Twenty Dollars

Cromemco

SDI

Graphics Interface

Instruction Manual

Cromemco

SDI GRAPHICS INTERFACE

INSTRUCTION MANUAL

CROMEMCO, Inc.
280 Bernardo Avenue
Mountain View, CA 94043

Part no. 023-2001

April 1980

This manual was produced on a Cromemco System Three computer using the Cromemco Screen Editor. The edited text was formatted using the Cromemco Word Processing System Formatter. Final camera-ready copy was printed on a Cromemco 3355A printer.

Table of Contents

1. THEORY OF OPERATION

1.0 INTRODUCTION

The information contained in this manual provides a detailed description of the Cromemco SDI graphics interface hardware. Please refer to the Cromemco SDI software manual if you will be using Cromemco SDI software exclusively. Most of the SDI operation described below is handled automatically by SDI software.

1.1 THE VERSATILE SDI

The Cromemco SDI is the most versatile video interface in the microcomputer industry today. With its high point resolution, Color Map selection, dual page windowing function, automatic area fill mode, and NTSC broadcast compatibility, the most demanding requirements for a video interface can be met. The SDI is designed to meet the challenges of professional and industrial environments where uncompromising performance, reliability, and continued compatibility are essential.

The SDI consists of two circuit boards which plug directly into the S-100 bus of any Cromemco microcomputer system. It acts as an interface between the computer and an RGB color monitor. The SDI performs its function by turning digital information which is stored in the computer's memory into a high resolution color image possessing strikingly pure and undistorted features. When used in conjunction with the Cromemco model RGB-19 high resolution color monitor and Cromemco Two Port memory the resulting picture is of a quality unparalleled in the industry.

In addition, the SDI offers a wide selection of operational modes which allow the selection of the size, location, resolution, and type of DMA suited to a particular application. All of this is completely controlled by software. Thus, one program can employ one operational mode and a later program can use an alternate mode. This manual contains information on utilizing the many options offered by the SDI. However, for more automatic handling of the SDI's features in FORTRAN, RATFOR,

or STRUCTURED BASIC, the Cromemco SDI software package is available.

As an indication of the power and flexibility of the SDI, consider the following applications:

* Where extremely high point resolution (482V x 756H) is required and 2 colors are sufficient, the **bit-mapped** mode may be selected. This is useful for:

  1. Multi-dimensional mathematical functions and graphs
  2. Engineering design
  3. Aircraft simulation
  4. Text generation, editing, and storage
  5. Curve plotting

* Where normal resolution (241V x 378H) is sufficient and wide color choice is important (any 16 from a 4096 color menu), the **nybble-mapped** mode is available. Just a few applications of this mode are:

  1. Picture information for inventory data
  2. Industrial control systems
  3. Medical monitoring
  4. Image processing
  5. Full color financial reports containing curves, bar charts, graphs, and digitized video pictures of products
  6. LSI circuit design
  7. Television studio special effects

* Where both of the above modes apply in a single application, portions of the same screen can be run in separate modes. Some applications would be:

  1. High resolution text in an inventory description of an item with a full color digitized picture included on a portion of the screen
  2. Video art
  3. Fine line smooth curve drawings adjacent to full color images

* Because of the special SDI Color Map technique, the 16 colors can be instantly changed creating simulated motion. This is useful in video animation.

* The SDI allows 2 pages of picture information to be stored simultaneously and windows to be opened from one onto the other. This technique is useful in:

  1. Process control
  2. Educational programs which open and close windows onto clues and answers
  3. Product advertising displays
  4. Computer aided design
  5. Animation

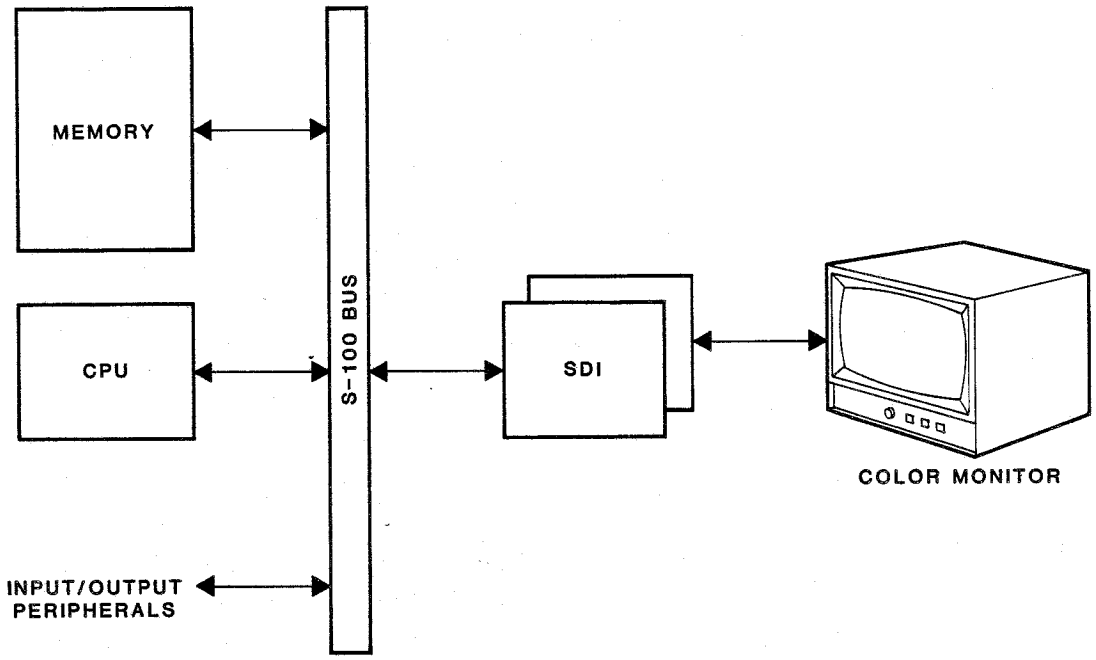
* With the SDI auto-fill mode a solid colored image can be created in the time it takes to draw the outline of the shape.

* Finally, by adding extra SDI interfaces to your system, the aforementioned color and resolution choices can be expanded. This is possible because of the flexibility afforded by separate red, green, and blue outputs.

1.2 BASIC SYSTEM LAYOUT

Any Cromemco microcomputer can be made to drive a full color RGB monitor through use of the 2 circuit boards which make up the SDI. By plugging the 2 SDI boards into the S-100 motherboard and connecting board #2 to an RGB color monitor you can attain the minimum system necessary to achieve high resolution color graphics. This system configuration is shown in the following figure.

As described in section 1.4 the SDI uses direct memory access (DMA) to take the picture information directly from the computer memory. The computer can be simultaneously engaged in the execution of a user program. In the basic system the SDI and the main CPU must share the data bus. This will cause an increase in the execution time of a user program by decreasing the CPU efficiency from 100% down to 65% at best, and only 8% in the most severe case. The actual CPU slowdown experienced due to bus sharing will be determined by the amount of memory used to store the TV image and the type of instructions sent to the SDI. These controls are discussed in section 2.4. This restrictive slowdown can be overcome through use of the Cromemco Two Port RAM boards described in the next section.



The Basic SDI System

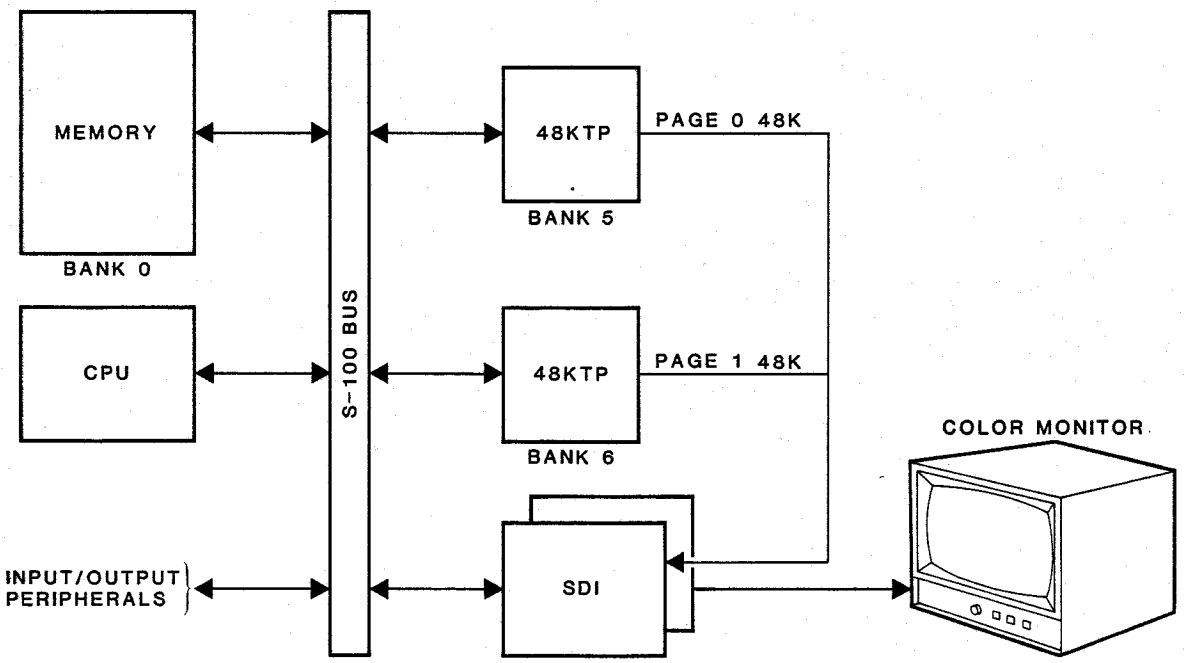
1.3 INCORPORATING THE 48KTP

If the CPU is to perform at 100% efficiency during SDI operation or if additional RAM is needed for use as a dedicated Frame Buffer, one or more 48KTP memory boards may be added to the system. These are Two Port memory RAMs which were specifically developed by Cromemco for use with the SDI to eliminate CPU slowdown during SDI operation. A typical system configuration employing two 48KTP boards is shown in the following figure.

This system will allow two 48K pictures to be buffered in dedicated memory simultaneously without taxing the main computer memory or impairing CPU efficiency. The Two Port boards also make it possible to write into one Frame Buffer while displaying another or to display parts of each buffer on different portions of the screen simultaneously. Other features such as independently controlled scrolling, animation, and stylized fade-outs can also be achieved with this system.

The Two Port RAMs are installed by first inserting

them into the S-100 bus. Then use the 50 conductor ribbon cable provided to connect connector J-2 of the SDI DMA board to the Two Port Memory. If more than one Two Port Memory board is used, they are **daisy-chained** together by means of additional connectors on this same 50 conductor ribbon cable. When using the 48KTP Two Port Memory, an additional 25 conductor cable is used to daisy-chain these memory boards to connector J-4 of the SDI DMA board. No connection is required to connector J-4 when using a 16KTP Two Port Memory.



An SDI System with Two 48KTP Boards

1.4 THE SDI FRAME BUFFER

Regardless of the system configuration the SDI continuously polls a region of memory called the Frame Buffer and converts this digital information

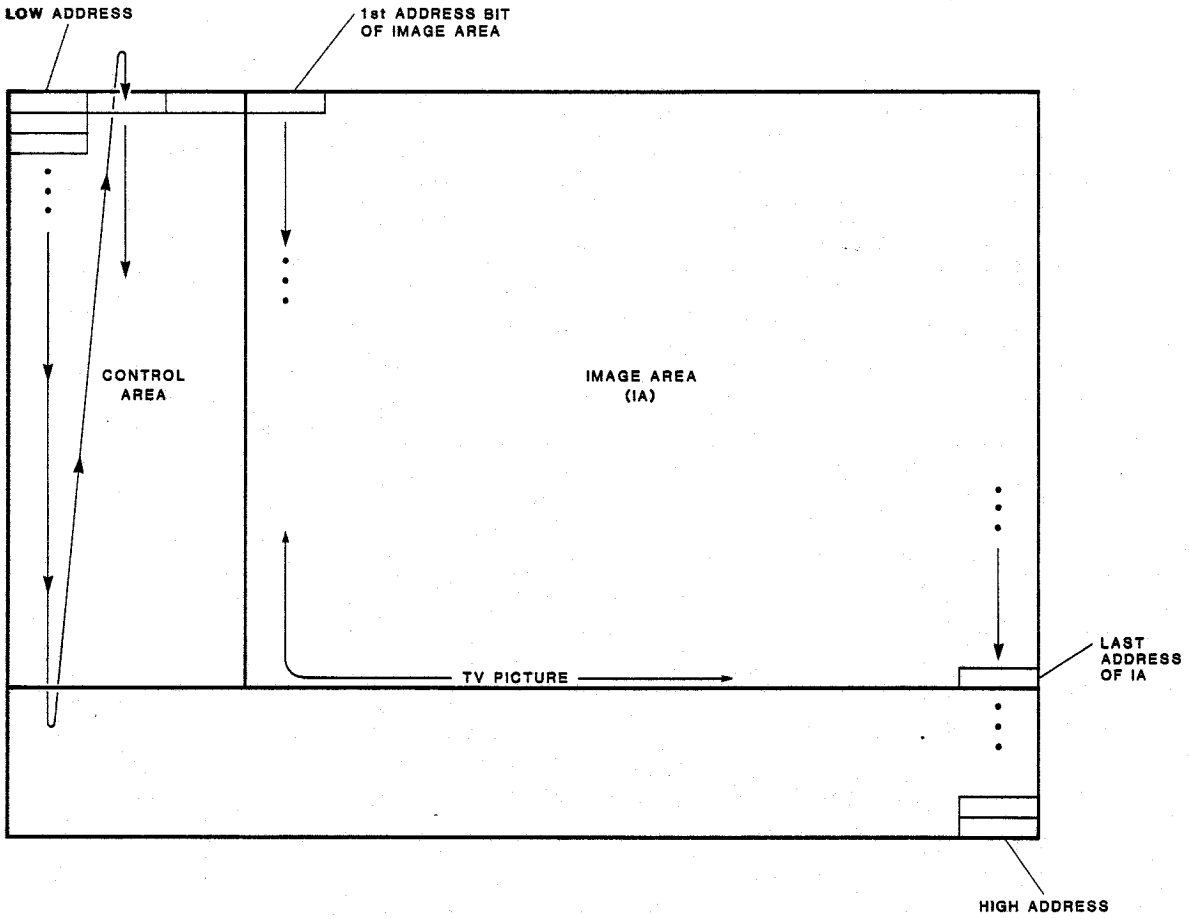
into a standard RGB video signal. The location, size, and format of the Frame Buffer depend on how the various SDI switches are set. (Most of the switches are under software control.) The basic arrangement of the Frame Buffer is, however, the same in all cases.

IMAGE AREA

The major portion of the Frame Buffer consists of an Image Area. Each bit or nybble of the Image Area corresponds to one pixel on the CRT screen. The value of the bit or nybble determines the color and intensity of the pixel. The location of the value (bit or nybble within the Image Area) which will determine the color of a given pixel is not fixed. The determination of this value-color correspondence is called **setting the Color Map** and is discussed in detail in Chapter 2.

CONTROL AREA

The Control Area is a smaller portion of the Frame Buffer which allows windows to be formed and more than one resolution to be displayed. The bits in the Control Area do not correspond in a one-to-one fashion with pixels on the screen, nor do they determine color/intensity in the picture. Rather than being **seen**, these bits can be used to **cut windows** in selected parts of a foreground picture, giving the viewer a glance at a background picture (see section 2.7 on Page Interleaving). These control bits also serve a second function by allowing different resolutions to be displayed on different parts of the screen simultaneously (see section 2.6 on Dual Resolution).

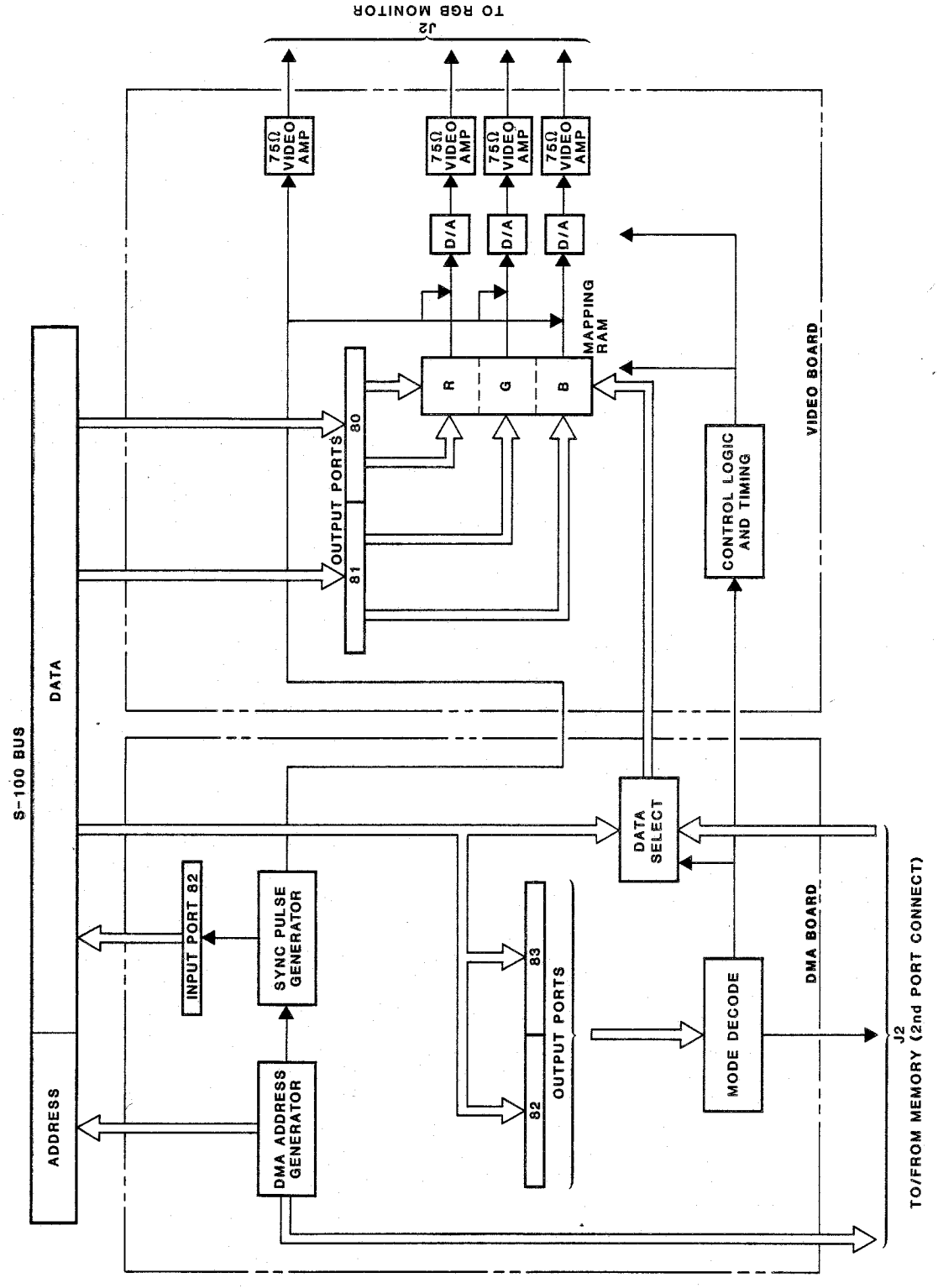


The Frame Buffer

1.5 THE SDI BOARDS

There are 2 circuit boards which make up the SDI. Board 1, the DMA board, is responsible for memory access, mode control, and sync pulse generation. Board 2, the video board, interprets the digital information which the DMA board supplies and converts it to an analog signal for video output.

