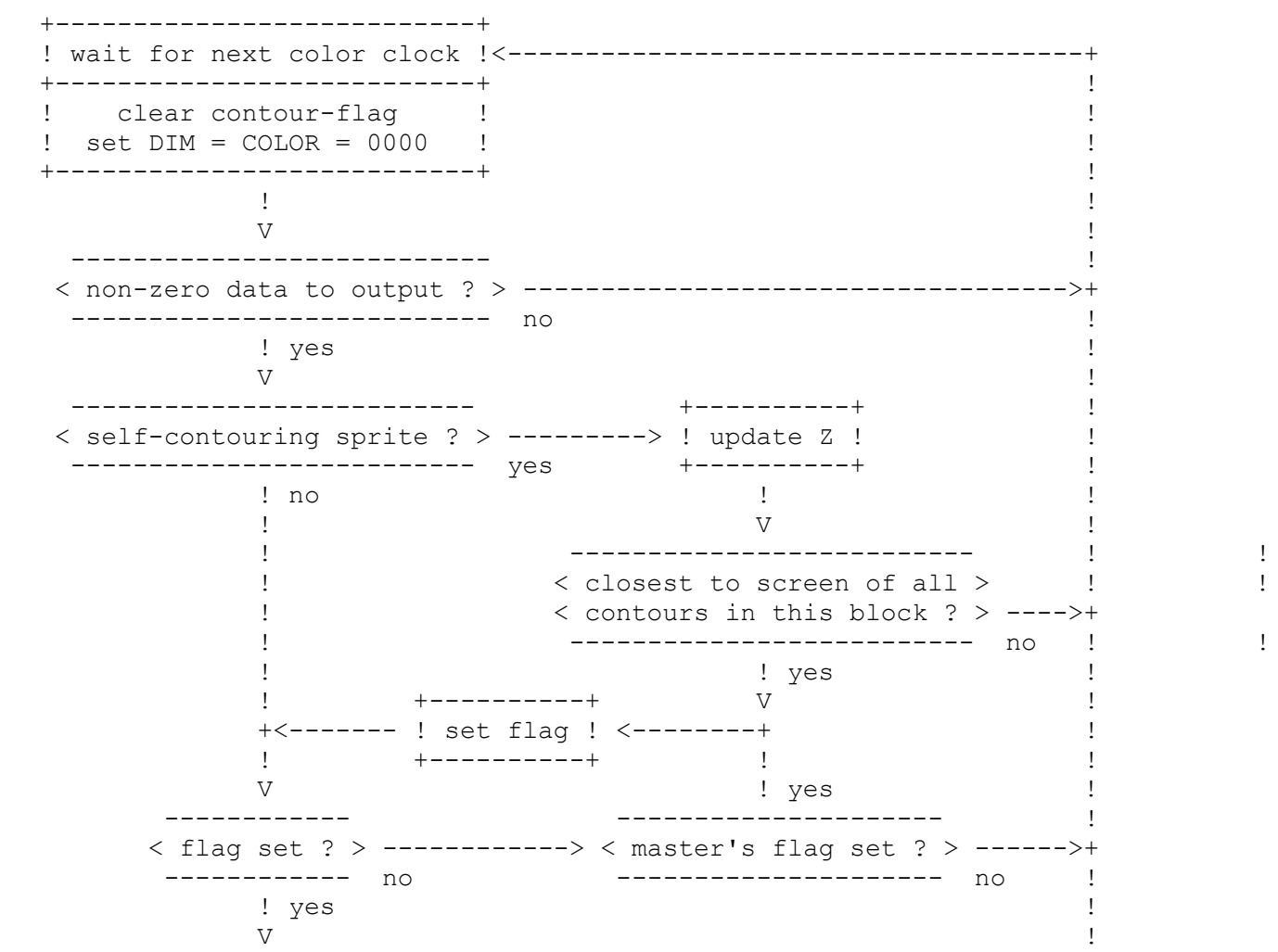
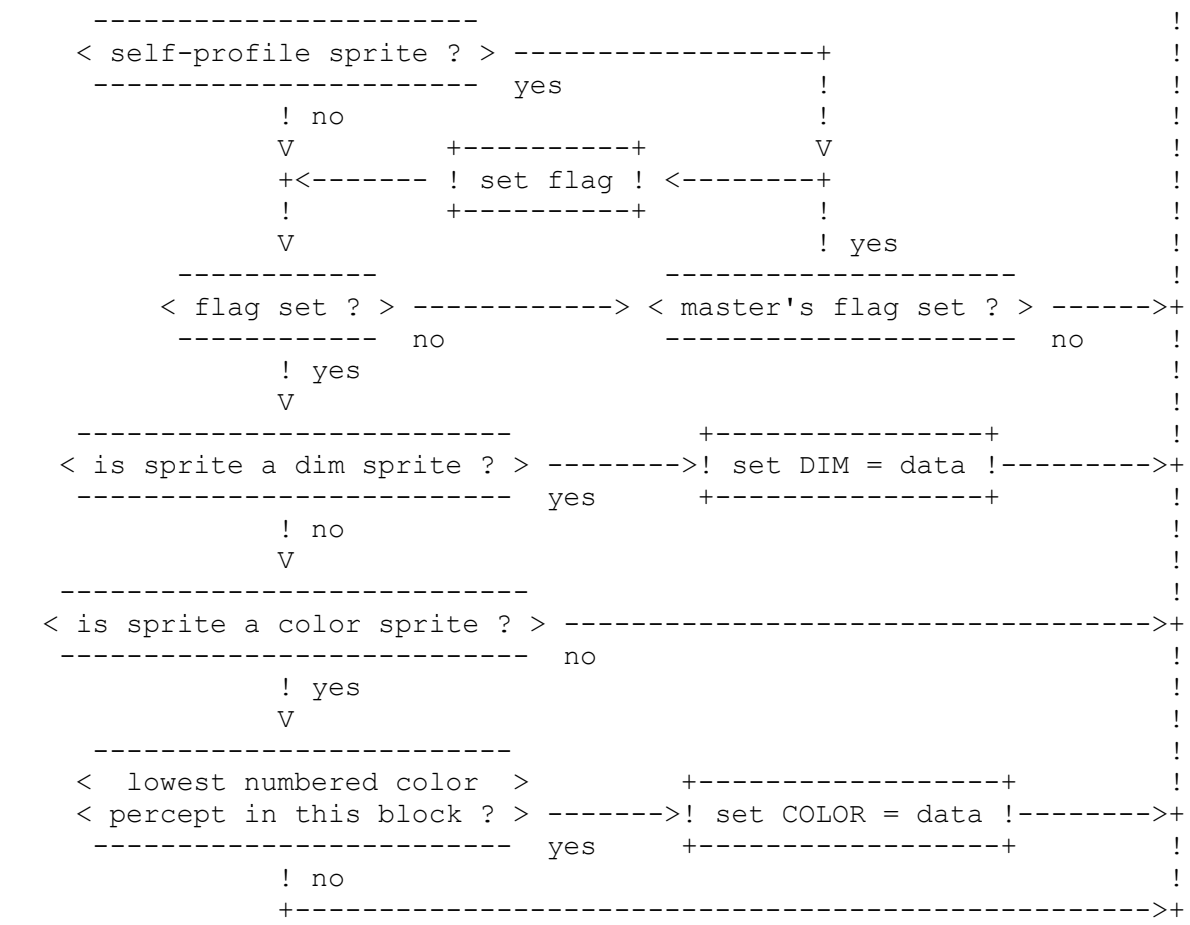
Display prioritization:





