REX Overview

Initial Release

Steve Adolph

Oct 2008

What is REX?

REX:

a 1 MB Flash memory card for

- Model 100 / Tandy 102
- Tandy 200

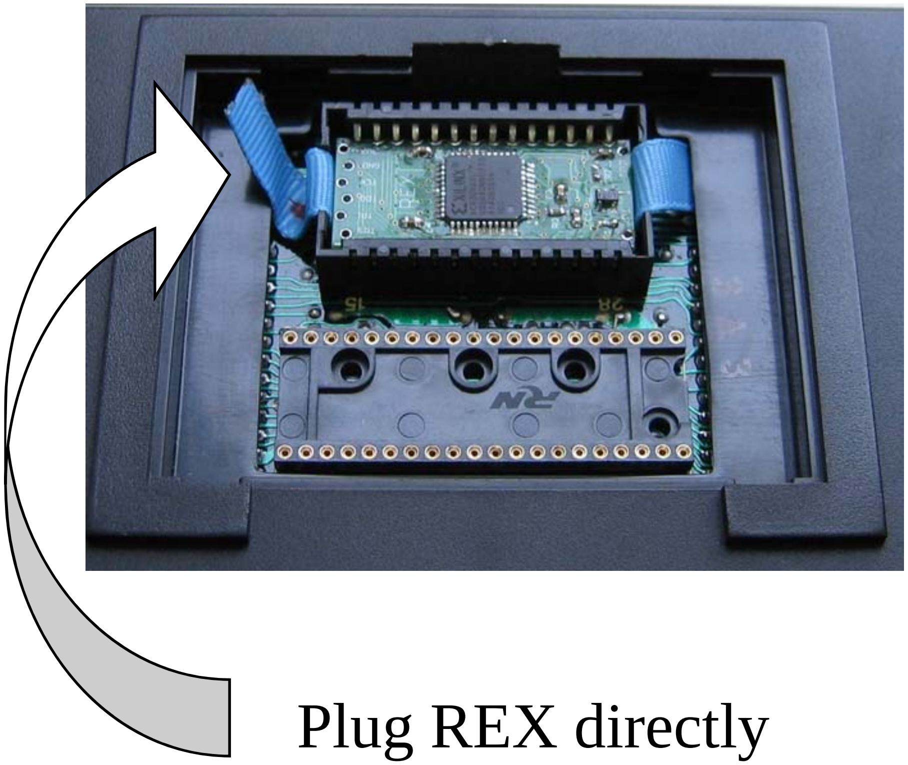
plugs into the OPTION ROM socket

eliminates the need for individual romware on a variety of modules.

provides new capabilities:

- Multi- romware access
- RAM image storage
- File storage

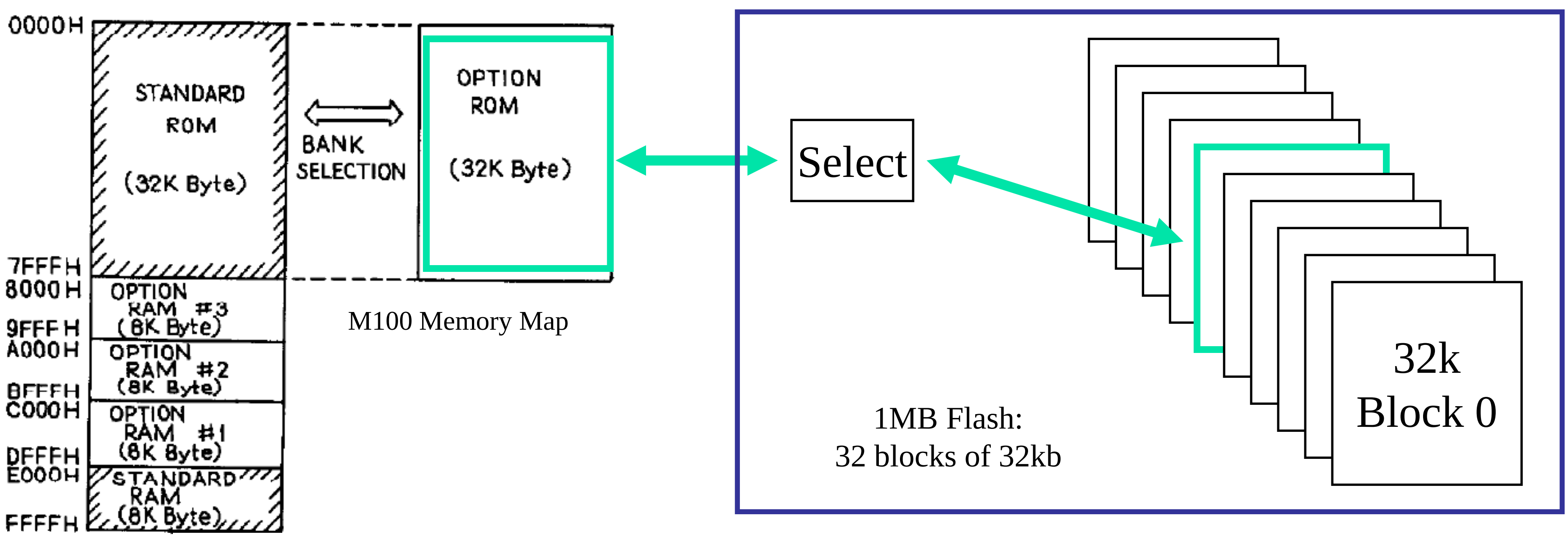
For advanced users, REX also replaces the standard ROM, allowing flash upgradeable main ROM.



Plug REX directly into the Option ROM socket-

NO WIRES!

