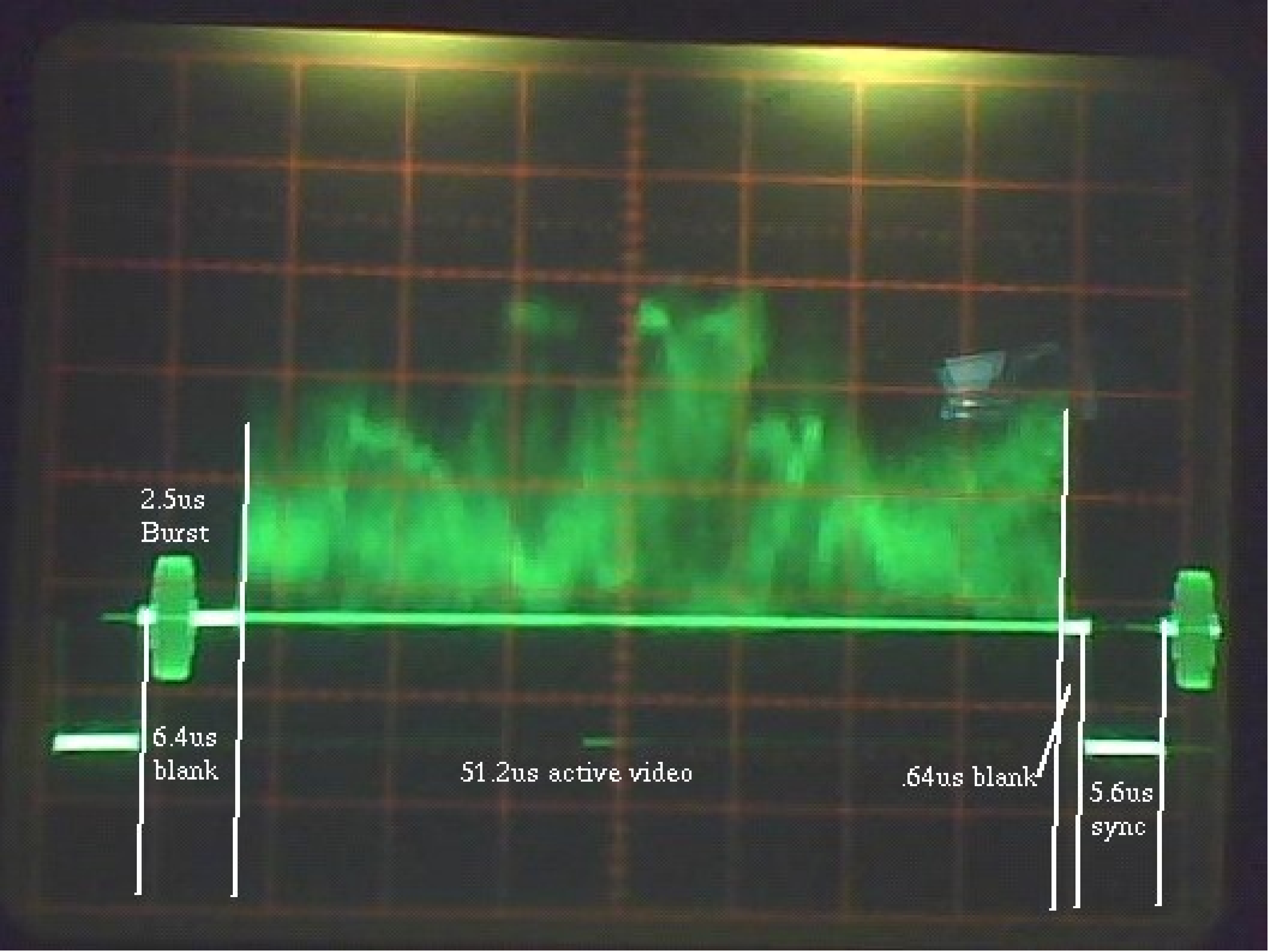
Converting Apple][NTSC to VGA

1. The NTSC TV Standard
2. Apple][video signals
3. The VGA “convention”
4. NTSC to VGA conversion
5. The Apple][VGA
6. Future thoughts

1. The NTSC TV Standard

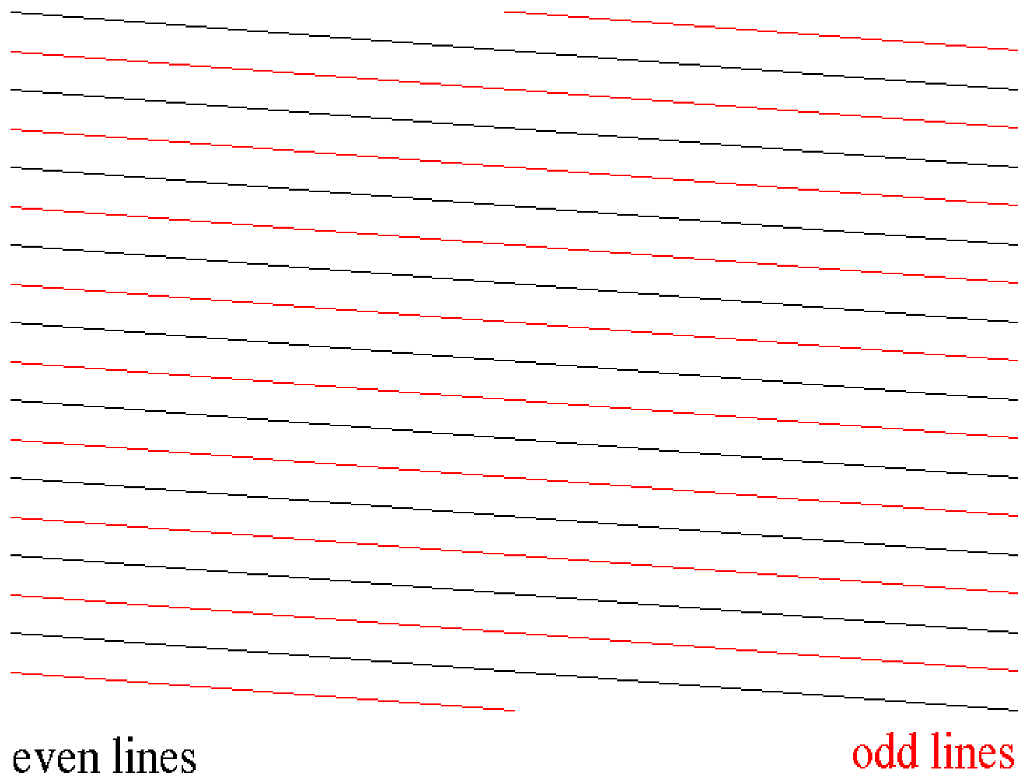
1. monchrome NTSC
   1. Timing
   2. The video signal
   3. Bandwidth
2. Color NTSC
   1. General Thoughts
   2. Making it compatible
   3. Chroma/Luma interference
   4. Quadrature modulation
   5. Receiver matrix


2.5us
Burst
6.4us
blank
51.2us active video
.64us blank
5.6us
sync

Vertical Timing

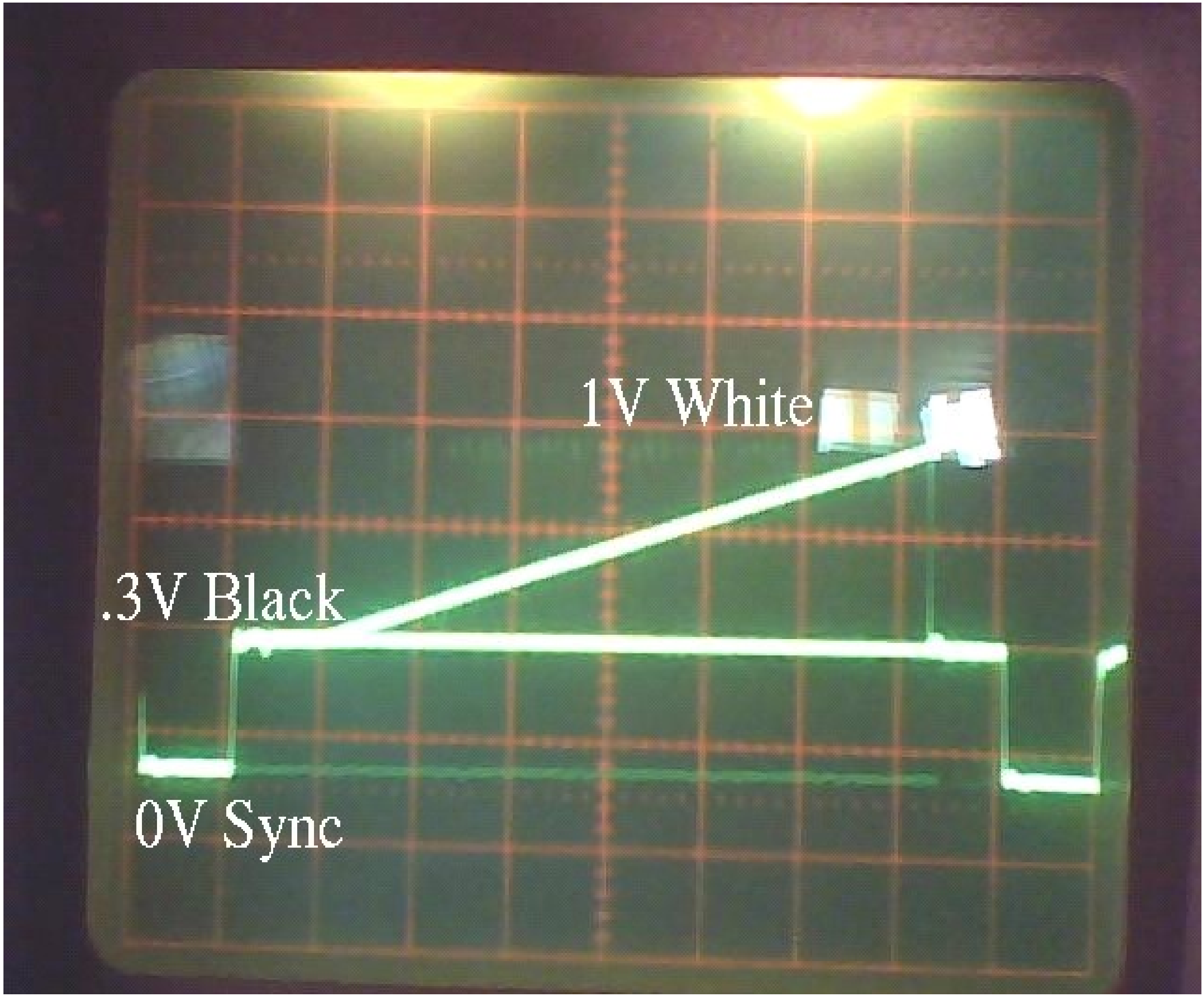
- 242 even lines active video
- 2 lines blanking
- 2 lines sync
- 16.5 lines blanking (retrace)
- 242 odd lines (starting and ending with ½ line)
- Another blanking/sync pattern

- Total 525 lines
- => 16.67 ms between syncs
- => 60Hz repetition rate
- Vertical rate has to equal mains frequency for hum suppression



1.1.2. The video signal

- Coax “CVBS” into 75 Ω:
- RF Power:
  - 100% Sync
  - 56 % Black
  - 1% White



1.1.3 Bandwidth

Analog TV has no “pixels”

Try to reproduce square dots at maximum vertical resolution

Aspect is 4:3 => 640 dots per line=>12.5 M dots/s

Video signal will oscillate at half dot rate(6.25 MHz)

NTSC chose 4Mhz to get 6Mhz channel spacing.

