd worden met behulp van een (micro)programma. De besturing en het rekenkundige orgaan deelden drie registers en het optelmechanisme en waren daardoor niet meer van elkaar te onderscheiden.⁶⁷

Later werkte Van der Poel deze vereenvoudiging uit tot de vraag *Welke opdrachten zijn essentieel in een rekenmachine?*⁶⁸ Dezelfde exercitie zou hij in het derde deel van zijn proefschrift herhalen.⁶⁹ Hij toonde aan dat er maar één opdracht werkelijk essentieel was, namelijk de nieuw geïntroduceerde opdracht Nn : trek het getal op adres n af van de accumulator en berg het resultaat weer op in n . Met behulp van enkele speciale adressen die een bepaalde actie uitvoerden bij het schrijven in of lezen uit zo’n adres, bijvoorbeeld voor in- en uitvoer of het springen in het programma, kon alle functionaliteit van een computer met deze opdracht N gerealiseerd worden.

De ZERO kende een zevental verschillende opdrachten (zie ook Figuur 2): het ophalen van een instructie uit het geheugen; het opbergen van de inhoud van de accumulator in het geheugen, met of zonder schoonmaken van de accumulator achteraf; en het optellen en aftrekken van een getal uit het geheugen bij de inhoud van de accumulator, eventueel met het vooraf schoonmaken van de accumulator. Deze opdrachten konden verdeeld worden in een aantal klassen. Behoorde een opdracht tot een bepaalde klasse, dan had het bit dat die klasse representeerde in de opdrachtcode de waarde 0 en anders de waarde 1.

De opdrachtcode bestond uit vier functionele bits. Het eerste bit, X (extract), bepaalde of de opdracht de opdracht Xn was of een van de andere opdrachten. Met het bit R (read) kon aangegeven worden of een opdracht uit het geheugen las of juist in het geheugen schreef. Het schoonmaken van de accumulator werd gespecificeerd met het derde bit (S). Het laatste bit, I, werd gebruikt om het onderscheid tussen optellen en aftrekken te maken. De func-

⁶⁷W.L. van der Poel, ‘A simple electronic digital computer’, *Applied Scientific Research B* 2:5 (1952), 367–400. In dit artikel beschrijft Van der Poel de ZERO en, tenzij anders aangegeven, is deze bespreking van de ZERO gebaseerd op de inhoud van dit artikel.

⁶⁸Ibidem

⁶⁹Van der Poel, ‘The logical principles of some simple computers’, 95–106

van de accumulator naar het hulpregister en naar het instructieregister.

Met behulp van de X-opdracht kon een sprong gemaakt worden, bijvoorbeeld naar een subroutine. Laat in een programma de sprongopdracht X_m voorkomen en deze opdracht geschreven worden in het instructieregister door de instructie X_n . Nadat X_m was uitgevoerd bevond zich in de accumulator de instructie X_{m+1} en in het hulpregister X_{n+1} . Met andere woorden, de terugkeerinstructie was in het hulpregister te vinden. Aan het begin van de subroutine moest dan deze instructie achteraan de subroutine geplaatst worden.

Met de X-opdracht konden ook conditionele sprongen gemaakt worden doordat bij het verplaatsen van de inhoud van accumulator naar het hulpregister het laatste bit, het tekenbit, van dat getal in het tekenregister werd geschreven (in Figuur 3 aangeduid met het vakje “sign”) dat uit een flip-flop bestond. Bij het uitvoeren van een X-opdracht werd dit teken tegelijkertijd met het getal één opgeteld bij die X-opdracht en in de accumulator geschreven. Dit teken was na het uitvoeren van een niet-X-opdracht altijd nul en had normaal gesproken dus geen invloed op de gang van zaken.

Enkel bij het uitvoeren van een expliciete X-opdracht kon dit tekenbit een rol spelen. Om een normale sprong uit te voeren moest de accumulator dan ook in de vorige opdracht schoongemaakt zijn of een positief getal bevatten. Voor het uitvoeren van een conditionele sprong werd een dummy X-opdracht uitgevoerd, een expliciete X-opdracht naar de volgende instructie. Als tijdens het uitvoeren van de vorige instructie in het programma de waarde in de accumulator negatief was, en het tekenbit van de accumulator dus 1, werd de volgende instructie in het programma overgeslagen omdat bij de volgende impliciete X-opdracht nu niet het getal één maar het getal twee werd opgeteld.

In- en uitvoer was in de ZERO mogelijk door het schrijven of lezen van twee speciale vijf-bits adressen, 30 en 31. De vijf bits representeerden een symbool. Het lezen van de band gebeurde door de opdrachten P31, Q31, O31 of A31 waardoor de vijf bits uit adres 31 op de vijf minst significante bits van de accumulator werden geschreven, terwijl tegelijkertijd de band een regel opschoof. De uitvoer verliep op een vergelijkbare manier: de opdrachten R30 en S30 plaatsten de vijf meest significante bits van het hulpregister in adres 30 die daarop uitgetypt of uitgeponst werden.

De ZERO was een eenvoudige machine, de programmering was flexibel en niet moeilijk te leren omdat er maar weinig opdrachten waren. Door veel subroutines te gebruiken was het schrijven van grotere programma's ook geen probleem. Toch was de machine niet praktisch bruikbaar, het was gewoonweg te langzaam. Aan de andere kant maakte dat de ZERO uitermate geschikt om mee te spelen; op initiatief van Kosten programmeerde Van der Poel het spelletje NIM voor de ZERO.⁷⁰

Eind 1951 zond Van der Poel een artikel over de ZERO in bij het tijdschrift Applied Scientific Research met de titel: *A simple electronic digital computer*. Door dit in 1952 verschenen artikel verkreeg Van der Poel internationale bekendheid.⁷¹

⁷⁰Van der Poel, 'A simple electronic digital computer', 396

⁷¹'Interview met W.L. van der Poel'

4 ZEBRA

4.1 Het ontstaan van de Stantec ZEBRA

Al snel na de ingebruikname van de PTERA bleek dat de PTERA het groeiende rekenwerk al snel niet meer aan zou kunnen en in 1954 besloot de PTT daarom om een betere rekenmachine te ontwikkelen. Begin 1955 had Van der Poel het logische ontwerp van deze nieuwe machine af. Verschillende andere rijksinstellingen, zoals de Technische Hogeschool Delft en het Nationaal Luchtvaartlaboratorium (NLL), toonden interesse voor deze nieuwe rekenmachine.⁷²

