OHIO SCIENTIFIC TECH NEWSLETTER #18

August 31, 1979

SALES NOTES

FREIGHT ON RA'S

Returned units coming in under an RA number are shipped back to the dealer one of two ways:

1) If the items returned were under warranty, the freight back to the dealer would be paid by Ohio Scientific.

2) If the items returned were out of warranty, then it is shipped back to the dealer freight collect.

Ohio Scientific reserves the right to decide by what means an item will be returned.

AUGUST SALE

Please note that the August CD2+2 Sale ends August 31. Any items ordered during this sale but not yet paid for will be shipped later at the regular price, not the sale price. Any items ordered under the August CD2+2 Sale and paid for will still be shipped at the sale price.

QUALITY CONTROL SURVEY

Just a reminder for the 50 dealers choosen to participate in the quality control survey, please return your input to Ohio Scientific as soon as possible. Also, thank you all in advance for your help and cooperation.

QUALITY CONTROL TASK FORCE

Ohio Scientific has implemented a Quality Control Task Force (QCTF). This is to better our service to our dealers. The QCTF will be headed by our Quality Control Manager, Mr. John Yellenic. John was recently promoted to head Quality Control from heading the Returns Department.

Also on the QCTF is Mr. James Burkett, Executive of Operations and Mr. David Clapp, Technician/Union President.

If any of our dealers have any input for the QCTF, please provide documented examples to your Sales Coordinator.

MONTHLY ADVERTISING

As many of our dealers already know, Ohio Scientific has an advertising campaign. This consists of two general ads of our Challenger III line and quotes the prices as ranging from $6,000 to $20,000 depending on the system's requirements. Many of our dealers participate in this every month. For the mutual benefit of both our dealers and the Sales Department, below is a coupon to enroll the dealer in this program on a continuous basis. This includes both Business Week and the Wall Street Journal each month. If you are interested, please fill this out and return to your Sales Coordinator. A photostat of this will be returned to each dealer upon receipt by Ohio Scientific. This will prevent exclusion of any dealers interested.

ENROLLMENT FORM FOR MONTHLY ADVERTISING

Dealer Name: _____

Street Address: _____

City: _____

State: _____

Zip: _____

Phone Number: _____

INFORMATION TO BE LISTED IN ADS

Company Name: _____

City: _____

State: _____

Phone Number: _____

The undersigned is hereby enrolled in the Ohio Scientific Advertising Campaign for business systems. It is acknowledged that a bill for each month's participation will be sent and withdrawal from this cooperative venture will be accepted upon written notification only.

