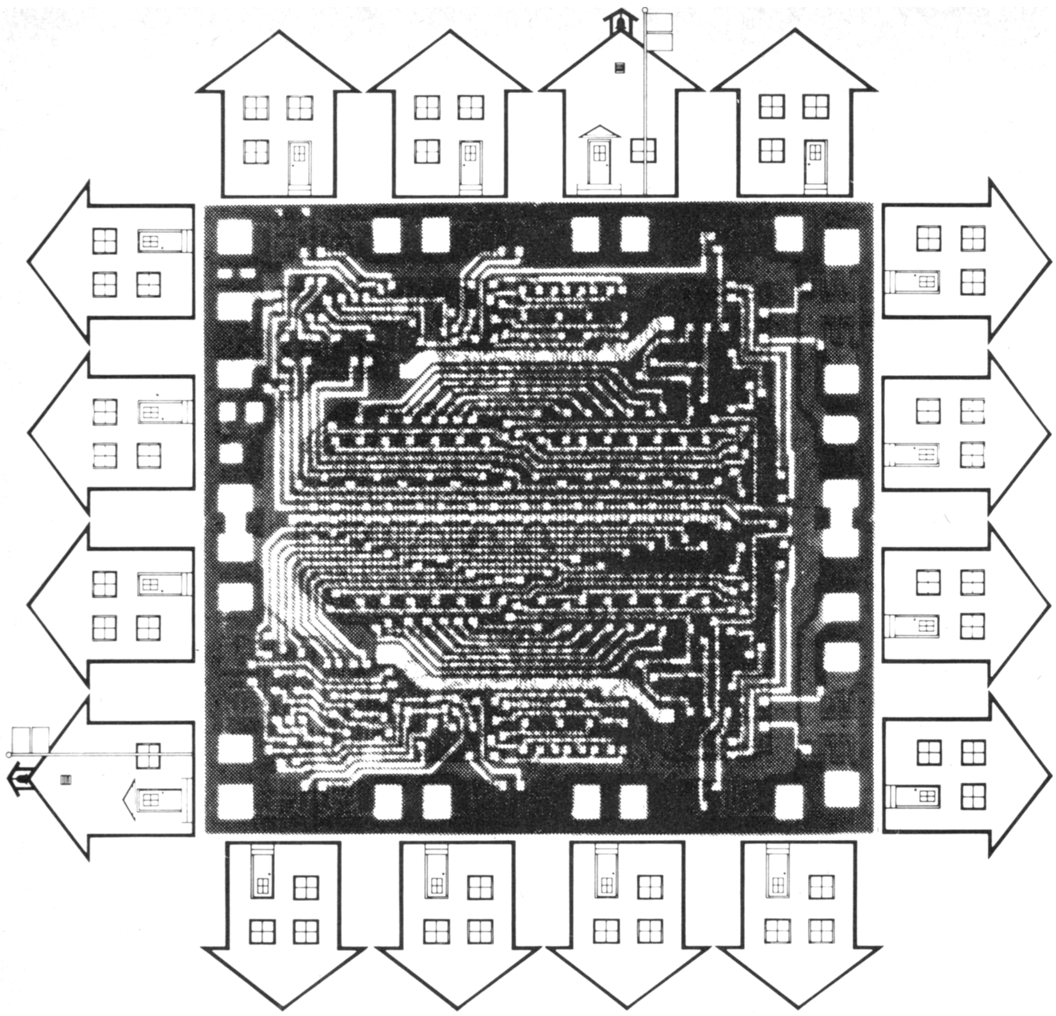


The mention of low-cost computers ususally evokes one of two images. Some of us see a super calculator; others picture a large data-base processor. The system described here is a more modest machine that could sell for under $500 in the relatively near future. Not much has been written on practical computers of this size. Nevertheless, prototypes of this low-cost, mass-market, free-standing computer system have been constructed, programmed, and operated in a home environment over the past several years.

A PRACTICAL, LOW-COST, HOME/SCHOOL MICROPROCESSOR SYSTEM

Joe Weisbecker
RCA Laboratories

Meet Fred

Despite the recreational and educational potential of stored-program computers, the single factor of cost has kept them out of the economic reach of most people. But with the advent of LSI microprocessor and memory chips, this may all change – particularly if we take a more modest applications approach and place reasonable limitations on hardware capability.

Of course, if we need conventional input, output, and bulk storage devices, or if our applications are playing chess, printing pages of data, or accessing large on-line digital/video data bases, the cost will still be prohibitive. However, prototypes of such modest home/school systems have been constructed and in use for several years.

This system, called FRED (Flexible Recreational and Educational Device), has been developed using the RCA COSMAC microprocessor.

A computer of this type could have major social value. As an interactive, open-ended, adaptive, recreational and educational device, it could stimulate the development of analytical and other intellectual abilities. One can easily imagine the formation of a whole new group of computer hobbyists, complete with user groups and publications for the exchange of programs and ideas. In short, the inexpensive home/school computer could open the door to an entirely new environment that stimulates experimentation, analysis, and creativity.

Application and System Overview

In schools, FRED could provide a powerful educational tool. It could be used to drill and test students from first grade on. It could be used in educational games, simulation exercises, and reading readiness, as well as in teaching programming, as an adjunct to math courses, and as an accessible student tool in almost any subject. FRED could be used to set up stimulating demonstrations and experiments in a wide variety of areas, to help correct learning disabilities, and to stimulate the development of creative abilities. Cost per student hour would be measured in pennies.

In the home, FRED has already functioned as a sophisticated entertainment center for the whole family. It provides a variety of games, simulates a calculator, and even provides a controllable TV puppet for the youngest member of the family. FRED permits a number of creative activities including TV picture drawing, low-fidelity music synthesis, and programming at a variety of skill levels. FRED also provides a shooting gallery, a variety of puzzles, and animated TV greeting cards for holidays.

Since FRED is a stored-program computer, it requires a program to be loaded into memory before use. Program loading is performed with an inexpensive audio cassette player which also gives the computer its voice, music, and sound effect capabilities. Prerecorded program cassettes can be loaded in less than 30 seconds.

After a program cassette is selected and loaded, FRED is operated with a small 16-position keyboard. For a game, the player presses appropriate keys to indicate the moves. Overlay cards are provided so that keyboard labeling can be changed for different programs.

FRED is attached to the antenna terminals of any TV set. This provides an inexpensive, flexible, dynamic output display which is ideally suited for home/school use. Numbers, words, or simple pictures can be displayed on the TV screen in the form of dot patterns.

The basic FRED system comprises the RCA COSMAC microprocessor, 1024 bytes of RAM, a simple hex keyboard, an inexpensive audio cassette player, and the user's own TV set. One would be hard-pressed to imagine a less expensive free-standing computer system. This system is supported by a library of cassette programs in the same way that a phonograph is supported by a record library. A continuing supply of new programs could be provided by the manufacturer of the system together with a selection of optional hardware attachments.

Adding a \$25 punched card reader and \$10 manual punch to the basic system increases its usefulness and provides more sophisticated users with the ability to prepare and save short parameter lists or programs. Adding a module for recording the contents of memory on cassettes turns the basic FRED system into a user-programmable computer for serious hobbyists. Other possible attachments include light guns, extra memory (RAM), pre-stored programs or tables (ROM), and output relays for control uses.