The 4 computer output ports that control the SDI can be divided into color data ports (80H and 81H) and control information ports (82H and 83H). The single input port (82H) supplies the CPU with complete timing information necessary for software/video synchronization. A complete understanding of the operation of these I/O ports is important for anyone designing custom software for the SDI.



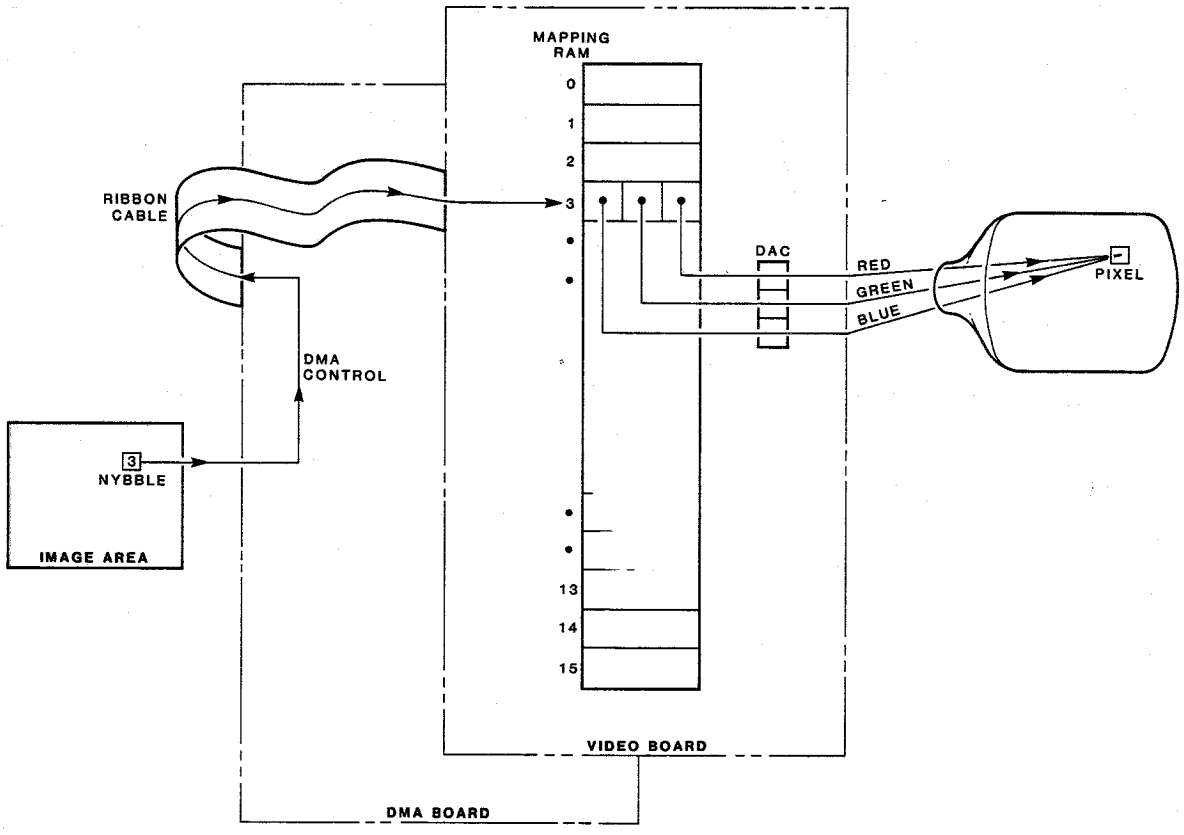
Block Diagram of the SDI Boards

1.6 THE COLOR MAPPING RAM

The color flexibility of the SDI is attributable to three 16X4 RAM chips which are located on the video board and collectively called the Mapping RAM. The Mapping RAM is used as sixteen 12-bit locations. The locations represent the color codes 0 to 15 (FH). The contents of each location is the color associated with that code.

Color mapping works as follows. The DMA calls up an address in the Image Area of the Frame Buffer. A nybble (or bit) is then fetched from that address. The value of that nybble (bit) is used to address one of the 16 locations in the Mapping RAM. Finally, the contents of this (Mapping RAM) location are read and converted to three analog signals which in turn drive the Red, Green, and Blue guns of a color monitor display.

Since a 12 bit word can drive the RGB guns in 4096 different ways, it is up to the programmer to load the Mapping RAM with the 16 colors that may be displayed in any given frame of the SDI output. This process is called **setting the Color Map.**



Operation of the Mapping RAM

2. PROGRAMMER'S GUIDE

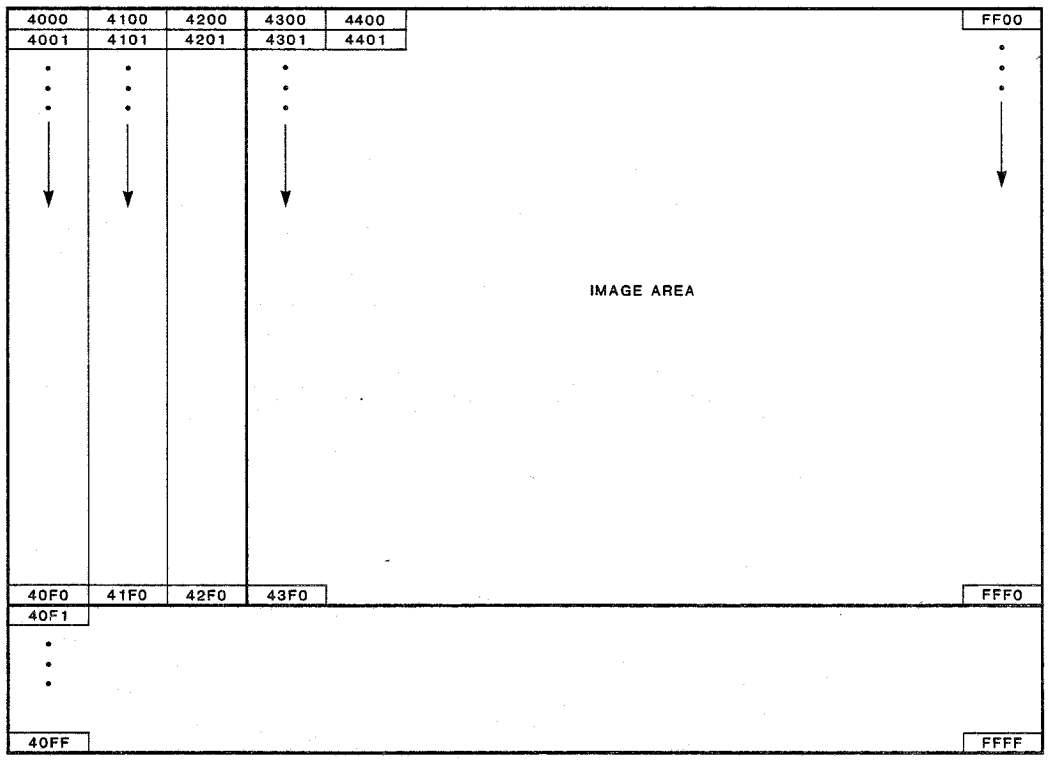
2.1 INTRODUCTION

There are 4 basic modes of operation for the SDI. These are determined by the size of the Frame Buffer (12K vs. 48K) and the Mapping mode (bit vs. nybble). The 4 combinations determined by these choices lead to differences in picture resolution and Frame Buffer layout. These differences are seen in Figures 6 and 7. Note that a 12K image may **not** be stored in a 48KTP RAM and that a 48K image may **not** be stored in a 64KZ RAM.

	12 K	48 K
NYBBLE-MAPPED	121 (V) × 189 (H)	241 (V) × 378 (H)
BIT-MAPPED	241 (V) × 378 (H)	482 (V) × 754 (H)

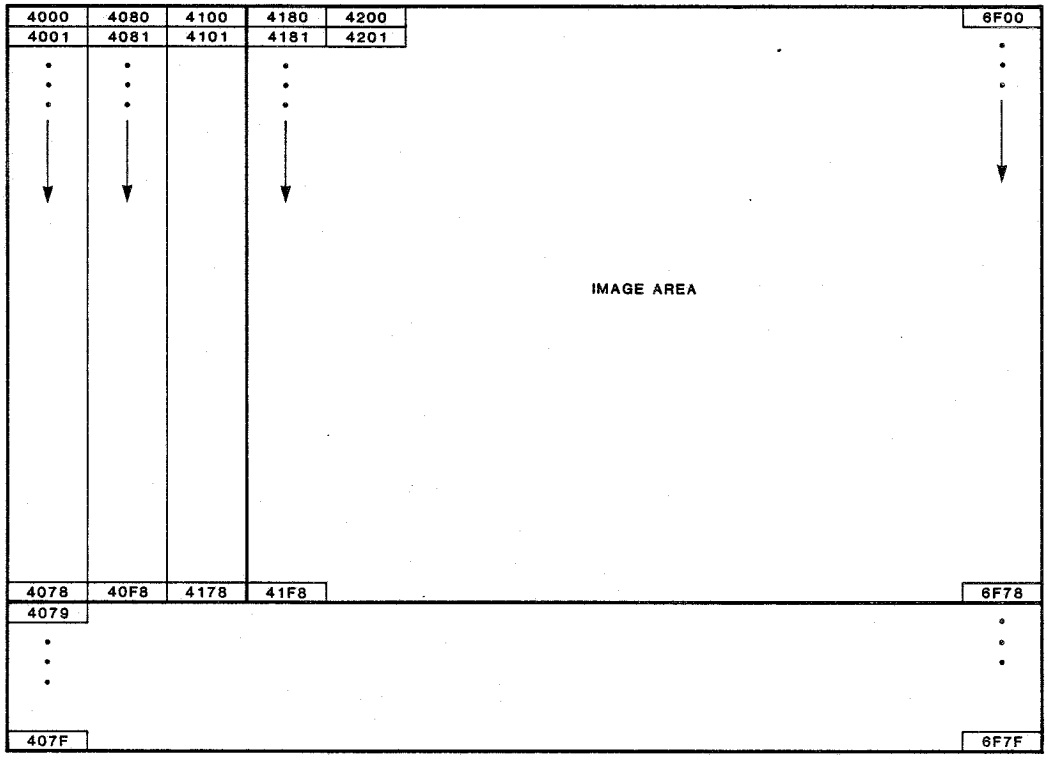
Resolution in the 4 Basic Modes

NYBBLE MAPPED MODE 48K FRAME BUFFER



BASE ADDRESS = 4000H MANDATORY

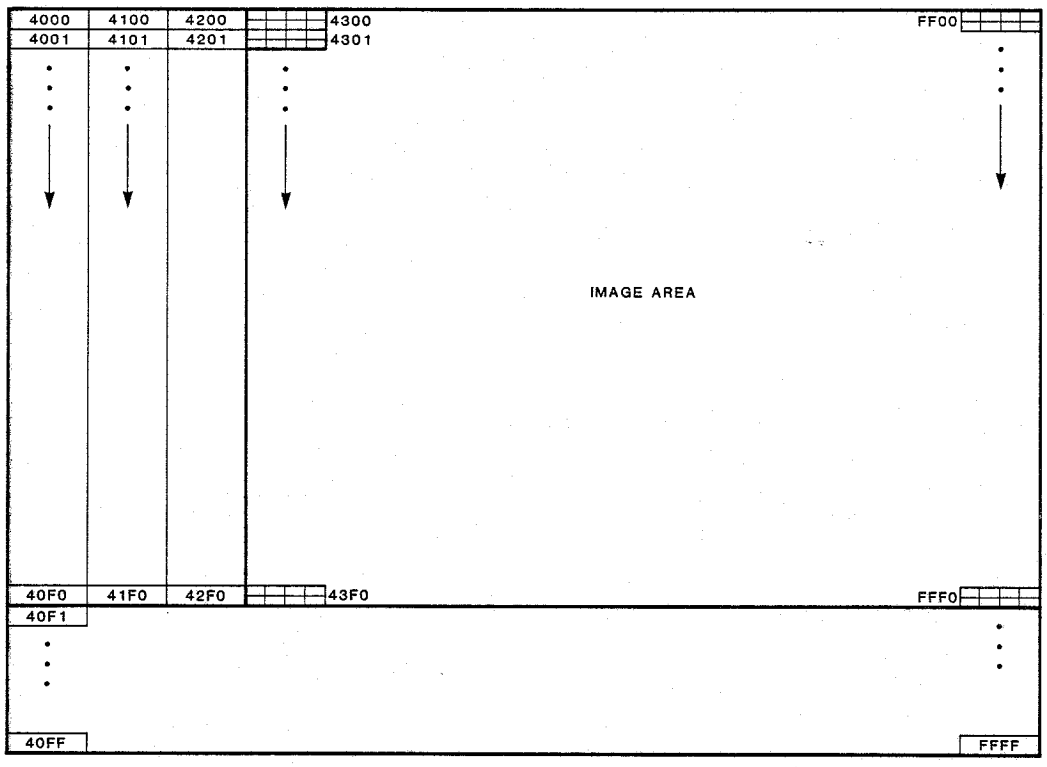
NYBBLE MAPPED MODE 12K FRAME BUFFER



BASE ADDRESS ASSUMED = 4000H, CAN BE 0H, 4000H, 8000H OR C000H

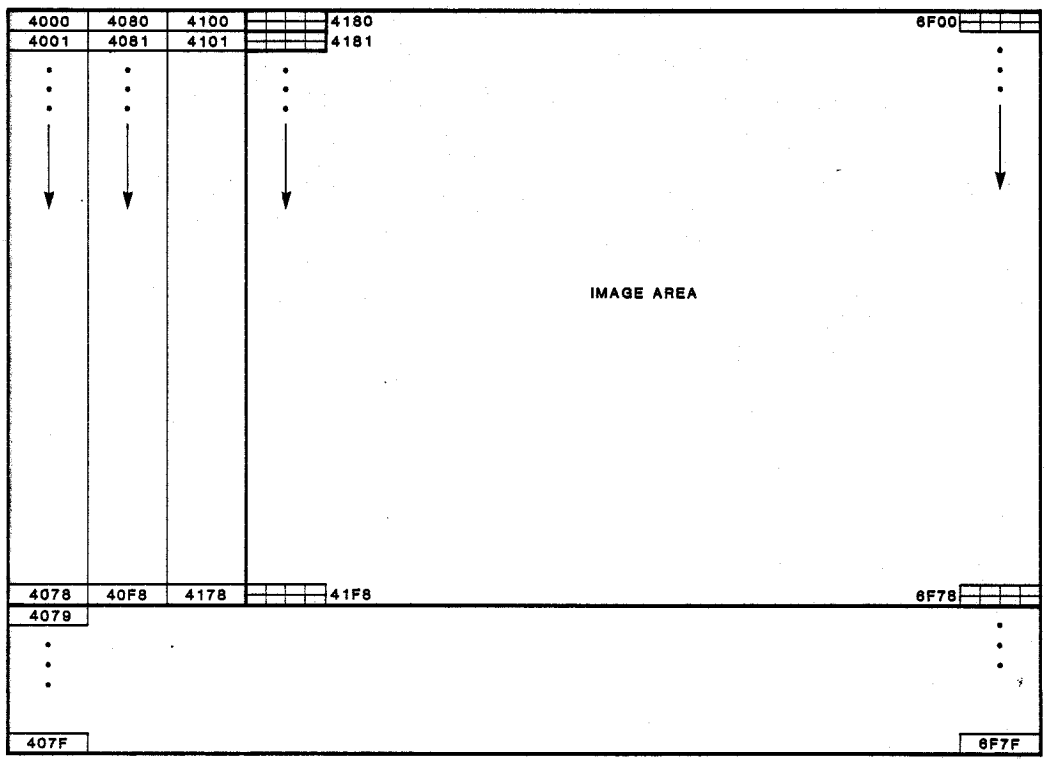
Frame Buffer in the Basic Modes, Nybble Mapped

BIT MAPPED MODE 48K FRAME BUFFER



BASE ADDRESS = 4000H MANDATORY

BIT MAPPED MODE 12K FRAME BUFFER



BASE ADDRESS ASSUMED = 4000H, CAN BE 0H, 4000H, 8000H OR C000H

Frame Buffer in the Basic Modes, Bit Mapped

2.2 THE I/O PORTS IN BRIEF

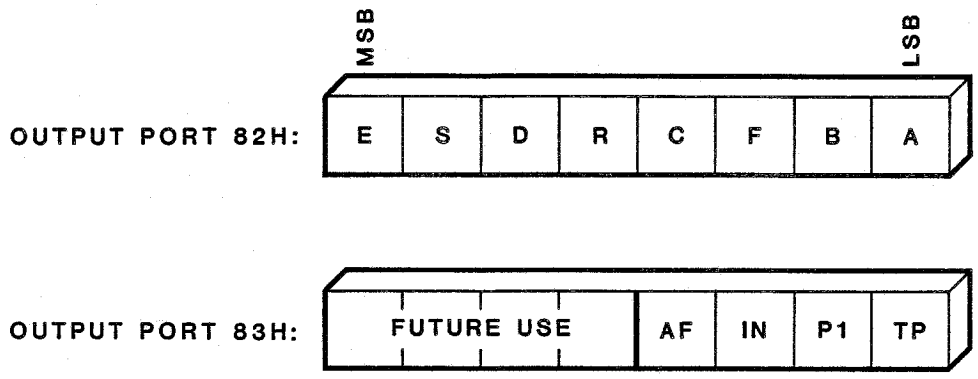
The SDI creates a TV image by continually polling the Frame Buffer and interpreting each bit or nybble in the Image Area as the state of a pixel on the screen. The SDI requires two pieces of information to convert the Image Area into a CRT image:

1. Image Format - size and location of the Frame Buffer and resolution of the picture

2. Color Map - the code for interpreting each bit or nybble in the Image Area as a color/intensity setting for its corresponding pixel on the screen

As the example at the end of this section illustrates, Image Format information is sent to output control ports 82H and 83H. Color Map information is sent to output ports 80H and 81H.

