MultiMod Programming Pad

This document contains a blank summary programming page to plan and document a ROM layout in your MultiMod ROM emulator. Examples are provided to guide your own layout.

The first example is the layout and commands for J-F's Ultimate ROM. This example illustrates how the ROM structure description consists of different size ROM chips from the HP71B perspective. Each "chip" is the same size as the ROM image it holds, even though it maps to one or more internal 16KB flash block.

The second example shows how you can mix chip sizes, as well as create a sequence of 16KB chips to hold a large ROM (64KB C71 ROM in this case). A hard ROM is not needed so the space dedicated to its use can be repurposed to hold other ROMs.

At the end is a blank programming sheet that you can print and fill out for your own layout. The programming sheet consists of three tables. The first table is where you list the ROMs you want your MultiMod to host. You would normally have ROMs that total no more than seven 16KB "chips" but an 8KB ROM can be stored in the top half of flash block 0. It is marked as a 16KB ROM in this and subsequent blocks, even though only 8KB of content can be held.

The second block is the layout of the ROM Enumeration Table as it appears in MultiMod SRAM. Filling this out is somewhat optional, but very useful in visualizing what you are actually presenting to the HP-71B O/S during its startup processing. It also serves as a check on your ROM structure prior to filling out the last two tables.

The last two tables are the serial monitor commands that you use to define the ROM structure and upload the ROM image data. The commands used are separated into two tables because you can often reuse the same structure to hold different ROM content. For example, replacing one ROM in the Ultimate ROM layout would only require the ERASE command and the IMAGE command to upload the new ROM image.

Naming and keeping a sheet for each ROM layout you create is a helpful reminder of what you have, or what you'd like to recreate. Build your own set of favorite layouts and quickly switch from one to the other using just a few serial monitor commands!

MultiMod Programming Pad

Name: J-F's Ultimate ROM using variable-sized chips

Content:

ROM Image File	ROM Size	# of 16KB Chips
HP-82478A_FORTH-ASSEMBLER_ROM.BIN	16KB	1
HP-82478A_FORTH-ASSEMBLER_HRD-FIX.BIN	32KB	2
MATH2B.BIN	32KB	2
JPCF05.BIN	32KB	2
ULIB52.BIN	8KB	1

Table Layout

ROM Image Filename	**Size**	**Resv**	**Type**	**Class**	**Last**	Flag	Addr	Chip	Line
forth1b.dat	0x0a	0x00	0x01	0x00	0x08	0	0x00	1	1
math2b.dat	0x09	0x00	0x01	0x00	0x08	0	0x00	2	2
jpcf05.dat	0x09	0x00	0x01	0x00	0x08	0	0x00	4	3
ulib52.dat	0x0a	0x00	0x01	0x00	0x08	1	0x00	0	4
empty	0x0a	0x00	0x01	0x00	0x08	0	0x00	5	5
forth1bhrd.dat	0x0a	0x00	0x01	0x00	0x08	2	0x00	6	6
forth1bhrd.dat	0x0a	0x00	0x01	0x00	0x08	2	0x00	7	7
MMIO Address	0x00	0x00	0x00	0x0C	0x02	0x00	0x00	0x00	8

Field	Data	Desc
Size	0x0A	16KB
	0x09	32KB
	0x08	64KB
Last	0x00	No
	0x08	Yes
Flag	0x00	Cont.
	0x01	Stop
	0x02	H-Rom

Command				**Display**	**Content**
R	1	1	1	ROM 1 16K 1	forth1b
R	2	3	2	ROM 1 32K 2	math2b
R	3	3	4	ROM 1 32K 4	jpcf05
R	4	1	0	ROM 1 16K 0	ulib52
R					
R					
R					

ROM Structure Commands

Command		**Display**
L	4	Last 4
H	Y	Hard Yes

Command		**Display**
E	0	Erase 0
E	1	Erase 1
E	2	Erase 2
E	3	Erase 3
E	4	Erase 4
E	5	Erase 5
E	6	Erase 6
E	7	Erase 7