Name of proprietor: _____

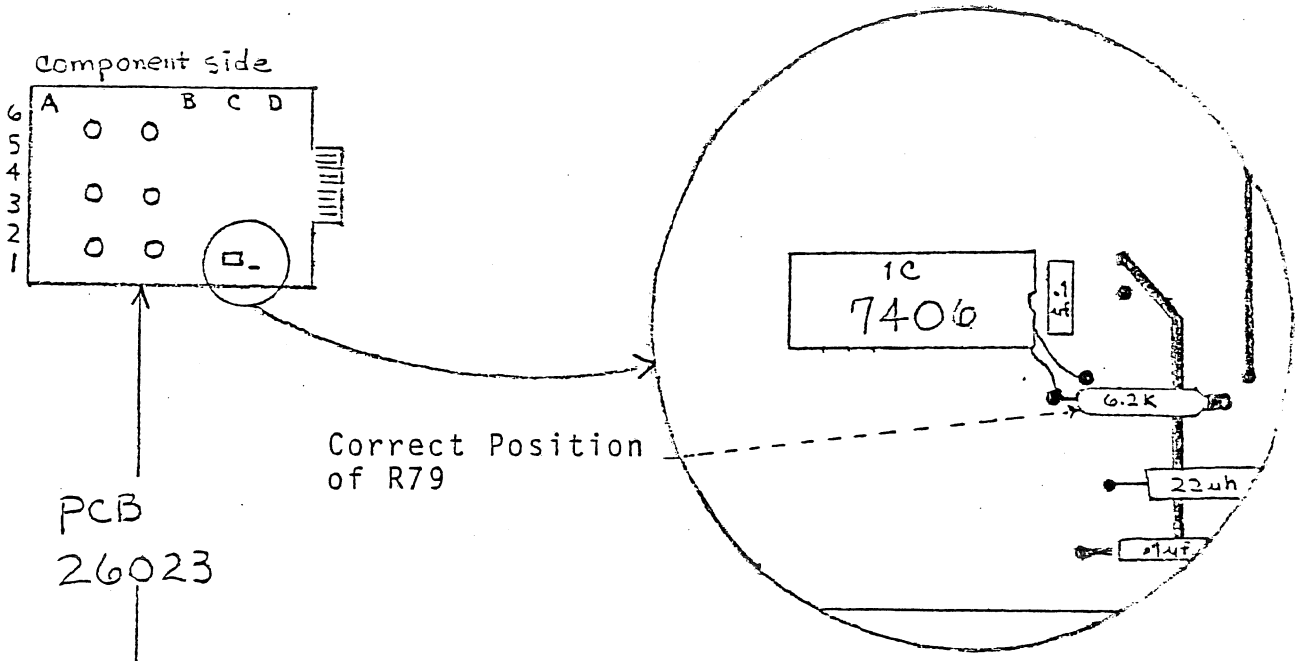
Signature of proprietor: _____

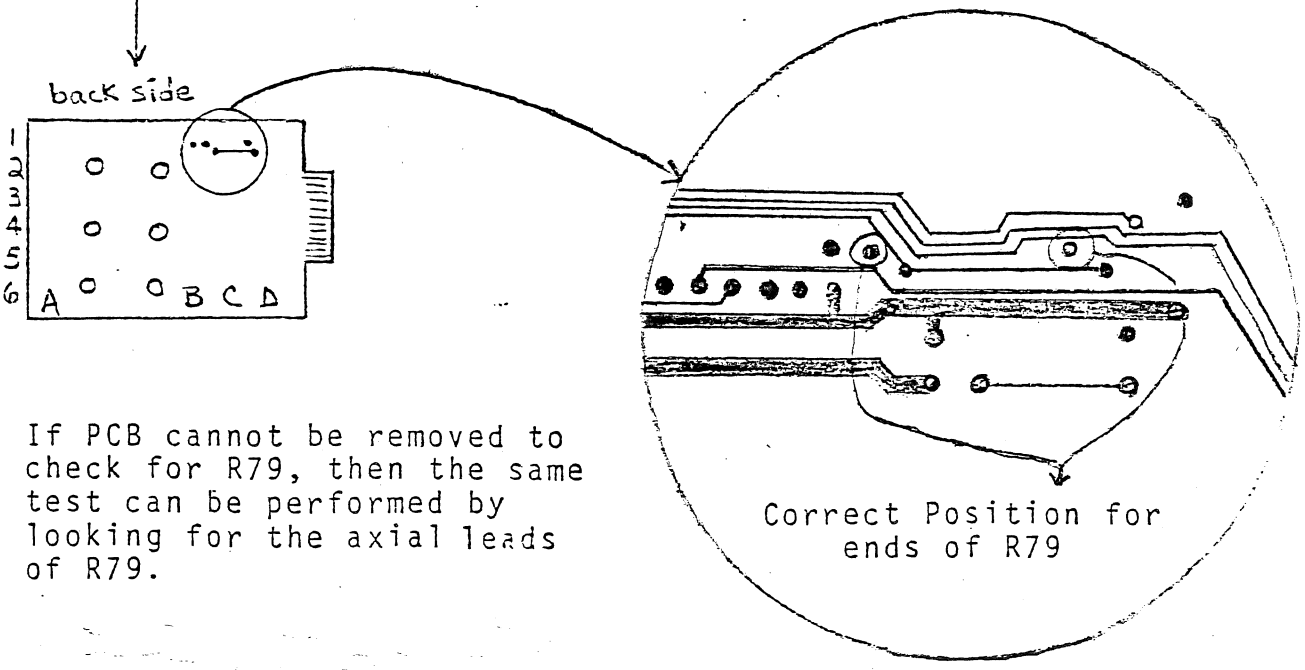
Shugart Associates
435 Oakmead Parkway, Sunnyvale, CA 94086 Telephone: (408) 733-0100

Technical Memo

Product: SA4000 Date: 7/27/79 Technical Memo #:3

There is a potential problem on any SA4000 manufactured between March 1979 and July 1979. On the R/W PCB, R79 may have been stuffed into the wrong position. Depending on the values of the other components, this condition could manifest itself as a write glitch when powering up or down. This glitch will affect data cells under the selected head only. Depending on your power supply decay time, the glitched area could be from a few bits up to half a sector.





If PCB cannot be removed to check for R79, then the same test can be performed by looking for the axial leads of R79.

OS-65U POKES / FORCING MONEY-MODE CONVERSIONS

The POKEs below will force the next numeric to string conversion to be in the money-mode format. For example:

To force $R money-mode conversion
N = number to be converted to a string
POKE 9734,128: N$ = STR$(N)

To force $L money-mode conversion
POKE 9734,129: N$ = STR$(N)

WP-2 AND THE "HOLD" COMMAND

In order for the "HOLD at end of page" feature to work, the "F" command must be used. If "FØ" is used, no page numbers will be printed.

For example:

HOLD ON <CR>

LØ7Ø,FØ,ON8 <CR> (without page numbers)

OR

HOLD ON <CR>

LØ7Ø,FlØ,)N8 <CR> (pages numbered starting with page #lØ)

9 - DIGIT BASIC VARIABLES

This article shall attempt to explain how BASIC stores and accesses variables. There are three types of variables in BASIC. They are floating point, integer, and string variables. The three types of variables may be non-subscripted, i.e. simple variables or subscripted variables.

VARIABLE STORAGE

Diagram one shows where in memory variables are stored. The BASIC program starts at the address pointed to by TXTTAB. Simple variables start just after the BASIC program. Immediately following simple variables in memory are array variables. Following the array variables in memory is the free memory space. After the free memory the actual string data is stored.

VARIABLE DESCRIPTORS

BASIC must have a way of "keeping track" of all variables. In order to do this, BASIC retains a descriptor for each variable in the program. A descriptor contains, in the case of numeric variables, the current value of that variable. String descriptors contain information on the length of the string and it's location in memory.

Upon encountering a variable for the first time, BASIC constructs a descriptor for that variable. If the variable is a subscripted variable, BASIC constructs an array descriptor. New simple variables are tacked on to the end of the other simple variables. Insertion of a simple variable requires that the array variables first be moved to make room. Subscripted variables are "tacked" on to the end of existing subscripted variables. Adding a subscripted variable decreases the amount of free memory space. Because array variables must be shifted upward to accommodate new simple variables, simple variables should be defined early in the program. DIM statements, when encountered, force BASIC to set up an array descriptor large enough to accommodate the size given in the DIM statement.

SIMPLE FLOATING POINT DESCRIPTORS

Simple floating point numeric variables have a descriptor containing the variable name and it's value. Diagram 2A shows the descriptor. Two bytes are always reserved for the variable name regardless of it's length in the program.

