

Please do not upload this copyright pdf document to any other website. Breach of copyright may result in a criminal conviction.

This Acrobat document was generated by me, Colin Hinson, from a document held by me. I requested permission to publish this from Texas Instruments (twice) but received no reply. It is presented here (for free) and this pdf version of the document is my copyright in much the same way as a photograph would be. If you believe the document to be under other copyright, please contact me.

The document should have been downloaded from my website <https://blunham.com/Radar>, or any mirror site named on that site. If you downloaded it from elsewhere, please let me know (particularly if you were charged for it). You can contact me via my Genuki email page: <https://www.genuki.org.uk/big/eng/YKS/various?recipient=colin>

You may not copy the file for onward transmission of the data nor attempt to make monetary gain by the use of these files. If you want someone else to have a copy of the file, point them at the website. (<https://blunham.com/Radar>). Please do not point them at the file itself as it may move or the site may be updated.

It should be noted that most of the pages are identifiable as having been processed by me.

I put a lot of time into producing these files which is why you are met with this page when you open the file.

In order to generate this file, I need to scan the pages, split the double pages and remove any edge marks such as punch holes, clean up the pages, set the relevant pages to be all the same size and alignment. I then run Omnipage (OCR) to generate the searchable text and then generate the pdf file.

Hopefully after all that, I end up with a presentable file. If you find missing pages, pages in the wrong order, anything else wrong with the file or simply want to make a comment, please drop me a line (see above).

It is my hope that you find the file of use to you personally – I know that I would have liked to have found some of these files years ago – they would have saved me a lot of time !

Colin Hinson

In the village of Blunham, Bedfordshire.



TEXAS INSTRUMENTS

MINI MEMORY

Solid State Cartridge
Cassette Software



- GB MINI MEMORY
- D MINI MEMORY
- F MINI MÉMOIRE
- I MINI MEMORIA
- NL MINI MEMORY




Ⓢ General Information

Before you slide a program module into your Home Computer please see »Read this first«.

Ⓢ Generelle Information

Bevor Sie ein Programm-Modul in Ihren Home Computer einstecken, lesen sie bitte zuerst »Bitte beachten«.

Ⓢ Informations générales

Veillez consulter le »Instructions d'installation« avant d'introduire un module d'application dans l'Ordinateur familial.

Ⓢ Indicazione generale

Prima di inserire un modulo di comando nel Vostro Home Computer vogliate consultare la guida »Prima di cominciare«.

Ⓢ Algemene aanwijzing

Voordat u een programma module in uw huiscomputer steekt lees a.u.b. eerst de »Lees dit eerst«.

GB	1
D	7
F	13
I	19
NL	25

Ⓢ Necessary System

HomeComputer TI 99/4A, Television Set; Cassette Recorder and Cassette Interface Cable optional

Ⓢ Notwendige Geräteausstattung

Home-Computer TI 99/4A, Fernseher; Kassettenrecorder und Anschlußkabel sinnvoll

Ⓢ Equipement nécessaire

Ordinateur modèle TI 99/4A, poste de télévision; magnétophone à cassette et câble d'interface de câble préconisés.

Ⓢ Sistema richiesto

HomeComputer TI 99/4A, Televisione; Registratore a cassetta e interfaccia cassetta Cavetto raccomandato

Ⓢ Benodigde uitrusting

Huiscomputer TI 99/4A, televisietoestel; Cassette Recorder en Cassette Scheidingsvlak kabel aanbevolen



INTRODUCTION

MINI MEMORY SOLID STATE CARTRIDGE

The Mini Memory Solid State Cartridge Command Module increases the versatility of your Texas Instruments TI99/4 or TI99/4A Home Computer by providing additional memory for your system and important tools for program development. In addition, the module contains a built-in battery, which permits the programs and data stored in the module's Random Access Memory (RAM) to be retained when the computer console is turned off, even if the module is removed from the console.

THE FEATURES OF THE MINI MEMORY MODULE INCLUDE:

- A total of 14 K bytes of memory. This memory consists of 6 K bytes of Graphics Read Only Memory (GROM), 4 K bytes of Read Only Memory (ROM), and 4 K bytes of RAM. The programs resident in the GROM and ROM provide additional important program development tools. The RAM provides additional memory space for data and program storage.
- A built-in battery in the module to preserve the data or programs stored in the RAM memory
- Additional files. Besides the 4 K-byte RAM file in Mini Memory module itself, the 24 K-byte segment of the Memory Expansion unit, if attached, can be used by TI BASIC programs.
- Assembly language capabilities. With the Mini Memory module, assembly language object programs can be loaded into the module itself or into the Memory Expansion unit, if attached.
- Additional TI BASIC subprograms. With the Mini Memory module, several additional subprograms can be called with TI BASIC statements. These subprograms include the ability to PEEK and POKE values.
- Additional utility routines. The Mini Memory module includes several program routines which permit access to the computer's resources; for example, interfacing user programs with ROM- and GROM- resident programs, interfacing assembly language with the TI BASIC interpreter, and accessing the VDP RAM.
- A resident debug program. The EASY BUG debug program is a useful program-development tool with which you can access the internal resources of the computer system and trouble-shoot your programs.

APPLICATIONS

You can use the RAM in the Mini Memory module to store either data or programs. This memory is »CPU memory,« which means it is fast-access memory. You can take advantage of this fast storage and retrieval to store data which is used frequently in an application or to store assembly language programs which perform rapid computations.

A TI BASIC program which you frequently use can be stored in the Mini Memory module, rather than on a cassette tape or diskette, for quicker loading.

Generally, Random Access Memory (for example, the »user« memory in the computer console) loses its contents when the console is turned off. The battery-activated Mini Memory module, however, retains its contents when you turn the console off. programs or data can be stored in the battery-packed RAM, the console turned off, and the module removed from the console. Then, when you reinsert the module and turn the console on, your data or program is ready to use.



CAUTION: When you remove or insert the Mini Memory module, the computer console should be OFF to prevent the possibility of losing or altering any data or programs stored in the module's RAM.

An important feature of the Mini Memory module is its capability of implementing assembly language programs. The module allows you to load your own assembly language programs for direct access to the programmable components in the computer (such as the TMS9900 microprocessor or the TMS9918 Video Display Processor). Assembly language programs can also directly access devices such as the Wired Remote Controllers or cassette tape recorders through their interface ports on the console.

In addition, the module makes it possible for assembly language subroutines to be called from TI BASIC programs. These assembly language subroutines can perform functions which would be inefficient or impossible to implement in BASIC. Program routines resident in the module's GROM and ROM provide a convenient interface between TI BASIC programs and assembly language programs.

Also included in the module is the EASY BUG debugging program. With EASY BUG, you can access the memory and programmable components in the computer. EASY BUG also includes commands for loading and storing memory-image data on cassette tape.

GENERAL MEMORY INFORMATION

Note: In the users manual, the greater than symbol (>) indicates that the following number is a hexadecimal (base 16) number.

The Mini Memory module contains a total of 14 K bytes of memory, consisting of 4 K bytes of battery-powered Random Access Memory (RAM), 4 K bytes of Read Only Memory (ROM), and 6 K bytes of Graphics Read Only Memory (GROM). Resident in the ROM and GROM are a number of routines which add additional callable subprograms to TI BASIC and which are useful for interfacing assembly language programs with TI BASIC programs.

Appendix A shows the memory organization for the computer's entire memory of the user manual space. The 4K bytes or ROM in the Mini Memory module occupy the memory space from address 6000 – 6FFF (or from 24576 through 28671). Appendix B illustrates the details of the module's ROM organization.

The 4K bytes of RAM in the module occupy the memory space from addresses >7000 through <7FFF (or from 28672 through 32767). Appendix C shows the details of the RAM organization when it is used for TI BASIC files. Appendix D describes how the RAM is organized when it is used for assembly language storage. The GROM occupies memory space which is not directly mapped into the CPU memory address space. Appendices E and F contain information about the RAM in the Video Display Processor (VDP).

SOURCES FOR BACKGROUND INFORMATION

A number of references can give you helpful background information as you begin to use the Mini Memory module. These are:

1. The Texas Instruments Home Computer User's Reference Guide;
2. The Texas Instruments Home Computer TI Extended BASIC owner's manual;
3. And, if you are creating assembly language programs, the Texas Instruments Home Computer Editor/Assembler owner's manual. This book is a complete reference guide to the TMS9900 assembly language and also contains details about the internal architecture of the TI99/4 and TI99/4A Home Computers.



HOW TO USE THIS MANUAL

The manual assumes that you are already experienced in programming with TI BASIC. Statements, commands, and functions that are the same as in TI BASIC are only discussed briefly. For a complete description, see the User's Reference Guide included with your TI99/4 or TI99/4A Home Computer.

If you intend to use the Mini Memory module for creating your own assembly language programs, it is assumed that you are experienced in TMS9900 assembly language programming and that you are familiar with the internal organization of data and file structures used by the Home Computer. For a complete discussion of these topics, see the Editor/Assembler owner's manual.

The remainder of this manual explains the various features included in the Mini Memory module. The section entitled »Using the Solid State Software™ Command Module« describes the various options that are presented on the selection screen when the Mini Memory module is first brought into operation and explains how to select these options.

The section on »Loading and Saving TI BASIC Files« explains how to use the Mini Memory module for accessing data files, and the »Loading and Saving TI BASIC Programs« section describes how to use the module for loading and storing BASIC program files.

The section on »Additional TI BASIC Subprograms« explains the additional subprograms the Mini Memory module provides to interface with assembly language programs and the computer system.

»Loading Assembly Language Programs« describes the procedures for loading assembly language programs and subprograms, and the »System Utility Routines« section explains the Mini Memory module's utilities which access CPU ROM and GROM routines.

The »EASY BUG DEBUGGER« section describes the operation of the EASY BUG debugging program, and the six appendices contain information about the memory organization of the Home Computer.

INTRODUCTION

LINE-BY-LINE ASSEMBLER

The cassette tape enclosed with the Mini Memory Solid State Cartridge Command Module contains a line-by-line symbolic assembler and a graphics demonstration program named LINES. The Line-by-Line Assembler allows you to enter TMS9900 assembly language source code, one line at a time, directly from the computer keyboard. The LINES demonstration program (operational only on the TI99/4A Home Computer) automatically draws colorful lines on the screen.

When the Assembler program is loaded in the Mini Memory module, each source statement you enter is immediately assembled into object code and stored in the memory locations specified by your source code. Therefore, as soon as you complete the entry of your program and store its name and address in the Symbol Table, it is ready to be run.

IMPORTANT NOTE: Since code is assembled and stored directly, as you enter each line, be sure that the memory addresses specified in the program are available. Otherwise, no code will be generated or stored.

Although the Assembler converts each instruction into machine code as it is entered, some source code is retained in a nine-page text buffer. You can scroll the screen to review previously entered lines of source code by pressing the up- and down-arrow keys.



This manual discusses the features of the Line-by-Line Assembler, assuming that you already know assembly language programming. For a complete reference guide to the TMS9900 assembly language, see the Editor/Assembler owner's manual. Instructions for running the LINES demonstration program are also given in this manual (see »Loading the Line-by-Line Assembler«).

