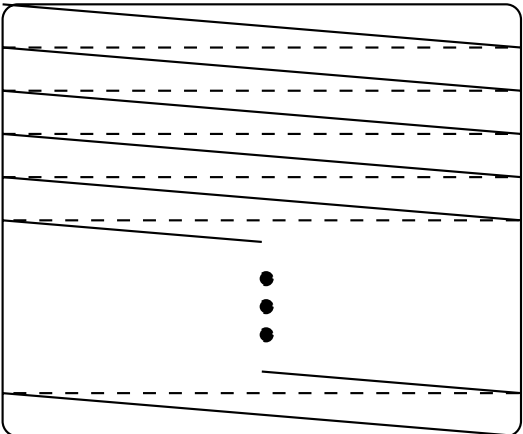
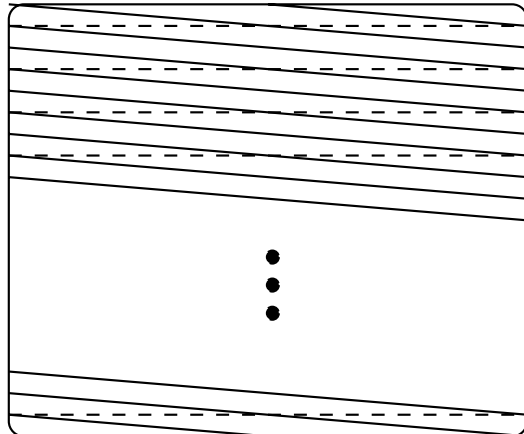