Alhoewel de PTT graag aan de verschillende verzoeken wilde voldoen om een computer te leveren, was van meet af aan ook duidelijk dat PTT geen grootschalige computerbouwactiviteiten wilde ontplooiën of helpen een computerindustrie op te starten. Voor die laatste optie was veel interesse van professor Engelfried van levensverzekeringsmaatschappij de Nillmij: het bedrijf was bereid een substantieel bedrag te investeren in een al bestaande industrie om computeractiviteiten te ontwikkelen of om zelfs een volledig nieuwe computerindustrie op te zetten. PTT wilde echter niet de kennis en ervaring van de mensen van de Mathematische Afdeling opgeven; het produceren van computers was niet de taak van het Dr. Neher Laboratorium.⁷³

De directie besloot om op zoek te gaan naar een bestaande industrie die bereid zou zijn om de nieuwe PTT computer verder door te ontwikkelen, te produceren en op de markt te brengen. J. Schoo, de directeur-generaal van PTT, zond daarop op 18 februari 1955 een brief naar zowel Philips Telecommunicatie als naar de Nederlandsche Standard Electric Mij (NSEM) met het verzoek om informatie over computerbouwactiviteiten en -mogelijkheden bij beide bedrijven.⁷⁴ PTT had goede contacten en ervaringen met beide bedrijven, die, net al PTT, actief waren in de Nederlandse telecommunicatie. Verder waren er contacten met Zuse in Duitsland, K. Zuse was in 1955 hiervoor bij het Centraal Laboratorium op bezoek geweest, maar zonder verder resultaat.⁷⁵

Nog voor NSEM of Philips reageerde, kwam van de Nederlandsche Röntgen-Apparatenfabriek een brief waarin het bedrijf aanbood de PTT rekenmachine te gaan bouwen.⁷⁶ De PTT ging hier niet verder op in. De Nederlandse Seintoe-stellenfabriek, onderdeel van Philips Telecommunicatie Industrie (PTI), toonde grote interesse in de bouw van de PTT rekenmachine⁷⁷, toch haakte Philips snel af. Philips kon zich vanwege contractuele verplichtingen met IBM niet met de commerciële productie van computers bezighouden. Overigens was Philips wel

⁷²Brief van F. Loonstra, voorzitter Afdeling Algemene Wetenschappen van de TH Delft aan H.C.A. van Duuren, directeur van het Dr. Neher Laboratorium, 25 januari 1955; Brief van de directeur-generaal van PTT aan J. Schoo, hoofddirecteur van de Nederlandse Standard Electric Mij en H. Fürstner, voorzitter van de directie van N.V. Philips Telecommunicatie, 18 februari 1955; verschillende nota's, verslagen en brieven. 'Nationaal Archief, Den Haag, Ministerie Verkeer en Waterstaat (NL-HANA), Staatsbedrijf der PTT 1955–1988, inventarisnummer 4103'; Th.J. Burgerhout, 'De ZEBRA. Deel II: motivering van de keuze', Technisch rapport NLL-TM F. 180 (Nationaal luchtvaartlaboratorium februari 1956), 4

⁷³Verslagen van de vergaderingen van de laboratoriumbestuursraad van 25 mei 1955 en 9 september 1955. 'Nationaal Archief, Den Haag, Ministerie Verkeer en Waterstaat (NL-HANA), Staatsbedrijf der PTT 1955–1988, inventarisnummer 9994'; Van der Poel, *Een leven met computers*, 20

⁷⁴'NL-HANA, Staatsbedrijf PTT, 1955-1988, inv.nr. 4103'

⁷⁵Poel2007a

⁷⁶9 maart 1955, Ibidem

⁷⁷Van der Poel, 'De ontwikkeling van de computer in Nederland.', 6

bereid om voor de PTT de nieuwe computer bij PTI te monteren, natuurlijk met componenten van Philips.⁷⁸ PTT zou dan alsnog als computerfabrikant moeten optreden en dat was juist wat PTT wilde vermijden.

Het contact met de NSEM verliep voorspoediger. De NSEM was geïnteresseerd, net zoals de moedermaatschappij in België, Bell Telephone Manufacturing Company, en het zusterbedrijf in Engeland, Standard Telephones and Cables Ltd. (STC). Beide bedrijven hadden al ervaring opgedaan met de bouw van grote rekenmachines. Bell met de bouw van “La Machine Mathématique” en STC was bezig geweest met het zelf ontwikkelen van een computer, overigens zonder succes.⁷⁹ Er werd besloten tot een bijeenkomst tussen PTT, NSEM en STC; besproken werd de mogelijkheid om tot een overeenstemming te komen over het produceren van computers door STC naar ontwerp van PTT en het aanschaffen van een aantal van die computers door PTT.⁸⁰

Technische details werden besproken in aanwezigheid van Kosten en Van der Poel. STC was gecharmeerd van Van der Poels ideeën. Overeenstemming op technisch gebied was dan ook geen groot probleem, te meer daar Van der Poels logische ontwerp nog helemaal uitgewerkt moest worden⁸¹ en PTT gaf STC daarin de vrije hand.⁸²

Ondertussen gingen de onderhandelingen ook op juridisch en finantiëel terrein verder. Een punt van bespreking waren patenten: bij wie zouden die komen te liggen? Dit punt was des te meer nijpend omdat Van der Poel in 1956 op zijn ontwerp van de ZEBRA (Zeer Eenvoudig Binair RekenApparaat), zoals de computer was gaan heten, wilde gaan promoveren. Besloten werd dat STC de rechten buiten Nederland zou verkrijgen en PTT die in Nederland. PTT kon de rechten in Nederland namelijk niet weggeven in geval er vanuit de overheid besloten zou worden tot de oprichting van een rekenmachinemaatschappij.⁸³ PTT en STC dienden samen een patentaanvraag in op 23 december 1955; als uitvinders werden aangemerkt W.L. van der Poel, H.H. Clement en J. Rice.⁸⁴

Verder zou STC samen met de NSEM de ZEBRA uitwerken, produceren en de marketing ervan verzorgen. STC nam een licentie af van PTT van 2% van de verkopen van de ZEBRA. STC kreeg het recht de naam van de computer aan te passen om hun aandeel in de ZEBRA zichtbaar te maken. Op 7 december 1955 werden de onderhandelingen afgesloten met het tekenen van een overeenkomst tussen PTT, STC en NSEM⁸⁵: de Stantec ZEBRA was een feit.

