This is the ONLY contemporary reference that I have yet been able to find of the KENBAK-1 computer:

RCA Engineer Magazine issue Feb 1974.

This is about 3 years after the KENBAK-1's debut.
Let's see what they have to say...

Design philosophy

Minimum system cost is the primary goal. To achieve this goal, an architecture is required that is both simple and flexible. Simplified computer architecture has received relatively little attention in the literature. Prior approaches toward simplified computers appear to be incompatible with microcomputer application goals.[4,5,9]

The architecture that was finally developed evolved from examining proposed applications. Another approach would have started with a more or less conventional minicomputer architecture and instruction set. This latter approach was discarded due to fundamental differences in minicomputer and microcomputer applications. It was also felt that a minicomputer starting point would not yield the simplest architecture.

Since a single-chip microcomputer promises minimum cost, the architecture was constrained to a 40-pin interface. Smaller microcomputer interfaces tend to require extensive multiplexing of interface signals which adds demultiplexing logic external to the microcomputer chip. This increases system cost.

An 8-bit parallel (or byte) architecture was chosen. This yields maximum performance consistent with interface pin constraints and is compatible with input/output requirements. One and four-bit organizations unduly restrict the range of potential applications. Sixteen or more bits exceed single-chip pin constraints or impose the need for multiplexed word transfers.

Since continued memory cost reduction is anticipated, techniques using memory to reduce hardwired logic complexity are heavily relied on. It is also apparent that many microcomputer applications will fall into the intelligent buffer category. For these reasons, direct memory addressing capability of up to 64K bytes is provided. Random-access memory (RAM) and read-only memory (ROM) can be mixed in any combination via a common memory interface. This is a distinct simplification over an architecture that provides separate RAM and ROM interfaces. The common RAM/ROM interface also enhances flexibility. System simplicity results since a single LSI chip containing both ROM and RAM segments will suffice for many applications.

Ouch! Reference 9 specifically calls out the KENBAK-1 as an example of what NOT to do in the future.

In fairness to the KENBAK-1, not even the 4004 had yet been invented (first commercially produced microprocessor).

It can be said though that this would be the only legacy of the KENBAK-1 if it had not won the 1986 "First PC" contest.

It might also be assumed that it being reference 9, while the others are numbered 4 & 5, the author had to dig pretty deep to find out about the KENBAK-1 Programming Manual and include it as just another example of a "simplified computer incompatible with application goals."

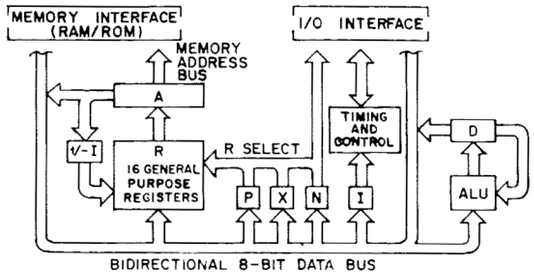

Fig. 1 — Microcomputer architecture.

While low memory costs can be expected, very low-cost systems will result only from minimizing memory capacity requirements. A unique architecture was devised which uses an 8-bit instruction format. This permits compact programs and subroutines. Useful systems requiring 1024 bytes or less of memory are possible with this format.

Random control logic uses chip area less efficiently than register/memory arrays. For this reason a simple, fixed cycle, microinstruction set was developed to reduce required control logic. The user has the option of programming directly in microcode, using a set of subroutines stored in memory (ROM/RAM), or a combination of these approaches. Sets of subroutines stored in memory are equivalent to applications-oriented macroinstructions and can readily be provided where ease of programming is important. On the other hand, many systems will utilize the microcomputer as a substitute for special purpose logic and can be programmed directly and efficiently in microcode. This approach retains most of the advantages of a microprogrammed computer but eliminates much of the specialized, hardwired sequencing and control logic usually associated with microprogrammed systems.[6] Simple, short-subroutine-calling byte sequences provide flexible macroinstruction definition.