How it works



M100 Memory Map

REX divides the 1MB of flash into 32 blocks of 32kb each

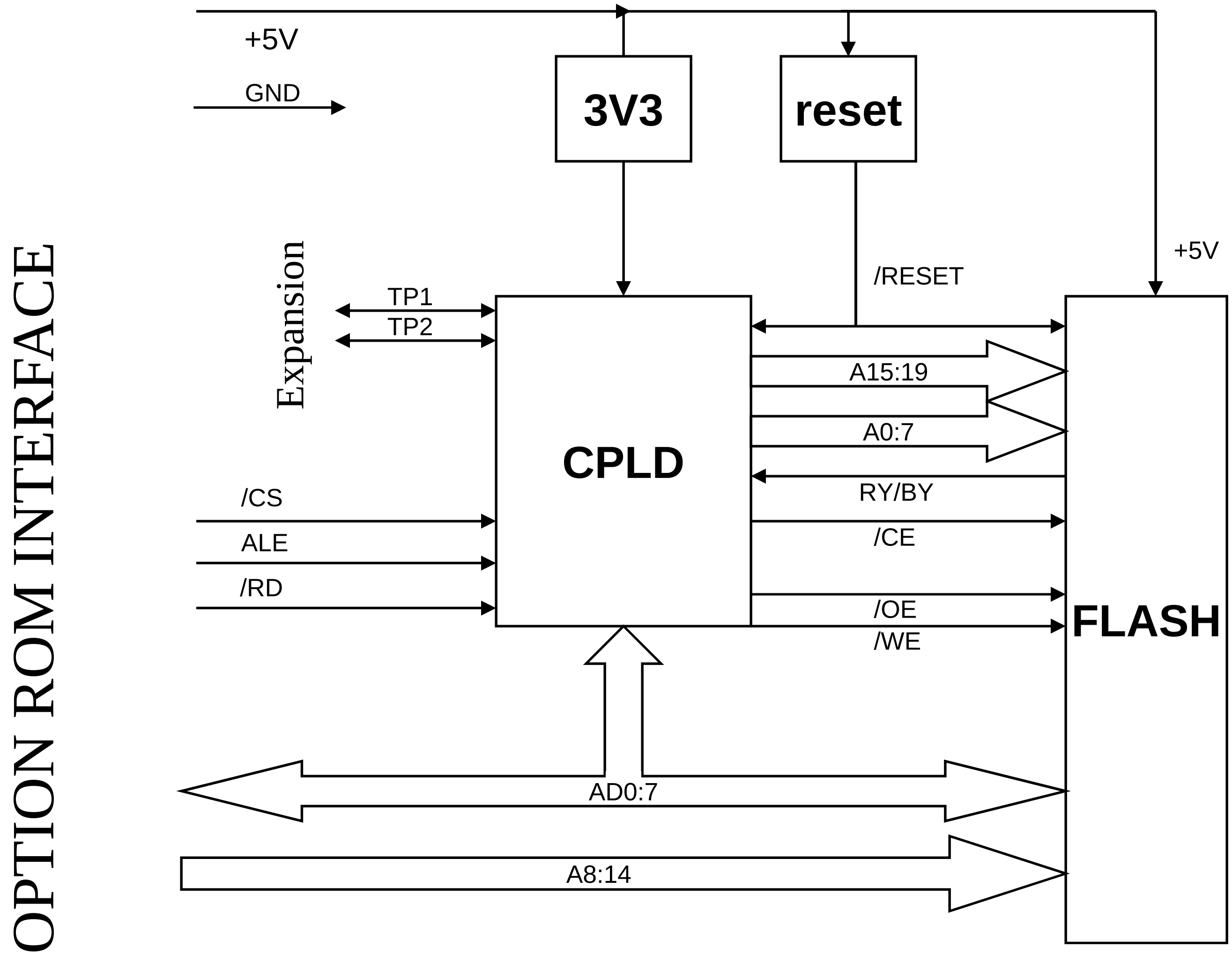
- user activates one block of 32k for access at any one time via the OPTROM bank

Once a block is activated:

- Data can be read normally from the flash, just like it was a ROM
- Using special commands, data can be written to the flash
- Blocks of flash can be erased and re-used as required

REX can be auto-configured on power up, so it is always the same way you left it.

REX Architecture



REX Physical Specifications

Physical Dimensions

- To fit within OPTION ROM socket

FLASH

- 1MB AM29F800B 5V FLASH

CPLD

- Xilinx Coolrunner XCR3064XL-10VQ44

Power Supplies

- CPLD: LP2980 (3.3 fixed, derived from system +5V)
- FLASH: system +5V (external to REX)