Floating point numbers are always stored in memory in a "NORMALIZED" form. "NORMALIZED" means that the binary number is shifted to the left until the most significant bit (MSB) is a one. The fractional value's sign bit is then placed into the MSB of the fractional value.

The MSB of the fractional value is implied to be a one unless the number is zero. A variable whose value is zero has it's exponent set to zero. In all cases, the binary point is implied to be to the left of the MSB in the fractional value.

SIMPLE INTEGER VARIABLE DESCRIPTORS

Diagram 2B shows the format of a simple integer descriptor. As with floating point variables, two bytes are always allocated for the name. BASIC requires a method by which it may differentiate between floating point, integer and string variables. The method used is a simple one involving the variable name. Floating point names are not changed, however string and integer names are affected. Integer names always have $8Ø added to the two bytes reserved for it's name. String variables have $8Ø added to the second byte reserved for the name. Integers are stored in two's complement form with bit 7 of the most significant byte reflecting the sign. Integers are stored in two bytes with 15 bits reserved for the value. This limits the range of an integer variable to be between -32767 and +32767.

SIMPLE STRING VARIABLE DESCRIPTORS

Diagram 2C illustrates the format used for simple string descriptors. A string variable always has $8Ø added to the second byte reserved for the name. String descriptors, unlike numeric variable descriptors, do not contain the value of the variable. A string descriptor contains the length of the string and a pointer to the actual string data. Strings maybe located within the actual program text or within the string space stored at the top of memory. The location of the string is dependent on how the string was defined. If a string variable is equated to string data within quotes, then that string's descriptor would point into the program text. A string variable equated to a CHR$, LEFT$, RIGHT$, or MID$ function will create a string in string space. INPUT statements will also cause a string to be formed in string space. If one string variable is equated to another, that string's pointer may or may not point into string space. Where the string descriptor points to is dependent on where the string variable to the right of the equal sign is located. If it is located within the program, then both descriptors will point into the work space. If the string that is being equated to resides in string space, then that string will be copied to the bottom of free space and the second string's descriptor will point to the copy.

ARRAY VARIABLE DESCRIPTORS

Diagram 3 illustrates the format of array variable descriptors. The array descriptors are basically the same as simple variable descriptors. The differences are information that defines the length of the array, the number of subscripts and the maximum value for each subscript, i.e. DIM A$ (2Ø,1Ø) would set the maximum subscripts to be 2Ø and 1Ø respectively. If an array variable is encountered that has not been DIMensioned, BASIC will default to a DIMension of 1Ø. In other words the array will be set up as if a DIMension of 1Ø had been executed. It is important to note that DIM statements must be executed to be effective. In memory, arrays are stored as sequential lists. The formula for accessing a array entry is:

For a List Array A$ (N):

Address of descriptor = N * Length of descriptor + Starting Array address

e.g., N=5Ø: Array data starts at $6ØØA

Address = (5Ø*3) + $6ØØA = $96 + $6ØØA = $6ØAØ

Multiple subscripted variables are slightly more involved. Diagram 4 shows a string array in memory. The array has been DIMensioned to 2 by 3. The formula for calculating the offset into an array to access a specific variable is shown below.

Offset = Length of descriptor * (S1+((S1max+1)*S2))

Working through an example should clarify the procedure. The example will use the array A$ (X,Y). The array has been DIMensioned as DIM A$ (2,3). The descriptor length will equal 3, one byte for the string length plus two bytes for the pointer to the actual string data. Since BASIC permits use of a subscript of zero, there are actually 12 variables in the array. The legal subscripts range from A$ (Ø,Ø) to A$ (2,3). We shall refer to the first subscript as S1 and to the second subscript as S2. The maximum subscript for S1 and S2, as defined in the DIMension statement is 2 and 3 respectively. The array is organized with A$ (Ø,Ø) at the front and A$ (2,3) as the last entry. If subscript S1 increases by one the offset increases by the length of one descriptor i.e. by 3 bytes. If S2 increases by one the offset would increase by (S1max+1)*3. This may seem a little strange, so let's examine the array format a little closer. The first variable in the array is A$(Ø,Ø). The next variable is A$ (1,Ø). In other words the array is in the order A$ (Ø,Ø), A$(1,Ø), A$(2,Ø), A$(Ø,1), A$ (1,1), etc.

This then indicates that when S1 increases by one, we are moving forward by one descriptor. However, if S2 increases by one, we must move past the descriptors containing the values for A$ (Ø,S2) through A$ (S1max,S2).

Using the values in our example, the relative offset to A$ (2,3) would be:

Relative offset = descriptor length *(S1+((S1max+1)*S2))

Relative offset = 3*(2+((2+1)*3))

Relative offset = 3*(2+9) = 3*11 = 33

Therefore, if one listed to access A$ (2,3) one would set up a pointer to the start of the array descriptors. Then the relative offset would be added to the array pointer. Next one would pick up the length of the string and the two byte pointer to the actual string data. At this point we have all the information required to access the string data of A$ (2,3). Variable descriptors for array variables with more than two subscripts follow the same BASIC format. A generalized form for accessing a descriptor of a variable having N subscripts is shown below.

Relative offset = DL (S1+(S1max+1)*S2+(S2max+1)*S3+.....)

Where DL is the length of the descriptor

S1 is the first subscript
S1max is the maximum value of S1
S2 is the second subscript
S2max is the maximum value of S2
S3 is the third subscript
etc.
etc.