With introduction of color, effective bandwidth was reduced to about 2.5MHz.

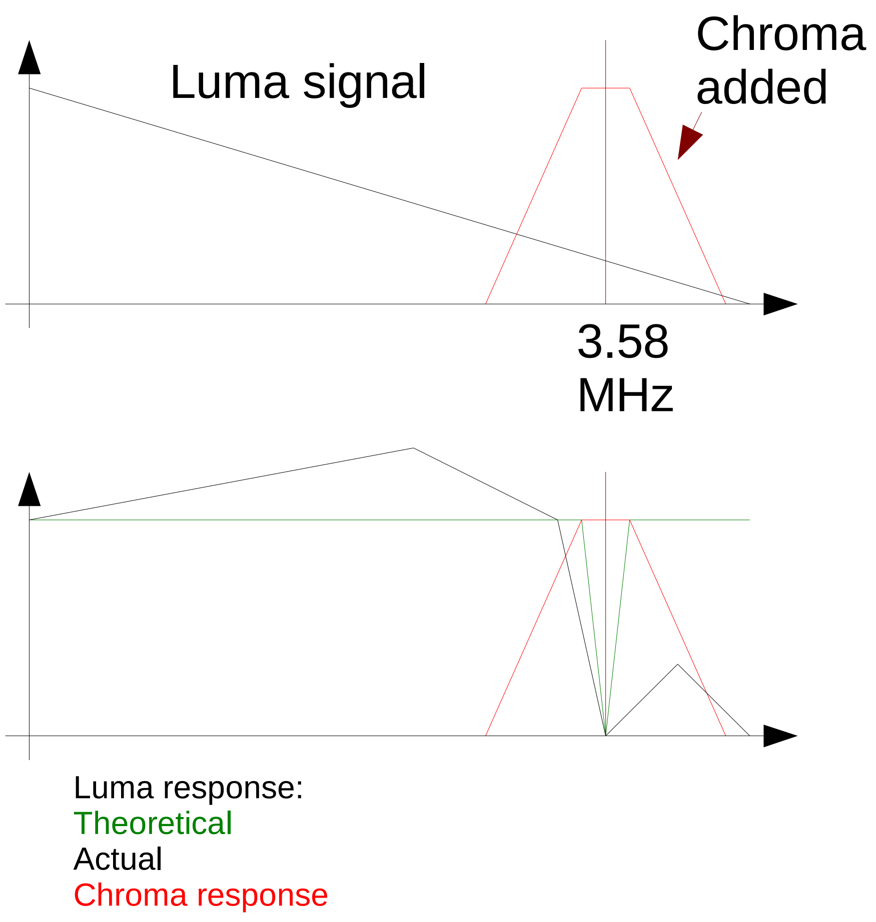
→ The limited bandwidth is insufficient for square pixel display (e.g. 80-column mode)

1.2.1 Color: General thoughts

- We want compatibility with existing B/W-TVs and transmitters
  - extra bandwidth will be very limited
- The color camera gives us three color signals: R,G,B (primary color signals)
- We need at least the brightness for a B/W TV: Y=R+G+B (We'll call it Luminance or Luma)
- Then we need two color signals. It was chosen to use U=R-Y and V=B-Y, because reduced bandwidth will not be very obvious with these.

1.2.2 Making it compatible

- Signal is split up in a luminance ("luma", Y) component and a two color components (U,V). A monochrome receiver displays this component ONLY.
- These are combined and modulated onto a 3.58 MHz subcarrier and form the chrominance signal ("Chroma", C)
- The signals are mixed before transmission
- The Receiver has to filter the signal to separate the two components



