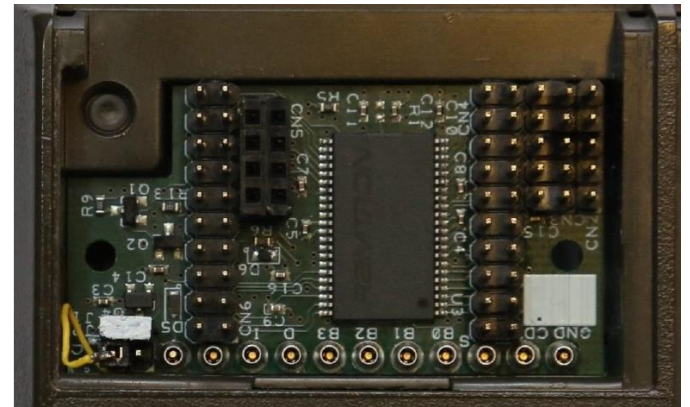# Introduction to FRAM71

How much RAM can a FRAM CRAM?
(into an HP-71B)

Bob Prosperi

Overview

- What is FRAM71 and what it's not
- How much RAM in a FRAM? And in an HP-71B?
- How to configure FRAM71
- Sample Configuration Worksheet
- Tricks and Special Features
- Hardware Config Options
- How does it work?
- Updating
- History
- Recent Community Activity and Future

What is FRAM71?

- A memory expansion board for the HP-71B, utilizing <u>F</u>erroelectric <u>RAM</u> for non-volatility
- Conceived, designed, made by Hans Brueggemann
- A great value! (worth ~$11M @ cur '71 RAM prices)
- Mostly invisible – adds no specific functionality
- FRAM71 is Awesome!
- What's it for?
  - Maximize 71B memory available for Programs and Data
  - Host ROM images, including Hard-Configured ROMs
  - Upgrade your 71B OS, Run Diagnostics
  - "Replace" defective ROM or RAM chips

What FRAM71 Is Not

- Not Difficult to configure or use

- Not For speed-up 71Bs

- Does not contain library of application ROMs

- Does not replace, speed-up, or enhance CPU or OS

- No harder to use than EMU/71 ROMs

	41CL	FRAM	CLONIX
CPU Speedup	X		
User RAM	X	X	(HEPAX)
Host ROMs	X	X	X
NV User RAM	X	X	
ROM Library	X		
OS Enhanced	X		

How much RAM in a FRAM?

- 16 x 32KB FRAM modules (F-Blocks) => 512KB (71 is a nibble-oriented machine, so 1MN - MegaNibble)

- 512KB possible, but not really (384KB useful)

- Build Memory Modules out of emulated chips
  - Ex: 4 x 32KB FB combine into a single 128KB MM

- One full F-Block for each chip regardless of size (so an 8KB MM costs a full 32KB F-Block)

- But that's OK … there is ultimately more FRAM than you can CRAM

Twice the FRAM, for those that <u>really</u> CRAM

- Current FRAM71 can be upgraded to 1MB!
- Supported by Hans' design, but not currently offered for sale in this configuration
- Add'l 512KB chip soldered over 1st chip (skill!)
- 2 Complete 71B machines, each w/full 512KB
- Selected/isolated by jumper
- 2 HP-71Bs – 1 for use, 1 to experiment… safely

HP-71B Memory Map

Address Block (32KB)	Use
00000	71B OS
10000	71B OS
20000	Vid Ram, System RAM values
30000	Avail
40000	Avail
50000	Avail
60000	Avail
70000	Avail
80000	Avail
90000 – D0000	Avail
E0000	HC ROMs
F0000-FFBFF	N/A if HC ROM present (DBGR)
FFC00-FFFFF	System Config Area (N/A)

How do I configure FRAM71?

