ROS User's Manual

by

R. M. Smets

Introduction

Certain parts of systems software which are necessary for the execution of almost any program are stored in read-only memory instead of the regular read-write semiconductor memory. This offers several advantages:

* The read-only memories are cheaper than read-write memories.
* This portion of software need not be loaded.
* This portion of software cannot be destroyed by a malfunctioning program.
* Changes can be made to systems software without requiring that all existing programs be altered or recompiled.
* Less power is consumed by read-only memories.

The types of software which are included are as follows:

1. Input/Output Device Routines
2. Interrupt Handling
3. PL/1 Intermediate Code Interpreter
4. Conversion Subroutines to Convert Data Types (Binary, Decimal, Character)
5. Decimal and Binary Arithmetic

This software is collectively known as the Read-Only Operating System (ROS). It is made up of three modules: Console, Disk, and Interpreter. Each of these modules may be changed independently. The console module may be changed if a different keyboard display and printer are to be used. The disk module may be changed if a different mass storage device is to be used or removed if there is to be no mass storage device. The interpreter module may be removed if PL/1 object code is not used.

Memory Organization

Figure 1 gives the overall layout of memory which is assumed by the operating system. 128 bytes of memory are referred to as "miscellaneous ROS data," which is described in more detail below:

Address	Name	Function
4080-4082	RA	Jump to start of program or subroutine return address
4083-4085		Jump to interrupt processing address
4086-4088		Jump to address for processing when system is waiting for printer or keyboard
4089	PLC	Least significant byte of address of last character loaded into printer buffer
408A	PTC	Least significant byte of address of last character output to printer
408B-408C	POS	Printer carriage position in 60th's of an inch
408D	RIB	Flag to indicate if a printable character was output on last printer operation
408E		Unused
408F	HEXX	2 if last key as "HEX" key; 1 if "HEX" was next to last; 0 otherwise
4090	INSF	1 if in insert mode
4091	FUNKEY	Function key which terminated last line of input

Address	Name	Function
4092	TOOK	Number of characters of last keyboard line which have been used
4093	CURSE	Cursor position
4094	UNDER	Character under cursor
4095	KSIZ	Current size of keyboard input line
4096-4097	OSEZ	Number of characters used for display output
4098	ACTK	Zero if keyboard input line is open, otherwise code of key which terminated line
4099-409A	THERE	Address of beginning of disk transfer
409B	NRT	Disk record count
409C	SNRT	Number of records to be transferred to or from disk
409D	ERC	Disk error count
409E-409F	INDEX	Current record number on index
40A0	DISK	Selected disk drive number
40A1-40A4	TRKS	Track numbers of drives 1 to 4
40A5	AD	Bit 0 set will prevent files from being opened on drive 1. Bit 1 for drive 2; bit 2 for drive 3; bit 3 for drive 4
40A6-40B5		File description for index file

Address	Name	Function
40B6-40B7	PART1	Length of portion of record to be transferred
40B8-40B9	PART2	Length of unused portion of record
40BA-40BF		Unused
40C0-40CF	TABB	One bits indicate keyboard tab positions
40D0-40E7	LFILE	File description of file to be loaded by loader
40E8-40F4		Unused
40F5-40F6	ONCODE	Oncode for PL/1
40F7-40F8	ERRA	Error address from ON ERROR statement
40F9-40FA	EOFA	Address from ON ENDFILE statement
40FB-40FD	OA	Jump to output subroutine for put
40FE-40FF	PC	Interpretive program counter

0 7FF	Console Module of ROS
800 FFF	Disk Module of ROS
1000 17FF	Unused
1800 1FFF	PL/1 Interpreter Module
2000 3FFF	Unused
4000 407F	Processor Stack
4080 40FF	Miscellaneous ROS Data
4100 417F	Keyboard Buffer
4180 41FF	Printer Buffer
4200 42FF	System Scratch (Used by Loader and Other Routines as Scratch)
4300	User Area

Figure 1 – Memory Layout

Console Module

The console module handles input/output and initialization for the keyboard, display, and printer. Since all of these devices use ASCII codes, there are number to ASCII and ASCII to number conversion subroutines for both binary and decimal numbers. The printer and keyboard are used on an interrupt basis.