Note: The LINES program is operational only on the TI99/4A Home Computer, which has the enhanced graphics processor required by the program.

LOADING THE LINE-BY-LINE ASSEMBLER

Both the Line-by-Line Assembler and the LINES graphics program are loaded at the same time from the cassette tape by the L (LOAD) command of the EASY BUG debugger option. The steps below describe the loading procedure.

1. With the Mini Memory module installed in the computer console, attach your cassette recorder/player to the console, as described in the User's Reference Guide.
 2. Press any key to make the master selection list appear, and select the Mini Memory option from the list. When the Mini memory selection list appears, press 3 for REINITIALIZE to prepare the memory for loading a new program. Then press QUIT to return to the computer's master title screen.
 3. Insert the Assembler cassette into the recorder, and rewind the cassette.
 4. Press any key to make the master selection list appear, and then select the EASYBUG option from the list.
 5. When the EASY BUG command description screen appears, press any key to clear the screen. Then type L and press ENTER to start the loading process. From this point on, the screen displays instructions to help you through the procedure. Follow these directions to load the program.
 6. After the Assembler program is loaded, press QUIT to return to the master title screen. Then press any key to go on to the master selection list, and select the Mini Memory option.
 7. When the Mini Memory selection list appears, press 2 to select the RUN option. The screen clears, and the program prompts you for the name of the program you want to run.
- A) If you want to run the LINES program (available only on the TI99/4A, type LINES and press ENTER. The program draws a colorful line design on the screen. If you press the C key, you can »freeze« the color of the current line, and all subsequent lines will be that color. Pressing C again cancels the single-color effect. To stop the program, press QUIT.
- B) If you want to create a new assembly language program, type NEW and press ENTER. The program enters the Assembler, clears the Symbol Table (more on this later), and waits for your first program line entry.
- C) If you want to continue writing a program that you began previously, type OLD and press ENTER. The old Symbol Table is retained, and the screen displays the next memory location, ready for you to continue your program.

Note: Your Mini Memory module may already be loaded with the Assembler and LINES programs. To check, select the Mini Memory RUN option, and enter the appropriate program name when the prompt appears. If present, the program immediately begins to run. If the program has not been loaded into the module, the screen reports »PROGRAM NOT FOUND«.



You may also review the LINES code by means of the M command in EASYBUG. Just enter M7CD6, and keep pressing ENTER to review each line of code without changing any data.

IMPORTANT NOTE: When you enter and assemble a program, the Symbol Table may overwrite part of the LINES program stored in the Mini Memory module. When you want to run LINES again, simply load the program again from the cassette tape into the module.

ASSEMBLER SYNTAX

Each line (or record) of your source program consists of four sections known as fields. These fields, if present (some are optional), must appear in the order and format (syntax) required by the Assembler. In this manual, the following conventions are used in the syntax definitions for machine instructions and directives.

- Items in capital letters, including special characters, must be entered exactly as shown.
- Items within brackets ([]) are optional.
- Items within angle brackets (< >) are required fields.
- A lower-case b indicates a single space.
- A lower-case b followed by three dots (b...) indicates one or more spaces.

The general syntax for an Assembler directive is as follows.

[label] b... <opcode> b [operand] [,operand] [b... comment]

The label field requires either a space (when there is no label) or one or two characters. The first character must be alphabetic; the second character, if present, may be alphanumeric. The label is followed by one or more spaces. If you do not type a label pressing the SPACE bar moves the cursor automatically to the beginning of the opcode field.

The opcode field contains the operation code of the task to be performed by the source statement. The field consists of one to four alphabetic characters, such as A for Add or AORG for the Absolute Origin directive, followed by a single space.

The operand field contains one or two operands, as required by the particular instruction. Note that the operand field has no spaces within it, and multiple operands are separated by commas. The operand field is concluded by pressing the SPACE bar (cursor advances to the comment field) or ENTER (signifies the end of the line). If an instruction has no operand, the operand field is omitted.

The comment field, if used, may include any character, and it continues until you press ENTER to end the line.

EXAMPLES:

XY MOV R1, VP
Z S R1, R2

Save R1 in VP.
Calculate difference.



The Line-by-Line Assembler predefines certain symbols. When an operand includes a dollar sign (\$) as an initial character, it is considered to refer to the contents of the location counter. For example, at location 7D00, the statements

JMP \$+8 and JMP 7D08

are considered to be equivalent.

In calling out register operands, you can use the symbol R, followed by a decimal number. Thus,

MOV R1, R15 and MOV 1, 15

are equivalent.

Note: The default number system for the Line-by-Line Assembler is decimal; hexadecimal numbers are indicated by the greater than (>) prefix.



EINFÜHRUNG

MINI MEMORY SOLID STATE CARTRIDGE

Das Mini Memory Solid State Cartridge-Befehls-Modul erhöht die vielseitige Verwendbarkeit Ihres TI99/4 oder TI99/4A Home-Computers von TEXAS INSTRUMENTS durch zusätzliche Speicherkapazität und wichtige Instrumente für die Programm-Entwicklung. Durch die im Modul integrierte Batterie bleiben Programme und Informationen, die im Schreib/Lesespeicher (RAM) gespeichert sind, auch dann erhalten, wenn die Computer-Konsole ausgeschaltet oder das Modul aus der Konsole herausgenommen wird.

CHARAKTERISTIKA DES MINI-MEMORY-MODULS

- Insgesamt 14 K Bytes Speicherkapazität. Der Speicher besteht aus einem 6 K Byte Grafik-Lese-speicher (GROM), einem 4 K Byte Festwertspeicher (ROM) und einem 4 K Byte RAM. Die GROM- und ROM-residenten Programme ermöglichen zusätzliche wichtige Programm-Entwicklungen. Das RAM stellt zusätzliche Speicherkapazitäten zur Informations- und Programm-speicherung bereit.
- Eine im Modul eingebaute Batterie zur Erhaltung von Informationen und Programmen, die im RAM-Speicher gespeichert sind.
- Zusätzliche Daten. Außer der 4 K Byte RAM-Datei im Mini-Memory-Modul selbst kann das 4 K Byte Segment bei der Verwendung der Speicher-Erweiterungs-Einheit für TI BASIC-Programme benutzt werden.
- Möglichkeit der Abarbeitung von Assembler-Sprachen. Mit dem Mini-Memory-Modul können Assembler-Sprachen-Programme in das Modul selbst oder in die Speicher-Erweiterungs-Einheit eingegeben werden.
- Zusätzliche TI BASIC-Unterprogramme. Mit dem Mini-Memory können durch TI-BASIC-Befehle zusätzliche Unterprogramme abgerufen werden. Diese Unterprogramme ermitteln Werte und führen diese mit.
- Zusätzliche Hilfsprogramm-Routinen. Das Mini-Memory-Modul verfügt über verschiedene Programm-Routinen, die den Zugriff zu den Computer-Ressourcen erlauben; z. B. Schnittstellen zwischen Anwender-Programmen und ROM- und GROM-residenten Programmen, Schnittstellen zwischen Assembler-Sprachen-Programmen und dem TI-BASIC-Interpreter und Zugriffsmöglichkeit zum VDP RAM.
- Fehlersuchprogramm. Das »Easy Bug«-Fehlersuchprogramm, das im Modul integriert ist, stellt ein nützliches Instrument zur Programmentwicklung dar und ermöglicht Ihnen den Zugriff zu den computereigenen Ressourcen um Fehler in Ihren Programmen aufzuspüren und zu beseitigen.

EINSATZMÖGLICHKEITEN

Sie können das RAM des Mini-Memory-Moduls zum Speichern von Daten oder Programmen benutzen. Dieser Speicher ist ein CPU-Speicher und ermöglicht somit einen schnellen Speicher-Zugriff, der bei der Speicherung häufig wiederkehrender Daten in einer Applikation oder der Abarbeitung von Assembler-Sprachen-Programmen von großem Vorteil ist.

Ein häufig benutztes TI-BASIC-Programm kann mit dem Mini-Memory-Modul schneller gespeichert werden als mit einer Cassette oder Diskette.



Normalerweise verliert der Schreib/Lesespeicher (z. B. der »Anwender«-Speicher in der Computer Konsole) seinen Inhalt, wenn die Konsole abgeschaltet wird. Das durch eine Batterie aktivierte Mini-Memory-Modul dagegen erhält seinen Inhalt bei Abschalten der Konsole. Programme oder Daten können im batteriegespeisten RAM gespeichert, die Konsole abgeschaltet und das Modul aus der Konsole entfernt werden. Werden Modul und Konsole wieder in Betrieb genommen, sind die Daten oder Programme abrufbereit.

Achtung! Wenn das Mini-Memory-Modul entfernt oder wieder eingelegt wird, sollte die Computer-Konsole ausgeschaltet sein, um den Verlust oder die Veränderung irgendwelcher, im RAM des Moduls gespeicherter Daten oder Programme zu verhindern.

Eine wichtige Eigenschaft des Mini-Memory-Moduls ist die Abarbeitung von Assemblersprachen-Programmen. Das Modul erlaubt Ihnen die Eingabe Ihrer eigenen Assemblersprachen-Programme für den direkten Zugriff zu den programmierbaren Bauelementen im Computer (z. B.) TMS9900 Mikroprozessor oder TMS9918 Video Display Prozessor).

Durch die Schnittstellen-Ausgänge an der Konsole können Assemblersprachen-Programme auch direkt auf Geräte wie Fernsteuerungen oder Kassettenrekorder Zugriff nehmen.

Außerdem ermöglicht das Modul das Abrufen von Assemblersprachen-Unterprogrammen durch TI-BASIC-Programme. Diese Assemblersprachen-Unterprogramme können Funktionen ausführen, die in BASIC nur ungenügend oder sogar unmöglich sind. Programm-Routinen, die GROM- und ROM-resident und im Modul integriert sind, erlauben eine bequeme Verknüpfung von TI-BASIC-Programmen und Assemblersprachen-Programmen.

Ebenfalls fest in das Modul integriert ist das »Easy-Bug«-Fehlersuchprogramm. Mit »Easy Bug« haben Sie Zugriff zum Speicher und den programmierbaren Bauelementen des Computers. Easy Bug beinhaltet auch Befehle zum Eingeben und Speichern von Bild-Daten auf Kassette.

ALLGEMEINE SPEICHER-INFORMATIONEN

Anmerkung: Das »größer als«-Zeichen (>) im Anwender-Handbuch bedeutet, daß die folgende Nummer eine hexadezimale (Basis 16) Zahl ist.

Das Mini-Modul hat einen Speicher von insgesamt 14 K Bytes, bestehend aus einem 4 K Bytes batteriegetriebenen Schreib/Lesespeicher (RAM), einem 4 K Bytes Festwertspeicher (ROM) und einem 6 K Bytes Grafik-Lesespeicher (GROM).

Eine Anzahl ROM- und GROM-residenter Routinen bieten zusätzlich abrufbare Unterprogramme zu TI-BASIC, die für die Verknüpfung von Assemblersprachen-programmen und TI-BASIC-Programmen nützlich sind.