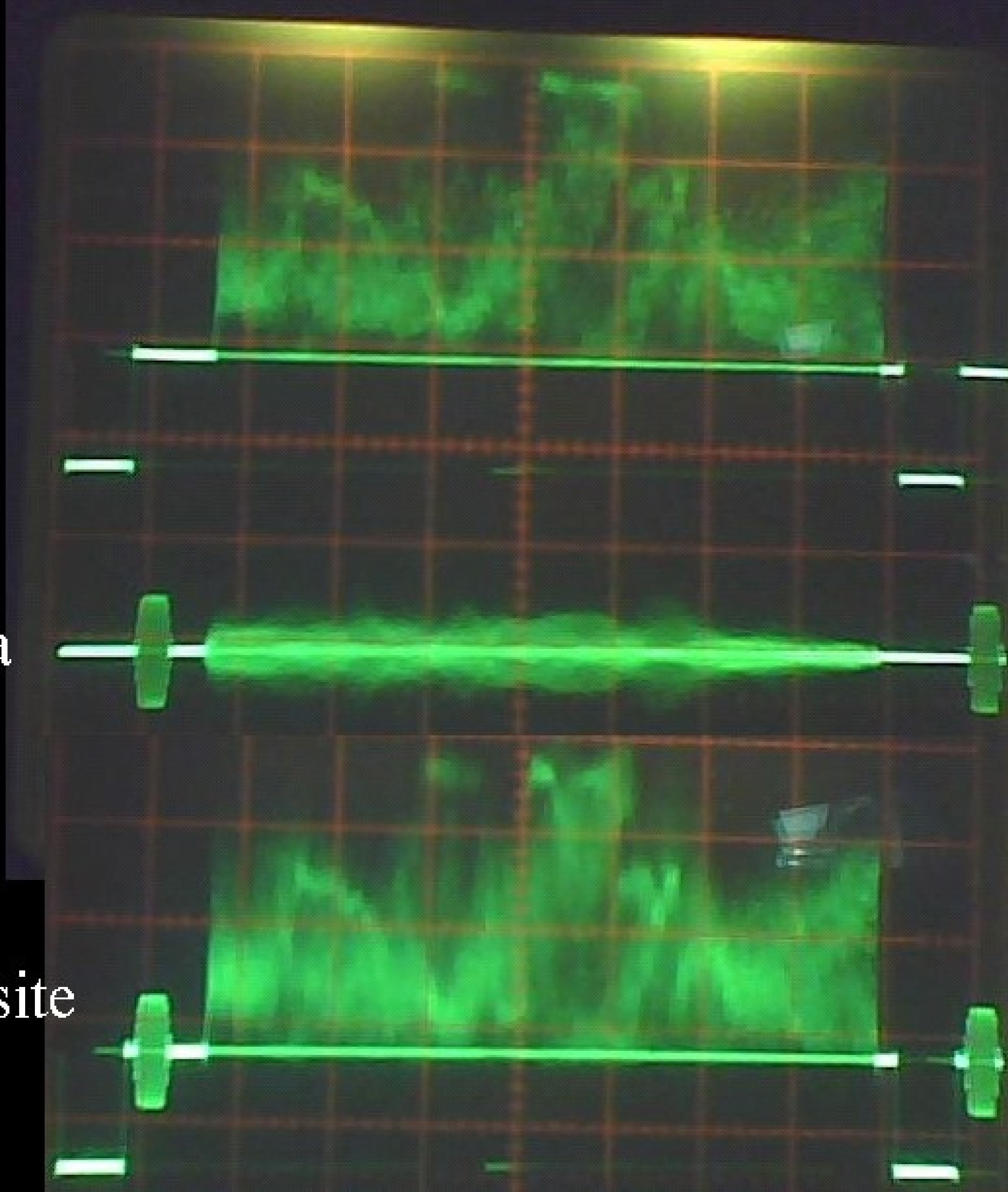
Luma

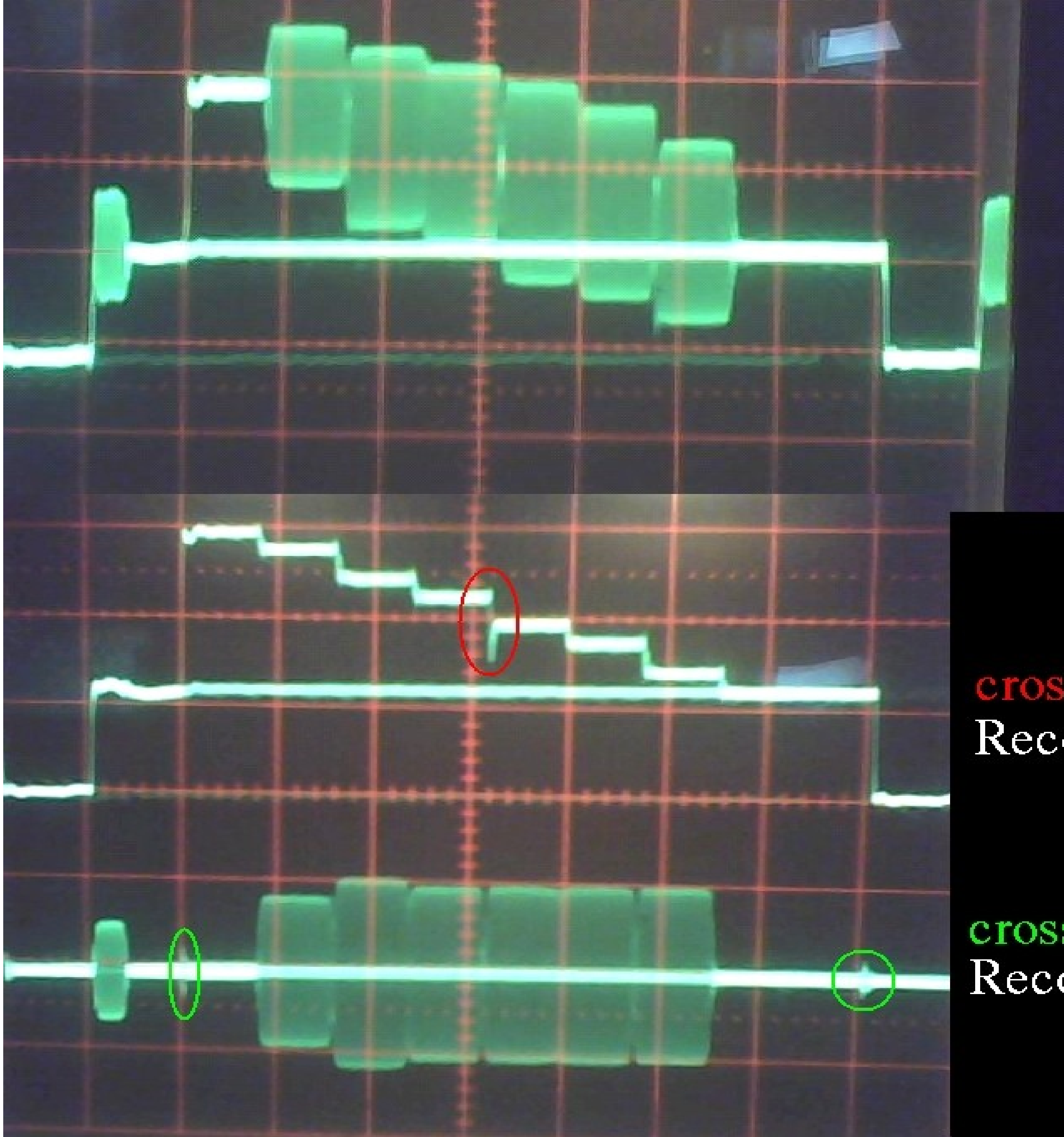
+

Chroma

=

Composite

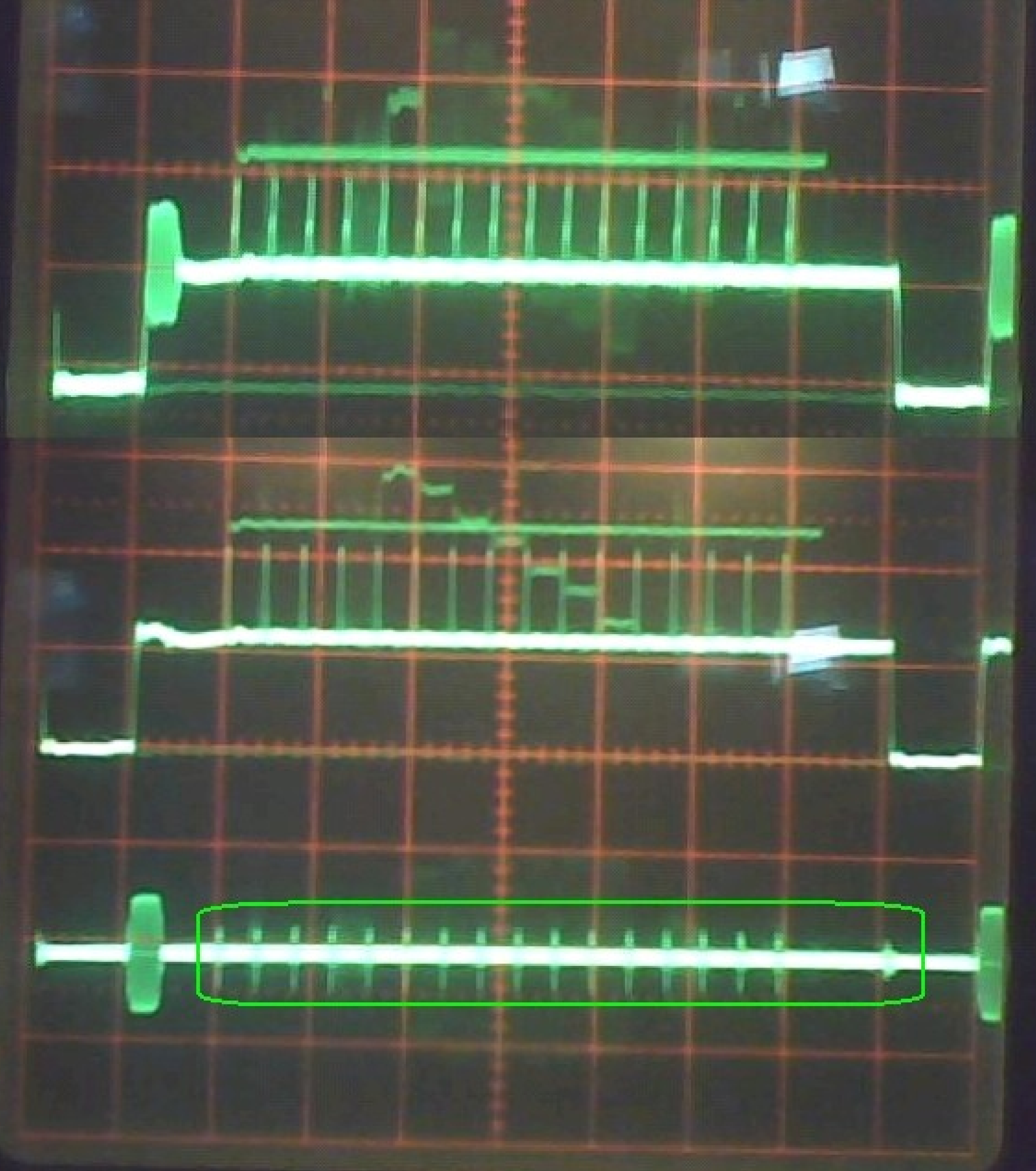




Composite

cross-color
Recovered Luma

cross-luma
Recovered Chroma



Composite

Recovered Luma

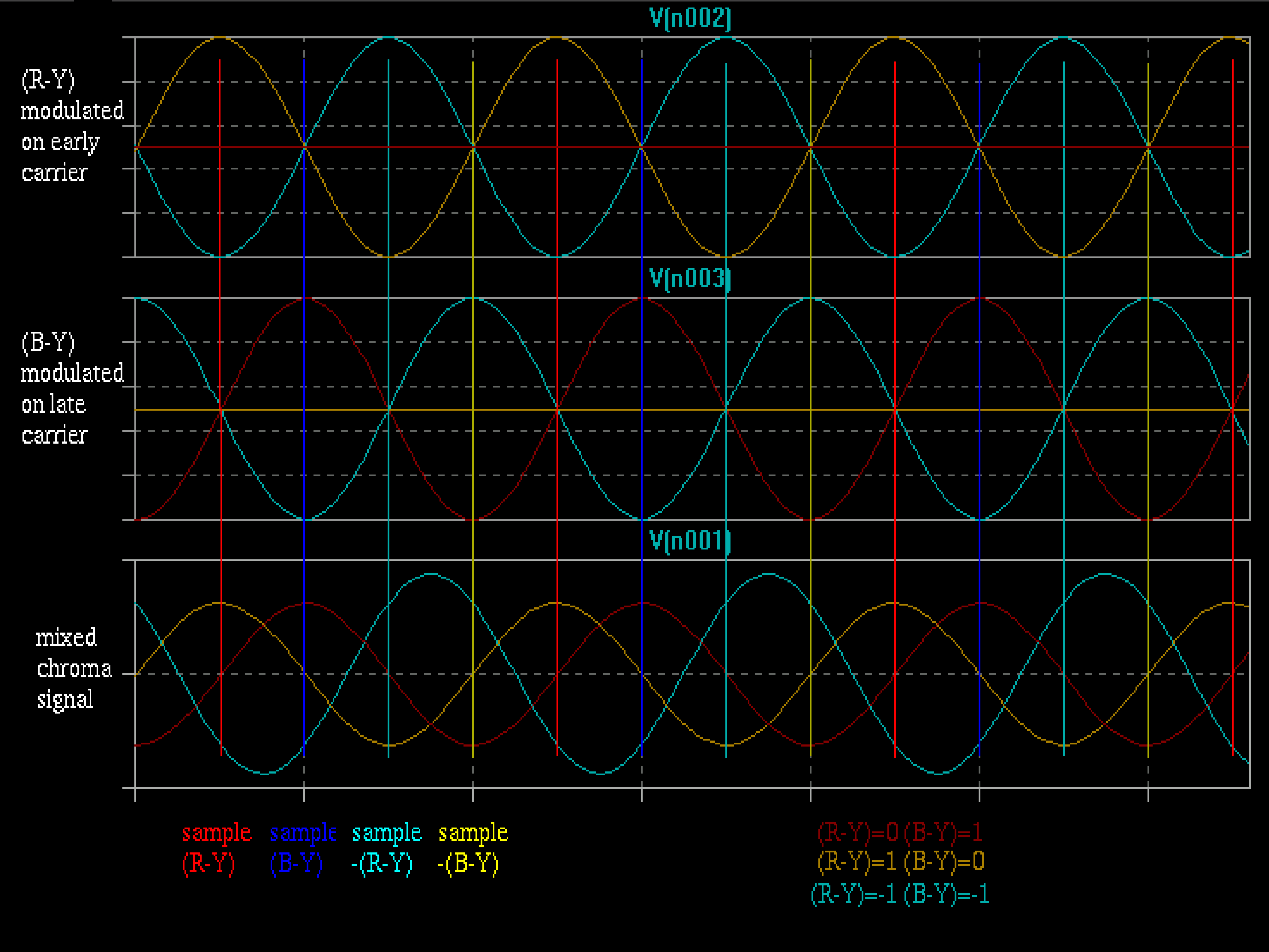
cross-luma
Recovered Chroma

1.2.3. Chroma-Luma interference

- The quality of a TV set (or its replacement – the Apple][VGA) boils down on its ability to separate luma and chroma.
- Cross-Color is visible as crawling patterns at color edges and as a pattern in intensely colored areas.
- Cross-Luminance is visible as color flicker in high detail areas.