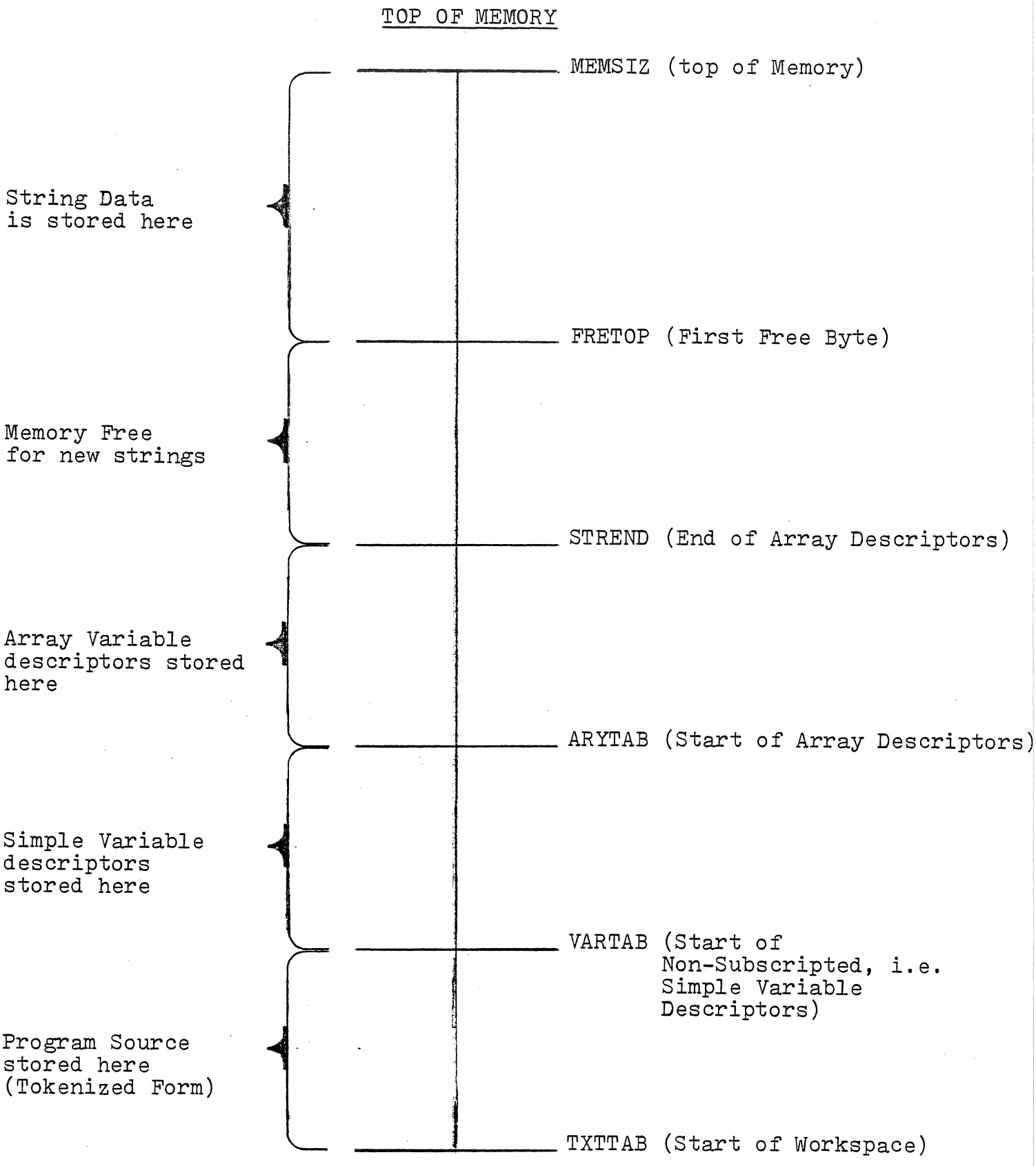
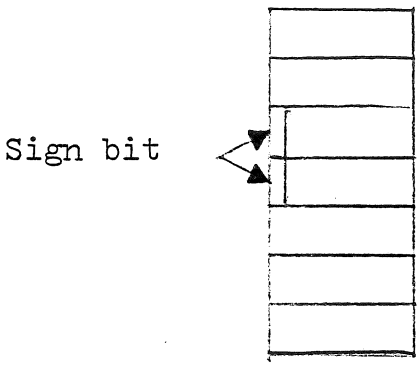


DIAGRAM 1

BASIC PROGRAM AND VARIABLE STORAGE

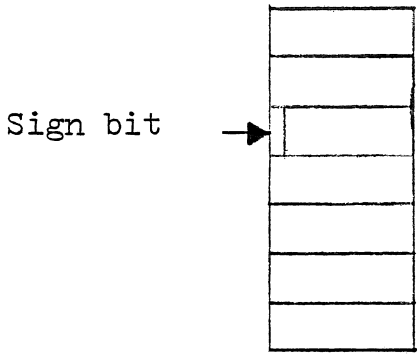
A



SIMPLE FLOATING POINT VARIABLE DESCRIPTOR

First character of name
Reserved for second character of name
Signed exponent with the sign bit complement
Fraction value sign bit, MSB fraction value
Fraction value
Fraction value
LSB of fractional value

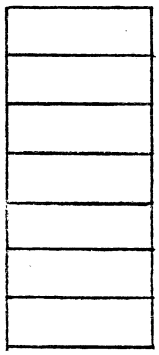
B



SIMPLE INTEGER VARIABLE DESCRIPTOR

First character of name +$8Ø
Second character of name +$8Ø
Sign bit, MSB
LSB
Ø - meaningless
Ø - meaningless
Ø - meaningless