ENGINEERING DETAIL - ALGORITHM FOR GENERATING YAW:

YAW	FACT1	FACT2	FACT3	FACT4	dz	dx

0	-	-	-	-	0	1
1	4	5	4	4	4	80
2	2	3	2	3	4	24
3	2	2	2	2	4	16
4	1	2	1	1	4	10
5	1	1	1	1	4	8
6	0	1	0	1	4	6
7	0	1	0	0	4	5
8	0	0	0	0	4	4
9	0	1	0	0	5	4
A	0	1	0	1	6	4
B	1	1	1	1	8	4
C	1	2	1	1	10	4
D	2	2	2	2	16	4
E	2	3	2	3	24	4
F	4	5	4	4	80	4

yaw angle = arctan(dz/dx)

The basic idea here is as follows:

For yaw < 45 degrees, when x has incremented through 2^FACTn pixels, z increments/decrements by 1. This happens for 'n' = 1, 2, 3, & 4 successively for as many repetitions as the length of the data will permit.

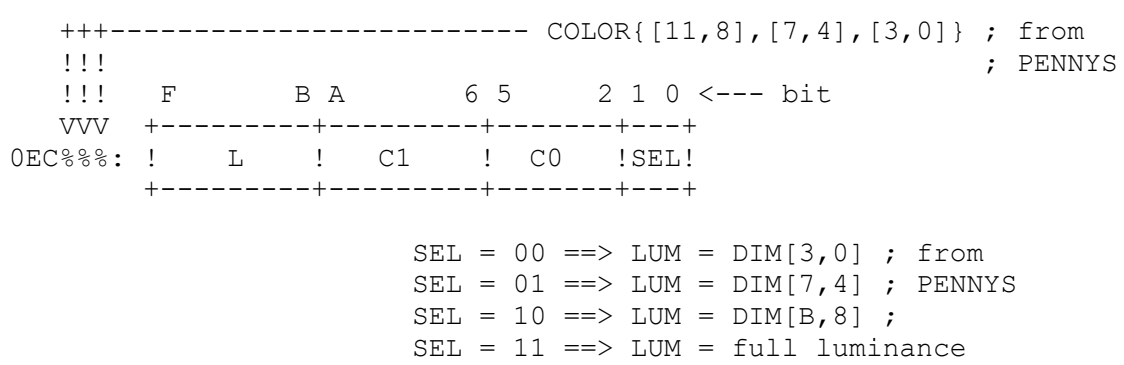
For yaw >= 45 degrees, as x increments by one for each pixel, z increments/decrements by 2^FACTn. This happens for 'n' = 1, 2, 3, & 4 successively for as many repetitions as the length of the data will permit.

During the generation of the present line, the initial z value is incremented/decremented PINC times to arrive at the initial z value for the next line.

The reasoning behind the '2^FACTn' increment/decrement instead of simply adding or subtracting a binary is that a power-of-two up/down counter is smaller than an adder. As can be seen, it

still yields reasonably usable and accurate values of angle if
four intervals are used.

PALETTE MEMORY MAP:



SEL = 00 ==> LUM = DIM[3,0] ; from
SEL = 01 ==> LUM = DIM[7,4] ; PENNYS
SEL = 10 ==> LUM = DIM[B,8] ;
SEL = 11 ==> LUM = full luminance

TWO METHODS OF GENERATING PURE COLORS:

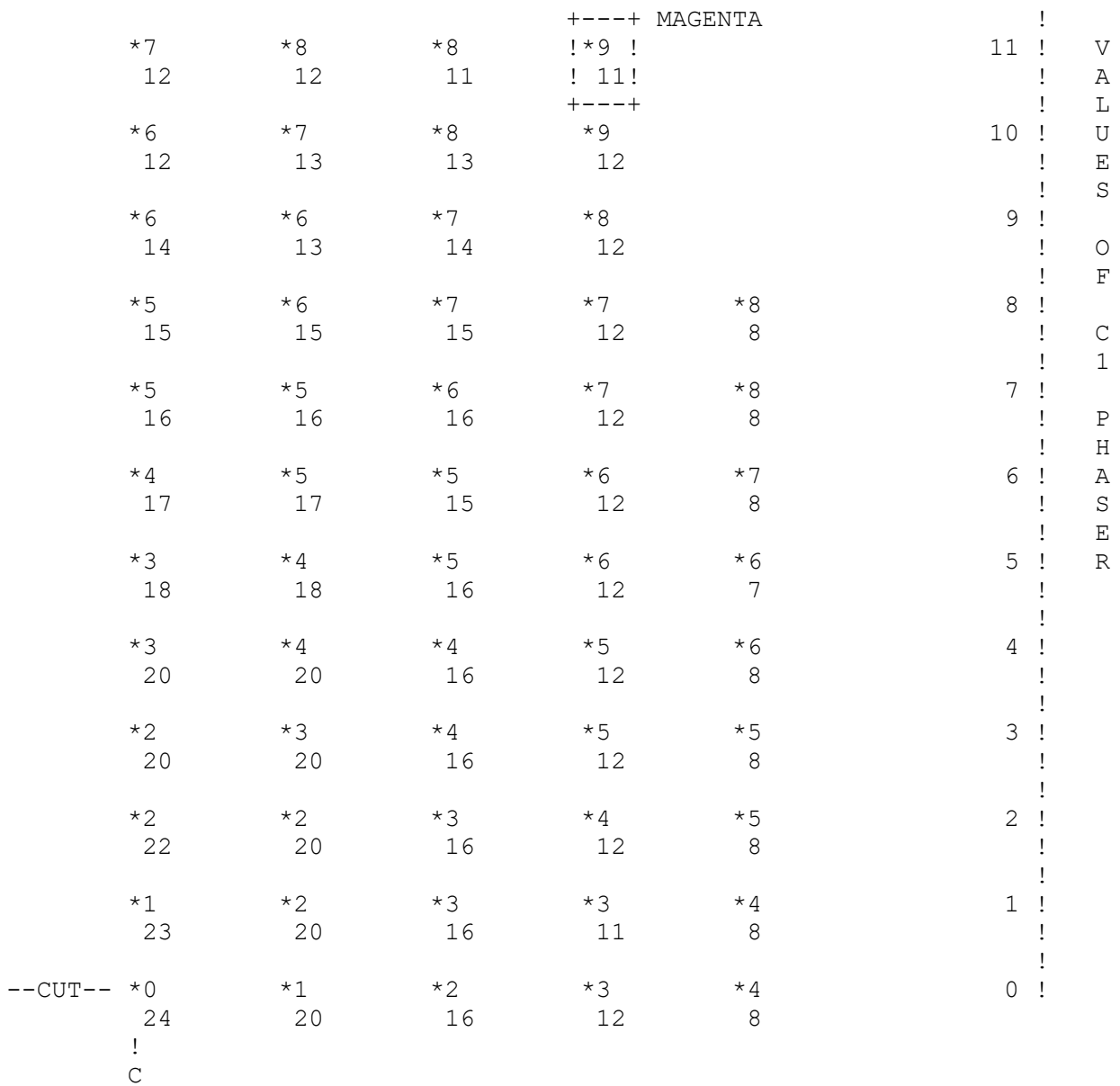
1, SELECT VALUES FOR L, C1 & C0 FROM THE COLOR CHARTS ON THE NEXT
FOUR PAGES

====== OR ======

2, CALCULATE VALUES FOR L, C1 & C0 AS FOLLOWS:

```
        ;; given desired amount of intensity of primary colors

        off = 0 =< Red =< 23.9 = bright red
        off = 0 =< Green =< 23.9 = bright green
        off = 0 =< Blue =< 23.9 = bright blue

        ;; store Luminance
```

```
        L = INTEGER[.30*R+.59*G+.11*B+.49]

        ;; store Chrominance 1

        TEMP = .877*(.70*R-.59*G-.11*B)
        C1 = INTEGER[TEMP-.49]          ; TEMP < 0
        C1 = INTEGER[TEMP+.49]          ; 0 =< TEMP

        ;; & store Chrominance 0

        TEMP = .493*(.89*B-.30*R-.59*G)
        C0 = 5-INTEGER[TEMP-.49]        ; TEMP < 0
        C0 = 5-INTEGER[TEMP/2+.49]      ; 0 =< TEMP



           color point
                V
        LEGEND  *m <--- minimum luminance
        ======   n <--- maximum luminance

                                  COLOR CHART (FIRST QUADRANT)

             V A L U E S   O F   C 0   P H A S E R
        -------------------------------------------------
        5         4         3         2         1        0

                                                           15 !
                                                              !
                                                              !
        *9                                                 14 !
         9                                                    !
                                                              !
        *8        *9        *10                            13 !
         9         9         10                               !
                                                              !
        *7        *8        *9        *10                  12 !
         10        10        10        11  PURE                !
```



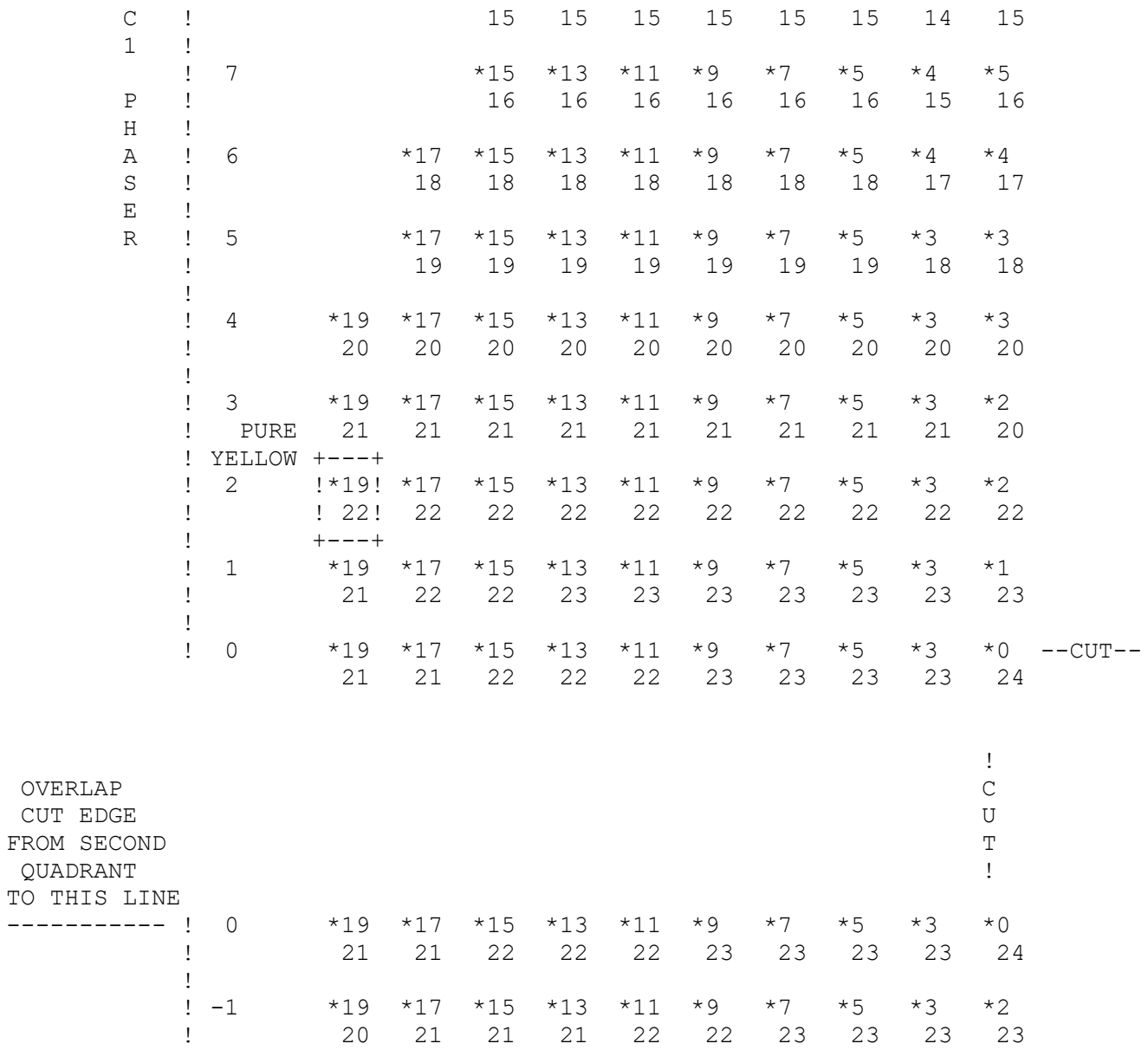
U
T
!

