



# Z-8-ontwikkelsysteem

H. J. C. Otten

Zilog, de ontwerper van de Z80-microprocessor, brengt een nieuwe familie microcomputers op één chip op de markt onder de naam Z-8.

Door Tekelec Airtronic wordt een ontwikkelsysteem geleverd rond een „Tiny Basic”-versie van de Z-8 om programma's voor de Z-8 in Basic te ontwikkelen, te testen en in EPROM te branden.

Een microcomputer op één chip bevat alle delen van een compleet microcomputersysteem: een CPU, RAM, ROM, en in/uit-schakelingen.

De Z-8 is een microcomputer volgens deze definitie. Alleen externe RAM kan nodig zijn om de Z-8 te laten draaien. Het nadeel van ROM op de chip is de bij fabricage vastgelegde software.

Zilog heeft het probleem van de vaste software in ROM voor een versie van de Z-8 opgelost door een Tiny Basic-interpretator in de interne ROM te zetten. Op deze microcomputer hoeft alleen maar een terminal en RAM te worden aangesloten om Basic-programma's te kunnen draaien. Natuurlijk is het niet de bedoeling geweest een personal computer te bouwen, maar meer een makkelijk te programmeren microcomputer voor besturingsdoeleinden met een minimum aan componenten.

De Z-8-architectuur is erg krachtig. Een minimaal systeem heeft de beschikking over 32 in/uit-lijnen. Als extern geheugen nodig is, kan

tot 60K programmeergeugen en 60K datageheugen worden aangesloten.

## Z-8-architectuur

Z-8 is de verzamelnaam voor een familie microcomputers op één chip van Zilog. Het is een 8bit-microcomputer met een 16bit-adresruimte. In afbeelding 1 is de interne architectuur van de Z-8 te zien. Er zijn twee timers aanwezig met programmeerbare instellingen. Een van de twee kan door een externe bron worden aangedreven. Een volledige UART (Universal Asynchronous Receiver/Transmitter) is ingebouwd met programmeerbare baudrate via een van de timers. Met de UART kan bijvoor-

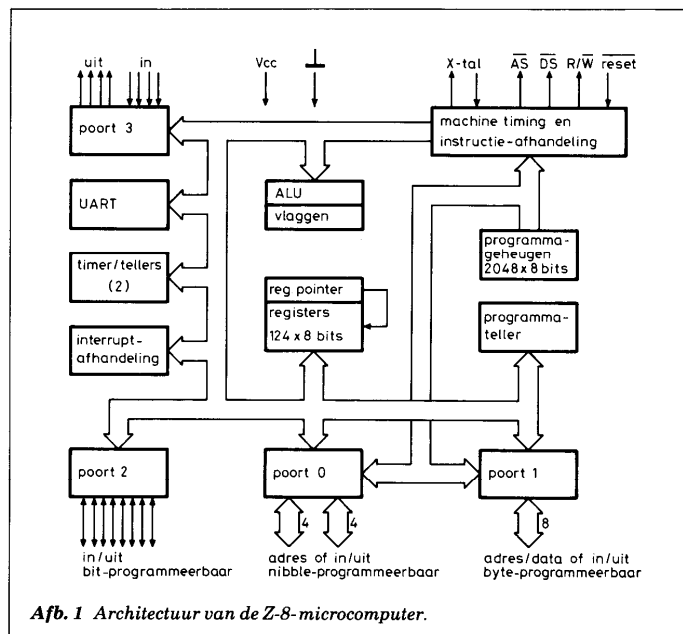
beeld een terminal worden aangesloten.

Een register file van 144 RAM-locaties bevat 124 algemene registers, vier in/uit-registers en 16 besturings- en statusregisters.

De vier in/uit-registers bepalen het gebruik van de 4 in/uit-poorten van ieder 8 bit breed.

Poort 1 kan als gecombineerde adres- en databus worden geconfigureerd. Als in/uit-poort is deze poort per byte programmeerbaar.

Poort 0 is per nibble als in/uit programmeerbaar. Als adresbus wordt het verdere geheugengebied bestreken. Poort 2 is per bit als in/uit te programmeren en poort 3 als vier ingangslijnen en vier uitgangslijnen. De Z-8 heeft 6 inter-



Afb. 1 Architectuur van de Z-8-microcomputer.