A table of jumps at the beginning of the module makes it possible to change the operating system without changing the addresses of the entry points. The function of each of these entry points is described below:

Name: PI
Address: 00

This entry point is called by the system automatically when an interrupt occurs. If the interrupt was caused by the restart button, the system is initialized. The miscellaneous ROS data area is zeroed, except for the three jumps at the beginning of the area. The printer is reset and the display is initialized to all spaces. Then the START entry point (described later) is entered.

If the interrupt was not due to the restart button, the program jumps to address 4083. If this instruction has not been altered by the user program, it jumps to ROS where the keyboard and printer are tested to see if IO is required. If the printer is ready and there is data in the print buffer, all data is output up to the next printable character. If the keyboard input line is not closed (see description of PROCH entry point) a test is made for keyboard data and if it is present, the PROCH entry point is called to process the character.

Name: TOSTR
Address: 03

Converts a decimal number to ASCII. Intended only for use by the PL/1 interpreter. A decimal number is removed from the stack

(see the Advanced PL/1 Programmers Manual for a description of the format) and converted to a character string in the system scratch area. Returns the string address and length in the stack.

Name: **TODEC**
Address: **06**

Converts a character string to decimal. Intended only for use by the PL/1 interpreter. Removes the string address and length from the stack and puts a decimal number into the stack.

Name: **UPDIS**
Address: **09**

Updates the display to show what is in the keyboard buffer.

Name: **MUL**
Address: **0C**

Multiplies two 16 bit signed integers in registers DE and BC. Result is in HL.

Name: **DIV**
Address: **0F**

Divides the 16 bit signed integer in HL by DE. Remainder in HL quotient in DE.

Name: **BICHAR**
Address: **12**
Registers: **HL – Binary Number**

Converts the number in HL to ASCII. Stores the rightmost character at address in DE. Return string address in DE and string size in C.

Name: NHL
Address: 15

Negates the binary integer in HL. Result is in HL.

Name: START
Address: 18

Used to allow programs to return control to the operating system when they have completed execution. If the disk module of ROS is not present, a jump to address 8000 will be made where user programs can reside in ROM in systems without disks. Otherwise, the message "Q1/LMC AT YOUR SERVICE" will be put on the display and the name of a file will be accepted from the keyboard and given to the loader.

Name: KFILE
Address: 1B

Accepts a file name from the keyboard input using the first non-space character up to the first character with an ASCII code of less than 30 up to a maximum of eight characters. The file name goes into address 40D2 using the KEYIN entry point.

Name: KEYIN
Address: 1E
Registers: HL – string address
C – string length

Inputs a character string from the keyboard buffer. If there is no input, it waits for input while repeatedly calling address 4086, which is initialized to jump to CLRDK which raises the disk heads. The address of this jump may be modified for multiprogramming purposes. If the available input does not fill the string, it will be padded with blanks.

Name: GETDN
Address: 21

Uses keyboard input up to the first numeric character that is followed by a non-numeric character. A numeric character is 0–9 or "." or "E" for scientific notation. The characters are transferred to the system scratch area using KEYIN and converted to decimal and put in the stack by TODEC. Intended for use by the PL/1 interpreter.

Name: NKEY
Address: 24

If a keyboard input line has been partially used by KEYIN, the remainder of the input line will be discarded.

Name: DISPLAY
Address: ~~24~~ 27
Registers: HL – string address
C – string length

The specified character string will be output to the display and the keyboard input line will be moved to the right of the last character output. The following control codes are used:

0D clear display
00 stop outputting

Name: PRINTER
Address: 2A
Registers: HL – string address
C – string length

The print buffer is loaded with the specified character strings. If the print buffer is full, address 408B, which is initialized to a jump to CLRDK, will be called repeatedly while the print buffer is being emptied by the interrupt service routine. The following control codes are used:

00	stop transfer
01	move print position up one half line
02	move right 1/60 inch
03	move down 1/48 inch
0A	move down 1 line
0D	move to left margin and down 1 line

Name: CARB
Address: 2D
Registers: HL – string address
C – string length

Convers a character string to binary. Sets the M flag on overflow. Result is in ~~DE~~ DE

Name: STOP
Address: 30

Clears keyboard input and holds the processor in a loop until the GO key is depressed.