Design Considerations

Two different approaches can be taken toward developing an under-\$500 home/school computer. One approach involves specifying a desired set of system characteristics and then attempting to achieve cost objectives. The danger in this approach is the tendency to overspecify the hardware so that price targets can't be met. The approach we took in developing FRED was to define a minimum cost, nontrivial hardware system that could easily meet our low price goal, and then testing its usefulness. This approach ensures that applications development effort won't be wasted. Applications developed for our minimum system are easily transferred to a larger system.

Any free-standing computer must include a CPU, main memory, input and output device(s), and bulk storage device(s).

The FRED system required the development of a system philosophy that was consistent with utilizing minimum cost hardware. This system philosophy included asking what we could do with cheap devices instead of asking what types of devices we might ideally want. Because of our low price goal, the implications of each of our choices were magnified in importance.

For our purposes, competitive cost-performance ratios were largely ignored. Reliability was also sacrificed to some extent to achieve low cost. Of course, permanent system failures cannot be tolerated, but occasional transient errors, provided they are not catastrophic, may be permitted during program loading. The need for memory and processor parity checking was also felt to be an unaffordable luxury.

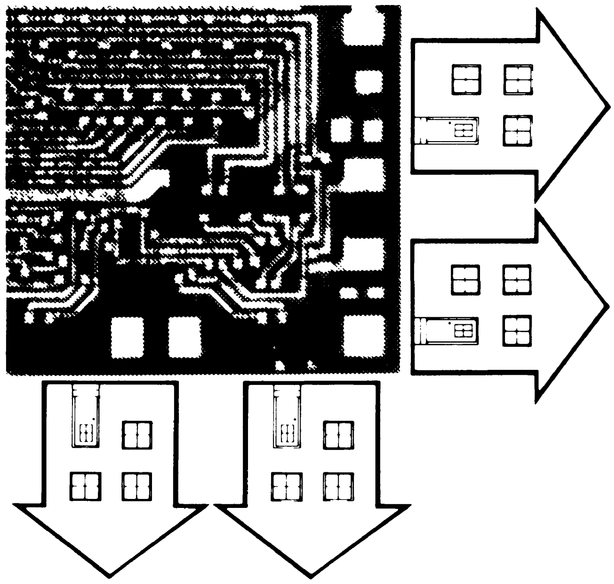
Ease of use is a primary requirement. So that the system would not appear overly complex to the novice user, a turn-key system philosophy was adopted. The user simply loads a program from a library to obtain a desired function. He is not expected to program the machine. This approach eliminates the need for program debugging and maximizes ease of use. The need for an expensive control and diagnostic panel is also eliminated. In fact, only two switches are required for basic use: LOAD and RUN. Utility routines and optional hardware are provided for the sophisticated user who wants to develop his own machine language programs. The minimum system also provides for small user-generated programs via a variety of simulation languages.

A block diagram of the basic system is shown in Figure 1. System considerations will be included in the discussion of individual elements.

Central Processing Unit and Memory

As mentioned earlier, the advent of single chip LSI microprocessors has made the under-\$500 system possible. Suitable microprocessor chips should be available for less than \$25 within the next several years. The choice of a microprocessor has a large influence on total system chip count and system cost. This influence is an important consideration since the microprocessor itself is only a small part of total system cost. The COSMAC architecture immediately eliminates the need for a read only memory (ROM) in the minimum system. Only one supply voltage is required. COS/MOS circuits further reduce system power supply costs. A self-contained direct memory access (DMA) channel facilities initial program loading and display refresh. A single-phase clock is another minimum cost feature. High output drive capability eliminates external buffer circuits.

The 8-bit COSMAC architecture is compatible with the intended uses of the system. The short instruction format permits compact programs with small memory require-



ments. Since the average user will never see the processor micro-instruction set, ease of programming is secondary to efficient memory utilization.

A complete description of the COSMAC microprocessor has appeared previously[1,2] and will not be repeated here. This architecture has demonstrated its advantages in prototypes of the low cost home/school system.

Due to the nature of our application, RAM is required for both program and data storage. It's well known that programs tend to expand to fill available memory space. Providing a 4096 byte memory only insures that no program will be written requiring a smaller memory. Even projecting a cost of 2¢ per byte would yield a cost of $82 for a 4096 byte memory. This size memory would add $200 or more to the selling price of the system. Instead of asking how much RAM we could use, we provide 1024 bytes in the minimum system. This is consistent with keeping memory cost equal to projected microprocessor chip cost. Should LSI memory costs drop below 2¢ per byte we can increase minimum system capacity to 2048 bytes or lower the price of the 1024-byte system. Based on current trends, we can safely predict one microsecond LSI RAM costs of 2 to 3¢ per byte. Dynamic RAM chips are at this cost level now, while static, single voltage RAM chips are currently available at 7 to 8¢ per byte.

The challenge of a 1024-byte memory seems to stimulate cleverness in programming and makes a future 2048-byte memory seem large by comparison. If we had initially provided a 4096-byte memory, subsequent size reduction to meet cost targets would have been extremely difficult. Assuming 4x1024 bit RAM chips are available within the next several years, a minimum system would require only two chips for memory.

Limiting the minimum system memory to 1024 bytes also provides several system cost advantages. Power supply cost is reduced, memory address drivers are eliminated, and printed circuit board space is saved. A less obvious system implication is the effect of memory size on program loading costs.

In general, the user should be able to load a program in half a minute or so. This coincides with observed user-patience factors.

An occasional error requiring reload can be tolerated for short load times, so that lower reliability loading devices may be used. To load a 1024-byte memory in 30 seconds only requires a serial transfer rate of 300 bits/second. This assumes a parity bit for each byte. For a 4096-byte memory, the required rate jumps to 1200 bits/second. The required transfer rate influences the choice of a program loading technique. Lower rates can generally be translated into lower costs and better reliability.

It's in the area of input, output, and bulk storage that we encounter the major cost problems. The choice of I/O and bulk storage techniques also has a major effect on the