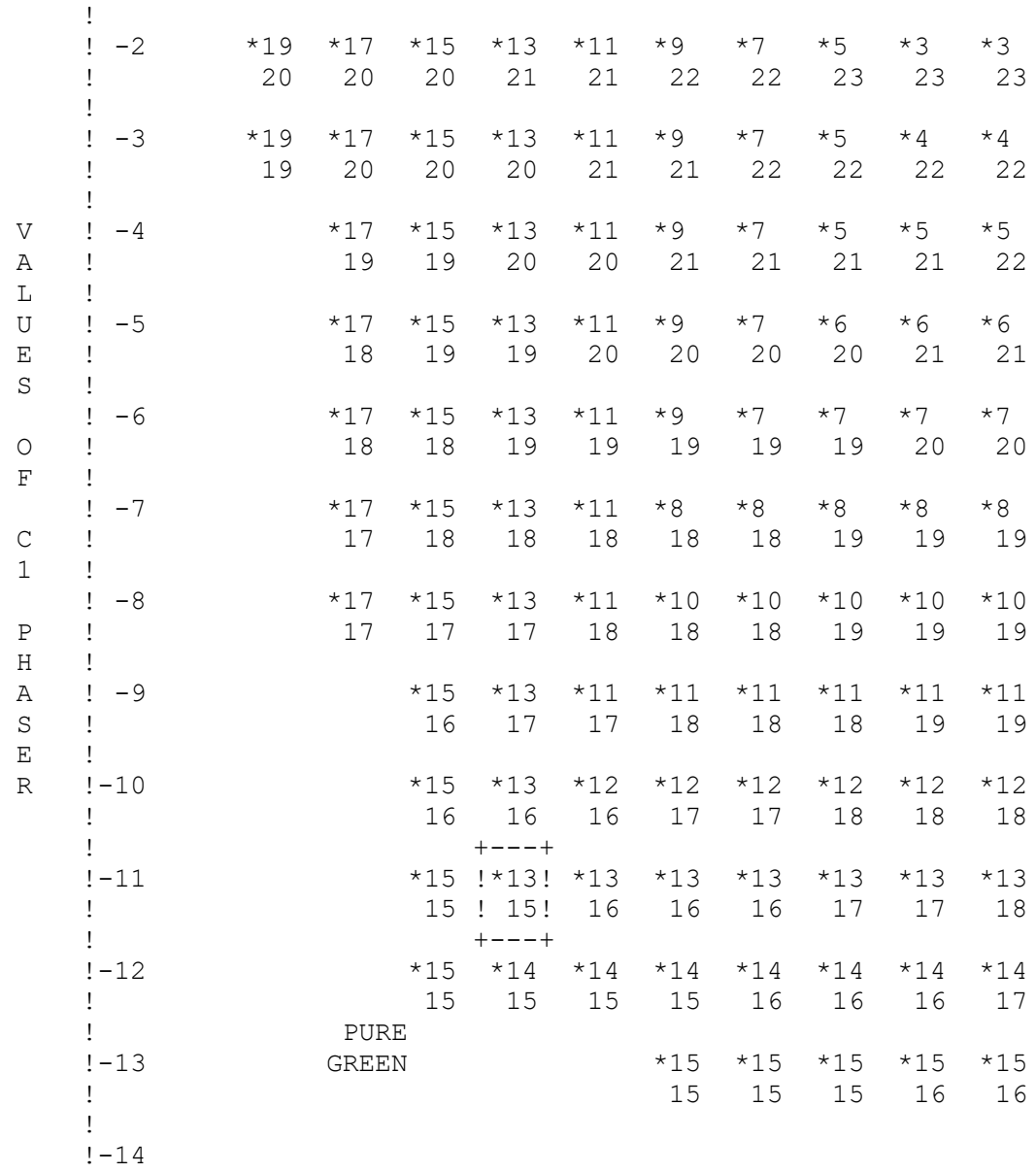
COLOR CHART (SECOND QUADRANT)