⁷⁸Verslag van de vergadering van de laboratoriumbestuursraad van 25 mei 1955. ‘NL-HANA, Staatsbedrijf PTT, 1955-1988, inv.nr. 9994’

⁷⁹Brieven van de NSEM aan de directeur-generaal van de PTT, 10 maart 1955 en 18 maart 1955. ‘NL-HANA, Staatsbedrijf PTT, 1955-1988, inv.nr. 4103’; R.J. Ord-Smith, ‘Memories of forty years with Computers’, in: C. Pronk en W.J. Toetenel ed., *Vooruitgang, bit voor bit* (Delft: Delftse Universitaire Pers 1988), 30–34, aldaar 31–32

⁸⁰Verslag van de bijeenkomst tussen PTT, NSEM en STC, 24 en 25 mei 1955. ‘NL-HANA, Staatsbedrijf PTT, 1955-1988, inv.nr. 4103’

⁸¹Er waren al gedetailleerde schema’s uitgewerkt door Van der Poel toen er nog van uit gegaan werd dat PTT de computer zelf zou gaan bouwen.

⁸²New Dutch P.T.T. computer, technical notes 17th June 1955. Ibidem

⁸³Verslag van een bespreking tussen STC, NSEM en PTT op 12 augustus 1955. Ibidem

⁸⁴W.L. van der Poel, H.H. Clement en J. Rice, ‘Patent Specification GB802,745: Improvements in or relating to Electrical Digital Computers’ (1958), (URL:<http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=GB802745&F=0>)

⁸⁵Heads of agreement dated the 7th day of December 1955 Between Netherlands Postal and Telecommunication Services (PTT), Standard Telephones and Cables Limited (STC) and Nederlandsche Standard Electric Maatschappij N.V. (NSEM) Relating to Computers. ‘NL-HANA, Staatsbedrijf PTT, 1955-1988, inv.nr. 4103’

<i>Naam</i>	<i>jaar</i>
DNL	1958
NLL	1958
TH Delft	1958
Marine Amsterdam ²	????
Rijksuniversiteit Groningen	1959
Rijksuniversiteit Utrecht	1960
TNO	1960
Heemaf ⁰	1960
Meetkundige Dienst Rijkswaterstaat ¹	196?

0: De Heemaf uit Hengelo was een machine-industrie. Het was samen met de Marine de enige niet wetenschappelijke instelling die een Stantec ZEBRA in gebruik nam.

1: De meetkundige dienst in Delft had samen met het International Training Centre for Aerial Survey één ZEBRA.

2: Over de computer van de Marine is niets bekend, dat was geheim.

Tabel 1: Instelling en jaar van ingebruikname van de negen Stantec ZEBRA computers in Nederland.

In april 1958 kon het Nationaal Luchtvaartlaboratorium, alhoewel iets vertraagd, de eerste Stantec ZEBRA in Nederland in gebruik nemen.⁸⁶ Al tijdens de onderhandelingen groeide het aantal orders. Niet alleen van Nederlandse instellingen, maar ook uit het buitenland.⁸⁷ Van de ZEBRA werden er ongeveer 55 verkocht⁸⁸, waarvan negen in Nederland (zie Tabel 1) en de rest verspreid over heel de wereld.⁸⁹

In 1958 nam de Mathematische Afdeling van het Dr. Neher laboratorium een Stantec ZEBRA in gebruik, waarop ‘[we] in een paar uur tijd de PTERA met een welhaast sadistisch genoeg grondig gesloopt [hebben]’.⁹⁰ De eerste 40 machines bestonden voor ongeveer de helft uit transistors en de andere helft was gerealiseerd met radiobuizen. De latere machines werden volledig getransistoriseerd geleverd.⁹¹ Omdat de verschillende ZEBRA installaties grotendeels gelijk waren konden programma’s uitgewisseld worden tussen de verschillen-

⁸⁶Th.J. Burgerhout, ‘Enige beschouwingen over de elektronische rekenmachine “ZEBRA” van het Nationaal Luchtvaartlaboratorium’, *De Ingenieur (Luchtvaarttechniek)* 71:23 (juni 1959), 37–43, aldaar 37; Van der Poel, *Een leven met computers*, 20

⁸⁷Verslag van bijeenkomst op 20 en 21 juli 1955 tussen STC en PTT. ‘NL-HANA, Staatsbedrijf PTT, 1955-1988, inv.nr. 4103’. Genoemd werden PTT, TH Delft, NLL, TNO, Philips Hilversum, het Weissman Institute uit Israël en de Universiteit Freiburg. Over die laatste werd opgemerkt: ‘Professor Fromme of this university has contributed in some small degree to the P.T.T. design and it has been agreed that he should have the right to build one machine for himself.’, pagina 3.

⁸⁸STC gaf een aantal leaflets en brochures over de Stantec ZEBRA uit: (STC) Standard Telephones & Cables Limited, ‘Stantec Zebra Electronic Digital Computer’ (1957), (URL:<http://archive.computerhistory.org/resources/text/Standard/Standard.StantecZebra.1957.102646083.pdf>), (STC) Standard Telephones & Cables Limited, ‘Stantec Zebra Electronic Digital Computer’ (1958), (URL:<http://archive.computerhistory.org/resources/text/Standard/Stantec.Zebra.1958.102646085.pdf>), (STC) Standard Telephones & Cables Limited, ‘Stantec ZEBRA Electronic Digital Computer’ (1959), (URL:<http://archive.computerhistory.org/resources/text/Standard/Standard.StantecZebraSpec.1959.102646084.pdf>) en (STC) Standard Telephones & Cables Limited, ‘Stantec ZEBRA’ (1961), (URL:<http://archive.computerhistory.org/resources/text/Standard/Stantec.Zebra.1961.102646082.pdf>).

⁸⁹Van der Poel, ‘Curriculum Vitae of Willem Louis van der Poel’

⁹⁰Van der Poel, *Een leven met computers*, 13

⁹¹Van der Poel, ‘De ontwikkeling van de computer in Nederland.’, 6

de gebruikers. Zo ontstond de ZEBRA-club.⁹² De wenselijkheid van contact tussen gebruikers werd overigens al in mei 1955 uitgesproken tijdens de eerste bespreking tussen PTT en STC.⁹³

4.2 Het logische ontwerp van de ZEBRA

Het idee van de ZEBRA was om te proberen of met de toepassing van de functionele bits zoals die in de ZERO gebruikt waren een eenvoudige machine te bouwen was die toch praktisch bruikbaar zou zijn.⁹⁴ De machine zou geen vermenigvuldiger of deler hebben en moest in staat zijn een aantal nadelen van de PTERA te overkomen: bij de PTERA kostte het een hele omwenteling van de geheugentrommel om een getal van een spoor van 32 getallen in te lezen; de parallelle opteller was complex en bevatte een groot aantal radiobuizen, een seriële opteller was veel eenvoudiger en dus betrouwbaarder; het rekenen met getallen van dubbele lengte kon in de PTERA enkel via de vermenigvuldiging; en het A register werd niet gebruikt tijdens het ophalen van een instructie. Deze vier problemen konden eenvoudig opgelost worden.

