



SAMSON EN SATELLITE

GETEST

H. J. C. OTTEN

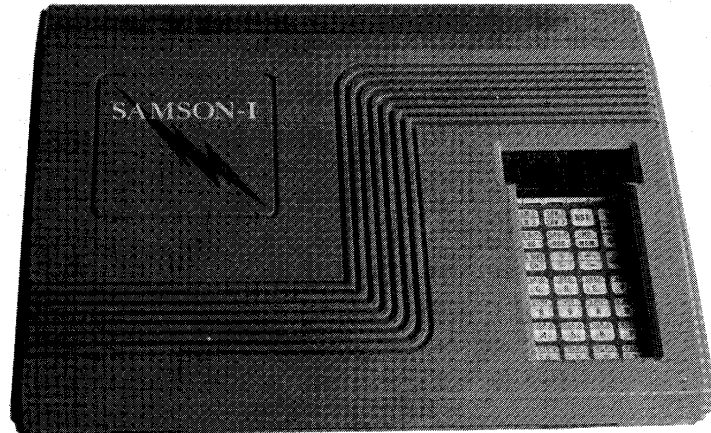
Achter de namen Samson en Satellite gaan twee oude bekenden schuil. In de Samson behuizing zit een SYM-1 singleboard computer en in de Satellite een KTM-2 videoterminal.

Deze apparaten zijn al apart besproken in Radio Bulletin maar de combinatie van beide, in een nette behuizing, aangevuld met geavanceerde software zoals de SYM supermon monitor, de RAE resident assembler en editor en de Microsoft Basic levert een zeer fraaie microcomputer.

Hardware

Omdat de SYM-1 al besproken is in Radio Bulletin (december 1978) zullen we ons als geheugensteun beperken tot een opsomming van de hardware mogelijkheden. Dit geldt voor zowel de kale SYM-1 als de SAMSON-1. Hierbij gaan we uit van een geheel uitgebreid systeem.

- 6502 microprocessor
- 6532 RAM, timer, in/uit-IC
- 6522 VIA, 2 timers, schuifregister, in/uit (3 stuks)
- toetsenbord met 28 toetsen
- 6 cijferig LED-display
- audio cassette interface met motorbesturing
- teletype 20 mA interface
- RS232 terminal interface
- oscilloscoop interface
- 4K RAM (uit te breiden tot max. 32K)
- 4K ROM met monitor supermon



- 8K ROM met RAE assembler/editor
 - 8K ROM met Basic interpreter
- De in maart 1980 in Radio Bulletin besproken KTM-2 heeft de volgende mogelijkheden:
- 24 regels met 40 karakters video
 - versie met 80 karakters ook beschikbaar
 - volledige ASCII upper en lowercase karakterset
 - 128 grafische karakters
 - RS232 interface
 - extra printer RS232 interface
 - instelbaar dataformaat
 - relatieve en absolute cursor adressering
 - diverse escape sequence mogelijkheden
 - reverse video

Samson- en Satellite-software

De SYM en de KTM-2 worden zonder behuizing geleverd, en zonder voeding. Bij de Samson en Satellite zijn deze kale printplaten voorzien van een nette kunststof behuizing. Deze behuizingen zijn overigens ook los te verkrijgen. Met deze bij elkaar passende behuizingen wordt het probleem dat de kale SYM en KTM-2 leveren keurig opgelost.

Monitor

De Supermon monitor van de SYM-1 bevat de noodzakelijke basis software voor de SYM zoals de reset initialisatie en diverse in/uit-drivers voor teletype, RS232 terminal en het toetsenbord plus LED-display. Daarnaast is een uitgebreide machinetaal monitor beschikbaar met een commando interpreter.

Een opvallende en fraaie eigenschap van de SYM-software is de scheiding tussen de in/uit van een programma en het in/uit-device. Daartoe staan in een stuk systeem RAM een aantal vectoren, sprongadressen. Alle in/uit verloopt via deze vectoren. Bij de reset initialisatie worden deze vectoren ingevuld met standaard adressen, maar onder software besturing kunnen deze vectoren naar wens worden ingevuld. Dit kunnen de al genoemde in/uit-drivers voor terminal of teletype of toetsenbord plus display zijn, maar ook een door de gebruiker toegevoegde in/uit-mogelijkheid.

De commando interpreter van de monitor haalt zijn input van de inputvector en de output loopt via de outputvector. Hetzelfde commando kan zo via elk in/uit-device worden ingevoerd. Alle SYM-software zoals Basic en assem-



bler RAE behandelen de in/uit op deze flexibele wijze.

Commando's voor de monitor bestaan uit een letter, gevolgd door een aantal, maximaal drie, parameters. De commando's hebben de gebruikelijke functies van een monitor om een machinetaal programma in te voeren en uit te proberen. Een fraaie mogelijkheid is het door een gebruiker zelf laten definiëren van extra commando's, ook weer gerealiseerd door een vector. De monitor is zo naar wens uit te breiden met commando's die voor een bepaalde toepassing van belang kunnen zijn.

