THE RM65 CRTC MODULE AND THE AIM-65 COMPUTER.

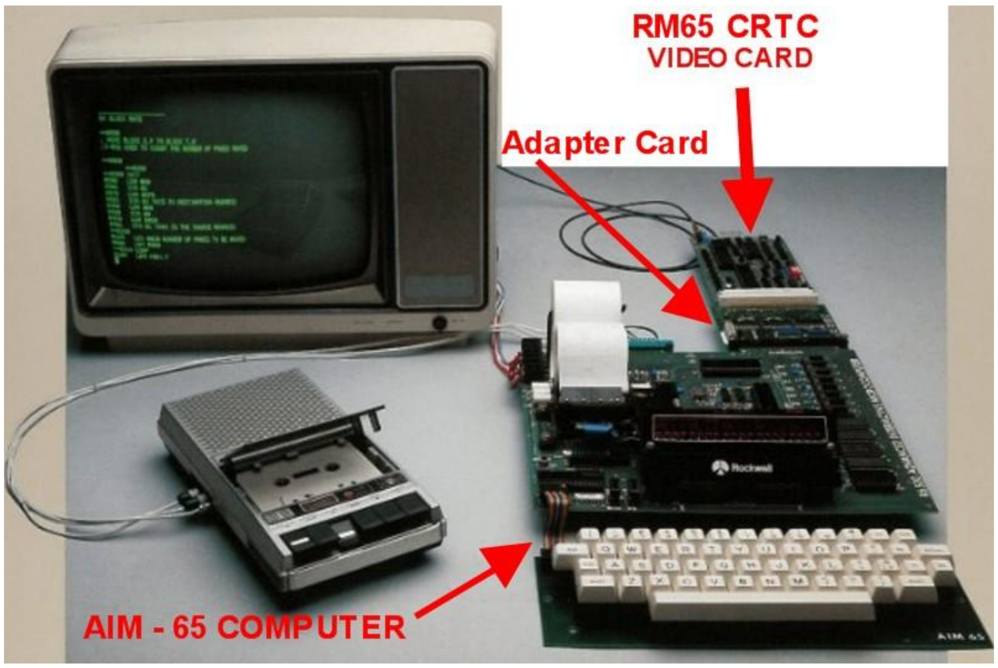
(Dr. H. Holden June, 2021)

BACKGROUND:

The Rockwell AIM-65 computer is a familiar type for those interested in vintage computers. It was similar in many ways to other 6502 computers such as the KIM-1. However, the AIM-65 was better featured with a single line Alpha-Numeric LED display, based on five of the DL1416 LED modules. It also and has an assembler, disassembler and BASIC all in ROM when fitted. Also the AIM-65 has an elaborate ROM monitor to make using the computer very easy, including transferring files to tape. In addition it has a well equipped keyboard. If one thing was missing from the standard AIM-65, that could have been there, it was a video output for a multi-row video display.

(The AIM-65 computer motherboard also sports a very interesting and compact Thermal Printer module, made by Shimadzu in Kyoto, Japan. Their company goes back to the 1870's and they became experts in high quality medical electronics. This extraordinary, compact and brilliant printer module was also used in equipment made by Tektronix. It is the subject of another article I'm working on, under construction. This printer is worthy of an article all on its own. It is quite an exceptional little machine).

While browsing the internet and learning about the AIM-65 computer, I came across the following image that stirred my imagination:



After some research and help (from members of the Vintage Computer forum) the video card shown in the photo above was recognised as the Rockwell RM65 CRTC card, and the other card an Adapter card, which allows it to interface directly to the AIM-65's 44 way expansion connector. It turned out much of the firmware on this CRTC card was designed to make it work effectively with the AIM-65.

ENTHUSIASM FOR ROCKWELL COMPUTERS:

Before I bought the Rockwell AIM-65 computer, I didn't know much about Rockwell. But after studying their AIM-65 computer and the Rockwell documentation, I have become very impressed by what their company created with the AIM-65 computer and their RM-65 computer systems too. Also the "modular" approach to computing systems was very advanced.

The AIM-65 computer was dedicated to the notion of Education and being a "trainer" computer. This is an admirable feature and Rockwell hoped that the AIM-65 users would benefit from their 6502 computer as a teaching tool. I suspect it inspired a generation of Engineers. I have only owned the AIM-65 computer for a short period of time and I am inspired by it. Rockwell also made most of the LSI IC's they used in their own computers and they also made parts for the Space program:

A heritage of unsurpassed quality and technical excellence

Electronic Devices Division
3310 Miraloma Avenue
P.O. Box 3669
Anaheim, CA 92803
Toll Free (800) 854-8099
in California (800) 422-4230

©Rockwell International Corporation 1981
All Rights Reserved
Printed in the U.S.A.

Rockwell International

Document No. 1
Rev. 2, June 1981

Information on the RM65 CRTC card was found in a 1984 Rockwell manual, it is a very advanced Video card and while there have been other video cards designed which will work with the AIM-65, this one is by far the most sophisticated and compact:

RM 65 MICROCOMPUTER MODULES

The RM65-5102E CRT Controller Module is one of the hardware options available for the RM 65 Microcomputer Module family.

RM 65 Microcomputer Module products are designed for OEM and end user microcomputer applications requiring state-of-the-art performance, compact size, modular design and low cost. Software for RM 65 systems can be developed in R6500 Assembly Language, PL/65, BASIC and FORTH. Both BASIC and FORTH are available in ROM and can be incorporated into the user's system.

RM 65 modules use a motherboard interconnect concept and accept any card in any slot. The 64-line RM 65 Bus offers memory addressing up to 128K bytes, high immunity to electrical noise and includes growth provisions for user functions. A selection of card cages provides packaging flexibility. RM 65 products may also be used with Rockwell AIM 65 and AIM 65/40 Microcomputers for product development and for a broad variety of portable or desktop microcomputer applications.

PRODUCT OVERVIEW

The CRT Controller (CRTC) Module interfaces the RM 65 to a CRT monitor or television receiver. The CRTC module outputs HSYNC, VSYNC, and raw video signals for direct connection to a CRT Monitor, and composite video for connection to a CRT monitor or to a TV receiver through an RF modulator. A socketed on-board ROM generates 5 × 7 characters with two descenders in a 7 × 10 dot matrix field to provide upper and lower case alphanumerics and special symbols. The 2K bytes of on-board display RAM are memory-mapped.

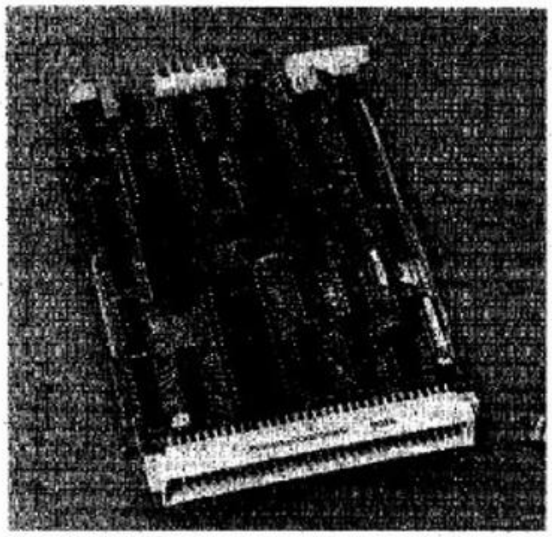
A 2K-byte program ROM provides firmware to configure the display format for 80 columns by 25 rows or 40 columns by 16 rows, scan rate of 50 or 60 Hz, and a CRT display driver for AIM 65. There are also cursor control, screen editing, and utility routines.

ORDERING INFORMATION

Part No.	Description
RM65-5102E	CRT Controller (CRTC) Module
Order No.	**Description**
814	CRT Controller (CRTC) Module User's Manual (included with RM65-5102E)

FEATURES

- Compact size—about 4" × 6¼" (100 mm × 160 mm)
- RM 65 bus compatible
- 4K Byte character generator ROM with:
  - —Upper and lower case alphabetics
  - —Special characters
  - —Numbers including subscripts and superscripts
  - —Math symbols
  - —Semi-graphics
- On-board ROM firmware supports:
  - —Scrolling
  - —Screen editing
  - —Full cursor movement control
  - —Full screen standard or inverse video
  - —Predefined formats for
    - 80 column by 25 row (50/60 Hz)
    - 72 column by 22 row (50/60 Hz)
    - 40 column by 25 row (60 Hz)
    - 40 column by 16 row (60 Hz)
  - —Selectable format from 1 to 80 columns by 1 to 25 rows
  - —NTSC (60 Hz, 525 lines per frame) and European (50 Hz, 625 lines per frame) raster format
  - —CRT display driver for AIM 65
- Single 5 volt operation
- Fully assembled, tested and warranted


RM65-5102E CRT Controller (CRTC) Module

FUNCTIONAL DESCRIPTION

The Data Transceivers invert and transfer 8 bits of parallel data between the CRTC Module and the RM 65 bus, based on control signals from the Base Address Decoder and the Control Buffers. The read/write control line determines the direction, while the bus active enables the Data Transceivers.

The Address Buffers invert and transfer the 16-bit parallel address lines from the RM 65 bus to the Base Address Decoders, the R2316 ROM, the CRT Controller (CRTC) device, and to the Refresh RAM device.

The Control Buffers invert and transfer the phase 2 clock and read/write control signals from the RM 65 bus onto the module.

The Bank Select Control circuit detects when the module's assigned memory bank is addressed, by comparing the bank address signal from the RM 65 bus to the Bank Select and Bank Select Enable switches. The Bank Select Enable switch allows the board to reside in common memory (both Bank 0 and Bank 1) or only in the Bank set by the Bank Select switch (either Bank 0 or Bank 1).

