ROAM-0A

A COLLECTION OF HP41C MICROCODE ROUTINES FOR THE ProtoCODER AND MLDL

TABLE OF CONTENTS

HP-41C ROM IMAGE ROAM-0A

OVERVIEW

ROAM-0A was assembled as a personal tool to help control a PROTOCODER setup encompassing both PROTOCODER 2 and MLDL type ROME.

ROAM-0 is an earlier version of ROAM-0A designed to work with PROTOCODER 1 / MLDL systems. It is included only for the temporary convenience of users who have not yet had their PROTOCODER upgraded to mark 2. The following notes are for ROAM-0A, with the main differences for ROAM-0 generation noted in the addendum. ROAM-0A and ROAM-0 XROM NUMBERS ARE NOT COMPATIBLE.

ROAM-0A functions stand alone, but the RPN “driver” routines require the presence of ASSEMBLER-3 AND X-FUNCTIONS.

In ROAM-0 the REGROMP function transfers (port independently) to the ASSEMBLER-3 REG>ROM function if the destination address is in MLDL.

While using these routines, NB:

THE PROTOCODER DEVICE
SWITCHES SHOULD BE
SET EQUAL TO THEIR
RESPECTIVE “ROM ADR”
SWITCHES

HEXIDECIMAL CALCULATIONS

XROM	05 , 01	/ND	As IL Development module AND, NOT, OR, XOR with the vowel replaced by “/” to avoid name clashes with the ASSEMBLER-3 functions
	, 02	N/T	
	, 03	/R	
	, 04	X/R	
	05 , 05	BIT?	As IL Development module functions of the same name
	, 06	ROTXY	
	, 07	BININ	
	, 08	BINVIEW	
	, 09	HEXIN	
	, 10	HEXVIEW	
	, 11	OCTIN	
	, 12	OCTVIEW	
	05 , 13	ROMCHKS	As IL Development module function “ROMCHKX” except that the correct checksum is left in (X) 2:0, with the correct address in (X) 6:3, ready to write to ROME (see “MCWT”, later).

The functions listed are from the HPIL Development module.

GENERAL 1

XROM	05 , 14	“0”	Most of the RPN routines send the program pointer to an empty program “0”, at the top of RAM, when complete. In the event it is not there, a copy is here.
	, 15	B?	Returns the number of BYTES in current RAM program to display (stack not affected). Non-programmable. Use trace mode to print the result.
	, 16	GLP	Sends RPN program pointer to ,END. Non-programmable
	, 17	RST	Resets register d to FIX 2, etc., ready to count the cost.
	, 47	D?	Returns distance in BYTES between the result of GETPC (ASSEMBLER-3) in (X) and the current RPN program pointer. Result is not meaningful if (x) 3:0 doesn’t contain MM-format pointer data, or is not from the same 4k if in ROM (works for both ROM and RAM). Result is sent to the display (stack is not affected). Non-programmable. Use trace mode to print.

PRIVATE

XROM	05 , 18	WRPA	Allows IL/Mass Storage ROM (Rev. 1H) to record “WRTA” files with “PRIVATE” programs in RAM.
	, 19	WALP	Allows Card-Reader (Rev. 1E, 1F, & 1G) to record “WALL” sets with “PRIVATE” programs in RAM.
	, 20	SPV	Sets current RAM program to “PRIVATE” without the necessity for a Card-Reader.
	, 21	CPV	It is not known what this function does.

CODE LOADING AND MOVING

XROM	05 , 22	CXISA?	Standard 41C conditional: TRUE if (X) 2:0 is equal to the contents of ROM(E) at address specified in (X) 6:3
	, 23	ROMP?	Takes the value in (X) 6:3 as an ROM(E) address. Clears flag 8 if address is in MLDL; or sets flag 8 if address is in PROTOCODER 2; otherwise: clears user flag 8 and gives “ROM” error.
	, 24	WTMC	Takes the value in (X) 6:3 as a ROME address and writes the value in (X) 2:0 to it; always writes to MLDL, but only writes to PROTOCODER 2 if user flag 8 is set.
	, 25	REGROMP	Works as REG>ROM for either MLDL or PROTOCODER 2.
	, 26	RDROMP	Works as “READROM” in the MC EPROM, but for either MLDL or PROTOCODER 2; place filename in alpha, decimal ROME page number in (X); then XEQ. Doesn’t affect 8-bit RAM.
	, 27	WRTROM	As MC EPROM “WRTROM”: place filename in alpha, decimal ROME page number in (X); the XEQ. Doesn’t affect 8-bit RAM.
	, 28	COMC	Given a “source-block” start address in (X) 10:7, a “destination block” start address in (X) 6:3, and the number of words (in hex) in (X) 2:0, will copy source code to destination address. Blocks may overlap in any order. Doesn’t affect 8-bit RAM. Will not write or read across 4k boundaries. Writes to either MLDL or PROTOCODER 2.