Power Consumption

- Negligible during reads, TBD during writes

REX Memory Assignment

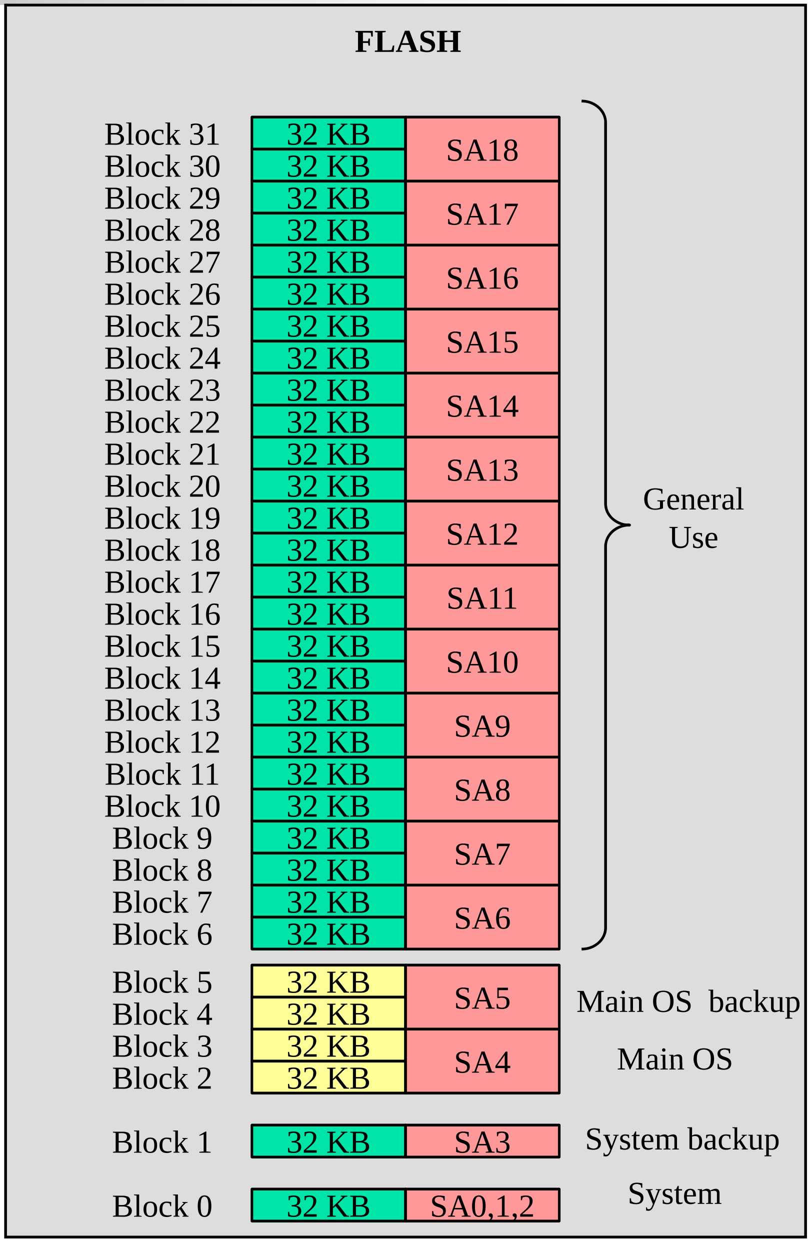
192k is reserved for system use

- 32k for REX Manager, 32k backup
- 128k for main ROM replacement

732k is free for image storage

- Images can be
  - OPTION ROMs
  - RAM backup images
  - Separate OS ROMs

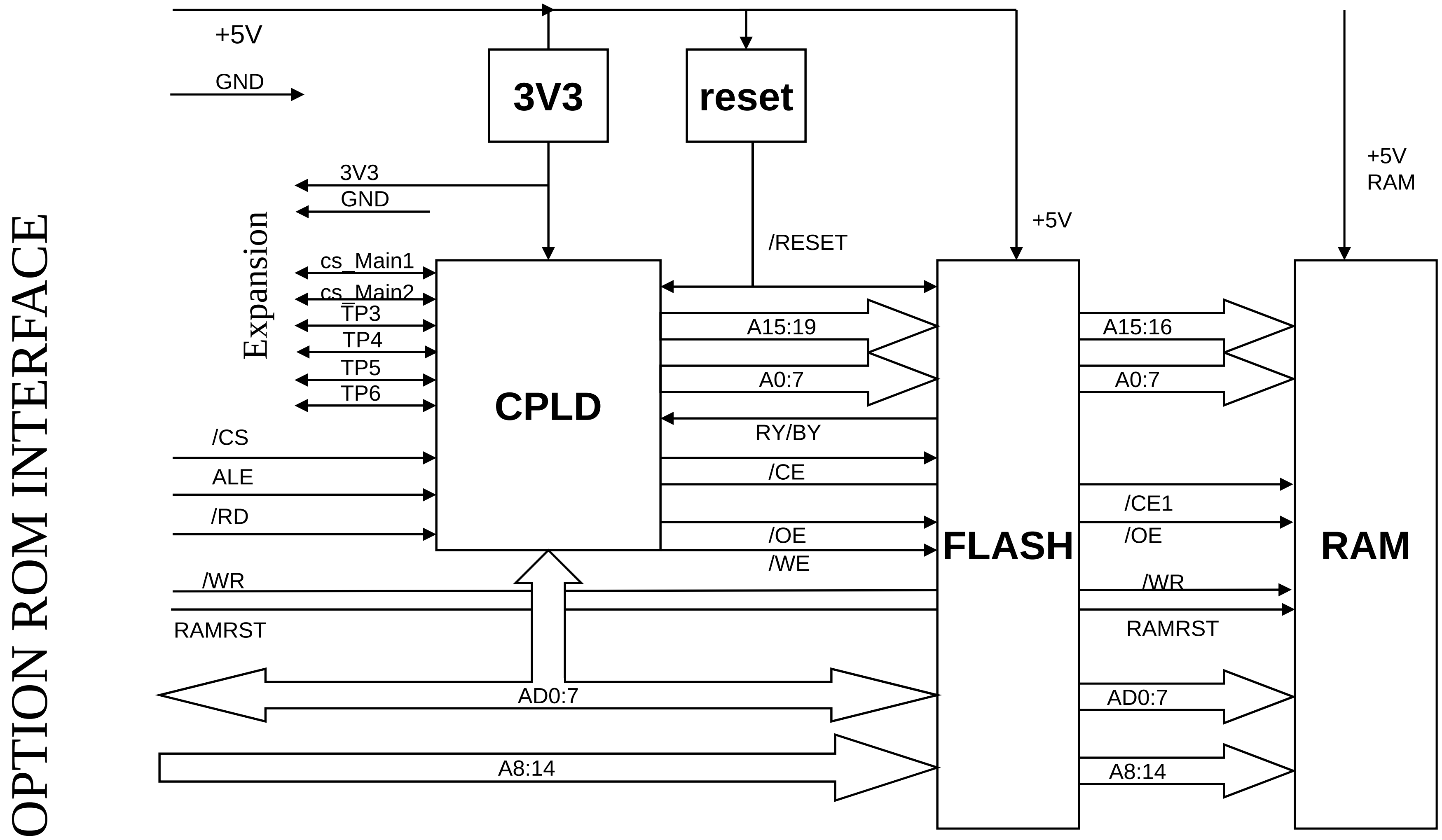
REX is shipped with REX Manager software installed in block 0 and block 1.



FLASH

Block 31	32 KB	SA18	General Use
Block 30	32 KB		
Block 29	32 KB	SA17	
Block 28	32 KB		
Block 27	32 KB	SA16	
Block 26	32 KB		
Block 25	32 KB	SA15	
Block 24	32 KB		
Block 23	32 KB	SA14	
Block 22	32 KB		
Block 21	32 KB	SA13	
Block 20	32 KB		
Block 19	32 KB	SA12	
Block 18	32 KB		
Block 17	32 KB	SA11	
Block 16	32 KB		
Block 15	32 KB	SA10	
Block 14	32 KB		
Block 13	32 KB	SA9	
Block 12	32 KB		
Block 11	32 KB	SA8	
Block 10	32 KB		
Block 9	32 KB	SA7	
Block 8	32 KB		
Block 7	32 KB	SA6	
Block 6	32 KB		
Block 5	32 KB	SA5	Main OS backup
Block 4	32 KB		
Block 3	32 KB	SA4	Main OS
Block 2	32 KB		
Block 1	32 KB	SA3	System backup
Block 0	32 KB	SA0,1,2	System