Anhang A des Anwenderhandbuchs zeigt die Speicher-Organisation der gesamten Speicherkapazität des Computers. Die 4 K Bytes des ROM im Mini-Memory-Modul belegen die Speicherkapazität von Adresse 6000 – 6FFF (oder von 24576 bis 28671). Anhang B zeigt die Einzelheiten der ROM-Organisation im Modul.

Die 4 K Bytes des RAM im Modul belegen die Speicherkapazitäten von Adresse 7000 bis 7FFF (oder von 28672 bis 32767). Anhang C des Anwenderhandbuchs zeigt die Einzelheiten der RAM-Aufteilung, wenn es für Assemblersprachen-Speicherung genutzt wird.

Das GROM belegt Speicherkapazität, die nicht direkt im CPU-Speicher-Adress-Raum angegeben ist.

Anhang E und F des Anwenderhandbuchs beinhalten Informationen über das RAM im Video Display-Prozessor (VDP).



QUELLEN FÜR HINTERGRUND-INFORMATIONEN

Eine Reihe von Referenzen können Ihnen hilfreiche Hintergrund-Informationen geben, wenn Sie beginnen, das Mini-Memory-Modul zu benutzen:

1. Anwenderhandbuch zum TEXAS INSTRUMENTS Home-Computer
2. Anwenderhandbuch EXTENDED BASIC
3. EDITOR/ASSEMBLER Handbuch zum TEXAS INSTRUMENTS Home-Computer zur Erstellung von Assemblersprachen. Dieses Buch vermittelt umfassendes Wissen für die TMS9900-Assembler-Sprache und beschreibt ferner Einzelheiten der inneren Architektur des TI-Home-Computers TI99/4 und TI99/4A.

BENUTZUNG DES ANWENDERHANDBUCHS

Das Anwenderhandbuch setzt voraus, daß Sie bereits Erfahrung im Programmieren mit TI-BASIC haben. Statements, Befehle und Funktionen, die auch in TI-BASIC Verwendung finden, werden nur kurz angesprochen. Vollständige Erläuterungen finden Sie im Anwenderhandbuch zu Ihrem TI-Home-Computer TI99/4 oder TI99/4A.

Die Verwendung des Mini-Memory-Modul zum Aufbau Ihrer eigenen Assemblersprachen-Programme setzt Erfahrung in der TMS9900 Assemblersprachen-Programmierung voraus. Ebenso sollten Sie mit der internen Organisation von Daten- und Dateistrukturen Ihres Home-Computers vertraut sein. Einzelheiten hierzu gibt das Editor/Assembler-Handbuch.

Der Rest des vorliegenden Handbuchs erläutert die verschiedenen Charakteristika des Mini-Memory-Moduls.

Der Abschnitt mit dem Titel »Anwendung des Solid State Software™ Befehls-Moduls« erläutert die verschiedenen Optionen die beim Aktivieren des Mini-Memory-Moduls auf dem Bildschirm erscheinen. Ebenso wird die Wahl der einzelnen Optionen erklärt.

Der Abschnitt »Eingabe und Speichern von TI-BASIC-Dateien« erläutert den Einsatz des Mini-Memory-Moduls als Zugriffselement zu Daten-Dateien und der Abschnitt »Eingabe und Speichern von TI-BASIC-Programmen« beschreibt, wie das Modul zur Eingabe und Speicherung von BASIC-Programm-Dateien verwendet wird.

Unter »Zusätzliche TI-BASIC-Unterprogramme« finden Sie Erläuterungen zu den zusätzlichen Unterprogrammen, die das Mini-Memory-Modul bereitstellt und die zur Verknüpfung von Assemblersprachen-Programmen mit dem Computer-System verwendet werden.

»Eingabe von Assemblersprachen-Programmen« beschreibt das Vorgehen bei der Eingabe von Assemblersprachen-Programmen und Unterprogrammen, und der Abschnitt »System-Hilfs-Routinen« erklärt die Hilfs-Routinen des Mini-Memory-Moduls, die Zugriff zu GROM- und ROM-Routinen der CPU geben.

»Easy Bug«-Fehlersuchprogramm beschreibt die Handhabung des Easy-Bug-Fehlersuchprogramms und die 6 Anhänge beinhalten Informationen über die Speicher-Organisation des Home-Computers.



EINFÜHRUNG

LINE-BY-LINE-ASSEMBLER (Zeilen-Assembler)

Die dem Mini-Memory-Solid State Cartridge-Befehls-Modul beigefügte Cassette enthält einen symbolischen Zeilen-Assembler und ein grafisches Demonstrations-Programm mit der Bezeichnung LINES (Zeilen).

Mit dem Zeilen-Assembler können Sie direkt über die Computer-Tastatur Zeile für Zeile den TMS9900 Assemblersprachen-Quell-Code eingeben. Das LINE-Demonstrations-Programm (nur in Verbindung mit TI99/4A) zeichnet automatisch farbige Linien auf dem Bildschirm.

Wenn das Assembler-Programm in das Mini-Memory-Modul eingegeben wird, wird jedes Quell-Statement sofort in Objekt-Code umgewandelt und an dem Speicherplatz abgespeichert, den Sie mit Ihrem Quell-Code bezeichnet haben. Somit ist das Programm sofort nach beendeter Eingabe und Speicherung von Name und Adresse in der Symbol-Überschrift abrufbereit.

Wichtig! Da der Code bei Eingabe der Zeilen direkt assembliert und gespeichert wird, versichern Sie sich, daß die im Programm angegebenen Speicheradressen verfügbar sind. Sonst wird der Code weder erstellt noch gespeichert.

Obwohl der Assembler jede Instruktion in Maschinen-Code umwandelt, wird ein Teil des Quell-Codes in einem 9-Seiten Text-Buffer zurückgehalten.

Früher eingegebenen Zeilen mit Quell-Code können Sie zur Überprüfung über den Bildschirm abrufen. Hierzu betätigen Sie die Pfeiletasten (Richtungstasten auf – nieder). ↕

LADEN DES ZEILEN-ASSEMBLERS

Sowohl der Zeilen-Assembler als auch das Grafik-Programm LINES werden gleichzeitig von der Kassette in die Computer-Konsole geladen. Hierzu bedient man sich des L (LOAD)-Befehls der »Easy Bug«-Fehlersuchprogramm-Option.

Die nachfolgenden Schritte beschreiben den Lade-Vorgang:

1. Nachdem das Mini-Memory-Modul in die Computer-Konsole eingelegt ist, verbinden Sie Ihren Kassettenrekorder mit der Konsole gemäß den Anweisungen im Anwenderhandbuch.
2. Durch einer beliebigen Taste erscheint das Inhaltsverzeichnis auf dem Bildschirm, aus dem Sie die Mini-Memory-Option auswählen. Wenn die Mini-Memory-Options-Liste erscheint, bereiten Sie, durch Drücken der Taste 3 (REINITIALIZE), den Speicher für die Aufnahme eines neuen Programmes vor. Um zum Titelbild des Computers zurückzukehren, betätigen Sie die Taste QUIT.
3. Nun wird die Assembler-Cassette in den Recorder eingelegt und zurückgespult.
4. Zum Abrufen des Hauptinhaltsverzeichnisses drücken Sie eine beliebige Taste und wählen dann die Easy Bug-Option.
5. Zum Löschen der Easy-Bug-Befehls-Beschreibung vom Bildschirm drücken Sie eine beliebige Taste. Um den Ladeprozeß zu starten ist erst L, dann ENTER zu drücken. Nun erscheinen auf dem Bildschirm Anweisungen für die weiteren Maßnahmen zur Programm-Eingabe.
6. Nachdem das Assembler-Programm eingegeben ist, kehren Sie durch QUIT zum Titelbild zurück. Durch Drücken einer beliebigen Taste rufen Sie das Hauptinhaltsverzeichnis ab, aus dem Sie die Mini-Memory-Option auswählen.
7. Nach Erscheinen der Mini-Memory-Liste starten Sie die RUN-Funktion durch Betätigen der Taste 2. Der Bildschirm wird frei und das Programm ruft den Namen des Programms ab, das Sie abarbeiten möchten.



- A) Wenn Sie das LINES-Programm abrufen möchten (nur auf dem TI 99/4A möglich) tippen Sie LINES und drücken die ENTER-Taste. Das Programm zeichnet nun farbige Linien auf den Bildschirm. Mit der C-Taste können Sie die Farbe der zuletzt erschienenen Zeile »einfrieren« und alle nachfolgenden Linien erscheinen in dieser Farbe. Nochmaliges Betätigen der C-Taste löscht diesen Effekt. Um das Programm zu stoppen wird QUIT betätigt.
- B) Wenn Sie ein neues Assemblersprachen-Programm entwickeln wollen, schreiben Sie NEW und drücken die Taste ENTER. Das Programm geht in den Assembler über, löscht die Symbol-Übersicht (mehr darüber später) und wartet auf die Eingabe Ihrer ersten Programmzeile.
- C) Wenn Sie ein bereits früher begonnenes Programm weiterschreiben wollen, tippen Sie OLD, dann ENTER. Die alte Symbol-Überschrift ist erhalten und der Bildschirm zeigt den nächsten Speicherplatz zur Fortsetzung Ihres Programms.

Anmerkung: Ihr Mini-Memory-Modul kann bereits mit dem Assembler- und dem LINES-Programm versehen sein. Um dies zu prüfen, wählen Sie die Mini-Memory-RUN-Option und geben nach Aufforderung den entsprechenden Programm-Namen ein. Ist das Programm vorhanden, läuft es sofort ab. Fehlt das Programm, zeigt der Bildschirm »Program not found« (Programm nicht gefunden).

Durch den M-Befehl des EASY BUG können Sie auch den LINES Code überprüfen. Nach Eingabe von M7CD6 und konstantem Drücken von ENTER können Sie jede Code-Zeile nachprüfen, ohne irgendetwelche Daten zu verändern.

Wichtig! Wenn Sie ein Programm eingeben und assemblieren, kann die Symbol-Übersicht einen Teil des im Mini-Memory-Modul gespeicherten LINES-Programms überschreiben. Wenn Sie dann LINES wieder abrufen wollen, laden Sie das Programm einfach noch einmal von der Kassette in das Modul über.

ALLEMBLER-SYNTAX

Jede Zeile (oder Datensatz) Ihres Quell-Programms besteht aus 4 Teilen, die als Felder bezeichnet werden. Diese Felder müssen – wenn vorhanden (einige sind wahlweise) – in der vom Assembler vorgegebenen Reihenfolge und Syntax erscheinen.

In diesem Handbuch werden für Syntax-Definitionen und Maschinenbefehle folgende Richtlinien benutzt:

- Angaben in Großbuchstaben, incl. spezieller Zeichen, müssen genau nach Vorgabe eingegeben werden.
- Angaben in eckigen Klammern ([]) können wahlweise gemacht werden.
- Angaben in Winkelklammern (< >) sind benötigte Felder.
- Ein kleines b bezeichnet einen Zwischenraum.
- Ein kleines b, gefolgt von 3 Punkten (b ...) bezeichnet einen oder mehrere Zwischenräume.
- Die allgemeine Syntax für eine Assembler-Anweisung lautet wie folgt:
[label] b ... <opcode> b [,operand] [b ... comment]

Das Adressfeld verlangt entweder 1 Zwischenraum (wenn keine Adresse vorhanden ist) oder bis zu zwei Zeichen. Das erste Zeichen muß alphabetisch, das zweite Zeichen (wenn vorhanden) kann alphanumerisch sein. Der Adresse folgen 1 oder 2 Zwischenräume. Wenn Sie keine Adresse eingeben, bewegt sich der Cursor nach Betätigen der Leertaste automatisch an den Anfang des Opcode-Feldes.