, 29	“MC”	“Driver” routine to automate microcode assembly & disassembly. See addendum for user instructions.
, 30	“MMC”	“Driver” routine to automate microcode moving. See addendum for user instructions

ASSEMBLER-3 UTILITIES

XROM	05 , 31	+F	Renamed “UPDFAT” routine from the ASSEMBLER-3 manual.
	, 32	GTOPP	Renamed “GTOEND” routine from the ASSEMBLER-3 manual.
	, 33	APPLBL	As in the ASSEMBLER-3 manual.
	, 34	GORAM	
	, 35	XQ>XR	
	, 36	“XI”	

GENERAL 2

XROM	05 , 37	“XC”	“CATALOG” XROMS: XEQ to see menu: PAGE ?? XROM Press A, then fill the prompt for ROM page number, R/S, or press C, to list all XROMs; or press E, then input XROM number, R/S. Outputs are displayed if the printer is not present: CF 21 to “review” of SF 21 to stop at each output.
	, 38	MCCF	Compares two blocks of microcode and returns addresses of the first difference between the blocks. Start addresses are taken from (X) 7:4 and (X) 3:0, result is returned in the same format.
	, 39	RYLX	Rotates (Y) left by the number of nibbles specified in decimal in (X). Roll the stack down to place the result in (X) and rotation parameters in (T). Initial (Y) value is put in (L).
	, 40	$>X	Sets (X) 13 to 0.
	, 41	MPPKX	Takes hex pre/post key assignment byte from (X) and effects the requested assignment, or crashes, as specified.

NO XROM #: Tapani Tarvainen’s “off-interupt” routine from PPC TN #15.

LOOP CONTROL

XROM	05 , 42	TOGF	Toggles user flag specified in (X). As for NFCROM TOGF, except when executed from the keyboard (only) it displays “YES” or “NO” to indicate which way the flag was toggled. (Does not work as a conditional when executed as a program line).
	, 43	EL	Clears user flag 33
	, 44	DL	Sets user flag 33 and clears user flags 55 and 21.
	, 45	“OPP”	Selects printer (IL accessory ID:32)
	, 46	“OPV”	Selects video (IL accessory ID:48)

“OPP” and “OPV” require the Extended I/O Module. They should automatically call EL, if required, but are miscoded; alter line 08 to “FS?33”.

Update: *The coding error mentioned has now been corrected, but on the ROAM-0A file only.*

(DIS)ASSEMBLY

XROM 05 , 29 “MC”

XEQ “MC” to activate the local labels:

[D] DISASSEMBLE MICROCODE WITH MNEMONICS
Fill prompts for the beginning and end addresses, pressing R/S after inputting each. (R/S only at END prompt to specify FFFF). R/S to stop or restart listing. Set or clear flag 21 to control printing, pausing, or stopping.

[d] DISASSEMBLE MICROCODE WITHOUT MNEMONICS
At the “ALPHA?” prompt: press R/S to list hexcode only, or ALPHA to specify simultaneous character output: then R/S only (or [R], R/S) to print ROM characters, or [A]. R/S to print ASCII characters. Flag 21 controls output as above.

[g] ASSEMBLE MICROCODE
Fill prompt for the beginning address, R/S thereafter, input hexcode 10-bit words, in order, R/S after each. R/S only: increments pointer without writing. Shift, R/S to exit. Leading zeros need not be input.

[A] ASSEMBLE MNEMONICS
Fill prompt for the beginning address, R/S thereafter, input ASSEMBLER-3 mnemonics, in order, R/S after each. R/S only: increments pointer without writing. To obtain the ASSEMBLER-3/MLI string loading facility for the PROTOCODER address use a period (.) instead of a dollar sign ($) before the string. Switch out of Alpha to exit.

After each address is specified for assembly, “MC” determines its nature ROM, PROTOCODER 2, or MLDL (MLI) and signals the result.

Each input is available for the printer-trace if required by setting or clearing flag 21.