REX2 Architecture



REX2 Physical Specifications

Physical Dimensions

- To fit within option rom socket

FLASH

- 1MB AM29F800B 5V FLASH

CPLD

- Xilinx Coolrunner XCR3064XL-10VQ44

RAM (REX2 only)

- Toshiba TC551001 5V SRAM

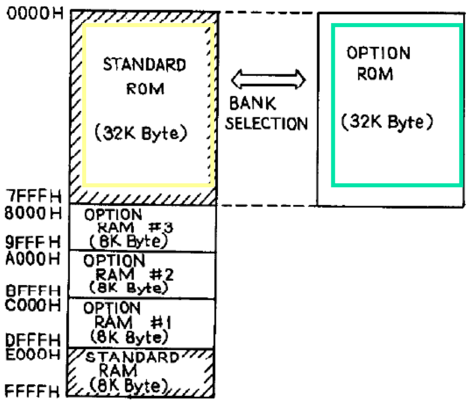
Power Supplies

- CPLD: LP2980 (3.3 fixed)

Power Consumption

- Negligible during reads, TBD during writes

REX2 Memory Assignment

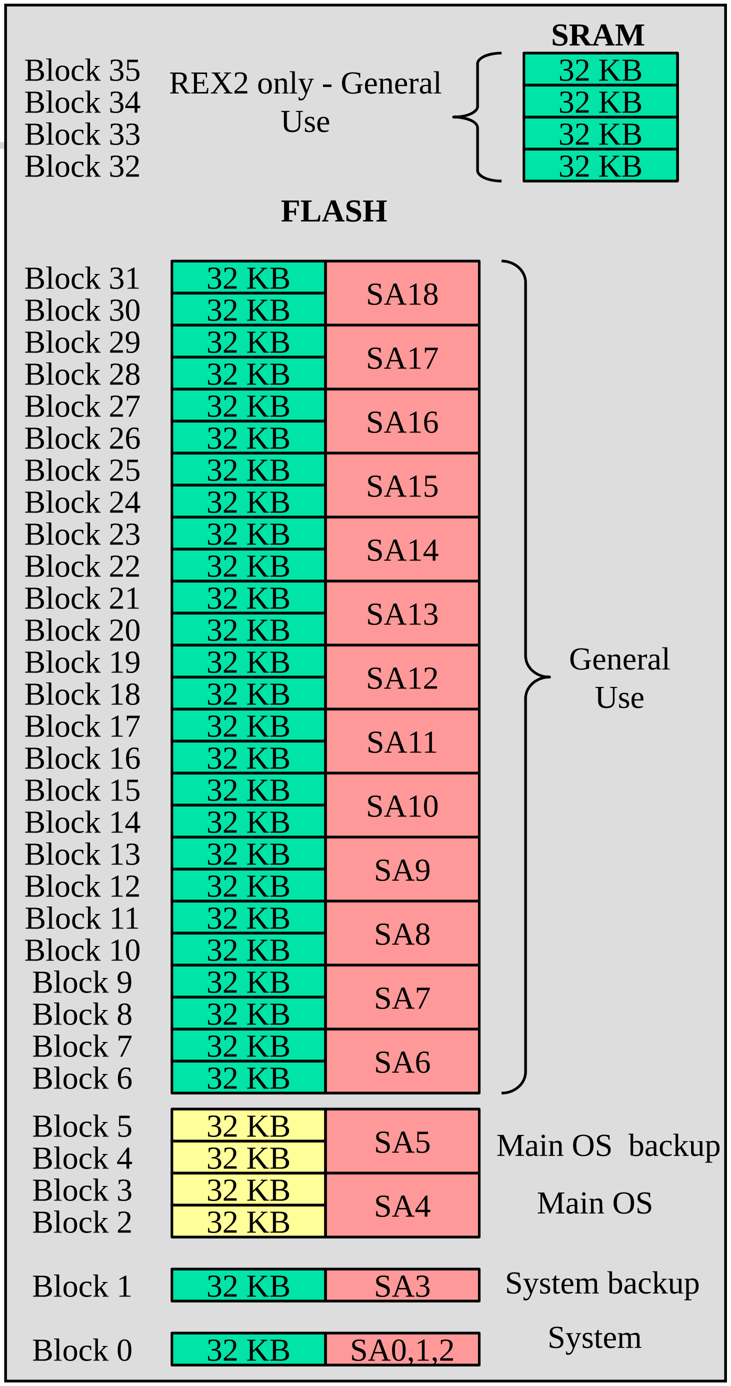


M100 Memory Map

- Green = 32k visible in option rom bank
- Yellow = 32k visible in main or option rom bank

- REX/REX2: Flash allocation
  - 32k for system use, 32k backup
  - 64k for main rom replacement OS1, 64k for OS2
  - 26 x 32k for general use
    - Option ROMs, RAM images

- REX2 SRAM allocation
  - 4 x 32k for general use
    - Files, images, roms etc.



		SRAM
Block 35	REX2 only - General Use	🟩 32 KB
Block 34		🟩 32 KB
Block 33		🟩 32 KB
Block 32		🟩 32 KB

FLASH

Block 31	🟩 32 KB	SA18	General Use
Block 30	🟩 32 KB		
Block 29	🟩 32 KB	SA17	
Block 28	🟩 32 KB		
Block 27	🟩 32 KB	SA16	
Block 26	🟩 32 KB		
Block 25	🟩 32 KB	SA15	
Block 24	🟩 32 KB		
Block 23	🟩 32 KB	SA14	
Block 22	🟩 32 KB		
Block 21	🟩 32 KB	SA13	
Block 20	🟩 32 KB		
Block 19	🟩 32 KB	SA12	
Block 18	🟩 32 KB		
Block 17	🟩 32 KB	SA11	
Block 16	🟩 32 KB		
Block 15	🟩 32 KB	SA10	
Block 14	🟩 32 KB		
Block 13	🟩 32 KB	SA9	
Block 12	🟩 32 KB		
Block 11	🟩 32 KB	SA8	
Block 10	🟩 32 KB		
Block 9	🟩 32 KB	SA7	
Block 8	🟩 32 KB		
Block 7	🟩 32 KB	SA6	
Block 6	🟩 32 KB		
Block 5	🟨 32 KB	SA5	Main OS backup
Block 4	🟨 32 KB		
Block 3	🟨 32 KB	SA4	Main OS
Block 2	🟨 32 KB		
Block 1	🟩 32 KB	SA3	System backup
Block 0	🟩 32 KB	SA0,1,2	System

The ACTIVE block

What is the ACTIVE Block?