Whether used as a component of larger systems or as a freestanding computer, the microcomputer architecture requires efficient, flexible, input/output capability. This is provided via programmed byte transfers and a built-in direct memory access (DMA) channel. Programmed input/output byte transfer instructions provide maximum flexibility for input/output selection, control, and data transfer. The DMA channel facilitates high-speed I/O block transfer, TV display refresh, and initial program loading with a minimum of external logic. While the inclusion of a DMA channel adds negligible complexity to the microcomputer architecture, it greatly simplifies system design, leading to reduced overall cost. In addition to the two basic I/O modes, four uncommitted flag lines are provided for activation by external devices. These flags can be tested as required by program. A flexible program interrupt capability also exists. Individual reset and load lines minimize external initializing logic.

The overall design philosophy consisted of developing a simple, flexible, microcomputer architecture which satisfies a wide range of potential applications at minimum cost. Performance levels were chosen to satisfy large-volume applications without overkill. The resulting architecture can be implemented initially on one or two chips using slow MOS technology.

Instruction execution times in the range of 4 to 8 μs are anticipated with LSI technologies that approach current TTL speeds. Experimental work has demonstrated that this performance level is adequate for most anticipated applications.

Microcomputer architecture

Fig. 1 illustrates the microcomputer architecture. "R" represents an array of sixteen, 16-bit general purpose registers. (This is essentially a 16x16-bit RAM.)

Registers P, X, and N are three 4-bit registers. The contents of P, X, or N select one of the 16 R registers. Register R(N) will be used to denote the specific R

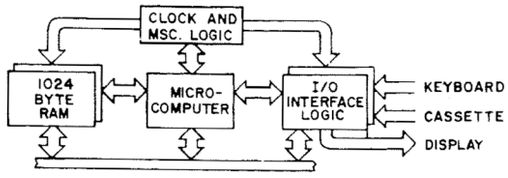

Fig. 4 — Six-chip, stand-alone system.

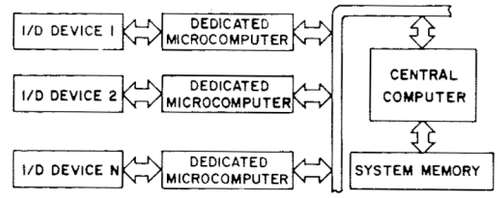

Fig. 5 — Dedicated multi-microcomputer system.

business system.

For microcode programming, an assembly language has been developed. This approach simplifies machine language coding considerably. An interactive simulator greatly facilitates initial program debugging. Both of these microcomputer software support systems are readily modified to run on existing time-sharing systems.

In general, the simplified microcomputer presents no difficulty in programming. It provides a simple set of short, easy to understand microinstructions that do not require high skill levels to use. For specific applications, tailored macroinstructions are readily provided via a flexible subroutine handling system.

Typical systems

Several systems using the microcomputer can be outlined. Many others are, of course, possible.

Fig. 3 indicates a possible microcomputer-based calculator. ROM and RAM might be provided on one chip resulting in a basic three-chip calculator. Functions could easily be added with ROM increments. This type of system could also provide a programmable calculator.

Fig. 4 illustrates a stand-alone system which might require only six LSI chips total.

It is assumed that 4x1024-bit memory chips will be available within the next several years. Subsequent LSI improvements could further reduce the chip count. Use of a small keyboard, audio cassette,[10,11] and CRT display might reduce system cost to a few hundred dollars. Such a system could have wide application in consumer and educational markets. This system, with more memory, hardcopy output, and low-cost floppy disk (or magnetic bubble bulk storage), would provide the basis for a wide range of inexpensive, turnkey, small business systems.

Fig. 5 illustrates a large computer system in which each I/O device is controlled by a dedicated microcomputer providing an intelligent buffer, as well as a replacement for special purpose logic. RAM, ROM, microcomputer, I/O device and central computer interface circuits could readily be provided on a small set of LSI chips. The microcomputer DMA feature is extremely useful for high-speed block transfers in this type of system. Downline loading of the microcomputer memory can immediately change its mode of operation. Off-line editing and maintenance is provided free. This type of large-scale system approach will become more popular in the future as microcomputer costs decrease.