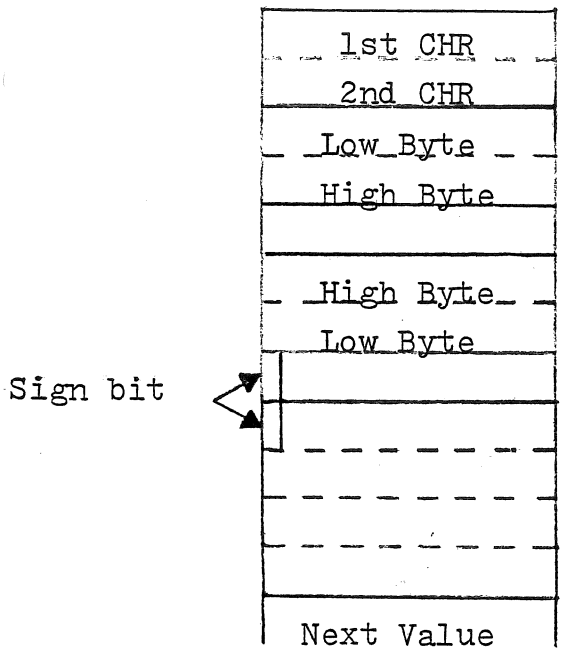
C



SIMPLE STRING VARIABLE DESCRIPTOR

First character of name
Second character of name + $8Ø
Length of string
Low address of string
High address of string
Ø - meaningless
Ø - meaningless

Diagram 2



FLOATING POINT ARRAY VARIABLE DESCRIPTOR

Name

Total length of array in bytes

Number of subscripts

Maximum subscript +1 (May repeat, one for each subscript)

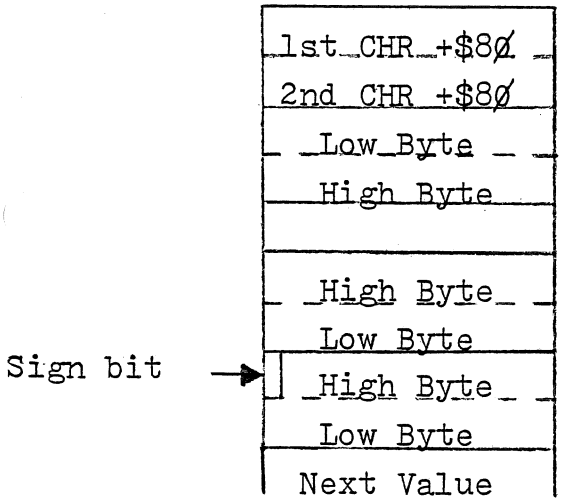
Signed exponent with sign bit complemented

MSB of Fractional value

Fractional value continued

Fractional value continued

LSB of fractional value



INTEGER ARRAY VARIABLE DESCRIPTOR

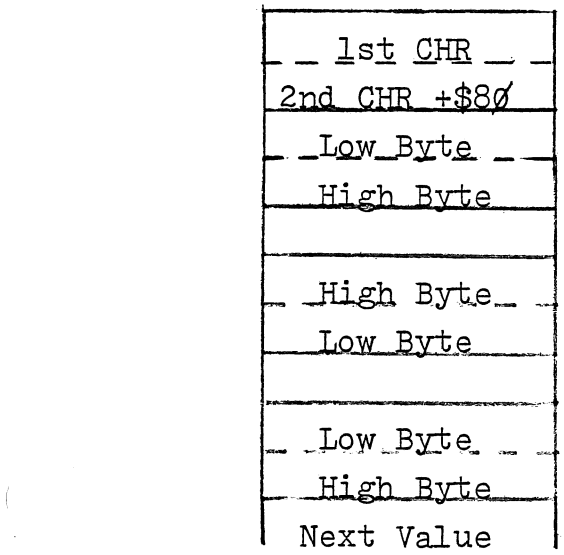
Name

Total length of array in bytes

Number of subscripts

Maximum subscript +1 (may repeat, one for each subscript)