- Config string simply POKE'd into SYS Config area
- One nibble for size/type of chip and one nibble to specify which F-Block
- Ex: POKE "2C000","9394A500000000…00"
  - 2 32KB MM, 32KB using FB3, 32KB using FB4
  - 1 16KB MM using FB5
- New Ports seen as 5.0X (so here, 5.00, 5.01, 5.02)
- MM Size names (9, A above) from 71B OS Config
- IRAM and HC ROM created as RAM, loaded, then converted to ROM (e.g. 9 -> D)

Sample FRAM71 Configuration

- Plan your MM usage assuming you can "make" any size modules you want

- Assemble the commands, recalling* the 71B Config code allocates RAM (Lg->sm), then IRAM/ROM (Lg->sm)

  * http://www.hpmuseum.org/journals/71b.htm

- I want it all! IRAM for LEX & Utility files, IRAM/eROM for HP App ROMs, Custom ROM, Forth/Assembler ROM, and lots of user RAM!

Sample FRAM71 Config

| Title: FRAM71 Config for HP-71B Machine #1 (s/n 2405A00304), Bottom FRAM								
Date: 7/19/15								
POKE "2C000",		**"**	Comment	(LCIM)	Device Type	Size (kB)	Module PORT 5.xx
Chip_0*)	**2C000**	CONF	**9 (D)**	HC E0000 for Forth ROM	**1**	**ROM**	**32**	**n/a**
	2C001	F-BLOCK	**3**	Jumper CN2-5 to Protect E0000 as ROM				
Chip_1	**2C002**	CONF	**1**	System RAM		**RAM**	**32**	**5.00**
	2C003	F-BLOCK	**4**					
(SNIP)								
Chip_6	**2C00C**	CONF	**9**	System RAM	**1**	**RAM**	**32 (192)**	**5.00**
	2C00D	F-BLOCK	**9**					
Chip_7	**2C00E**	CONF	**9**	IRAM to hold LEX files	**1**	**RAM**	**32**	**5.01**
	2C00F	F-BLOCK	**A**					
Chip_8	**2C010**	CONF	**9**	IRAM to hold 32KB Application ROM image	**1**	**RAM**	**32**	**5.02**
	2C011	F-BLOCK	**B**					
Chip_9	**2C012**	CONF	**9 (D)**	IRAM to hold 32KB JPC ROM image	**1**	**ROM**	**32**	**5.03**
	2C013	F-BLOCK	**C**	Convert to ROM (9 => D) after loading				
Chip_A	**2C014**	CONF	**A (E)**	IRAM to hold 16KB SC Forth ROM	**1**	**ROM**	**16**	**5.04**
	2C015	F-BLOCK	**D**	Convert to ROM (A => E) after Loading				
Chip_B	**2C016**	CONF						
	2C017	F-BLOCK						
Chip_C	**2C018**	CONF						
	2C019	F-BLOCK						
Chip_D	**2C01A**	CONF						
	2C01B	F-BLOCK						
Chip_E	**2C01C**	CONF						
	2C01D	F-BLOCK						
Chip_F	**2C01E**	CONF						
	2C01F	F-BLOCK						
Note: Do not use F-block 2. *) for E0000, Chip_0 = 32 K		**"**	**32K:** D-9-5-1 **16K:** E-A-6-2 **8K:** F-B-7-3 LCIMRO-LCIMRA-RO-RA				

- POKE "2C000","931415161718999A9B9CAD0000000000"

Tricks (how to use side-effects as features)

- Bank-switching – load 2 ROM modules into different F-Blocks, then reconfigure – just chg one nibble. No restart required, so can be done "on-the-fly" under program control.

- FRAM71's best feature can also be its biggest problem. FRAM is Non-Volatile so even corrupted memory is preserved – INIT:3 may not restore.

  - Config memory to hide intact F-Blocks, config damaged F-Blocks as normal RAM.

  - Turn-on and let the 71 ROM System Init code initialize away the problem.

Special Features - SYSRAM

- Allows loading alternate OS ROM (run 2CDCC on any machine)
- Load 'TakeOver' ROMs – HP's 71 Diagnostic ROM
- Controlled by dedicated jumper – no permanent chg
- Poke the ROM contents into a dedicated F-Block
- Enabled by OD signal – floats internal ROM to let some "other" ROM run
- Hardware mod inside or special Port-1 jumper
- Modified 4KB RAM modules, custom module, and now 3D Printed Port cover
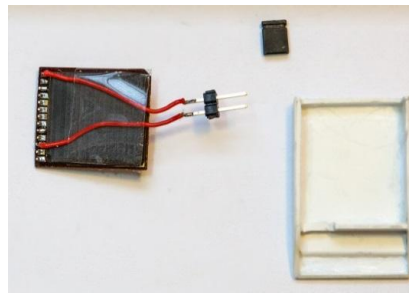
Special Features – HC ROMs

- Configure Chip_0 for special ROMs with fixed address (Hard Configured) ROM image

- FORTH/Assembler ROM and HP-41 Translator ROM

- Hard-Configured ROM (32KB) permanently mapped to E0000-EFFFF

- Both ROMs also have a normal (Soft-Configured) 16KB ROM as well (FORTHROM LEX, etc.)