The Base Address Decoder, with the Base Address Select switches, the Bank Select Control circuit, the ROM Disable Switch and the read/write and phase 2 clock signals, generates device selects for the on-board ROM, RAM, and I/O (CRTC device and Display Enable Status Buffer). The Base Address Select switches allow the module to be selected to any 4K block. Within the selected 4K block, the RAM is assigned to the lower 8 pages (2K bytes), and the I/O to the first 256 byte page of the upper 2K bytes. When the ROM is disabled, only the RAM and I/O can be selected and the module is assigned 9 pages (2304 bytes) in memory. When the ROM is enabled, the module is assigned the full 4K bytes, with 7 pages for ROM, in addition to the RAM and I/O.

The Controller Clock uses a crystal-controlled oscillator to derive a 6 MHz or 12 MHz reference for the shift register dot clock depending on the Dot Clock Select jumper position. With the 6 MHz clock, up to 40 characters per line can be displayed on any monitor or standard television using an RF modulator. Up to 80 characters per line can be achieved with the 12 MHz clock and a high bandwidth monitor. The dot clock is divided by seven to provide a Character Clock for the CRTC device and a load character signal for the shift register.

The Refresh RAM provides 2K bytes of display memory, for screen densities of up to 25 lines with as many as 80 characters each. The RAM is directly mapped into the RM 65 memory map, so the display can be updated by a block memory move or under DMA control. The Refresh RAM Multiplexer and RAM Transceiver allow the RM 65 bus and the CRTC device to both access the Refresh RAM, with the RM 65 bus having priority when any conflict occurs.

The Display Enable Status Buffer allows the RM 65 bus to monitor the active display times, so that display memory transfers can be made with no visible distortion.

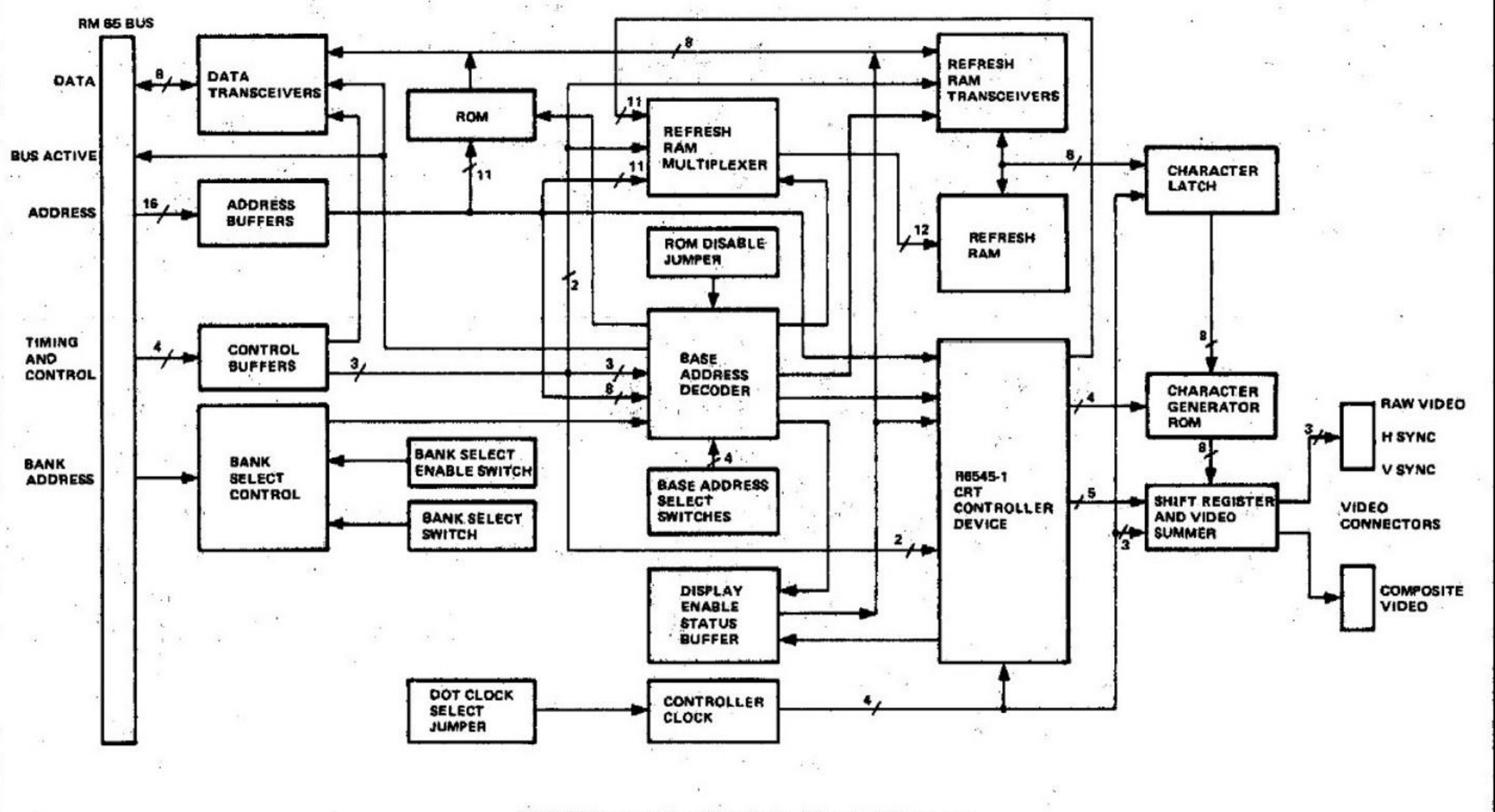
The Character Generator ROM holds the fonts for the character set. These fonts are stored as 256 characters, each with 10 seven-bit rows. The four CRTC device row address lines and the eight Character Latch bits, which hold the character being refreshed, create an address for the character generator ROM. The output data of the ROM, which is seven parallel bits, represents the display pattern. The Shift Register takes this data and forms the serial video data. The Video Summer combines and buffers the serial video data with CRTC device timing signals to form a composite video output and a separate video, horizontal sync, and vertical sync.

The Program ROM contains the firmware for an intelligent CRT driver, in addition to utilities to aid in custom CRT display application software. There are six predefined screen formats, including 25 lines of 80 characters (50 or 60 Hz), 22 lines of 72 characters (50 or 60 Hz), 25 lines of 40 characters (50 Hz), and 16 lines of 40 characters (60 Hz). For other formats, any dimensions from 1 to 25 lines of from 1 to 80 characters can be defined (50 or 60 Hz). Full screen inverse video and 256 display characters allow flexible display capabilities.

The intelligent display driver controls all screen updating and cursor movement for the selected screen format. The cursor can be on, off, or blinking with movements including up, down, left, right, home, and carriage return, as well as to any row and column position. There are many commands to facilitate screen editing, such as:

Insert character or line
Delete character or line
Clear to end of line
Clear to end of screen
Clear line or screen
Set or Clear special character mode

The firmware utilities are useful for special applications. There is also a display driver which replaces the AIM 65 on-board display with a CRT monitor and an AIM 65 Assembler listing reformatter which takes advantage of the longer display lines.



CRT Controller Module Block Diagram

RM65-5102E CRT Controller (CRTC) Module

RM 65 CRTC Control Commands

Hex Code	Character	Description	Hex Code	Character	Description
00	CTRL @	*	10	CTRL P	Pass Through Next Character
01	CTRL A	Clear Line	11	CTRL Q	*
02	CTRL B	Clear to End of Line	12	CTRL R	*
03	CTRL C	Clear Screen	13	CTRL S	Toggle Insert Character Mode
04	CTRL D	Clear to End of Screen	14	CTRL T	Delete One Character
05	CTRL E	Clear Screen	15	CTRL U	Insert One Line
06	CTRL F	Clear to End of Screen	16	CTRL V	Delete One Line
07	CTRL G	*	17	CTRL W	Display Cursor
08	CTRL H	Backspace (←)	18	CTRL X	Blank Cursor
09	CTRL I	Horizontal Tab (→)	19	CTRL Y	Relink AIM 65 Display
0A	CTRL J	Line Feed (↓)	1A	CTRL Z	*
0B	CTRL K	Vertical Tab (↑)	1B	CTRL [	Escape Character (ESC) (1)
0C	CTRL L	Form Feed (Clear Screen)	1C	CTRL \	Blinking Cursor
0D	CTRL M	Carriage Return (Home on Line)	1D	CTRL]	Enter Normal Characters
0E	CTRL N	Home on Screen	1E	CTRL ^	Perform Self Test
0F	CTRL O	Home on Screen	1F	CTRL _	Reverse Video

*These characters have no effect.