Het ontwerp van de ZEBRA was gebaseerd op een drietal punten: het besturingsgedeelte dat zowel besturing als rekenorgaan was, werd erg klein gehouden ten opzichte van het geheugen; dit besturingsgedeelte was zo eenvoudig mogelijk gemaakt, dus geen vermenigvuldiger of deler en alles in serie; het aantal elementen was geminimaliseerd om de betrouwbaarheid te vergroten.⁹⁵ In een lezing voor het Colloquium Moderne Rekenmachines verwoordde Van der Poel deze grondbeginselen van het ontwerp van de ZEBRA anders. Het moest technisch eenvoudig zijn, dus seriële werking was noodzakelijk. Het moest snel zijn en dat bereikte Van der Poel door een aantal snelle geheugenplaatsen te gebruiken. Het coderen moest erg flexibel zijn om een zo groot mogelijke vereenvoudiging te verkrijgen.⁹⁶

Het geheugen van de ZEBRA was een magnetische trommel met 8192 woorden van 33 bits, de woorden werden nu achter elkaar op een spoor geplaatst in plaats van bit voor bit zoals bij de PTERA. In het operationele deel kwamen vier registers voor, A, B, C en D. A en B waren accumulatoren van 33 bits elk waarbij A de head en B de tail was van een dubbele lengte accumulator. C was het besturingsregister van 33 bits waarin de volgende uit te voeren instructie werd opgeborgen. D was een hulpregister voor de besturing. Daarnaast waren er nog 32 korte registers in het snelle geheugen van 33 bits elk. Een aantal van deze korte registers hadden een speciale waarde, zoals constanten of speelden een rol bij de in- en uitvoer.

Een instructiewoord was opgebouwd zoals weergegeven in Figuur 4. Net zoals in de ZERO werden opdrachten gevormd door een aantal functionele bits. Het eerste bit had ook dezelfde functie: dat gaf aan of de instructie een op-

⁹²W.L. van der Poel, 'De ZEBRAclub. Causerie ter gelegenheid van het afscheid van Prof. dr. A. van der Sluis', in: *Symposium ter gelegenheid van de 60-ste verjaardag van A. van der Sluis* (Utrecht 1988), 16–17

⁹³NL-HANA, Staatsbedrijf PTT, 1955-1988, inv.nr. 4103'

⁹⁴De tekst van deze paragraaf is, tenzij anders aangegeven, gebaseerd op Van der Poel, 'The logical principles of some simple computers'.

⁹⁵Ibidem, 50

⁹⁶W.L. van der Poel, 'Logical structure of the ZEBRA', in: *Nederlands rekenmachine genootschap 1959 – 1969. Colloquium Moderne Rekenmachine II* (Amsterdam 1969), 48–53, aldaar 48

bit	naam	0	1
c_0	A-digit	jump	add
c_1	K-digit	short register for accumulator	short register for control
c_2	Q-digit		$(B) \pm \varepsilon \rightarrow B$
c_3	L-digit		shift AB left
c_4	R-digit		shift AB right
c_5	I-digit	additive	subtractive
c_6	B-digit	use A	use B
c_7	C-digit	do not clear	clear
c_8	D-digit	read	write on drum
c_9	E-digit	read	write in short store
c_{10}	V-digit	selection	test
c_{11}	V_4 -digit		V_4 : test on b_{32}
c_{12}	V_2 -digit	Un: test KSn	V_2 : test on b_0
c_{13}	V_1 -digit		V_1 : test on a_0
c_{14}	W-digit	wait	do not wait for drum
$c_{15} - c_{19}$	Short address. Number of digits < 3 .		
$c_{20} - c_{32}$	Drum address. Number of digits ≥ 3 .		

Figuur 4: Opbouw van een instructie in een woord van de ZEBRA. Uit: Van der Poel, *The logical principles of some simple computers*, 52–53.

haal instructie (X) of niet (A). De V-cijfers zijn testcijfers waarmee een operatie wel of niet uitgevoerd werd al naargelang het criterium gespecificeerd met de cijfers V. Mocht de instructie niet uitgevoerd worden, dan voerde de ZEBRA de instructie A0 uit. Het W-cijfer gaf de mogelijkheid de trommel wel of niet in de opdracht op te nemen, dat wil zeggen dat als er niet gewacht werd op de trommel er niet geschreven kon worden en er altijd nul gelezen werd. De instructie bestond naast deze vijftien functionele bits uit een kort adres en een lang adres voor respectievelijk het snelle en het langzame geheugen. De ZEBRA was dus een twee-adrescode machine.

Elke instructie begon met een A of een X. Omdat een sprong naar de volgende opdracht vaak voorkwam, elke tweede opdracht die de machine uitvoerde was zo'n opdracht, introduceerde Van der Poel daarvoor een nieuwe naam: $(p) = X_{p+1}$ werd geschreven als N. Deze opdracht N voegde de ZEBRA automatisch toe om de volgende instructie op te halen, de programmeur hoefde daar geen rekening mee te houden.

Normaal gesproken was een instructie opgebouwd met een beginsymbool A, X, of N gevolgd door een of meer adressen gemengd met functionele cijfers in een speciale volgorde: AKQLRIBCDEVV₄V₂V₁. Als de programmeur eerst een trommeladres schreef, dan zette het invoerprogramma automatisch het functie-cijfer W op 0, anders op 1. Het korte adres 0 mocht sowieso worden overgeslagen. De twee adressen werden gescheiden door functionele bits of een punt. Mocht eerst een kort adres geschreven zijn, gevolgd door een lang adres, dan was $W = 1$ en het trommeladres effectief niet meer gezien een adres.

De invoer geschiedde door standaard teletype band die door een Ferranti optische bandlezer met 200 symbolen per seconde inlas. Elk symbool was opgebouwd uit vijf gaten. Een invoerprogramma las deze vijf gaten en combineerde het tot een symbool. Hiervoor dienden de registers 26 tot en met 30. Deze

registers bevatten een nul als er geen gat was en een een als er wel een gat op de band was. Register 26 was voor het vijfde gat en register 30 voor het eerste gat, de andere gaten voor de andere registers.

Er was een teletypewriter om te printen bestuurd door een 7-unit code. De communicatie met deze printer ging via register 25. Zodra register 25 werd geselecteerd, maakte de ZEBRA het teletypesignaal gelijk aan het teken van 25 en schreef de inhoud van A naar 25. Verder kon er een hoge snelheidponser worden aangesloten die door het schrijven naar registers 26 tot en met 30 kon bandponsen.

