De KIM 1

D. M. de Boer

Na de 3 inleidende artikelen over de microprocessor gaan we nu een en ander in de praktijk brengen. Dit in de praktijk brengen houdt natuurlijk niet in dat we met een soldeerbout in de hand een eigen computer gaan bouwen. Ten eerste omdat een computer bouwen een tijdrovend en geestdodend werk is, met daarbij nog de kans dat het geheel niet werkt wanneer het klaar is. Ten tweede omdat er op de markt reeds uitgekiende systemen zijn, die door de grote massaproductie vrij goedkoop zijn en bovendien volledig gegarandeerd worden.
Het zelfbouwen van een computer is eigenlijk alleen aan te raden voor mensen die reeds de nodige ervaring op dit gebied hebben. Nee, RB gaat meteen beginnen met de leuke kant van de nieuwe hobby . . . het programmeren en gebruiken van de microcomputer.

Welk systeem?
Op de markt zijn reeds ontelbare systemen uitgebracht, hierdoor is het voor de leek haast onmogelijk een goede keus te maken. Sommige systemen zijn in verhouding tot hun prestaties veel te duur, en het enthousiasme van een beginnende computerhobbyist wordt al gauw de kop ingedrukt als hij merkt dat je voor een kleine duizend gulden alleen wat lampjes kunt laten knipperen.
RB heeft daarom voor u de markt bekeken. We hebben daarbij vooral gelet op de prijs/prestatie verhouding. Al snel stuitten wij op de KIM 1, een doordacht systeem voor niet al te veel geld. Voorlopig zullen wij al onze ontwerpen op deze microcomputer baseren.

Wat heeft de KIM 1 ons te bieden?
De KIM 1 is een complete microcomputer op één printplaat. Op deze print bevindt zich ook een toetsenbord en een display. Tijdens het programmeren kunnen we op dit display zien welke code op welk adres staat. Na het aansluiten van de voeding is het systeem geheel gebruiksklaar. Tot uw beschikking staat dan een ruime geheugencapaciteit van 1024 bytes (voor programma- en gegevensopslag), 15 aansluitingen welke afzonderlijk te programmeren zijn als in- of uitgang, een cassetterecorder aansluiting, een teletype aansluiting, twee interval timers en tot slot een monitor- en een audiotape programma, welke in twee ROM's staan.

Het monitor programma
Wanneer de KIM 1 wordt aangezet, zal de microprocessor geen specifieke opdrachten

zien, de inhoud van het geheugen is nu immers volkomen willekeurig. Wanneer we een toets op het toetsenbordje indrukken zal alleen de ingang waar de toets op aangesloten zit laag worden maar omdat de microprocessor niet kijkt naar die ingang zal er verder niets gebeuren. Het systeem is volkomen eigenzinnig bezig, en lijkt voor de buitenwereld ontoegankelijk. Gelukkig bezit de KIM 1 een zgn. monitorprogramma. Dit programma (dat in een van beide ROM's staat) zorgt voor de verbinding tussen het toetsenbord, display's en geheugen. De resettoets op het toetsenbord dirigeert de microprocessor naar het begin van dit programma.
Vanaf dit moment lichten de display's op en hebben we de mogelijkheid om adressen en informatie (data) in te typen. Ingetypte adressen verschijnen (hexa-decimaal) op de vier meest linkse 7-segment display's. De inhoud van dit betreffende adres kunnen we nu zien op de twee rechtse display's. Ogenschijnlijk lijkt het of we zo rechtstreeks in het geheugen kunnen kijken, en de geheugeninhoud (indien gewenst) kunnen wijzigen. In werkelijkheid loopt alles via de microprocessor, die zijn opdrachten krijgt van het monitor programma. We hebben dus het monitor programma nodig om de microcomputer te kunnen programmeren, zonder dit programma blijft het display donker en reageert de microprocessor niet op het indrukken van een toets. Wanneer we een zelfgeschreven programma starten, is de verbinding tussen geheugen, display en toetsenbord verbroken.
Alleen de stop-toets en de reset-toets veroorzaken weer een sprong terug naar het monitor programma.
We kunnen met een zelfgeschreven programma ook het toetsenbord of het display een andere functie geven. Bij de melodiant, die elders in dit nummer wordt beschreven, verschijnen bv. de namen van verschillende tonen op het display, terwijl het toetsenbord niet wordt gebruikt en ook niet reageert. Hieruit blijkt weer hoe flexibel een microprocessorsysteem is. Ogenschijnlijke verbindingen tussen schakelaars, lampjes, display's enz. kunnen zonder soldeerbout anders gemaakt worden, door eenvoudig andere opdrachten te geven.