(1) There are two escape sequences as follows:

Hex Code	Character Sequence	Function
1B 3D YY XX	ESC = y x	Move the cursor to the row y and column x position, with row y between top ($00) and bottom ($19), and column x between leftmost ($00) and rightmost ($4F).
1B 47	ESC G	Enter Graphics Character Mode

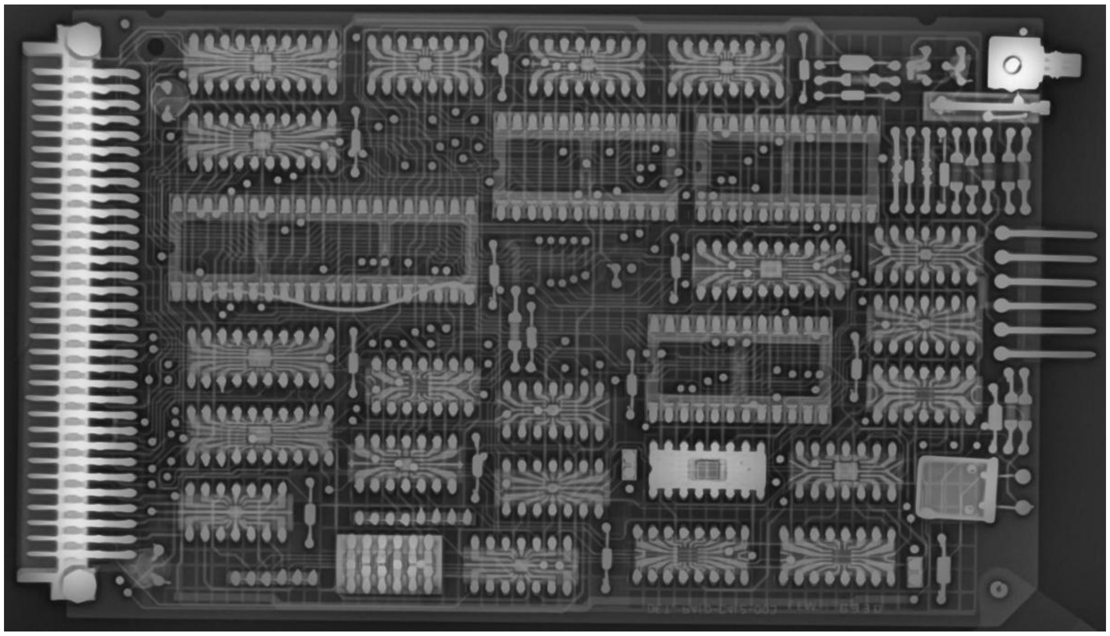
As can be seen from the data sheets above, it is a highly elaborate controller module. It has a 6545 CRTC IC, a 2316 ROM with the firmware in it, a 2332 ROM with the Character stored in it and a 2k, 2016 RAM and all the support IC's to make it operate.

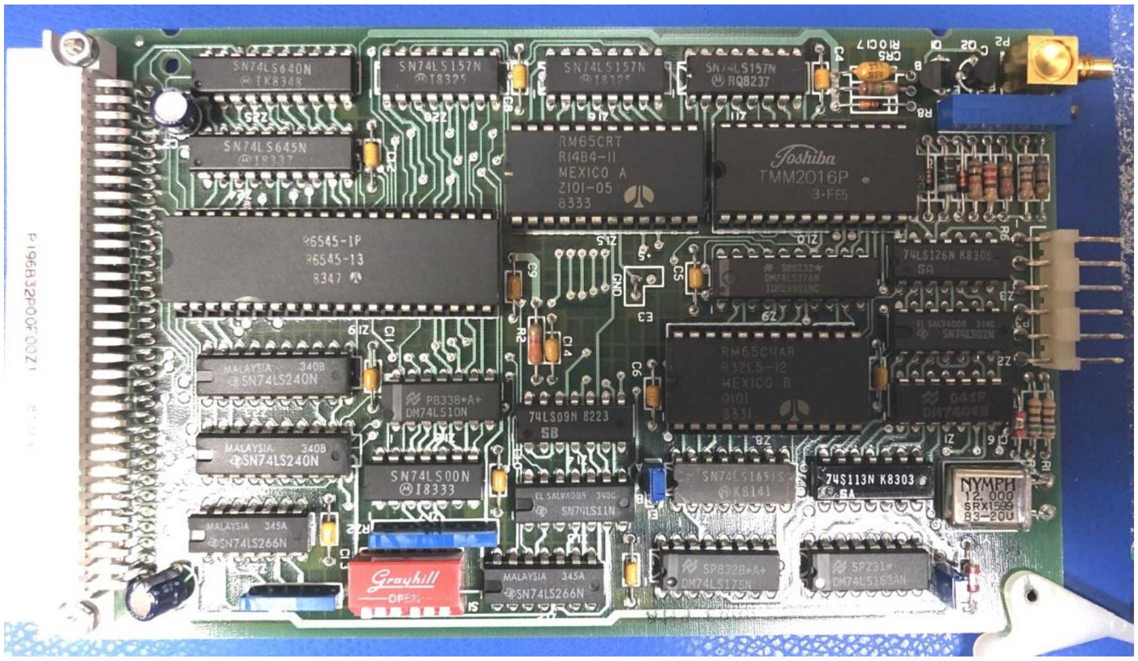
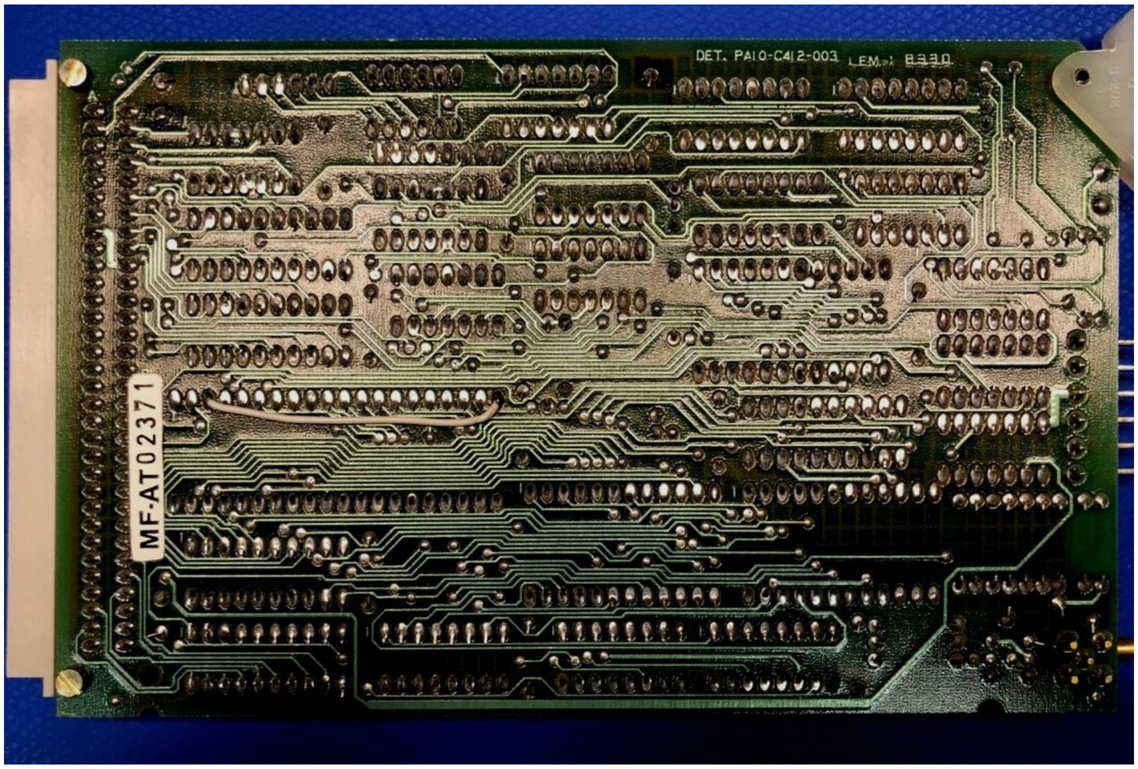
REPLICATING THE RM65 CRTC BOARD AND THE ADAPTER BOARD:

At this point in history, the RM65 CRTC board and the adapter board have become very rare. Fortunately a Vintage Computer forum member (dfnr2) had one of these cards and was able to provide detailed photographic images and even an X-ray of the card. In addition he was above to dump the contents of the two ROMS.

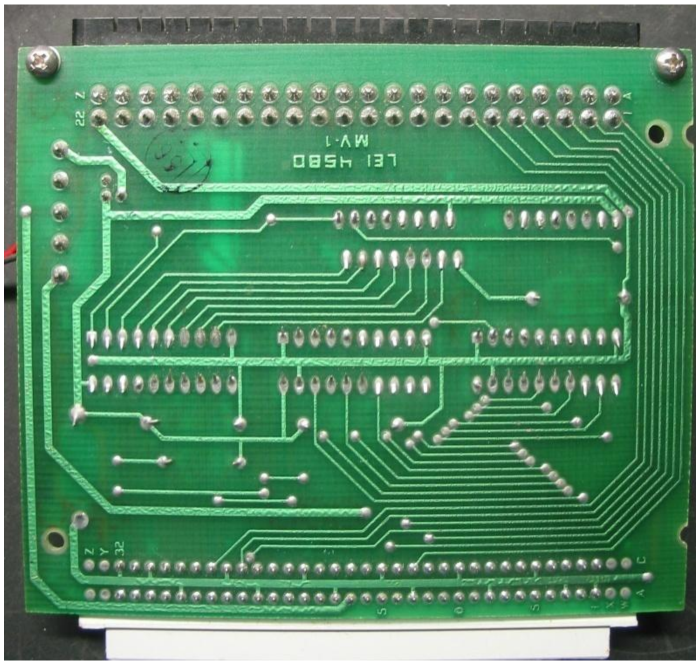
With this information I was able to replicate the RM-65 video card. Not without difficulty though. My previous vintage computer related pcb replicas were simple double sided pcb's. The RM-65 CRTC board turned out to be a 4 layer type and in addition one of the internal layers also contained an 11 bit address bus !