Basic

Ook de Basic-interpretator voor de SYM is een produkt van Microsoft. De mogelijkheden komen ongeveer overeen met andere Microsoft Basic interpreters zoals van de Apple, de PET en de TRS-80.

Deze Basic heeft een rekennauwkeurigheid van negen cijfers.

Basic programma's kunnen onder vermelding van een letter op audio-cassette worden opgeborgen.

De USR-functie, waarmee machinetaal subroutines vanuit Basic kunnen worden opgeroepen, heeft naast de gebruikelijke functie bij de SYM nog een tweede gebruiksmogelijkheid. Als meer dan één argument wordt opgegeven, is het eerste argument het adres van de subroutine en daarop volgende argumenten worden via de stack aan de subroutine meegegeven. Handig is dat deze SYM-Basic ook hex getallen accepteert, vooral bij de Peek, Poke en Usr statements is het gebruik van hexadecimale getallen duidelijker dan decimale.

RAE assembler

De RAE (Resident Assembler en Editor) maakt van de SYM of Samson een comfortabel ontwikkelsysteem voor machinetaalprogramma's. Nu is een editor/assembler programma voor de 6502 niet iets bijzonders, voor elke 6502 computer is dit wel beschikbaar. De RAE-assembler is echter wel iets bijzonders. Dat bijzondere zit voornamelijk in de assembler die macro's, conditional assembly en een relocating loader faciliteiten biedt. Om het belang van deze faciliteiten te kunnen doorzien is het nodig wat meer over een assembler te vertellen.

De instructies die de microprocessor begrijpt zijn binaire woorden, een stel



afb. 1. Satellite video terminal.
afb. 2. De Samson van binnen.

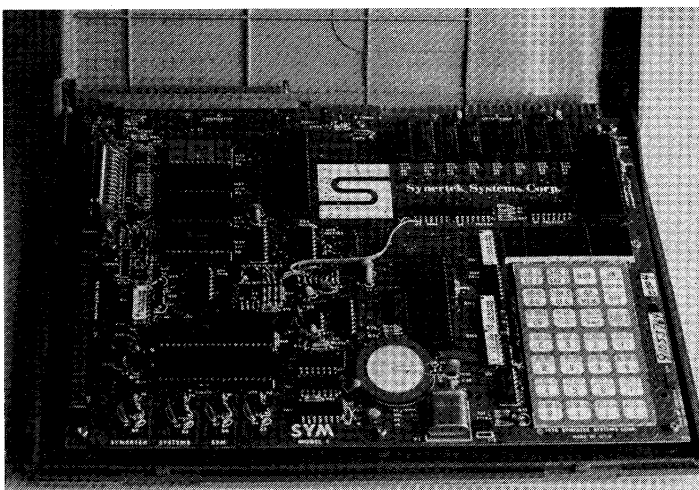
nullen en enen. Voor een mens is dat zo goed als onbegrijpelijk, zodat om de communicatie tussen mens en machine mogelijk te maken, de programmeertalen zijn ontwikkeld. Hoe 'hogere' de programmeertaal, des te beter is het programma door een mens te begrijpen en te onderhouden. Als een systeem te klein is om zo'n hogere programmeertaal te verwerken moet het programma voor de computer uit machine instructies bestaan.

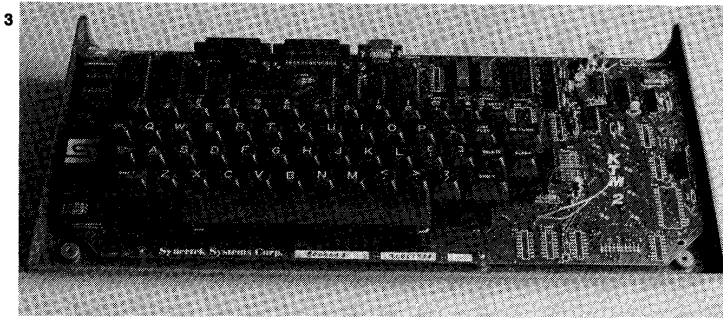
Om zo'n machinetaal programma te schrijven is ook een programmeertaal ontwikkeld: de assembler. De assembler accepteert de door de mens ingevoerde programma code: de source code. En genereert hieruit de binaire instructies. De assembler programmeertaal staat natuurlijk zeer dicht bij de machine, de meeste statements zijn machine-instructies, alleen aangegeven met een symbolische voor een

mens begrijpelijke naam: de mnemonic.

Daarnaast biedt de assembler de mogelijkheid data op verschillende wijze voor te stellen, zoals in hex, decimaal of ASCII. De assembler heeft ook de mogelijkheid de definitie van een bepaald adres of data uit te stellen door daarvoor een symbool: label te gebruiken. We kunnen bijvoorbeeld vooruit wijzen naar een subroutine door daar een label voor te gebruiken. Bij de subroutine wordt dat label vermeld. Bij het assembleren vervangt de assembler dit symbolische label door de assembler vastgestelde adres.