CONTROL PORTS



E 1 = DMA enable
0 = DMA disable

E is effectively the on/off switch for the monitor.

S 1 = 48K mode
0 = 12K mode

Note that a 12K picture cannot be read from a 48KTP board and that a 48K picture may not be stored in a 64KZ RAM board.

D 1 = vertical resolution reduction
0 = normal resolution

When set, the vertical resolution is reduced and CPU efficiency is improved. Used in 48K mode only, especially with the basic **bus sharing** system configuration.

R 1 = reduce vertical range
0 = normal vertical range

Cuts vertical picture size by 25%. Set for increased CPU efficiency, especially with the basic **bus sharing** system configuration.

C 1 = key code enable
0 = normal

Used in bit-mapped mode only. If reset, a 0 in the Image Area will correspond to color 0 in the Mapping RAM. If set, a 0 in the Image Area will correspond to color code 1.

F 1 = bit-mapped option enabled
0 = nybble-mapped mode

When set, the bits in the Control Area of the Frame Buffer are used to determine whether segments of the screen are displayed in bit or nybble mode (see Section 2.6).

B = A15 of the address bus

Locates a 12K base address (ignored in the 48K mode).

A = A14 of the address bus

IN 1 = interleave
0 = no interleave

Allows the pictures in two 48K buffers to share the screen. When set, the bits in the Control Area of the Frame Buffer are used.

P1 1 = select page 1 of Two Port memory
0 = select page 0 of Two Port

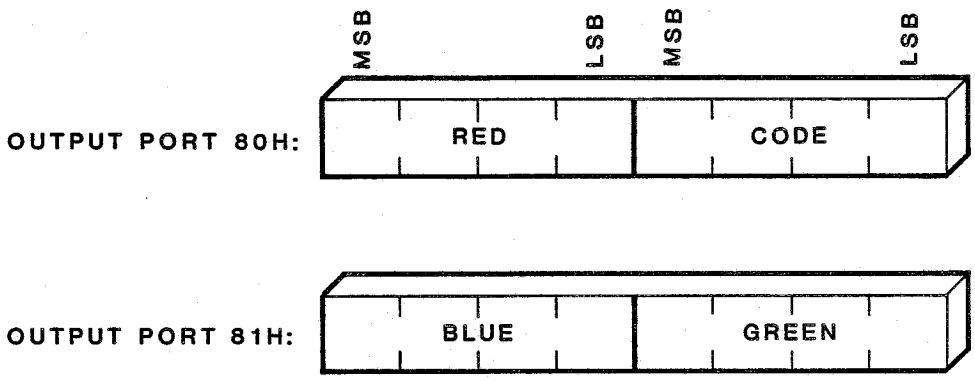
This bit is ignored if the TP bit is reset (thus choosing S-100 rather than second port DMA).

TP 1 = select the Two Port RAMs via second port
0 = select main memory via S-100

Selects location of Frame Buffer.

AF 1 = enable auto-fill mode
0 = disable auto-fill mode

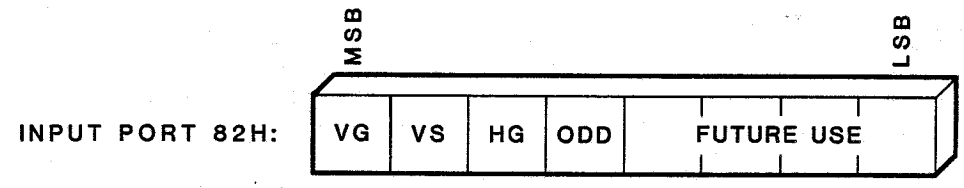
COLOR PORTS



Ports 80H and 81H are used to associate a color on the screen with a code in the Image Area. In nybble-mapped mode there are 16 codes which can be used (0H, 1H,...,FH). In bit-mapped mode there are two codes (0H and FH, or, if the key code bit C is set, 1H and FH). These ports are described in detail in section 2.3.

INPUT PORT

Input port 82H signals the state of the video scanner by reporting the status of the separate sync pulses.



VG Vertical Scan Gate

True during vertical image extent (59.94 HZ).

VS Vertical Blank Indicator

True pulse near start of blank time between fields.

HG Horizontal Scan Gate

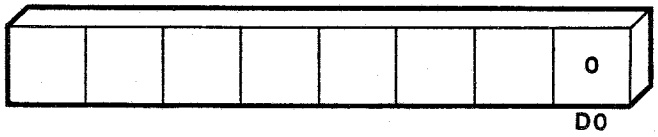
True during horizontal image extent(15.734 KHZ)

ODD True for first field of each frame.

EXAMPLE 1 - TURNING ON THE SDI

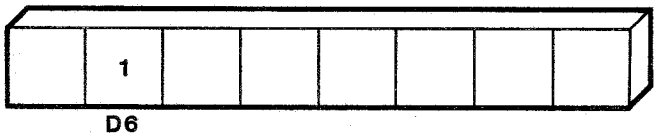
To illustrate the use of control ports 82H and 83H let us assume we wish to display a **48K picture** using **nybble-mapped mode.** The image is stored in the **host memory** so that it must be read using S-100 DMA.

The S-100 DMA is selected by resetting bit D0 of output port 83H:



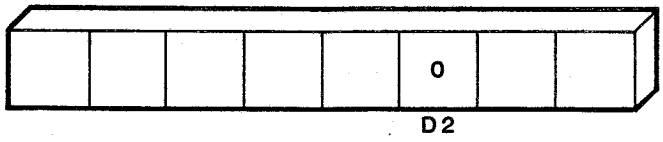
= scan main memory via S-100 bus DMA
= scan TP's via second port DMA

The picture size is communicated to the SDI through bit D6 of port 82H:



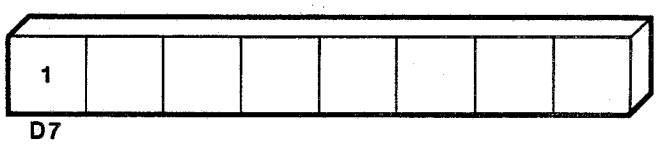
0 = 12K picture
1 = 48K picture

Mapping mode is determined by bit D2 of port 82H:



0 = nybble-mapped mode
1 = bit-mapped mode (Control Area must be properly set)

Finally, the DMA is enabled (i.e., the screen is turned on) using bit D7 of output port 82H:



0 = no DMA (off)
1 = DMA (on)

Putting this all together, we can display the 48K, nybble-mapped picture with the outputs:

```
0000 0000 --------> port 83H
1100 0000 --------> port 82H
```

In Z80 assembly code this can be accomplished with the instructions:

```
ld   a,0h
out  83h,a
ld   a,0C0h
out  82h,a
```

or, in FORTRAN:

```
CALL OUT(Z'83',Z'00')
CALL OUT(Z'82',Z'C0')
```

or, in DEBUG or RDOS:

```
-O 00 83
-O C0 82
```

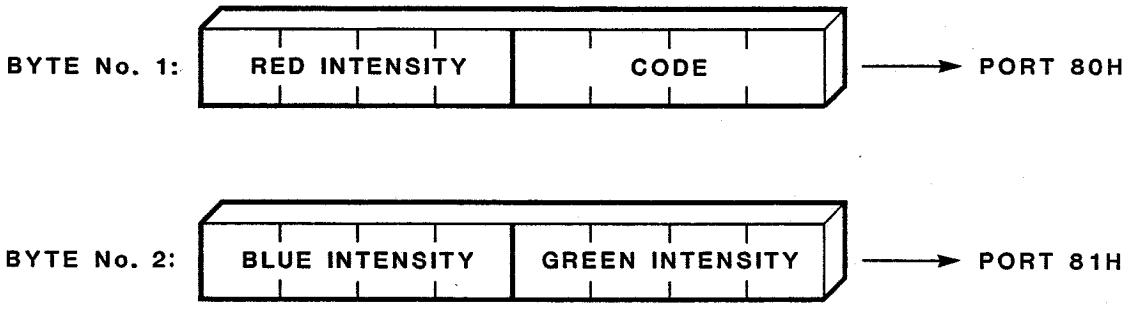
When these instructions are executed the SDI will display the high 48K of user RAM on the color monitor. Note that if the Color Map has not been set the screen may interpret all colors as black and appear blank.

2.3 SETTING THE COLOR MAP

NYBBLE MAPPED MODE

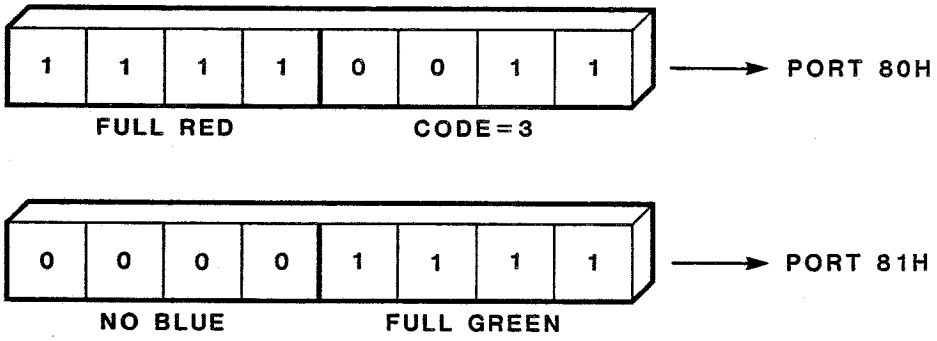
There is a choice of 4096 color/intensity settings. Any 16 of these settings can be displayed on the screen simultaneously. A code from 0H (0000) to FH (1111) is assigned to each chosen color/intensity setting. This information must be output to ports 80H and 81H upon (or prior to) initiation of the image. It need not be updated unless a color which was not among the original 16 is to be displayed.

The format of the Color Map word is:



For example, the following output will cause a 3 in the Image Area to stand for bright yellow (i.e.,

saturated red + saturated green):



This could be accomplished through assembly code:

```
ld   a,0F3h
out  80h,a
ld   a,0Fh
out  81h,a
```

or by using a high level language such as FORTRAN:

```
CALL OUT(Z'80',Z'F3')
CALL OUT(Z'81',Z'0F')
```

The SDI Mapping RAM will now interpret a 0011 nybble in the Image Area as bright yellow.

Two rules should be observed when outputting Color Maps to ports 80H and 81H:

1. Only change the Color Map when the video scanner is not in the picture, that is, during the **blank time.** This avoids irregularities in the TV image. The VSYNC bit of the SDI input port can be used to time such outputs.

2. Always output to port 80H first, then to 81H. This assures that byte #2(blue/green) gets associated with byte #1(red/code).

EXAMPLE 2 - OUTPUTTING A COLOR MAP

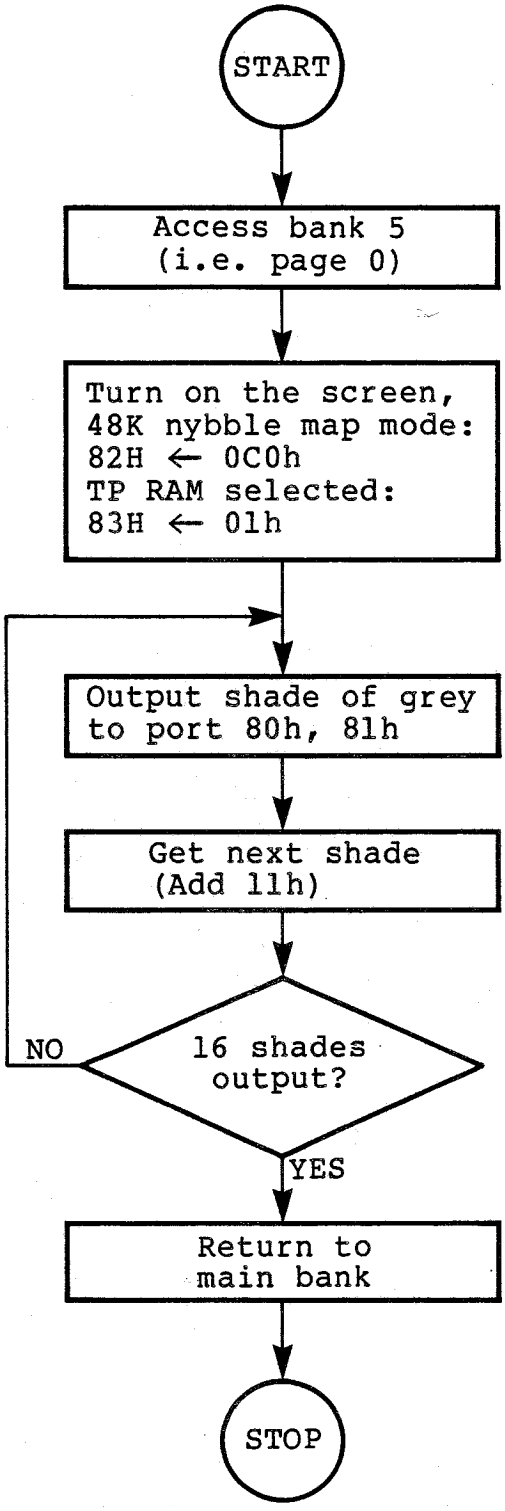
Assume a 48K, nybble-mapped image is stored in the Two Port memory (page 0) and that it has the format of a grey scale image. This means that 0H is the code for black and as the codes increase from 0H to FH they represent increasingly light shades of grey until FH represents pure white.

The software is to 1) turn on the screen, and 2)

output the Color Map. The following program accomplishes this and returns control to CDOS.

```
                ;**************************************
                ;Example 2a : GREYSCALE                *
                ;**************************************


;Acces bank 5 (page 0)
;
START:  ld      a,20h
        out     40h,a           ; select bank 5
;
;Turn on the screen,48k nybble map mode,TP RAM selected
        ld      a,0c0h
        out     82h,a
        ld      a,01
        out     83h,a
;
;Now for the greys...
        ld      a,0             ; start with black
;
loop:   out     80h,a
        out     81h,a
        add     11h             ; and get next shade of grey
        jr      nc,loop         ; if overflow,we are done
;
;Now back to bank 0 where the host memory (and the operating
;system) reside...
        ld      a,1
        out     40h,a           ;select main bank
;
        jp      0               ;reboot
        END     START
```

Program for Example 2a



Flow Chart for Example 2a

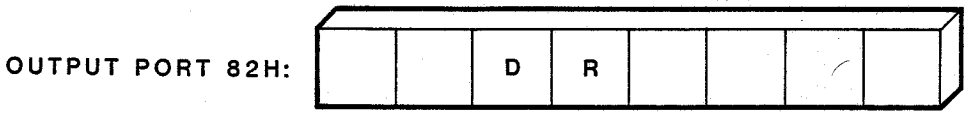
A subroutine which outputs a variety of colors and codes can be written as follows:

```
                ;**************************************
                ;Example 2b: Subroutine CLROUT         *
                ;**************************************

        ENTRY   CLROUT

CLROUT: ld      a,00h
        out     80h,a
        ld      a,00h
        out     81h,a

        ld      a,41h
        out     80h,a
        ld      a,10h
        out     81h,a

        ld      a,12h
        out     80h,a
        ld      a,31h
        out     81h,a

        ld      a,93h
        out     80h,a
        ld      a,5ah
        out     81h,a

        ld      a,44h
        out     80h,a
        ld      a,03h
        out     81h,a

        ld      a,0f5h
        out     80h,a
        ld      a,00h
        out     81h,a

        ld      a,66h
        out     80h,a
        ld      a,52h
        out     81h,a

        ld      a,37h
        out     80h,a
        ld      a,0d3h
        out     81h,a

        ld      a,58h
        out     80h,a
        ld      a,85h
        out     81h,a

        ld      a,69h
        out     80h,a
        ld      a,7ah
        out     81h,a

        ld      a,0bah
        out     80h,a
        ld      a,69h
```

```
out     81h,a

ld      a,09bh
out     80h,a
ld      a,5ch
out     81h,a

ld      a,5ch
out     80h,a
ld      a,0f5h
out     81h,a

ld      a,9dh
out     80h,a
ld      a,9fh
out     81h,a

ld      a,0feh
out     80h,a
ld      a,2fh
out     81h,a

ld      a,0ffh
out     80h,a
ld      a,0ffh
out     81h,a
ret
end     clrout
```

BIT MAPPED MODE

In the bit-mapped mode only 2 colors may be selected for display simultaneously. A 0 in the Image Area causes its associated pixel to have the color assigned by the code 0000. A 1 in that bit gives the pixel the color having code 1111. However, if the key code bit C is set, a 0 in the Image Area will color the pixel the color 0001 (instead of 0000). This is useful if, for example, one wishes to change the background color while in the bit-mapped mode without altering the color map. (Refer to section 2.6.)