1.2.4 Quadrature modulation I

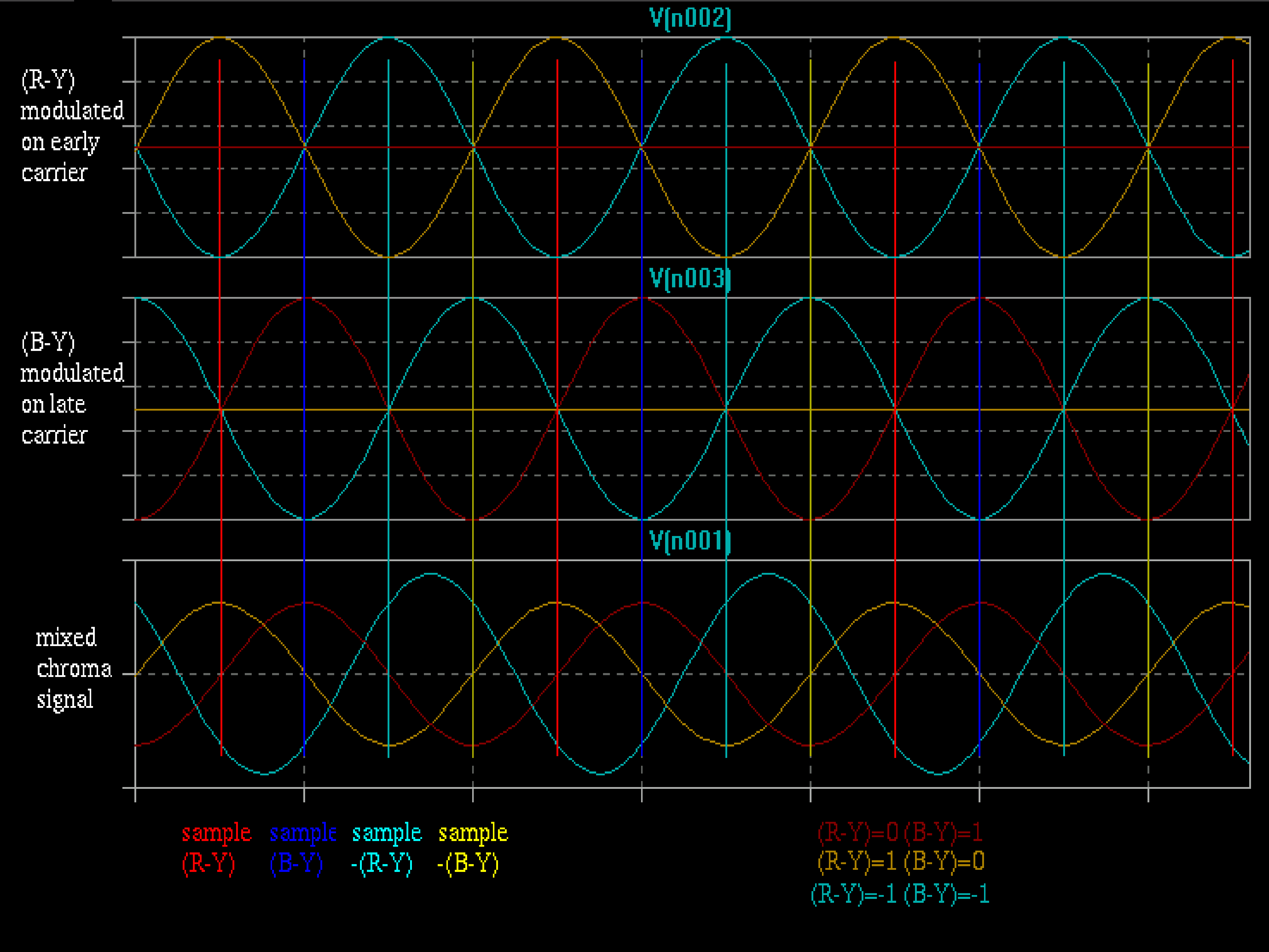
- Modulate 2 signals on 1 carrier
- 2 color carriers: 45° leading / 45° trailing
- Each carrier is multiplied by one color difference signal
- Both carriers are added.



V(n002)
(R-Y) modulated on early carrier
V(n003)
(B-Y) modulated on late carrier
V(n001)
mixed chroma signal
sample (R-Y)
sample (B-Y)
sample -(R-Y)
sample -(B-Y)
(R-Y)=0 (B-Y)=1
(R-Y)=1 (B-Y)=0
(R-Y)=-1 (B-Y)=-1

1.2.4 Quadrature modulation II

- Receiver recovers COLOR REFERENCE from the burst
- Phase error shall be <5° or visible color deviation will occur
  - Hue knob adjusts COLOR REFERENCE phase
- Demodulation by sample-and-hold
  - (R-Y)=”red”-”cyan”
  - (B-Y)=”blue”-”yellow”
- Low-pass filter


V(n002)
(R-Y) modulated on early carrier
V(n003)
(B-Y) modulated on late carrier
V(n001)
mixed chroma signal
sample (R-Y)
sample (B-Y)
sample -(R-Y)
sample -(B-Y)
(R-Y)=0 (B-Y)=1
(R-Y)=1 (B-Y)=0
(R-Y)=-1 (B-Y)=-1

1.2.5. Receiver matrix

- Convert U (R-Y) and V (B-Y) to R,G,B
- 1st matrix:
  - (G-Y)=-(R-Y)-(B-Y)
- 2nd matrix
  - R=(R-Y)+Y
  - G=(G-Y)+Y
  - B=(B-Y)+Y
- Combined matrix:
  - G=Y-(R-Y)-(G-Y)

2. Apple][video signals

1. Clocks
2. Text display
3. LORES/DHIRES colors
4. The GS

2.1 Video clocks

- Master clock (14M): 14.31818MHz
- TEXT/HIRES clock (7M):14M/2 (7.15909 MHz)
- COLOR REFERENCE: 14M/4 (3.5754545MHz)
- PHASE 0 (P0): 14M/14(1.023 MHz)
- Phase of CREF alters for every cycle of P0
- Every line ends with a “long cycle” that ensures a constant phase relationship between P0 and CREF

V(p0)
V(cref)
8.4V
7.8V
7.2V
6.6V
6.0V
5.4V
4.8V
4.2V
3.6V
3.0V
2.4V
1.8V
0.0µs
0.3µs
0.6µs
0.9µs
1.2µs
1.5µs
1.8µs
2.1µs
2.4µs
2.7µs
3.0µs