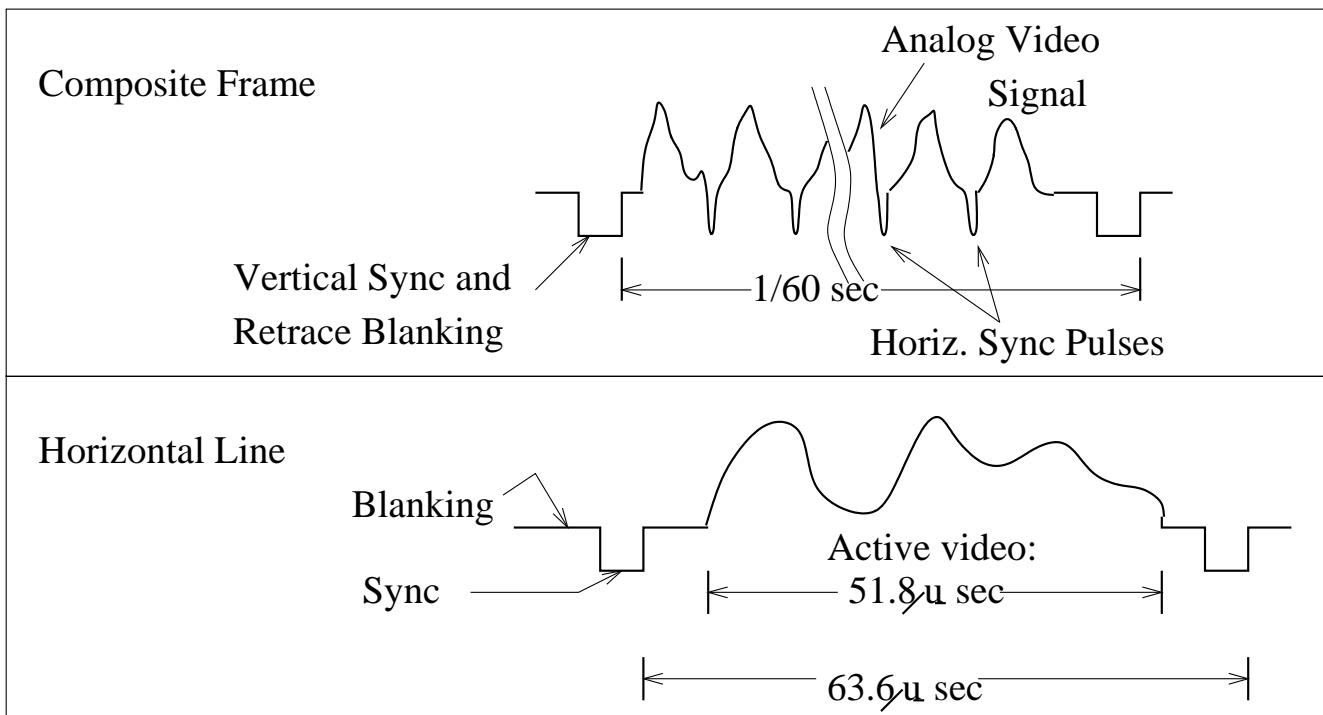
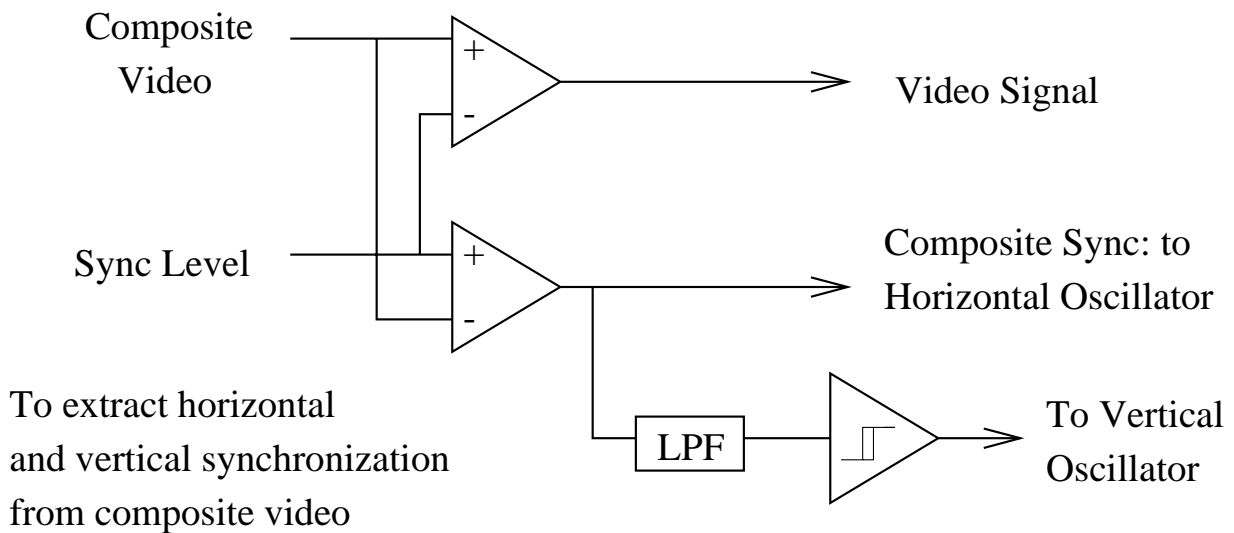