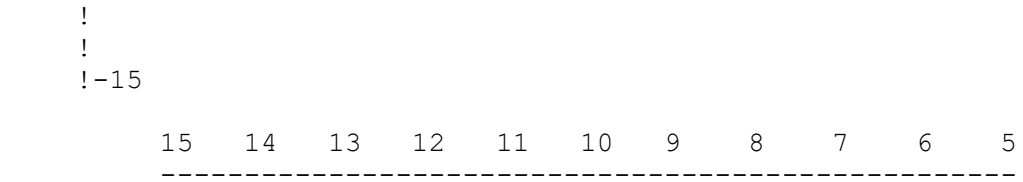
V A L U E S O F C 0 P H A S E R

! OVERLAP
! CUT EDGE
! OF FIRST
! QUADRANT
! TO THIS
! LINE

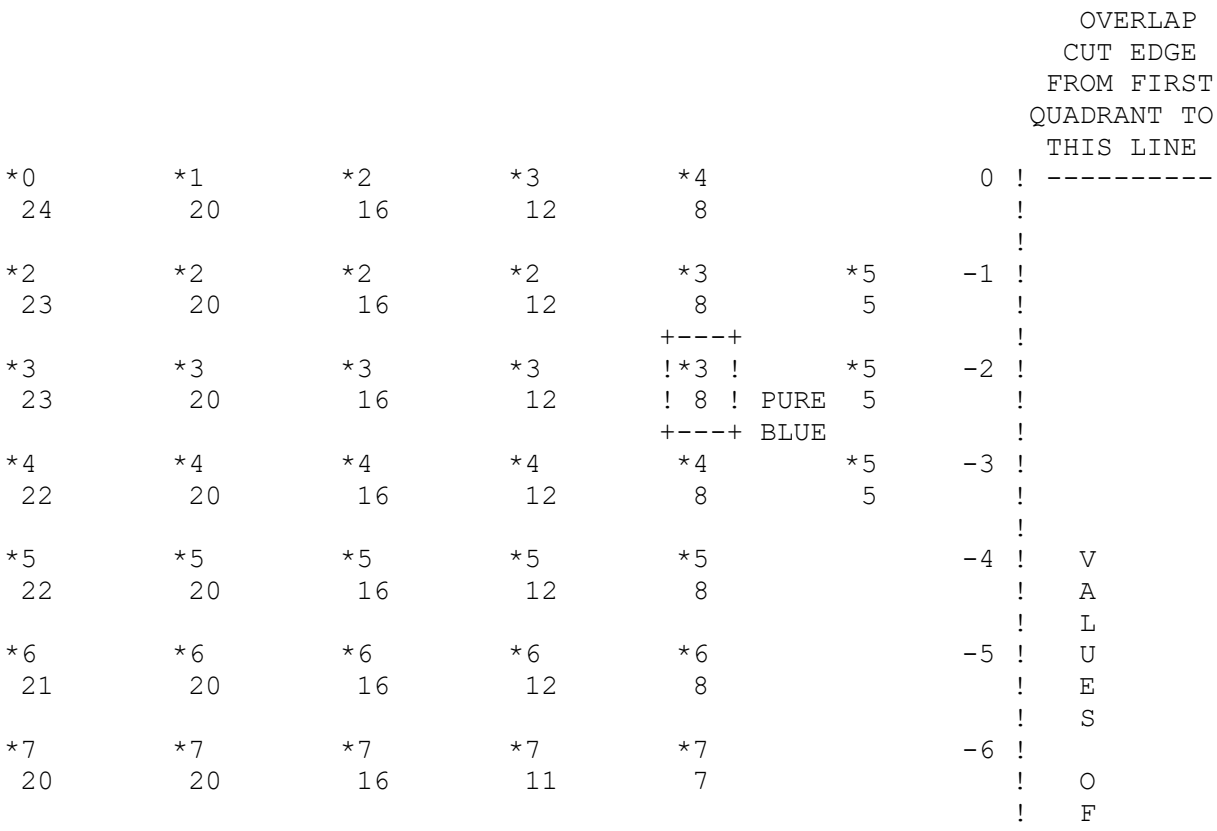
VALUES OF	15	14	13	12	11	10	9	8	7	6	5
	15										
	14						PURE RED	*9 9	*9 9	*9 9	*9 9
	13						*9 10	*8 10	*8 8	*8 9	*8 9
	12					*11 11	*9 11	*7 11	*7 9	*7 9	*7 10
V	11					*11 12	*9 12	*7 12	*6 10	*6 10	*7 12
A	10				*13 13	*11 13	*9 13	*7 13	*6 12	*6 12	*6 12
L	9				*13 14	*11 14	*9 14	*7 14	*5 14	*5 13	*6 14
U	8			*15	*13	*11	*9	*7	*5	*5	*5
E											
S											
O											
F											