4.3 Programmatuur voor de ZEBRA: de Simple Code

Voor de ZEBRA werd allerlei software geschreven, bijvoorbeeld LISP interpreters en ALGOL 60 vertalers. In deze paragraaf gaat het over de software die al in 1956 was ontworpen als onderdeel van de ZEBRA, dus het inzetprogramma, het adresboekprogramma en meer van dergelijke programma's. Opvallend was de toevoeging ten opzichte van de PTERA van een interpretatief programma voor de zogenaamde Simple Code. De Simple Code maakt van de ZEBRA die erg ingewikkeld was om te programmeren een erg eenvoudig te programmeren computer die meer leek op andere computers met een vaste instructieset.

Doordat de ZEBRA erg flexibel te programmeren was door het gebruik van functionele bits was de machine moeilijker te programmeren dan vergelijkbare machines.⁹⁷ Voor een ervaren programmeur was die flexibiliteit geen probleem, integendeel, hij kon zeer efficiënte programma's schrijven. Voor minder ervaren programmeurs die niet veel tijd konden of wilden investeren in het leren van de machinecode was een nieuwe krachtige code ontwikkeld: de Simple Code. Deze Simple Code gaf de programmeur het idee dat hij een machine programmeerde die de Simple Code als zijn machine code had. Het idee hierachter was dat door deze Simple Code meer mensen de ZEBRA zouden gaan gebruiken.

Een programma bestond niet enkel uit instructies die een wiskundig probleem berekenden. Er moest ook allerlei administratieve handelingen verricht worden, zoals het opmaken van de uitvoer of het bijhouden hoeveel getallen gelezen moeten worden. Daarnaast werd normaal gesproken gebruik gemaakt van getallen met een vaste komma waardoor berekeningen geschaald moesten worden om in het bereik van dergelijke getallen te blijven. Een oplossing hiervoor was het rekenen met getallen met een drijvende komma. Om deze problemen voor de programmeur op te lossen bedacht Van der Poel de Simple Code waarin bijvoorbeeld het rekenen met drijvende komma ingebouwd was.

De Simple Code machine had een apart geheugen voor getallen en een apart geheugen voor instructies die eventueel wel door elkaar gebruikt konden worden. Beide geheugens waren ongeveer 1200 plaatsen groot. Een instructie bestond uit twee gedeelten: een functiegedeelte aangegeven door een letter en het adres van de operand van de functie. De Simple Code was dus een een-adrescode, dit in tegenstelling tot de machinecode van de ZEBRA.

Verder kende de Simple Code een accumulator en een zevental kleine registers voor tellen en andere speciale doeleinden. Via de Simple code was het

⁹⁷Deze paragraaf is, tenzij anders aangegeven, gebaseerd op W.L. van der Poel, 'The simple code for ZEBRA', *Het P T T-bedrijf* 9:2 (augustus 1959), 31-66; zie ook R.J. Ord-Smith, 'The STANTEC-ZEBRA Simple Code and its Interpretation', in: R. Goodman ed., *Annual Review in Automatic Programming*, deel 1 (Londen 1960), 146-168.

ook mogelijk om de in- en uitvoer aan te sturen. De Simple Code opdrachten waren verdeeld in zes verschillende typen: aritmetische instructies die met de accumulator werkten; invoer- en uitvoerinstructies; besturingsinstructies zoals sprongen naar gelabelde instructies; invoercodes die niet tot het programma behoorden maar het Simple Code interpretatief programma machine aanwijzingen gaven hoe de invoer te lezen; speciale instructies, de zogenaamde Z-instructies waarmee allerhande standaardfuncties direct beschikbaar waren voor de programmeur; en telinstructies om repetities mee uit te voeren.

Een invoerinstructie had de vorm An.p, n een adres en p een label, waarbij de instructie n plaatsnemen na de instructie met het label p werd geplaatst. Subroutines moesten ingelezen worden voordat het programma dat ze gebruikte ingelezen kon worden, dit om de labels in de subroutine schoon te maken voor gebruik door het hoofdprogramma.

Om tussentijdse resultaten te kunnen zien, bijvoorbeeld bij het debuggen, was er de I-mogelijkheid ingebouwd. Als een I achter een instructie wordt geplaatst verving de machine die instructie met een speciale instructie die een bepaald aantal tussenresultaten uitprintte. Het drukte het adres af van de met een I gemarkeerde instructie, het printte daarna de inhoud van de accumulator uit voor en na de uitvoering van die instructie. Omdat Simple Code niet altijd voldoende was om alles te kunnen programmeren, kon de programmeur gebruik maken van de machine code van de ZEBRA.

De snelheid van een programma geschreven in Simple Code was 30 milliseconden per instructie. Het inlezen van een symbool ging in 10 milliseconden en het inlezen van een instructie of getal kostte 20 ms. De uitvoer naar ponsband kostte 20 milliseconden per symbool en 30 milliseconden per getal. De printer deed 150 milliseconden over het printen van een symbool en 30 milliseconden over het printen van een getal. Dit waren gemiddelde waarden.

4.4 Het gebruik van de ZEBRA: de ZEBRA-club, onderwaterprogrammeren en trucologie

4.4.1 Het gebruik van de ZEBRA

Zoals eerder gezegd produceerde en verkocht STC ongeveer 59 ZEBRA machines. De meeste hiervan gingen naar rekencentra van onderzoeksinstellingen en universiteiten. Van een aantal Nederlandse instellingen, zoals de Technische Hogeschool Delft, het Nationaal Luchtvaartlaboratorium en Rijksuniversiteit Groningen, zijn verslagen geschreven over het gebruik van de machine en de in gebruikname van de machine.

In 1956 schreef T.J. Burgerhout een rapport voor het Nationaal Luchtvaartlaboratorium waarin hij de keuze voor de ZEBRA machine motiveerde.⁹⁸ In dat rapport werd eerst een introductie gegeven over de automatische rekenmachines in Nederland. De conclusie was dat de ARRA en FERTA, gebouwd door het Mathematisch Centrum in Amsterdam, niet geschikt waren voor het werk van het Nationaal Luchtvaartlaboratorium.

Het Luchtvaartlaboratorium stelde namelijk een drietal eisen: de snelheid van berekeningen met drijvende komma zou zo snel moeten zijn als de snelheid van berekeningen op de ARRA met vaste komma, de machine zou een groot geheugen moeten hebben en het zou eenvoudig te programmeren moeten zijn.