Het audiotape programma

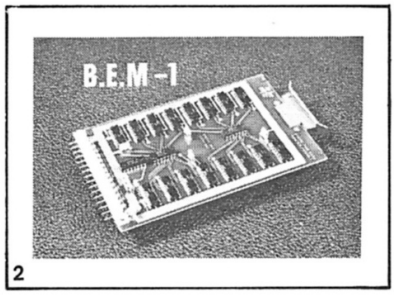
Op de print bevindt zich ook een cassette recorder aansluiting. De benodigde onderdelen om van de toontjes op de band een digitaal signaal te maken zijn aanwezig. Het decoderen van deze digitale signalen wordt weer door de microprocessor opgeknapt. Geen ingewikkelde decodeer schakeling dus, alleen een audiotape programma. Dit programma is ondergebracht in de tweede ROM, en geeft de microprocessor de nodige instructies om informatie welke op de band staat in de juiste geheugenlocaties te plaatsen. Ook bevat het audiotape programma de nodige instructies om een programma welke in de KIM 1 staat, op cassette te zetten. We moeten daarbij opgeven welk deel van het geheugen moet worden opgenomen. Of dit gedeelte nu een programma dan wel gegevens (bv. meetwaarden) bevat is niet belangrijk. Ook moet een identificatienummer worden gegeven om later de

mogelijkheid te hebben een bepaald blok te selecteren. Deze cassetterecorder aansluiting is voor de hobbyist een belangrijk pluspunt. Hij heeft nu de mogelijkheid een eigen bibliotheek van programma's aan te leggen en kan hierdoor de KIM 1 in enkele minuten omtoveren tot Melodiant, rekenmachine, amateur-schaker, wekker en wat hij maar kan bedenken.

1

De mogelijkheden van de KIM 1

Na al deze informatie zal de lezer zich afvragen: 'Wat kan die KIM 1 nu allemaal?' Het antwoord op die vraag is niet makkelijk te geven. Op de eerste plaats hangt dit natuurlijk van de programmeur af. Een feit is dat veel KIM 1-bezitters zich steeds weer verbazen over de mogelijkheden van de KIM 1. Door de grote populariteit die de KIM 1 in Amerika reeds geniet (ook in Nederland groeit z'n populariteit) zijn er al 'n groot aantal programma's speciaal voor de KIM 1 geschreven. U heeft vast wel eens gehoord van een computer die kan schaken. We denken hierbij dan aan een grote kast, vol met knipperende lampjes. Het klinkt misschien ongelooflijk, maar de KIM 1 presteert het ook! (zie afb. 1). Het is erg frustrerend om te moeten verliezen van een printplaat van nog geen 22 x 28 cm! U moet natuurlijk van deze microcomputer met z'n 1k geheugen geen schaakspel op topniveau verwachten, een redelijke schaker kan echter op een goed tegenspel rekenen.



2

Uitbreidings mogelijkheden van de KIM 1

Op de printsteker van de KIM 1 komen alle adres- data- en controllijnen naar buiten, waardoor we de mogelijkheid hebben het systeem uit te breiden. Deze uitbreiding kan bestaan uit extra programmeerbare in- en uitgangen, extra intervaltimers, vaste programma's in ROM's of een uitbreiding van het RAM. Speciaal voor de KIM 1 zijn door MOS Technology de KIM 2 . . . 6 ontwikkeld. De KIM 2 en 3 zijn RAM uitbreidingen (een RAM is een geheugen waarin zowel geschreven als gelezen kan worden) van resp. 4k en 8k. De KIM 4 is een moederkaart, die wordt gebruikt om de verschillende KIM-kaarten te koppelen. De KIM 5

is een ROM uitbreidingskaart. Op het ogenblik zijn 2 KIM ROM's te leveren, nl. een assembler/editor programma en een rekenpakket. Het assembler/editor programma werkt in combinatie met een teletype en dient om ingetypte opdrachten (zoals LDA, STA, INC enz.) te vertalen in machinecodes. In een later stadium zullen we hier nog uitvoerig op terug komen. De KIM 6 tenslotte is een EPROM-kaart, met een capaciteit van 2k bytes. We hebben het nog niet eerder gehad over de PROM en de EPROM. De PROM is een Programmable Read Only Memory, een geheugen (ROM) dat éénmaal geprogrammeerd kan worden. In dit geheugen kan dus door de microprocessor niet worden geschreven. De EPROM (Erasable PROM) is een PROM die met behulp van UV-licht weer gewist kan worden.
Ook andere fabrikanten maken geheugenkaarten die op de KIM 1 aangesloten kunnen worden. Zo is er b.v. de BEM-1, een 2k RAM-kaart (afb. 2). De BEM-geheugenkaarten zullen in de komende maanden nog uitgebreid worden met de BEM 2 . . . 5. Wij zullen u op de hoogte houden omtrent de ontwikkelingen op deze markt.