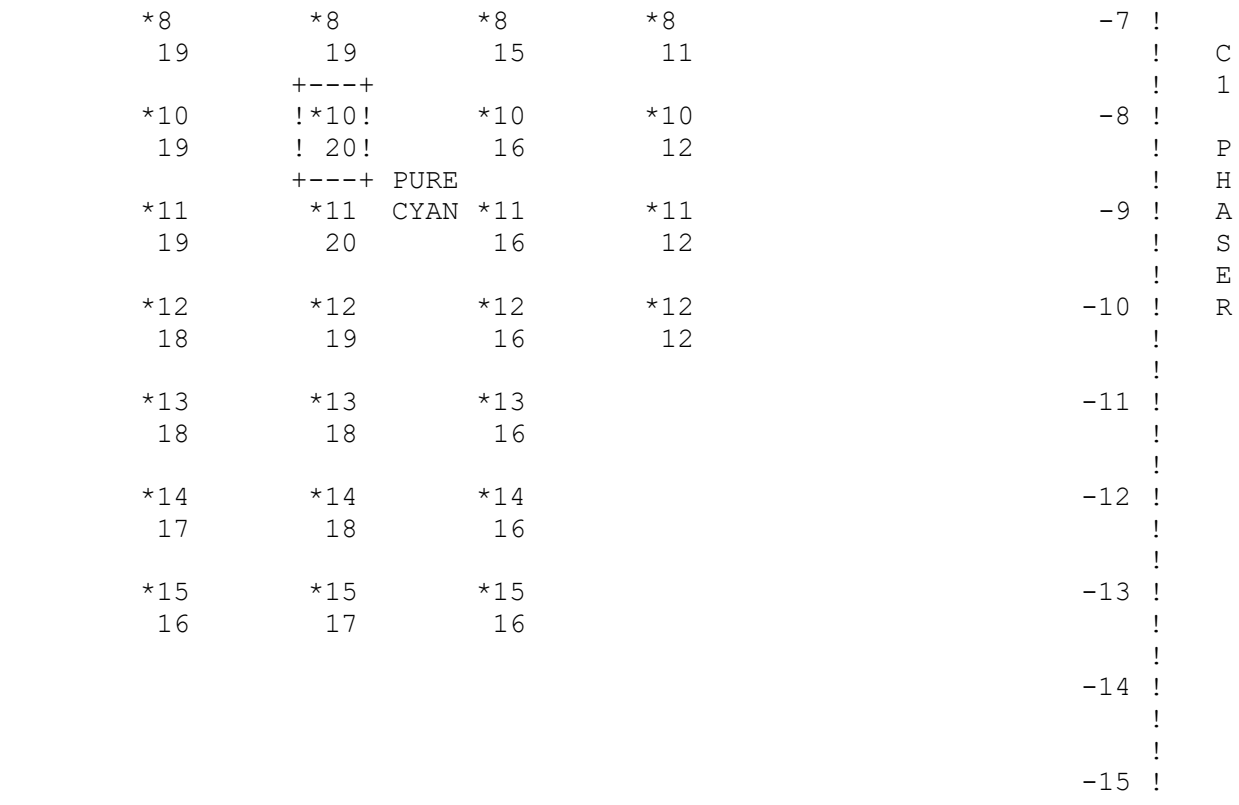






COLOR CHART (THIRD QUADRANT)





DERIVATION OF COLOR CHARTS:

1. I PLOTTED ALL POSSIBLE COMBINATIONS OF PHASERS (IE, 16X31)

2. I THREW OUT ALL COMBINATIONS WHICH EXCEEDED THE LIMITS

        0 =< R,G,B =< 23.9

3. I FOUND THE MAXIMUM SATURATED R, G & B POINTS AND DREW LINES FROM
   THE ORIGIN TO THOSE POINTS TO DIVIDE THE GRAPH INTO THREE REGIONS

        AN R TO G REGION,
        A G TO B REGION &
        A B TO R REGION.

4. I ASSIGNED MINIMUM REQUIRED LUMINANCE FOR EACH SURVIVING POINT

        FOR R TO G REGION

                C0 = .493*(-.30*R-.59*G)
                C1 = .877*(.70*R-.59*G)
                LMIN = .30*R+.59*G = -C0/.493

        FOR G TO B REGION

                C0 = .493*(.89*B-.59*G)
                C1 = .877*(-.11*B-.59*G)
                LMIN = .11*B+.59*G = -C1/.877

        FOR B TO R REGION

                C0 = .493*(-.30*R+.89*B)
                C1 = .877*(.70*R-.11*B)
                LMIN = .30*R+.11*B = .3782*C0+.5798*C1

        IN THE ABOVE CALCULATIONS, 'LMIN' IS THE LUMINANCE WHICH IS
        REQUIRED JUST TO SUPPORT THE CHROMINANCE VECTORS WITHOUT

ANY ADDED LUMINANCE (IE, FULLY SATURATED CHROMINANCE).

5. I READJUSTED THE MINIMUM LUMINANCE UPWARD FOR POINTS WHICH WERE OVERSATURATED (IE, THE SIGNAL DIPPED BELOW -20IRE)

```
   OVERSATURATED SIGNAL < -20IRE
LMIN(+/-)C0,LMIN(+/-)C1 < -20IRE = -5
```

6. I THREW OUT ALL POINTS WHICH WERE OVERMODUATED (IE, THE SIGNAL OVERSHOT 120IRE) WITH EVEN THE MINIMUM LUMINANCE

```
     120IRE < OVERMODUATED SIGNAL
29 = 120IRE < LMIN(+/-)C0,LMIN(+/-)C1
```

DERIVATION OF COLOR CHARTS (CONTINUED):

7. I FOUND THE AMOUNT OF PRIMARY COLORS IN EACH POINT AT MINIMUM LUMINANCE

FOR R TO G REGION

```
C0 = .493*(-.30*R-.59*G)
C1 = .877*(.70*R-.59*G)
R = -C0/.493+C1/.877
G = -2.4066*C0-.5798*C1
```

FOR G TO B REGION

```
C0 = .493*(.89*B-.59*G)
C1 = .877*(-.11*B-.59*G)
B = C0/.493-C1/.877
G = -.3782*C0-1.7200*C1
```

FOR B TO R REGION

C0 = .493*(-.30*R+.89*B)
C1 = .877*(.70*R-.11*B)
R = .3782*C0+1.7200*C1
B = 2.4066*C0+.5798*C1

8. I SOLVED FOR MAXIMUM LUMINANCE FOR EACH POINT BY ADDING DELTA TO ALL THREE COLORS UP TO THE MAXIMUM OF 24 FOR ANY ONE COLOR