Das Opcode-Feld enthält den Betriebs-Code der Aufgabe, die durch die Quell-Anweisung ausgeführt werden soll. Das Feld hat 1 – 4 alphabetische Zeichen, wie z. B. A für ADD oder AORG für Absolute Origin-Anweisung, gefolgt von einem Zwischenraum.

Das Operanden-Feld hat 1 oder 2 Operanden, entsprechend der Instruktionen. Das Operanden-Feld hat keine Zwischenräume und mehrere Operanden werden durch Kommata getrennt. Durch Betätigen der Leertaste (Cursor rückt zum Comment-Feld vor) oder ENTER (Ende der Zeile) wird das Operanden-Feld abgeschlossen. Wenn eine Instruktion keinen Operanden enthält, fällt das Operanden-Feld weg.

Das Comment-Feld kann jedes Zeichen beinhalten und es läuft solange bis ENTER betätigt wird um die Zeile zu beenden.

BEISPIELE:

XY MOV R1, VP
Z S R1, R2

RL in VP speichern
Differenz berechnen

Der Zeilen-Assembler bezeichnet gewisse Symbole im voraus. Wenn ein Operand das Dollar-Zeichen (\$) als Anfangszeichen beinhaltet, wird angenommen, daß es sich auf den Inhalt des Befehlszählers bezieht. So werden z. B. die Statements auf Zählerstelle 7D00

JMP und JMP 7D08

als gleich angesehen.

Zum Abruf von Register-Operanden kann das Symbol R verwendet werden, gefolgt von einer Dezimal-Zahl. Also

MOV R1, R15 oder MOV 1, 15

Anmerkung: Das Default-Nummern-System des Zeilen-Assemblers ist dezimal; hexadezimale Zahlen werden mit dem »größer als«-Zeichen (>) bezeichnet.



INTRODUCTION

Le module *Solid State Cartridge* »Mini Mémoire« dote l'Ordinateur Familial d'une mémoire de stockage supplémentaire et offre à l'utilisateur la possibilité de développer des programmes plus facilement. Ce module contient une pile permettant de conserver dans sa mémoire vive (RAM) des données ou programmes, même lorsqu'on le retire de l'Ordinateur Familial ou que l'on éteint ce dernier. Ce module présente les caractéristiques suivantes:

- 14Ko de mémoire au total: 6Ko de mémoire morte-graphique (GROM), 4Ko de mémoire morte (ROM), et 4Ko de mémoire vive (RAM). Les 10Ko de mémoire morte contiennent des utilitaires de programmation, utilisables à partir de TI BASIC. Les 4Ko de mémoire vive constituent une mémoire de masse supplémentaire.
- une pile située à l'intérieur du module permet de conserver des données dans les 4Ko de mémoire vive.
- en plus de ces 4Ko de mémoire vive, 24 des 32Ko de l'extension mémoire peuvent être utilisés à partir du TI-BASIC dans le cas où la carte d'Extension Mémoire* est insérée dans le Système d'Extension Périphérique*.
- des programmes-objet écrits en langage ASSEMBLEUR peuvent être chargés soit dans la »Mini-Mémoire«, soit dans l'Extension Mémoire (si elle est connectée).
- la »Mini-Mémoire permet, à l'aide d'instructions TI-BASIC, d'appeler certains sous-programmes normalement non accessibles et, en particulier, d'utiliser les commandes PEEK et POKE.
- différents utilitaires permettant d'exploiter les ressources du TI99/4A deviennent disponibles: interfaçage de programmes-utilisateur avec des programmes résidant en GROM ou ROM et de programmes ASSEMBLEUR avec l'interpréteur TI-BASIC, accès à la MEV (RAM) du microprocesseur de gestion écran (VDP).
- le programme de recherche d'erreurs EASY BUG donne accès à toutes les possibilités du TI99/4A et permet de déboguer plus efficacement les programmes.

* accessoire vendu séparément

APPLICATIONS

La mémoire vive (RAM) offerte par la Mini Mémoire est une mémoire dite »CPU«, à accès très rapide, pouvant être utilisée pour stocker des programmes en ASSEMBLEUR.

- Plutôt que d'utiliser une disquette ou une cassette, on peut stocker un programme en TI-BASIC sur la Mini Mémoire de façon à y avoir accès immédiatement.
- Lorsqu'on éteint l'ordinateur, tout le contenu de la mémoire-utilisateur (RAM) est effacé. Néanmoins, grâce à sa pile incorporée, la Mini Mémoire conserve dans tous les cas l'information contenue dans ses 4Ko de mémoire vive. Une fois que des programmes ou des données sont stockés dans la Mini Mémoire, on peut débrancher l'ordinateur et enlever le module sans qu'il y ait perte d'information. De même, lorsque la Mini Mémoire est réintroduite dans l'ordinateur et que celui-ci est rebranché, cette information est immédiatement accessible.

Attention! Avant d'enlever ou de réintroduire le module Mini-Mémoire, vérifier que l'ordinateur n'est pas sous tension de façon à éviter toute perte d'information.



La Mini-Mémoire permet aussi d'utiliser le langage ASSEMBLEUR. Il est possible de charger son propre programme en ASSEMBLEUR et d'avoir un accès direct au microprocesseur central (CPU) TMS9900 ou au microprocesseur vidéo (VDP) TMS9918. De plus, les manettes de contrôle ou un magnétophone à cassette peuvent être adressés directement par des programmes en ASSEMBLEUR et ce, par l'intermédiaire de leurs ports d'interface qui se trouvent dans l'ordinateur.

- Il est aussi possible d'appeler des sous-programmes ASSEMBLEUR à partir de programmes TI-BASIC, et d'avoir ainsi accès à des fonctions et des possibilités normalement inaccessibles en BASIC.

Les programmes résidant dans les 6Ko de GROM et 4Ko de ROM réalisent l'interface entre ces sous-programmes et un programme BASIC.

INFORMATIONS GENERALES

Remarque: le signe »plus grand que« (>) utilisé dans ce manuel d'utilisation indique que le nombre qui suit est exprimé en base 16 (nombre hexadécimal).

L'appendice A du manuel d'utilisation indique la façon dont la mémoire de l'ordinateur est agencée. Les 4Ko de mémoire morte (ROM) de la Mini-Mémoire occupe l'espace délimité par les étiquettes >6000 (24576) et >6FFF (28671). L'appendice B donne de plus amples détails sur cet agencement.

Les 4Ko de mémoire vive (RAM) sont délimités par les étiquettes >7000 (28672) et >7FFF (32767). L'appendice C (respectivement D) donne de plus amples détails sur l'agencement de cette mémoire lorsque celle-ci est utilisée pour des fichiers en BASIC (respectivement en ASSEMBLEUR).

La partie de mémoire correspondant à la GROM n'est pas adressable directement par l'unité centrale (CPU).

Les appendices E et F concernent la mémoire vive (RAM) du processeur d'image (Vidéo Display Processor).

QU' TROUVER DES INFORMATIONS SUPPLÉMENTAIRES?

Plusieurs documents peuvent aider à utiliser la Mini-Mémoire:

1. Manuel d'utilisation du TI99/4A
2. Manuel d'utilisation de module »EXTENDED BASIC«
3. Manuel d'utilisation du module »EDITOR/ASSEMBLER« dans le cas où l'on veut programmer en Assembleur. Ce manuel donne toutes les informations possibles concernant l'Assembleur TMS9900 et l'architecture interne du TI 99/4A.

COMMENT UTILISER CE MANUEL ?

Ce manuel d'utilisation suppose que le lecteur a déjà des connaissances en TI-BASIC; les instructions, commandes et fonctions qui sont les mêmes en TI-BASIC ne sont décrites que très brièvement. Pour en obtenir une description complète, se reporter au manuel d'utilisation du TI 99/4A.

Il faut avoir au préalable une certaine connaissance de l'Assembleur TMS9900 et de l'organisation interne des données et fichiers utilisés par le TI 99/4A pour utiliser la Mini-Mémoire en programmation en Assembleur. Pour de plus amples détails sur ce sujet, se reporter au manuel d'utilisation du module »Editor/Assembler«.

Le paragraphe intitulé »Using the Solid State Software Command Modul«™ décrit les différentes options qui deviennent disponibles lorsqu'on introduit la Mini-Mémoire dans le TI 99/4A.



Le paragraphe intitulé «Loading and Saving TI BASIC files (respectivement subprograms)» montre comment utiliser la Mini-Mémoire pour avoir accès à des fichiers de données (respectivement, de programmes).

Le paragraphe intitulé «**Additional TI-BASIC subprograms**» décrit les sous-programmes fournis par la Mini-Mémoire et permettant l'interface avec les programmes en Assembleur et l'ordinateur.

Le paragraphe «**Loading assembly language Programs**» indique comment charger les programmes et sous-programmes écrits en Assembleur.

Le paragraphe «**System Utility Routines**» décrit les programmes utilitaires de la Mini-Mémoire qui permettent l'accès aux routines de la ROM et de la GROM de l'unité centrale.

Le paragraphe «**Easy Bug Debugger**» explique le fonctionnement du programme de recherche de défauts (debugger).

Les 6 appendices décrivent l'organisation interne de la mémoire du TI 99/4A.

INTRODUCTION:

La cassette fournie avec la Mini-Mémoire contient un assembleur «ligne par ligne» ainsi qu'un programme de démonstration graphique appelé LINES. Cet assembleur permet l'accès, à partir du clavier du TI 99/4A, au code-source de l'assembleur TMS 9900 et ce, une ligne à la fois. Quant au programme de démonstration, il dessine automatiquement sur l'écran des droites de couleurs différentes.

Une fois que le programme ASSEMBLER est chargé dans la Mini-Mémoire, chaque instruction-source introduite est immédiatement assemblée en code-objet et stockée en mémoire à l'endroit précisé dans le code-source. Ainsi, dès qu'il est introduit dans l'ordinateur et que son nom et son adresse sont stockés dans la table des symboles, le programme peut être exécuté.

Remarque importante: Puisque le code est assemblé et stocké directement à chaque fois qu'une ligne est introduite, il faut s'assurer au préalable que les adresses de mémoire spécifiées dans le programme sont valides. Sinon, aucun code ne pourra être généré ou stocké.

Bien que l'Assembleur transforme en code-machine chaque instruction dès qu'elle est introduite, une partie du code-source est conservé dans une mémoire-tampon (9 pages). Il est possible, grâce aux touches ↑ (FCTN E) et ↓ (FCTN X), de réafficher sur l'écran des lignes de code-source précédemment introduites.