Integer in two's complement form



STRING ARRAY VARIABLE DESCRIPTOR

Name

Total length of array in bytes

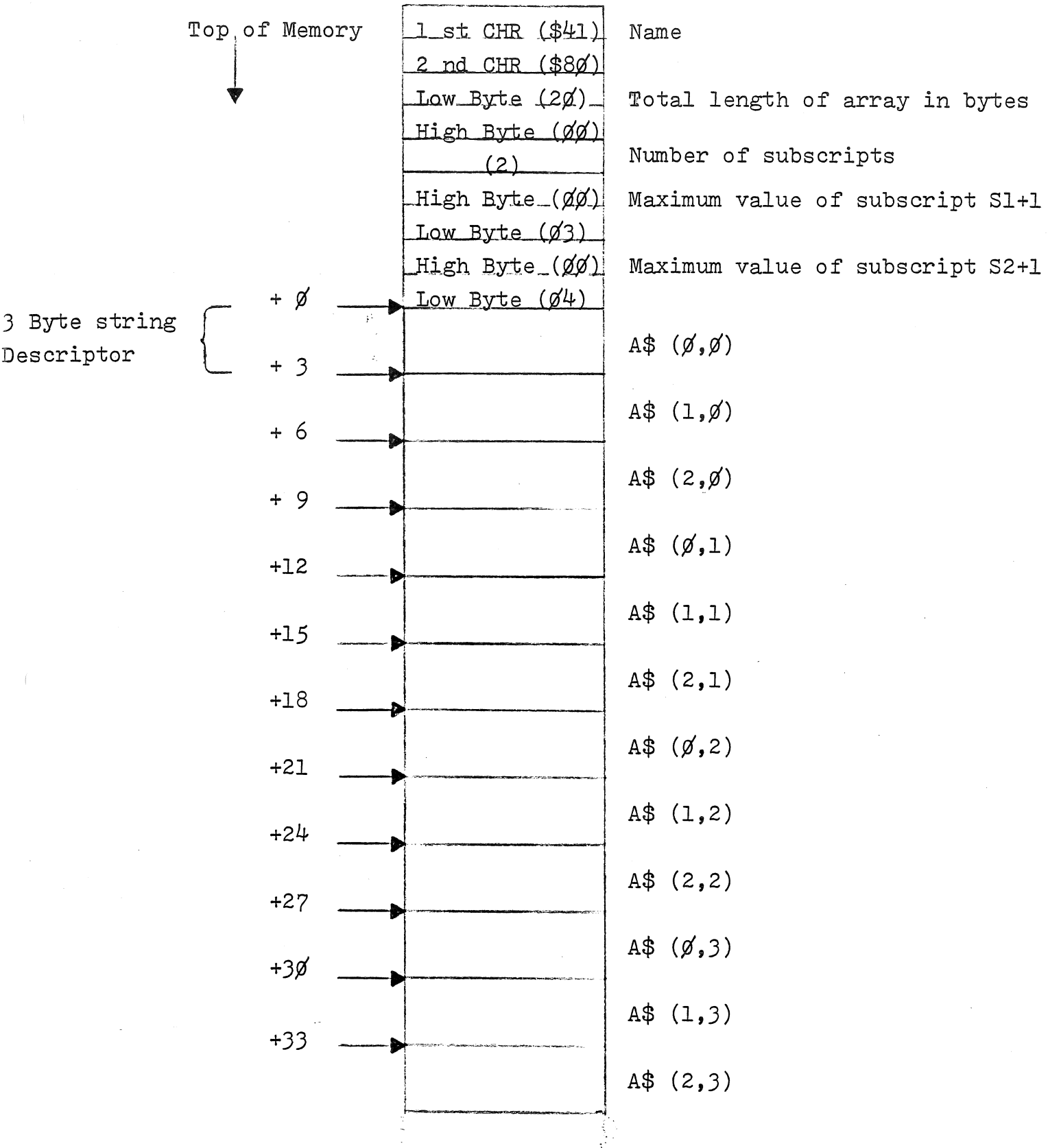
Number of subscripts

Maximum subscript +1 (may repeat, one for each subscript)

Length of string

Address pointer to actual string data

Diagram 3



ARRAY A$ (2,3) IN MEMORY

Diagram 4

Personal Computer Cassette Descriptions

EDUCATIONAL PROGRAMS

Hangman

Hangman is a popular educational game in which a person attempts to guess the spelling of a word from clues. As the user makes false guesses, the hangman is constructed. The object of the game is to guess the word before the man in the gallows is hanged.

Homonyms

Homonyms is an example of an academic test in which the student is supposed to select homonyms. This is an example of one type of testing which is possible on a computer. The data for the program can be easily changed for other specific applications. In this method of testing, the computer runs all the test questions through and, at the end, gives a summary of the right and wrong answers.

Counter

Counter is a combination educational game and cartoon for youngsters. It is specifically meant to be a youngster's first encounter with the computer. It presents a concept of the numbers one through ten in conjunction with interaction with the computer. It can be successfully used with children from preschool to about first grade. It also is an example of how a computer's cartoon capability can be used effectively.

Trig Tutor

Trig Tutor is an interactive educational program which provides tutoring and testing in conjunction with interesting graphics. It is an introduction to trigonometry including sine, cosine and tangent. This program provides an excellent example of a personal computer's capability to convey topics in an interesting fashion when its graphics and conditional response capability is utilized.

Mathink

Mathink is an intellectual mathematical game in which the user attempts to guess an equation in which three numbers will be utilized to obtain the desired answer. It is an interesting game which cultivates mathematical thinking.

Bar Graph

Bar Graph is a stand alone program which generates bar graphs on the screen. This program will be particularly useful if utilized as a subroutine to some other program. It allows you to plot up to 30 distinct points on the screen with reference lines as desired.

Definite Integral

Definite Integral calculates definite integrals for a single integral of one variable. It utilizes an iterative process (Simpson's method) in which the user can specify the number of iterations.

Basic Math

Basic Math is an excellent math tutoring exercise which randomly generates addition, subtraction, multiplication and division problems for the user. It is an excellent example of how the computer system can be utilized in repetitive drilling exercises.

Presidents

Presidents is a question and answer quiz on the Presidents of the United States. This sample educational program contains a large data base of information on the presidents. However, it doesn't really check the answers, it simply gives you the right answer after your answer is entered. It is a good example of a particular class of exercise or drill that can be performed on a computer.

Powers

Powers is a program which will calculate the power of a specified number up to 60 digits in length. If the user enters one, it generates factorials instead. It is an interesting program for showing the trends in large numbers. It is also interesting because it demonstrates multiple precision operation in BASIC up to 60 digit numbers.

NOTE: C1P Version - Generates powers up to 22 digits in length.

Trend Line

Trend Line calculates the slope of the straight line that is the "best fit" to a set of user specified points. This is accomplished by a least squares analysis. The program will then provide a graph of the points and calculate x or y values that fall on the line of best fit. Trend Line is an excellent demonstration of mathematical analysis by computer.

BASIC Tutor Series

This cassette contains six separate programs which cover the elementary statements and operations in the BASIC computer language. This instruction tape is aimed at the beginning programmer and is only intended to be an introduction to BASIC.