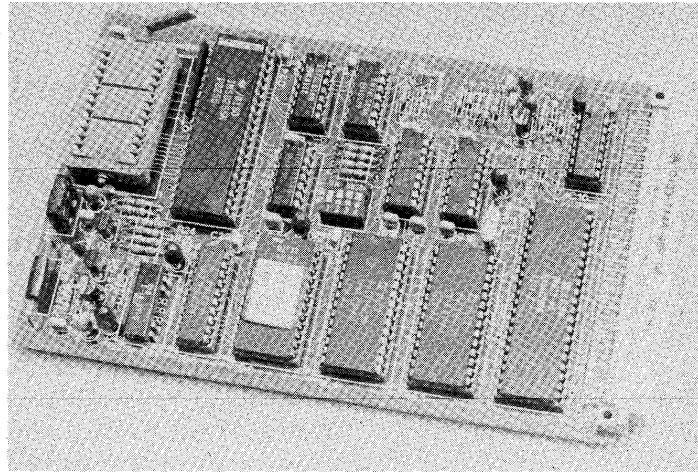


## Z-8-ontwikkelsysteem

**Afb. 2** Het Z-8-ontwikkelsysteem op een print van eurokaart-formaat.

**Afb. 3** De bij het Z-8-ontwikkelsysteem behorende voedingsprint met het cassette-interface.

**Tabel 1** Overzicht van Z-8-Tiny Basic.



ruptniveaus, waarvan er twee intern zijn. De Z-8 is een snelle microcomputer. De meeste instructies worden in 1,5 tot 2,5 microseconde uitgevoerd. De Z-8 maakt gebruik van zogenoemde pipelining, waarmee de volgende instructie alvast wordt opgehaald om tijd te winnen. Onderdeel van de Z-8 chip is 2K ROM; 4K-versies zijn ook leverbaar. Deze ROM is bij de fabricage al gevuld en niet meer te veranderen. De Z-8 is dus alleen geschikt voor grote productiehoeveelheden of moet, zoals de Z8671-versie, met een Basic-interpret in ROM toch weer kunnen worden geprogrammeerd met extern programmagesheugen.

### Instructieset Z-8

Een eerste indruk van de Z-8 is dat de instructieset erg op die van de Z80 lijkt. Meer registers en optimalisatie voor besturingsdoeleinden geven natuurlijk wel verschillen.

De Z-8 heeft 47 instructies en 9 adresseringsmogelijkheden. Vertegenwoordigd zijn de gebruikelijke groepen zoals verplaatsings, arithmetische, logische, schuif-, rotatie- en bit-manipulatie-instructies, alsmede instructies voor de besturing van het programma en de CPU.

De bloktransportinstructies lijken op de overeenkomstige Z80-instructies.

De scheiding tussen in/uit-poorten en geheugen is vervangen door een scheiding tussen interne register file en extern geheugen of memory-mapped in/uit.

### Z6132 quasi-statische RAM

De Z6132 is een quasi-statische RAM, georganiseerd als een 4096 woorden bij 8 bits geheugen. Al-

hoewel de interne celstructuur dynamisch is, worden alle refresh-handelingen intern uitgevoerd en ziet de RAM er voor de buitenwereld uit als een statische RAM. De Z6132 is pen-overeenkomstig met

de 2716 of 2732 EPROM.

De Z6132 kan zonder andere IC's direct op de Z-8 worden aangesloten om 4K RAM werkruimte toe te voegen. Toepassen van de Z6132 is niet beperkt tot de Z-8.

**Tabel 1**

Begrip	Waarden	Omschrijving
constanten	-32768 tot 32767 0000 tot FFFF	decimaal (twee bytes) hexadecimaal
variabelen	A tot Z	(twee byte lang)
operatoren	+ - * / \	optellen aftrekken vermenigvuldigen delen absolute deling en naar rechts schuiven
	= => > <= < <>	gelijk aan groter of gelijk groter kleiner of gelijk kleiner ongelijk aan
geheugenreferenties	@ -	byte referentie woord (twee byte) referentie
functies	AND USER	Boolean AND sprong naar machinetaal routine met parameter en argument als resultaat
statements	GOTO GO @ GOSUB IF...THEN INPUT/IN LET LIST NEW PRINT RETURN RUN STOP	sprong naar regelnummer sprong naar machinetaalroutine sprong naar subroutine conditionele sprong invoer toekenning waarde aan variabele lijst van programma nieuw programma uitvoer (evt, in hex) terugkeer uit subroutine start programma einde programma



### Z-8-Basic/debug

De versie van de Z-8 met in de 2K ROM een Tiny Basic-interpretter heeft het nummer Z8671 gekregen. Tiny Basic is een kleine subset van standaard Basic met uitbreidingen en optimalisaties voor het schrijven van besturingsprogramma's. Direct manipuleren van geheugen en registers van de Z-8 is eenvoudig. Externe interrupts kunnen ook worden verwerkt. Optimalisaties zijn ondermeer het kunnen werken met hexadecimale of decimale getallen; getallen zijn alleen integers van twee bytes. Machinetaalsubroutines zijn mogelijk. De Tiny Basic-programma's kunnen in ROM worden gezet. Een minimaal besturingssysteem kan daarom bestaan uit een Z-8 en een ROM.

De positie van iedere variabele en pointer is gedocumenteerd, zodat makkelijk machinetaalprogramma's met Tiny Basic kunnen samenwerken.

Als de Z-8 opstart wordt eerst gekeken of er RAM op het systeem is aangesloten. Stack en andere pointers worden bij externe RAM daarnaar gezet, anders wordt naar de interne registers verwezen als werkruimte. Vervolgens gaat de Z-8 kijken of er ROM aanwezig is met een geldig Tiny Basic-programma, vanaf adres \$1020.