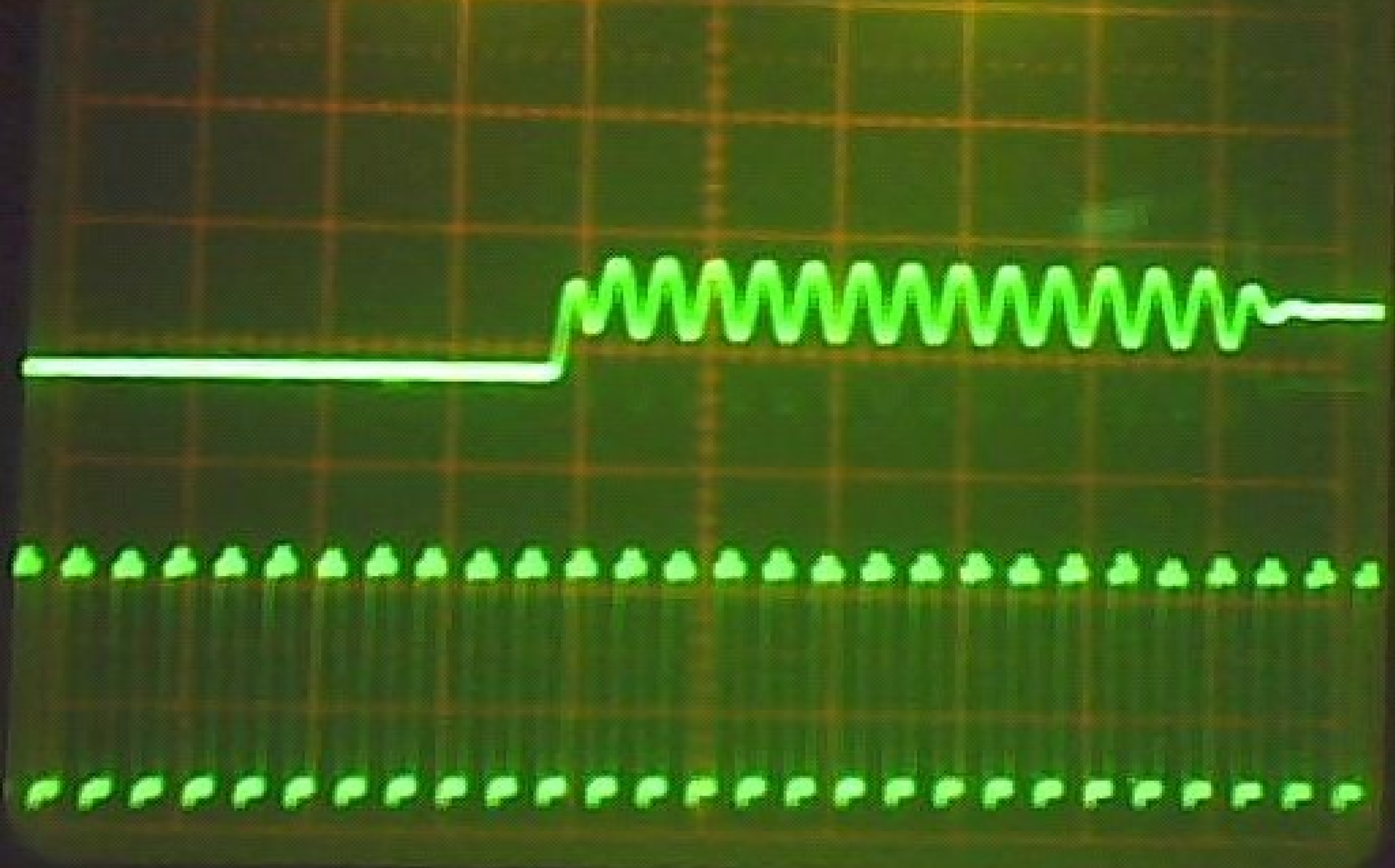
2.2 Text display

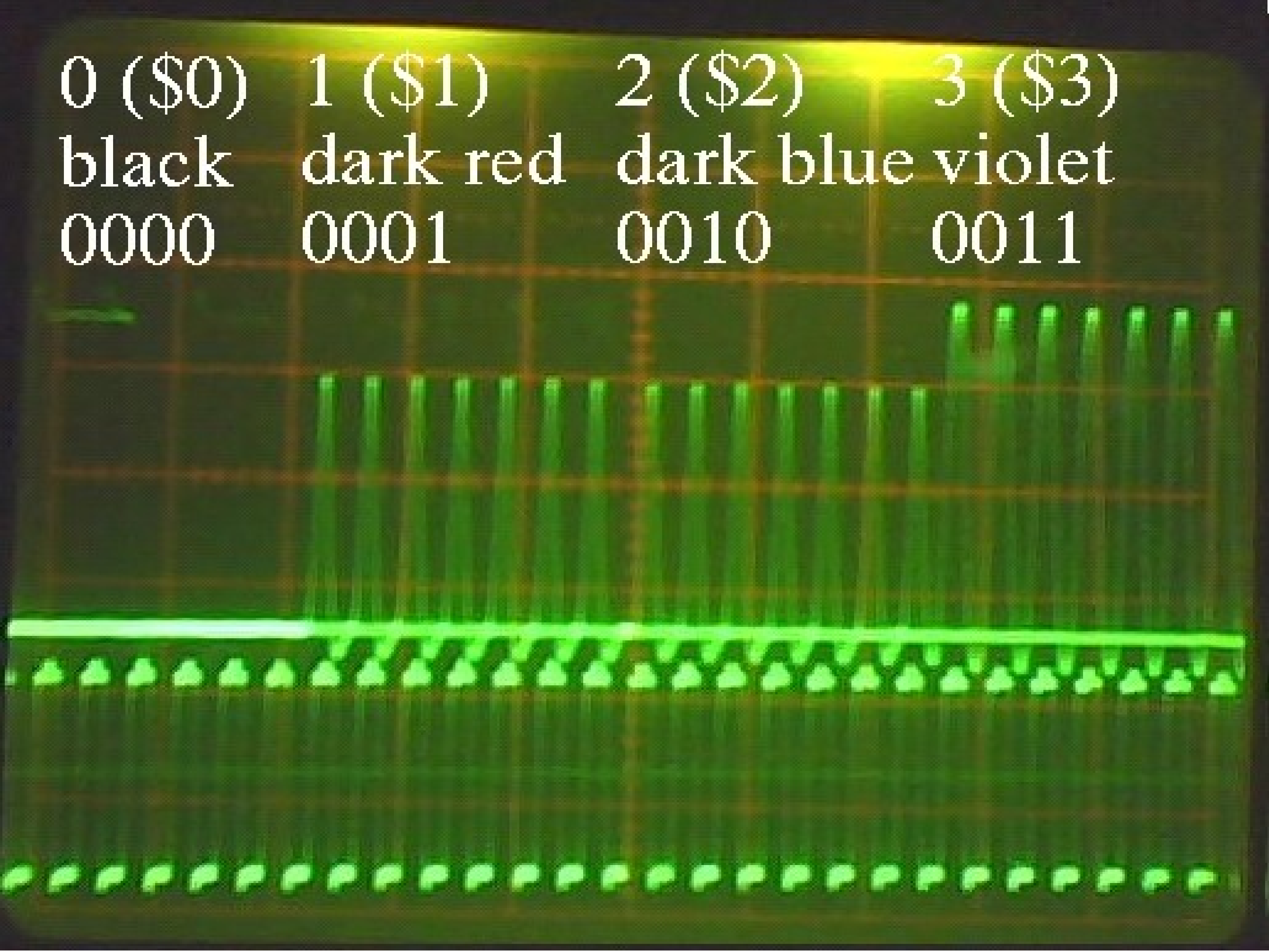
- The Apple][uses no interlacing. Every frame has 262 lines. Video is displayed in 192 lines. Other lines are blank.
- A line is 65 P0 cycles long. Video is displayed in 40 P0 cycles per line.
- 40-col TEXT: 1 character per P0 cycle, pixel clock 7M, minimum BW 3.5MHz
- 80-col TEXT: 2 characters per P1 cycle, pixel clock 14M, minimum BW 7MHz

2.3 Color graphics display

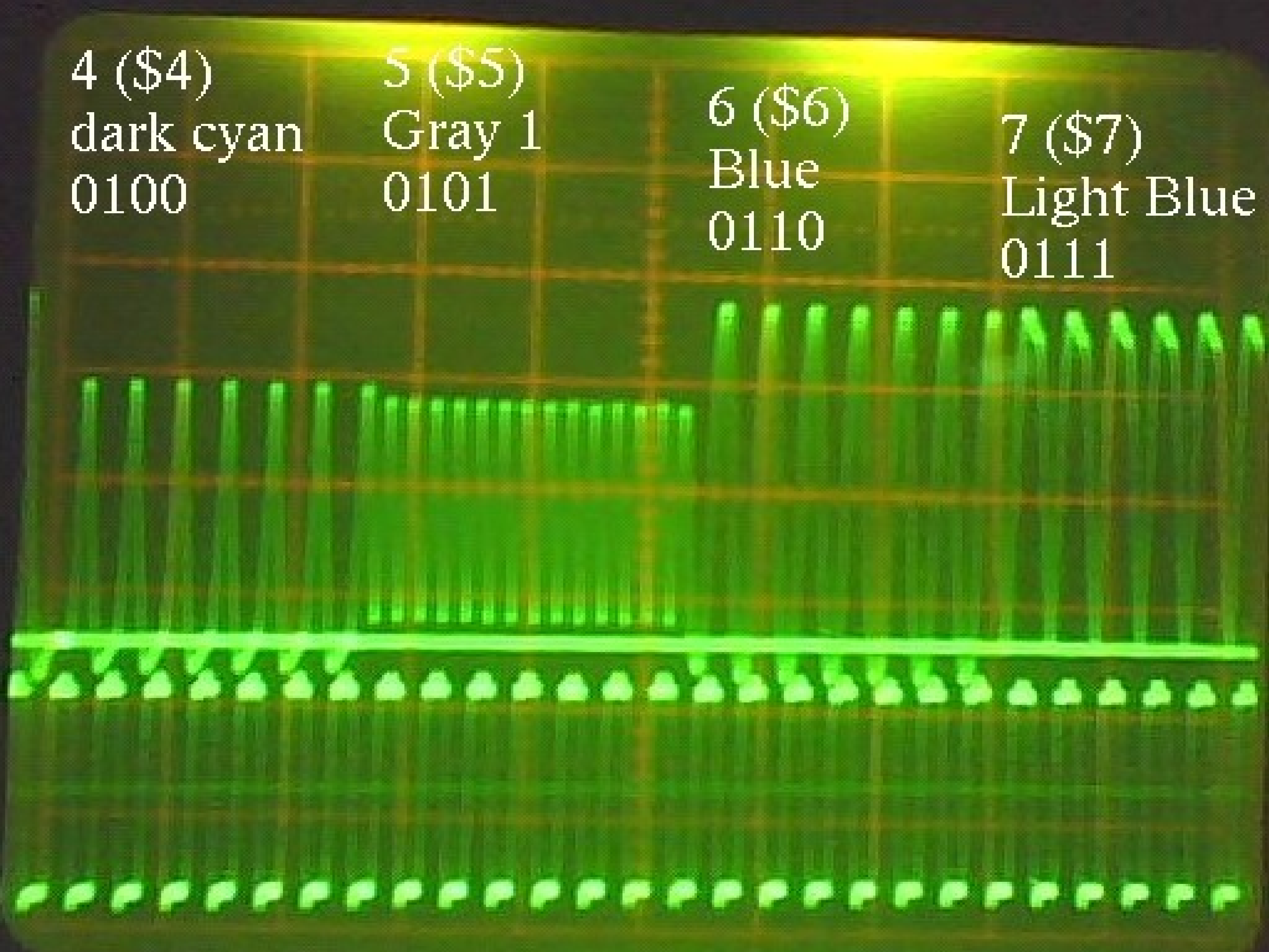
- Turn on color burst to enable TV's color circuit
  - Burst is inverted CREF so the TV syncs to **INVERTED** CREF
- **There is no chroma generator in the Apple][!**
- (D)HIRES: Generate bit patterns in software that will be interpreted as chroma signals
- LORES: Repeat a given bit pattern (LORES color number) in hardware
-