FOR R TO G REGION

DELTA = SMALLEST{24-R,24-G}
LMAX = LMIN+DELTA

FOR G TO B REGION

DELTA = SMALLEST{24-G,24-B}
LMAX = LMIN+DELTA

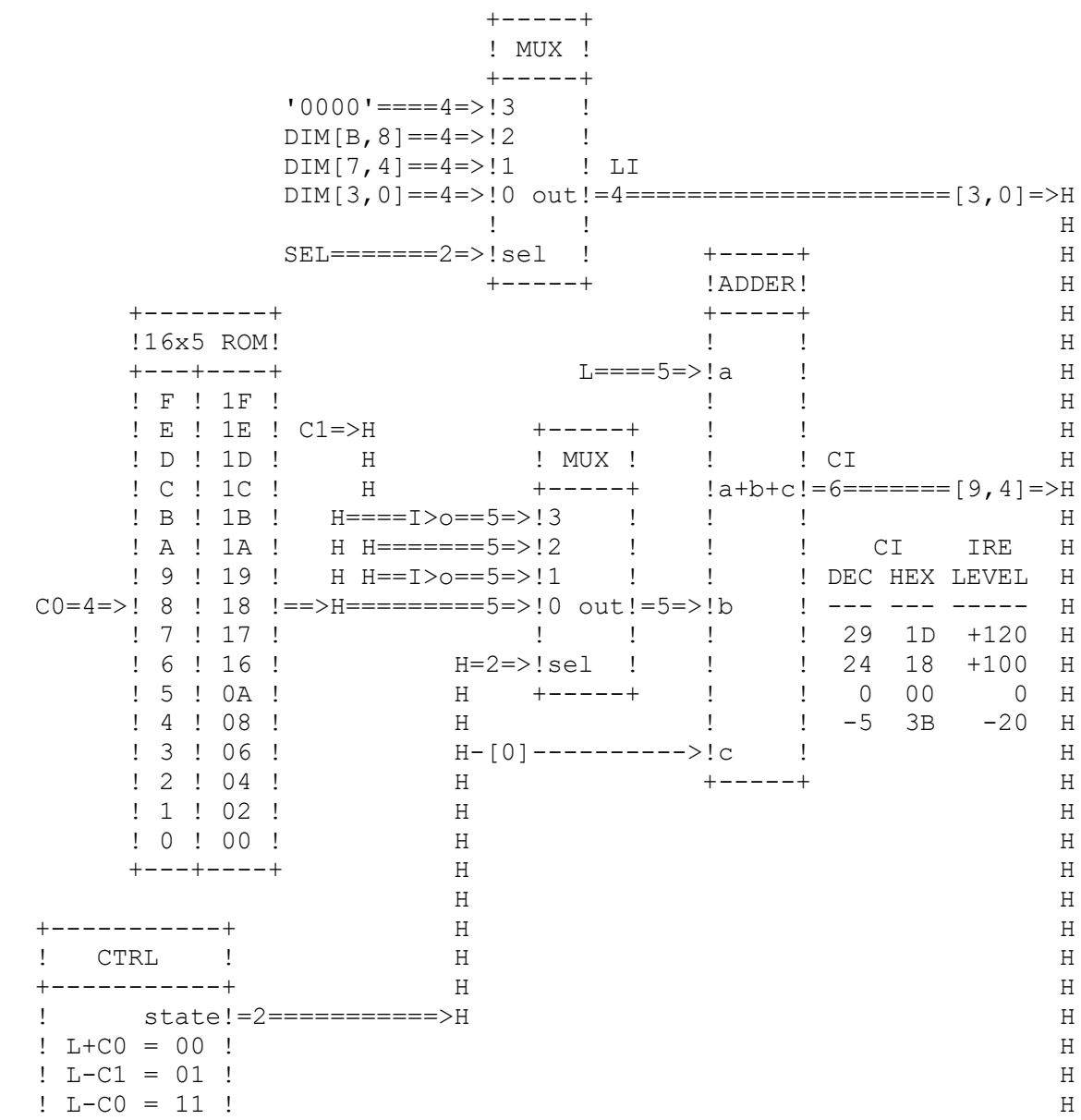
FOR B TO R REGION
DELTA = SMALLEST{24-R,24-B}
LMAX = LMIN+DELTA

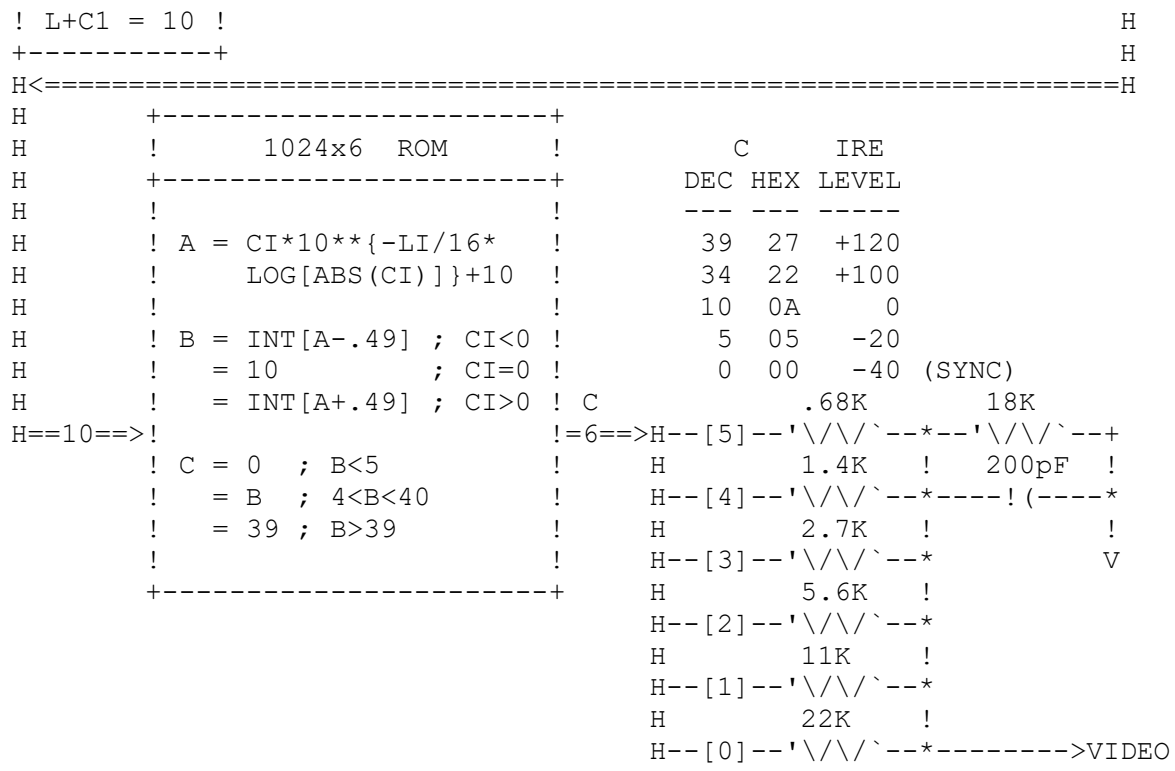
9. I READJUSTED THE MAXIMUM LUMINANCE DOWNWARD FOR POINTS WHICH WERE OVERMODULATED (IE, THE SIGNAL OVERSHOT 120IRE)

120IRE < OVERMODUATED SIGNAL
29 = 120IRE < LMAX(+/-)C0,LMAX(+/-)C1

10. I ENTERED LMIN AND LMAX FOR EACH POINT ON THE CHARTS BESIDE EACH POINT.

GENERATION OF VIDEO OUTPUT





VIDEO OUTPUT ROM PROGRAMMING TABLE

bits 9,4 in HEX (DEC)	bits 3,0 in HEX 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00 (0)	06	05	05	04	04	03	03	03	02	02	02	02	01	01	01	01
01 (1)	07	06	05	05					02	02	02	02	01	01	01	01
02 (2)	08	07	06	05					03	02	02	02	02	01	01	01