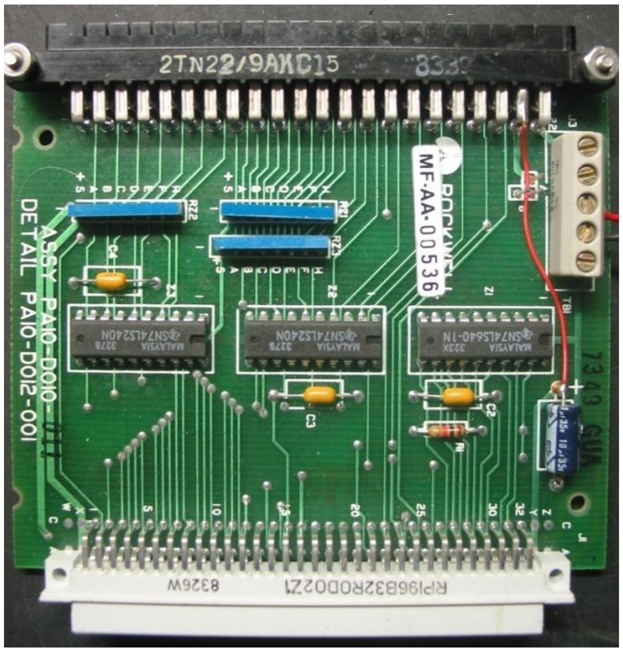
The following images are the X-Rays and the photos that were used to help replicate the RM-65 CRTC pcb and the adapter board:



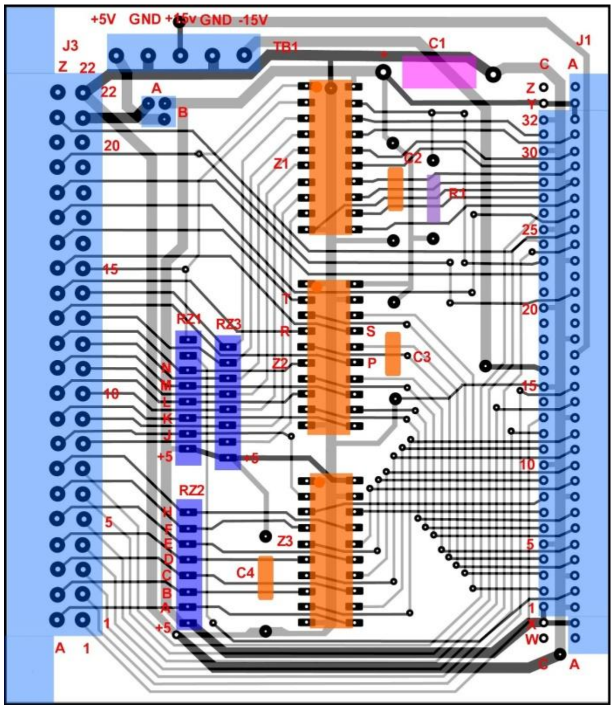



THE ADAPTER BOARD:





Being a simple two layer board, the Adapter board was far less of a challenge, and I could draw up the pcb foil patterns in short order:

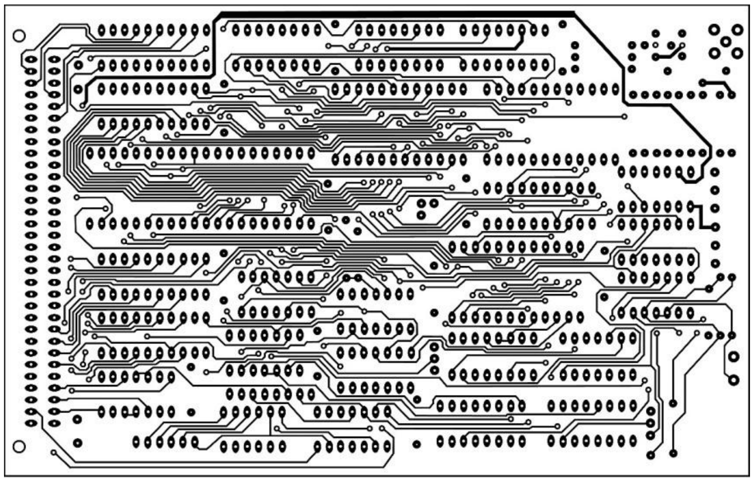
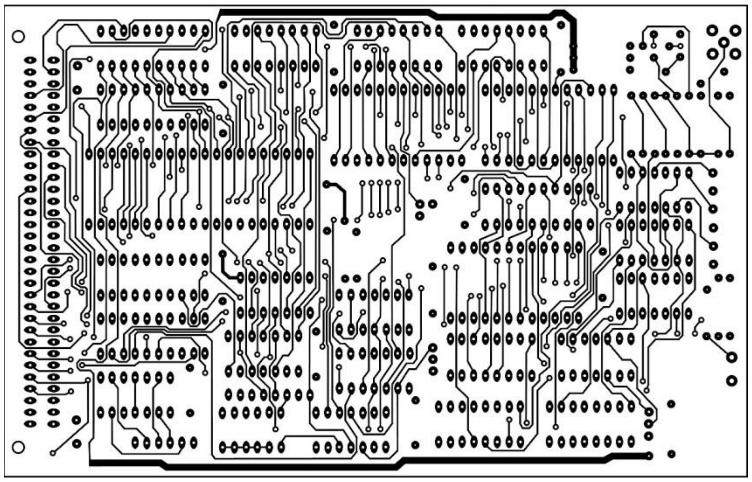
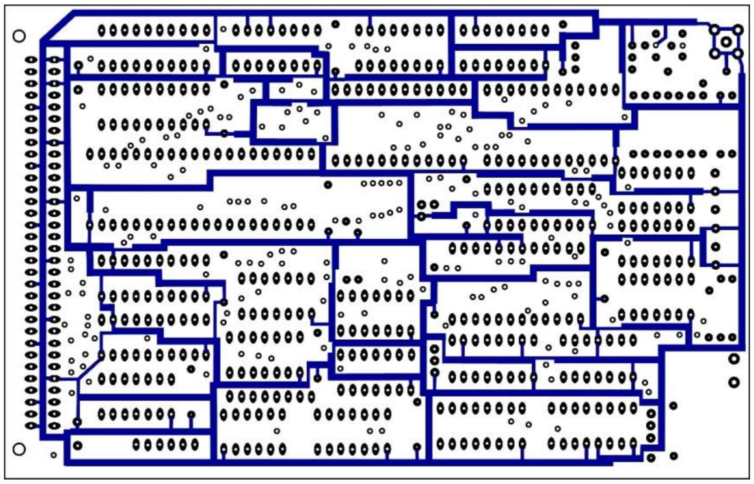
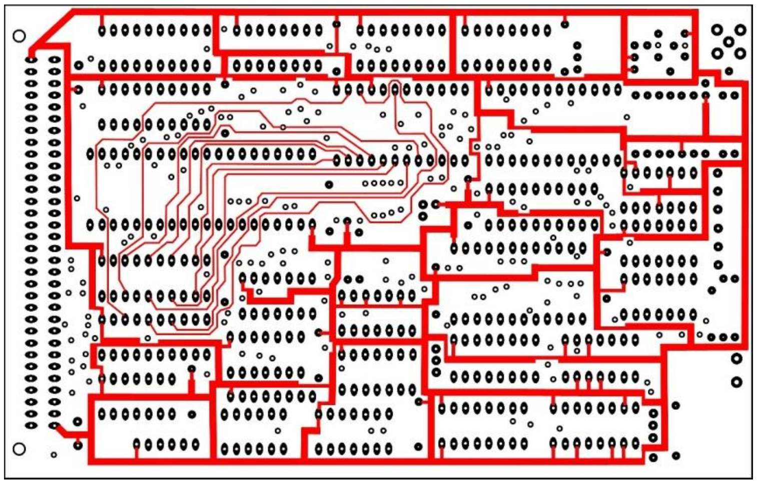


However, the four layer board of the RM-65 CRTC was a major challenge.

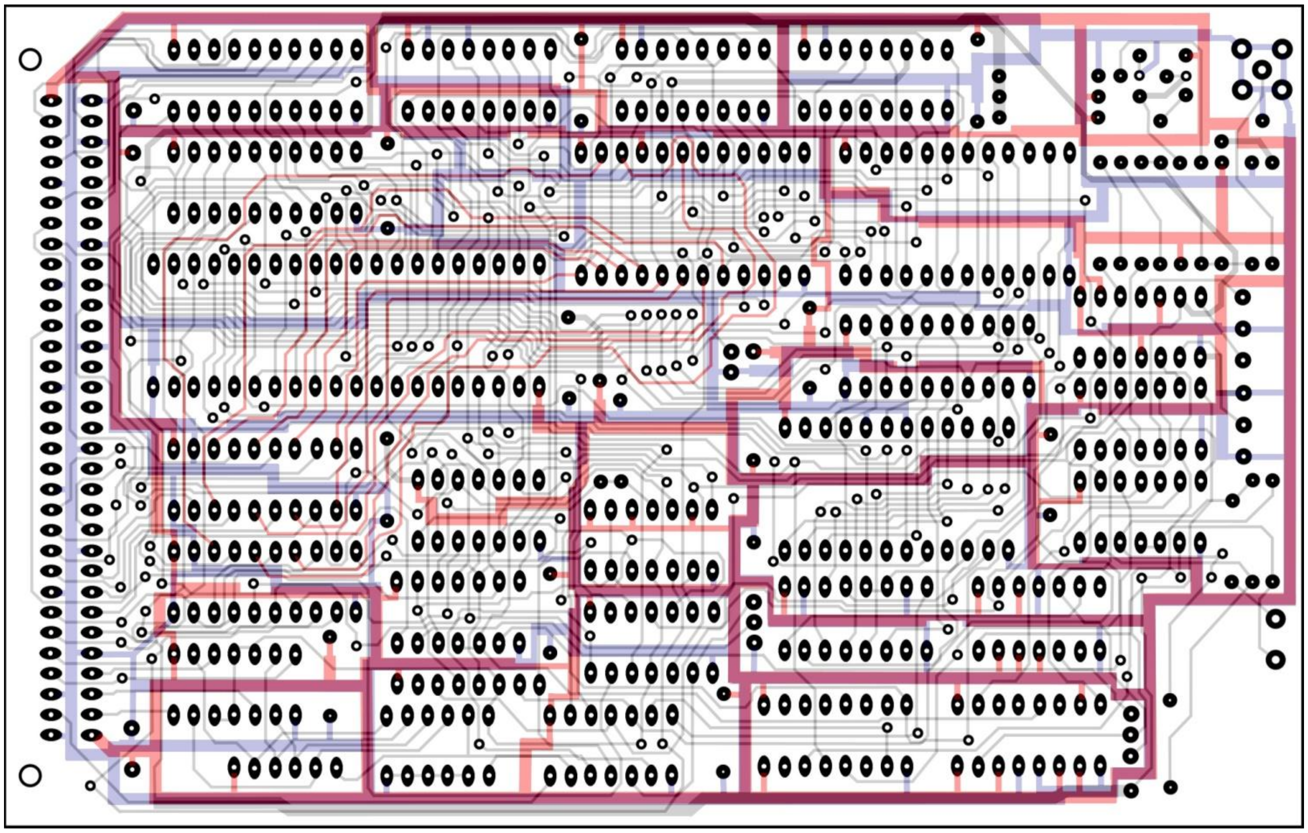
This was the first time I had attempted to replicate a four layer pcb. And although the top and bottom layers of the pcb had a track layout, which was fairly easy to determine from photos and schematic, this was not the case for the two middle layers which included the pcb's ground plane and the +5v power supply plane.