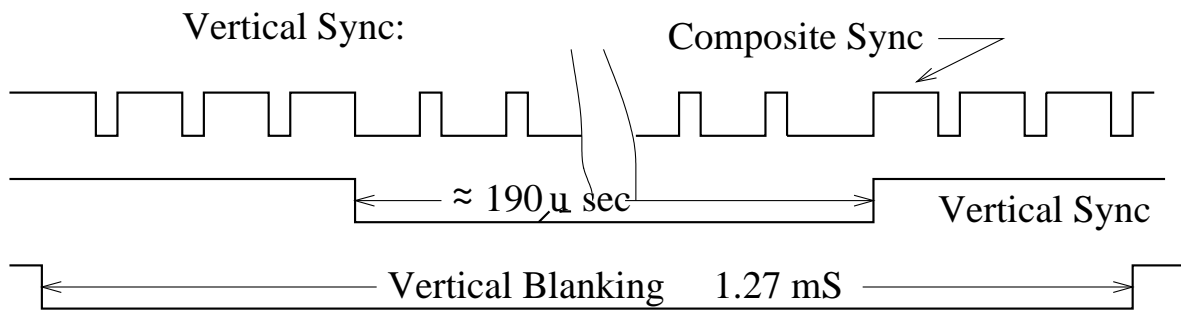
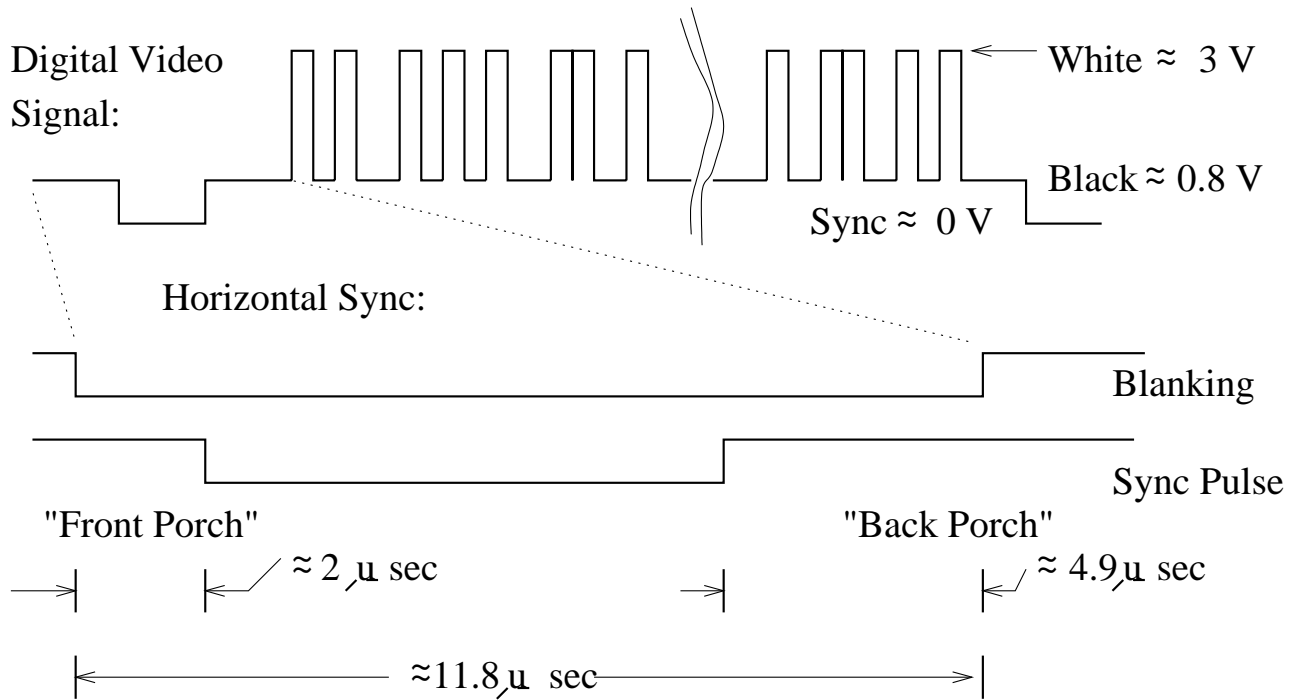