De adresseer mogelijkheden
In de vorige afleveringen over de microprocessor hebben we al gezien dat een complete instructie 1, 2 of 3 bytes in beslag neemt. Het eerste byte is de OP-code, deze geeft aan wat er moet gebeuren. Op de volgende byte(s) staat waar de operand in het geheugen te vinden is. De operand geeft aan waarmee iets moet gebeuren. Nu bestaan er verschillende mogelijkheden om aan te geven waar de operand in het geheugen staat. Deze mogelijkheden worden de adresseermogelijkheden genoemd.

De 6502 microprocessor, die het hart vormt van de KIM 1 microcomputer, heeft de volgende adresseermogelijkheden (voor een uitgebreide beschrijving zie ook RB juni 1977 pag. 216):

Immediate addressing (2 bytes)
De operand staat in het 2e byte van de instructie.

Absolute addressing (3 bytes)
Bij de absolute addressing (ook wel extended addressing genoemd) wordt een volledig 16 bits adres gespecificeerd. Op dit adres staat de operand.

Zero page addressing (2 bytes)
Bij de zero page addressing (ook wel direct addressing genoemd) geeft het tweede byte van de instructie een adres op pagina 00. Op dit adres staat de operand.

Accumulator addressing (1 byte)
Alle instructies hebben betrekking op de accumulator (register A). Een aanduiding van de operand is dus niet nodig.

Implied addressing (1 byte)
De OP-code bevat het adres van de operand. Deze instructies hebben betrekking op de andere registers in de microprocessor.

Indexed indirect addressing (2 bytes)
Bij de indexed indirect addressing (afgekort (IND,X)) wordt de inhoud van het tweede byte van de instructie opgeteld bij de inhoud van het indexregister X. Het resultaat van deze optelling geeft een adres op pagina 00. Op dit adres zijn de laatste 8 bits van het uiteindelijke adres te vinden. De volgende locatie op pagina 00 bevat de eerste 8 bits van het uiteindelijke adres. Op dit uiteindelijke adres staat de operand.

Indirect indexed addressing (2 bytes)
Bij de indirect indexed addressing (afgekort (IND),Y) bevat het tweede byte van de instructie het adres van een geheugenlocatie op pag. 00. De inhoud van deze geheugenlocatie wordt opgeteld bij de inhoud van indexregister Y. Het resultaat van deze optelling geeft de laatste 8 bits van het uiteindelijke adres. Op de volgende locatie op pag. 00 zijn de eerste 8 bits van het uiteindelijke adres te vinden. Op het uiteindelijke adres staat de operand.

Indexed zero page addressing (2 bytes)
De inhoud van het tweede byte van de instructie wordt opgeteld bij de inhoud van indexregister X of Y. Het resultaat geeft een adres op pag. 00. Op dit adres staat de operand.

Indexed absolute addressing (3 bytes)
De inhoud van het tweede en derde byte van de instructie bevat een 16 bits- basisadres. Bij dit basisadres wordt de inhoud van het indexregister X of Y opgeteld. Op het resulterende adres staat de operand.

Relative addressing (2 bytes)
De inhoud van het tweede byte van de instructie wordt opgeteld bij de inhoud van de instructieteller (program counter). Deze adresseermogelijkheid wordt uitsluitend gebruikt voor sprongopdrachten (branches) in het programma.

Absolute indirect addressing (3 bytes)
De inhoud van het 2e en 3e byte van de instructie geven een volledig gespecificeerde geheugenlocatie. Op deze geheugenlocatie staan de laatste 8 bits van het uiteindelijke adres. De eerste 8 bits staan op de volgende geheugenlocatie. Het uiteindelijke adres bevat de operand.