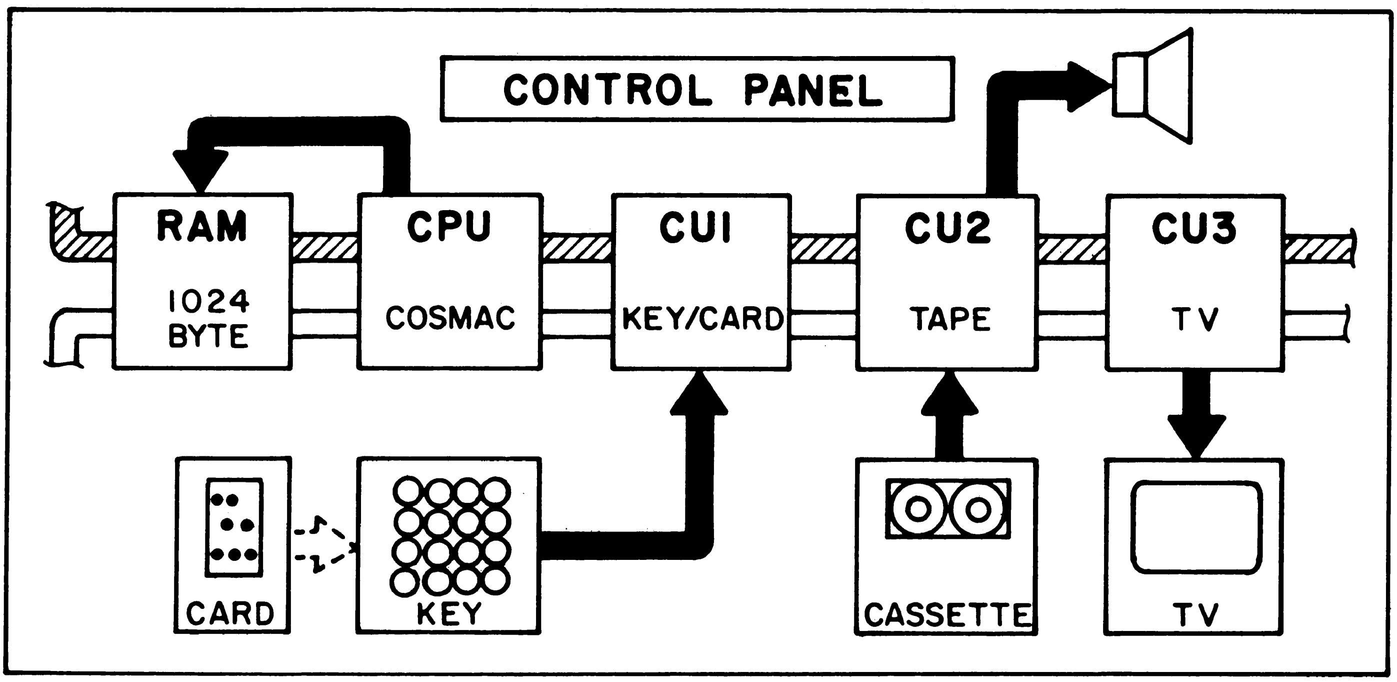


Figure 1. Basic System

range of possible system applications. Obviously, a single switch input and single light output would achieve minimum cost but would also result in a trivial system relative to use.

Output Display

Fortunately, an ideal, low-cost output device for home/school applications already exists: a standard TV set provides a flexible, dynamic output display device which most users already own.

The choice of a TV display format involves a number of system considerations. These include types of applications, display refresh memory requirements, and complexity of control circuits. A low resolution, black and white dot matrix was chosen for maximum flexibility at minimum cost. An array of white dots is displayed on a black background. The black background avoids potential picture noise problems. Arrays of 32x32, 16x64, and 32x64 dots are provided. Figure 2 illustrates the flexibility of this format for displaying small game boards, simple pictures, words, numbers, or symbols. Each dot represents the state of a main memory bit. If the bit is "1" the dot is on; if the bit is "Ø" the dot is off.

Changing the memory bit pattern immediately changes the TV picture accordingly. Bit patterns are readily moved in and out of the displayed memory area by normal programming procedures. Simple animation can be achieved by modifying memory bit patterns at appropriate time intervals. Any contiguous 128 or 256-byte section of memory can be selected for display by setting a microprocessor address pointer. This display pointer can be modified at any point in a program, thus allowing the user to step through various memory display areas at any desired rate. It is easy to flash selected portions of a picture by alternating between two display areas in memory.

For 32x32 and 16x64 displays only 1024 bits (128 bytes) of memory are required for display refresh. This is only 12.5 percent of the minimum 1024 byte memory. The 32x64 display option utilizes 25 percent of the minimum system memory but provides a larger area picture when required. It is also useful in expanded memory systems. It should be emphasized that no ROM is required for TV display in the minimum system and that frame refresh storage is provided via main memory.

The TV control unit (CU3 in Figure 1) contains the circuits for generating TV sync signals and for requesting memory bytes via the COSMAC DMA channel as required for display refresh. The individual bits of each byte are used to generate a video signal. The composite sync and video signal modulates the output of a simple RF oscillator. This modulated RF can be applied to the antenna terminals of any standard TV set.

Figure 3 illustrates the detailed timing for displaying dots on the TV screen. A magnified view of four dots is shown. Each dot is two horizontal TV lines high with a two-line space between dots. An 8-byte row buffer is provided in the TV control unit. Each TV line time is 65 microseconds. During the two blank line times, between rows of dots, up to 8 bytes (64 bits) are retrieved from main memory and stored in the row buffer. During the next two TV line times the bits in the row buffer modulate the TV beam to display the proper dot row pattern. By spreading the dots as shown, the low resolution display fills up the TV screen without requiring a high refresh rate.

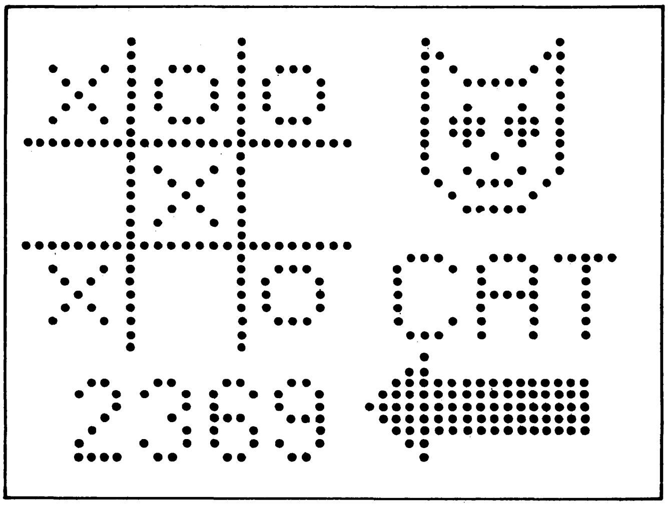


Figure 2. Display Flexibility

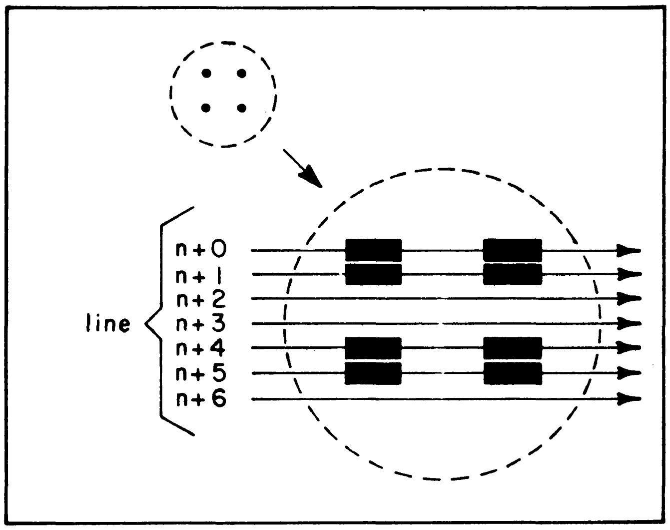
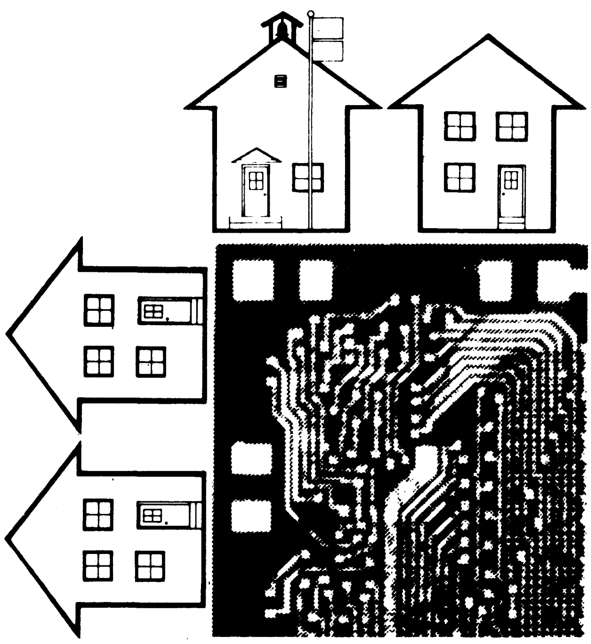


Figure 3. TV Dot Detail



The TV control unit also generates a program interrupt signal at the beginning of each TV frame. This interrupt permits the program to initialize the microprocessor display address pointer at the appropriate time. Since TV program interrupt occurs 60 times/second, a free real-time clock exists when needed. This clock capability is useful for timing purposes in a number of applications.