- Config F-Block, write 32KB ROM image, set special jumper, pwr-cycle, load SC ROM

Hardware Configuration

- Jumpers do it all…

- Hard-Config E0000 ROMs

- Enable/Disable writing to SYSRAM

- Disable Bus Writes – for regaining control of locked-up configuration

- Disable output during Config - For Clearing all IRAMs / troubleshooting

- Select Top RAM – select 2nd FRAM 512KB block – 2 full 512KB HP-71B Memory sets

- Enable/Disable SYSRAM
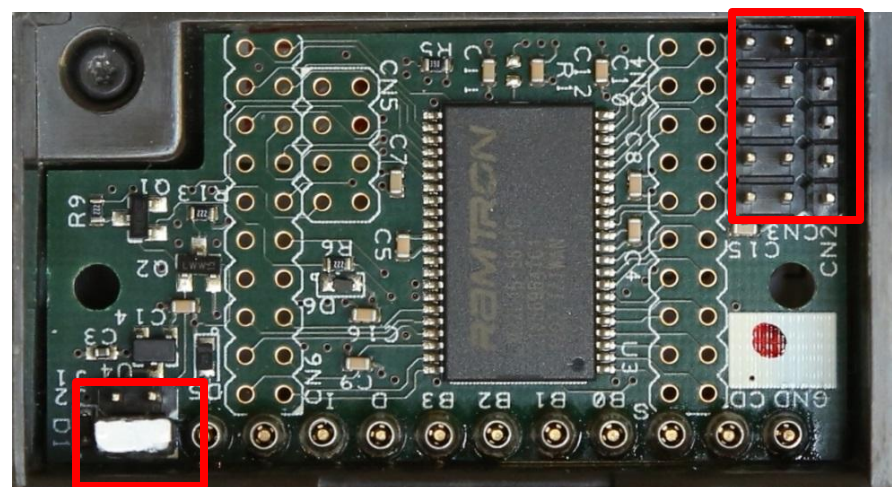
FRAM71 Block Diagram
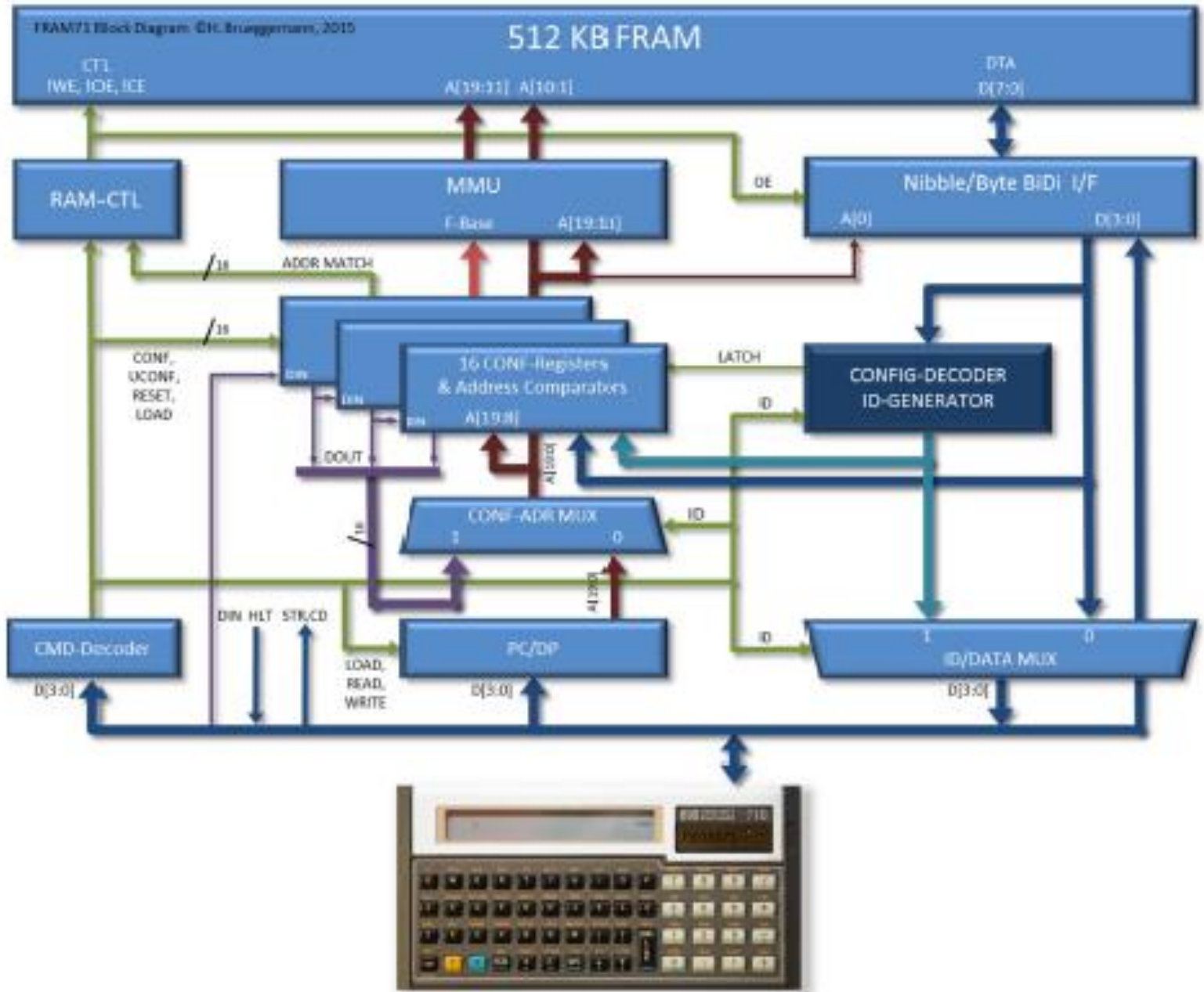
Updating FRAM71 Firmware