2.4 CONTROL BITS D AND R

The D (bit 5) and R (bit 4) bits of output byte 82H have the effect of increasing CPU efficiency at the expense of the size and resolution of the CRT image. Since CPU efficiency is not affected in the Two Port system configuration, these bits should not be used in this case (i.e., they should remain reset). Notice that bit D is ignored in 12K mode.

	12 K	48 K
TP READ	D (NO EFFECT) R=0 RECOMMENDED	D=0, R=0 } RECOMMENDED
S-100 DMA	D (NO EFFECT) R=0 or 1	D=0 or 1 R=0 or 1

a

EFFICACY OF D AND R DURING S-100 DMA (not 2-port):

SWITCH SETTINGS		CPU EFFICIENCY*	
D	R	12 K MODE	48 K MODE
0	0	43.2	8.2
0	1	65.6	32.2
1	0	43.2	43.2
1	1	65.6	65.6

* PERCENT OF TIME CPU HAS ACCESS TO MEMORY

b

Use of Control Bits D and R

R If set, this bit causes the reduction of the vertical range of the screen by 25% (12.5% off top and 12.5% off bottom). The portion of Image Area mapped onto that part of the screen is ignored by the DMA controller. In 48K mode

this eliminates 40H horizontals, total height 177 pixels; in 12K it eliminates 20H horizontals, total height 82 pixels. The utility of this switch is felt if the Frame Buffer resides in main CPU memory. In that situation, if R is set, the reduction in image size buys additional CPU efficiency (see figure).

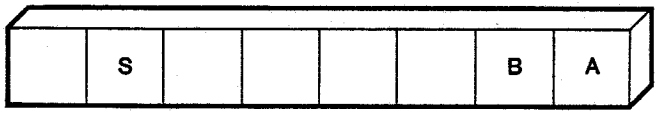
D (48K mode only) Another mechanism for reducing DMA erosion of CPU time is the reduction of vertical resolution. D=1 cuts DMA in half by producing two identical horizontal sweeps of the video scanner, per single horizontal of Frame Buffer. Alternate horizontals in the Image Area are not scanned. This switch eases CPU overhead from 95% to 55%. If both D and R are set simultaneously during DMA of main memory the reduction of CPU efficiency is only 35%.

VERTICAL HEIGHT (IN PIXELS)

SWITCH SETTING	12 K MODE	48 K MODE
R=0	121 (V)	241 (V)
R=1	89 (V)	178 (V)

Vertical Height and the R Switch

OUTPUT PORT 82H:



2.5 CONTROL BITS S, B, AND A

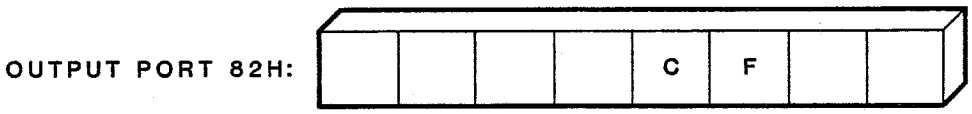
S When set, S selects the 48K (as opposed to the 12K) mode. In the 48K mode the Image Area can be located either in the CPU memory or in a 48K Two Port memory. In either case the image is addressed from 4000H to FFFFH. For 12K operation the image may be stored in any of 4 areas of CPU memory or one of the 16KTP boards (but not in the 48KTP board).

In the 12K mode, the location of the Frame Buffer is determined by control bits A and B.

B,A (12K mode only) These bits act as offsets for DMA addressing. They are equivalent to bits A15 and A14 of the address bus.

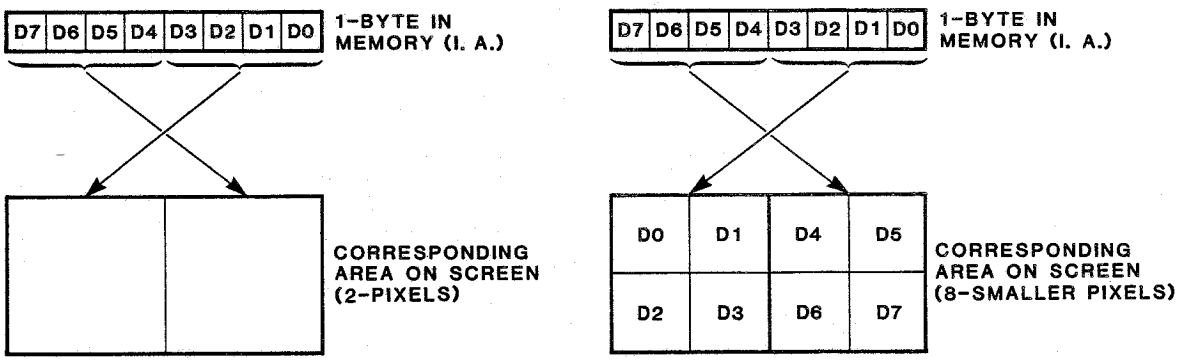
B	A	DMA select (TP=0)	Two Port select (TP=1)
0	0	0 - 2F7FH	board 1
0	1	4000H - 6F7FH	board 2
1	0	8000H - AF7FH	board 3
1	1	C000H - EF7FH	-------

An important consideration when locating a Frame Buffer is the location of the disk operating system. For example, a 48K image stored in main CPU memory would overwrite a 64K CDOS. This may be undesirable for programs making use of the CDOS system calls. If special Two Port memory RAMs are used to hold the Frame Buffer, this difficulty is overcome and all 64K of user RAM may be used by CDOS and SDI programs.



2.6 CONTROL BITS C AND F: BIT MAPPED MODE & THE CONTROL AREA

In super high resolution (bit-mapped) mode, each pixel is divided into four independently controlled sub-pixels. Each bit of the nybble controlling the original pixel then controls its own, smaller, pixel giving four times the point resolution of the nybble-mapped mode. In bit-mapped mode, since there is only one bit in the Image Area per pixel on the screen, only two colors may be displayed simultaneously. A 1 in the Image Area will correspond to the color in Mapping Ram location FH, and a 0 in the Image Area will correspond to the color in Mapping Ram location 0H. Usually, the code 0H will correspond to the color black, so 0's should be used in a bit-mapped Image Area as the background. However, if a colored background is desired, this can be accomplished through use of control bit C, described later in this section.



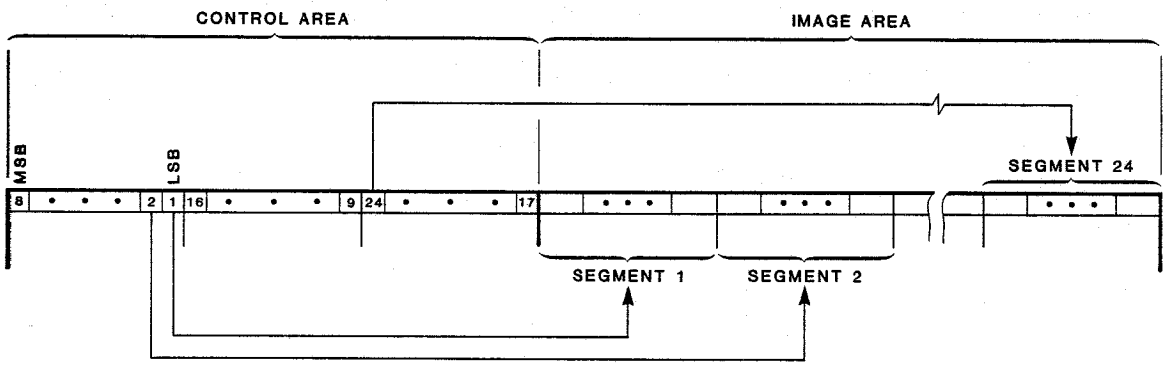
Byte/Pixel Correspondence in the 2 Modes

F The bit-mapped option is selected by setting bit F of output control word 82H, to 1. When reset, normal resolution prevails and nybble-mapped mode is selected. The term **option** rather than **mode** is used because bit F does not, in itself, cause a high resolution format picture to be displayed. Instead, it allows the specification of an area in which a high resolution (bit-mapped) image is to be displayed. This allows part of the screen to

be displayed in the nybble-mapped mode at the same time.

The final determination of high resolution vs. normal resolution is exercised through the bits in the Control Area (that portion of the Frame Buffer to the left of the Image Area). Each bit in the Control Area controls a small part of the Image Area called a **segment.** If F is set, then a 1 in the Control Area causes its corresponding segment to be displayed in high resolution mode. A 0 in this position yields a normal resolution (nybble-mapped) interpretation of the segment.

The association of bits in the Control Area with segments in the Image Area is set by hardware. As an example, we consider the top horizontal of the Image Area (assume a 48K Frame Buffer). This horizontal is divided into 24 segments as shown in Figure 10. The 3 bytes in the Control Area to the 'left' of this horizontal contain the 24 control bits associated with the 24 segments of this horizontal.



IN 48 K MODE EACH BIT IN THE CONTROL AREA CONTROLS A SEGMENT 8 BYTES LONG (except 1st)

The 24 Segments of the Image Area

C Since it is useful to reserve color code 0000 for black (blue off, red off, and green off) it follows that bit-mapped images should be constructed with a background of 0's. This will give a color on black image. To enable the user to display an image on a colored background (color on color image) without making any changes in the Color Map (such as loading a non black color in position 0000 of

the Color Mapping RAM), control bit C is provided. When set, a 0 in the Image Area will correspond to the color stored in location 0001 of the the Mapping RAM, that is, the color having the color code 1. A 1 in the Image Area will remain associated with the color code 1111 (FH).

EXAMPLE 3 - HIGH RESOLUTION

Assume that there is a single 48KTP RAM in bank 5 and the program is stored in the low 16K of memory which is shared by all banks. Furthermore, assume this picture is stored in high resolution format. In order to turn on the screen and display this picture appropriately the following bytes must be output.

Select TP memory:

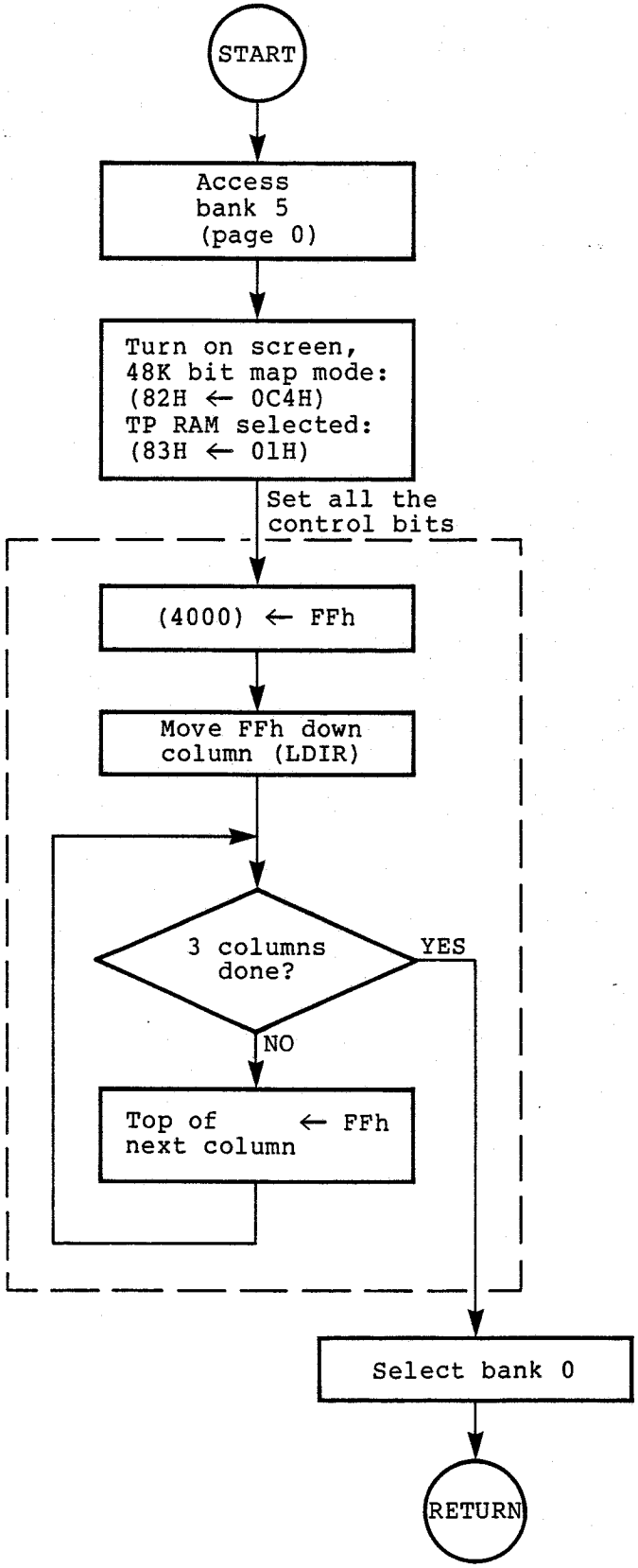
0000 0001 ----> PORT 83H

Turn on screen (set E), signal 48K DMA (set S) and enable bit-mapped mode (set F):

1100 0100 ----> PORT 82H

Finally, to be sure we are displaying the entire image in high resolution mode, all of the control bits in the Control Area of the Frame Buffer should be set.

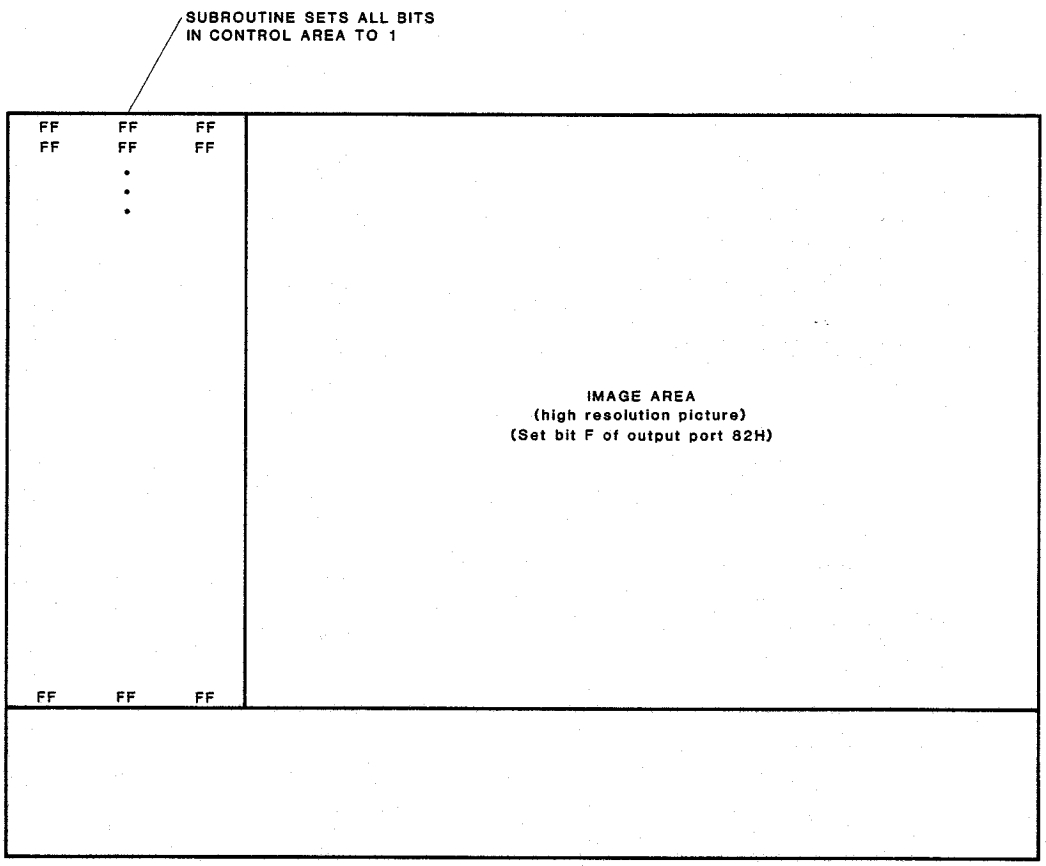
A Z80 flowchart and subroutine follow.