Bulk Storage

Program library storage and loading presents another major problem area in a low cost system. The high cost of existing computer devices such as floppy discs and digital tape units immediately rule out their use. Paper tape is awkward and still fairly expensive. Conventional punched card readers are expensive and inconvenient.

This problem was solved by using another existing, inexpensive consumer device – the audio cassette recorder. Suitable portable units sell for under forty dollars. A built-in unit could be provided for less than $20. Several methods for storing bit serial digital data on audio cassettes have been described[3,4] and others are possible. We developed a proprietary, pulse counting technique that yields a 50 byte per second transfer rate, tolerates missing or extra pulses, and permits tape speed variations of 30 percent. This system works well even for cheap portable audio units. Only single track capability is required in the system.

Since errors can be expected on occasion, a parity bit is added to each byte on tape. The cassette control unit checks the parity of input data read from tape and turns on an error light for incorrect parity. Reloading a program when an error occurs is a simple and quick procedure.

Figure 4 shows the single track, cassette tape format which was used. Digital or audio blocks are always framed by 4 kHz stop tones (T). The stop tone detection circuit is designed to respond only to long (.5 sec) continuous tones so that voice or music frames will not cause false triggers.

Figure 5 shows how a standard cassette player is used in the system. Most cassette recorders provide an external speaker or earphone output jack. This output is connected to the control unit as shown. Stop tones and digital data are detected via this cassette output line. A relay is also provided which permits the cassette output to be connected to a speaker under program control. This permits selected tape frames to be passed inaudibly.

The majority of inexpensive cassette recorders have a remote start-stop control jack. This is designed for use with a microphone or foot switch. For use in our system the cassette remote jack is connected to a program controlled relay. This gives the computer the ability to start and stop tape, providing the user has previously placed the cassette recorder in its PLAY mode.

The primary system operating controls comprise two toggle switches – LOAD and RUN. The LOAD switch activates the cassette control unit (CU2 in Fig. 1). The desired program cassette is selected by the user, rewound, and the recorder set to PLAY. When the first stop tone is encountered the data reading circuits are automatically turned on. Waiting for this stop tone eliminates possible noise problems at the beginning of tape. The digital data representing the program is loaded sequentially into memory at 50 bytes/second. The second stop tone automatically stops the tape via the tape control relay. Turning off the LOAD switch resets the computer. The RUN switch initiates execution of the program which was just loaded.

During program execution the tape can be automatically restarted so that the user will hear audio frame #1 at a desired time. The stop tone following audio frame #1 will automatically stop the tape. The program can monitor the state of the control relay to determine when the end of

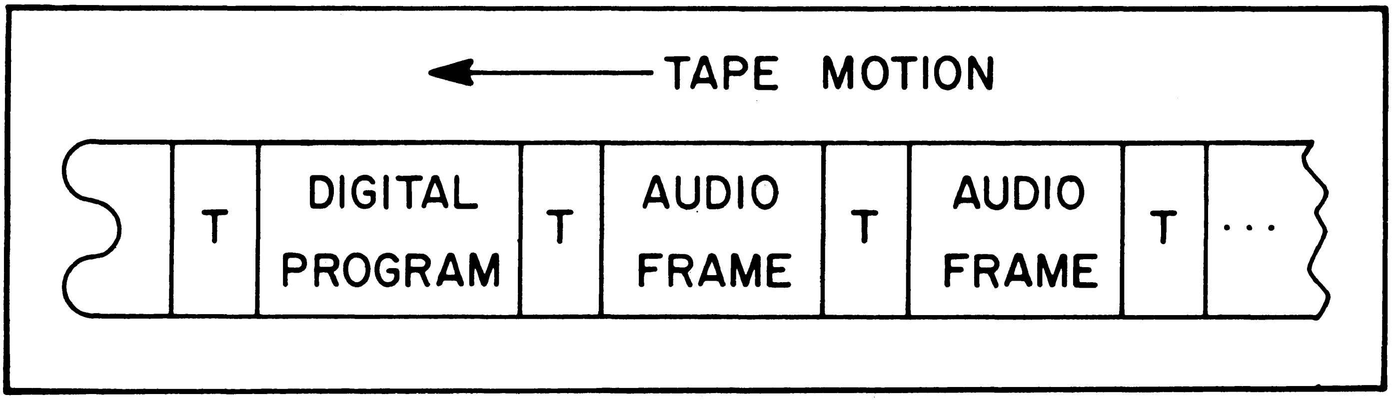


Figure 4. Cassette Tape Format

data/audio frames occur. This permits synchronizing audio material on cassettes with a program.

The provision for program controlled audio segments has an important system implication. The ability to provide instructions, questions, or other data in the form of voice frames on tape minimizes the need for a high resolution, alphanumeric TV display with its attendant requirement for large refresh and back-up digital storage capacity.

The speaker provided for use with the cassette recorder provides a useful output device. A flip-flop which can be set and reset by program drives the speaker when it's disconnected from the cassette output. Programs can, therefore, create many audible sequences of tones.

Input Devices

The primary input device for our system is a 16 position keyboard. A number of $5 to $10 keyboards of this type have been developed for use in pocket calculators. A flat, printed circuit type was chosen to facilitate an overlay feature. A slight modification of the keyboard permits insertion of a printed card above the switch array. Various cards are provided to relabel the switch array for different programs. For educational programs keys can be labeled with colors, pictures, words, or possible answers to questions. For other programs, the keys might be labeled with direction arrows for manipulation of the TV display. The variable label keyboard is fundamental to meeting the ease-of-use criterion for this type of system.

Unfortunately, the flat keyboard which is ideal for variable labeling has no tactile feel. This objection was overcome by taking a systems approach. Since a speaker already exists, switch depressions need only be coupled into this speaker to provide an audible "click." This has proven to be an adequate substitute for tactile feel. The scanning approach used to decode the switch panel minimizes the cost of this approach. Specific programs can also generate various tones for switch depressions which again substitute for tactile feel.

The 16-position keyboard normally causes an 8-bit byte to be stored in memory for each key depression. The most significant four bits (digit) are normally ØØØØ. A shift switch pressed in conjunction with a hex key causes the most significant four bits to be ØØØ1. The least significant four bits of a stored byte represent the code for one of the 16 possible hex digits shown in Figure 6. The hex keyboard in conjunction with a shift switch permits entry of 32 different codes.

An alternate mode of keyboard entry is also provided. In this mode two key depressions per byte are required. The first key specifies the most significant hex digit of the byte to be entered. The second key provides the least significant hex digit of the byte. This mode provides the sophisticated user with a convenient way to manually load his own machine language programs. It's also a useful mode for certain turnkey programs.