⁹⁸Burgerhout, 'Motivering ZEBRA NLL'

Deze eisen kwamen voort uit de overweging dat bij automatisering geprobeerd moest worden om zo veel mogelijk werkzaamheden tot een minimum te beperken.

Op het laboratorium kwamen veel kleine berekeningen voor die in een of enkele dagen behandeld moesten worden. Daarnaast was er een aantal berekeningen van grote omvang die te ingewikkeld waren om zonder drijvende komma te programmeren. Het programmeren met een drijvende komma was des te meer belangrijk omdat dat ook voor vele systematisch verlopende berekeningen gebruikt moest kunnen worden. Omdat de mogelijkheid om met drijvende komma te rekenen zo belangrijk was, moest het interpretatieve programma dat de drijvende komma verzorgde standaard in het geheugen van de machine kunnen staan.

Een ARRA machine was dus niet geschikt, maar de werkzaamheden bij het Mathematisch Centrum werden wel in de gaten gehouden. Bij het Nationaal Luchtvaartlaboratorium was men ondertussen gewaar geworden van de ontwikkeling van de ZEBRA bij de PTT en die machine voldeed aan alle eisen. Bij het Mathematisch Centrum werd ondertussen aan de ARMAC gewerkt en na vergelijking van ARMAC en ZEBRA kwam Burgerhout tot de conclusie dat de ZEBRA de beste keus was: het was goedkoper, het was eenvoudig te programmeren via de Simple Code⁹⁹ en de machine zou betrouwbaarder zijn.

Drie jaar later, in 1959, rapporteerde Burgerhout in de Ingenieur over de ervaringen met de ZEBRA die bij het Nationaal Luchtvaartlaboratorium opgedaan waren sinds april 1958.¹⁰⁰ Hij classificeerde de ZEBRA nu als een 'binaire automatische digitale rekenmachine van de middenklasse, welke klasse gekenmerkt wordt door een vrij groot trommelgeheugen, een beperkt aantal direct bereikbare registers en reketijden, die uitgedrukt werden in milliseconden'.¹⁰¹ Hierna ging hij verder met het beschrijven van de ZEBRA, over het gebruik als zodanig zei hij echter niets.

Ook de Technische Hogeschool Delft schafte zich een ZEBRA aan die door problemen niet eind 1957 werd geleverd maar in juli 1958. Er werden twee afnemingsbeproevingen gehouden, een in Engeland en later een in Delft waarbij de ZEBRA werd getest door het 46 uur onafgebroken te laten werken op een matrix-inversie.¹⁰² Het was de bedoeling de ZEBRA met een programmeer-afdeling in te zetten voor rekenwerk van alle afdelingen van de hogeschool en eventueel ook voor rekenwerk van derden.¹⁰³

In 1960 werd door de Commissie Rekenmachines van de Technische Hogeschool Delft een rapport geschreven over rekenmachines, zowel analoog als digitaal, die door de Hogeschool gebruikt werden.¹⁰⁴ In dit rapport werd ook de ZEBRA besproken, het bestaan van een gebruikersvereniging, de ZEBRA-club, waarin kennis en programma's uitgewisseld kon worden, werd als een groot pluspunt aangemerkt.¹⁰⁵

De tijd die het leren werken met de Simple Code kostte werd volgens de

⁹⁹Burgerhout noemde het de ZEBRA code

¹⁰⁰Burgerhout, 'Enige beschouwingen over de elektronische rekenmachine "ZEBRA" van het Nationaal Luchtvaartlaboratorium'

¹⁰¹Ibidem, 38

¹⁰²L. Kosten, 'De ZEBRA-rekenautomaat van de Technische Hogeschool is in gebruik genomen', *T.H. mededelingen* 6:2 (december 1958), 38-44, aldaar 40

¹⁰³Ibidem, 44

¹⁰⁴C.J.D.M. Verhagen, *Rekenmachines in Delft* (Delft: Commissie Rekenmachines 1960)

¹⁰⁵Ibidem, 29

auteurs van dit stuk gemeten in uren en niet in dagen. Studenten konden in een programmeercursus van een aantal middagen leren werken met de machine, maar de ZEBRA werd voor dergelijke onderwijstaken maar 3% van de tijd gebruikt.¹⁰⁶ Medewerkers konden een aangepast lesprogramma volgen. De rest van de tijd ging op aan rekenwerk voor de verschillende afdelingen van de hogeschool die daarbij geholpen werden door de leiding van het Instituut voor Toegepaste Wiskunde en een tiental programmeurs. De meeste programma's, ongeveer 90%, werden in Simple Code geschreven door deze programmeurs en een enkeling die zelf zijn eigen problemen wilde programmeren.¹⁰⁷

De Rijksuniversiteit Groningen bestelde in 1957 een Stantec ZEBRA nadat Universiteit Utrecht al zo'n machine had besteld. De ZEBRA kwam onder de hoede van de faculteit Wiskunde. Het programmeren leerde ze met vier man in Utrecht. De machine werd dag en nacht gebruikt in twee ploegdiensten.¹⁰⁸

[Een aantal Stantec ZEBRA gebruikers worden besproken in het rapport: Computergebruik in Nederland.]

4.4.2 de ZEBRA-club: meester-trucologen

De ZEBRA-club was een club van gebruikers van de ZEBRA waarin gebruikers kennis, ervaring, programma's en bovenal trucks uwisselden.¹⁰⁹ Eigenlijk was de echte club in Engeland, maar ook in Nederland kwamen gebruikers uit Nederland en België eens in de twee maanden bijeen. Verschillende programma's werden uitgewisseld op bandjes; als iemand iets leuks had geprogrammeerd dan maakte hij een aantal kopieën en verdeelde die dan op een ZEBRA-clubbijeenkomst. Op zo'n bijeenkomst waren meestal een paar personen per ZEBRA installatie. In totaal ongeveer een man of twintig die trouw de bijeenkomsten bezochten. Het belangrijkste van de ZEBRA-club was het vaststellen van de LOT, de verzameling standaardprogramma's die op de Stantec ZEBRA draaiden zoals de Simple Code, standaard in- en uitvoer, en meer van dergelijke programma's.¹¹⁰

De ZEBRA werd geprogrammeerd door instructies die door 15 functionele bits werden gevormd en door hier handig gebruik van te maken was het mogelijk zeer efficiënte programma's te schrijven. 'Van elk stukje code werd tot op het bot nagepluisd of er niet één opdracht uit weggepraat kon worden. Het was onze gevestigde, maar absurde mening dat dat altijd wel zou lukken. (...) De elegantie lag erin dat de programmeur tot onverwachte denkkronkels kwam waaraan hij dikwijls zoiets als een bewijs kon ophangen.'¹¹¹ Het schrijven van dergelijke snelle programma's hing met dergelijke trucks aan elkaar en de programmeurs die hier erg goed in waren werden dan ook 'meester-trucologen' genoemd.¹¹²