Name: PROCH
Address: 33
Registers: A – character to be processed

Processes a character as keyboard input. With the exception of the keys listed below, the seven-bit ASCII codes are put into successive memory location in the keyboard buffer. Keys with codes 5--7, 0C--19, and 83–9F will act as function keys which close out the input line. This means that the operating system will not test for keyboard input until the input line is used up by calls to KEYIN or the keyboard input is cleared out by some other means. The following keys are used to manipulate the input line.

Key	Code	Action
TAB CLR	2	The current cursor position will not be a tab position

Key	Code	Action
TAB SET	3	The current cursor position will be a tab position
CORR	4	The cursor will be moved back one position
TAB	9	The cursor is moved to the next tab or the 128th position
STOP	F	Keyboard input is cleared and the processor is held in a loop until the "GO" key is depressed
REV TAB	10	The cursor is moved to the previous tab or the left margin
CLEAR ENTRY	1B	All keyboard data is cleared and the cursor goes to the left margin
CHAR ADV	1C	Moves the cursor right one position
HEX	1A	The next two character keys will be interpreted as a hexadecimal code for a character
DEL CHAR	1D	Deletes the character under the cursor and moves data to the right of the cursor left one position
INSERT MODE	1E	All data to the right of the cursor and under the cursor will be moved right one position before a new character is put in the input line, until the "INSERT MODE" key is depressed again or the input line is closed.

Name: INTRET
Address: 36

For use by programs that bypass normal interrupt handling by changing the address of the jump at address 4083. A jump to this entry point will restore the registers and flags and enable the interrupts in the proper order.

Name: INDEX
Address: 39
Registers: DE – address of configuration to be found
B – length of configuration
HL – address the string to be searched
C – length of string

Searches a string of bytes for a specified configuration of bytes. Returns byte number at which the configuration starts in HL. The first byte of the string is byte one. Returns zero if not found.

Name: SHIFTY
Address: 3C

Shifts the first 16 bytes of the system scratch area left one half byte. Used by PL/1 interpreter to handle decimal numbers.

Disk Module

The disk module provides subroutines for input/output to the floppy disks. The subroutines assume a disk format which is described in Figure 2. Track 0 is a special file called INDEX. Each record of this file is a description of one of the files on the disk. The memory area into which one of these records is read is called a file description.

Each record on a file is preceded by a sector header. The number of sector headers on a track depends on the record length of the file.

The subroutines which detect errors all use the same codes to report errors. The error code is returned in the A-register. If there were no errors, the A-register is zero and the Z flag is set.

Error Code	Error
1	Sector header not found
2	Read Error
3	Write Error
4	Key not found
5	Disk has been removed or file closed since file was opened
6	An attempt was made to access a record which was after the end of a file
7	An attempt was made to write on a protected file

The entry points are as follows:

Name: READ
Address: 800
Registers: A – number of records to be transferred
BC – file description
DE – memory space allocated per record
HL – address of first record

A group of records is transferred from disk into memory. If the memory per record is greater than the record length of the file, the unused memory will be unaffected. If it is less, the excess portion of the record will not appear in memory. The record number in the file description will be advanced by the number of records transferred.

Name: WRITE
Address: 803
Registers: same as READ

A group of records is transferred from memory onto disk. If the memory per record is greater than the record length, the excess memory will not be written on the record. If it is less, the record will be padded with binary zeros. The record number in the file description will be advanced by the number of records transferred.

Name: REWRITE
Address: 806
Registers: same as READ

The last group of records read or written will be written.

Name: KEY
Address: 809
Registers: A – key length
BC – file description address
DE – position of key in record
HL – key address

The file will be scanned for a record which is identical to the key at the specified position. The record number in the file description

will be set to the number of the matching record.