Each assembled hexcode is echoed in the display: set or clear flag 20 to control its printing

Set flags 20 and 21 and specify a beginning address in ROM to get an instruction/hexcode listing without writing to ROME.

“MC” does not send the program pointer to “0”.

“MC” checks each location after writing to it. ASSEMBLER-3 errors are signaled by “ERROR” and a beep. Writing errors (i.e., a ROME location does not match the data just sent to it) are signaled by :$:ERR: and a beep. The dollar symbol is added as a reminder that when a string is written to an MLDL this type of error is generated, even though it is not one (as well as a reminder that if it really is a writing error, you could well be into the :$:ERR: getting it fixed).

After most errors “MC” recycles to re-present the preceeding prompt, except when a string input to an MLDL generates a pseudo write error, when it moves into the next location to be written to.

“MC” uses R00, stack, Alpha, and user flags.

CODE MOVING

XROM 05 , 30 "MMC"

XEQ "MML" to activate local labels:

[A] ROM>REG
Fill prompts for block beginning address and block end address, R/S after each. "Header" is put into R05, data and up.

[E] REG>ROM
Fill prompt for beginning address, R/S. "Header" is assumed to be in R05.

[e] POSITION R>R DATA
When R>R file is read into the 41 by "READR" or "RDTA", the "Header" will be put into R00. Press [e] to move the file up so that the "Header" is in R05.

[C] COPY MICROCODE, no affect on 8-bit RAM.
Fill prompts* for source beginning, source end, destination beginning, and destination end, R/S after each.

[B] COPY MICROCODE leaving R>R file of source in 8-bit RAM.
Fill prompts* for source beginning, source end, destination beginning, and destination end, R/S after each. Copy left in R05 up.

[D] COPY MICROCODE leaving R>R file of destination in 8-bit RAM.
Fill prompts* for source beginning, source end, destination beginning, and destination end, R/S after each. Copy left in R05 up.

[b] SWAP MICROCODE leaving R>R file of destination in 8-bit RAM.
Fill prompts* for source beginning, source end, destination beginning, and destination end, R/S after each. Copy left in R05 up.

* Any one prompt of the 4 may be left out, press R/S only. In all routines blocks may overlap either way, but copying across a 4k boundary is "no go".

[c] COPY PAGE OF MICROCODE
Fill prompts for source page and destination page, R/S after each.
NB: Location 000 (XROM #) is not copied.

"MMC" uses R00 to R04. Uses but restores user flags 00 and 01; uses but does not restore user flags 08-10

After each move (all labels except [e]) "MMC" sends the program pointer to "0": re-XEQ "MMC" for another move.

“MMC” is a code mover, not a code relocator: post dependent jumps may require subsequent adjustment to work properly.

R>R files are put into, and taken from, R05 up.

When “MMC” puts R>R files into 8-bit RAM a bbb.eee control number is left in (X) for subsequent data recording via “WDTRX”, “WRTRX”, etc. Another copy of the control number may be found in R01.

ADDENDUM

ROAM-0 FOR THE PROTOCODER 1

ROAM-0 is a limited and slightly different version of ROAM-0A,

Flag 20 is used as a PROTOCODER 1 “write protect” on “MC”, and flag 19 for hexcode listing control.

ROMP? sets flag 8 for PROTOCODER and ROM, clears flag 8 for MLDL.

REGROMP will write to the PROTOCODER only from a running program. (Use REG>ROM to write MLDL from the keyboard.) Transfers to REG>ROM (in ASSEMBLER-3) if the destination is MLDL.

PCFC formats an address and word in (X) 6:0 ready for “SIGN” to write it to the PROTOCODER 1. If flag 8 is set the write will be done automatically. It is replaced in ROAM-0A with “WTMC”.

COPC and COPM are combined into COMC in ROAM-0A. COPM may be executed from the keyboard to copy to MLDL. COPC (which transfers to COPM if address is in MLDL) will write to the PROTOCODER only from a running program.

Inputs for REGROMP and COPC are the same for REGROMP and COPC in ROAM-0A, however ROAM-0 alters these in the stack when executed, ROAM-0A does not.

READROM works only for MLDL in ROAM-0. (RDROMP in ROAM-0A works for both MLDL and PROTOCODER 2). ROAM-0 requires hex page numbers in (X) for WRTROM & READROM. ROMCHKX in ROAM-0 is identical to ROMCHKS in ROAM-0A,