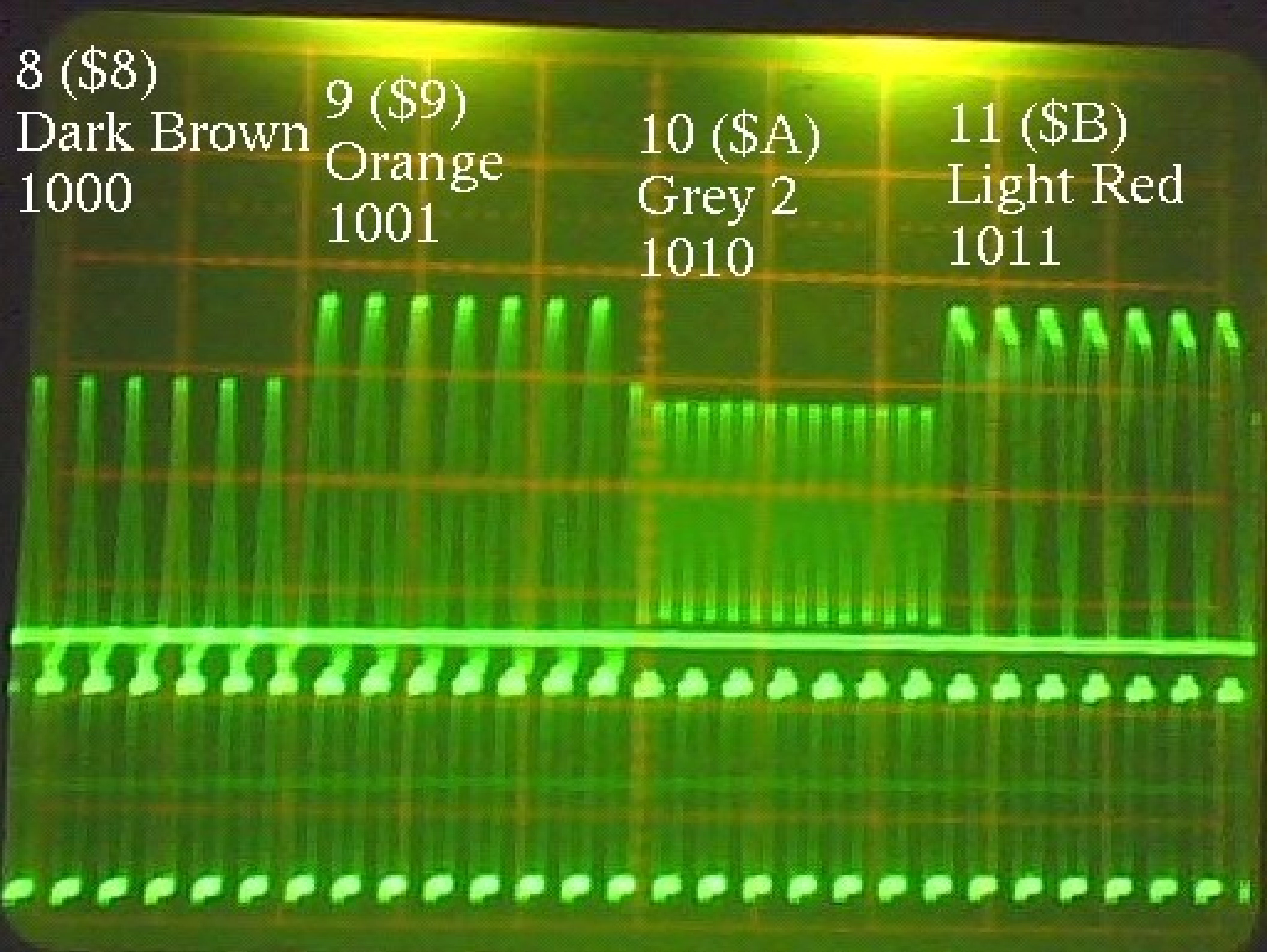
Apple][color burst

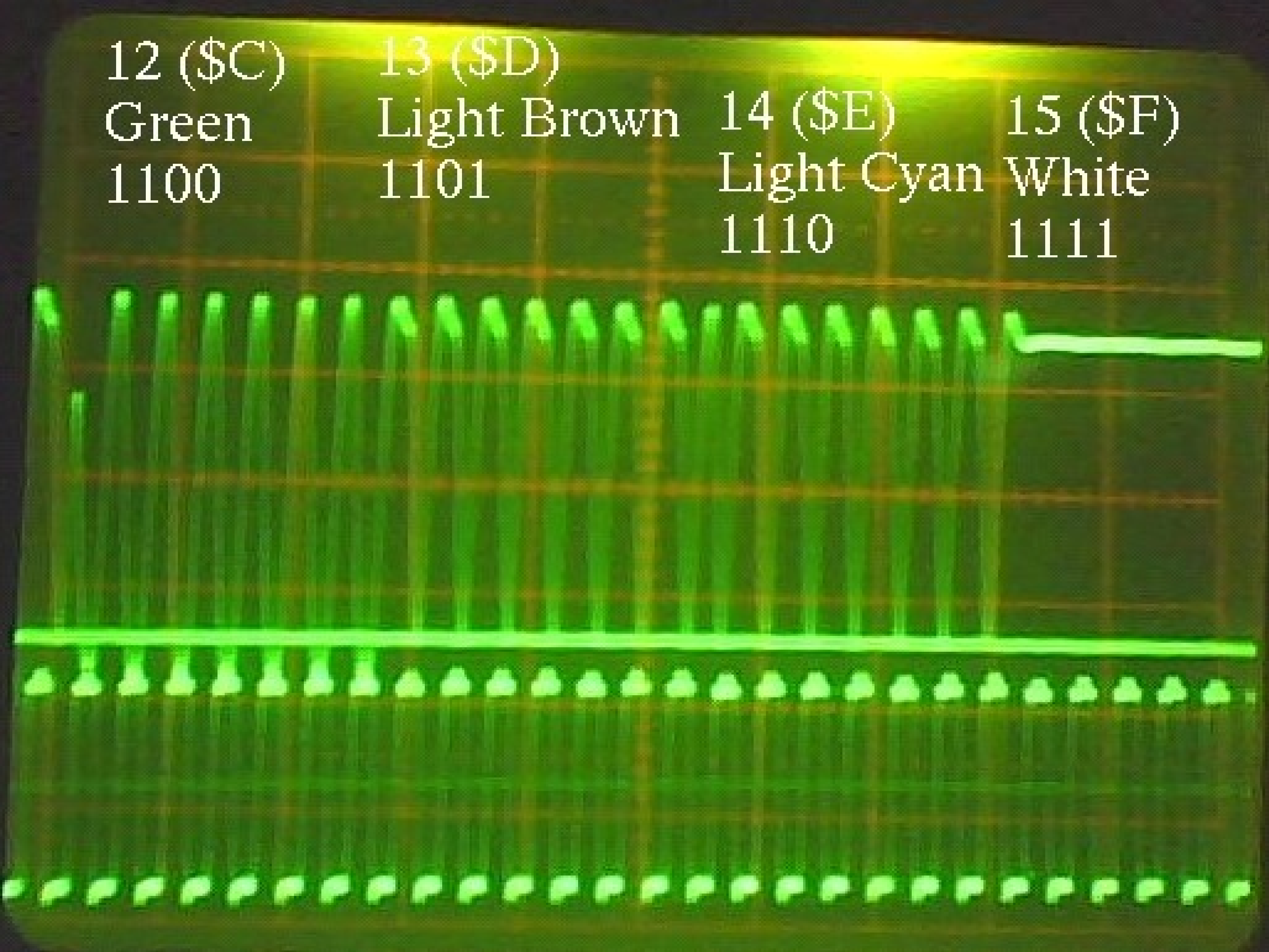




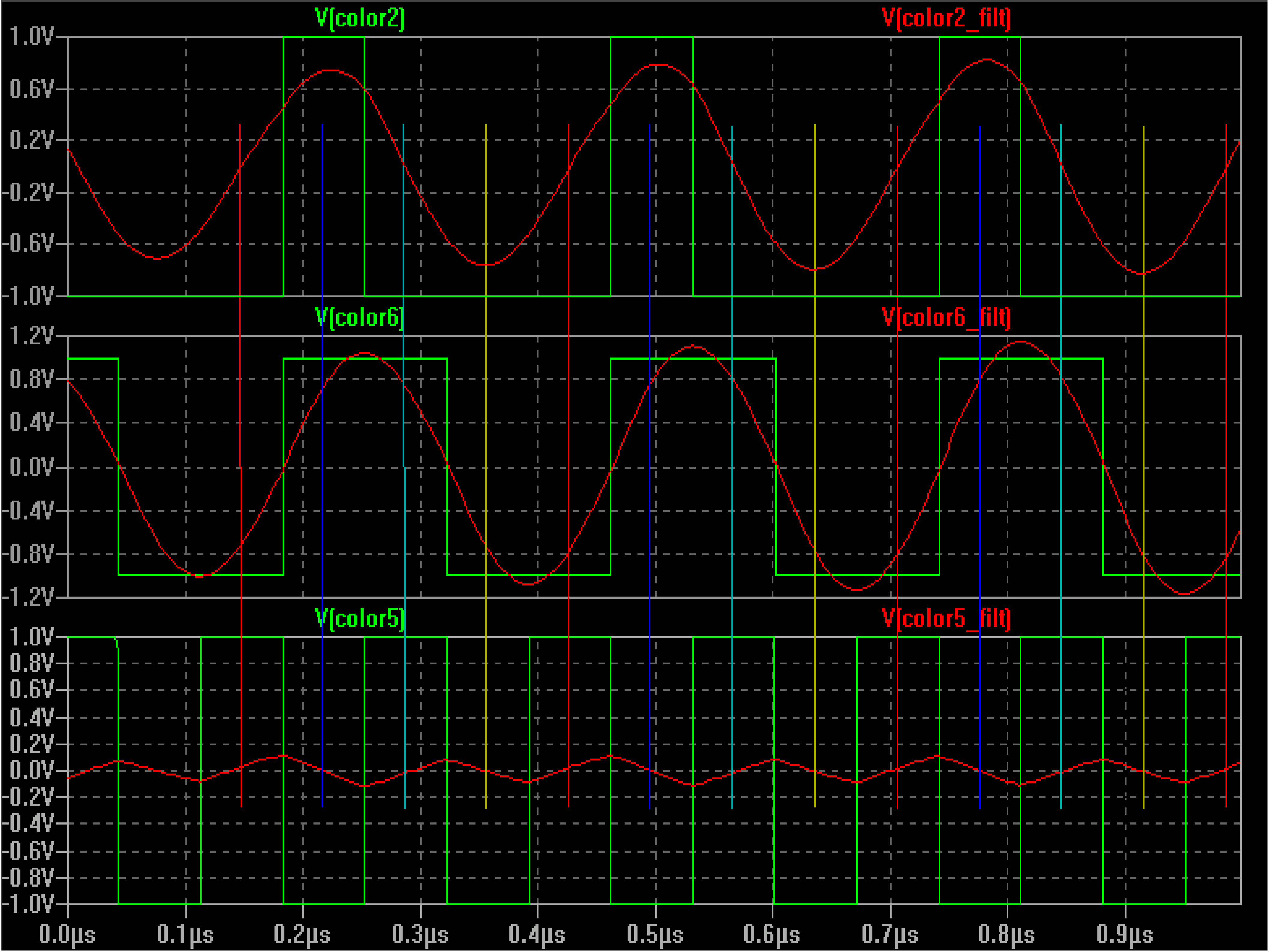
0 ($0)
black
0000
1 ($1)
dark red
0001
2 ($2)
dark blue
0010
3 ($3)
violet
0011


4 ($4)
dark cyan
0100
5 ($5)
Gray 1
0101
6 ($6)
Blue
0110
7 ($7)
Light Blue
0111


8 ($8)
Dark Brown
1000
9 ($9)
Orange
1001
10 ($A)
Grey 2
1010
11 ($B)
Light Red
1011



12 ($C)
Green
1100
13 ($D)
Light Brown
1101
14 ($E)
Light Cyan
1110
15 ($F)
White
1111



V(color2)
V(color2_filt)
1.0V
0.6V
0.2V
-0.2V
-0.6V
-1.0V
V(color6)
V(color6_filt)
1.2V
0.8V
0.4V
0.0V
-0.4V
-0.8V
-1.2V
V(color5)
V(color5_filt)
1.0V
0.8V
0.6V
0.4V
0.2V
0.0V
-0.2V
-0.4V
-0.6V
-0.8V
-1.0V
0.0µs
0.1µs
0.2µs
0.3µs
0.4µs
0.5µs
0.6µs
0.7µs
0.8µs
0.9µs

2.4 The Apple//gs

- True 12-bit RGB color system
- 4 pixel clocks: 7,8,14,16 MHz
- Video output timing like //e
- “old” modes are emulated by “Mega][“ chip and converted to RGB
- A MC1377 converts RGB to NTSC (and does a fairly poor job because of Apple saving $1)
- S-Video out is available as well as RGB

3. The VGA “convention”

1. Pinout and signal levels
2. CRT Timing
3. LCD Timing

3.1 VGA pinout and signals

Pin	Function	Level	Notes
1	Red Video	0.7Vpp	terminated 75Ω
2	Green Video	0.7Vpp	terminated 75Ω
3	Blue Video	0.7Vpp	terminated 75Ω
5-8,10	GND		Pin 8 open triggers “no signal” message
13	HSYNC	3.5Vpp (TTL)	no polarity
14	VSYNC	3.5Vpp (TTL)	specified
9,11,12,15	DDC	various	Plug'n'Play – don't care

3.2 CRT Timing

- The CRT cares about frequencies ONLY
- Pixel clock is a “soft limit”
  - Higher Pixel clock will result in a blurry picture

Monitor	Vsync	Hsync	Pixel clock
NTSC TV	60 Hz	15750 Hz	<5MHz
Green screen	60 Hz	15750 Hz	<15Mhz
CGA/RGB	60-70 Hz	15750 Hz	<15Mhz
Classic VGA	60-70 Hz	31500 Hz	<30MHz
Low-end VGA	50-100 Hz	30-60 kHz	<100MHz
High-end VGA	50-152 Hz	30-108 kHz	<400MHz

