



# Lilith

## Personal computer

H. J. C. Otten

De personal computer Lilith is ontworpen door N. Wirth, evenals de bekende programmeertaal Pascal. Vooral Pascal heeft in de microcomputerwereld veel succes en Wirth is een invloedrijk figuur in de informatica. De ontwerpfilosofie van Lilith is afwijkend van de normale gang van zaken. Bij Lilith is namelijk uitgegaan van de programmeertaal Modula-2 en de gehele soft- en hardware is daarop afgestemd.

Lilith is een studieproject zonder directe commerciële bedoelingen. Lilith is helaas ook niet te koop. Het belang van Lilith is het ontwerpconcept. De computer is ontworpen als een persoonlijk werkstation met eventueel netwerkkoppelingen. Het display heeft een bijzonder hoog oplossend vermogen om de gebruiker zoveel mogelijk informatie te geven. Daarmee lijkt Lilith op de Alto computer van Xerox waar meer overeenkomsten mee zijn aan te wijzen. Dat is niet

verwonderlijk omdat mensen van Xerox Palos Alto aan Lilith hebben meegewerkt.

De moderne halfgeleidertechnologie heeft het mogelijk gemaakt een geavanceerde programmeertaal, zoals Modula-2, te implementeren op een relatief kleine en goedkope machine die voor een enkele gebruiker is gereserveerd.

Het project Lilith bestond uit de volgende onderdelen. Eerst is de taal Modula-2 ontworpen en een compiler voor de PDP11. Daarmee

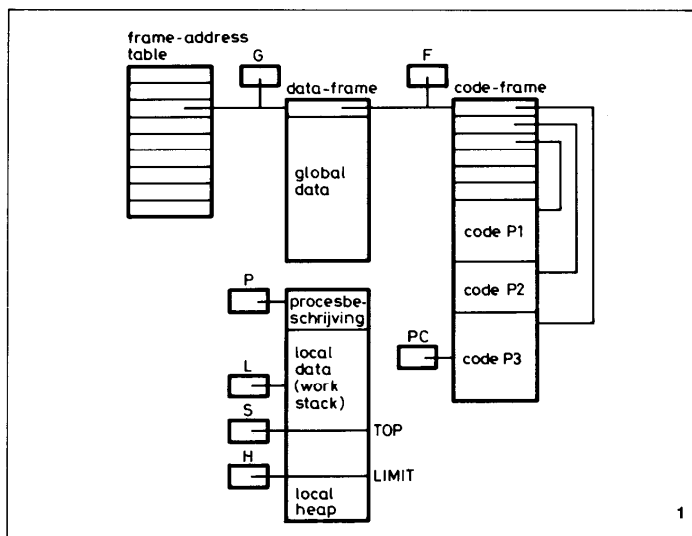
zijn een operating system met file system en linking loader, een screen editor en diverse utility's ontwikkeld.

Vervolgens is een geschikte machine-architectuur tussen compiler en hardware ontworpen en is deze architectuur in micro-code geprogrammeerd: de virtuele M-code machine. De hardware om efficiënt de micro-code te interpreteren en de randapparaten te ondersteunen was de laatste stap naar Lilith.

### Modula-2

De taal Modula-2 is ontworpen als een algemene systeemprogrammeertaal die als enige taal op een computer beschikbaar is. Dat vereist door het ontbreken van een assembler dat de taal zowel geschikt moet zijn voor machine-onafhankelijk programmeren op een hoog niveau, zoals de compiler, als voor machine-afhankelijk programmeren op laag niveau zoals het aansturen van randapparatuur. De taal Modula-2 heeft veel van Pascal geleend. Een belangrijke uitbreiding is het idee van de module. Het doel van een module is bij een modulair opgebouwd programma de samenwerking tussen de modules zo goed mogelijk te definiëren. Objecten in een module, zoals variabelen etc., zijn alleen voor andere modules bereikbaar als ze in een speciale import- of exportlijst zijn opgenomen. Deze afscherming van objecten heeft als doel de zoveel naderigheid veroorzakende gevaren van neveneffecten te vermijden. De bescherming die procedures door lokale variabelen etc. geven is niet voldoende omdat we veel objecten wel bestaand maar niet zichtbaar voor andere modules willen houden.