Video (Raster-Scan) Displays are like Television

	
<p>Non-Interlaced: Frame rate may be 60, 72, etc. frames/sec</p> <p>————— Scan line</p> <p>----- Retrace line</p>	<p>Interlaced. Frames alternate</p> <p>This is like Television: 60 half frames/sec.</p>

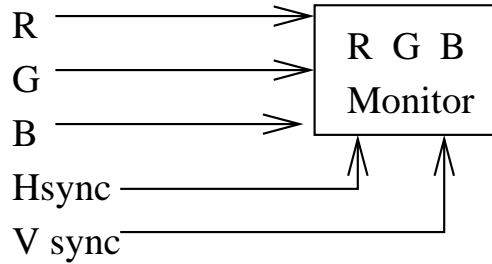
Electron beam "scans" tube. Beam location is shown here. Beam current determines brightness of display.



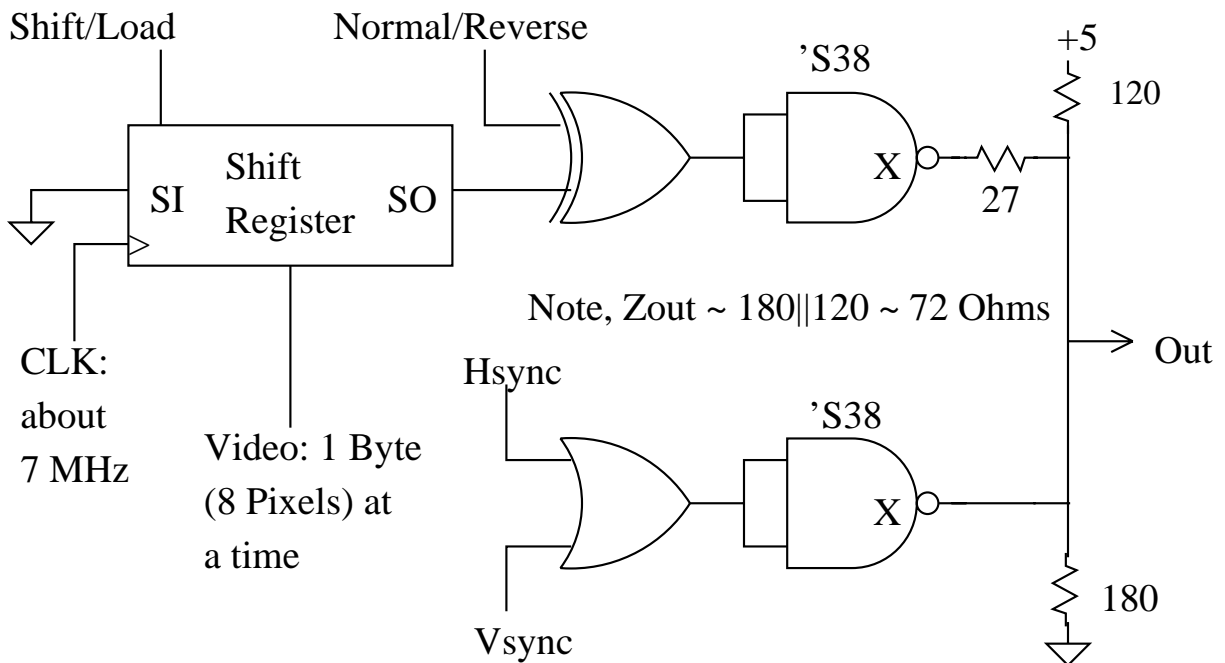


"Color" Television or displays are like 3 white-on-black systems operated together.

Sometimes sync signals are supplied separately.



	R	G	B
Black	0	0	0
Blue	0	0	1
Green	0	1	0
Cyan	0	1	1
Red	1	0	0
Purple	1	0	1
Yellow	1	1	0
White	1	1	1



Here is a simple video "Front End" to produce a composite, binary, white-on-black or black-on-white signal that is one bit deep.