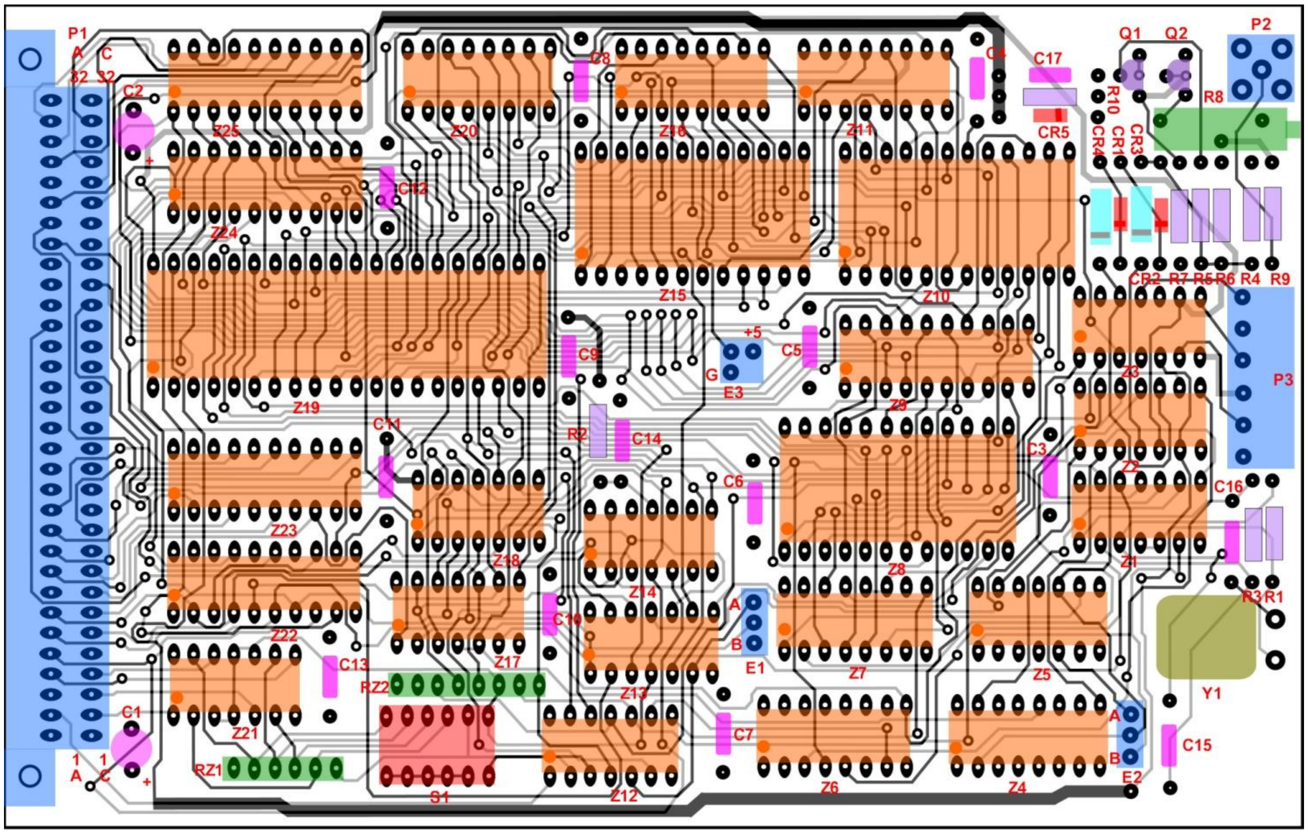
One complication was also the fact that the +5V plane appeared to contain an 11 track address bus. However, with the help of the X-Rays and photos provided by Dave (dfnr2), I was able to route the 11 wire address bus inside the +5V plane close to the original layout. Also, I simply designed my own ground plane and +5V plane to suit the task.

The following diagrams show the images that I drew and sent to my pcb maker here in Australia:

The diagram below shows all the planes together and the pcb component overlay:





The images above have no relationship to any kind of pcb drawing software.

I drew them on the vintage Microsoft drawing program called “Picture It”. This was done by using the original photographs and in this case, the X-ray, and drawing over the images and a superimposed 2.54 x 2.54mm grid.

The ground plane was drawn in blue and the +5v plane drawn in red, including the 11 bit address lines that are in that +5V plane.

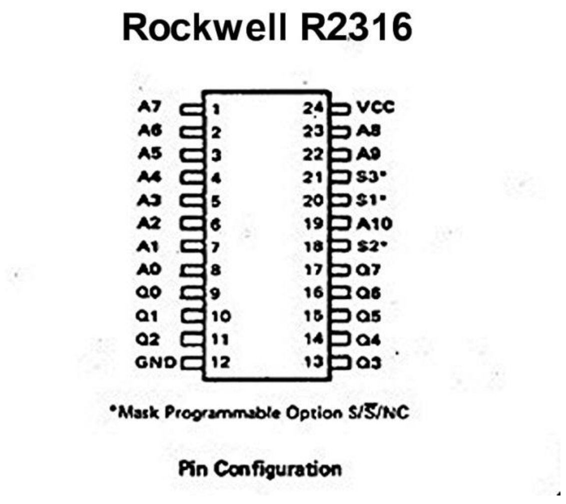
The reason that this somewhat oddball process of vintage pcb replication works, is that George, from LD electronics, here in Australia, is able to use such jpg files to create a replica of them in Altium pcb design software. This is remarkable really, because looking at attempts to replicate other vintage and historically important pcb’s, from computer history, the Apple 1 computer motherboard for example, some replicas had tracks which did not replicate the originals in the shape and curvature of the tracks, so they didn’t look right.

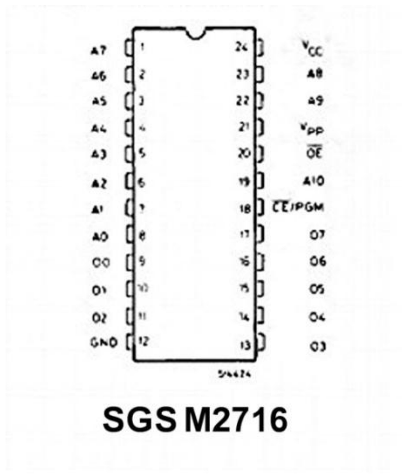
However, George is able to make the real pcb tracks exactly match the artwork of the tracks drawn on the .jpg images. Therefore, this process, although somewhat unorthodox, creates replica pcb’s very close to the originals.

This process is a great asset in preserving vintage computer systems, which otherwise might be so rare, they get lost to the sands of time. It appears that most vintage computer enthusiasts share the same goal of preserving computer history. In a sense, just like any other kind of Historians wanting to save, restore and perpetuate innovative past knowledge or technology.

SUBSTITUTE ROMS:

Analysis of the circuitry indicated that a 2716 UV Eprom could be plugged directly into the socket for the 2316 OTP ROM without modification to the pcb track work.



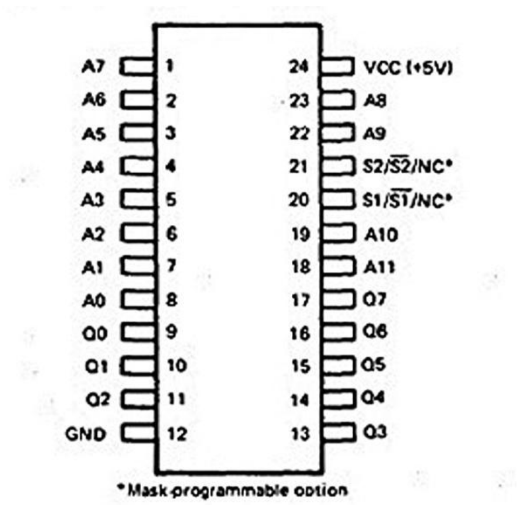


The application of the R2316 ROM in the CRTC controller is that it holds the firmware to make the controller run. The R2316 has three mask programmable chip select pins 18, 21 & 23, labelled as S2*, S1* and S3*.