The hex keyboard control unit (CU1 in Figure 1) also supports the addition of an inexpensive card reader to the minimum system. This unique device uses 3-inch x 5-inch punched cards. Data is punched in the form of rows of holes. Figure 7 shows four such rows (A,B,C,D) punched on one side of a card (both sides can be punched). Each row represents the 4-bit code for one hex digit. A fifth hole is added and encoded with odd parity. At least one hole will

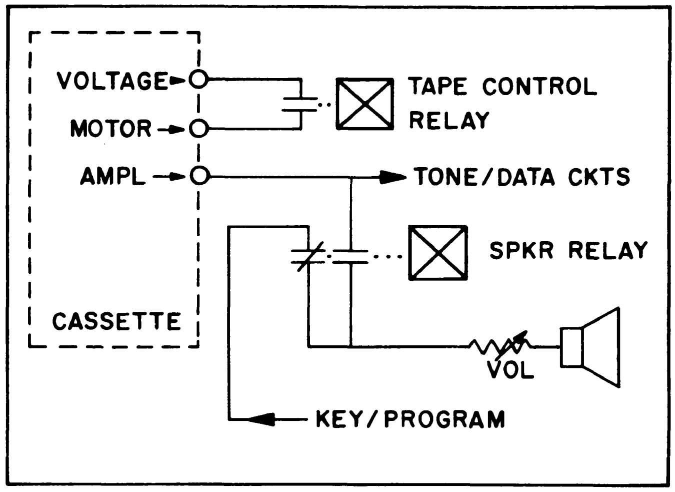


Figure 5. Cassette Attachment

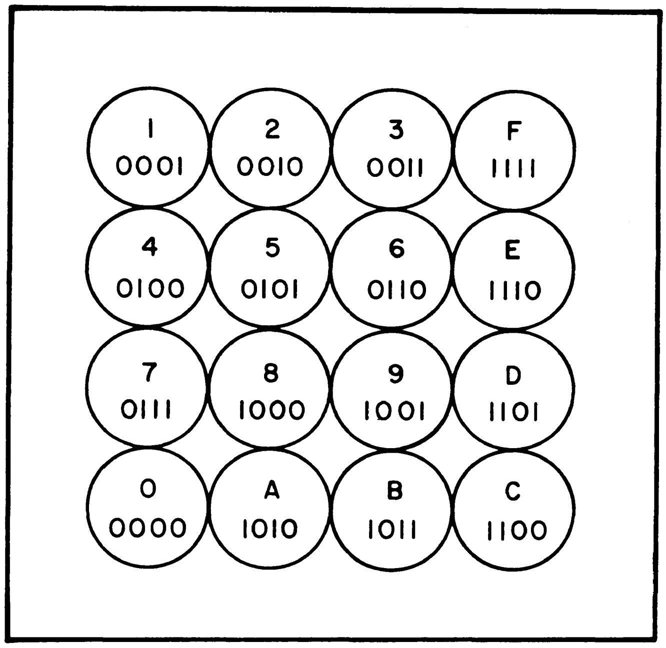


Figure 6. Keyboard Control

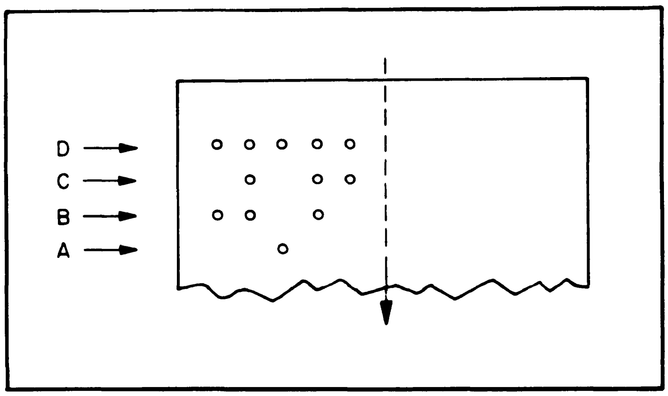
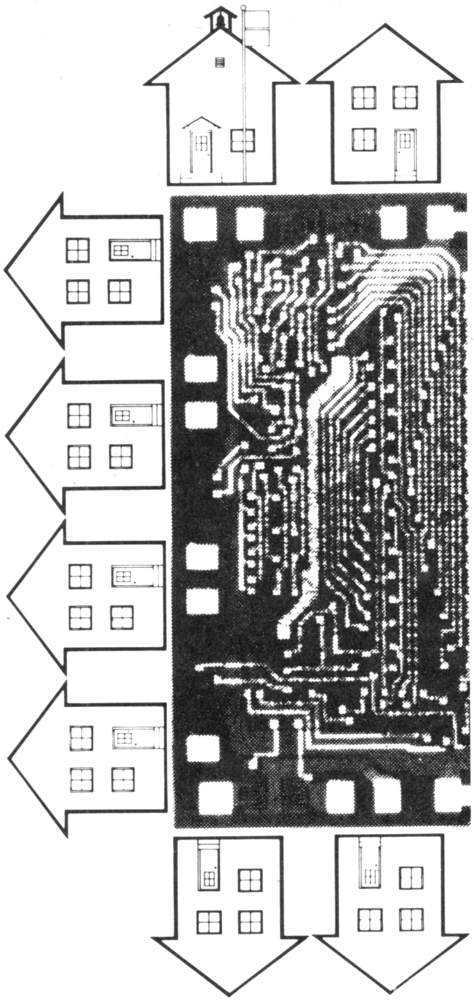


Figure 7. Punched Card



be punched for each of the possible hex digit codes. Cards are read by dropping them into a 3-inch slot. They fall past a light source and six photodiodes. One photodiode senses the presence of the card and conditions the control unit circuits accordingly. The other five photodiodes read the self clocking hex digit codes into the system. Hex digits are paired to form bytes before storage in memory.

By limiting the information content of a card to 16 hex digits per side, the mechanical tolerances of the reader can be considerably relaxed. The reader has no moving parts, and photodiodes can drive the COS/MOS control unit circuits directly. These factors combine to provide a very low cost input device ($25 or less). A simple, manual card punch can also be provided permitting users to punch their own cards.

The low cost card reader can be used to enter short lists of parameters or short user prepared programs. In a classroom, the teacher might use parameter cards to set up test/drill programs. Picture cards used by the student could contain the spelling of the word pictured for checking by the computer. The cards also facilitate certain simulation languages and permit users to save simulation language programs that they develop.

Another low cost, optional input device is a simple light gun. This contains a lens system and a photodiode. The computer detects when the gun is pointed at any lighted area of the TV screen. The light gun facilitates various computerized target shooting games. By alternately flashing portions of the TV display a program can determine the area at which the light gun is pointed. This permits the user to indicate various types of choices by pointing the gun at appropriate portions of the display.

Applications Philosophy

The open ended aspect of a stored program computer differentiates it from other types of recreational and educational devices. Any number of special purpose devices such as TV games, shuffleboard tables, electric football games, and educational toys are ideally suited to their intended function. None of these, however, will change their characteristics as user moods or interests change. Many of these special purpose devices are seldom used after their initial novelty expires. The stored program computer is a general purpose device. New programs can adapt it to changing moods and interests without the expense of new hardware. It can satisfy the needs of young and old and can grow with individual abilities.

The real value of the home/school system lies in its ability to stimulate and develop human capabilities that are often ignored or discouraged by conventional recreational and educational devices. The computer system provides an environment that stimulates experimentation, analysis, and creativity. For example, contemporary TV encourages passive viewing. However, the computer attached to a TV set enables the user to interact and play a game with the TV set. As the games played increase in sophistication, the user is encouraged to improve his analytical abilities. The user can subsequently be encouraged to experiment via specific programs or eventually to write his own programs.