ROM Content Commands

Command		**Display**	**File Upload**
I	0	IMAGE 0	ulib52.dat
I	1	IMAGE 1	forth1b.dat
I	2	IMAGE 2	math2b.dat
I	4	IMAGE 4	jpcf05.dat
I	6	IMAGE 6	forth1bhrd.dat
I			
I			
I			

MultiMod Programming Pad

Name: Example using a multi-chip sequence, no hard ROM

Content:

ROM Image File	ROM Size	# of 16KB Chips
MATH2B.BIN	32KB	2
HP-82485A_TEXT-EDITOR_ROM.BIN	16KB	1
C71.BIN	64KB	4
ULIB52.BIN	8KB	1

Table Layout

ROM Image Filename	Size	Resv	Type	Class	Last	Flag	Addr	Chip	Line
math2b.dat	0x09	0x00	0x01	0x00	0x08	0	0x00	1	1
text-edit.dat	0x0a	0x00	0x01	0x00	0x08	0	0x00	3	2
c71.dat	0x0a	0x00	0x01	0x00	0x00	0	0x00	4	3
c71.dat	0x0a	0x00	0x01	0x00	0x00	0	0x00	5	4
c71.dat	0x0a	0x00	0x01	0x00	0x00	0	0x00	6	5
c71.dat	0x0a	0x00	0x01	0x00	0x08	0	0x00	7	6
ulib52.dat	0x0a	0x00	0x01	0x00	0x08	1	0x00	0	7
MMIO Address	0x00	0x00	0x00	0x0C	0x02	0x00	0x00	0x00	8

Field	Data	Desc
Size	0x0A	16KB
	0x09	32KB
	0x08	64KB
Last	0x00	No
	0x08	Yes
Flag	0x00	Cont.
	0x01	Stop
	0x02	H-Rom

Command				Display	Content
R	1	3	1	ROM 1 32K 2	math2b
R	2	1	2	ROM 2 16K 2	text-edit
R	3	c	4	ROM 3 CHIP 4	c71
R	4	c	5	ROM 4 CHIP 5	c71
R	5	c	6	ROM 5 CHIP 6	c71
R	6	1	7	ROM 6 16K 7	c71
R	7	1	0	ROM 7 16K 0	ulib52

ROM Structure Commands

Command		Display
L	7	Last 7
H	n	Hard No

Command		Display
E	0	Erase 0
E	1	Erase 1
E	2	Erase 2
E	3	Erase 3
E	4	Erase 4
E	5	Erase 5
E	6	Erase 6
E	7	Erase 7

ROM Content Commands

Command		Display	File Upload
I	0	IMAGE 0	ulib52.dat
I	1	IMAGE 1	math2b.dat
I	2	IMAGE 3	text-edit.dat
I	4	IMAGE 4	c71.dat
I			
I			
I			
I			

MultiMod Programming Pad

Name: _____

Content:

ROM Image File	ROM Size	# of 16KB Chips

Table Layout

ROM Image Filename	Size	Resv	Type	Class	Last	Flag	Addr	Chip	Line
		0x00	0x01	0x00			0x00		1
		0x00	0x01	0x00			0x00		2
		0x00	0x01	0x00			0x00		3
		0x00	0x01	0x00			0x00		4
		0x00	0x01	0x00			0x00		5
		0x00	0x01	0x00			0x00		6
		0x00	0x01	0x00			0x00		7
MMIO Address	0x00	0x00	0x00	0x0C	0x02	0x00	0x00	0x00	8

Field	Data	Desc
Size	0x0A	16KB
	0x09	32KB
	0x08	64KB
Last	0x00	No
	0x08	Yes
Flag	0x00	Cont.
	0x01	Stop
	0x02	H-Rom

Command				Display	Content
R					
R					
R					
R					
R					
R					
R					

ROM Structure Commands

Command		Display
L		
H		

Command		Display
E		Erase
E		Erase
E		Erase
E		Erase
E		Erase
E		Erase
E		Erase
E		Erase

ROM Content Commands

Command		Display	File Upload
I		IMAGE	
I		IMAGE	
I		IMAGE	
I		IMAGE	
I		IMAGE	
I			
I			
I			