03 (3)	09 08 07 06	TO BE REVISED	03 02 02 02	02 01 01 01
04 (4)	0A 09 07 06		03 03 02 02	02 01 01 01
05 (5)	0B 09 08 07		03 03 02 02	02 01 01 01
06 (6)	0C 0A 09 07	06 05 05 04	03 03 02 02	02 01 01 01
07 (7)	0D 0B 09 08	07 06 05 04	03 03 02 02	02 02 01 01
08 (8)	0E 0C 0A 08	07 06 05 04	04 03 03 02	02 02 01 01
09 (9)	0F 0D 0B 09	07 06 05 04	04 03 03 02	02 02 01 01
0A (10)	10 0D 0B 09	08 07 06 05	04 03 03 02	02 02 01 01
0B (11)	11 0E 0C 0A	08 07 06 05	04 03 03 02	02 02 01 01
0C (12)	12 0F 0C 0A	09 07 06 05	04 03 03 02	02 02 01 01
0D (13)	13 10 0D 0B	09 07 06 05	04 04 03 02	02 02 01 01
0E (14)	14 11 0D 0B	09 08 06 05	04 04 03 02	02 02 01 01
0F (15)	15 11 0E 0C	0A 08 07 05	04 04 03 02	02 02 01 01
10 (16)	16 12 0F 0C	0A 08 07 06	05 04 03 03	02 02 01 01
11 (17)	17 13 0F 0D	0A 09 07 06	05 04 03 03	02 02 01 01
12 (18)	18 14 10 0D	0B 09 07 06	05 04 03 03	02 02 01 01
13 (19)	19 14 11 0E	0B 09 07 06	05 04 03 03	02 02 01 01
14 (20)	1A 15 11 0E	0B 09 08 06	05 04 03 03	02 02 01 01
15 (21)	1B 16 12 0E	0C 0A 08 06	05 04 03 03	02 02 01 01
16 (22)	1C 17 12 0F	0C 0A 08 06	05 04 03 03	02 02 01 01
17 (23)	1D 17 13 0F	0C 0A 08 07	05 04 03 03	02 02 01 01
18 (24)	1E 18 14 10	0D 0A 08 07	05 04 03 03	02 02 01 01
19 (25)	1F 19 14 10	0D 0A 08 07	05 04 04 03	02 02 01 01
1A (26)	20 1A 15 11	0D 0B 09 07	06 04 04 03	02 02 01 01
1B (27)	21 1A 15 11	0D 0B 09 07	06 05 04 03	02 02 01 01
1C (28)	22 1B 16 11	0E 0B 09 07	06 05 04 03	02 02 02 01
1D (29)	23 1C 16 12	0E 0B 09 07	06 05 04 03	02 02 02 01
1E (30)	24 1D 17 12	0F 0C 09 07	06 05 04 03	02 02 02 01
1F (31)	25 1D 17 13	0F 0C 09 08	06 05 04 03	02 02 02 01

VIDEO OUTPUT ROM PROGRAMMING TABLE (CONTINUED)

bits 9,4 in HEX (DEC)	bits 3,0 in HEX 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20 (32)	26	1E	18	13	0F	0C	0A	08	06	05	04	03	02	02	02	01
21 (33)	27	1F	19	14	0F	0C	0A	08	06	05	04	03	02	02	02	01
22 (34)	28	20	19	14	10	0D	0A	08	06	05	04	03	02	02	02	01
23 (35)	29	20	1A	14	10	0D	0A	08	06	05	04	03	02	02	02	01
24 (36)	2A	21	1A	15	10	0D	0A	08	06	05	04	03	02	02	02	01
25 (37)	2B	22	1B	15	11	0D	0A	08	06	05	04	03	03	02	02	01
26 (38)	2B	23	1B	16	11	0D	0B	08	07	05	04	03	03	02	02	01
27 (39)	2B	23	1C	16	11	0E	0B	08	07	05	04	03	03	02	02	01
28 (40)	2B	24	1C	16	12	0E	0B	09	07	05	04	03	03	02	02	01
29 (41)	2B	25	1D	17	12	0E	0B	09	07	05	04	03	03	02	02	01
2A (42)	2B	26	1D	17	12	0E	0B	09	07	05	04	03	03	02	02	01
2B (43)	2B	26	1E	18	12	0E	0B	09	07	05	04	03	03	02	02	01
2C (44)	2B	27	1F	18	13	0F	0B	09	07	05	04	03	03	02	02	01
2D (45)	2B	28	1F	18	13	0F	0C	09	07	05	04	03	03	02	02	01
2E (46)	2B	29	20	19	13	0F	0C	09	07	06	04	03	03	02	02	01
2F	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
30	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
31 (-15)	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
32 (-14)	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
33 (-13)	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
34 (-12)	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
35 (-11)	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
36 (-10)	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
37 (-9)	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

```
38 (-8)   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00
39 (-7)   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00
3A (-6)   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00
3B (-5)   01 01 01 01   01 01 01 01   01 01 01 01   01 01 01 01

3C (-4)   02 02 02 02   02 01 01 01   01 01 01 01   01 01 01 01
3D (-3)   03 03 03 02   02 02 02 02   01 01 01 01   01 01 01 01
3E (-2)   04 04 03 03   03 02 02 02   02 02 01 01   01 01 01 01
3F (-1)   05 04 04 04   03 03 03 02   02 02 02 01   01 01 01 01
```

PENNY PINOUTS

#pins	Input/Output (srce/dest)	symbol	description
1	I (pwr sply)	VDD	+5v
1	I (pwr sply)	VSS	ground
1	I (VIVIAN)	CLK	system CLocK (7.16 MHz)
1	I (VIVIAN)	SYNC	horizontal/vertical SYNC
16	I/O (memory)	MDB	Memory Data Bus
8	O (memory)	MAB	Memory Address Bus
1	O (memory)	MCAS	Memory Column Address Strobe
1	O (memory)	MRAS	Memory Row Address Strobe
1	O (memory)	MRES	Memory Refresh Enable Strobe
1	O (memory)	MWES	Memory Write Enable Strobe
4	O (palette)	COLOR	COLOR
4	O (HEATHER)	DIM	DIMness factor
40			