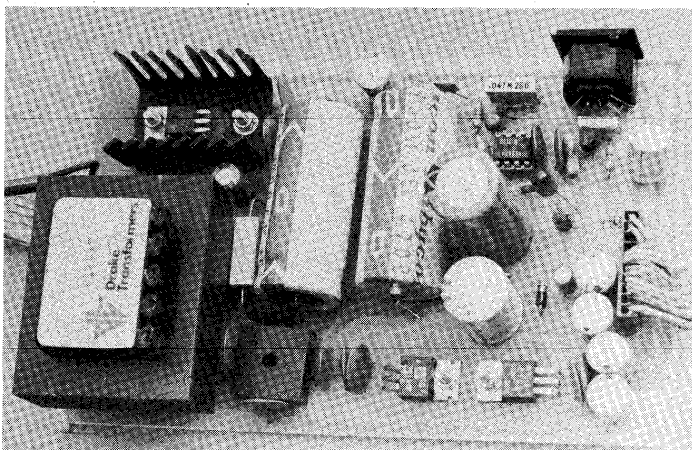
Als dit niet het geval is meldt Tiny Basic zich aan de terminal, aangesloten op de Z-8-seriële poort, klaar om in Basic te worden geprogrammeerd. Bij een programma in ROM wordt dat programma opgestart. De baudrate van de terminal wordt bij het opstarten ingelezen van locatie \$FFFD, bijvoorbeeld van een dip-schakelaar. Op deze manier is een zo flexibel mogelijke toepassing ontstaan. Een alleenstaande besturingseenheid met een programma in ROM of een ontwikkelsysteem met RAM vereist geen andere componenten.

Een overzicht van Tiny Basic is in tabel 1 te vinden.

### Z-8-ontwikkelsysteem

Het Z-8-ontwikkelsysteem is op een eurokaart-print geplaatst met connectoren voor voeding en eventuele uitbreidingen.

Centraal staat natuurlijk een Z8671, de Tiny Basic-versie van de



Z-8. 8K RAM gerealiseerd met twee Z6132 quasi-statische RAM-IC's, een 2732 EPROM voor 4K ROM, RS232-interface-elektronica voor de seriële poort van de Z-8 met een dip-schakelaar voor de baudrate, een 8255 in/uit-IC met bijbehorende elektronica en een IC-voet voor een EPROM-programmer zijn daarnaast op de kaart gezet. Een bijbehorende voedingseenheid, ook op een print geplaatst ter grootte van een eurokaart, levert alle benodigde spanningen, inclusief de 25 V voor de EPROM-programmer en de + en - 12 V voor het RS232-interface.

Aansluiten van een terminal is voldoende om met het Z-8-ontwikkelsysteem te werken in Basic.

De 4K ROM bevat een aantal Tiny Basic-programma's.

1. Data in EPROM zetten via de EPROM-programmer.
2. Data uit de te programmeren EPROM lezen.
3. Data in het geheugen verplaatsen.
4. Hex- en ASCII-dump van het geheugen.
5. Inhoud van het geheugen bekijken en veranderen.
6. Cassette lees- en schrijfroutines, in Z-8-machinetaal geschreven.

Dit minimale systeem is voldoende om Tiny Basic-programma's te ontwikkelen, te testen en daarna in EPROM te zetten voor een toepassing als besturingsapparaat.

In een assembler is niet voorzien bij het Z-8-ontwikkelsysteem. Zi-

log beschikt wel over een goede assembler. In de onlangs uitgegeven Computer Bulletin special is een artikel opgenomen met voorbeelden van deze gestructureerde assembler.

### Conclusie

Een grote computer, al is het een goedkope microcomputer is te duur als eenvoudig besturingsapparaat. Het is niet moeilijk goedkope randapparatuur aan te sluiten, maar dat grote systeem heeft de beschikking nodig over software-hulpmiddelen en hardware-interface-faciliteiten en is daarom nogal duur.

De Z-8-Basic-microcomputer met een interpretatieve taal heeft geen behoefte aan teksteditors, assemblers etc., die veel geheugen eisen en is toch in staat comfortabel en snel de ontwikkeling van eenvoudige en goedkope besturingsapparatuur mogelijk te maken.

Z-8-Tiny Basic is een Basic-interpretter zonder pretenties, maar wel zeer geschikt voor juist die besturingsprogramma's waarbij beheersing van de microcomputer voorop staat. Het Z-8-ontwikkelsysteem zoals dat door Tekelec Airtronic op de markt wordt gebracht is een goed hulpmiddel voor het ontwikkelen van Z-8-Basic-programma's. Een getest programma kan dan met het Z-8-ontwikkelsysteem in een EPROM worden gezet en in een minimaal (3 à 4 IC's) Z-8-besturingssysteem worden gedraaid.