- Any of the 32 blocks of flash memory can be “active” – that means that the block is currently visible in the option ROM bank.
- When a block is “active”, the 32k binary image stored in that block looks exactly as if an option ROM with that image was plugged in.
- In this way REX allows the user to have access to all 1MB of the flash chip

Setting or changing the ACTIVE Block:

- The user can define what block is “active” IE “in use”, using the integrated REX Manager software.

POWER UP:

- When the laptop is powered up, REX defaults to block 0 always.
- Auto-activation: REX installs some special software in the laptop that redirects the power up routines to REX Manager, which searches for the active block and sets it up for the user, on every power up.

Installing REX/REX2

Installing REX

REX installs in the OPTION ROM socket just like any other ROM.

Three things are supplied with REX-

- REX card
- Ribbon (used to extract REX if need be)
- Balsa spacer (used to guarantee REX seats properly in the socket

Installing REX: Step 1

Insert balsa spacer gently GENTLY into the OPTION ROM socket. You may have to press against the springy contacts of the socket to get the spacer to sit flat against the bottom of the socket.

Installing REX: Step 2

Lay the ribbon on top of the balsa spacer such that it is evenly centered in the socket, and the tails of the ribbon are even on both sides of the socket.

Installing REX: Step 3

REX is placed in the ROM socket with the flash chip hidden, and the CPLD visible. You must orient REX correctly inside the socket. Look closely at the OPTROM socket, and you will see that one end of the socket has a single vertical notch in the center of one end, and the other has a dual vertical notch.

Installing REX: Step 3

REX has a single white dash marking on the CPLD side on one end (opposite end from the row of six holes). This end (which also has 2 holes marked TP1 and TP2) must line up with the single notch of the socket, such that the white dash marking lines up with the single notch.

Installing REX: Step 3

Once you understand how REX needs to be oriented in the socket, gently place REX into the OPTROM socket so that it is resting evenly on the springy contacts. Now, firmly press REX downward so that it evenly and flatly sits on top of the spacer and the ribbon, and all springy contacts are evenly contacting REX.

Initializing REX

To initialize the environment, REX Manager’s initialization routine must be started. The initialization sequence is as follows-

First, REX Manager boot up software is installed.

- This software uses the timer hook to re-direct every power up to the REX Manager application
- This allows the active block to be detected and provisioned automatically
- A hidden file RXHOOK is injected as the first basic program
- REXMGR.BA is injected, and allows manual starting of REX Manager

Initialization from a COLD RESTART:

- Enter BASIC
- Type

  CALL 63012 <enter>
- You will be directed to the initialization routine

Re-initializing the environment (if needed):

- Everytime REX Manager runs, it verifies that it is properly installed, and it re-installs all files and the timer hook settings needed.

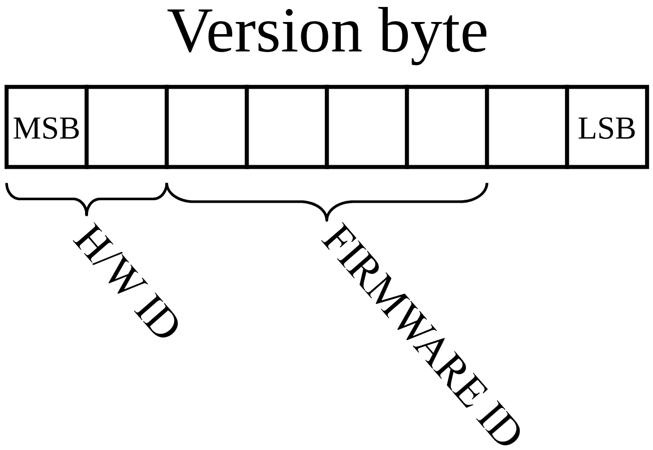
Starting REX Manager

Starting REX Manager:

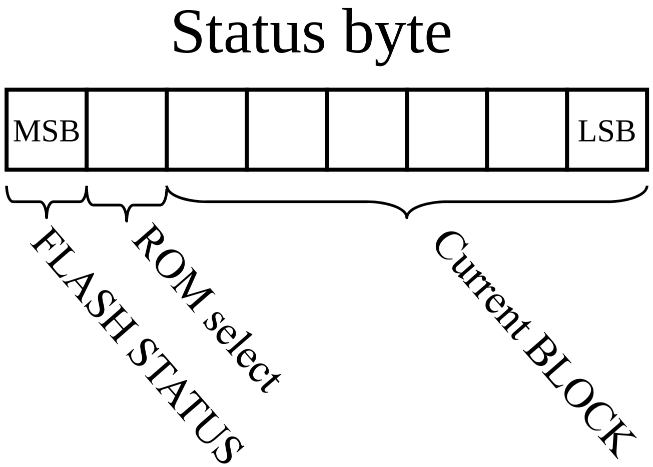
- Load and run the basic program REXMGR.BA
- You will be directed to REX Manager