For a child, the computer may initially provide arithmetic or spelling drills. Even this kind of memory development can be made more interesting via interaction with the computer. However, the child will eventually begin to wonder about the computer. Programs are made available which stimulate this curiosity and let him experiment with changing game rules. He can even begin to formulate and develop his own simple programs in a variety of simulation languages. While the initial use of the computer involves memory skills, it eventually encourages experimentation and the development of analytical and other capabilities.

The creation of programs that stimulate the user to develop mentally is a challenging task with a high payoff in terms of satisfaction. We have only begun to explore this area of use for very small, inexpensive, practical computers of the type described here. Even so, the number and richness of uses for this type of system are surprising. Those of us who are experienced with 64,000 byte main memories and large disc files may be inclined to dismiss a 1024 byte memory system as unuseable. But, in fact, such a system can be adapted to a wide range of uses. Over 80 specific applications of the inexpensive home/school system will be listed in the following sections. Many represent classes of programs which could be developed.

Four general areas of use are identified in Figure 8. These areas will be discussed individually although there is a high degree of overlap between them. Most of the listed uses only require the basic system. Reference 18 also describes a number of uses (mostly games) that have been programmed on larger computers with hard copy output. Many of these are readily adapted to the low-cost computer.

Utility Applications

This category of applications involves use of the computer to achieve specialized functions such as those listed in Table 1.

*Four Function Decimal Calculator
Hex Binary Calculator
Game Score Keeper
*Number Base Converter
Weight/Measure Converter (Metric)
Secret Code Computer
Logic Machine[5]
Classification Computer
Gambling Strategy Computer
Other Specialized Calculators
(temperature conversion, interest, etc.)
Electronic Dice
Random Number Generator
Simulation Game Computer
Bar Graph
Interactive Audio-Visual Toy
*TV Greeting Card
*Electronic "Etch a Sketch"
TV Puppet
*Audio-Visual Demonstrator
Mind Reading Computer
Party Compatibility Computer
Programmed Timer/Controller
Stop Watch/Game Timer
Simple Electronic Organ
Metronome
Advertising Display

***Already developed for the COSMAC miniprocessor.**

Table 1. Utility Applications

A simulated four-function decimal calculator has been implemented on the basic 1024 byte memory system. This includes display refresh, digit pattern tables, and decimal arithmetic algorithms with 20-digit operand and result capability. A 2048 byte memory would permit development of a programmable calculator with multi-line display. Optional ROM chips could provide a permanently resident calculator capability if desired.

A variety of specialized calculators can be implemented on the basic system. Programs to provide scorekeeping for card, war, or commercial games could be provided. Children could have their own secret code computer. For several years a plastic toy rock identification computer has been on the market. Certain tests are performed (color, hardness, etc.) on a mineral sample. The plastic computer and a set of cards is then used to identify the sample. The basic home/school system could readily be programmed as a classification computer of this type.

Logic machines have held a certain fascination for years.[5] The computer readily simulates a variety of machines of this type. It can also be programmed to simulate gambling algorithms. A pair of dice is easily simulated for use in a number of games. Random number generating machines find use in various school courses and experiments. Serious war game fans can use computer generated battle results and score keeping to advantage. The leading magazine in the field, "Strategy & Tactics," has over 20,000 subscribers indicating a wide interest in this type of activity.

For very young children the computer simulates a variety of interactive, audio-visual toys that make sounds and change TV pictures in response to key depressions. Customized, animated, TV Greeting Cards/Decorations for Birthdays, Christmas, or Halloween can be provided. Simple, key operated TV puppets are possible. Stepping a spot around the screen permits drawing TV pictures.

The ability to synchronize audio tape frames with programs permits programmed audio-visual tutorials for home and school or eye-catching advertising displays. The basic system real-time clock facilitates key operated game timer or stop watch capability. The program controlled speaker turns the computer into a simple electronic organ or metronome. TV display can be included with the sound generation.

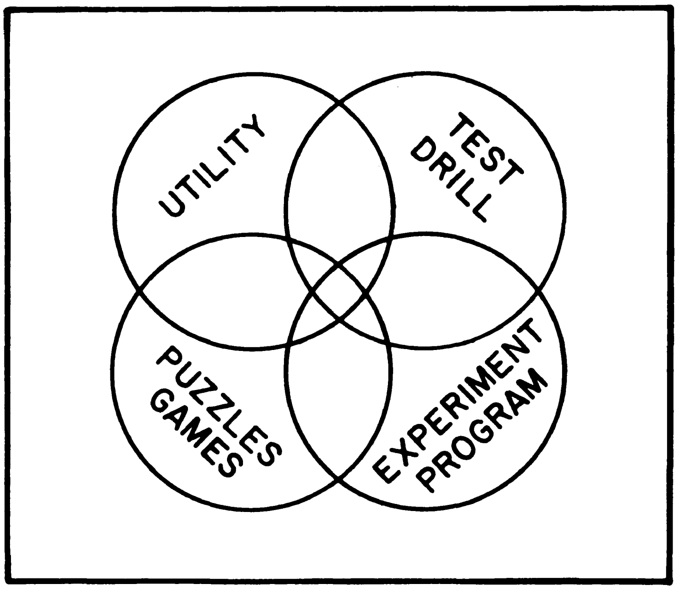


Figure 8. Areas of Use

Test and Drill Applications

These are probably the first types of uses that come to mind when education is mentioned. Drills involve the development of memory or conditioned reflexes. Testing can involve the development of other skills, as well. The infinite patience of a computer makes it ideal for drills. Interactive capability adds interest and motivation. Some specific examples are included in Table 2.

*TV Arithmetic Drill
*Word Spelling Drill
*Word Recognition Test
*Pattern Recognition (Superimposed, Complex)
Electronic Flash Cards
Classroom Group Games
Preschool Shape/Color Recognition
Up-Down, Left-Right Discrimination
Sound-Picture Matching
Reading Readiness Skill Drills
Logical Aptitude Test[6]
*Number Base Conversion Drill
Flap Board Simulator[7]
Morse Code Drill
Reflex Testing
*Logical Deduction Test (21 Questions)
Logidex[8]
Memory Training (Sobriety Test)
Individual Testing & Scoring Aid
Change Making Drill
X-Y Curve Plotting Drill
Time Sense Development

***Already developed for the COSMAC miniprocessor.**

Table 2. Test and Drill Applications

The number of different programs possible in this area is virtually unlimited. Programs of this type are ideal for individual use to overcome specific weaknesses. Unlike the teacher or parent, the computer does not make value judgements about the child during a drill. The drills can also be made to appear as games with the computer providing added motivation.

A simple arithmetic drill might appear as shown in Figure 9. Addition problems are randomly generated on the TV screen. The child must enter correct answers via the keyboard in time to prevent the boat from completely sinking. The rate at which the boat sinks can be preset by the teacher and changed from session to session to maintain challenge as the child's speed and accuracy improves. The computer displays the child's score when the teacher enters a special code (key/card).

Spelling drills can be implemented in several ways. A cassette voice could ask for the spelling of a word which the student then spells via the keyboard. The TV display and/or audio tone responds if he is right. The computer again keeps score and times the answers. A simple word recognition drill involves displaying a word on TV and asking for the corresponding picture via keyboard or card input. Patterns can be superimposed on the TV screen and the student asked to identify the components of the picture. Simple preschool shape, color, or sound recognition programs are possible. Up-down, left-right concepts are readily presented via taped voice and animated TV displays. Most test and drill programs become classroom games. Team members take turns answering computer questions and the computer announces the winner.