The performance level of the simplified microcomputer described is more than adequate for the above types of systems as well as many others.

Conclusions

Much current microcomputer development effort appears to be directed toward improved performance. There is, however, a need for simple, minimum cost structures that will satisfy large-volume applications which do not require minicomputer performance levels. These microcomputers must also be organized to reduce total system cost. One such microcomputer architecture has been developed. It promises low cost, together with minimum external memory and system logic requirements. Hopefully, microcomputers of this class will accelerate the development of major new markets.

Currently high input/output device costs might be used as an argument against minimizing microcomputer cost. This is extremely short-sighted. The availability of ten-dollar microcomputer chips will, by itself, exert considerable pressure on the development of compatible low-cost I/O and bulk storage devices. Even now there are many potential new products that demand minimum cost microcomputers of the type described.

Because of its flexibility and potential for low-cost systems, RCA is currently developing a COS/MOS-LSI version of this microcomputer — COSMAC, SOS versions are also being investigated for applications requiring higher instruction execution rates. Both implementations are expected to find wide application in a variety of future products.

Acknowledgments

The following people have devoted considerable effort toward evaluating and developing software for the microcomputer described here: S. Heiss, A.R. Marcantonio, J.T. O'Neil, A.D. Robbi, P.M. Russo, R.O. Winder, and C.T. Wu. Without this effort, validation of the architecture would have been impossible.

References

1. Lapidus, G., "MOS/LSI Launches the Low-Cost Processor," *IEEE Spectrum* (Nov. 1972) pp. 33-40.
2. Sideris, G., "Microcomputers Muscle In," *Electronics* (March 1, 1973) pp. 63-64.
3. Hoff, M.E., Jr., "Applications for Microcomputers in Instrumentation," 1973 IEEE Intercon Papers, Session 21/1.
4. Hitt, D.C., Ottaway, G.H., and Shirk, R.W., "The Minicomputer—A New Approach to Computer Design," *Proc. of* 1968 *Fall Joint Computer Conference*, pp. 655-662.
5. Conn, R.W., "The Dinkiac I," *Proc. of* 1971 *Spring Joint Computer Conference*, pp. 1-9.
6. Reyling, G., Jr., "LSI Building Blocks for Parallel Digital Processors," 1973 IEEE Intercon Papers, Session 21/3.
7. "Backer Sought for Information Center," *Electronics* (Mar. 29, 1973) pp. 33-34.
8. "Personal Lifetime Computer Foreseen," *Electronics* (Sept. 11, 1972) pp. 40-42.
9. *KENBAK-1 Computer Programming Reference Manual*, KENBAK Corp. (April 1971).
10. "Putting Data on an Ordinary Audio Recorder," *The Electronic Engineer* (May 1972) p. DC-9.
11. Wolf, E., "Ratio Recording for Lower Cassette Recorder Cost," *Computer Design*, (Dec. 1972) p. 76.

Likely referencing the 8008 chip.

The second sentence here might have got a whiff of the incredibly influential 8080 chip that would be released in 2 months, but given the next paragraph, seems to actually refer to RCA's own 1802 chip that would soon be released.

That chip by the way, wouldn't see much usage in personal computers, but did get used in many spacecraft, such as Galileo, Magellan, & Hubble Space Telescope. :-)

This is kind of "The Point" of this RCA article (of which I omitted a few pages).

I think they don't like the KENBAK-1 because it doesn't support very many "CPU" instructions, doesn't seem to support subroutines (from my short skimming of the cited Programming Reference Manual), and some instructions may have unexpected behaviour (see PR-10).

The also-slandered "Dinkiac I", (which may have never existed), has a similarly over-reduced instruction set, proposed on pg 6 of this referenced document.