Additional REX information

REX Registers

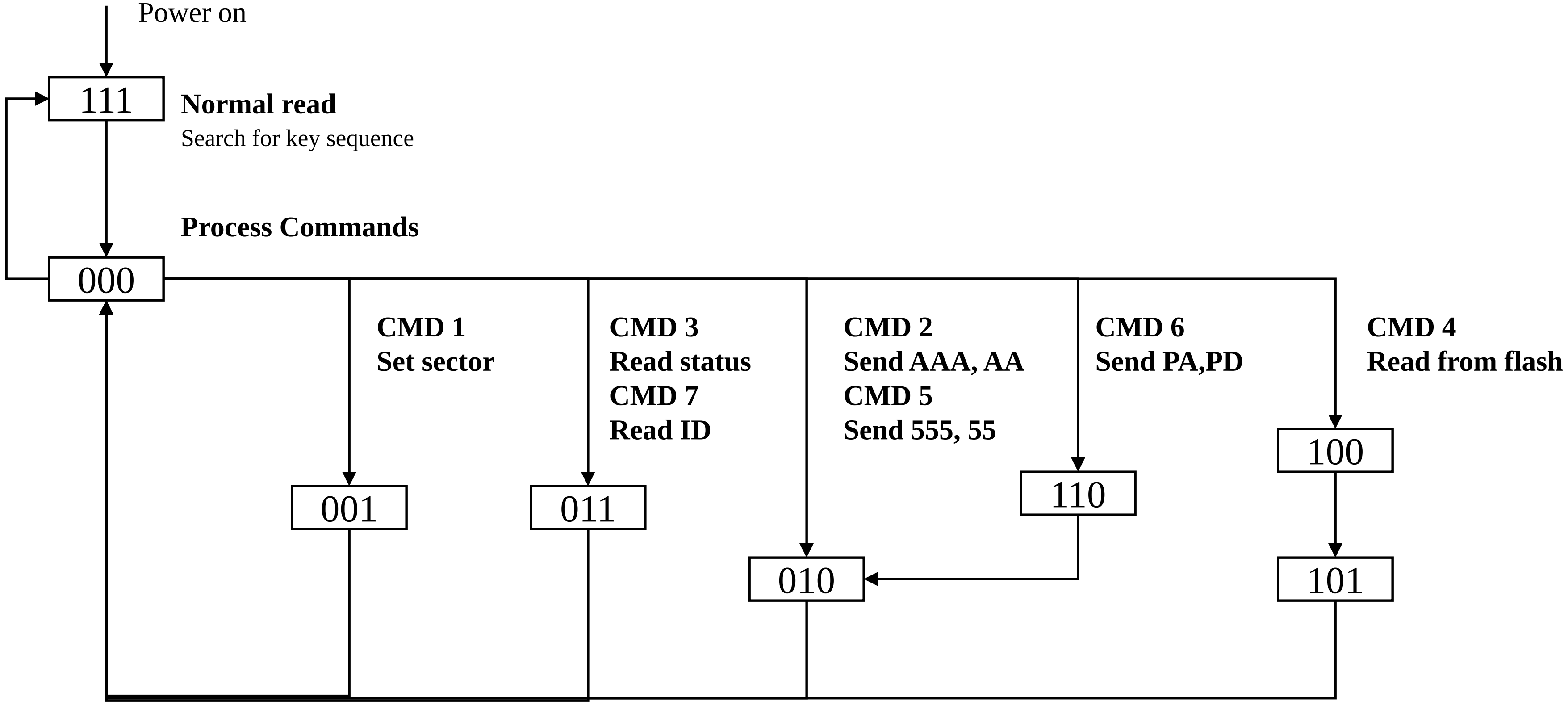


- This register is accessed (read) by writing the unlock key to REX, and reading back the version byte.
- The version byte provides-
  - Bits 7-6: hardware version
  - Bits 5-2: firmware version
  - Bits 1, 0: internal use only



- This register is accessed (read) by writing the unlock key to REX, and reading back the status byte.
- The status byte provides-
  - MSB: status of the flash chip – 1 = ready, 0 = busy
  - Bit 6: ROM select for main rom
  - Bits 5-0: 6 bit block selection – the block currently in use is indicated here (0-35)

REX state machine



Power on

111

Normal read

Search for key sequence

Process Commands

000

CMD 1
Set sector

CMD 3
Read status
CMD 7
Read ID

CMD 2
Send AAA, AA
CMD 5
Send 555, 55

CMD 6
Send PA,PD

CMD 4
Read from flash

State machine advances on falling edge of ALE when OPTROM is selected.
CPLD is reset by first rising ALE edge.

REX Manager software

REX has it’s own management software built in – REX Manager

- This software is provided as a option rom, and is loaded into the default power up block, block 0.
- It is also backed up into block 1.
- REX Manager comes in 2 versions – M100/T102 and T200 – they are not interchangeable, however one may reprogram the software to suit.

REX Manager (Release 4.2) supports

- Select – user identifies a block to process
- Activate – user identifies a block to be “in use”
- Copy – copy an image from “selected” block to another block
- Erase – erases the “selected” block
- Program – programs the “selected” block from a 32k binary file stored on TPDD
- Name – assign a name to the “selected” block

REX Manager keeps track of the “active” block in a 1kb directory, and the names of all the blocks in another 3kb directory.

- Directories are stored in block 0, backed up in block 1.
- 1000 active block assignments can be made before the directory needs to be rebuilt
- 368 block name assignments can be made before the directory needs to be rebuilt
- Directory rebuilding is an automatic function, transparent to the user.