Flow Chart for Example 3

```
                ;**************************************
                ;Example3    : HIGH RESOLUTION          *
                ;**************************************


        ENTRY   SUPHRO

;
SUPHRO: ld      a,20h           ;select bank 5 (page 0)
        out     40h,a           ;to write into it
;
;Issue control outputs to SDI...
        ld      a,0c4h          ;turn on screen,48k bit mapped mode
        out     82h,a
;
        ld      a,1             ;selet TP RAM
        out     83h,a
;
        ld      a,0
        out     80h,a
        ld      a,0
        out     81h,a           ;color code 0=black
;
        ld      a,0ffh
        out     80h,a
        ld      a,0ffh
        out     81h,a           ;color code 15=white
;
;Now set all the control bits...
        ld      hl,4000h        ;point to control area
        ld      de,4001h
nxtcol: ld      a,0ffh          ;set 1st byte of column of control area
        ld      (hl),a
        ld      bc,0f0h         ;define length of column
        ldir
;
        inc     h               ;do next column
        inc     d
        ld      l,0
        ld      e,l
        ld      a,h             ;check whether 3 columns of control area
        cp      43h             ;have been filled ?
        jr      nz,nxtcol       ;go if not being filled
;
;Back to main bank...
        ld      a,01
        out     40h,a
;
        ret
        end     suphro
```

Program for Example 3



Frame Buffer after Routine Suphr

EXAMPLE 4 - DUAL RESOLUTION

Assume there is a 12K Frame Buffer in main memory starting at base address 8000H. The image is stored in this buffer in such a way that the upper left hand corner is to be normal full color resolution and the remainder of the Image Area is stored in high resolution format (see figure, a dual resolution image).

The control bits in such a situation could be set as in Figure 12b.

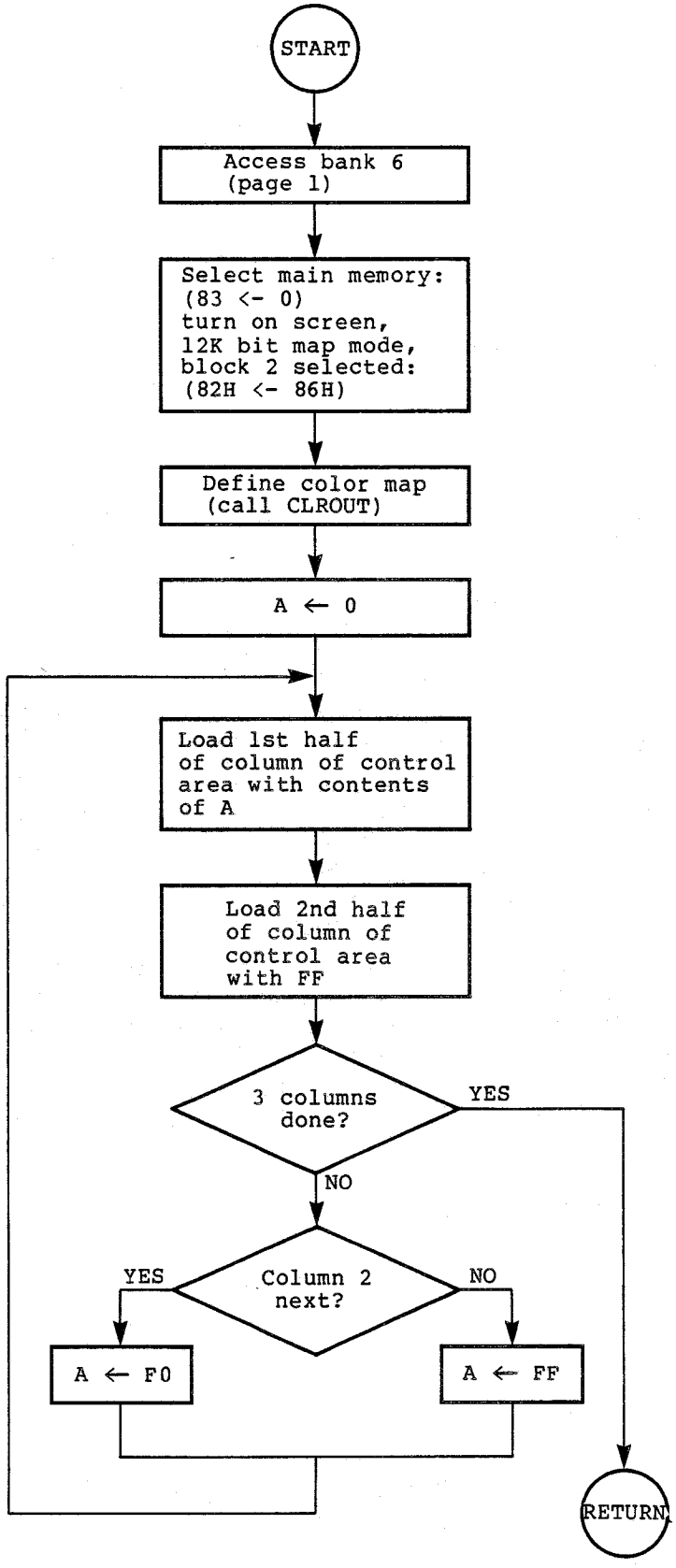
The outputs to be made are:

Select main memory:

0000 0000 ----->PORT 83H

Turn on screen (set E), signal 12K DMA (reset S), enable bit map (set F), and locate Frame Buffer in 8000H (bits B,A = 1,0):

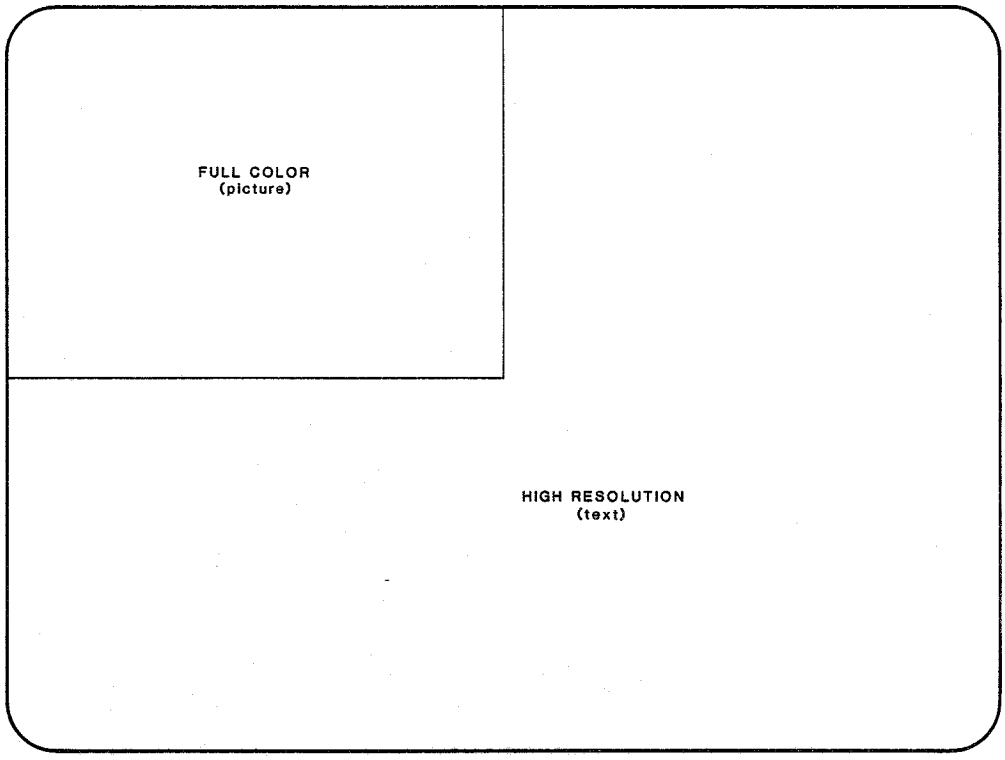
0100 0110 ----->PORT 82H

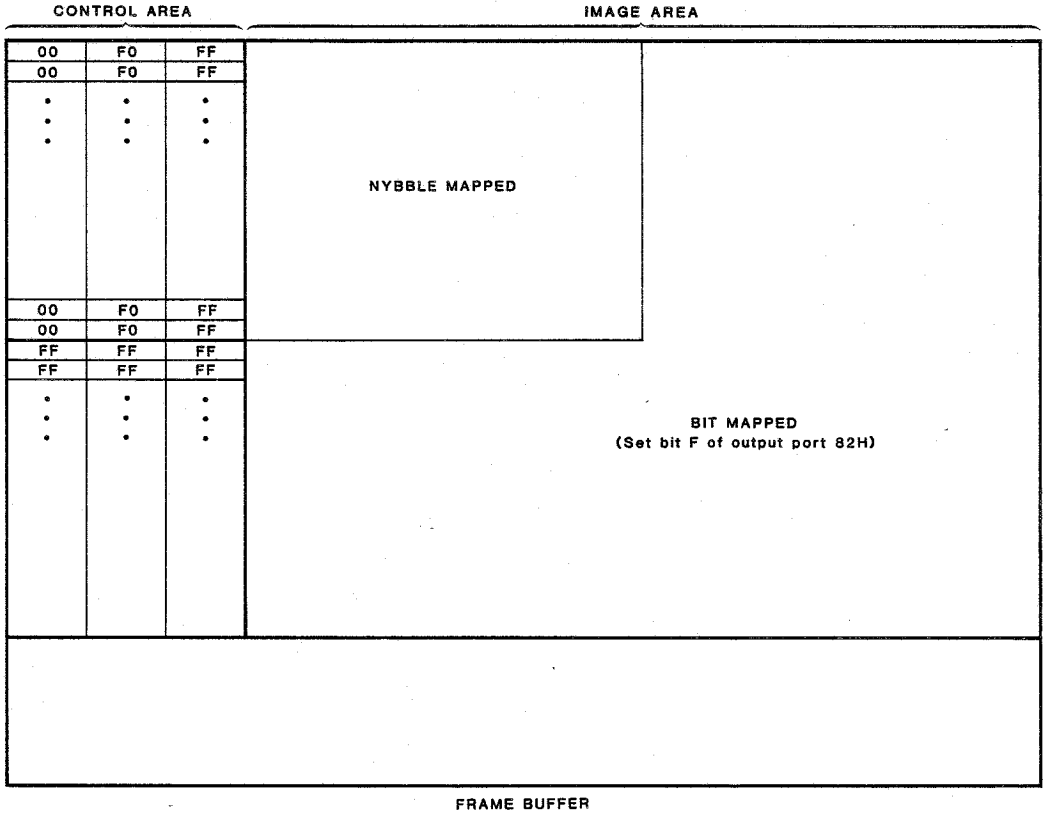


Flow Chart for Example 4

```
                ;*******************************************
                ; Example4  : DUAL RESOLUTION                *
                ;*******************************************


        ENTRY   PICTXT
        EXT     CLROUT


;Access bank 6 (page 1)
PICTXT: ld      a,40h
        out     40h,a
;
;Issue control outputs...
        ld      a,0
        out     83h,a
;
        ld      a,86h
        out     82h,a           ;8000 page of main memory
;
;Subroutine CLROUT (example 2b) sets color map
        call    clrout
;
;Point to the control area
        ld      hl,8000h
        ld      de,8001h
;
        ld      a,0
nxtcol: ld      (hl),a          ;fill 1st half of column
        ld      bc,3ch          ;length of half a column
        ldir
;
        ld      a,0ffh          ;fill 2nd half of column
        ld      (hl),a
        ld      bc,3ch
        ldir
;
        ld      a,h             ;3 columns done?
        cp      81h
        jr      z,out
;
;Point to next column of control area
        ld      bc,8h           ;now point to next column...
        add     hl,bc
        ex      de,hl
        add     hl,bc
        ex      de,hl
;
;do we do column 2 or 3 next?
        ld      a,1
        cp      0
        jr      z,col3
;
        ld      a,0f0h          ;column 2
        jr      nxtcol
;
col3:   ld      a,0ffh          ;column 3
        jr      nxtcol
;
out:    ld      a,1             ;back to main bank
        out     40h,a
;
        RET
        END     PICTXT
```

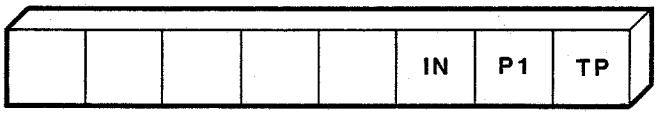
Program for Example 4





A Dual Resolution Image

OUTPUT PORT 83H:



2.7 OUTPUT PORT 83H: TP, P1, AND IN; THE CONTROL AREA REVISITED

The bits in output port 83H are of interest only if the user has one or more Two Port memory boards. Otherwise they should remain reset at all times.

In the Two Port system configuration (Figure 2) there is a choice of where the Frame Buffer can be stored. For a 48K picture, the buffer may be in the high 48K of main memory or either of the 48K TP memories (page 0 or 1). For a 12K picture there are 4 choices of Frame Buffer location within each 64K of memory.

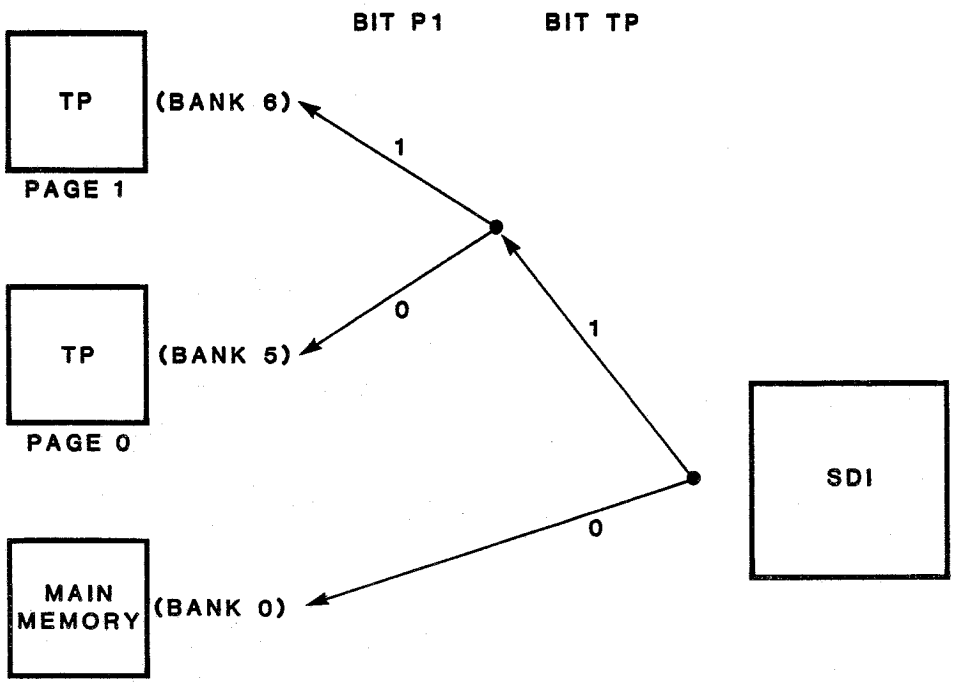
Port 83H is used to communicate the bank location of the Frame Buffer to the SDI. In addition, if a 12K picture is being displayed, port 82H (bits B and A) is needed to determine the exact location within the bank (see section 2.5).

Finally, it is possible to display more than one Frame Buffer on the screen simultaneously. In a fashion which parallels the use of control bits in dual resolution (section 2.6), the programmer can assign distinct portions of the screen to different Frame Buffers. Examples are supplied at the end of this section.

TP When set, the dedicated Two Port RAM is selected for DMA scanning. When reset, bank 0, main memory, is selected for DMA.

48K MODE:
The switch settings recommended for these boards will locate the Frame Buffer in the 4000H to FFFFH range of either bank 5 (page 0) or bank 6 (page 1). If there are two 48KTP buffer areas and TP is set, the SDI needs to know which 48K page to scan. Control bit P1 determines this choice (see below).

12K MODE:
In addition to the bank select considerations just mentioned, a 12K Frame Buffer must be located within the selected bank. Bits B and A of control byte 82H communicate this information to the SDI (see section 2.5).



Selection of Bank for DMA

P1 P1 = 0 selects page 0 (bank 5) for DMA
P1 = 1 selects page 1 (bank 6)

If TP is reset to 0 the state of bit P1 is ignored by the SDI. This bit is only used if 1) two or more TP RAMs are employed and 2) the TP RAMs are addressed as separate pages from the second port. This is the situation in Figure 2. If these conditions are not met, any TP RAMs should be addressed as page 0 and P1 should remain reset at all times.

Control bit P1 only becomes relevant if bit TP is set, indicating that the SDI is reading from the second port of the Two Port RAM.

IN (Page Windowing): This bit controls the capability of the SDI to view both pages of image memory simultaneously.

When reset, the displayed page is either page 0 (bank 5), page 1 (bank 6) or main memory (bank 0) depending on the settings of control bits P1 and TP.

When set, portions of page 0 and page 1 may appear on different parts of the screen simultaneously. That is, page 0 will have certain areas of the screen reserved for its

display, while the complement of the screen will be reserved for page 1. **The areas reserved for each page are determined by the Control Area in the Frame Buffer of page 0.**

As in the case of dual resolution mode, each bit in the page 0 Control Area dominates a certain horizontal segment in the Image Area. In the present case, however, a control bit set to 1 means that the segment dominated by that bit will "window through to" (i.e., be reserved for) page 1. If reset to 0, its corresponding segment will display page 0. Please see the figure in section 2.6, the 24 segments of the Image Area.

Note that only the control bits on page 0 are used for page windowing. The control bits on page 1 have absolutely no effect when bit IN is set to 1. (See programming examples 5 and 6.)

THE SIMULTANEOUS USE OF CONTROL BITS IN AND F

When IN and F are both 0, the Control Areas of each Frame Buffer are inactive and the values in these areas are irrelevant. It was shown in section 2.6 that if bit F is set, the Control Area of the page being displayed controls the layout of high resolution areas on the screen. In section 2.7 it was shown that when bit IN is set, the Control Area of page 0 is used for reserving certain areas of the screen for page 1 (and the page 1 Control Area is ignored).