Ce manuel décrit les caractéristiques de l'ASSEMBLEUR LIGNE/LIGNE (**LINE BY LINE ASSEMBLER**) et permet aussi d'utiliser le programme de démonstration LINES (Voir le Paragraphe «Chargement de l'Assembleur»). Il est recommandé de consulter le Manuel d'Utilisation du module EDITOR/ASSEMBLER pour obtenir tout renseignement concernant ce langage.



CHARGEMENT DE L'ASSEMBLEUR LIGNE/LIGNE

L'Assembleur ligne/ligne et le programme de démonstration graphique peuvent être chargés en même temps à partir de la cassette à l'aide de la commande L (LOAD), une fois sélectionnée l'option 2 (EASY BUG) de la Mini-Mémoire. Suivre les étapes suivantes pour charger le contenu de la cassette:

1. Une fois que la Mini-Mémoire est enfilée dans l'ordinateur, connecter le magnétophone à cassette à ce dernier (voir le Manuel d'Utilisation du TI 99/4A).
 2. Appuyer sur une touche quelconque pour faire apparaître le tableau de sélection et appuyer sur 3 (option MINI-MEMORY). Lorsque la liste des options de la Mini-Mémoire est affiché, appuyer sur 3 (option REINITIALIZE) pour préparer la mémoire à recevoir un nouveau programme. Appuyer ensuite sur QUIT (FCTN=) pour revenir à l'image d'introduction initiale.
 3. Placer la cassette du LINE-BY-LINE ASSEMBLER dans le magnétophone et la rembobiner.
 4. Appuyer sur une touche quelconque pour faire apparaître le tableau de sélection et sur 2 pour choisir l'option EASY BUG.
 5. Une fois que l'écran décrivant l'option EASY BUG apparaît, appuyer sur une touche quelconque pour effacer l'écran. Taper ensuite L et appuyer sur ENTER pour initier le processus de chargement. A partir de ce moment-là, des instructions apparaissent en clair sur l'écran et guident l'utilisateur tout au long de la procédure.
 6. Une fois le programme ASSEMBLER chargé, appuyer sur QUIT (FCTN=) pour revenir à l'image d'introduction initiale et appuyer sur 3 pour choisir l'option MINI MEMORY.
 7. Quand le tableau de sélection des options de MINI MEMORY apparaît, appuyer sur 2 pour choisir l'option RUN (exécution).
- A) Pour exécuter le programme LINES, taper LINES une fois que le message »PROGRAM NAME?« est affiché et appuyer sur ENTER. Ce programme affiche un dessin multicolore composé à partir de droites. En appuyant sur C, on peut figer la couleur de la dernière droite dessinée de façon à ce que toutes les droites suivantes aient la même couleur. En réappuyant sur C, on supprime cet effet. Pour interrompre le programme, appuyer sur QUIT (FCTN=).
- B) Pour créer un programme en Assembleur, taper NEW et appuyer sur ENTER. L'Assembleur devient alors accessible, la table des symboles est effacée et la première ligne de programme peut être introduite.
- C) Pour continuer un programme introduit antérieurement, taper OLD et appuyer sur ENTER. La table des symboles existante est conservée et l'adresse-mémoire suivante est affichée sur l'écran. A partir de ce moment-là, le programme peut être complété.

Remarques: Il se peut que le module MINI-MEMORY contienne déjà le programme Assembleur et le programme Assembleur et le programme LINES. En guise de test, choisir l'option RUN de la MINI-MEMOIRE et introduire le nom du programme lorsque le caractère - guide (prompteur) apparaît. Si le programme est déjà chargé, son exécution commence tout de suite. Sinon, le message »PROGRAM NOT FOUND« (Programme non trouvé) est affiché sur l'écran.

La commande M de EASY BUG permet de voir le programme - code de LINES. Taper M7CD6 et appuyer à chaque fois sur ENTER pour visualiser une ligne différente de ce programme, sans pour cela en modifier les données.



Important Lorsqu'un programme est introduit et assemblé, il se peut que la table des symboles »ef-face« une partie du programme LINES stocké dans la Mini-Mémoire: Pour exécuter ce programme, il suffit dans ce cas de le recharger (à partir de la cassette) dans la Mini-Mémoire.

SYNTAXE D'ASSEMBLAGE

Chaque ligne ou enregistrement d'un programme-source est constituée de 4 secteurs. Lorsqu'ils sont utilisés certains sont facultatifs, ces secteurs doivent être dans un ordre et selon un format bien déterminés. Les conventions suivantes sont utilisées dans ce manuel:

- tout ce qui apparait en lettres majuscules, y compris les caractères spéciaux, doivent être introduits tels quels dans l'ordinateur.
- tout ce qui est entre [] est facultatif
- tout ce qui est entre <> représente des secteurs indispensables.
- un b minuscule représente un espace
- un b minuscule suivi de 3 points (...) représente un ou plusieurs espaces

La syntaxe utilisée pour les instructions en Assembleur est la suivante:

[LABEL] b... <OPCODE> b [OPERAND] [,OPERAND] [b... COMMENT]

- le secteur LABEL (étiquette) comporte soit un espace (pas d'étiquette), soit un maximum de 2 caractères dont le premier doit être alphabétique et le second, le cas échéant, alphanumérique. Ce secteur est suivi de un ou plusieurs espaces. Si aucune étiquette n'est utilisée, le fait d'appuyer sur la barre d'espacement fait automatiquement avancer le curseur au début du secteur OPCODE.
- le secteur OPCODE (code-opération) contient le code de l'opération qui doit être exécutée par l'instruction source. Il contient de 1 jusqu'à 4 caractères (tels que A pour addition ou AORG pour Origine Absolute) suivi(s) d'un espace.
- le secteur OPERAND (opérande) contient 1 ou 2 opérandes selon l'instruction utilisée. Noter que ce secteur ne contient pas d'espace et que différents opérandes sont séparés entre eux par une virgule. On termine un secteur OPERAND en appuyant sur la barre d'espacement (ce qui fait avancer le curseur au secteur COMMENT) ou sur ENTER (ce qui indique la fin de la ligne). Lorsqu'une instruction n'a pas d'opérande, le secteur correspondant est simplement omis.
- lorsqu'il est utilisé, le secteur COMMENT (commentaire) peut contenir un caractère quelconque et se prolonge jusqu' à ce que la touche ENTER soit appuyée pour signifier »fin de ligne«.

EXEMPLES:

XY MOV R1, VP : Stocke R1 dans VP
Z S R1, R2 : Calcule la différence

L'Assembleur Ligne/ligne prédéfinit certains symboles. Lorsque le premier caractère d'un opérande est le signe dollar (\$), cet opérande se rapporte forcément à une adresse-mémoire. Par exemple, à l'adresse >7D00, les instructions JMP \$ + 8 et JMP >7D08 sont équivalentes.



On peut utiliser le symbole R, suivi d'un nombre décimal, pour appeler des opérandes de registre. Ainsi, les instructions

MOV R1, R15 et MOV 1,15

sont équivalentes.

Remarque: Le Système numérique utilisé par défaut par l'Assembleur ligne/ligne est le système décimal; les nombres hexadécimaux sont précédés du signe >.



INTRODUZIONE

ASSEMBLER ISTRUZIONE PER ISTRUZIONE

La cassetta acclusa al modulo di comando Mini Memoria contiene un Assembler simbolico "istruzione per istruzione" e un programma per la dimostrazione di grafici denominato LINES. L'assembler "istruzione per istruzione" vi consente di digitare un programma "SORGENTE" nel linguaggio Assembly TMS9900, una istruzione alla volta, direttamente dalla tastiera del computer. Il programma di dimostrazione LINES (operante solo sul TI-99/4A) traccia automaticamente sullo schermo le righe colorate.

Quando il programma Assembler è caricato nel modulo di Mini Memoria, qualsiasi istruzione "SORGENTE" Voi digitate viene immediatamente compilata nel codice oggetto e caricata nelle locazioni di memoria specificate dal Vostro codice sorgente.

Pertanto, non appena avrete completato l'inserimento del Vostro programma e collocato il suo nome ed indirizzo nella tavola dei simboli, questo è pronto per essere eseguito.

NOTA IMPORTANTE: Poiché il programma è compilato e memorizzato direttamente, nel momento in cui Voi digitate una istruzione siate sicuri che gli indirizzi di memoria specificati nel programma siano disponibili.

In caso contrario, nessun codice sarà prodotto o memorizzato.

Sebbene l'Assembler converta ogni istruzione digitata nel codice della macchina, appena queste sono digitate alcune sono memorizzate in una area di 9 pagine di video.

Potete controllare sullo schermo per rivedere preventivamente le istruzioni digitate premendo i tasti contrassegnati dalla freccia in su ed giù.

Questo manuale evidenzia le caratteristiche dell'Assembler "istruzione per istruzione", presupponendo che comosciate già il linguaggio di programmazione Assembler. Per una completa guida di riferimento sul linguaggio Assembly TMS9900, consultate il manuale Editor/Assembler. Vengono inoltre fornite in questo manuale (consultate: "utilizzo Assembler" istruzione per istruzione), istruzioni per lanciare il programma di dimostrazione LINES.

Nota: Il programma LINES opera soltanto sul TI-99/4A Home Computer, che contiene il potente processore grafico richiesto dal programma.

UTILIZZO DELL'ASSEMBLER "ISTRUZIONE PER ISTRUZIONE"

Sia l'Assembler "istruzione per istruzione" che il programma di grafica LINES sono caricati nello stesso momento dalla cassetta tramite il comando L (CARICA) dell'opzione EASY BUG. Le seguenti fasi descrivono il procedimento di caricamento.

1. Con il modulo di Mini Memoria installato nel computer collegate il Vostro registratore alla consolle, come descritto nel manuale d'uso.
2. Premete qualsiasi tasto per far apparire la lista di selezione principale selezionate l'opzione della Mini Memoria Quando la lista di selezione della Mini Memoria appare, premete 3 "Reinizializzare" per predisporre la memoria al caricamento di un nuovo programma quindi premete QUIT per riportarsi al titolo principale del computer.
3. Inserite la cassetta dell'Assembler nel registratore, e riavvolgetela.
4. Premete qualsiasi tasto per far apparire la lista di selezione principale, quindi selezionate l'opzione EASY BUG.



5. Quando appare sullo schermo la descrizione del comando EAGY BUG, premete qualsiasi tasto per ripulire lo schermo. Quindi premete ENTER per iniziare l'operazione di caricamento. Da questo momento in poi, lo schermo visualizza istruzioni per aiutarvi in questa procedura. Seguite queste istruzioni per caricare il programma.
6. Dopo che il programma Assembler è stato caricato, premete QUIT per ritornare al titolo principale. Quindi premete un tasto qualsiasi per passare alla lista di selezione principale scegliete l'opzione Mini Memoria.
7. Quando appare la lista di selezione dalla Mini Memoria premete 2 per selezionare RUN. Lo schermo si pulisce ed il programma Vi suggerisce il nome del programma che Volete lanciare.
- A. Se volete lanciare il programma LINES (a disposizione solo sul TI-99/4A), battete LINES e premete ENTER. Il programma traccia delle linee colorate sullo schermo. Se premete il tasto C, potete "congelare" il colore della riga corrente, e tutte le righe susseguenti saranno di quel colore. Premendo C di nuovo cancella l'effetto del singolo colore. per fermare il programma, premete QUIT.
- B. Se volete creare un nuovo programma in linguaggio Assembler battete New e premete ENTER. Il programma carica l'Assembler, pulisce la Tavola dei Simboli ed attende che digitiate la Vostra prima riga di programma.
- C. Se volete continuare a digitare un programma che avevate iniziato precedentemente, battete OLD e premete ENTER.
La vecchia Tavola dei Simboli viene trattenuta, e lo schermo visualizza la prossima collocazione di memoria, pronta per continuare il Vostro programma.