3.3 LCD Timing

- LCDs need to digitize the video signal
  - Controller measures H and V frequencies and guesses the pixel clock
  - Most controllers don't support modes the LCD would be capable of (e.g. 16 kHz RGB for the GS)
- LCDs have no size controls
  - Displaying at lower resolution will result in part of the screen being unused

Mode	**VSYNC**	**HSYNC**	**Pixel clock**
640x480	60 Hz	31.5 kHz	25.3 MHz
720x400	70 Hz	31.5 kHz	28.6 MHz
800x600	60 Hz	37.8 kHz	36 MHz
1024x768	60 Hz	48 kHz	65 MHz

4 Conversion NTSC to VGA

1. Line doubling
2. Timing generation

4.1 Line doubling

- Convert to 640x480 (60Hz V/31.5kHz H)
  - All parameters are equal except HSYNC which is twice as high
  - The IBM VGA card displays each line twice in CGA modes
- A basic converter has to
  - Read every line into memory
  - Output the previous line twice at double speed
  - Generate VGA sync signals
  - Decode NTSC color

4.2 Timing generation

- Extract VSYNC from composite sync pattern
- Time RAM read and write cycles
- Create correct Pixel clocks
  - Lock Pixel clock on HSYNC by PLL
  - Or cheat and get the clock from the computer
- Create VGA HSYNC from Pixel clock
  - Ensuring correct frequency (simple divider)
  - And phase (not so simple)

5. The Apple][VGA

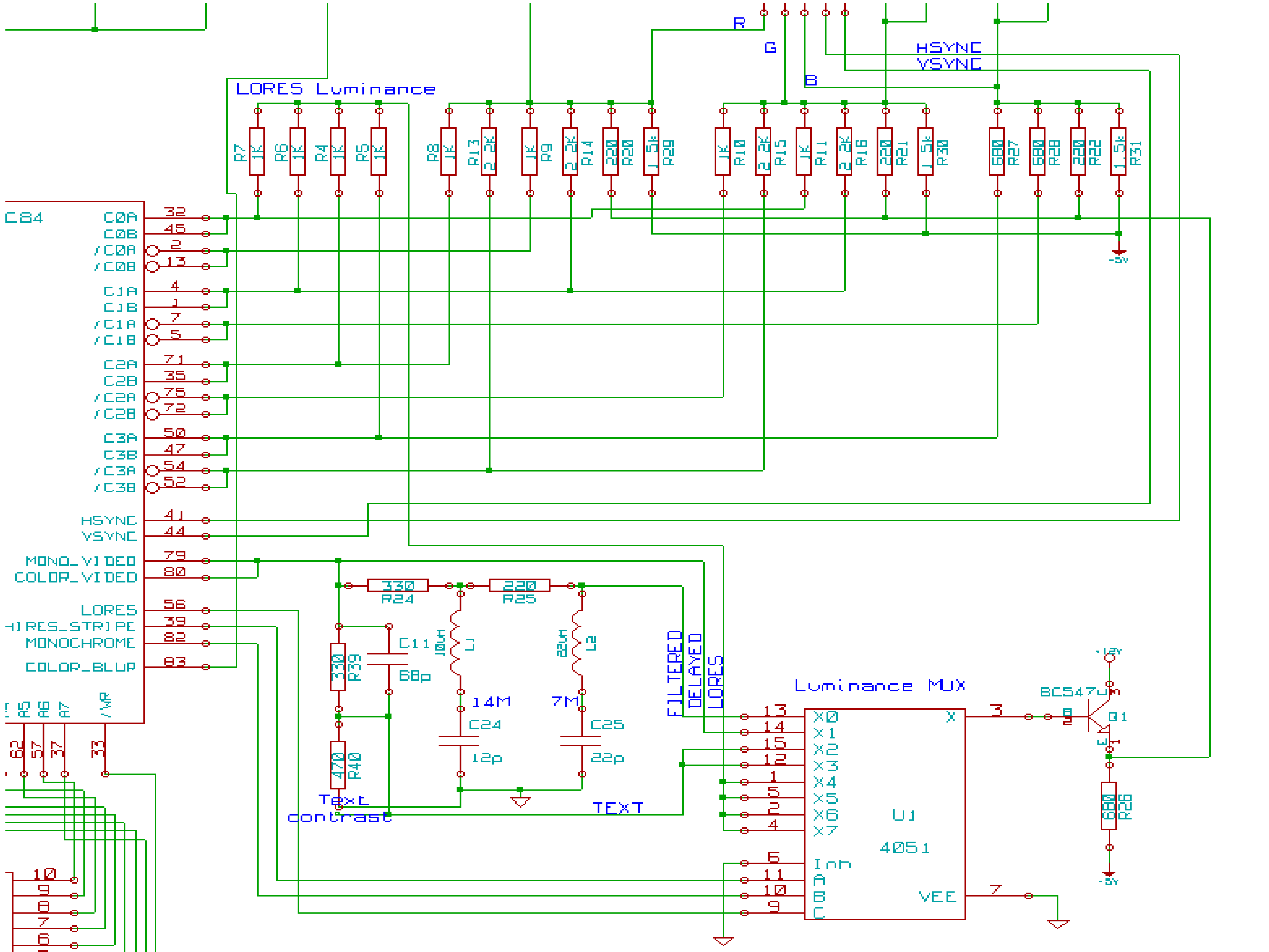
1. Line-doubling operation
2. NTSC decode
3. LCD issues

5.1 Apple][VGA linedoubling

- Double 14M to get 28M (VGA pixel clock)
- Sample the 1 bit video stream at 14M
- Combine 8 bits and store them in memory page 1 (every other CREF high)
- Read from page 2 back and output them (twice) at 28M (every CREF low)
- Swap pages every line

5.2 NTSC decode

- No extra chroma filter
  - So we still have a 1 bit stream
  - Sample/Hold can be a simple D-Flip/Flop
- 1st and 2nd matrix are combined
- Switchable chroma low-pass filter (by relay)
- Switchable luma filters (by 4051 MUX)
  - Additional RGB-card style luma signal from D-FFs
- Mode switching logic (in the CPLD) selects the filter settings after user requests and current display mode



LORES Luminance
R
G
B
HSYNC
VSYNC
C84
C0A
C0B
/C0A
/C0B
C1A
C1B
/C1A
/C1B
C2A
C2B
/C2A
/C2B
C3A
C3B
/C3A
/C3B
HSYNC
VSYNC
MONO_VIDEO
COLOR_VIDEO
LORES
HIRES_STRIPE
MONOCHROME
COLOR_BLUR
R24 330
R25 220
R39 330
R40 470
C11 68p
L1 10uH
L2 22uH
14M
7M
C24 12p
C25 22p
Text contrast
TEXT
FILTERED
DELAYED
LORES
Luminance MUX
U1
4051
X0
X1
X2
X3
X4
X5
X6
X7
X
Inh
A
B
C
VEE
BC547
Q1
R26 680
+12V
-5V

5.2.1 (D)HIRES decode

- Default: Filter luma and chroma
  - Luma filter is a 7M and a 14M (gray) trap
  - Chroma filter is 3 caps (1nF) between the R,G,B lines.
- Monochrome mode or unfiltered luma oe chroma both can be selected from software
  - Data is transferred through fast graphics mode switching patterns
  - Monochrome HIRES can be selected by jumper

5.2.2 LORES decode

- No chroma filtering
  - Filtering would create color fringes
- Luma taken from chroma Ffs (like RGB card
  - Luma is average of last 4 pixels

5.2.3 Filter control

- Chroma filters delay the signal
  - Chroma and luma need to be in phase
  - Switchable luma delay (shift reg)
- Mixed modes
  - Video output is one line late due to doublescanning
  - Mode switching would happen too early
- Software control

5.3 LCD issues

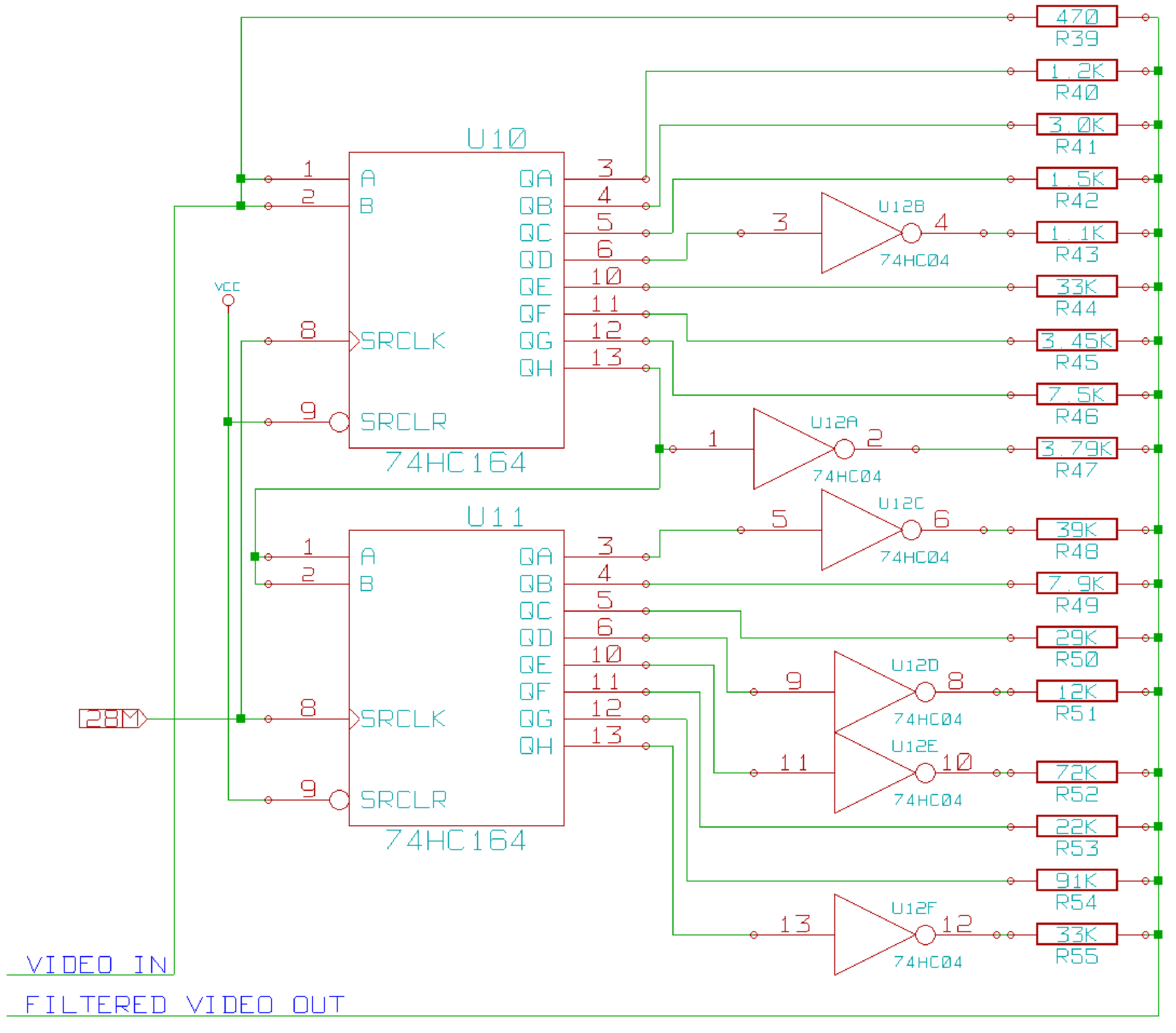
- Apple][VGA uses 28.6 MHz output clock
- LCD expects 25.3 MHz
  - LCD skips every 9th pixel
- LCD expects 640x480, not 560x192
  - Screen can not be filled completely
- Many LCDs fail to detect signal when running the A2VGA in a 50Hz apple