Wanneer u zelf gaat programmeren is het aan te raden voorlopig alleen de eerste 3

adresseer mogelijkheden toe te passen. Het blijft dan overzichtelijk, en wanneer de programma's ingewikkelder worden zal vanzelf de behoefte aan de andere adresseer mogelijkheden ontstaan. De instructies staan in tabel 1.
Voor alle instructies zijn standaard afkortingen.
Bij het schrijven van een programma worden altijd deze afkortingen gebruikt, omdat deze afkortingen ons meer zeggen dan de getallen die we voor een opdracht in moeten typen. Als het programma klaar is, gaan we de machinecodes bij de afkortingen schrijven. Dit wordt ook wel assembleren genoemd. In tabel 1 zijn deze afkortingen met hun betekenis gegeven.
In een volgend artikel zullen we verder ingaan op het zelf schrijven van een programma. Voor de geïnteresseerde lezers is het aan te raden het programma van de Melodiant (elders in dit nummer) te volgen.

Tabel 1. De instructies van de 6502 microprocessor

ADC (ADd with Carry) De operand en het carry-bit van het statusregister worden bij de inhoud van register A opgeteld.
AND (AND) De 'EN' bewerking wordt uitgevoerd tussen elk bit van de operand en register A. Het resultaat komt in register A te staan.
ASL (Arithmatic Shift Left) Alle bits van de operand schuiven één plaats naar links. Op bit 0 komt een '0' te staan, bit 7 komt in het carry-bit van het statusregister te staan.
BCC (Branch on Carry Clear) Voorwaardelijke sprong in het programma. De sprong wordt gemaakt als het carry-bit in het statusregister nul is.
BCS (Branch on Carry Set) Voorwaardelijke sprong in het programma. De sprong wordt gemaakt als het carry-bit in het statusregister '1' is.
BEQ (Branch on Result Zero) Voorwaardelijke sprong in het programma. De sprong wordt gemaakt als het zero-bit in het statusregister '1' is.
BNE (Branch on Result not Zero) Voorwaardelijke sprong in het programma. De sprong wordt gemaakt als het zero-bit in het statusregister '0' is.
BMI (Branch on result MInus) Voorwaardelijke sprong in het programma. De sprong wordt gemaakt als het negative-bit in het statusregister '1' is.
BPL (Branch on result PLus) Voorwaardelijke sprong in het programma. De sprong wordt gemaakt als het negative-bit in het statusregister '0' is.
BVC (Branch on V Clear) Voorwaardelijke sprong in het programma. De sprong wordt gemaakt als het V-bit (overflow) in het statusregister '0' is.
BVS (Branch on V Set) Voorwaardelijke sprong in het programma. De sprong wordt gemaakt als het V-bit (overflow) in het statusregister '1' is.
BIT (BIt Test) Zowel de operand als register A houden hun oorspronkelijke inhoud. De BIT-instructie veroorzaakt alleen veranderingen in de carry-, negative- en overflowbit in het statusregister.
BRK (BReaK) Bij deze instructie ontstaat er een interrupt, hij wordt alleen gebruikt bij het 'nakijken' van een programma. We kunnen dan op elke gewenste plaats het programma stoppen.
CLC (CLear Carry) Het carry-bit in het statusregister wordt '0'.
CLD (CLear Decimal mode) De optel en aftrek instructies worden binair uitgevoerd (Normale toestand).
CLI (CLear Interrupt disable bit) Interrupts worden toegestaan.
CLV (Clear V) Het overflow-bit in het statusregister wordt '0'.
CMP (CoMPare) Operand en register A worden vergeleken. Het status register wordt geactiveerd alsof de operand van register A wordt afgetrokken.
CPX (ComPare with index X). Zie CMP, alleen register A wordt indexregister X.
CPY (ComPare with index Y) Zie CMP, alleen register A wordt indexregister Y.
DEC (DECrement) Verminder de operand met 1.
DEX (DEcrement index X) Verminder indexregister X met 1.
DEY (DEcrement index Y) Verminder indexregister Y met 1.
EOR (Exclusive-OR) De excl. of bewerking wordt uitgevoerd tussen elk bit van de operand en register A. Het resultaat komt in register A te staan.