Nota: Il Vostro modulo di Mini Memoria può ancora essere caricato con l'Assembler ed i programmi LINES. Per provare, selezionate l'opzione RUN Mini Memoria e digitate il nome relativo al programma. Se il programma è presente viene lasciato immediatamente. Se il programma non è stato caricato nel modulo il video riporta "PROGRAM NOT FOUND".

Potete anche rivedere le istruzioni del programma LINES mediante il comando M nell'EASYBUG. Digitate soltanto M7CD6 e tenete premuto ENTER per rivedere ogni riga di programma senza cambiare i dati.

NOTA IMPORTANTE: Quando digitate e compilate un programma, la tavola dei simboli può scrivere sopra il programma LINES memorizzato nel modulo di Mini memoria. Quando volete lanciare nuovamente il LINES, caricate di nuovo il programma dalla cassetta nel modulo.

SINTASSI DELL'ASSEMBLER

Ogni istruzione (o record) del vostro programma sorgente consiste di 4 parti dette campi. Questi campi, se presenti (alcuni sono opzionali), devono apparire nell'ordine e formato (sintassi) richiesti dall'Assembler. In questo manuale, le seguenti convenzioni vengono impiegate nelle definizioni di sintassi per le istruzioni macchina.

- Voci in maiuscolo, che includono speciali caratteri devono essere digitati esattamente come spiegato.
- Voci tra parentesi ([]) sono opzionali.
- Voci tra parentesi (< >) sono campi richiesti.



- Una lettera minuscola b indica un singolo spazio.
- Una lettera minuscola b seguita da tre punti (b...) indica uno o più spazi.

la sintassi generale per una istruzione di Assembler é la seguente.
[Label] b...<opcode>b[operando][,operando][b...commento]

Il campo "label" richiede uno spazio (quando non vi é label) oppure uno o due caratteri. Il primo carattere deve essere alfabetico; il secondo carattere, se presente, può essere alfanumerico. Il "Label" é seguito da uno o più spazi. Se non battete un "label", premendo la barra "SPAZIO" si mette in movimento il cursore automaticamente verso l'inizio del campo opcode.

Il campo opcode contiene il codice operazione del lavoro da svolgere con l'istruzione sorgente. Il campo consiste da 1 a 4 caratteri alfabetici, quali A per Add (addizionare) oppure Aore per l'istruzione di INDIRIZZO ASSOLUTO, seguito da un singolo spazio.

Gli operandi in una istruzione possono essere uno o due a seconda delle particolarità dell'istruzione stessa.

Notate che il campo "operando" non ha spazio bianchi nel suo interno, e operandi multipli sono separati da virgole.

Il campo "operando" viene chiuso premendo o la barra SPACE (il cursore avanza verso il campo commento) o ENTER (vuol dire fine della riga). Se un'istruzione non ha alcun operando, il campo "operando" viene ommesso.

Il campo "commento", se impiegato, può includere qualsiasi carattere, e continua fino a che voi non premete ENTER per terminare la riga.

Esempi:

XY MOV R1, EVP	Salve R1 in VP.
Z S R1, R2	Calcolate la differenza

L'Assembler "istruzione per istruzione" predefinisce alcuni simboli. Quando un operando include un simbolo di dollaro (\$) come un carattere iniziale, viene preso in considerazione per fare riferimento ai contenuti della locazione del contatore. Ad esempio, nella localizzazione >7D00, lo statements

e JMP \$+B
JMB >7D08

sono considerati equivalenti. per richiamare dei registri negli operandi, potete usare il simbolo R, seguito da un numero decimale. Pertanto,

e MOV R1, R15
MOV 1, 15

sono equivalenti.

Nota: Il sistema Numerico utilizzato dell'Assembler "istruzione per istruzione" é il decimale; numeri esadecimali vengono indicati con il prefisso >.



MINI MEMORIA SOFTWARE ALLO STATO SOLIDO

Il Modulo di Comando del Software allo stato solido aumenta la versatilità del vostro Home Computer Texas Instruments TI-99/4A mettendo a disposizione una memoria aggiuntiva per il sistema e per importanti supporti per lo sviluppo dei programmi. Inoltre, il modulo contiene una batteria interna che mantiene i programmi e i dati memorizzati nel modulo RAM, anche quando la consolle del computer è spenta, o il modulo viene rimosso dalla consolle stessa.

CARATTERISTICHE DEL MODULO »MINI MEMORIA«:

- Un totale di 14K bytes di memoria. Questa memoria consiste in 6K bytes di memoria grafica di sola lettura (GROM), 4K bytes di memoria di sola lettura (ROM) e 4K bytes di RAM. I programmi che si trovano nella GROM e nella ROM mettono a disposizione supporti aggiuntivi per sviluppo di programmi importanti. La RAM fornisce uno spazio di memoria addizionale per la memorizzazione dei dati e del programma.
- Una batteria interna al modulo per conservare dati o programmi caricati nella memoria RAM.
- Files addizionali. Accanto ai 4K bytes RAM nel modulo »Mini Memoria«, il segmento di 24K bytes dell'unità di Espansione della Memoria, se inserita, può essere usato dai programmi TI BASIC.
- Capacità del linguaggio Assembler. Con il modulo »Mini Memoria«, i programmi in linguaggio assembler possono essere caricati entro il modulo stesso o nell'Unità di Espansione di Memoria, se inserita.
- Sottoprogrammi addizionali TI BASIC. Con il modulo Mini Memoria, molti sottoprogrammi addizionali possono essere chiamati dalle istruzioni del TI BASIC. Questi sottoprogrammi abilitano i valori PEEK e POKE.
- Routine di utilità aggiuntive. Il modulo Mini Memoria comprende parecchie routine di programma che permettono di accedere alle risorse del computer; ad esempio, programmi d'interfaccia dei programmi utente con i programmi ROM- e GROM- ivi presenti, programmi di interfaccia del linguaggio assembler con il traduttore TI Basic, e che accede al VDP RAM.
- Un programma in grado di eliminare gli errori. IL programma EASYBUG è uno strumento di sviluppo-programma con il quale potete accedere alle risorse interne del computer ed eliminare gli inconvenienti del vostro programma.

APPLICAZIONI

Potete usare la RAM nel modulo Mini Memoria per accumulare dati o programmi. Questa memoria è una »memoria CPU«, memoria di accesso veloce. Potete trarre vantaggio nell'usare questa memoria veloce per immagazzinare e recuperare i dati che vengono usati frequentemente in un'applicazione o per accumulare programmi in linguaggio assembler che debbono fornire calcoli rapidi.

Un programma TI BASIC che usate frequentemente può essere immagazzinato nel modulo Mini Memoria, piuttosto che su di una cassetta o dischetto, per un caricamento veloce.

Generalmente, la RAM: Memory Random Access (ad esempio, la memoria dell'»operatore« nella consolle del computer) perde i suoi contenuti quando la consolle è spenta. La batteria attiva, nel modulo Mini Memoria, consente di ritenere i contenuti della memoria stessa quando spegnete la consolle. I programmi o dati possono essere accumulati nel pacchetto: batteria-RAM e la consolle può essere poi spenta ed il modulo rimosso dalla consolle stessa. Quando reinserite il modulo ed accendete la consolle, i vostri dati o programma sono pronti per l'uso.



AVVERTENZA

Quando disinerite o inserite il modulo di Mini Memoria, la consolle del computer dovrà essere spenta, per prevenire la possibilità di alterare qualsiasi dato o programma accumulati nella RAM del modulo.

Un'importante caratteristica del modulo Mini Memoria è la sua capacità di integrare i programmi in linguaggio Assembler. Il modulo vi consente di caricare i vostri programmi in linguaggio Assembler per un accesso diretto ai componenti programmabili nel computer (quali il microprocessor TMS9900 oppure il TMS9918 processor video display). I programmi in linguaggio Assembler possono anche accedere direttamente a periferiche quali i comandi a distanza o registratori di cassette attraverso le porte di interfaccie presenti sulla consolle.

Inoltre, il modulo rende possibile chiamare dai programmi TI BASIC le subroutine in linguaggio Assembler. Queste subroutine possono fornire delle funzioni che sarebbero inefficienti o impossibili da integrare nel BASIC. Le routines dei programmi residenti nella GROM e ROM del modulo permettono una conveniente interfaccia tra i programmi TI BASIC ed i programmi in linguaggio assembler.

E' incluso inoltre nel modulo il programma EASYBUG. Con l'EASYBUG, potete accedere alla memoria e ai componenti programmabili del computer, L'EASYBUG include anche comandi per il caricamento e la registrazione di dati immagine-memoria sulla cassetta.

INFORMAZIONI GENERALI SULLA MEMORIA

Nota: Nella guida per l'operatore, il simbolo (¹⁶) indica che il numero seguente è un numero esadecimale (base 16).

Il modulo Mini Memoria contiene un totale di 14K bytes di memoria che consistono in: 4K bytes di (RAM), 4K bytes di ROM, e 6K bytes di GROM, Risiedono nella, ROM e GROM un numero di routine che forniscono sottoprogrammi addizionali che possono essere chiamati con il TI BASIC e che sono utili per i programmi in linguaggio Assembler.

L'appendice A mostra l'organizzazione di memoria per l'intero spazio di memoria del computer. I 4K bytes della ROM del modulo di Mini Memoria occupano lo spazio di memoria dall'indirizzo >6000, a >6FFF (o da 24576 a 28671). L'appendice B illustra i dettagli dell'organizzazione della ROM.

I 4K bytes della RAM nel modulo occupano lo spazio memoria dagli indirizzi >7000 a >7FFF (o da 28672 a 32767). L'appendice C indica i dettagli dell'organizzazione della RAM quando viene usata per files TI BASIC. L'appendice D descrive come la RAM è strutturata per la memorizzazione del linguaggio Assembler. La GROM occupa lo spazio di memoria che non e' direttamente nello stesso spazio di indirizzo di memoria CPU. Le appendici E e F contengono informazioni riguardanti la RAM nel Processor Video Display (V.D.P.)

INFORMAZIONI PRELIMINARI

Prima di accingervi ad usare il modulo di Mini Memoria, vi suggeriamo di leggere con attenzione:

- Manuale d'uso dell'Home Computer della Texas Instruments.
- Manuale del BASIC Esteso dell'Home Computer della Texas Instruments.
- Se state creando dei programmi in linguaggio Assembler, il manuale dell'Editor/Assembler Home Computer Texas Instruments. Questo libro è una guida/manuale completa per il linguaggio TMS9900 assembly e contiene inoltre dettagli circa la composizione interna dell'Home Computer TI-99/4 e del TI-99/4A.