Name: OPEN
Address: 80C
Registers: HL – file description address

A file will be found in the Index which has the name which is in the file description before the call to OPEN. The Index record will be read into the file description and the disk drive number will be filled in. The Indexes of the disks will be accessed in the order of the drive numbers. A drive may be skipped if specified by memory location AD (see Memory Organization). A file must be opened before any other disk operations may be done on it.

Name: CLOSE
Address: 812
Registers: HL – file description address

The end of the file data in the Index will be updated to the record number in the file description. A file must be closed immediately after records have been added to it by a succession of writes.

Name: LOADER
Address: 80F

Opens the file named at address 40D2 and loads it as a binary object file. The format of the file is described in Figure 3.

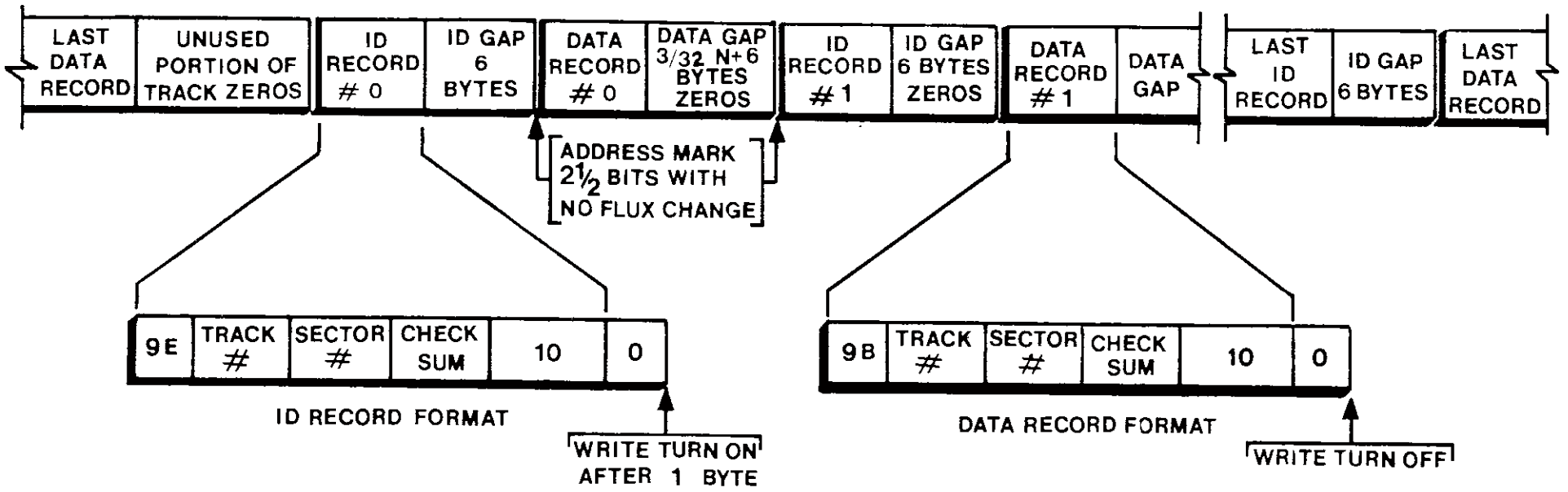
Name: CLRDK
Address: 815

All disk drives are deselected causing the heads to be raised.

Name: REPORT
Address: 818
Registers: BC – file description address
A – error number

Puts a message describing the error on the display, clears keyboard input and waits for a "GO" key depression.