Een ander belangrijk concept in





Modula-2 is dat van de corountine waarmee meerdere processen in het geheugen tegelijk aan het werk kunnen zijn.

De nadruk ligt op de bescherming door modulen, typecontrole etc. die aan de programmeur en eventueel een team van programmeurs wordt geboden tegen conflicten die in assembler programma's zo vaak voorkomen.

De Modula-2 compiler doet zijn werk in vier stappen: lexical en syntaxanalyse, declaraties, typecontrole en codegeneratie.

Het is mogelijk met Modula-2 apart modulen te compileren en toch de typecontrole etc. te handhaven die door de linking loader wordt uitgevoerd.

**Operating system**

Het operating system van Lilith, MEDOS, is op een enkele gebruiker geïntendeerd en daarom vrij van beschermende mechanismen tegen foutlopende programma's van andere gebruikers. MEDOS presenteert zich aan de gebruiker als een verzameling modulen, geschreven in MODULA-2. De modulen worden in een gebruikersprogramma geïmporteerd. Een aantal modulen, zoals het file system, worden door elk programma gebruikt en zijn daarom altijd in het geheugen aanwezig.

Het operating system bestaat uit drie hoofdmodulen: Program, Terminal en File system.

De module Program beheert het geheugen en laadt en start programma's.

Terminal is de module die de software voor het toetsenbord en display bevat.

File system is, zoals de naam al zegt, de module die files beheert op disk.

Alle modulen uit het operating system zijn geschreven in Modula-2. Zij zijn daarmee voorzien van de bescherming die Modula-2 biedt en zijn zonder uitstapjes naar assembler geschreven.

**Architectuur van Lilith**

Het afstemmen op Modula-2 heeft een niet erg conventionele architectuur tot gevolg. Er is bijvoorbeeld geen poging gedaan de instructieset voor een programmeur geschikt te maken, zelfs als er een assembler beschikbaar zou zijn is

programmeren in machine-code van Lilith niet eenvoudig.

De architectuur van Lilith is die van een stack-machine en wel zo dat de variabelen, die bij de berekening lokaal zijn, snel bereikbaar zijn omdat ze frequent nodig zijn. De bijbehorende instructies moeten dan ook kort zijn. De adresseringsmethoden zijn geheel op de modulaire opbouw gebaseerd. Elke module heeft een basisadres in een frame-tabel en een eigen code- en data-frame. Lokale data bevindt zich in het lokale frame en is met een korte instructie bereikbaar (zie afb. 1).

Elke in het geheugen zijnde module komt voor in de frametabel.

**Instructieset van Lilith**

Instructies bestaan uit een of meer bytes, die in de volgende categorieën zijn onder te brengen: Load en Store, Operators, Control en andere.

In afb. 2 is het formaat van de instructies te zien afhankelijk van de adresseringsmethode.

De instructieset is direct beïnvloed door de constructies in de programmeertaal Modula-2. Het resultaat is een hoge codedichtheid. Een vergelijking met de Modula-2 compiler voor de PDP11 levert een bijna viermaal zo hoge dichtheid voor Lilith op.

De instructies Load en Store zijn push- en pop-instructies van de stack. Operators, zoals compare, laten een boolean resultaat op de stack achter. Control-instructies bepalen de volgorde van het programma, zoals jumps etc. Bij de andere instructies horen de instruc-

**Afb. 1** Architectuur van Lilith.

Registers:

- L* pointer to local data segment.
- G* pointer to global data segment.
- S* pointer to top of stack.
- H* pointer to limit of stack.
- F* pointer to current code-frame.
- PC* pointer to current instruction.
- P* pointer to current proces.