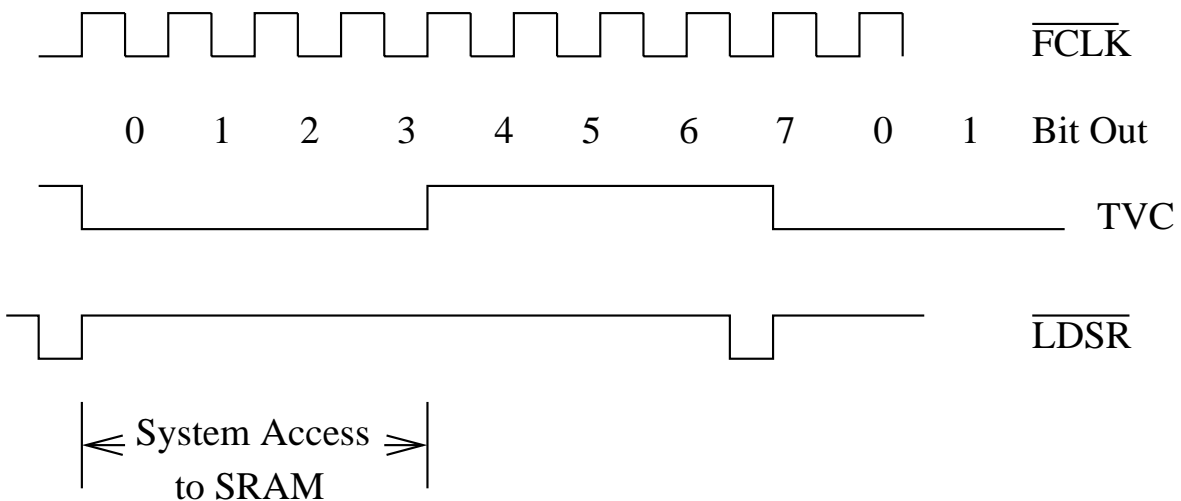
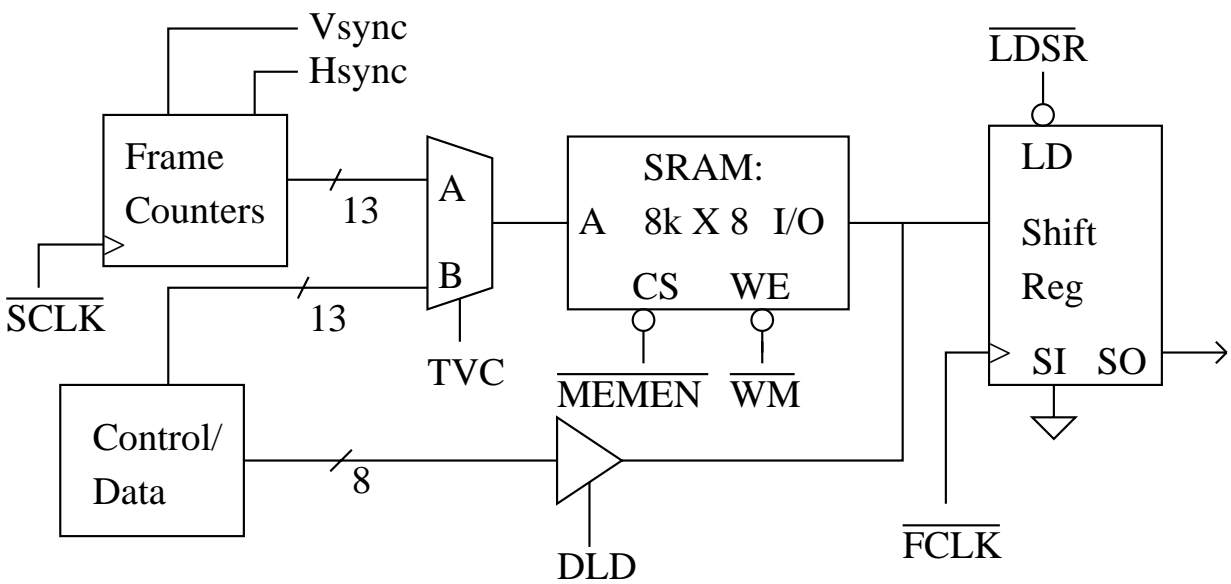
Here is one possible display format:

256 dots (pixels) across X 192 rows vertical

Clock = 7.16 MHz: .13966 microseconds per dot

Display time = 35.8 microseconds (active line)

256 X 192 = 48 X 1024 = 48 k pixels = 6 k X 8 bytes of memory



Horizontal and Vertical Addressing:

63.6 microseconds per row at 60 Hz: total of 262 rows

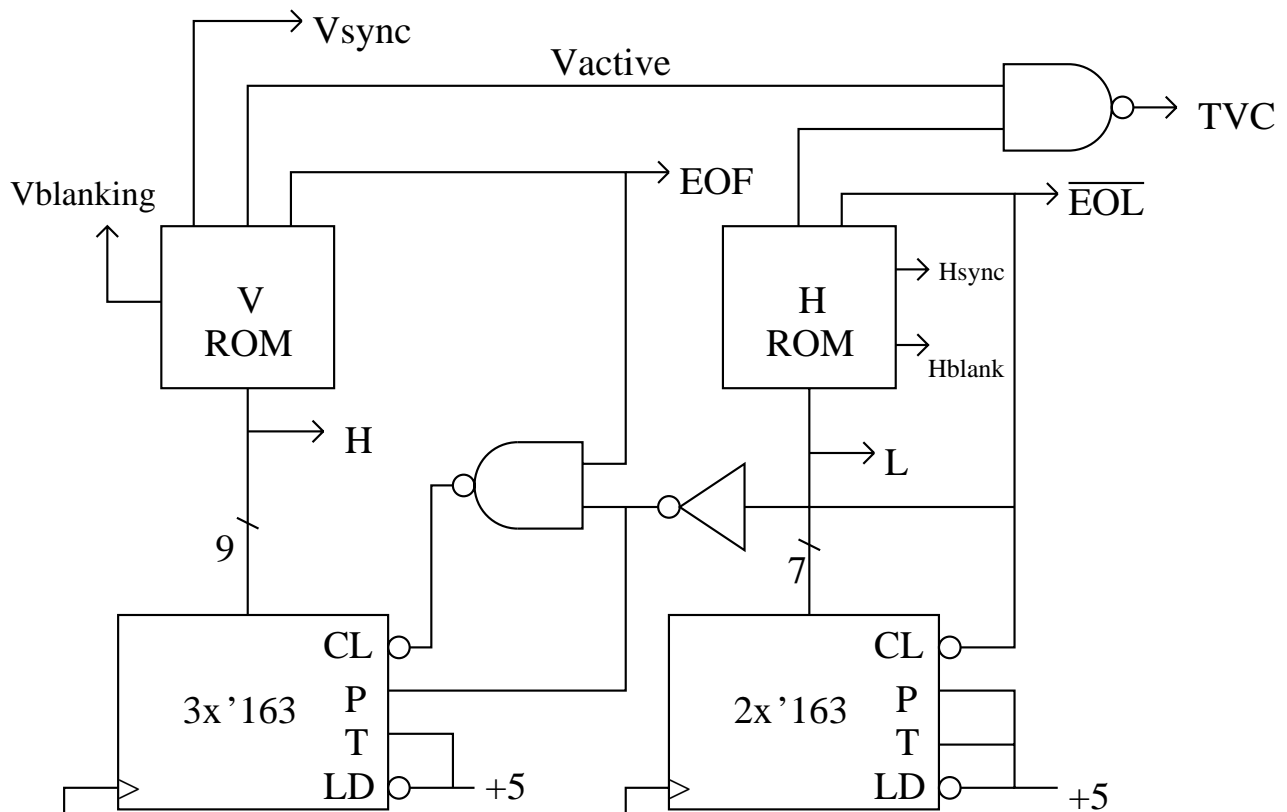
63.6 microseconds per row at 7.16 MHz: total of 455 dots/row

455 dots/row => 57 bytes (6 bits of counter)

262 rows => (9 bits of counter)

[Note that not all bits need to go to memory, since some addresses are outside of the displayed area.]

Here is a conceptual scheme for generating the necessary control signals:
Store information in ROMs (yet another use for them!)



$\overline{\text{CLK}} = \overline{\text{FCLK}} \div 4$

H: Bits 7:0 are high bits of memory address

L: Bits 5:1 are low bits of memory address

PROM Contents (for this example)

Vprom:	192	0 - 191	Vactive
	26	192 - 217	Vblanking
	6	218 - 223	Vsync
	37	224 - 260	Vblanking
	1	261	EOF
Hprom:	32	0 - 31	H active
	9	32 - 40	Hblanking
	7	41 - 47	Hsync
	8	48 - 55	Hblanking
	1	56	EOL