Since the Control Areas of pages 0 and 1 cannot perform both of these unrelated functions simultaneously, the following rule is observed when bits IN and F are both set:

1. The Control Areas of pages 0 and 1 function as they do in normal window mode. Thus parts of the screen will display page 0 and the complement will display page 1. This determination will be made by page 0's Control Area. Page 1's Control Area will be ignored.

2. Those areas which display page 0 will show that page in nybble-mapped, full color mode.

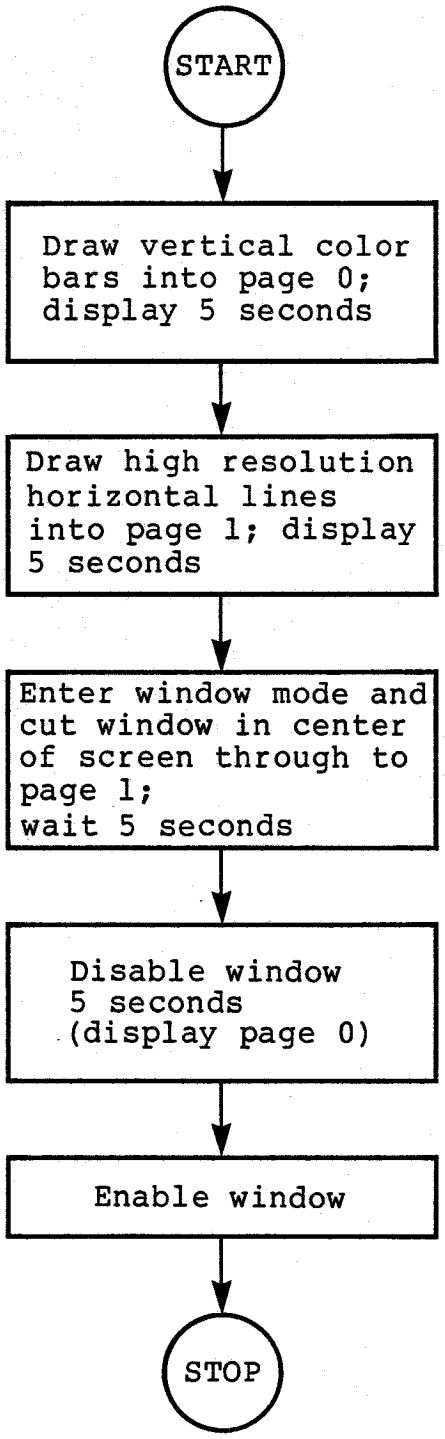
3. Those areas which display page 1 will show

that page in bit-mapped, high resolution mode.

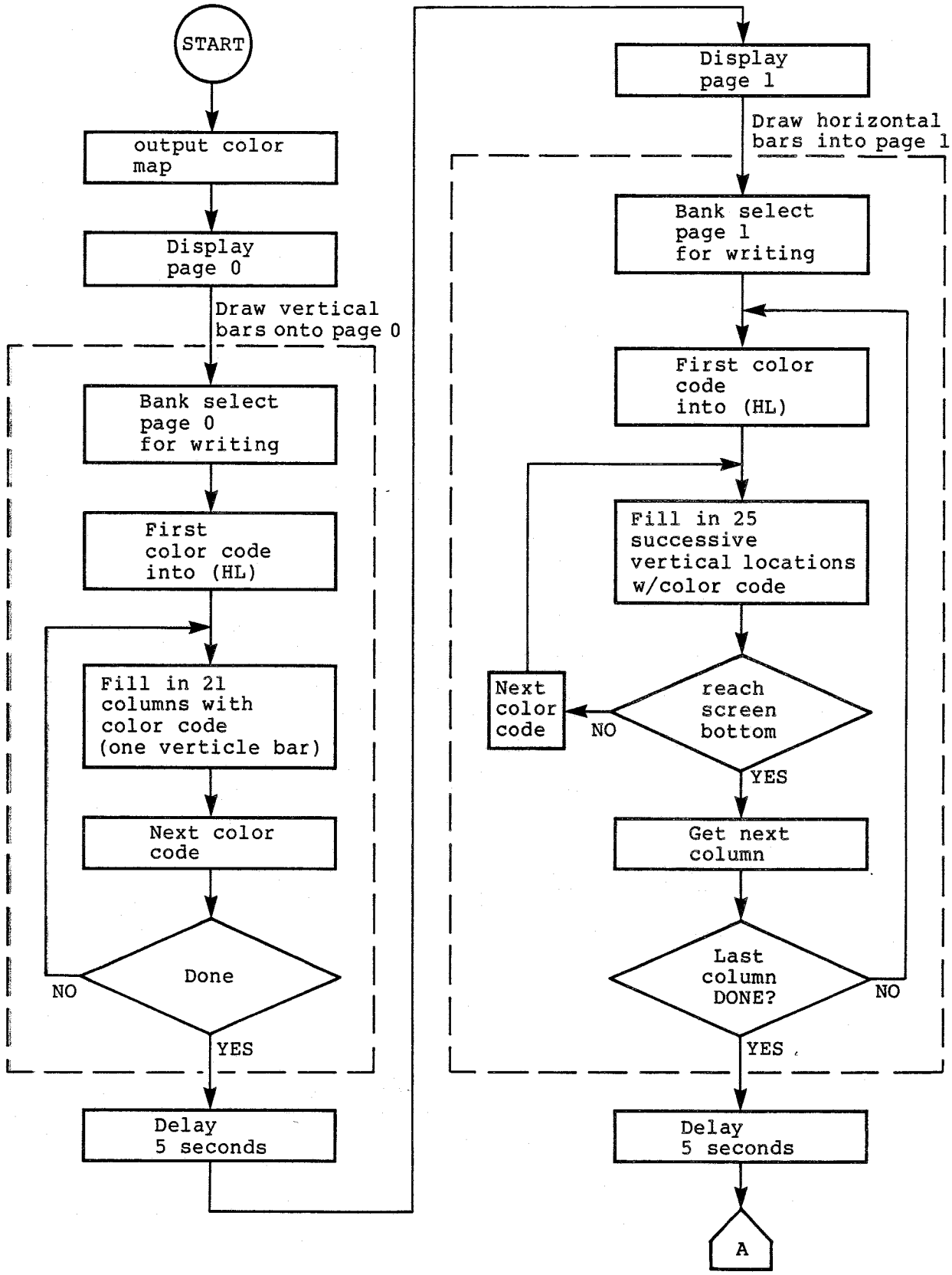
IN \ F	0	1
0	Control area of neither page used. Displayed page is either 0 or 1. Resolution is nybble mapped.	Control area of page being displayed used to specify hi-res (bit mapped) areas.
1	Control area of page 0 used to specify windows through to page 1. both pages in nybble mapped resolution.	Control area of page 0 used to specify windows through to page 1. Page 0 entirely nybble mapped res. Page 1 entirely bit mapped res.

Effect of Simultaneous Settings of Control Bits IN and F

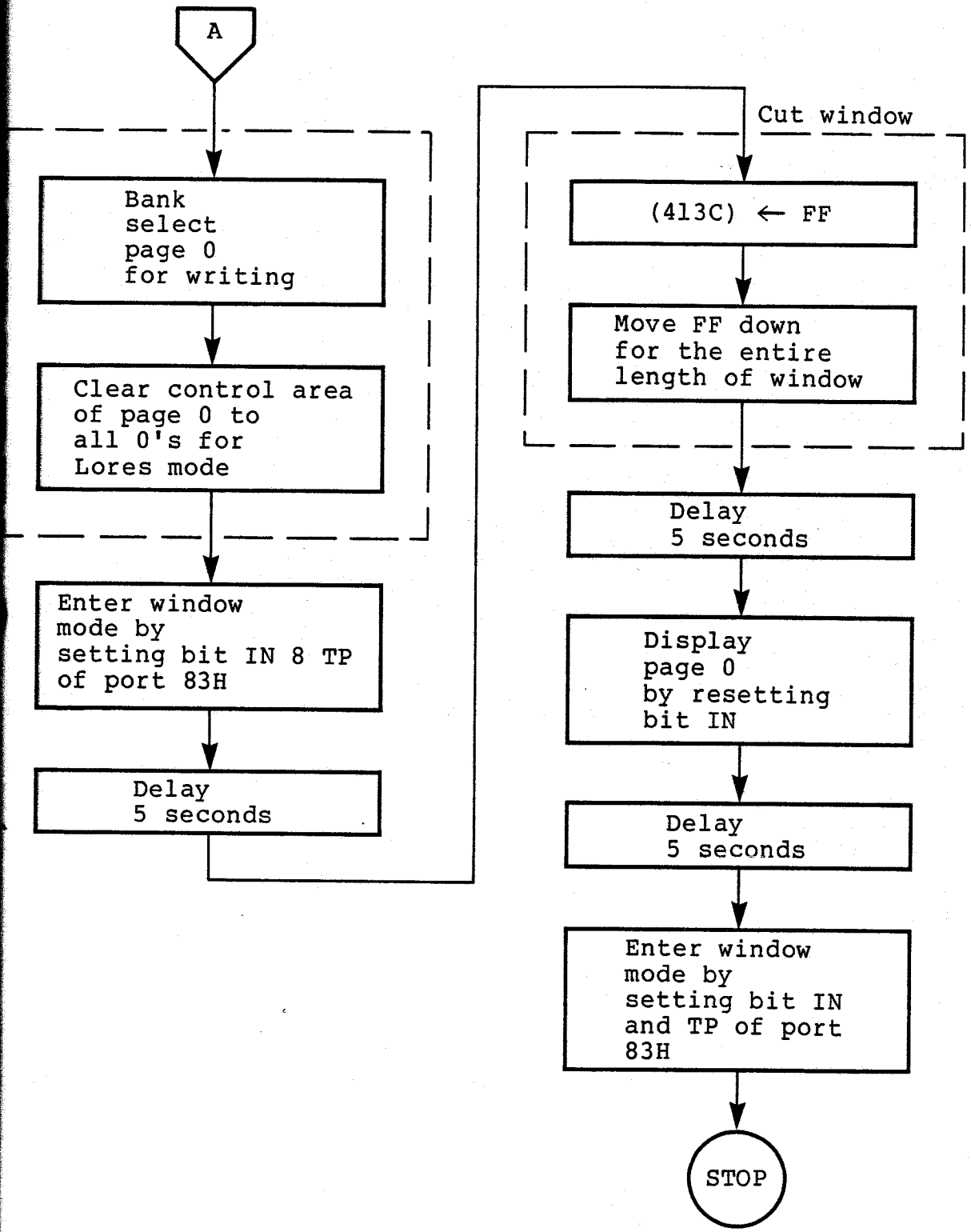
EXAMPLE 5 - WINDOW MODE



General Flow Chart for Example 5



Detailed Flow Chart 1 for Example 5



Detailed Flow Chart 2 for Example 5

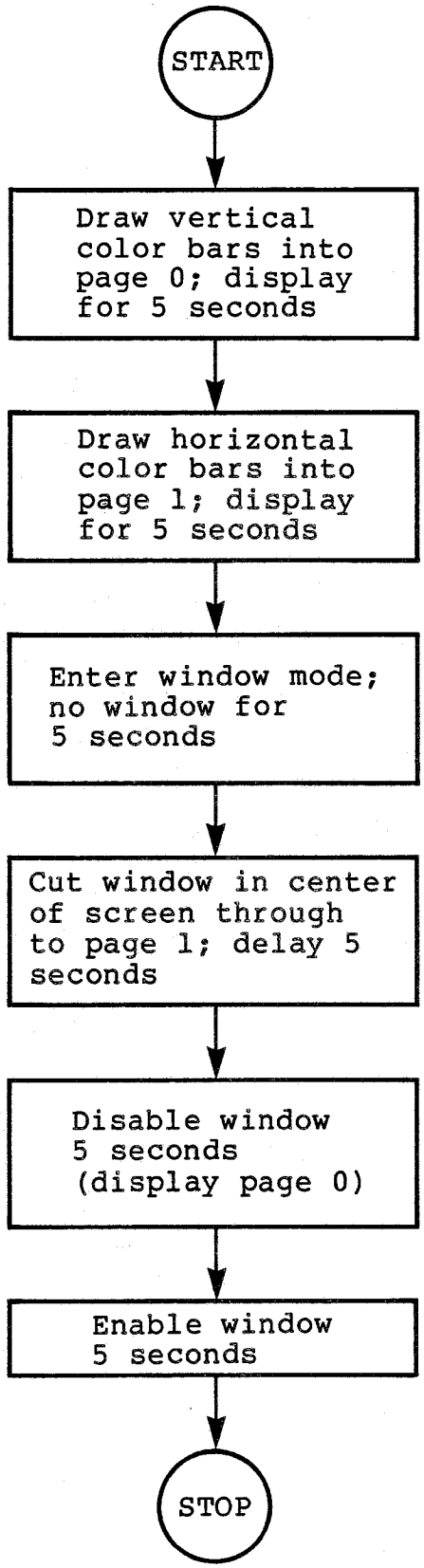
```
                ;*******************************
                ;Example5   :WINDOW MODE       *
                ;*******************************


        EXT     DELAY,CLROUT


        DS      30
STK:    DS      0
BEGIN:  ld      sp,stk

;Define color map
        call    clrout

;display page 0

        ld      a,0c0h          ;turn on the screen
        out     82h,a           ;select 48k nybble map mode
        ld      a,1             ;select page 0 via 2nd port DMA
        out     83h,a


;draw 9 vertical color bars of 21 columns each on page 0
        ld      a,20h           ;select page 0
        out     40h,a           ; & write into it via S100 bus
        ld      hl,4300h        ;hl points to 1st location of image area
        ld      de,4301h
        ld      a,11h           ;color code #1 ,doubled in a-register
nxtcol: ld      (hl),a
        ld      bc,5375         ;length of 21 columns  (256 *21 -1)
        ldir                    ;fill in 21 columns with color code
        add     a,11h           ;on to the next color code
        cp      0aah            ;check whether 9 bars have been drawn
        jp      z,next
        inc     hl              ;next 21 columns
        inc     de
        jp      nxtcol

;wait 5 seconds
NEXT:   call    delay

;display page 1
        ld      a,03h
        out     83h,a

;draw 10 horizontal color bars of 25 rows each on page 1
        ld      a,40h           ;select bank 6 and write to it
        out     40h,a
        ld      hl,4300h
        ld      de,4301h
        ld      a,0
ncolor: add     a,11h           ;color code in a-register
        ld      (hl),a
        ld      bc,24           ;bc is counter for 25 rows
        ldir                    ;fill in 25 succesive locations
                                ;of the column with color code
        ld      b,a             ;save color code in b-register
        ld      a,1
        cp      a,0f0h          ;have we reach screen bottom?
```

Program for Example 5

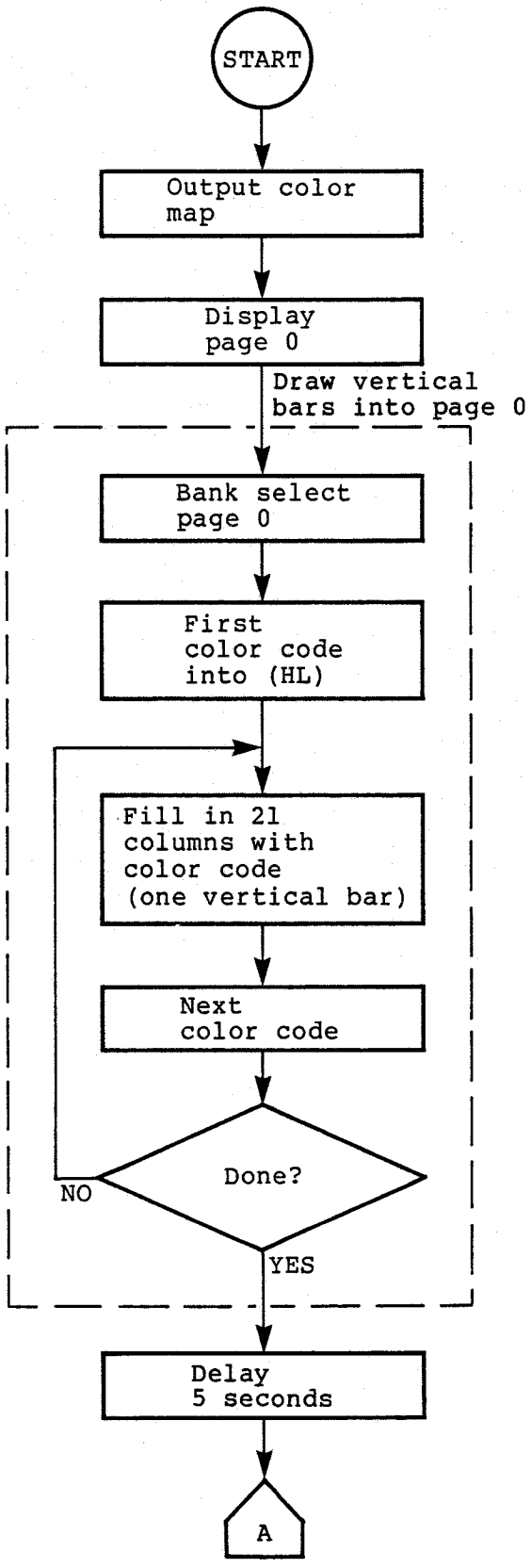
```
        jp      nc,ncolum       ;if so, do next column
        ld      a,b             ;if not,restore color code
        inc     l               ;next horizontal bar
        inc     e
        JP      ncolor          ;continue painting downwards-next color
ncolum: inc     h               ;hl points to next column
        jp      z,nextl         ;go if did reach last column
        inc     d
        ld      l,0
        ld      e,l
        ld      a,0
        jp      ncolor          ;do next column

;wait 5 seconds
NEXTl:  call    delay

;clear control area of page 0 to all 0's
        ld      a,20h
        out     40h,a           ;select bank 5
        ld      hl,4000h        ;hl points to lst location in control area
        ld      de,4001h
ncoll:  ld      a,0
        ld      (hl),a
        ld      bc,0f0h         ;length of column
        ldir                    ;fill in the entire column with 0
        inc     h
        inc     d
        ld      l,0
        ld      e,l
        ld      a,h
        cp      43h             ;check whether last column
                                ;of control area being filled
        jr      nz,ncoll        ;do next column of control area

;enter window mode
        ld      a,05h           ;set bit IN and TP of port 83h
        out     83h,a

;wait 5 seconds
        call    delay

;cut a window thru to page 1 in the center of screen
        ld      hl,413ch
        ld      de,413dh
        ld      a,0ffh
        ld      (hl),a
        ld      bc,78h          ;length of window
        ldir

;wait 5 seconds
        call    delay

;display page 0
        ld      a,1
        out     83h,a

;wait 5 seconds
        call    delay

:enter window mode
        ld      a,5
        out     83h,a

;Back to main bank
        ld      a,01h
        out     40h,a
        jp      0               ;so long for now...
        end     begin
```