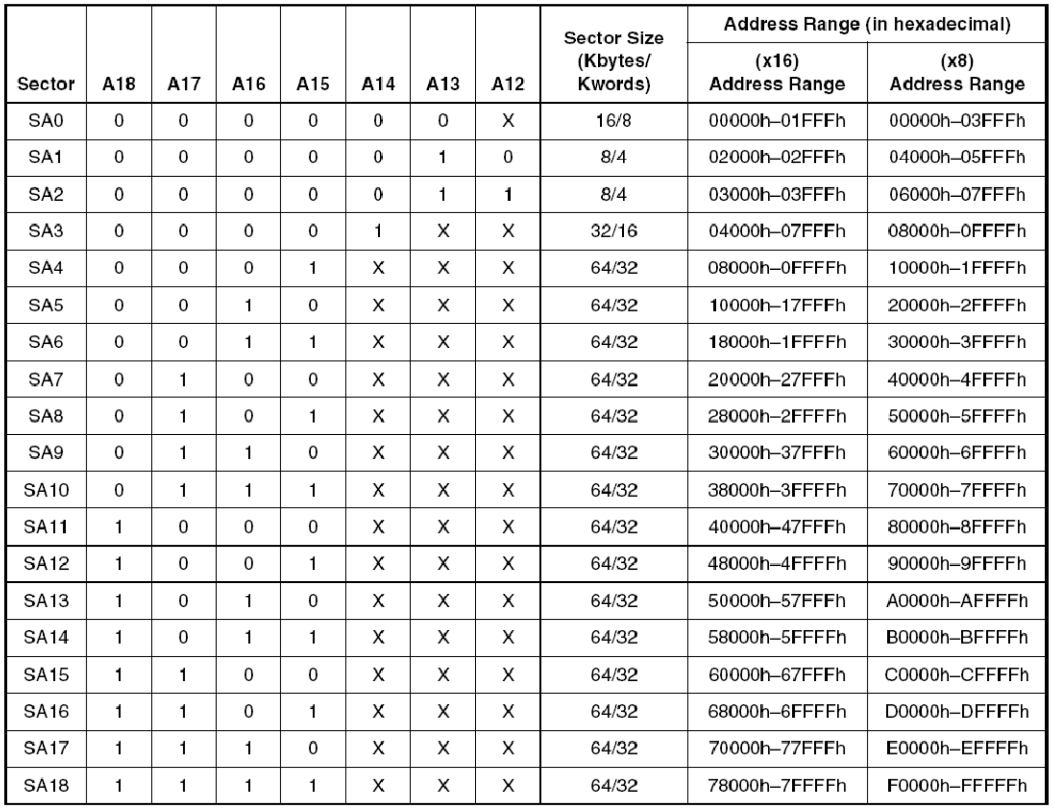
Flash organization

- The AM29F800B flash chip is designed with 19 SECTORS
  - A sector is defined as an independent, erasable area within the flash
  - The entire chip can be erased, or a single sector can be erased
- There are 4 small sectors
  - SA0 – 16 kbytes
  - SA1 – 8 kbytes
  - SA2 – 8 kbytes
  - SA3 – 32 kbytes
- The remaining sectors are 64 kbytes

- This poses a challenge to the M100 user. A rom image is typically 32kb, but the sector size is 64 kb. So, erasing one 32k block requires erasing another adjacent 32k block, and re-writing the contents later.

- In general, to program a rom, one must
  - Copy the other ½ of the sector to another sector
  - Erase the sector
  - Program the new rom image to half of the sector
  - Reprogram the other half of the sector

Table 3. Am29F800BB Bottom Boot Block Sector Address Table



Sector	A18	A17	A16	A15	A14	A13	A12	Sector Size (Kbytes/ Kwords)	Address Range (in hexadecimal)	
									(x16) Address Range	(x8) Address Range
SA0	0	0	0	0	0	0	X	16/8	00000h–01FFFh	00000h–03FFFh
SA1	0	0	0	0	0	1	0	8/4	02000h–02FFFh	04000h–05FFFh
SA2	0	0	0	0	0	1	1	8/4	03000h–03FFFh	06000h–07FFFh
SA3	0	0	0	0	1	X	X	32/16	04000h–07FFFh	08000h–0FFFFh
SA4	0	0	0	1	X	X	X	64/32	08000h–0FFFFh	10000h–1FFFFh
SA5	0	0	1	0	X	X	X	64/32	10000h–17FFFh	20000h–2FFFFh
SA6	0	0	1	1	X	X	X	64/32	18000h–1FFFFh	30000h–3FFFFh
SA7	0	1	0	0	X	X	X	64/32	20000h–27FFFh	40000h–4FFFFh
SA8	0	1	0	1	X	X	X	64/32	28000h–2FFFFh	50000h–5FFFFh
SA9	0	1	1	0	X	X	X	64/32	30000h–37FFFh	60000h–6FFFFh
SA10	0	1	1	1	X	X	X	64/32	38000h–3FFFFh	70000h–7FFFFh
SA11	1	0	0	0	X	X	X	64/32	40000h–47FFFh	80000h–8FFFFh
SA12	1	0	0	1	X	X	X	64/32	48000h–4FFFFh	90000h–9FFFFh
SA13	1	0	1	0	X	X	X	64/32	50000h–57FFFh	A0000h–AFFFFh
SA14	1	0	1	1	X	X	X	64/32	58000h–5FFFFh	B0000h–BFFFFh
SA15	1	1	0	0	X	X	X	64/32	60000h–67FFFh	C0000h–CFFFFh
SA16	1	1	0	1	X	X	X	64/32	68000h–6FFFFh	D0000h–DFFFFh
SA17	1	1	1	0	X	X	X	64/32	70000h–77FFFh	E0000h–EFFFFh
SA18	1	1	1	1	X	X	X	64/32	78000h–7FFFFh	F0000h–FFFFFh

SECTOR == memory regions defined in flash device

BLOCK == a 32 kbyte region mapped into the option rom memory, usable by the laptop

Backup

REX vs. REX2

REX (available now) has a single 1 MB flash chip.

REX2 (not available yet) incorporates a 128 KB static RAM chip as well as the 1MB flash chip.

- This increases the # of blocks from 32 to 36 (blocks 33-36 are SRAM).

Benefits:

- Additional memory for general use
- Supports all-ram operation (CP/M!)
- Supports ram based file storage.

SRAM access in REX2 requires installation of 3 additional wires

- Vram, /WR and RAMRST

REX Installation Levels

LEVEL 1: Wire Free basic installation

- Standard installation, all base features
- Simple plug and play install
- Valid for all firmware codes

LEVEL 2: ROM Replacement

- Allows the main rom to be provided by REX
- Requires 1 (M100/T102) or 2 (T200) wires to be installed
- The original main rom(s) must be removed or disabled
  - On M100/T200, the rom chip(s) can simply be unplugged
  - On T102, the rom chip must be removed or disabled