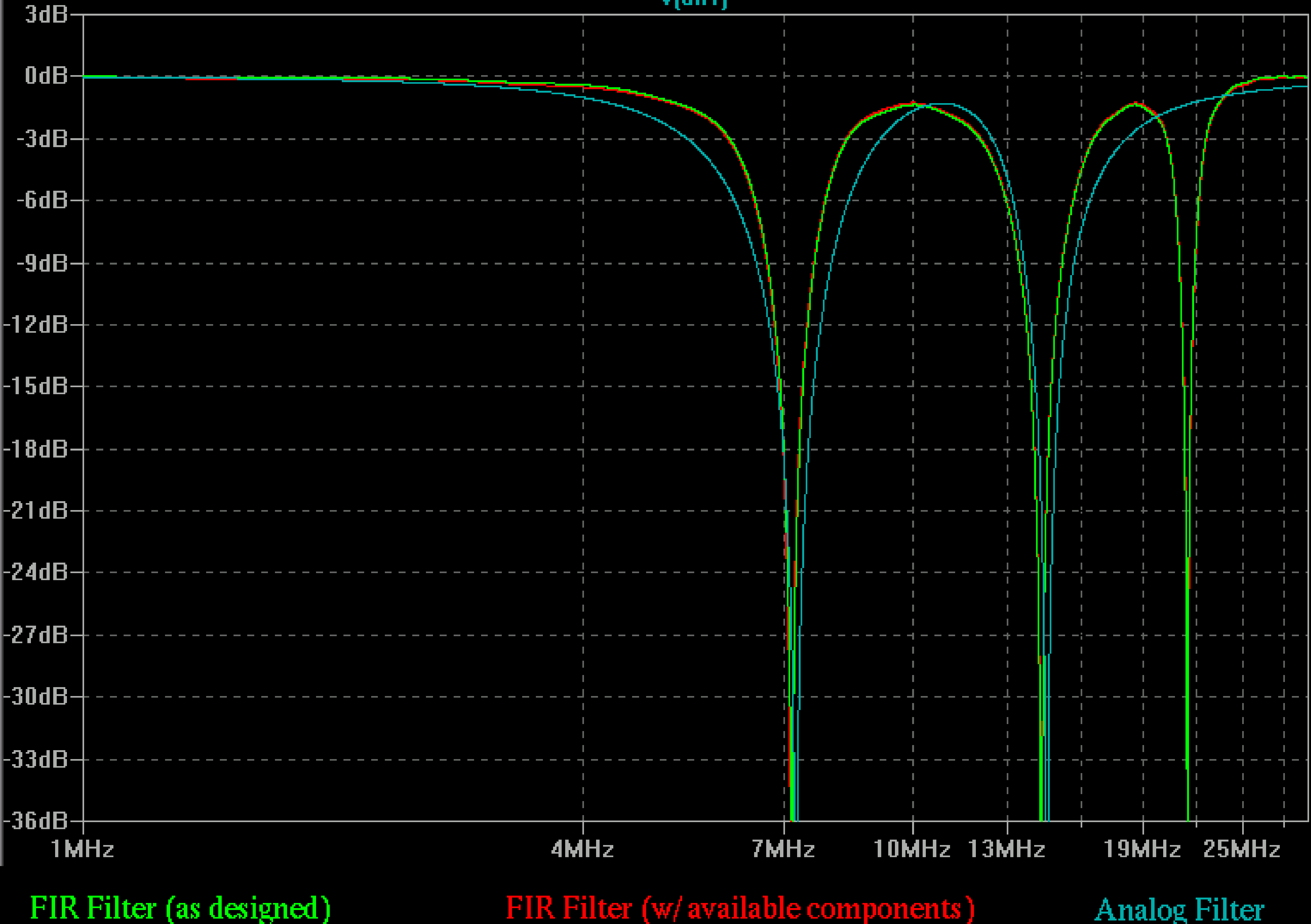
6 Future thoughts

1. Digital Luma filtering
2. Boards for the][and //e
3. The //gs

6.1 Digital Luma filtering

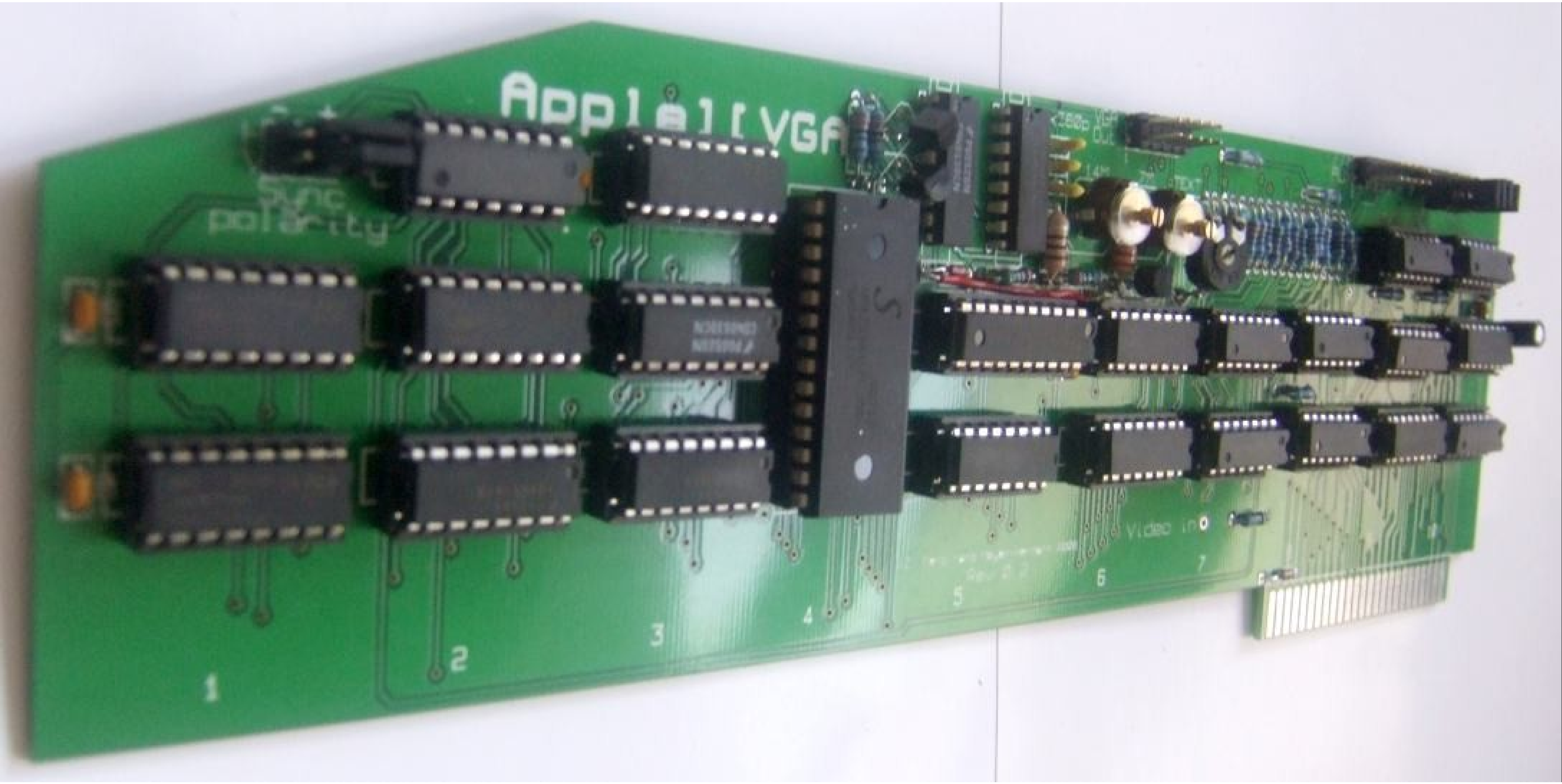
- Analog notch filters have advantages:
  - Simplicity
  - Extremely sharp notches possible
- But the required coils can make a lot of trouble
  - Losses reduce effectiveness
  - Interference
  - Detuning
- Solution: Digital filters!
  - FIR filtering: Sum of shift register outputs


U10
74HC164
A
B
SRCLK
SRCLR
QA
QB
QC
QD
QE
QF
QG
QH
U11
74HC164
U12A
U12B
U12C
U12D
U12E
U12F
74HC04
VCC
28M
470
R39
1.2K
R40
3.0K
R41
1.5K
R42
1.1K
R43
33K
R44
3.45K
R45
7.5K
R46
3.79K
R47
39K
R48
7.9K
R49
29K
R50
12K
R51
72K
R52
22K
R53
91K
R54
33K
R55
VIDEO IN
FILTERED VIDEO OUT


3dB
0dB
-3dB
-6dB
-9dB
-12dB
-15dB
-18dB
-21dB
-24dB
-27dB
-30dB
-33dB
-36dB
1MHz
4MHz
7MHz
10MHz
13MHz
19MHz
25MHz
FIR Filter (as designed)
FIR Filter (w/ available components)
Analog Filter

6.2 Boards for the][and //e

- Combined RAM and VGA boards for the //e's AUX slot (cooperation w/ReactiveMicro)
- Discrete “old-school” boards for the][



6.3 The //gs

- 12-bit RGB color system
  - About 12 times RAM speed required
- 4 different pixel clocks
  - Require pixel clock switching
  - Require pixel clock conversion
- No video expansion connector available
  - Maybe pulling a chip and inserting a connector (like e.g. the][language card)