Een bekende meester-trucologist was de doof-blinde G. van der Mey die grote delen van de systeemprogrammatuur voor de ZEBRA schreef, zoals de Simple

¹⁰⁶Ibidem, 32

¹⁰⁷Ibidem, 32–33

¹⁰⁸'Het begin van het Rekencentrum' (2005), (URL:<http://www.rug.nl/cit/organisatie/rc/geschiedenis/beginrc>); J. Kraak, 'Pioniers van de ZEBRA', *Pictrogram* 1 (2001), (URL:<http://www.rug.nl/cit/organisatie/pictogram/archief/2001-01/computerpioniers.htm>)

¹⁰⁹Van der Poel, 'De ZEBRAclub', 19

¹¹⁰Interview met W.L. van der Poel

¹¹¹I.S. Herschberg, 'In plaats van een afscheidstoespraak', in: C. Pronk en W.J. Toeteneel ed., *Vooruitgang, bit voor bit* (Delft: Delftse Universitaire Pers 1988), 83–86, aldaar 83

¹¹²Van der Poel, 'De ZEBRAclub', 17

Code, LISP, ALGOL 60 en LOT. Een andere was J van Leyden die ‘onleesbare maar correcte en zeer snelle programma’s [schreef] op kleine kladpapiertjes. Anderen werkten die dan uit tot distribueerbare en gedocumenteerde programma’s.’¹¹³ Volgens Van der Poel was het bedenken en gebruiken van trucs om meer complexe acties uit te voeren een wezenlijk onderdeel van het gebruik van de ZEBRA. Sterker nog, de ZEBRA zou zo goed als onbruikbaar zijn zonder deze trucs en de programmatuur die daarmee mogelijk werd gemaakt.¹¹⁴

Een andere truck was het zogenaamde onderwaterprogrammeren dat vijf jaar na het uitkomen van de ZEBRA werd ontdekt. Een onderwaterprogramma was een programma dat zijn eigen code aan kon passen. Een klein onderwaterprogramma kon leiden tot het uitvoeren van vele instructies die enkel in het snelle geheugen bestonden. ‘Alleen dank zij de aanwezigheid van een geraffineerde interpretatieve tracer was het mogelijk om dergelijke onzichtbare stukjes code te debuggen.’¹¹⁵ Er bestond ook zoets als super-onderwaterprogrammeren. Het was mogelijk zogenaamde tapeprogramma’s te schrijven die van de ponsband direct in het snelle geheugen gelezen werd en daar uitgevoerd werden zoals gewone onderwaterprogramma’s.¹¹⁶

Lijst van geraadpleegde archieven

Nationaal Archief, Den Haag, Ministerie Verkeer en Waterstaat, Staatsbedrijf der PTT 1955–1988, inventarisnummer 4103.

Nationaal Archief, Den Haag, Ministerie Verkeer en Waterstaat, Staatsbedrijf der PTT 1955–1988, inventarisnummer 9994.

Referenties

‘Nationaal Archief, Den Haag, Ministerie Verkeer en Waterstaat (NL-HANA), Staatsbedrijf der PTT 1955–1988, inventarisnummer 4103’.

‘Nationaal Archief, Den Haag, Ministerie Verkeer en Waterstaat (NL-HANA), Staatsbedrijf der PTT 1955–1988, inventarisnummer 9994’.

‘De wonderen zijn de wereld nog niet uit. Indienststelling van PTERA’, *PTT-bedrijfsbanden* 10 (oktober 1953), 295–297.

‘Het begin van het Rekencentrum’ (2005), (URL:<http://www.rug.nl/cit/organisatie/rc/geschiedenis/beginrc>).

‘Interview met W.L. van der Poel gehouden op 17 juli 2007 door G. Alberts en H.T. de’ (2007).

Alkemade, Fons, ‘OD95. 1925–2000: van Delftsch Hoogeschoolfonds naar UFD’, *Delft Integraal* 5 (2000), (URL:<http://www.delftintegraal.tudelft.nl/info/index7acb.html?hoofdstuk=Artikel&ArtID=1973>).

¹¹³Ibidem, 18

¹¹⁴W.L. van der Poel, ‘Micro-programming and trickology’, in: W. Hoffmann ed., *Digitale Informations wandler* (Braunschweig: Friedrich Vieweg und Sohn 1962), 269–311, aldaar 274

¹¹⁵Van der Poel, ‘De ZEBRAclub’, 19

¹¹⁶Ibidem, 20

- Burgerhout, Th.J., ‘De ZEBRA. Deel II: motivering van de keuze’, Technisch rapport NLL-TM F. 180 (Nationaal luchtvaartlaboratorium februari 1956).
- Burgerhout, Th.J., ‘Enige beschouwingen over de elektronische rekenmachine “ZEBRA” van het Nationaal Luchtvaartlaboratorium’, *De Ingenieur (Luchtvaarttechniek)* 71:23 (juni 1959), 37–43.
- Donselaar, P.J. van, ‘De ontwikkeling van elektronische rekenmachines in Nederland (Een historisch overzicht van Nederlandse computers)’, Technisch rapport (Amsterdam: Stichting Het Nederlands Studiecentrum voor Administratieve Automatisering en Bestuurlijke Informatieverwerking juli 1967).
- Hartree, D.R., ‘Lectures on modern calculating machines’, Technisch rapport CR-7 (Amsterdam: Mathematisch Centrum 1947).
- Herschberg, I.S., ‘In plaats van een afscheidstoespraak’, in: Pronk, C. en W.J. Toetenel ed., *Vooruitgang, bit voor bit* (Delft: Delftse Universitaire Pers 1988), 83–86.
- Kosten, L., ‘Over blokkeerings- en wachtproblemen’, Proefschrift, Technische Hogeschool Delft (Delft 1942).
- Kosten, L., ‘Electronische rekenmachines’, *Het P T T-bedrijf* 3:1 (mei 1950), 1–16.
- Kosten, L., ‘IV. Beschrijving van een machine voor kunstmatig verkeer op elektronische basis’, *De Ingenieur* 44 (Electrotechniek 7) (1951), 62–66.
- Kosten, L., ‘PTERA I. Wordingsgeschiedenis van PTERA; doel en mogelijkheden’, *Het P T T-bedrijf* 5:4 (december 1953), 116–123.
- Kosten, L., ‘PTERA IV. “A 7”’, *Het P T T-bedrijf* 5:4 (december 1953), 148–163.
- Kosten, L., ‘Octrooi No. 75090: tweewaardig electronisch lootorgaan’ (1955), (URL:<http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=GB742305&F=0&QPN=GB742305>), ingediend 18 december 1950, openbaar gemaakt op 16 november 1953, uitgegeven 15 juni 1954.
- Kosten, L., ‘De ZEBRA-rekenautomaat van de Technische Hogeschool is in gebruik genomen’, *T.H. mededelingen* 6:2 (december 1958), 38–44.
- Kraak, J., ‘Pioniers van de ZEBRA’, *Pictogram* 1 (2001), (URL:<http://www.rug.nl/cit/organisatie/pictogram/archief/2001-01/computerpioniers.htm>).
- Nieuwe Giessen, D. van de, *Onderzoek en ontwikkeling bij KPN. Een geschiedenis van de eerste honderd jaar* (Leidschendam: KPN Research 1996).
- Ord-Smith, R.J., ‘The STANTEC-ZEBRA Simple Code and its Interpretation’, in: Goodman, R. ed., *Annual Review in Automatic Programming*, deel 1 (Londen 1960), 146–168.