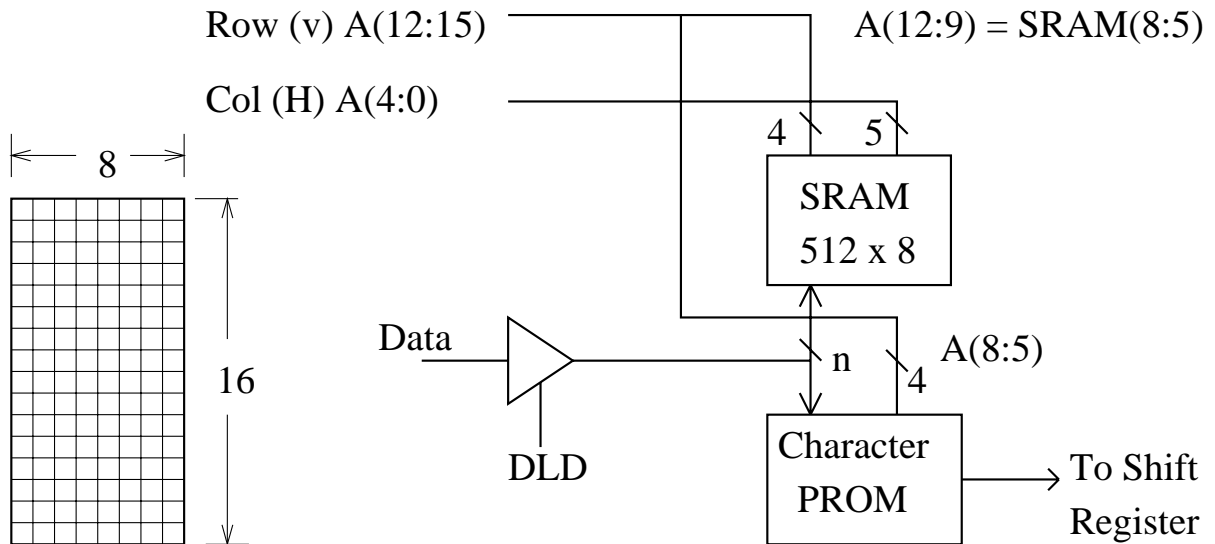
Blanking Locations (Not active) allow access to RAM

Character Displays

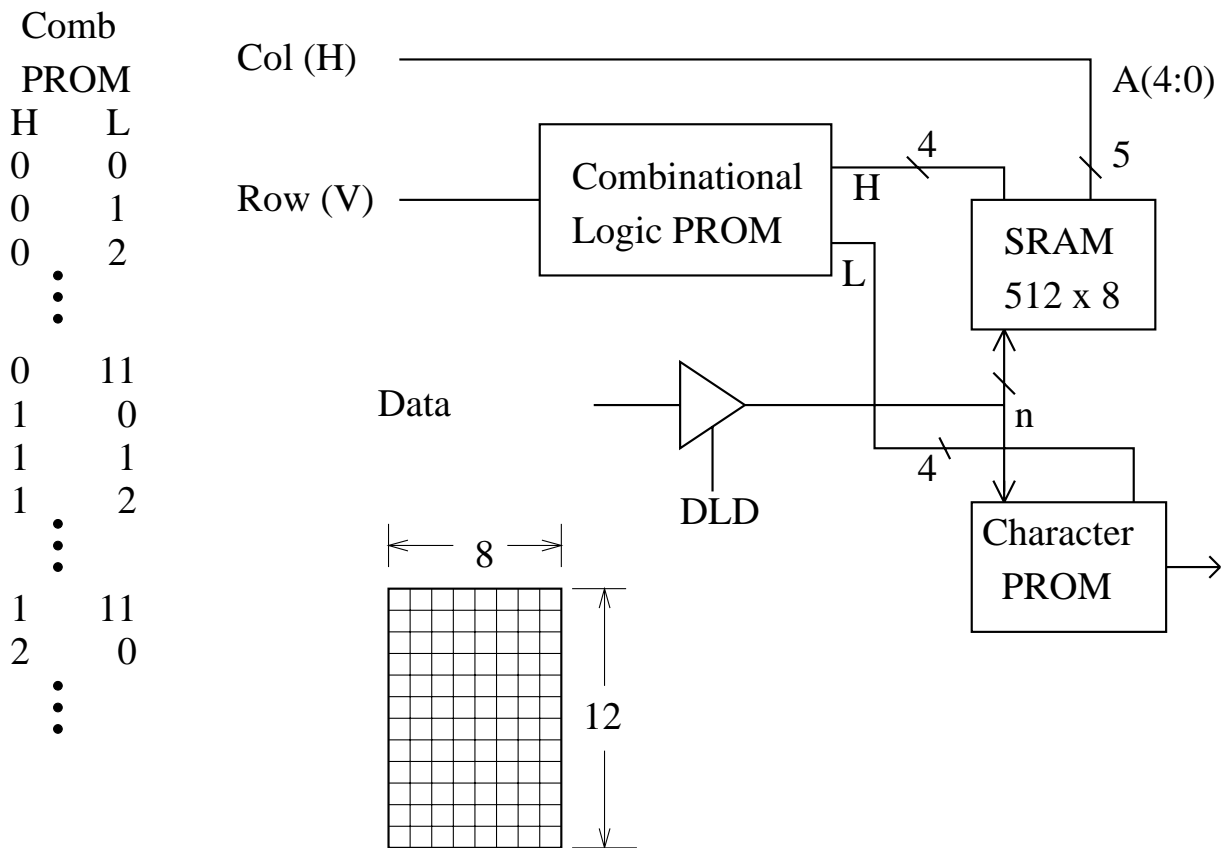
Fixed bit patterns which can be placed at different locations on the screen. The letters you are looking at are a good example. By pre-coding them they can save a lot of computation.