INC (INCrement) Vermeerder de operand met 1.
INX (INcrement index X) Vermeerder indexregister X met 1.
INY (INcrement index Y) Vermeerder indexregister Y met 1.
JMP (JuMP) Spring naar een nieuw adres.
JSR (Jump to Sub Routine) Spring naar een subroutine.
LDA (LoaD register A) Zet de operand in register A.
LDX (LoaD index X) Zet de operand in indexregister X.
LDY (Load index Y) Zet de operand in indexregister Y.
LSR (Logic Shift Right) Alle bits van de operand schuiven één plaats naar rechts. Bit 0 komt in het carry-bit van het statusregister te staan, in bit 7 komt een '0' te staan.
NOP (No OPeration) Er wordt geen opdracht uitgevoerd.
ORA (OR with register A) De 'OF'-bewerking wordt uitgevoerd tussen elk bit van de operand en register A. Het resultaat komt in register A te staan.
PHA (PusH register A) Zet de inhoud van register A in de stack.
PHP (PusH statusreg.) Zet het statusregister in de stack.
PLA (PuLl register A) Haal de inhoud van register A uit de stack.
PLP (PuLl statusreg.) Haal het statusregister uit de stack.
ROL (Rotate One bit Left) Alle bits van de operand schuiven één plaats naar links. Het carry-bit schuift in bit 0, bit 7 schuift in het carry-bit.
ROR (Rotate One bit Right) Alle bits van de operand schuiven één plaats naar rechts. Het carry-bit schuift in bit 7, bit 0 schuift in het carry-bit.
RTI (ReTurn from Interrupt) Ga weer verder met het hoofdprogramma.
RTS (ReTurn from Subroutine) Ga weer verder met het hoofdprogramma.
SBC (SuBtract with borrow) De operand en C worden van de inhoud van register A afgetrokken.
SEC (SEt Carry) Het carry-bit in het statusregister wordt '1'.
SED (SEd Decimal mode) De optel en aftrek instructies worden binair-decimaal uitgevoerd.
SEI (SEt Interrupt disable bit) De interrupts worden genegeerd.
STA (STore register A) Zet de inhoud van register A in de operand.
STX (STore index X) Zet de inhoud van indexregister X in de operand.
STY (STore index Y) Zet de inhoud van indexregister Y in de operand.
TAX (Transfer reg. A to index X) Zet de inhoud van register A in indexregister X.
TAY (Transfer reg. A to index Y) Zet de inhoud van register A in indexregister Y.
TXA (Transfer index X to reg. A) Zet de inhoud van indexregister X in register A.
TYA (Transfer index Y to reg. A) Zet de inhoud van indexregister Y in register A.
TSX (Transfer Stack pointer to index X) Zet de stack pointer in indexregister X.
TXS (Transfer index X to Stack pointer) Zet de inhoud van indexregister X in de stack pointer.

De KIM 1 tegen gereduceerde prijs voor abonnees van RB

De KIM 1 is normaal in de handel voor f 1110,- incl. BTW. Meegeleverd worden een stel van drie engelstalige handboeken en een printsteker. Niet inbegrepen is de voeding, waarvoor overigens ieder apparaat kan worden gebruikt dat 5V bij 1,2A èn 12V bij 0,1A kan leveren. Een schema van een eenvoudige voeding is te vinden in een van de drie handboeken.

Ten behoeve van de abonnees van RB heeft de uitgeverij De Muiderkring een partij KIM 1 microcomputers in voorraad genomen. Deze kunnen geleverd worden voor de gereduceerde stuks-prijs van f 998,- incl. BTW, excl. verzendkosten ad. f 8,15. Ook lezers die nog geen abonnee zijn kunnen van dit aanbod profiteren door zich nu als zodanig op te geven.

Bestelling van een KIM 1 en/of een abonnement op RB kan geschieden door inzending van onderstaande 'bon' of (als u het blad niet wilt beschadigen) van een briefje waarin de van toepassing zijnde gegevens zijn vermeld. Met de betaling kunt u wachten tot u een acceptgirokaart ontvangt. De bestellingen worden in volgorde van binnenkomst behandeld.

Via De Muiderkring bestelde microcomputers worden door de importeur volgens de gebruikelijke voorwaarden gegarandeerd.

BON

Uitknippen of overschrijven op ander papier en ingevuld opzenden in open, ongefrankeerde envelop aan: Redactie RB, machtiging 224, 1400VB Bussum.[ ] ik ben reeds RB-abonnee
[ ] ik geef mij op als RB-abonnee (nrs. okt., nov., dec. '77 en hele jaar '78 voor slechts f 40,-)
[ ] ik bestel een KIM 1 microcomputer voor de gereduceerde abonnee-prijs van f 998,- plus f 8,15 verzendkosten
Het totaalbedrag ad. f ..... zal ik direct na ontvangst van uw acceptgirokaart overmaken.

naam: _____

adres: _____

woonplaats: _____

handtekening: _____