Program for Example 5

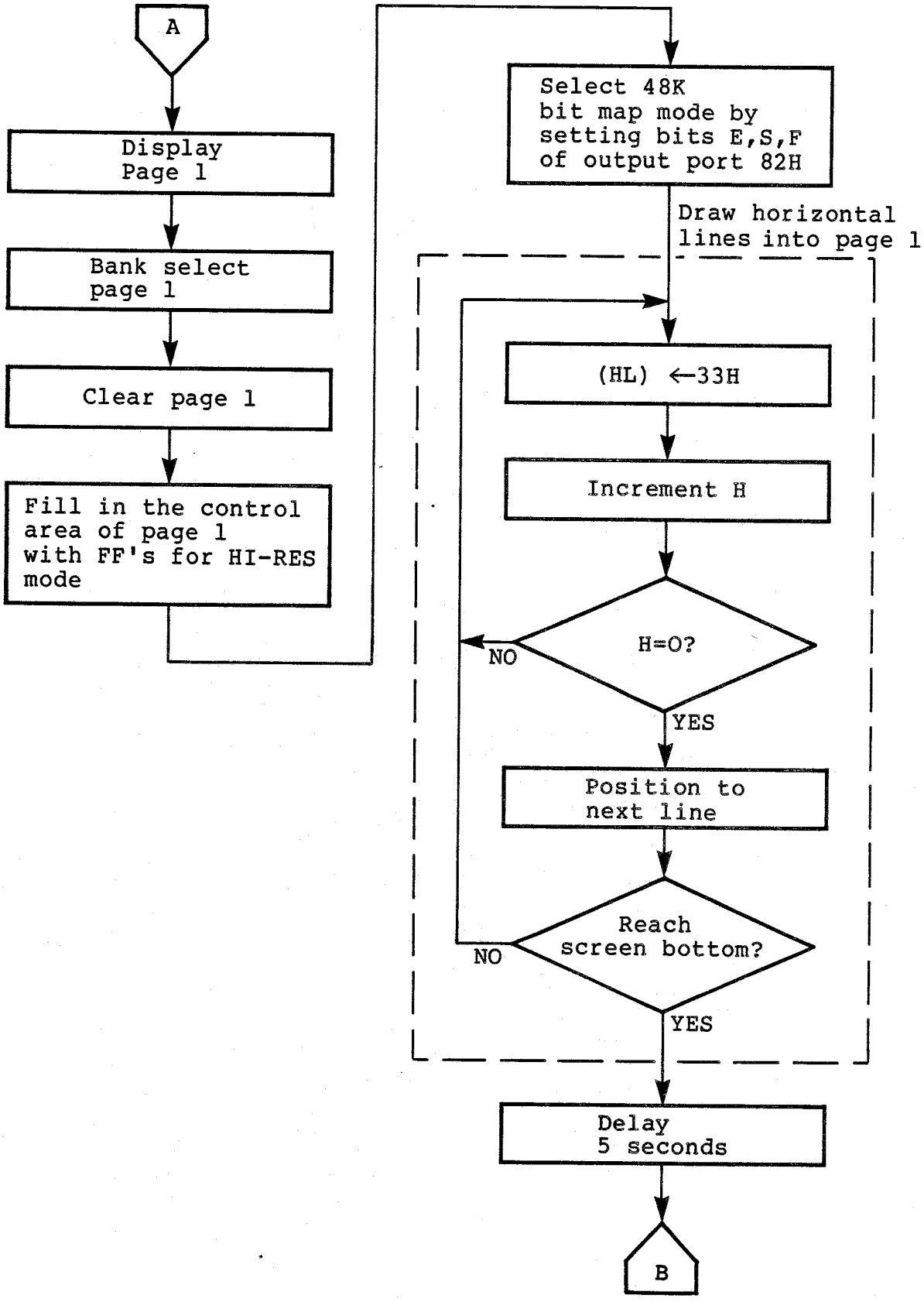
EXAMPLE 6 - WINDOW MODE & HIGH RESOLUTION



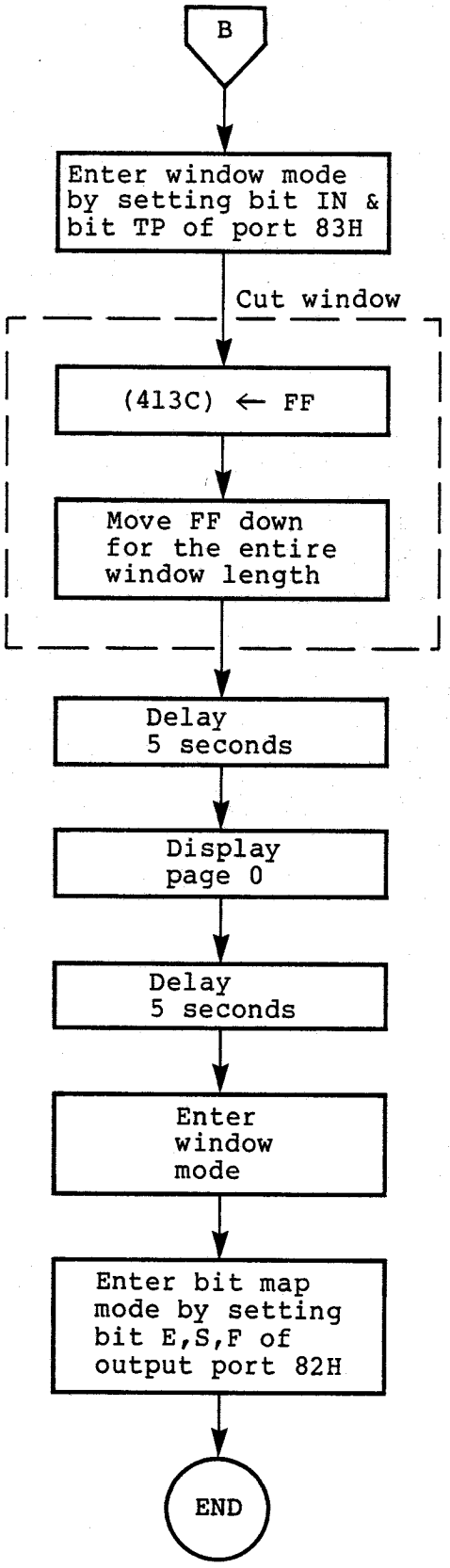
General Flow Chart for Example 6



Detailed Flow Chart 1 for Example 6



Detailed Flow Chart 2 for Example 6



Detailed Flow Chart 3 for Example 6

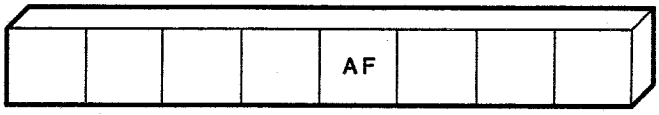
```
                ;*******************************************************
                ; Example6 : WINDOW MODE + HIGH RESOLUTION             *
                ;*******************************************************


        EXT     DELAY,CLROUT


        DS      30
STK:    DS      0


BEGIN:  ld      sp,stk

;Define color map
        call    clrout
;
;display page 0

wline:  ld      a,0c0h          ;turn on the screen
        out     82h,a           ;select 48k nybble map mode
        ld      a,1             ;select page 0 via 2nd port DMA
        out     83h,a


;draw 9 vertical color bars of 21 columns each on page 0
        ld      a,20h           ;select page 0
        out     40h,a           ; & write into it via S100 bus
        ld      hl,4300h        ;hl points to 1st location of image area
        ld      de,4301h
        ld      a,11h           ;color code #1 ,doubled in a-register
nxtcol: ld      (hl),a
        ld      bc,5375         ;length of 21 columns  (256 *21 -1)
        ldir                    ;fill in 21 columns with color code
        add     a,11h           ;on to the next color code
        cp      0aah            ;check whether 9 bars have been drawn
        jp      z,next
        inc     hl              ;next 21 columns
        inc     de
        jp      nxtcol

;wait 5 seconds
NEXT:   call    delay

;display page 1
        ld      a,03h
        out     83h,a

;clear page 1
        ld      a,40h           ;select bank 6 & write to it
        out     40h,a
        ld      hl,4000h        ;hl points to 1t location of control area
        ld      de,4001h
        ld      a,0
        ld      (hl),a
        ld      bc,0BFFFh       ;c000h is the total number of locations
                                ;of entire frame buffer(c000h=256*192)
        ldir
```

Program for Example 6

```
;set control area of page 1 to all ff's
        ld      hl,4000h        ;hl points to 1st location in control area
        ld      de,4001h
ncoll:  ld      a,0ffh
        ld      (hl),a
        ld      bc,0F0H         ;length of column in control area
        ldir                    ;fill in the entire column with 0ffh
        inc     h
        inc     d
        ld      l,0
        ld      e,l
        ld      a,h
        cp      43h             ;have we set the entire control area?
        jr      nz,ncoll        ;do next column of control area
;send out hi-res bit map mode
        ld      a,0c4h          ;dma enable,48k bit map mode select
        out     82h,a

;draw hi-res horizontal lines on page 1
        ld      hl,4300h        ;hl points to 1st location of image area
        ld      de,10h
again:  ld      a,33h           ;33h is a mask to select the 4 top
                                ;pixels corresponding to 4 bits of
                                ;a byte that are set
        ld      (hl),a
        inc     h
        jr      nz,again        ;go if not finish drawing the line
        add     hl,de
        ld      h,43h           ;move down 10h rows for next line
        ld      a,l
        cp      0               ;have we reach screen bottom?
        jr      nz,again        ;if no,loop back & draw next line

;wait 5 seconds
        call    delay

;enter window mode(bit map mode already set)
        ld      a,05h           ;set bit IN and TP of port 83h
        out     83h,a

;cut a window thru to page 1 in the center of screen
        ld      hl,413ch
        ld      de,413dh
        ld      a,0ffh
        ld      (hl),a
        ld      bc,78h          ;length of window
        ldir

;wait 5 seconds
        call    delay

;display page 0
        ld      a,0c0h          ;select 48k nybble map mode
        out     82h,a
        ld      a,1
        out     83h,a

;wait 5 seconds
        call    delay

;enter window mode & bit mapped mode
        ld      a,5
        out     83h,a
        ld      a,0c4h
        out     82h,a

;Back to main bank
        ld      a,01h
        out     40h,a
        jp      0               ;so long for now...
        END     BEGIN
```

Program for Example 6

OUTPUT PORT 83H:



2.8 OUTPUT PORT 83H: AF; AUTOMATIC AREA FILL

In normal nybble-mapped mode, in order to create a solid colored area (i.e., an area which is filled in with color), the desired color code must be written into memory a great number of times, especially for large areas. Indeed, virtually all color interfaces which work on the raster scan principle require a time consuming massive write operation in order to fill large areas with color.

The SDI offers an **Automatic Area Fill** mode whereby solid areas can be created in the same time it takes to draw the borders of the areas.

This is accomplished by changing the way in which the SDI interprets data in the Frame Buffer. For example, when Automatic Fill mode is enabled, a 0 in the Image Area is no longer interpreted as an instruction to color the given pixel with the color associated with code 0. Rather, it takes on a new meaning. It is now interpreted as an instruction to color the given pixel the same color as the pixel to the left. All non-zero codes (1-15) retain their old meaning as color codes. With this simple change automatic area fills are made possible.

It should be observed that since Auto-Fill treats data in the Frame Buffer differently from the normal nybble-mapped mode, the user should fix the mode (AF or normal) at the onset of image creation and keep the mode unchanged whenever that image is being displayed. Undesired results will be obtained if a picture is drawn in normal mode and later the AF bit is set, or visa versa. Images compatible with both modes can be generated if the programmer takes into account the effect of each mode.

AF When reset, all other modes of SDI operation are possible as described earlier in this manual. Auto-Fill is disabled. 2.9

When set, Auto-fill is enabled. Data in the Frame Buffer is interpreted as follows:

1. Color codes 1-15 represent colors in their Color Mapping RAM locations as in

normal nybble-mapped mode.

2. Code 0 has no color meaning; its presence signifies that the given pixel should take on the color of the pixel to its left.

DRAWING PICTURES IN AUTOMATIC-FILL MODE

The two rules for interpreting Auto-Fill data just given can lead to different systems for drawing pictures in this mode. Here we will describe a simple system for this purpose.

1. Define codes 0 and 1 to be black in Mapping RAM:

```
output    00--->80h
output    00--->81h
output    01--->80h
output    00--->81h
```

   Give codes 2-15 any desired color meaning.

2. Clear screen by filling Image Area with zeros and drawing a vertical line of 1's along left margin of Image Area.

   Note that the 1's on the left indicate that the given pixels are to be black and that the zeros indicate that the given pixels are to copy the color of the pixel to the left. The net result is a cleared screen.

3. To create a solid filled geometric object, draw the border on the left with the desired color code.

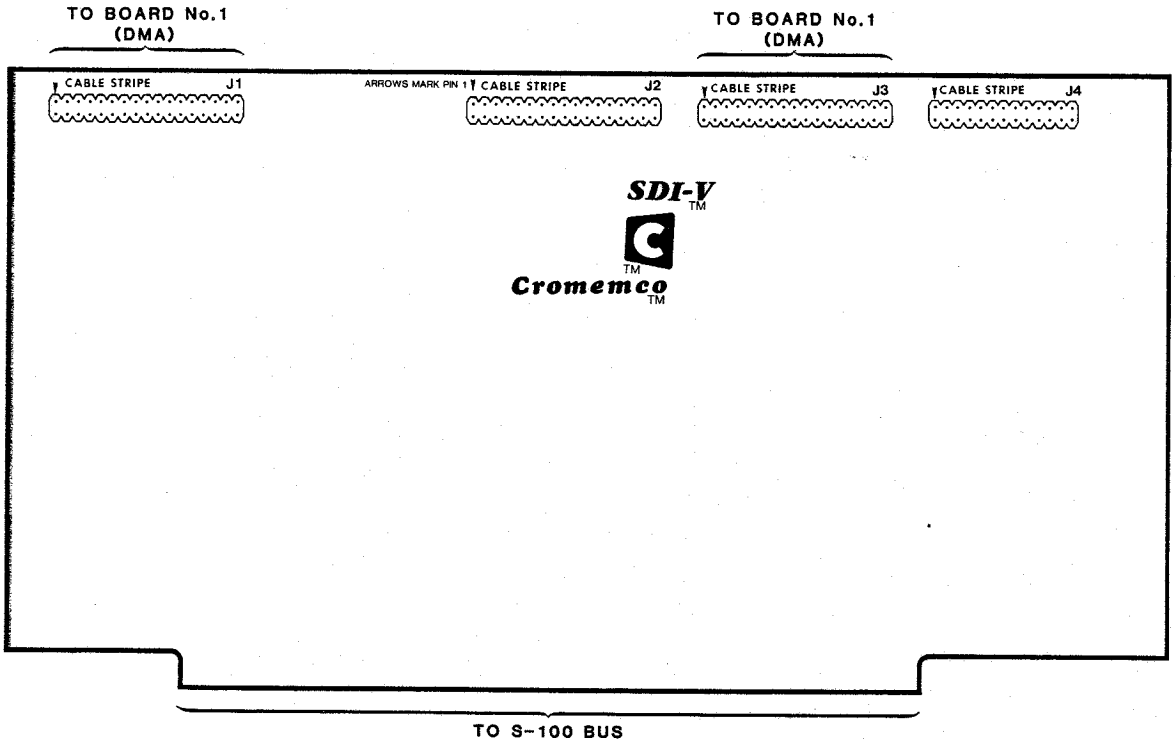
4. Complete the object by drawing 1's just to the right of the desired right hand border.

2.9 A NOTE ON BANK SELECT

When selecting bank 5 or 6 in preparation for writing into the Frame Buffer, only the low 16K (0-3FFFH) of host memory will remain visible to the CPU. For that reason, all write and read operations to the Two Port Frame Buffer should be stored in the low 16K of host memory.

If additional banks of host memory are to be used to store programs which will access the Frame Buffer, this memory should also appear in banks 5 and 6. The low 16K of these banks should be reserved for write and read operations.

