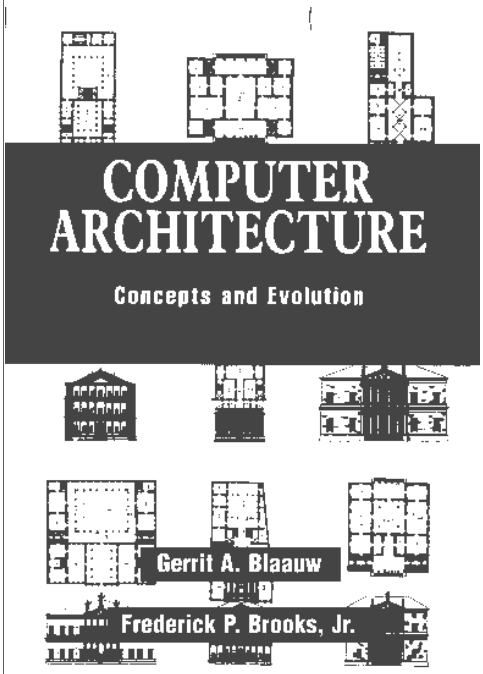
Der Ursprung des von Neumann-Computers

Computer Architecture - Concepts and Evolution; Gerrit A. Blaauw, Frederick P. Brooks jr.; Addison-Wesley; ISBN 0-201-10557-8; ca. 1200 Seiten; ATS 625,-

Norbert Bartos



11.1

Princeton IAS

11.1.1

Highlights

History

We start with the complete Princeton IAS design as published in [Burks et al., 1946], even though the IAS was not built exactly as described.

Dates. Part 1, the most often quoted paper is dated June 28/ 1946. Of *Part* II Volume 1 is dated April 1, 1947; Volume 2, April 15, 1948; and Volume 3, August 16, 1948. Volume 4 of Part II was announced in Volume 3, but apparently it was never published. Part II deals primarily with programming, but contains some changes to the Original proposal. We describe the Original design, but note the later changes.

Six years after the publication of the first IAS paper Bigelow completed the IAS Computer at the Institute of Advanced Study; and the Moore School of Electrical Engineering at the University of Pennsylvania completed the EDVAC, based on the IAS concepts.

Family tree. The IAS design inspired the designs of many other computers. We describe in this Zoo the Cambridge EDSAC (1949) and the IBM 701 (1953). Some others were the Los Alamos Maniac (1952), the University of Illinois Illiac (1952), the Ordvac (1952), the MIT Whirlwind (1953), the Rand Corporation

Johnniac (1954), the Stockholm Besk (1954), and the Amsterdam Mathematical Centre Arra (1954). The Manchester Ferranti Mark 1 (1951) and Univac I (1951), although influenced by the IAS, are nevertheless pioneering efforts of their own; we place them in the Pioneer House (Chapter 10).

Unfortunately all IAS-type machines were different, so programs were not interchangeable.

Noteworthy

Stored program. By uniting the instruction and the data store into one memory, the IAS solved the most bothersome programming problem of early Computers: addressing elements within an array. But this solution was almost immediately made obsolete by Kilburn's invention of address modification through indexing. The basic ability to treat instructions as data, however remained; it cleared the way for programming languages.

Consistency. The IAS design excels through consistency. To be sure, some of this consistency is the result of omitting necessary functions, such as logic and I/O, as well as of ignoring bit efficiency. Nevertheless, the design is

strikingly clean and thorough. Furthermore, this consistency is not accidental: Von Neumann explicitly emphasized it.

In a sense, the IAS design also exemplifies the difficulties encountered in achieving consistency: the allocation of incommensurate objects and pressures from the implementation. (See the following comments on *rounding.)*

Incommensurate units. The key architectural problem of the von Neumann design is the incommensurateness of its data and instructions - exactly the two items one would like to treat alike in the stored-program design. The IAS design solves this problem by using 40 bits for all data and 20 bits for all instructions, with addressing resolution always 40 bits. The cost of this solution is a Left and Right version of Branch and of Store Address. Most other members of the house choose to gain bit efficiency by having halfword data - at the expense of complication.

Load. A small example of clean design is the Load instruction. We might overlook it as obvious, but no design tradition had been established at this point; the EDSAC is an example of a less clean alternative: It has no Load, but combines clearing with storing.

Peculiarities

No logic. Having no logic also means having no logical shift. Logic emerges only slowly in the von Neumann house (another contrast with Kilburn's Manchester design). The arithmetic shifts are accurately called Multiply By Two and Divide By Two, deemphasizing their shifting property.

No integers. All numbers are treated as fractions; even such obvious integers as addresses are left-aligned in the instruction.

No multiple lengths. Product and dividend are single-length data - a defensible decision.