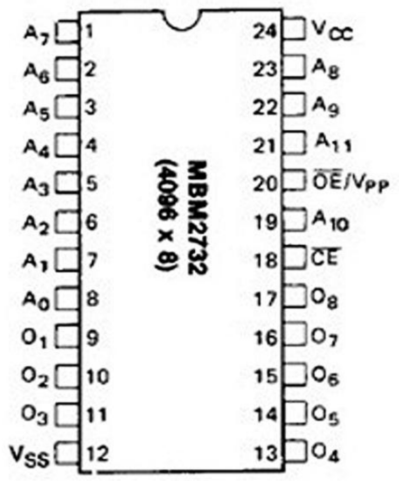
In the case of the RM65 CRTC controller circuit, Pin 18 is permanently grounded by a pcb link and pin 21 is tied high. Pin 20 is used for chip selection and the signal line is labelled as **/**CS1. This implies the mask programmable chip select inputs were programmed by Rockwell to: S3* = don't care (tied high), and S2* and S1* must both be low to select and enable the chip.

Looking at the M2716's connections this is convenient because in read mode, pin 21 is normally tied high. Pin 18 is the **/**CE and this gets tied low by the existing track work and pcb link. Pin 20 on the M2716, is the Output enable **/**OE, and this is connected to the **/**CE control line on the RM65 board. So the conclusion is; no tracks on the CRTC pcb should have to be changed to use the M2716 directly in place of the R2316.

However, one complication was that the original board used the R2332 character generator ROM. These are a one time programmable device. I decided to replace this device with an MBM2732 UV Eprom, partly because I have the ability to program this part. However, this required modification to the circuit track layout on the CRTC board.



R2332 Pin Configuration



As can be seen there are a few complications. On the R2332 Pin 21 and 20 are mask programmable chip selects. Inspecting the CRTC's schematic, indicated that the R2332 character generator ROM is permanently "chip selected & output enabled".

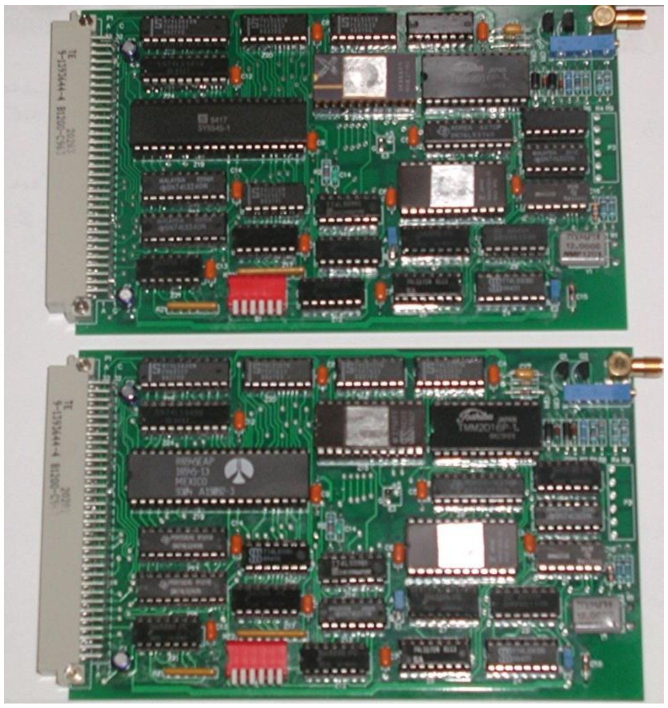
(**As an aside:** The idea behind mask programmable chip selects was that each chip in an array of chips could have a different "code" to activate them and the select inputs could all be "wired OR" together. So, say with 3 chip select inputs, you could select or activate one of the eight ROMs at a time with a 3 bit code. Or with two select inputs you could select one of 4 ROMs. This idea reduced the requirement for external gating).

Pin 21 of the R2332 was tied high (not used or programmed for don't care) and pin 20 used as the chip select and wired permanently low on the pcb.

In addition on the R2332, A11 is pin 18, but it is pin 21 on the MBM2732.

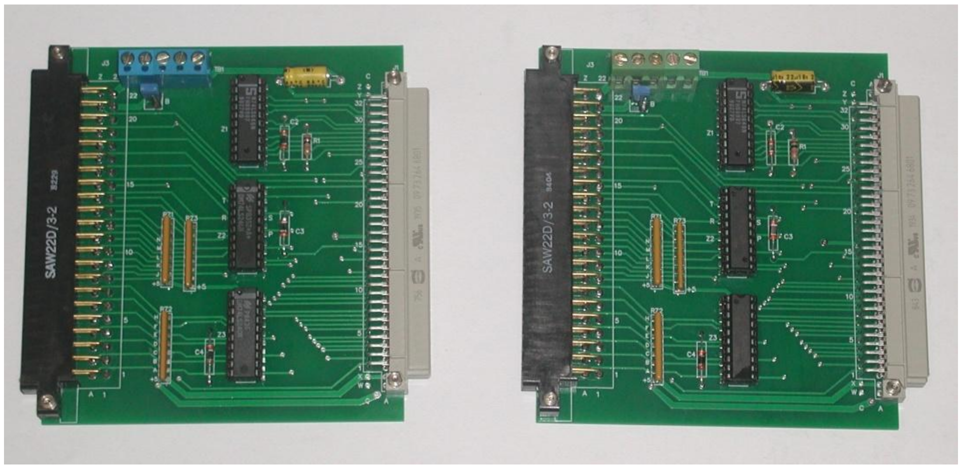
To operate the MBM272 in permanent read mode it requires that both pin 18 ***/***CE, and pin 20 ***/***OE are tied low. Therefore the pcb track wiring was changed so that pin 21 was disconnected from its original connection to +5V and that the signal line input to A11 was disconnected from pin 18 and re-routed to pin 21 and after that pin 18 was tied low. Pin 20 was already tied low in the original design, so it did not require changing.

THE FINISHED PCB's:



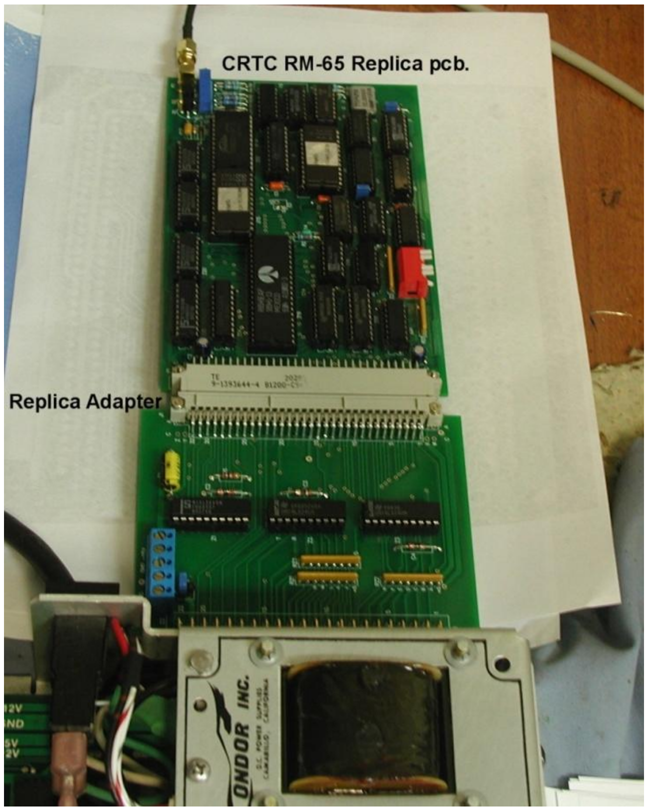
(My pcb artwork was 100% accurate, I went over it multiple times to eliminate errors. However there were manufacturing problems on the pcb's made by LD Electronics, with two tracks that did not quite reach their pads. This problem was detected and corrected with two link wires)**.**

ADAPTER PCB's:

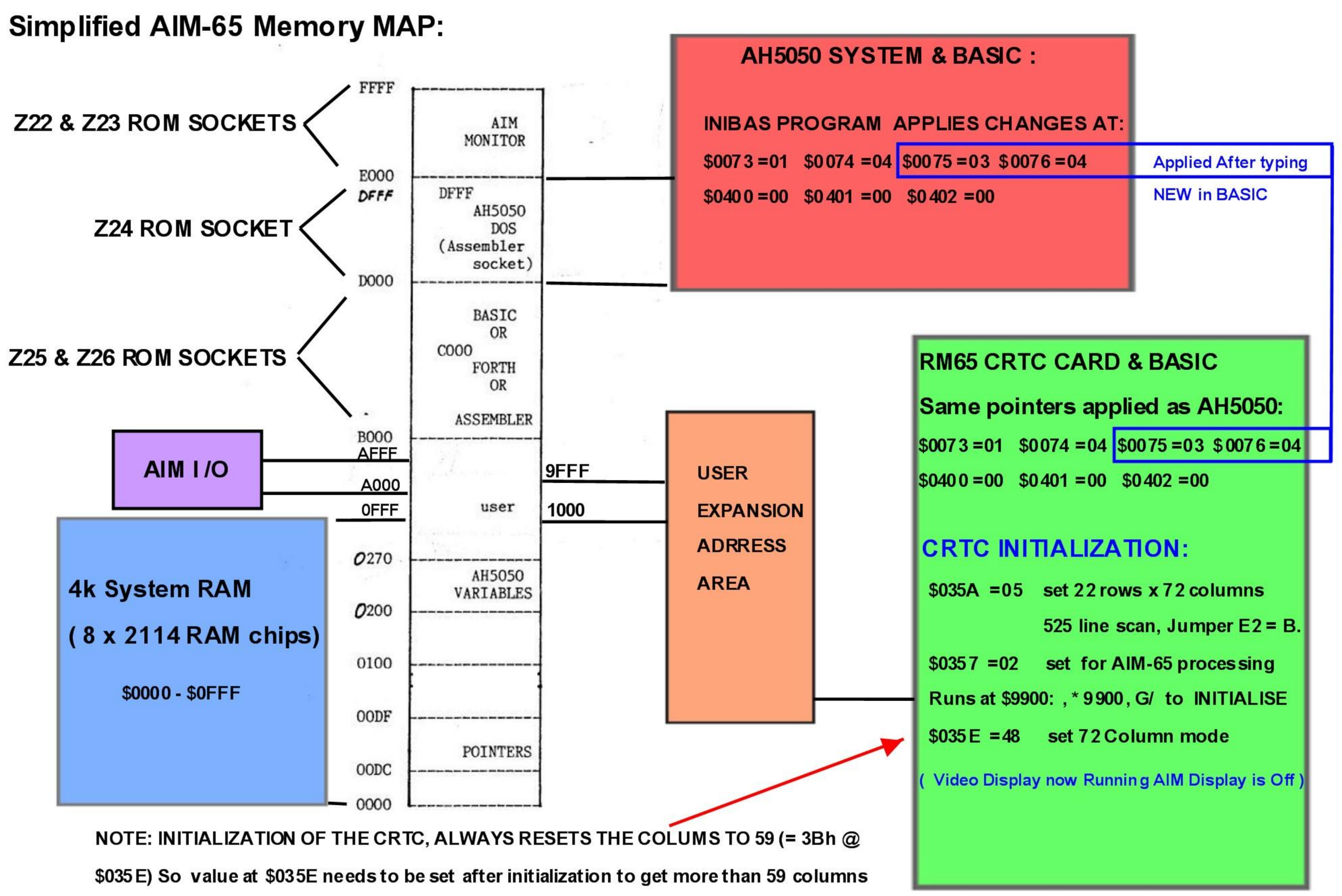


I built two of each pcb. This was done for the purpose of comparing one with the other. This could have helped if I happened to land something like a bad IC, in conjunction with other faults. One adapter was sent to dfnr2 (Dave) for analysis and trial with his original Rockwell RM-65 CRTC card on the AIM-65 computer.

The photo below shows a CRTC card plugged onto the replica adapter card and the AIM-65 Expansion connector.



How the RM-65 CRTC ROM firmware fits in:


Simplified AIM-65 Memory MAP:
Z22 & Z23 ROM SOCKETS
Z24 ROM SOCKET
Z25 & Z26 ROM SOCKETS
AIM I/O
4k System RAM
(8 x 2114 RAM chips)
$0000 - $0FFF
FFFF
AIM MONITOR
E000
DFFF
DFFF AH5050 DOS (Assembler socket)
D000
BASIC OR C000 FORTH OR ASSEMBLER
B000
AFFF
A000
0FFF
user
9FFF
1000
0270
AH5050 VARIABLES
0200
0100
00DF
POINTERS
00DC
0000
AH5050 SYSTEM & BASIC :
INIBAS PROGRAM APPLIES CHANGES AT:
$0073 =01 $0074 =04 $0075 =03 $0076 =04
$0400 =00 $0401 =00 $0402 =00
Applied After typing
NEW in BASIC
USER
EXPANSION
ADRRESS
AREA
RM65 CRTC CARD & BASIC
Same pointers applied as AH5050:
$0073 =01 $0074 =04 $0075 =03 $0076 =04
$0400 =00 $0401 =00 $0402 =00
CRTC INITIALIZATION:
$035A =05 set 22 rows x 72 columns
525 line scan, Jumper E2 = B.
$0357 =02 set for AIM-65 processing
Runs at $9900: , * 9900, G/ to INITIALISE
$035E =48 set 72 Column mode
(Video Display now Running AIM Display is Off)
NOTE: INITIALIZATION OF THE CRTC, ALWAYS RESETS THE COLUMS TO 59 (= 3Bh @
$035E) So value at $035E needs to be set after initialization to get more than 59 columns

THE CRTC CARD WITH BASIC AND INIBAS AND THE AH5050 FILER SYSTEM:

The firmware in the CRTC card, for use with the AIM-65, runs at address $9900. Also, prior to doing this, variables at $035A and $0357 must be set so as to initialize the CRTC controller IC in the card (more details below).

In addition, some interesting issues crop up when using BASIC. These issues are practically identical to the issues involved attempting to uses ABM's AH5050 Filer system (and the commodore 1541 disk drive) with the AIM-65. To avoid problems, pointers need to be altered and the BASIC starting address requires alteration.

A program had been created by members of the Vintage Computer Forum, called INIBAS to replace the unavailable original INIBAS program that was supplied by ABM along with the AH5050 PCB. This program allows the AH5050 system to work with BASIC, so that BASIC's SAVE & LOAD functions work properly and it fixes the pointer issues too.

The INIBAS program (shown below) with the pointer and BASIC start address corrections is suited to both the AH5050 Filer and the RM-65 CRTC card requirements.

Therefore, running INIBAS after BASIC has been cold started allows the AH5050 system to work properly, but it also sets the required pointer changes to allow the CRTC card to work with BASIC too. This is very handy.

INTERACTIONS BETWEEN THE CRTC CARD AND THE AH5050 FILER SYSTEM:

It appears that the AH5050 Filer system (which is used in association with the Commodore 1541 Disk Drive) has some conflict with the CRTC video card, when they both try to operate simultaneously.

The AH5050 Filer was designed by the company ABM and of course the AIM-65 & RM-65 CRTC card by Rockwell. There would have been no effort to necessarily make the two compatible.

The Filer system (even supported by INIBAS) fails to SAVE and LOAD BASIC files correctly, or (R)run saved disk files with the AH5050 file handler, if the CRTC card is concurrently active.

This problem can be solved by avoiding using the Filer, while the CRTC card is operational.

The CRTC card requires, as noted on the diagram above, that the pointers and the start of BASIC are corrected so as to run with BASIC:

$0073 =01 $0074=04 $0075=03 $0076=04

$0400=00 $0401=00 $0402=00

Also, to make BASIC's SAVE and LOAD function correctly with the AH5050 Filer, it is also necessary to alter the pointers to the same values after cold starting BASIC <5>

So the initial sequence of events to set up the pointers for both the AH5050 Filer system and the CRTC card, if both are being used with the AIM-65 computer:

1) Power AIM65 & Commodore disk drive. Insert Disk with INIBAS program (see below)

2) CTRL and PRINT keys to turn off the printer.

3) Cold start BASIC <5>, Then enter: Memory = 4000, Width = 72

4) Escape to AIM-65 monitor (esc key)

5) Initialise Filer <N> and press F1 key

6) R(run) INIBAS program from disk. DEV=D1 FILE= INIBAS,

This corrects the pointers above (except those at $0075 and $0076)

7) Warm start BASIC <6> and type NEW (This corrects pointers at $0075 and $0076)

8) Escape to monitor again to initialise CRTC card:

9) Manually set $035A =05, $0357=02

10) * 9900, G/ Runs CRTC software & initialises CRTC card.

CRTC video takes over from AIM-65 display

11) Manually set $035E=48 (sets CRTC card to 72 column mode)

12) Warm start BASIC <6>

Write BASIC programs , RUN & LIST etc…. but when it comes to SAVING or LOADING:

13) Type CTRL and Y keys. CRTC card is deactivated (screen freezes), AIM-65 display takes over.

14) Press esc key, back to AIM-65 monitor.

15) Re-initialise the Filer <N>

16) Warm start BASIC <6>

17) SAVE BASIC program to Disk: SAVE, OUT=U,DEV=D1, FILE= MYBAS

18) Type NEW to erase the BASIC program, if required for an operational test:

19) LOAD and test saved BASIC program. LOAD, IN=U,DEV=D1, FILE= MYBAS

To return to CRTC video card control of the display after LOADing any SAVE'd program and the CRTC card was already initialised in steps 1 through 12 above:

20) Escape to monitor, esc key

21) Manually set $010C=4C $010D=C0 $010E=0F

$0FC0=20 $0FC1=64 $0FC2=99 $0FC3=60

22) Press F1 key, this warm starts the CRTC card.

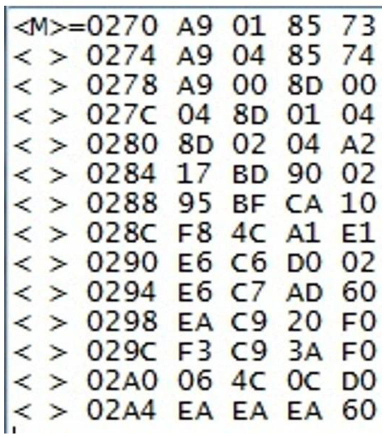
23) Warm start BASIC <6>.

24) Work on LOADed program with CRTC display active.

The above sequence avoids having the CRTC card and the AH5050 running together.