Electronics Equations

Electronics Equations contain ten commonly used equations from electronics: magnetic field at a solenoid, power dissipation (current and voltage), Ohm's law, series and parallel resistance, series and parallel inductance, series and parallel capacitance. The program solves for the variable indicated by the user - very handy.

Spelling Quiz

The student is given the phonetic representation of 27 commonly misspelled words and the opportunity to correctly spell the word. At the end of the quiz, the computer gives the student's answer, the correct answer and a grade.

Solar System Quiz

This program is aimed at elementary school students. The student is given a list that may be consulted at any time during the quiz. Fifteen basic questions are presented. At the end of the quiz, the student gets a grade and can compare his answers to the computer's.

Continents Quiz

The student (elementary level) is given a list of continents with each question. The answers are corrected during the quiz.

Add Game

Add Game combines an additional drill with a simple game that can be played at two levels of difficulty. The player and the computer take turns selecting numbers from one through nine and adding them up. The first one to reach 50 wins! Addition mistakes may be corrected during the game.

Math Intro

Math Intro gives addition and subtraction problems in the classic form, i.e.,

```
  18
+ 23
----
   ?
```

The user can predetermine the upper and lower limits of the numbers involved, thus, the program can be adapted to students at different levels.

Base Ten Converter

This program allows the user to convert any base ten integer (-99999 to 99999) to its equivalent in base N where N ranges between 2 and 36.

Math Blitz

Fifty problems (+, -, x, ÷) are flashed on the screen in random positions. Try to get as many right as possible - but remember, there is a time limit and it gets shorter near the end. This is a good drill to increase a student's skill and accuracy.

Personal Computer Cassette Descriptions

BUSINESS PROGRAMS

Straight & Constant Depreciation

This program calculates the depreciation for equipment via the straight and constant depreciation methods.

Inventory Demo

Inventory Demo demonstrates some of the basic principles of computer based inventories for small businesses. It maintains a file of inventory items as an integral part of the program including part number, description, quantity on hand, wholesale price, retail price and vendor. It allows the user to quickly access this information. This program is just a demo in that it is impractical for any real inventory situation because of its limited memory capacity and lack of random access storage, i.e., one needs a floppy disk drive to perform inventory on a computer.

Ratio Analysis I

The program is an aid to planning and evaluating a company's financial future by considering two classes of ratios: liquidity ratios and leverage ratios.

Ratio Analysis II

Ratio Analysis II calculates activity ratio and profitability ratio. More specifically, inventory turnover, collection period, asset ratio, pure profit, return on assets and return on net worth are evaluated. This program works well with Trend Line for long-term planning.

Bond Evaluation

Bond Evaluation is intended to be an aid in determining the initial price of a bond in order to achieve a specified rate of return.

Break Even Analysis

This program performs the analysis based on either dollar sales or units sold, making the assumption that income and total cost increase linearly. The break even point is reached when total costs equal total income. After this point, a profit is made.

Advertisement

Designed to be utilized as a display program, the user inputs up to ten lines of advertisement and selects one of three cartoon-type animations. Advertisement will continuously run the selected ad with the eye catching cartoon. Great for store windows and counter displays.

Address Book

This handy program gives the user a versatile method to store and retrieve names, addresses, phone numbers and related comments. The stored information may be accessed in a variety of methods, i.e., by name, address or phone number.

Word Processor - Cassette Version

Word Processor will edit up to 32 lines of text. Your options include: insert or delete lines or characters, global replacement of strings, find strings anywhere in the text, remove blank lines and print out edited material as data (with numbered lines) or text. This program comes with a user's manual.

NOTE: Word Processor must be used on computers with a minimum of 12K RAM.

Programmable Calculator

This program is designed to allow programmable calculator users to program the computer without any knowledge of BASIC. Five modes are available: immediate, program, clear, edit and run. In immediate mode, just key-in the steps as you would with any calculator. In program mode, the operations you key-in are stored in memory. Clear allows you to input a new program. Edit allows corrections and modifications of existing programs. When running the "program", you have a trace option allowing you to follow calculations step-by-step.

Graphics are available through GET and PUT commands. Also, diagnostics are built into Programmable Calculator. There are two types of logical branching: absolute and relative.

There is a 100-step program limit with up to 20 memories. Subroutines may be nested up to ten deep.

Programmable Calculator comes with a user's manual.

Statistics I

The user simply inputs a set of values and Statistics I outputs the arithmetic mean, geometric mean, median, mode and standard deviation.

Salary Demo

This program calculates a single biweekly paycheck for a salaried or hourly employee taking into account local, state and federal taxes. As with Inventory Demo, this program is designed to illustrate a specific computer application.

Personal Computer Cassette Descriptions

PERSONAL PROGRAMS

Annuity I

Annuity I is a program which calculates return on investments, either working from a desired resultant sum or initial sum, taking into account the amount of time money is invested, the interest rate and the method of interest compounding.

Annuity II

Annuity II calculates a resultant sum of money accumulated based on the payment in regular intervals of a fixed sum of money. The program takes into account an interest rate, period of time, payment period and method of compounding interest.

Interest on Loans

Interest on Loans calculates the classic payment table for typical home, car or equipment loans from banks. The user specifies the amount of loan, interest rate and length of loan. The program will calculate the monthly payment and generate a complete payment table.

Loan Finance

Loan Finance allows the user to specify several variables in conjunction with a possible loan. The program then creates a loan schedule from this input. Loan Finance will either determine the monthly payment or the number of years to pay off the loan from the input information such as amount of loan, interest rate and pay back period.