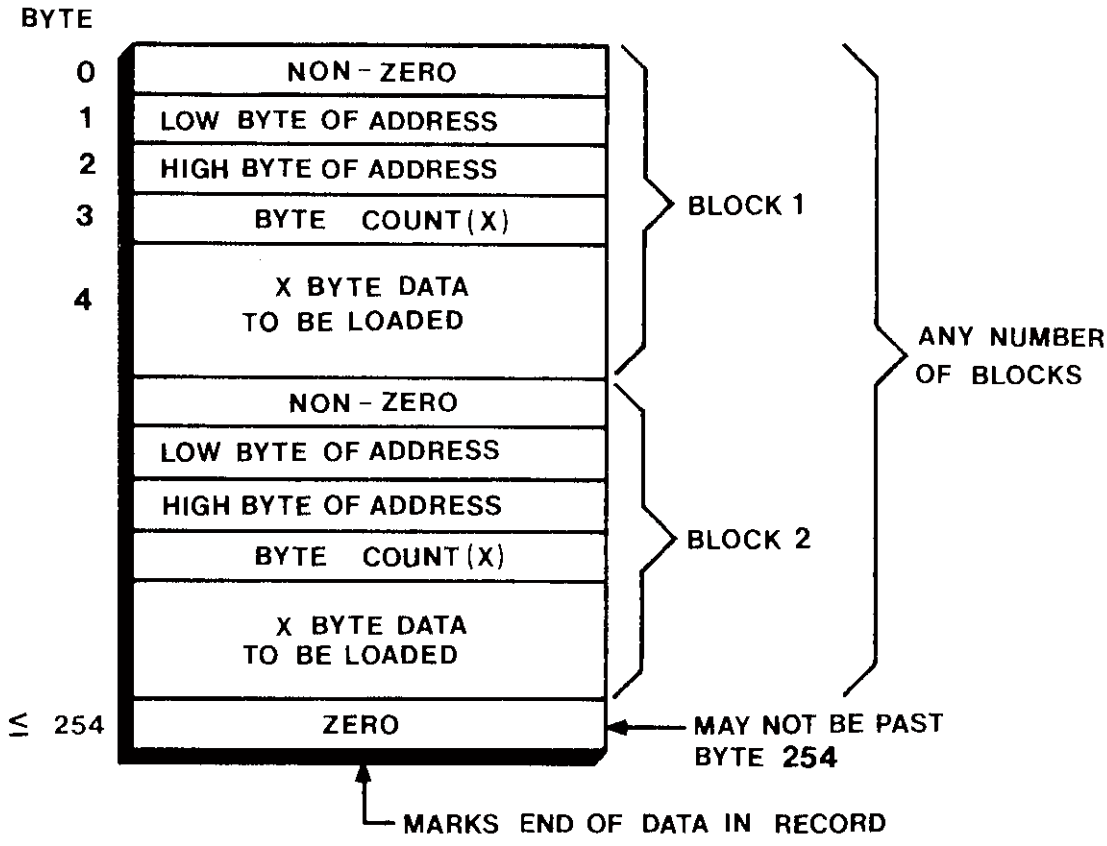
TRACK FORMAT



BYTE		
0	RECORD NUMBER	ZERO ON INDEX
2 4 6 8	NAME OF FILE IN ASCII PADDED WITH BLANKS ON RIGHT	ONLY THIS PORTION (FILE NAME) NEEDS TO BE IN MEMORY BEFORE CALL TO OPEN
A	NUMBER OF RECORDS	ENFILE DATA. EXPANDED TO MAXIMUM BY CALL TO WRITE
C	RECORD LENGTH	REFERED TO AS N ABOVE
E	RECORDS / TRACK; DISK #	DISK # IS ZERO ON INDEX
10	FIRST TRACK #	
12	LAST TRACK #	MOST SIGNIFICANT BIT IS SET IF PROTECTED
14	UNUSED	
16	RECORD # BEFORE LAST OPERATION	ZERO ON INDEX. USED FOR REWRITE AND RETRIES

FILE DESCRIPTION
AS IT APPEARS IN MEMORY AND ON INDEX

FIGURE 2



LOADER RECORD FORMAT

FIG. 3

PL/1 Interpreter Module

The operation of the PL/1 intermediate code interpreter is described in the "Advanced PL/1 Programmers Manual." However, for completeness, the entry points will be described here:

Name: NORMY
Address: 1800
Registers: HL – number address

Used to normalize a decimal number and put it in the stack.

Name: GIN
Address: 1803
Registers: HL – subroutine address

Used by assembly language programs to call interpretive subroutines.

Name: CLEAR
Address: 1806

Zeros the first 16 bytes of the system scratch area.

Name: START
Address: 1809

Used to start the processor executing an interpretive program.