REX2 Overview

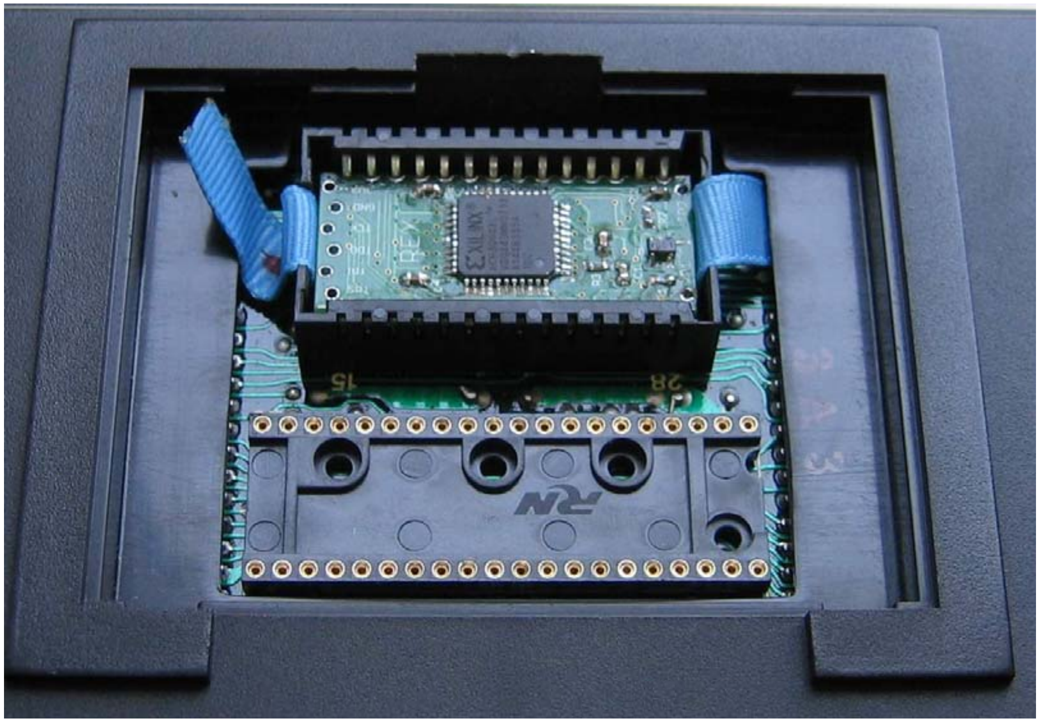
REX2 is a FLASH card that plugs into the option rom socket on M100, T102 and T200 laptops.

This upgrade provides Memory expansion via the option ROM bank

- 1 Mb of flash memory
- REX2 only: 128 kb of SRAM
- Memory used for
  - multiple rom storage
  - Flash File storage
  - System image backups
  - *Upgradeable main rom
  - *Rom development
  - *Ram file storage

REX2 also provides an expansion capability

- Power/ground + 4 unused I/O



* Level 2 and 3 installation features

REX Installation Requirements

Model	Level 1 install: Base features	Level 2 install: Rom Replacement	Wires needed
REX	●		No wires needed
REX		●	1 (M100/T102) 2 (T200)

LEVEL 1: Wire Free basic installation

- Standard installation, all base features
- Simple plug and play install

LEVEL 2: ROM Replacement

- REX provides the main ROMs
  - Two OS images are supported
- Requires some hardware modification
- The original main ROM(s) must be removed or disabled
  - M100 - unplug the ROM
  - T200 - unplug the ROMs
  - T102 - the ROM must be removed or disabled

REX Versions

Version 1: REX

- has a CPLD and a flash chip for a total of 2 ICs
- Hardware ID = 00

Version 2: REX2

- has a CPLD, a flash chip, and an SRAM chip for a total of 3 ICs
- hardware ID = 01

Each REX has a CPLD firmware load

- Firmware ID is a 6 bit value
- Readable from the REX version byte
- Firmware is not upgradable by the user, but it can be upgraded by returning it.

Can be used in either M100/T102 or T200

- Each REX can be shipped with a software load for either M100/T102 or T200
- The software load contains all the utilities and functions needed to operate REX.
- User can change or upgrade this load by using a special utility

ROM Replacement

- If level 2 installation is done, a valid ROM image MUST be present in block 2/3 to boot the machine
  - If this block is somehow corrupted, then REX cannot boot the machine.

- There is no emergency safeguard to having a bad image burned
  - Once the process to update block 2/3 is started, there is no way to detect and take action for a bad burn

- In the event of a corrupt OS preventing boot up, the original main rom needs to be installed again
  - This amounts to reverting to a level 1 installation.

- Advanced users could install a switch and pull up inside the ROM compartment such that the /CS signal to activate the main rom is either:
    - directed to REX, and the main rom CS is pulled high, or
    - Directed to the main rom and open circuited to REX (REX has a pullup)
    - This would allow a mechanism to safeguard against a non-booting machine.

REX Installation Levels

LEVEL 1: Wire Free basic installation

- Standard installation, all base features
- Simple plug and play install

LEVEL 2: ROM Replacement

- Additional feature included in firmware
- Allows the main rom to be provided by REX
- Requires 1 (M100/T102) or 2 (T200) wires to be installed
- The system rom must be removed or disabled
  - On M100/T200, the rom chip can simply be unplugged
  - On T102, the rom chip must be removed or disabled.

LEVEL 3: 128kb SRAM upgrade

- Advanced feature supported by REX2 only
- Allows SRAM to be accessed via the option rom bank
- Three wires must be connected from REX2 to system board (clips provided)

Future Development

Flash File System

- Sectors SA0-SA6 are for roms
- Sectors SA7-SA18 are for files
- At least 1 sector must remain free for use as scratchpad
  - This leaves 768k – 64k = 704kb as file storage
  - Scratchpad allows sectors to be purged of deleted files, and may be used by rom burning routines to copy blocks
- First 1k of every sector used in the file system will be a distributed file directory and block directory
  - Need 32 block directory entries, supporting 8 types of blocks
    - Type 0: rom binary image block
    - Type 1: ram binary image block (M100)
    - Type 2: ram binary image block #1 (T200)
    - Type 3: ram binary image block #2 (T200)
    - Type 4: ram binary image block #3 (T200)
    - Type 5: file storage block type 1 (contains directory)
    - Type 6: file storage block type 2 (contains only files)
    - Type 7: unused/scratchpad

- Byte 00
  - Indicates sector status
    - FF = unused, scratch pad possible
    - 00 = in use for file system
- Files will be stored on a 1kb record size similar to RAMPAC
- Files must be contained within a single sector
    - This limits file sizes to 32 kb max
    - Up to 63 files may be stored in one sector – 693 files max stored in REX
    - Directories occupy a directory entry, so directories reduce the # of files per sector

Flash File System

- Directory entries
  - Can be either file names or directory names
  - Up to 255 file directories can be named, nesting is supported
  - each directory entry is 14 bytes long
    - Byte 0 = type; 0=file, 01-FF = directory number for this record (if not a file)
    - Byte 1 = directory membership = number from 0 to FF
    - Byte 2 = file type descriptor
    - Byte 3,4 = file length
    - Bytes 5,6 = undefined
    - Bytes 7-12 = filename or directory name
    - Bytes 13-14 = extension