The computer can be programmed to momentarily flash a picture, word, pattern, or group of symbols on the TV screen to develop perception skills. Reflexes or time sense can be developed by requiring a specific keyboard response following programmed sounds or TV displays. Scoring adds a motivating game element to these types of drills. Morse code is taught by requiring the translation of tape voice passages into key depressions. The computer checks accuracy and gradually increases speed.

Reading readiness skills include simple shape recognition, word configuration recognition, and ability to maintain fixation on a moving object. The latter could involve having a child press direction-changing keys to prevent a moving TV spot from hitting obstacles. While not explicitly designed for this purpose, many of the games played with the system also develop reading readiness skills.

The computer can easily simulate logical aptitude testing devices[6], existing simple educational aids[7], or games[8]. The dot array TV display format is ideal for X-Y plotting practice. The computer can be used for individual testing and scoring in any subject area and at any grade level. The test questions are provided in printed page or booklet form.

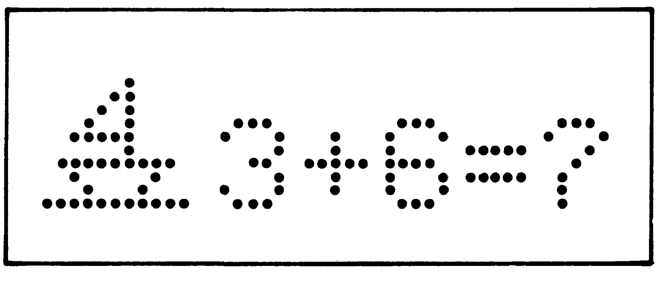
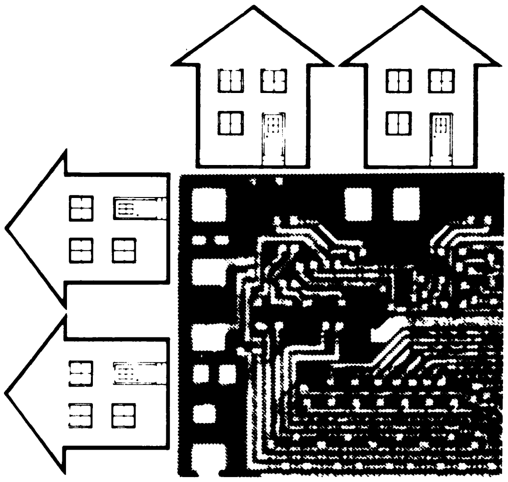


Figure 9. Add Drill Display



The computer specifies which questions are to be answered via taped voice or TV display. Answers can be in the form of multiple choice, numbers, or words which the computer can check against a prestored table of correct answers.

The ability to skip audio frames on tape via the program controlled speaker relay provides added flexibility for test and drill applications. Two sequential voice frames could be provided per question. One frame would "tell" the student that his answer was right. The other frame would "tell" him that his answer was wrong, and why. For each student response the computer "plays" the appropriate frame and "skips" the other.

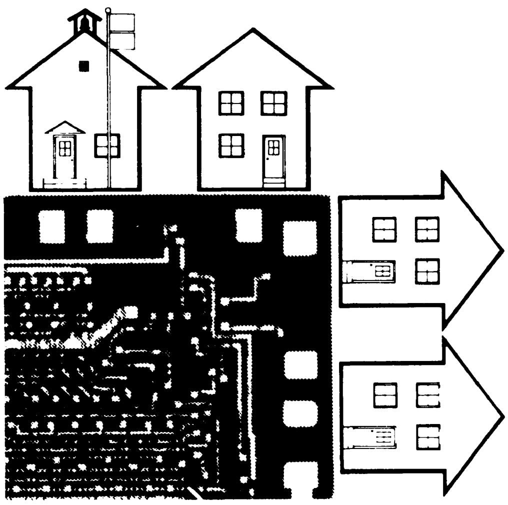
Games and Puzzles

Games and puzzles are normally associated with recreation. We have already seen that a number of utility programs have recreational aspects. The educational as well as recreational aspects of games and puzzles will be discussed here. Some of the possible uses of the computer in this area appear in Table 3.

*TIC TAC TOE
*Hexapawn[9]
*Sliding Block Puzzles
*State Change Games/Puzzles[10]
*Bowling
Football[11]
*Minikreig
*Target Shoot (Optional Gun)
*One Armed Bandit
*Network Games
*Twenty One
*Cell Matching Games
*Maze Tracing (Invisible, Changing)
*Race Games (Against Time)
*Space War
Bombs Away
Combinational/Sequential Puzzles[12]
Dodge Games (Space Ship & Asteroids)
Fish Card Game
Moon Landing
*NIM Games (Static/Dynamic)
Invisible Counter Board Games
Simulation Games[13]
Game Forms of Utility/Test/Drill Programs

***Already developed for the COSMAC miniprocessor.**

Table 3. Games and Puzzles



One of the most obvious aspects of this list is that there was no problem in motivating people to write game programs. TIC TAC TOE, Hexapawn, Twenty One, and Space War are played against the computer. Two player versions are, of course, possible. Hexapawn[9] was implemented as a learning program. The computer learns to play perfectly only after a number of games have been played. This type of learning program provides the basis for experiments where the user plots games played versus games lost by the computer to establish empirical learning curves. Heuristic versions of other games such as TIC TAC TOE are also possible.

A less familiar class of game involving interactive cell state changes is illustrated by a small plastic device called think-a-dot which has been sold for a number of years.[10] These types of games involve deducing the rules experimentally before any skill can be developed. The computer permits variations which would be otherwise impractical.

Bowling displays the pins on the TV screen. A ball spot randomly moves up and down at the opposite end of the alley. Pressing a key at the proper time rolls the ball. Sound effects, scorekeeping, and some random factors are incorporated in this two-player game. A variety of football games are possible. The simplest of these involves simulating commercial varieties of electric football[11]. Minikreig is a simplified war game. With the optional light gun a variety of computer controlled target shooting games can be devised.

Sliding block puzzles are easily simulated via the TV display. A move counter keeps the puzzle solver honest. Cell matching games involve momentarily flashing an array of cells on the TV screen at the beginning of the game. Each cell contains a symbol. Two players take turns trying to find cell pairs with matching symbols. The player with the most matches wins the game. Network games involve completing a path from one point to another before your opponent. A number of published combinational/sequential puzzles can be easily simulated[12].

Manipulating a moving spot through a racecourse or maze in the minimum time has proved to be a popular pastime. Spot acceleration and deceleration maximizes the challenge. The computer also permits invisible and changing mazes to be easily implemented. NIM type games can have a dynamic aspect included by using the computer. Board games can incorporate invisible counters whose positions must be deduced. Moon landing involves selecting fuel burning rates so as to avoid crashing. This type of game lends itself to experimentation, since it represents a simple simulation situation.

The whole area of using small computers in simulation games requires further investigation. Reference 13 contains brief descriptions of over 500 such games currently available for educational purposes. The availability of an inexpensive computer for referee, controller, and randomizing functions should be welcomed in this area.

Many of the utility, test, and drill uses previously discussed also provide the basis for games or puzzles. A widely available, mass market computer will undoubtedly stimulate the invention of many new games and puzzles for recreational and educational use.

Experimentation and Programming Uses

This area might be thought of as primarily educational. There is, however, a recreational aspect as well. The unique characteristics of a computer that make it ideal for experimentation also provide the basis of a fascinating hobby. Developing programs for your own computer embodies both educational and recreational aspects. Some specific experimentation and programming possibilities are listed in Table 4.