Uneven Cash Flows

This program gives you the effective interest rate from uneven cash flows. It is specifically designed to yield the interest rate on a loan on which there are uneven payments but can be applied to any uneven cash flow situation to generate the effective annual rate of return. The user specifies a number of uneven periods, the payment for each period and the number of months in each period.

Personal Calendar

Personal Calendar provides the user with a very flexible, easily expanded personal calendar. The program includes instructions for storing appointment information on the tape. Appointments may be added, deleted, reviewed or stored on the tape at any time.

Checking Account

Checking Account will balance a checkbook (even yours!) when given the initial amount, deposits and check amounts. Please note that there is a limit of 20 checks per run.

Savings Account

Savings Account calculates the balance on a savings account when given the initial amount, interest rate, compounding frequency, deposits, withdrawals and dates.

Calorie Counter

Calorie Counter considers your height, weight, sex and age, then estimates your metabolism. Next, by considering your activity levels, the computer predicts your average caloric requirement.

Biorhythm

Biorhythm graphs your personal biorhythms (physical, emotional and intellectual) for any specified thirty day period.

Kitchen Aid

Give the computer any recipe and it will give you the proper amount of all ingredients to serve as many or as few as the user desires.

Personal Computer Cassette Descriptions

GAME PROGRAMS

Tiger Tank

Tiger Tank is a highly sophisticated one or two player tank game similar to arcade tank games. However, it has an option to allow the computer to operate one of the tanks. Tiger Tank can provide hours of entertainment. It requires manual dexterity at the keyboard for game playing. (Requires graphics)

Lunar Lander

Lunar Lander is a classical computer game in which a person attempts to land successfully on the moon by varying the amount of fuel he burns per unit time. It is an interesting mathematical game. This version of Lunar Lander has no graphics. It can be used on virtually any OSI computer.

Breakout

Breakout is a pong type game in which the user can serve a ball and then steer a paddle to direct the ball to knock down targets. It is a challenging video game which requires manual dexterity. (Requires graphics)

4K-Trek

This is a Star Trekish version of a video arcade shooting gallery game where the Enterprise is fighting it out with turrets on a star base. 4K-Trek is a shortened version of Space War. (Requires graphics)

Tic-Tac-Toe

This is a classical implementation of the popular game Tic-Tac-Toe in which you play against the computer. You can go first or second. The computer will play a random or a reasonably tough game but it can be defeated. (Requires graphics)

Torpedo

Torpedo is a typical video arcade game in which the user fires a torpedo at a passing submarine. (Requires graphics)

Hectic

Hectic is an exotic video game which is ideally suited for microcomputers. In this game you have a turret in which you are attempting to shoot down bombs which can ultimately penetrate your shield. It has an outerspace setting and is a very challenging and intriguing game to play. (Requires graphics)

Frustration

Frustration is basically the playing board for a popular puzzle in which the player removes one item from a triangular grid and then jumps over items to remove them from the board. The object of the game is to leave only one. This particular game requires no inherent intelligence on the part of the computer, that is, it is not an interactive game. However, it is a clever implementation of this puzzle. (Requires graphics)

Star Trek

Star Trek is a classical implementation of the most popular computer game in existence including sensors, photon torpedoes, warp drives, phasers and galactic maps. It is one of the most challenging and interesting character oriented games available for computers.

Kaleidoscope

Kaleidoscope is a geometric pattern generator that continuously creates geometric images. This program is useful for display and is an example of some of the things you can do on a graphics system. There is no user intervention once it is started.

Bomber

You are the pilot of a bomber. The object of the game is to get as many points as possible by shooting down planes and bombing ground targets before one of the enemy planes get you! As you improve at this game, you can select a higher speed for oncoming aircraft. (Rquires graphics)

Destroyer

You are in command of a destroyer patrolling enemy waters. Try to sink the enemy submarines by controlling the level at which your depth charges explode. (Requires graphics)

Hide & Seek

The computer will randomly place one to ten objects in an opaque box. Try to find them by shining rays of light into the box and observing where they exit. (Requires graphics)

Star Wars

An advanced video game in which you move crosshairs around. The object of the game is to get the moving target ship into the crosshairs of your laser. You can fire at the target ship and the target ship fires back at you! This program requires keyboard dexterity. (Requires graphics)

Black Jack

The old gamblers favorite - young gamblers, too! For one to five players. As a player, you have the options of getting an extra card, doubling your bet or holding. The dealer keeps track of your winnings and losings. (Requires graphics)

NOTE: ClP Version - One player against the computer.

New York Taxi

New York Taxi is a game in which you are a pedestrian attempting to get a taxi in New York City without being run over. It is quite a challenge. (Requires graphics)

23 Matches

Here is an old strategy game - try to beat the computer at this one! The player goes first and removes one to three matches, then the computer gets a turn. Whoever picks up the last match loses. It sounds so easy!

Cryptography

Cryptography is a two player game. One player inputs a message of up to 20 letters and a clue. His opponent tries to decipher the computer's coded version of the message.

Etch-A-Sketch

Etch-A-Sketch is a computer version of the toy with the same name. Either you control the display or let the computer create random patterns. Your control options are: regular or reversed screen and create or erase lines. (Requires graphics)

Space War

Here is an outstanding two player arcade type video game. Two ground weapons fight it out with the Enterprise. The first to break through the other's energy shields is the winner. (Requires graphics)

Battleship

You are pitted against the computer in this nautical warfare game. Each of you 'hide' five ships on separate 10 x 10 grids, then try to sink the enemy's ships. At any time you may review the location of your remaining ships or a record of your shots.