THE INIBAS PROGRAM:

INIBAS:

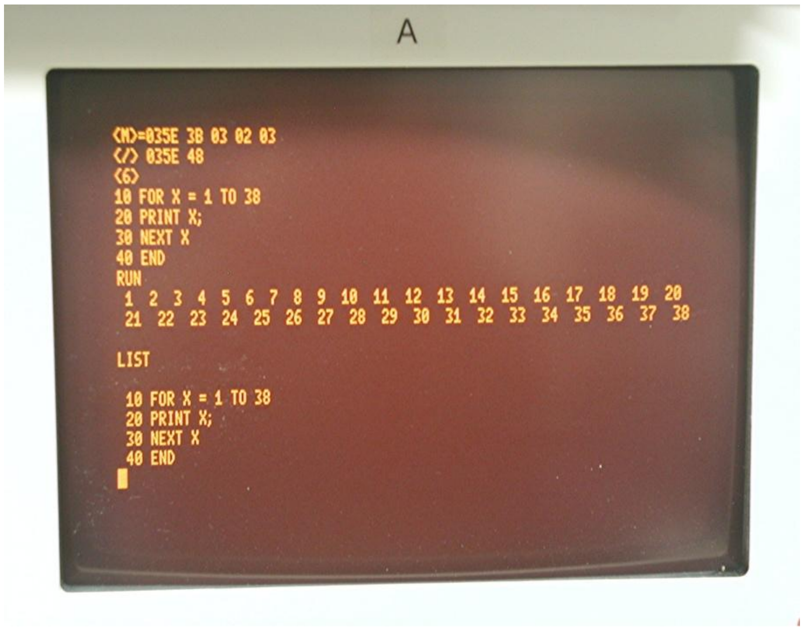


```
<M>=0270 A9 01 85 73
< > 0274 A9 04 85 74
< > 0278 A9 00 8D 00
< > 027C 04 8D 01 04
< > 0280 8D 02 04 A2
< > 0284 17 BD 90 02
< > 0288 95 BF CA 10
< > 028C F8 4C A1 E1
< > 0290 E6 C6 D0 02
< > 0294 E6 C7 AD 60
< > 0298 EA C9 20 F0
< > 029C F3 C9 3A F0
< > 02A0 06 4C 0C D0
< > 02A4 EA EA EA 60
```

This program is easily entered manually on the AIM-65, but when using the AH5050 Filer it is easy to save it to Disk as the named file INIBAS using the Filer's (P) or put command and saving the file from $0270 to $02A8. Then INIBAS is available from disk with the (R) run command anytime you need it, so as to run BASIC with the AH5050 Filer, or to set the pointers to run the CRTC card with BASIC.

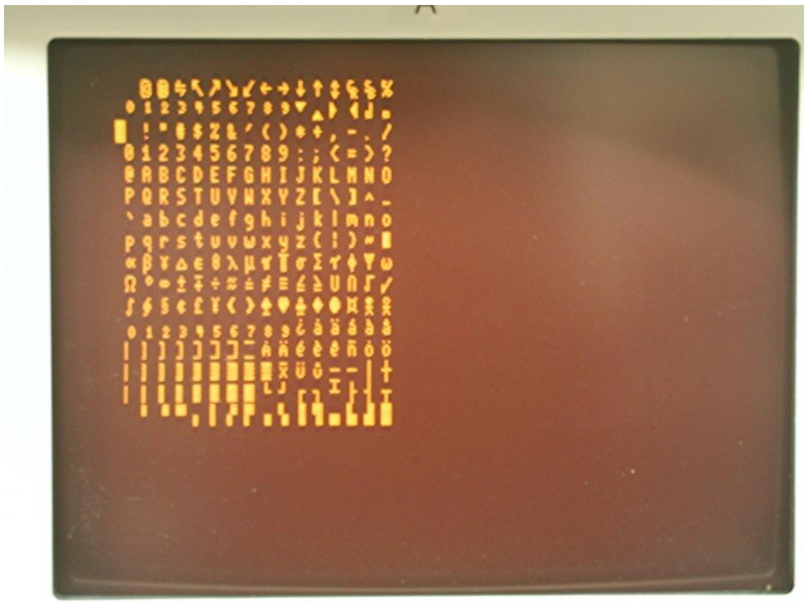
It is important in both cases, after re-entering BASIC <6> key, to type NEW, to make sure all of the pointers are correct.

The photo below is a screen shot running the CRTC card and using BASIC in 72 column mode:

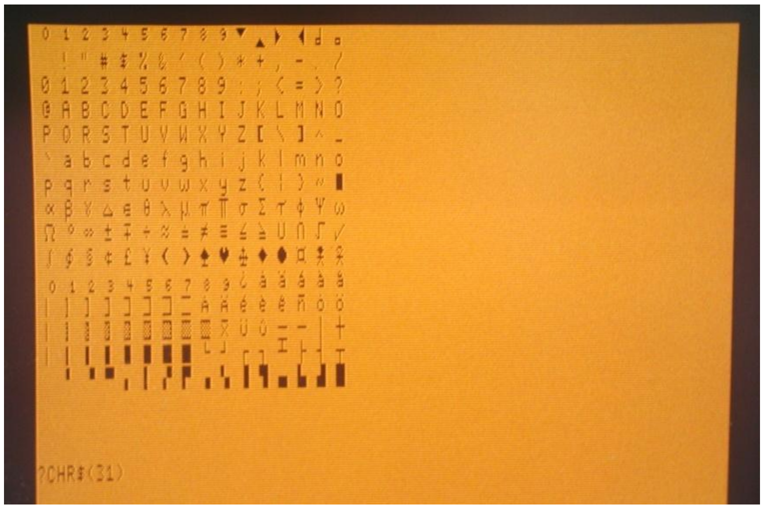


```
<M>=035E 38 03 02 03
</> 035E 48
<6>
10 FOR X = 1 TO 38
20 PRINT X;
30 NEXT X
40 END
RUN
 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20
 21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38

LIST

 10 FOR X = 1 TO 38
 20 PRINT X;
 30 NEXT X
 40 END
█
```

As noted, there are various control codes which alter the function of the CRTC card, if CTRL_ is sent to the card, it toggles the card to produce reverse video or black characters on an illuminated background. Since there is no "_" on the AIM-65 keyboard, another way to do it is to type ?CHR$(31) , each time this is done it toggles the screen from normal video to reverse video. Many of the control commands, such as CTRL-C , that clear the screen for example are available from the keyboard. The command ?CHR$(30) invokes a test to check the ROM characters:



At the hardware level , the video is inverted with an XOR gate on the CRTC card which is driven by a flip flop which is toggled each way by issuing the PRINT CHR$(31) or the ?CHR$(31) command in BASIC.



```
?CHR$(31)
```

(The CTRL _ has the ASCII code 1F = 31 decimal)

ADDITIONAL NOTES FOR RM-65 CRTC SETUP:

(The following notes were provided by dfnr2 (Dave) which assist the CRTC setup)

Configuring the hardware:

On-board firmware (Manual p 2-7, table 2-5)

Enable the on-board firmware by setting jumper E1 to position "A"

ROM base address (Manual page 2-5, Table 2-2)

The ROM is mapped with switches 1-4 of DIP switch S1. These select one of 16 4K blocks from $0000 to $F000, with the high bit at S1-4. An open switch is a "0" , so all open maps to $0000, and all closed maps to $F000.

To use the built-in firmware, the ROM must be mapped to $9000:

S1-4: 1 (closed)
S1-3: 0 (open)
S1-2: 0 (open)
S1-1: 1 (closed)

ROM Program enable:

Set Jumper E3 to the GND position to enable the firmware at $9900.

Dot clock jumper (E2):

For 40 columns (6 MHz), set to position A
for 80 columns (12 MHz), set to position B

Bank select:

Switches S1-5 and S1-6 are for a bank-addressing scheme. For most systems without bank-select hardware, just set S1-6 to OPEN, and S1-5 is ignored.

The CRTC firmware:

The CRTC firmware consists of initialization code, and a display routine (a crude terminal emulation) that intercepts characters via the I/O vectors, displays the characters on the screen, then passes complete lines to the active system program (e.g., BASIC).

While the display program can move the cursor around the screen, the code only processes characters received since the last <CR>, and sends those characters on the the AIM-65 software. That means that while it's possible to move the cursor on the screen, it's not possible to edit pre-existing content on the screen and send that to the AIM-65 software (as with the PET and Apple II)

The ROM contains initialization code at $9900.

The overview of the start-up process is:

- set screen geometry
- set the CRTC to work with BASIC (or FORTH, PL/65, and other standard firmware)
- Jump to $9900 to initialize the CRTC

The screen geometry is determined by the value at $035A. From the manual (Appendix C-5):

Value	Rows/Cols	Screen Lines	Jumper E2
0	25 x 80	625	B
1	22 x 72	525	B
2	25 x 40	625	A
3	16 x 40	525	A
4	User def	525/625	A/B
>4	22 x 72	525	B

-

Line Length (CMAX):

The max line length is stored at $035E. The default value is 59(dec). This matches the editor and PL/65. For BASIC, this can be increased to 72, by placing byte 48 at address $035E. This must be done after the card is initialised, because initialisation always resets the byte at location $035E to 3B (59 decimal).

AIM-65 processing:

To use the CRTC with the various standard ROM languages, write $02 to $0357. For a detailed description, see page 3-3 in the manual.

So the sequence suggested in the manual is

In the monitor:
(store $05 in $035A (for 22x72, or use another value from the above table)

```
<M>=035A
</>=05
```

(store $02 in $0357)

```
</M>=0357
</>=02
```

(jump to $9900)

```
<*>=9900
```

<G>/.

When using the CRTC with BASIC, some extra steps are needed to avoid the BASIC Start-up routine or BASIC storage space from overwriting the CRTC variables.

To avoid the conflict, BASIC is cold-started first, and the start of BASIC storage is moved above the CRTC variables. Then, the CRTC is configured, and finally BASIC is warm-started:

Since the CRTC default line length is 59 and BASIC's line length is 72, one of these values must be changed so the two values match.

The following sequence is given on manual page 3-8.

```
<5>               (Cold start BASIC)
Memory size?       (Return to test memory size)
Width? 72
AIM 65 BASIC V1.1  (Hit ESC to return to monitor)

<M>=0305         (Set the geometry)
</>=05           (22 x 72)

<M>=0357        (CRTC should send the lines to BASIC)
</>=02

<*>=9900         (jump to initialization routine at $9900)
<G>=.

<M>=035E         (set line length to 72(dec) = $48)
</>=48

<M>=0073         (set the start of BASIC storage to $0401)
</>=01 04

<M>=0400         (zero out the first 3 bytes of BASIC storage)
</>=00 00 00
```

An assembly snippet to accomplish the same sequence is also listed in the manual and can be bound to a function key.

****