Naast de machine-instructies in mnemonics vorm kent een assembler nog meer statements, pseudo ops genoemd. Dit zijn bijvoorbeeld statements om het startadres van het programma te definiëren, externe subroutines met een label aan te duiden etc. De kwaliteit van de assembler is vaak af te lezen aan het aantal bruikbare pseudo ops, wat bij de RAE assembler een





afb. 3. De Satellite van binnen.

indrukwekkend aantal is. Een aantal van deze pseudo ops zijn bijzonder bruikbaar, dit zijn de macro en conditional assembly faciliteiten.

Een assembler levert naast de gewenste machine code ook een assembler listing die voor de programmeur van groot belang is. Mits voorzien van zinvol commentaar is de assembler listing de documentatie van het programma, nodig om kleine of grote veranderingen aan te kunnen brengen, na enige tijd de werking van het programma opnieuw te doorgronden of een ander de werking duidelijk te maken.

Macro's

Een macro is een faciliteit waarbij het mogelijk is met een regel source code een functie bestaande uit vele machine-instructies voor te stellen. De bruikbaarheid van een macro is het best met een voorbeeld duidelijk te maken. De 6502 heeft geen instructie om een twee byte teller op te hogen. We kunnen daarvoor een macro definiëren als volgt:

```
!!!INCD .MD (LOC)
        INC LOC
        BNE SKIP
        INC LOC + 1
SKIP    .ME
```

.MD is de pseudo op waarmee de macro definitie start,

.ME geeft het einde van de macro aan. LOC is een dummy variabele die pas een waarde krijgt bij de aanroep van de macro, LOC stelt de locatie voor van de twee byte teller. In een assembler programma kan nu, als we een twee byte teller willen ophogen, bijvoorbeeld op locatie Count, worden volstaan met:

```
INCD (Count)
```

waarbij de macro opgeroepen wordt en in de macro LOC door Count wordt vervangen als resultaat van de aanroep. In plaats van Count kunnen we

verder in het programma de macro opnieuw gebruiken voor een andere teller.

Een macro wordt vaak verward met een subroutine, wat niet juist is. Bij een macro aanroep kunnen we meerdere argumenten meegeven die in de macro worden ingevuld en het resultaat is een gewoon stukje code en geen subroutine.

Een macro bespaart zo een hoop schrijfwerk en maakt de source code overzichtelijker. Met een bibliotheek met macro instructies wordt de instructieset van de processor aantrekkelijk uitgebreid.

Conditional assembly

De assembler is in staat een conditie te testen en aan de hand daarvan wel of niet een stuk source code te assembleren. De testen zijn van de vorm IF EXPRESSION etc.

Door conditional assembly te gebruiken is het mogelijk met een assembler source code verschillende programma's te ontwikkelen door bepaalde stukken source wel of niet te laten assembleren.

In combinatie met een macro is conditional assembly zeer sterk. Een zeer algemeen opgestelde macro voorzien van conditionele assembly kan door de argumenten bij de aanroep maar gedeeltelijk naar wens worden geassembleerd.

Relocating loader

Als een machinetaalprogramma in de vorm van de object code door de assembler is geproduceerd, dan is dit programma voor een bepaald geheugengebied geschikt en niet zo eenvoudig te verplaatsen (relocaten). Eigenlijk moet voor een ander startadres het programma opnieuw worden geassembleerd met een ander startadres. Met een relocating loader kan dit ver-

plaatsen op een veel minder omslachtige wijze worden gerealiseerd. De assembler is in staat een relocatable object file te produceren waarin het machinetaal programma en alle informatie is opgenomen die nodig is om het programma op elke willekeurige gewenste positie neer te zetten.

Editor

De editor in de RAE assembler/editor maakt het mogelijk de source code voor de assembler op een comfortabele wijze samen te stellen, in te voeren, te wijzigen, op cassette op te bergen of vanaf te halen.

Alhoewel bedoeld om assembler source code te schrijven is het gebruik van de editor daartoe niet beperkt maar is het een uitstekende tekst editor.

Uitbreidingen

Door middel van een interface kaart is het SYM- of Samson-systeem aan het BEM-systeem te koppelen. Het BEM-systeem is gebaseerd op eurokaart formaat modules zoals RAM, EPROM en in/uit (parallel en serie) modules, met een professionele kwaliteit.

Via het BEM-systeem kunnen ook digitale cassetterecorders en floppy disc drive's worden aangesloten, waarbij ook de benodigde software leverbaar is.

Conclusie

De SYM of Samson, gecombineerd met de KTM-2 of Satellite is van een singleboard computer uitgegroeid tot een volwaardige microcomputer, vergelijkbaar met PET en Apple. Het systeem is als personal computer met de Microsoft Basic bruikbaar en als ontwikkelingssysteem met de uitstekende RAE assembler/editor en de monitor voor het debuggen.

Inlichtingen: Brutech Electronics, Vinkeveen.