- Ord-Smith, R.J., ‘Memories of forty years with Computers’, in: Pronk, C. en W.J. Toetenel ed., *Vooruitgang, bit voor bit* (Delft: Delftse Universitaire Pers 1988), 30–34.
- Poel, W.L. van der, ‘Schakelingen van automatische cijfermachines’, *Nederlands Tijdschrift voor Natuurkunde* 15 (1949), 13–22, Overdruk uit Nederlands Tijdschrift voor Natuurkunde pagina’s 243–281.
- Poel, W.L. van der, ‘Het programmeren voor de PTERA’, Technisch rapport 12 MA (Centraal Laboratorium PTT 1952).
- Poel, W.L. van der, ‘A simple electronic digital computer’, *Applied Scientific Research B* 2:5 (1952), 367–400.
- Poel, W.L. van der, ‘PTERA II. De werking van PTERA’, *Het P T T-bedrijf* 5:4 (december 1953), 124–134.
- Poel, W.L. van der, ‘PTERA III. Het programmeren voor PTERA’, *Het P T T-bedrijf* 5:4 (december 1953), 135–147.
- Poel, W.L. van der, ‘Dead programmes for a magnetic drum automatic computer’, *Applied Scientific Research B* 3 (1954), 190–198.
- Poel, W.L. van der, ‘The logical principles of some simple computers’, Proefschrift, Universiteit van Amsterdam (1956).
- Poel, W.L. van der, ‘The simple code for ZEBRA’, *Het P T T-bedrijf* 9:2 (augustus 1959), 31–66.
- Poel, W.L. van der, ‘Micro-programming and trickology’, in: Hoffmann, W. ed., *Digitale Informations wandler* (Braunschweig: Friedrich Vieweg und Sohn 1962), 269–311.
- Poel, W.L. van der, ‘De PTERA’, in: *Nederlands rekenmachine genootschap 1959 – 1969. Colloquium Moderne Rekenmachine I* (Amsterdam 1969), 26 september 1953, 88–94.
- Poel, W.L. van der, ‘Floating addresses in an automatic digital computer’, in: *Nederlands rekenmachine genootschap 1959 – 1969. Colloquium Moderne Rekenmachine II* (Amsterdam 1969), 29 januari 1955, 19–27.
- Poel, W.L. van der, ‘Logical structure of the ZEBRA’, in: *Nederlands rekenmachine genootschap 1959 – 1969. Colloquium Moderne Rekenmachine II* (Amsterdam 1969), 27 februari 1956, 48–53.
- Poel, W.L. van der, ‘The Early History of Computing at PTT’, *Delft Progress Report* 5 (1980), 174–184.
- Poel, W.L. van der, ‘De ZEBRAclub. Causerie ter gelegenheid van het afscheid van Prof. dr. A. van der Sluis’, in: *Symposium ter gelegenheid van de 60-ste verjaardag van A. van der Sluis* (Utrecht 1988).
- Poel, W.L. van der, *Een leven met computers* (TUDelft 1988), afscheidsrede.

- Poel, W.L. van der, 'De ontwikkeling van de computer in Nederland.', *Histech-nicon* 15:1 (april 1989), 2–9, Samenvatting van de causerie van prof. dr. ir. van der Poel ter gelegenheid van de Nationale Wetenschapsdag in het ITC te Delft op zondag 16 oktober 1988.
- Poel, W.L. van der, 'Curriculum Vitae of Willem Louis van der Poel' (2003), (URL:<http://sepc.twi.tudelft.nl/~poel/packages/curricul.44q>).
- Poel, W.L. van der, H.H. Clement en J. Rice, 'Patent Specificati-on GB802,745: Improvements in or relating to Electrical Digital Computers' (1958), (URL:<http://v3.espacenet.com/textdoc?DB=EPDOC&IDX=GB802745&F=0>), ingediend 23 december 1955, uitgegeven 8 oktober 1958.
- Standard Telephones & Cables Limited, (STC), 'Stantec Zebra Electronic Di-gital Computer' (1957), (URL:<http://archive.computerhistory.org/resources/text/Standard/Standard.StantecZebra.1957.102646083.pdf>), brochure.
- Standard Telephones & Cables Limited, (STC), 'Stantec Zebra Electronic Di-gital Computer' (1958), (URL:<http://archive.computerhistory.org/resources/text/Standard/Stantec.Zebra.1958.102646085.pdf>), leaflet.
- Standard Telephones & Cables Limited, (STC), 'Stantec ZEBRA Electronic Digital Computer' (1959), (URL:<http://archive.computerhistory.org/resources/text/Standard/Standard.StantecZebraSpec.1959.102646084.pdf>), leaflet.
- Standard Telephones & Cables Limited, (STC), 'Stantec ZEBRA' (1961), (URL:<http://archive.computerhistory.org/resources/text/Standard/Stantec.Zebra.1961.102646082.pdf>), brochure.
- Veen, R. van der, 'Een schildpad als rekentuig. De Testudo, voorloper van de hui-dige p.c.', *Delft Integraal* 1 (2005), (URL:<http://www.tudelft.nl/live/pagina.jsp?id=339fb018-21db-4bcc-a5fe-a105ce2c8c14&lang=nl&binary=/doc/DI051-32-34.pdf>), 32–34.
- Verhagen, C.J.D.M., *Rekenmachines in Delft* (Delft: Commissie Rekenmachines 1960).