Two common examples: Assume a 256 x 192 screen

Char Size	Rows	Columns	Char Pos. (Memory per page)
8 x 16	12	32	384
8 x 12	16	32	512



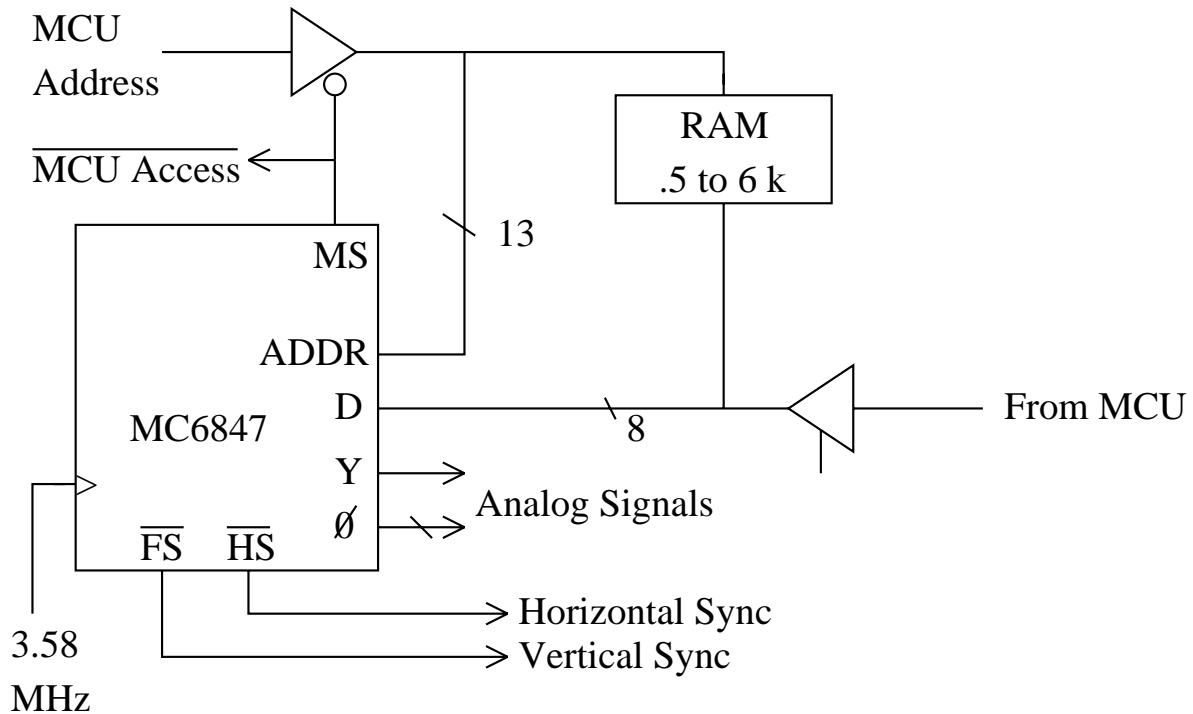
To do 32 x 16 characters (8 x 12 pixels each) a re-mapping of vertical address is required:



a.111 L18.8

March 29, 2000

Video Controller: MC 6847 Good for simple displays (Like Apple II)



The 6847 Provides a 13-bit address and reads 8 bit numbers and produces a video signal

Simplest mode is 256 x 192 pixels (sound familiar?)

This is "resolution graphics" mode. 1 bit per pixel, White on Black

MCU has access to RAM only during vertical blanking !