*LIFE[14]
Penny Matching Computer[15]
Turing Machine[16]
*Tutorial Computer
Picture Computer
Sound Computer
Machine Code Programming
Simulations
Variable Rule Games
Logic Simulator
Learning Machines
Probability & Monte Carlo Experiments
Heuristic Program Design

***Already developed for the COSMAC miniprocessor.**

Table 4. Experimentation and Programming Uses

A classic example of experimentation via computer is provided by Conway's game of LIFE described in Reference 14. The hours of bootlegged computer time devoted to this program at computer centers all over the country are a testimonial to its recreational value. LIFE simulates a succession of generations for a colony of cells. Cell birth, survival, and death are controlled by algorithms in the program. Watching the patterns of cell change for each generation on the TV screen is addicting. The program is extremely rich in experimental possibilities. New starting patterns that yield interesting and sometimes surprising life histories are constantly being discovered. The availability of an inexpensive computer would permit even unsophisticated users to experiment via programs of this type.

Heuristic programs for simple games such as Hexapawn and Penny Matching[9,15] let the user develop experimental learning curves. This approach has been used to add interest to grade school math even without the availability of a computer.[17] Letting the user modify program behavior via keyboard parameters stimulates more creative and sophisticated experimentation; even TIC TAC TOE becomes a fascinating educational device when the user is allowed to modify the rules that the computer uses.

The computer can provide a variety of simple simulations that encourage experimentation. A simple moon landing game or racecourse game with acceleration and deceleration controls are examples. A logic simulator would permit experimenting with arrays of logic elements. Commercial hardware logic trainers are quite expensive. A random number generating program facilitates experimental development and understanding of probability curves. Game theory experiments are a natural application. None of these uses requires programming ability on the part of the user.

The area of programming provides the richest and most valuable recreational and educational experiences. Programming capability can be provided at several levels. A simple set of simulated instructions to move a spot around the TV screen could be provided via card or key symbols. Programming this picture-drawing computer could be introduced as early as second or third grade. The ability to program sequences of audible tones (or music) via a simple simulation language is also easily provided.

At a slightly higher level of sophistication, various tutorial computers can be simulated. A simple fixed word, decimal computer was simulated on the basic 1024 byte system. This included ten instructions, 100 words of user memory, and a simulated control/debug panel. Teenage children were able to write and debug their own programs with as little as one hour of instruction. Simple, simulated, tutorial computers open the door to a variety of interesting educational projects. What better way is there for a student to learn than by teaching his computer.

The construction of a hardware Turing machine model for educational purposes is described in Reference 16. The authors list several disadvantages inherent in the alternative approach of computer simulation:

A. Computer time too expensive.
B. Students have to learn how to operate the computer which has nothing to do with the simulation.
C. Graphic output display is too expensive and printed output is too slow and inconvenient.

It is interesting to note that the home/school system readily overcomes all three objections. How many other valuable educational tools might be easily provided via simulation if this type of inexpensive computer was made widely available?

For the sophisticated hobbyist the area of machine code programming will be of major interest. All that is required is to add circuitry for writing cassette tapes. This is an inexpensive option. The use of a small set of programming conventions together with specialized subroutines permits the sophisticated user to develop, debug, and save his own machine code programs.

Conclusions

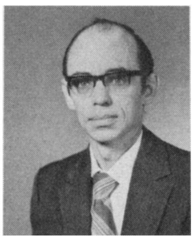
A practical, inexpensive, free standing computer system based on the RCA COSMAC microprocessor has been described. The fact that over 30 programs are already running on prototypes demonstrates its viability. Over 80 uses are listed to dispel any notion that limited capability machines of this class are necessarily trivial. Several basic system enhancements were mentioned and many others are possible.

A system of this type would, for the first time, permit widespread access to computers. Much of the public awe and confusion relative to computers would be dispelled. The creation of a group of home computer hobbyists will stimulate invention and development of new computer devices and applications. Educational benefits are unlimited.

Computers are considered to be useful tools with which to achieve a specific end result such as processing a payroll or calculating a trajectory. This view of computers has often carried over into educational applications with the computer cast in the role of teacher/tutor. The low-cost home/school system described here is intended as a flexible plaything which encourages experimentation and stimulates a desire to learn. This approach may be more significant than the improvement of teaching methods for unmotivated students.

Acknowledgements

A. R. Marcantonio, C. T. Wu, and B. J. Call have made a number of contributions to the development of this low-cost computer system. A. D. Robbi and P. Russo have provided continuing moral support, ideas, and major assistance in developing and evaluating the system. Most of all, R. O. Winder deserves credit for making the development of the system possible. ■



Joe Weisbecker has been developing new computers, communications terminals, I/O devices, and computer related commercial games for almost 20 years. He held engineering, advanced development and product planning positions within RCA's Computer Systems Division prior to joining the Lab in 1970. Mr. Weisbecker is a graduate of Drexel University, has written several articles, received two RCA Laboratories Achievement Awards, and holds 21 patents, with others pending. He is a Senior Member of the *IEEE* and a member of the Computer Society. The computer described in this article has been accepted as a member of the family by Jean (his wife), their two children (Joyce and Jean) and even their dog who punches random number cards with her teeth.

References

1. J. Weisbecker, "A Simplified Microprocessor Architecture," *Computer* (March 1974).

2. N. Swales and J. Weisbecker, "COSMAC – A Microprocessor for Minimum Cost Systems," 1974 INTERCON, Session 17/2.

3. "Putting Data on an Ordinary Audio Recorder," *The Electronic Engineer* (May 1972) p. DC-9.

4. E. Wolf, "Ratio Recording for Lower Cassette Recorder Cost," Computer Design (December 1972) p. 76.

5. M. Gardner, *Logic Machines and Diagrams,* McGraw-Hill, 1958.

6. A. Opler, "Testing Programming Aptitude," *Datamation* (October 1963) pp. 28-31.

7. J. Jones, "The Flap Board: A Simple Diagnostic and Remedial Tool," *Educational Technology* (January 1974) pp. 59-61.

8. H. Nurse, "Logidex," *Popular Electronics* (November 1973) pp. 63-66.

9. J. L. Hughes and K. J. Engvold, "Hexapawn: A Learning Demonstration," *Datamation* (March 1968) pp. 67-73.

10. B. L. Schwartz, "Mathematical Theory of Think-A-Dot," *Mathematics Magazine* (September-October 1967) pp. 187-193.

11. R. F. Graf and G. J. Whalen, "Electronic Football Lets You Play Like the Pros," *Popular Mechanics* (October 1967) pp. 147-149, 228.

12. J. W. Cuccia, "The Princeps Puzzle," *Popular Electronics* (May 1971) pp. 27-32.

13. D. W. Zuckerman and R. E. Horn, *The Guide to Simulations/ Games for Education and Training,* Information Resources, Inc., 1973.

14. M. Gardner, "John Conway's New Solitaire Game – LIFE," *Scientific American* (October 1970), pp. 120-123.

15. D. W. Hagelbarger, "Seer, A Sequence Extrapolating Robot," IRE Transactions on Electronic Computers (March 1956), pp. 1-7.

16. I. Gilbert and J. Coh, "A Simple Hardware Model of a Turing Machine: Its Educational Use," Proceedings of the ACM Annual Conference (August 1972) pp. 324-329.

17. J. Ackerman, "Computers Teach Math," *The Arithmetic Teacher* (May 1968), pp. 467-468.

18. D. H. Ahl, Ed., *101 Basic Computer Games,* Digital Equipment Corp., 1973.