**Afb. 2** Instructieformaten van Lilith.

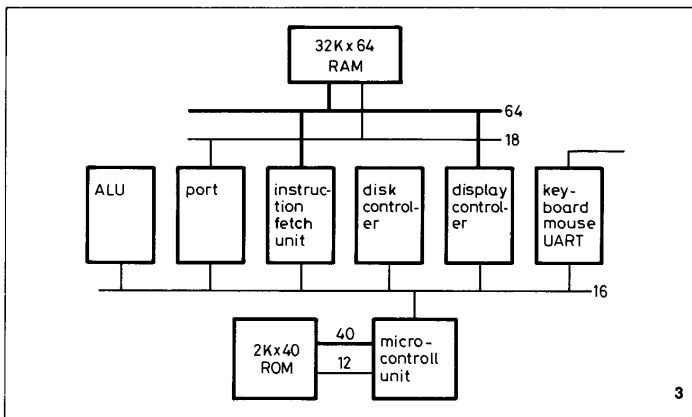
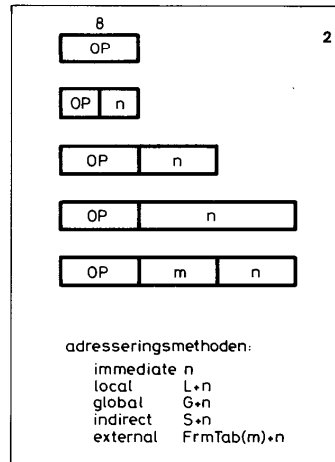
**Afb. 3** Hardwarestructuur van Lilith.

ties die de display bitmap besturen.

**Hardware van Lilith**

De volgende eisen werden aan de hardware van Lilith gesteld:

- Snelle implementatie van de M-code interpreter.
- Efficiënte instructies voor de display bitmap.



- – Eenvoudige structuur met een homogeen geheugen.
- Eenvoudig te onderhouden.

De hardwarestructuur van Lilith is in afb. 3 te zien. Het hart is de MCU (Micro Control Unit), opgebouwd met AM2911 bit-slice controllers. Samen met een  $2K \times 40$  ROM verwerkt deze eenheid de micro-code instructies. De bootstrap loader, de M-code interpreter, de instructies voor de display bitmap en de floating point routines zijn hierin opgenomen.

De ALU is opgebouwd rond een AM2911 bit-slice processor met een 16 bits brede databus.

Het geheugen is opgebouwd uit 16K dynamische RAM's van het type 4116 als een  $16K \times 64$  bits blok. Deze brede databus geeft snel toegang tot de display bitmap.

De IFU is de eenheid voor het ophalen van instructies. De communicatie met de buitenwereld verloopt via een 10Mbyte-hard disk controller, een printerpoort, een

toetsenbord en het hoog oplossend grafisch display van 594 bij 768 punten in zwart/wit.

Een belangrijk randapparaat is de Mouse, een apparaat om de cursor op het scherm te sturen met handbewegingen over de tafel. Dit gaat veel handiger dan het werken met cursorbesturingstoetsen. Ook de Alto-computer is met een Mouse uitgerust. De Mouse komt aan zijn naam doordat het een bol, goed in de hand liggend apparaatje is dat met een dun staartje aan de computer vast zit. De handbeweging over de tafel wordt (relatief) door een rollende bol in de Mouse omgezet in een cursorbeweging. De terugkoppeling tussen scherm en handbeweging geven een zeer nauwkeurige plaatsbepaling, in feite net zo goed als het oplossend vermogen van het display. De Mouse beschikt over drie druktoetsen die onder programmacontrole staan en het nut van de Mouse nog vergroten.

## Conclusie

Lilith geeft toekomstige ontwikkelingen van de personal computer aan. De personal computer heeft grote voordelen boven de gedeelde centrale computer als het gaat om beschikbaarheid en snelle respons. In Lilith zijn het display en de Mouse voorbeelden van de kracht die een personal computer heeft. Het opmerkelijke van Lilith is de wijze waarop de hard- en software zijn ontwikkeld. Het van de hogere programmeertaal uitgaan als enige en afdoende taal op een computer is een nieuw concept dat de ontwikkeling van VLSI-microcomputers sterk kan beïnvloeden.

Modula-2 is de programmeertaal waar Lilith op berust, een taal die concurrent processing, modulair opzetten van programma's en apart compileren van modulen met volledig behoud van de controle die de hogere taal op de integriteit van de data geeft, mogelijk maakt.