COME USARE QUESTO PROGRAMMA

Il manuale per l'operatore presume che stiate ancora sperimentando la programmazione con il TI BASIC. Istruzioni, comandi, e funzioni che sono le medesime del TI BASIC vengono solo brevemente accennate. Per una completa descrizione consultate il manuale d'uso incluso nel vostro Home Computer TI 99/4A.

Se avete intenzione di usare il modulo di Mini Memoria per creare i vostri programmi in linguaggio Assembler, si presuppone che abbiate sperimentato il linguaggio di programmazione assembler TMS9900 e che siate ormai pratici di organizzazione interna di dati e strutture di file impiegate nell'Home Computer. Per una completa trattazione di questi argomenti, consultate il manuale Editor/Assembler.

Modulo di Mini Memoria. La parte intitolata »Usando il Modulo di Comando Software allo stato solido« descrive le varie opzioni che vengono presentate sulla selezione dello schermo quando il modulo di Mini Memoria viene inserito per la prima volta e spiega come selezionare queste opzioni.

La parte riguardante »Caricamento e salvataggio di files BASIC« spiega come usare il modulo di Mini Memoria per accedere ai dati dei files, e la parte riguardante »Caricamento e custodia dei Programmi BASIC« descrive come usare il modulo per il caricamento e la custodia files di programma BASIC.

La parte sui »Programmi aggiuntivi del TI BASIC« spiega i sottoprogrammi addizionali. Il modulo di Mini Memoria è in grado di creare una interfaccia con i programmi di linguaggio assembler ed il computer.

»Caricamento di programmi Assembler« descrive le Procedure di caricamento di programmi in linguaggio Assembler e la parte »Routine di utilità del Sistema« fornisce delle spiegazioni circa l'utilità del modulo dei Mini Memoria che accede alle routines CPU ROM e GROM.

La parte »EASY BUG« descrive il funzionamento del programma EASY BUG e le sei appendici che contengono le informazioni riguardanti l'organizzazione di memoria dell'Home Computer.



INLEIDING

MINI MEMORY SOLID STATE CARTRIDGE

De Mini Memory Solid State Cartridge Opdrachtmodule geeft uw Texas Instruments TI-99/4 of TI-99/4A Home Computer een nog grotere veelzijdigheid. Deze opdrachtmodule biedt uw computersysteem namelijk extra geheugen en tevens belangrijke hulpmiddelen voor programma-ontwikkeling. Bovendien heeft de module een ingebouwde batterij, die ervoor zorgt dat de programma's en gegevens die in het Random Access Memory (RAM) van de module liggen opgeslagen, bewaard blijven wanneer de computer-console wordt uitgeschakeld, en zelfs wanneer de module uit de console verwijderd wordt.

DE EIGENSCHAPPEN VAN DE MINI MEMORY MODULE ZIJN:

- Een totaal van 14K bytes geheugen. Dit geheugen bestaat uit 6K bytes Graphics Read Only Memory (GROM), 4K bytes Read Only Memory (ROM), en 4K bytes RAM. De programma's in GROM en ROM beschikken nog over belangrijke programma-ontwikkeling-snufjes. Het RAM biedt extra geheugenruimte voor gegevens en het opslaan van programma's.
- Een ingebouwde batterij in de module om gegevens of programma's die in het RAM-geheugen zijn opgeslagen, te bewaren.
- Extra files. Behalve de 4K-byte RAM file in de Mini Memory module zelf, kan het 24K-byte segment van de Memory Expansion unit, indien aangesloten, gebruikt worden door TI BASIC programma's.
- Assembler-taal mogelijkheden. Met de Mini Memory module kunnen assembler-taal-object programma's in de module zelf of, indien aangesloten, in de Memory Expansion unit geladen worden.
- Extra TI BASIC subprogramma's. Met de Mini Memory module kunnen verscheidene extra subprogramma's m.b.v. TI BASIC instructieregels opgeroepen worden. Een van deze subprogramma's biedt de mogelijkheid om waarden na te gaan (PEEK en POKE).
- Extra gebruiksroutines. De Mini Memory module kent verscheidene programma-routines die toegang verschaffen tot de bronnen van de computer; bijvoorbeeld het koppelen van gebruikersprogramma's aan ROM- en GROM-programma's, het koppelen van assembleer-taal-programmas aan de TI BASIC interpreter en het toegang verschaffen tot VDP RAM.
- Een intern fouten-opsporingsprogramma. Het EASYBUG programma is een handig programma-ontwikkeling-hulpmiddel waarmee u de interne bronnen van het computersysteem kunt bereiken en de fouten in uw programma kunt opsporen.

TOEPASSINGEN

U kunt het RAM in de Mini Memory Module gebruiken om data of programma's op te slaan. Dit geheugen is CPU-geheugen, hetgeen betekent dat het snel toegankelijk is. U kunt van deze snelle opslag- en terugroepcapaciteiten profiteren door data op te slaan die herhaaldelijk in een bepaalde toepassing gebruikt worden of assembleer-taal-programma's die snelle berekeningen uitvoeren op te slaan.

Een TI BASIC programma dat u vaak gebruikt kan in de Mini Memory module opgeslagen worden; dat is handiger dan op een cassettebandje of diskette, in verband met de snellere laadtijd.



In het algemeen verliest het RAM (bijv. het "gebruikers" geheugen in de computerconsole) zijn inhoud wanneer de console uitgeschakeld wordt. De batterij-aangedreven Mini Memory module bewaart echter haar inhoud wanneer u de console uitschakelt. Programma's of data kunnen in het RAM van de module worden opgeslagen; vervolgens kan men de console uitschakelen en de module uit de console halen; brengt u dan de module terug in de console en schakelt u de console weer aan, dan kunt u uw data of programma onmiddellijk weer gebruiken.

Waarschuwing! Wanneer u de Mini Memory module verwijdt of inbrengt, dient de computerconsole uitgeschakeld te zijn om te voorkomen dat data of programma's die in het RAM van de module zijn opgeslagen, verloren gaan of gewijzigd worden.

Een belangrijke eigenschap van de Mini Memory module is haar vermogen assembler-taal-programma's uit te voeren. De module stelt uw in staat uw eigen assembler-taal-programma's te laden voor directe toegang tot de programmeerbare componenten in de computer (bijvoorbeeld de TMS9900 microprocessor of de TMS9918 Video Display Processor). Met behulp van assembler-taal-programma's hebt u ook direct toegang tot apparaten zoals de Wired Remote Controllers of cassetterecorders via hun interface-ingangen op de console.

Bovendien kunt met behulp van deze module assembleer-taal-subroutines van TI BASIC programma's oproepen. Deze assembler-taal-subroutines kunnen functies uitvoeren die slechts op inefficiënte of zelfs in het geheel niet in BASIC uitgevoerd kunnen worden. Programma-routines die in het GROM en ROM van de module zijn opgeslagen vormen een handige koppeling tussen TI BASIC programma's en assembler-taal-programma's.

De module bevat ook het EASY BUG foutenopsporingsprogramma. Met EASY BUG kunt u het geheugen en de programmeerbare componenten in de computer bereiken. EASY BUG bevat ook opdrachten voor het laden en opslaan van geheugen-beeld data op cassetteband.

ALGEMENE GEHEUGENINFORMATIE

N.B.: In de computer Handleiding betekent het groter-dan-symbool ($>$) dat het volgende getal een hexadecimaal (grondtal 16) getal is.

De Mini Memory module bevat in totaal 14K bytes geheugen, bestaande uit 4K bytes batterij-gevoed Random Acces Memory (RAM), 4K bytes Read Only Memory (ROM) en 6K bytes Graphics Read Only Memory (GROM). Het ROM en GROM bevatten nog een aantal routines die extra oproepbare sub-programma's aan TI BASIC toevoegen en die nuttig zijn voor het koppelen van assembleer-taal-programma's aan TI BASIC programma's.

Appendix A van de Handleiding toont de geheugenorganisatie van de gehele geheugenruimte van de computer. De 4K bytes ROM in de Mini Memory module bezetten de geheugenruimte van de adressen >6000 t/m $>6FFF$ (of van 24576 t/m 28671). Appendix B van de Handleiding geeft de details van de ROM-organisatie van de module.

De 4K bytes RAM van de module gebruiken de geheugenruimte van de adressen >7000 t/m $>7FFF$ (of van 28672 t/m 32767). Appendix C van de Handleiding toont de details van de RAM organisatie bij gebruik voor TI BASIC files. Appendix D van de Handleiding beschrijft hoe het RAM georganiseerd is wanneer het gebruikt wordt voor assembleer-taal-opslag. Het GROM gebruikt geheugenruimte die niet rechtstreeks in kaart gebracht is in de CPU-geheugen-adres-ruimte. Appendix E en F van de Handleiding bevatten informatie over het RAM in de Video Display Processor (VDP: Beeldschermbesturing).



VOOR MEER INFORMATIE:

Wanneer u de Mini Memory module voor het eerst gebruikt kunnen een aantal bronnen u nuttige achtergrondinformatie verschaffen:

1. De Texas Instruments HOME COMPUTER HANDLEIDING
2. De Texas Instruments HANDLEIDING VOOR TI EXTENDED BASIC
3. En, wanneer u assembleer-taal-programma's maakt, de Texas Instruments HOME COMPUTER EDITOR/ASSEMBLER HANDLEIDING. Dit boek vormt een complete handleiding voor de TMS9900 assembleertaal en biedt ook details over de interne opbouw van de TI-99/4 en TI-99/4A Home Computers.

GEBRUIK VAN DEZE HANDLEIDING

Deze Handleiding gaat ervan uit dat u al ervaring in het programmeren hebt opgedaan met TIBASIC. Instructies, opdrachten en functies die hetzelfde zijn als in TI BASIC worden slechts kort behandeld. Voor een complete beschrijving zie de handleiding die bij de TI-99/4A Home Computer geleverd is.

Wanneer u de Mini Memory module wilt gebruiken voor het maken van uw eigen assembleer-taal-programma's, dan gaan wij ervan uit dat u ervaring hebt met TMS9900 assembleer-taal-programmeren en dat u vertrouwd bent met de interne organisatie van data en file-structuren die door de Home Computer gebruikt worden. Voor een complete behandeling van deze onderwerpen verwijzen wij naar de Editor/Assembler Handleiding.

De rest van deze Handleiding legt de verschillende eigenschappen en mogelijkheden van de Mini Memory module uit. De paragraaf getiteld "Gebruik van de Solid State Software™ Opdrachtmodule" beschrijft de verschillende opties die op het selectiescherm verschijnen wanneer men de Mini Memory module in werking stelt en legt uit hoe men uit deze opties selecteren moet.

De paragraaf over "Laden en Opslaan van TI BASIC Files" legt uit hoe u de Mini Memory module kunt gebruiken voor het bereiken van data-files en de paragraaf "Laden en Opslaan van BASIC Programma's" beschrijft hoe de module gebruikt kan worden voor het laden en opslaan van BASIC programmapfiles.