Rounding. As an outstanding numerical analyst, von Neumann was fully aware of the rounding problem, and he treated it thoroughly. The product is a rounded high-order fraction with the remainder of the product placed in the MQ.

For the quotient, however von Neumann used "making odd" to simplify the implementation. He said that "this occasionally produces results with unfamiliar and even annoying aspects (e.g., when quotients like 0 / y or y / y are formed), but it is nevertheless unobjectionable and self-consistent on

the basis of our general principles" [Burks et al., 1946].

Address capacity. The address capacity is large for its time; it could even have been four times larger by using two spare instruction bits. Interestingly enough, other members of the house have smaller word and address sizes. At that time, reliable memory-implementation was the first concern; the need for a large memory was less urgent.

No input/output. The emission of a design for the I/O mechanism is conscious; the need for I/O is mentioned in the IAS paper but thc paper leaves that problem to be solved later.

11.1.2

Machine Language

Language Level

Design philosophy. The IAS design is aimed at scientific and engineering computation. Floating point was considered, but von Neumann and his collaboraters deemed programmed scaling te be mere amenable to error analysis, and therefore better for the programmer.

The only control information is the instruction; the IAS has no indicators.

Spaces

Memory name-space. Memory is homogeneous throughout its 4096 words; there is no embedding. We assume a configuration with the full addressing capacity populated.

Working store. Working store has two identical registers called A and R, or AR; for uniformity we use acc and mq.

Control store. The instruction address is treated as an integer, with or without a fraction of one-half. The only other control state is the stopped state.

Programming model. The IAS model epitomizes the von Neumann design. Notice the inability to save the Instruction Address.

Operand Specification

Number of addresses. Instructions are uniformly one-address, with a single simple address field as the address phrase.

Operation Specification. The 22 Operations are more or less classical, with the exception of the rounding in Multiply. Only 22 of the 64 codes are used. The coding, represented by position in Program 11-5, follows the numbering in the IAS paper. The mnemonics are our own.

Instruction Structure

Instruction format. There is only one format, so only one syntactic pattern.

Der vorangehende Abschnitt wurde (gekürzt) aus dem Buch übernommen.

Nun noch einige Bemerkungen zu dessen Inhalt. Nach dem ersten physikalischen Kontakt mit dem Buch erstaunt zunächst sein Gewicht: 1,81 kg! Das zweite Aha-Erlebnis stellt sich beim Lesen ein. Die Autoren behandeln die Frage des Computerdesigns dermaßen ausführlich und komplett abdeckend, sodass es durchaus als „Handbuch des Prozessorentwurfs“ gesehen werden kann, das einfach jeder Prozessordesigner gelesen haben muss. Für Anfänger auf dem Computersektor ist es zu stark ins Detail gehend und damit eher langatmig und verwirrend. Es kann daher nicht als Lehrbuch im üblichen Sinne verwendet werden, wohl aber für Seminare und Spezialvorlesungen im Universitäts- und Fachhochschulbereich. Für den Fortgeschrittenen ist es sicher eine Fundgrube interessanter Informationen.

Ungefähr 40% des Inhalts beschäftigt sich mit den Bereichen „Machine Language“, „Names and Addresses“, „Data“, „Operations“, „Instruction Sequencing“, „Supervision“ und „Input/Output“. In jedem Kapitel werden umfassend alle Möglichkeiten des Entwicklers diskutiert und verglichen, sowie praktische Richtlinien abgeleitet.

Die verbleibenden 60% füllt der „Computer Zoo“ aus. Hier schöpfen die beiden Autoren zum Nutzen der Leserschaft aus ihrer jahrzehntelangen und reichlichen Erfahrung im Computerdesign. Wichtige Maschinen und Konzepte aus der Pionierzeit (Harvard Mark I, diverse Zuse-Maschinen, Princeton IAS u.a.) werden ebenso besprochen, wie beispielsweise einige IBM-Systeme (1401, System/360 u.a.), Cray-Computer, PDPs und VAXes und viele andere mehr, sowie auch einige bekannte Mikroprozessoren. Jede Maschine wird nach einem einheitlichen Spezifikationsschema vorgestellt, sodass ein Vergleich leicht fällt. Eine dermaßen umfassende und detaillierte Übersicht über die Computersysteme der letzten Jahrzehnte ist mir persönlich bisher nicht bekannt gewesen.

Zusammenfassend kann gesagt werden, dass dies ein Buch ist, welches man nicht so leicht aus der Hand legt (Motto: „Wenn ich nur aufhör'n könnt'...“). Es ist ausgesprochen empfehlenswert für Insider, denn ich glaube, dass jeder der Leserinnen und Leser noch viele ihr oder ihm unbekannte Informationen darin vorfinden wird.