- Firmware can be updated - Bug fixes, new features, etc.

- Uses ACTEL FlashPro4/5 (~$25) w/custom cable

- Early batch machines should have header soldered-in, but still doable as-is

- Easy procedure, less than 10 minutes, usually with all memory contents intact!

Project History & Evolution

- 1/10 – Initial idea to create FPGA-based 32KB Front port module
- 1/11 – Successful FRAM Read/Write
- 8/11 – MATH ROM Loads OK
- 7/12 – Diag. ROM Loads OK
- 2/13 – FORTH ROM, SYSRAM OK
- 10/13 - Pwr. Sys. redesign Always-On
- 3/14 – 1st Prototype to field
- 6/14 – RedEye, UART working
- 10/14 – 1st batch sales begin
- 3/15 – 1MB & on-the-fly working
- 4/15 – 2nd batch sales begin
- 9/15 – IRAM Zombies finally killed

- Project log of 112 events (many bugs)
- 3 PCB iterations
- 50 early PCBs scrapped – layout bugs
- 3 early prototypes scrapped due to TX/RX weakness
- 1 production FRAM71 bad due to bad FRAM chip
- 1 FRAM71 killed by ESD
- Many bezel designs (hint – do not try cast resin. Ever. But it is pretty…)

Recent Community Activity

- FRAM71 Custom bezels
  - >20 Colors & materials, more coming
- OD Module Port Covers
  - 24 AWG wire in Port Cover with precise holes; custom labels
- Avail @ http://www.shapeways.com/shops/hpparts
- By Nate Martin (CAD) and Dave Frederickson (store)

FRAM71 Future…

- Need 30 orders to proceed (25 more please)
- Contact Hans Brueggemann, brueg@gmx.net
- Possible Future features – demand driven
  - UART - Serial com a-la RS-232C (demonstrated)
  - Infrared (demonstrated)
  - 1MB Upgrade board (vs. current challenging chip solder)
  - Additional interface details for custom add-ons
  - Top / Bottom bank-switching via program control (the ultimate machine backup?)