De paragraaf over "Extra TIBASIC Subprogramma's" beschrijft de extra subprogramma's die de Mini Memory module biedt en die een koppeling kunnen vormen tussen assembleer-taal-programma's en het computersysteem.

"Laden van Assembleer-taal-programma's" beschrijft de procedure voor het laden van assembleer-taal-programma's en -subprogramma's en de "Systeem-gebruikers-routines" legt uit hoe de Mini Memory module toegang heeft tot CPU ROM en GROM routines.

De "EASY BUG Foutenopsporing" -paragraaf beschrijft de werking van het foutenopsporingsprogramma en de zes appendixes bevatten informatie over de geheugenorganisatie van de Home Computer.

INLEIDING

REGEL-VOOR-REGEL ASSEMBLER (LINE-BY-LINE ASSEMBLER)

De cassetteband in de Mini Memory Solid State Cartridge Opdrachtmodule bevat een regel-voor-regel symbolische assembler en een grafiek-demonstratie-programma, LINES genaamd. De regel-voor-regel assembler stelt u in staat om, steeds per regel, rechtstreeks van het computer-toetsenbord TMS9900 assembleer-taal broncode in te voeren. Het LINES demonstratie-programma (alleen op de TI-99/4A Home Computer) trekt automatisch kleurrijke lijnen op het scherm.



Wanneer het assembler-programma in de Mini Memory module geladen is, wordt elke bron-instructie die u invoert onmiddellijk geassembleerd in objectcode en opgeslagen in de geheugenlocatie die door uw broncode wordt aangeduid. Zodra u dus de invoering van uw programma voltooid en naam en adres in de symbooltabel opgeslagen hebt, dan is het klaar voor gebruik.

N.B.: Daar de code direct wordt geassembleerd opgeslagen bij het invoeren van elke regel, dient u er op te letten dat de geheugenadressen die in het programma worden aangegeven, inderdaad beschikbaar zijn. Anders wordt geen code gegenereerd, noch opgeslagen.

Hoewel de assembler elke instructie omzet in machinecode tijdens de invoering, blijft sommige broncode bewaard in een negen pagina's lange tekstbuffer. U kunt het beeld op het scherm langzaam laten passeren om de eerder ingevoerde broncode-regels opnieuw te bekijken, door op de toetsen met de pijl omhoog en naar beneden te drukken.

Deze handleiding behandelt de eigenschappen van de regel-voor-regel assembler. Hierbij wordt ervan uitgegaan dat u al kunt programmeren in assembleer-taal. Zie voor een complete bespreking van de TMS9900 assembleer-taal de Redigeer/Assembleer handleiding. Instructies voor het laten lopen van het LINES demonstratie-programma vindt men ook in deze handleiding (zie "Laden van regel-voor-regel assembler").

N.B.: Het LINES programma kan alleen uitgevoerd worden op de TI-99/4A Home Computer; deze heeft namelijk de geavanceerde grafiekbesturing die voor dit programma vereist is.

LADEN VAN DE REGEL-VOOR-REGEL ASSEMBLER

Zowel de regel-voor-regel assembler als het LINES grafiek-programma worden – tegelijkertijd – geladen van de cassetteband door de L (LOAD)–opdracht van de EASY BUG foutenopsporing optie. De onderstaande stappen beschrijven de laad-procedure.

1. Sluit, wanneer de Mini Memory module zich in de computer console bevindt, uw cassetterecorder op de console aan, zoals beschreven in de HOME COMPUTER HANDLEIDING.
2. Druk op een willekeurige toets om de hoofdselectielijst te doen verschijnen en kies dan de MINI MEMORY optie van de lijst. Wanneer de Mini Memory selectielijst verschijnt, drukt u op 3 (RE-INITIALIZE) om het geheugen gereed te maken voor het laden van een nieuw programma. Druk vervolgens op QUIT om terug te keren naar het hoofdtitelscherm van de computer.
3. Breng de assembler-cassette in de recorder en breng het bandje in zijn beginstand.
4. Druk op een willekeurige toets om de hoofdselectielijst te doen verschijnen en kies dan de EASY BUG optie van de lijst.
5. Wanneer het EASY BUG-opdracht-scherm verschijnt, drukt u op een willekeurige toets om het scherm te wissen. Type vervolgens L en druk op ENTER om het laadproces te starten. Van nu af aan verschijnen instructies op het scherm die u door de procedure heen loodsen. Volg deze aanwijzingen om het programma te laden.
6. Wanneer het assembleerprogramma geladen is, drukt u op QUIT om terug te keren naar het hoofdtitelscherm. Druk vervolgens op een willekeurige toets om verder te gaan naar de hoofdselectielijst en selecteer de MINI MEMORY optie.
7. Wanneer de Mini Memory selectielijst verschijnt, drukt u op 2 om de RUN optie te selecteren. Het scherm wordt gewist en het programma vraagt u naar de naam van het programma dat u wilt laten lopen.



- A. Wanneer u het LINES programma wilt laten lopen (dit kan alleen op de TI-99/4A), typt u LINES en drukt u op ENTER. Het programma tekent een kleurrijk lijnen-ontwerp op uw scherm. Drukt u op de C-toets, dan kunt u de kleur van de lijn die op dat moment getekend wordt "bevrozen"; alle daarna getekende lijnen houden dan die kleur. Drukt u nogmaals op de C-toets, dan wordt het één-kleur-effect weer opgeheven. Om het programma stop te zetten drukt u op QUIT.
- B. Wanneer u een nieuw assembleer-taal-programma wilt creëren, dan drukt u op NEW en vervolgens op ENTER. Het programma voert de Assembler in, wist de symbooltabel (hierover later meer) en wacht tot u eerste programmaregel invoert.
- C. Wanneer u met een programma waar u eerder mee begon verder wilt gaan, dan drukt u op OLD en vervolgens op ENTER. De oude symbooltabel blijft bewaard en het scherm laat de volgende geheugenlocatie zien; u kunt met uw programma verder gaan.

N.B.: Het kan zijn dat uw Mini Memory module al geladen is met het Assembler- en LINES-programma. Om dit te controleren selecteert u de Mini Memory RUN-optie en voert u de juiste programmanaam in wanneer daarom gevraagd wordt. Is het programma aanwezig dan begint het onmiddellijk te lopen. Is het programma niet in de module geladen, dan verschijnt op het scherm de boodschap "PROGRAM NOT FOUND" (programma niet gevonden).

U kunt ook de LINES code nalopen door in EASY BUG de M-opdracht te gebruiken. Voer M7CD6 in en blijf op ENTER drukken om elke code-lijn na te lopen zonder gegevens te wijzigen.

Belangrijke Opmerking: Wanneer u een programma invoert en assembleert kan de symbooltabel over een gedeelte van het LINES-programma dat in de Mini Memory module ligt opgeslagen heen schrijven. Wanneer u LINES weer wilt laten lopen, dan laadt u het programma gewoon opnieuw van de cassetteband in de module.

ASSEMBLER SYNTAXIS

Elke regel (of record) van uw bronprogramma bestaat uit vier secties, gebieden genaamd. Deze gebieden moeten, indien aanwezig (sommige zijn facultatief) verschijnen in de volgorde en vorm (syntaxis) die door de Assembler vereist worden. In deze handleiding worden de volgende afspraken gehanteerd bij de syntaxis-definities voor machineinstructies en richtlijnen.

- Items in hoofdletters, inclusief speciale tekens, moeten precies zo als ze er staan, ingevoerd worden.
- Items tussen vierkante haakjes ([]) zijn facultatief.
- Items tussen hoek-haakjes (<>) zijn vereiste gebieden.
- Een kleine letter b duidt op een enkele spatie.
- Een kleine letter b gevolgd door drie stippen (b...) duidt op een of meer spaties.

De algemene syntaxis voor een Assembler-richtlijn is als volgt

[label] b... <opcode> b [operand] [,operand] [b...opmerking]

Het label-gebied vereist ofwel een spatie (wanneer er geen label is) ofwel een of twee tekens. Het eerste teken moet alfabetisch zijn; het tweede teken, indien aanwezig, mag alfanumeriek zijn. Het label wordt gevolgd door een of meer spaties. Wanneer u geen label typt, dan zorgt het indrukken van de spatiebalk er automatisch voor dat de loper naar het begin van het opcode-gebied gaat.



Het opcode-gebied bevat de bewerkingscode van de taak die door de broninstructie uitgevoerd moet worden. Het gebied bestaat uit een tot maximaal vier alfabetische tekens, bijvoorbeeld A voor Add of AORG voor de Absolute Origin instructie, gevolgd door een enkele spatie.

Het operand-gebied bevat een of twee operands, al naar gelang de vereiste voor de bepaalde instructie. Merk op dat het operand-gebied geen spaties bevat en dat meerdere operands door komma's gescheiden worden. Het operand-gebied wordt afgesloten door op de spatiebalk te drukken (de looper gaat verder naar het opmerking-gebied) of op ENTER (geeft het eind van de regel aan). Wanneer een instructie geen operand heeft, wordt het operand-gebied weggelaten.

Het opmerking-gebied kan, indien gebruikt, elk teken bevatten en loopt tot u ENTER indrukt om de regel te voltooien.

Voorbeelden:

XY	MOV R1,	VP	Sla R1 in VP op
Z	SR1, R2		Bereken het verschil

De regel-voor-regel assembler predefinieert bepaalde symbolen. Wanneer een operand een dollar-teken (\$) bevat als eerste teken, dan betekent dit dat verwezen wordt naar de inhoud van de locatie-teller. Bij locatie >7000, bijvoorbeeld, zijn de instructies

en JMP \$ + 8
 JMP >7D08 equivalent. Bij het geven van register operands kunt u het symbool R gebruiken, gevolgd door een decimaal getal.

en MOV R1, R15
 MOV 1, 15

zijn equivalent.

N.B.: Het default getallen systeem voor de regel-voor-regel assembler is decimaal; hexadecimale getallen worden aangeduid door het groter dan (>) voorvoegsel.

GB The Mini Memory Solid State Cartridge Command Module increases the versatility of your Texas Instruments TI99/4 or TI99/4A Home Computer by providing additional memory for your system and important tools for program development.

D Das Mini Memory Solid Cartridge-Befehls-Modul erhöht die vielseitige Verwendbarkeit Ihres TI99/4 oder TI99/4A Home-Computers von TEXAS INSTRUMENTS durch zusätzliche Speicherkapazität und wichtige Instrumente für die Programm-Entwicklung.

F Le module *Solid State Cartridge* »Mini Mémoire« dote l'Ordinateur Familial d'une mémoire de stockage supplémentaire et offre à l'utilisateur la possibilité de développer des programmes plus facilement.

I Il Modulo di Comando del Cartridge allo stato solido aumenta la versatilità del vostro Home Computer Texas Instruments TI-99/4A mettendo a disposizione una memoria aggiuntiva per il sistema e per importanti supporti per lo sviluppo dei programmi.

NL De Mini Memory Solid State Cartridge Opmachtmodule geeft uw Texas Instruments TI-99/4 of TI-99/4A Home Computer een nog grotere veelzijdigheid. Deze opdrachtmodule biedt uw computersysteem namelijk extra geheugen en tevens belangrijke hulpmiddelen voor programma-ontwikkeling.