3. EXTERNAL CONNECTIONS TO/FROM THE SDI



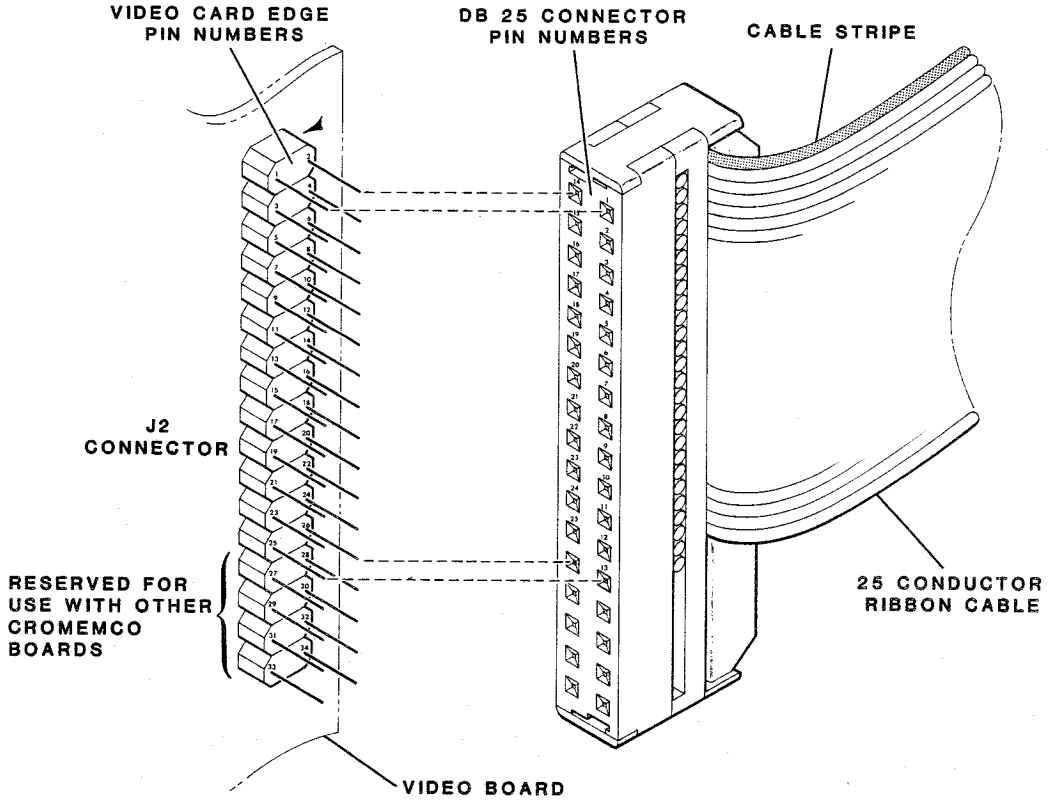
SDI Video Board External Connections

The video board contains provisions for 4 separate cable connections on the card edge. These pin groups, labelled J-1, J-2, J-3, and J-4 are shown in the illustration above.

The two pin groups J-1 and J-3 are for internal SDI communication and connect directly to the SDI board 1 (DMA board).

Connector J-2 is the main output to an RGB monitor. Connector J-4 contains both input and output signals for the SDI. If the SDI is to run on external composite video sync, for example, that signal should be input to the SDI via connector J-4.

The following pages contain detailed pinout descriptions of connectors J-2 and J-4. Notice that the SDI pin numbers differ from the standard DB-25 cable pin numbers which plug into these connectors. The correspondence between the SDI numbers and DB-25 numbers is shown in these pages.

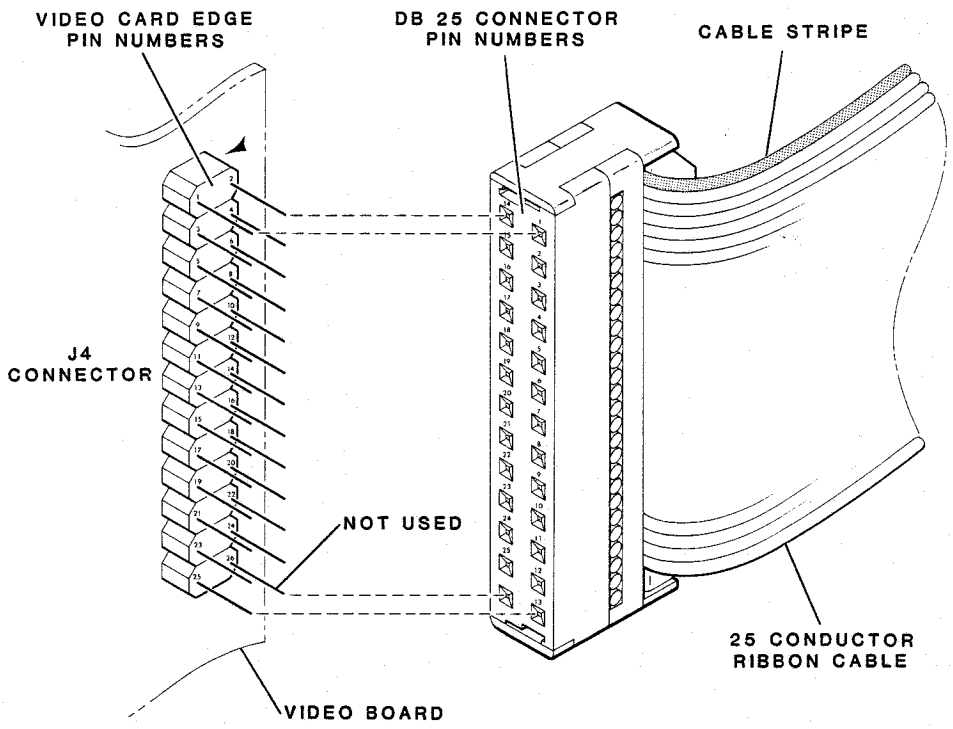


Video Connector J-2

DB-25 CONNECTOR NUMBERS	SDI VIDEO CARD EDGE NUMBER		DB-25 CONNECTOR NUMBERS	SDI VIDEO CARD EDGE NUMBER	
1	1	GND	14	2	RED VIDEO OUT (75 Ω)
2	3		15	4	GND
3	5		16	6	GREEN VIDEO OUT (75 Ω)
4	7		17	8	GND
5	9		18	10	BLUE VIDEO OUT (75 Ω)
6	11		19	12	GND
7	13		20	14	$\overline{\text{COMPOSIT SYNC}}$ OUT (75 Ω)
8	15		21	16	GND
9	17		22	18	KEY PULSE OUT* (75 Ω)
10	19		23	20	FOR USE WITH OTHER CROMEMCO PRODUCTS
11	21	FOR USE WITH OTHER CROMEMCO PRODUCTS	24	22	
12	23		25	24	
13	25		—	26	
—	27		—	28	
—	29		—	30	
—	31		—	32	
—	33		—	34	

* KEY PULSE OUT IS HIGH (2V INTO 75 Ω) IF COLOR CODE BEING DISPLAYED IS NON-ZERO. OTHERWISE IT IS LOW (GROUND).

Video Connector J-2 Pin Description



Video Connector J-4

DB-25 CONNECTOR	SDI VIDEO CARD EDGE		DB-25 CONNECTOR	SDI VIDEO CARD EDGE	
1	1	GND	14	2	COMPOSIT VIDEO/$\overline{\text{SYNC}}$ IN (75 Ω)
2	3	GND	15	4	N.C.
3	5	GND	16	6	$\overline{\text{COMPOSIT SYNC}}$ OUT (75 Ω)
4	7	GND	17	8	N.C.
5	9	GND	18	10	COLOR SUBCARRIER OUT (75 Ω)
6	11	GND	19	12	C PHASE OUT
7	13	GND	20	14	COLOR SUBCARRIER IN (75 Ω)
8	15	GND	21	16	C PHASE IN
9	17	GND	22	18	$\overline{\text{BLINKING}}$ OUT (75 Ω)
10	19	GND	23	20	$\overline{\text{H SYNC}}$ IN
11	21	GND	24	22	$\overline{\text{V SYNC}}$ IN
12	23	GND	25	24	$\overline{\text{ODD CLR}}$ IN
13	25	ODD OUT		26	N.C.

Video Connector J-4 Pin Description

VIDEO CONNECTOR J-4 PIN FUNCTIONS

Pin Function

1,3,5,...,23 **gnd** (ground)

25 **odd out**
This is a digital output square wave which is high during the odd (first) frame of each video field.

2 **composite video/sync in**
This analog input is for external sync lock of the SDI. It accepts a 75 ohm composite sync (-2V to -8V) or a composite video (+1V) signal. (Color subcarrier (pin 14) and color phase/color frequency adjustment is also necessary.)

4 **N.C.**
There is no connection to SDI circuitry.

6 **composite sync out**
This is the analog sync output. -2V to -8V into 75 ohm for external sync lock, for monitor drive use the output on connector J-2.

8 **N.C.**

10 **Color subcarrier out**
Analog 3.58 MHZ sine wave. 1-4V peak-to-peak into 75 ohm.

12 **Cphase out**
This is the digital color phase output, for sync locking two or more SDI systems. The SDI generating the sync must have a connection from this pin to the Cphase in (pin 16) of other SDI boards. (See figure.)

14 **color subcarrier in**
This is the analog input from a 75 ohm source which is used for sync locking SDI with an external source, 1-4V peak to peak. (Composite video/sync (pin 2) and color phase/color frequency adjustment is also necessary.)

16 **Cphase in**
This is the digital color phase input which is used for sync locking two or more SDI systems. (See comment on pin 12 and figure.)

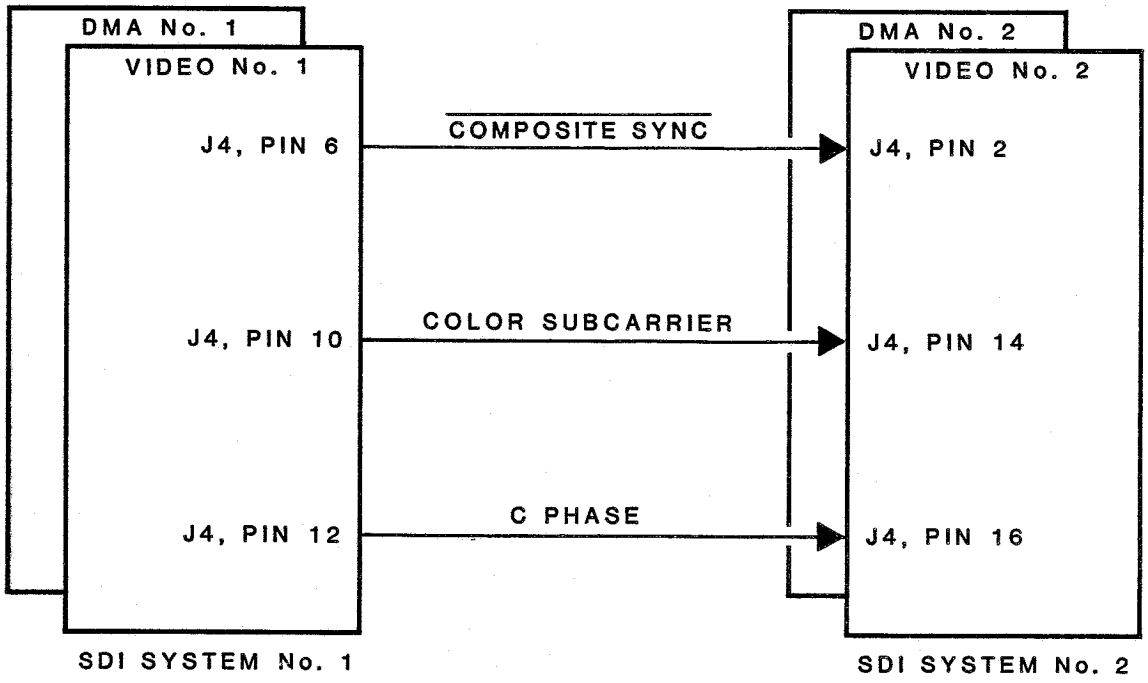
18 **blanking out**
Analog blanking output into 75 ohm load.

20 **HSYNC in**
Separate digital horizontal sync input.

22 **VSYNC in**
Separate digital vertical sync input.

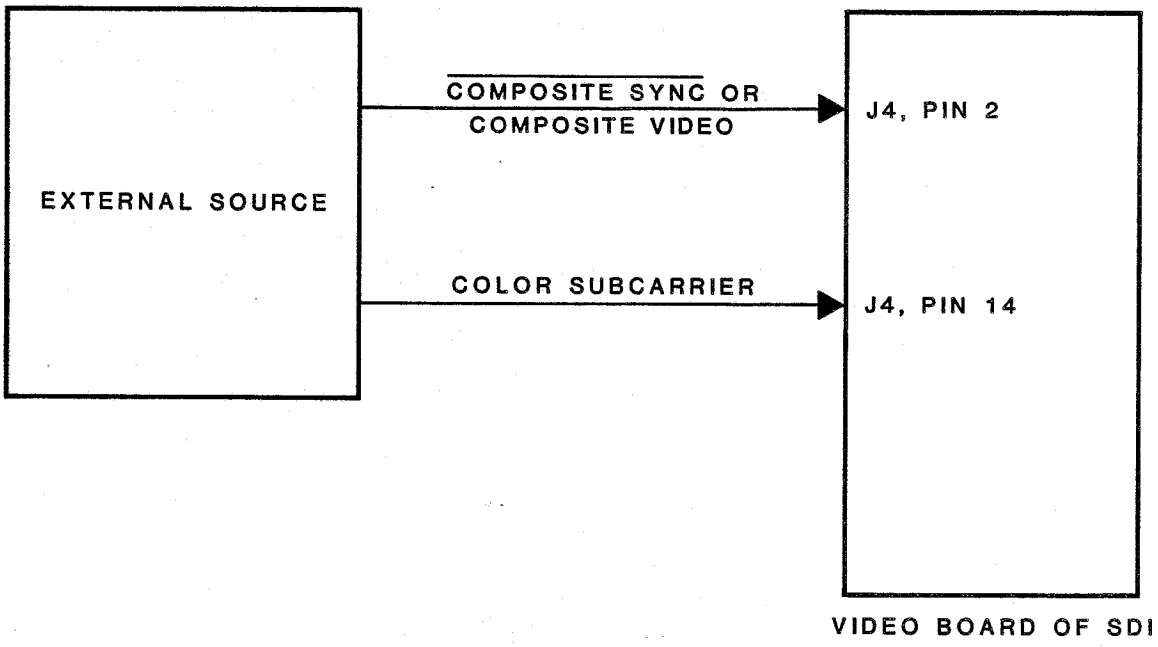
24 **ODDCLR in**

SYNC LOCKING TWO SDI SYSTEMS*



(a)

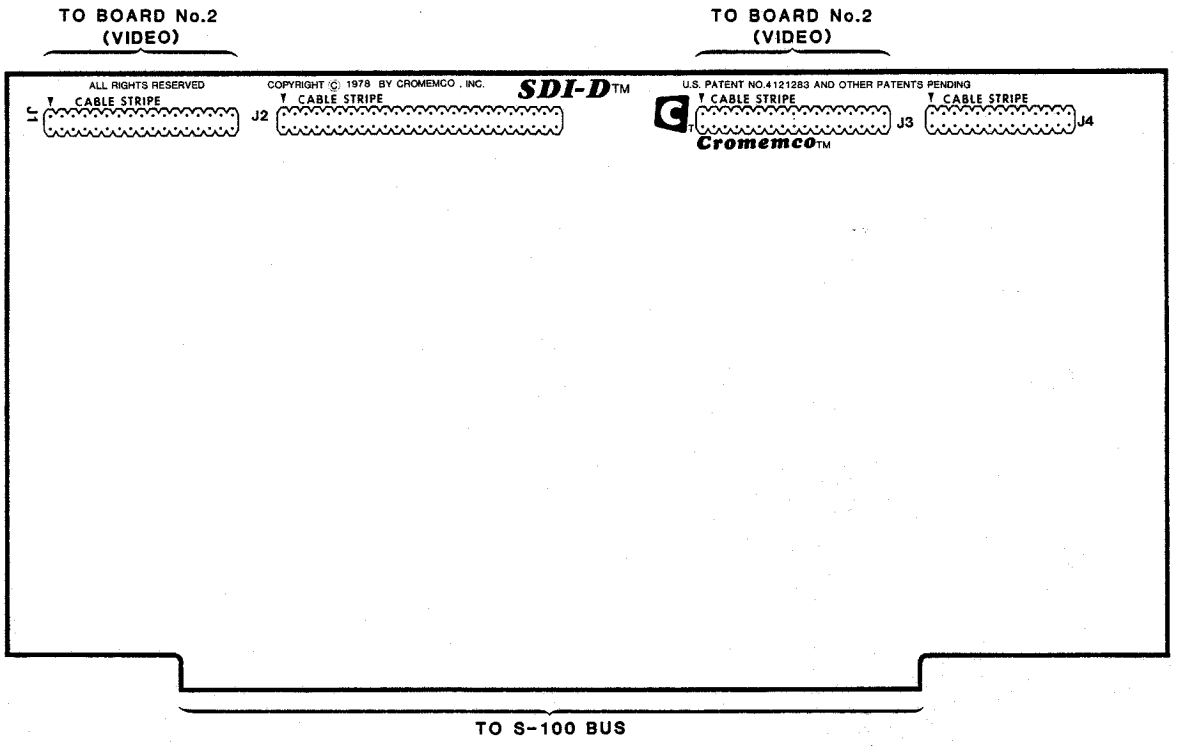
SYNC LOCKING SDI FROM EXTERNAL SOURCE*



(b)

* ADJUST COLOR PHASE FOR PROPER OUTPUT PHASE AND ADJUST COLOR FREQUENCY FOR 3.579545 MHz OF COLOR SUBCARRIER OUT AFTER SYSTEM RESET

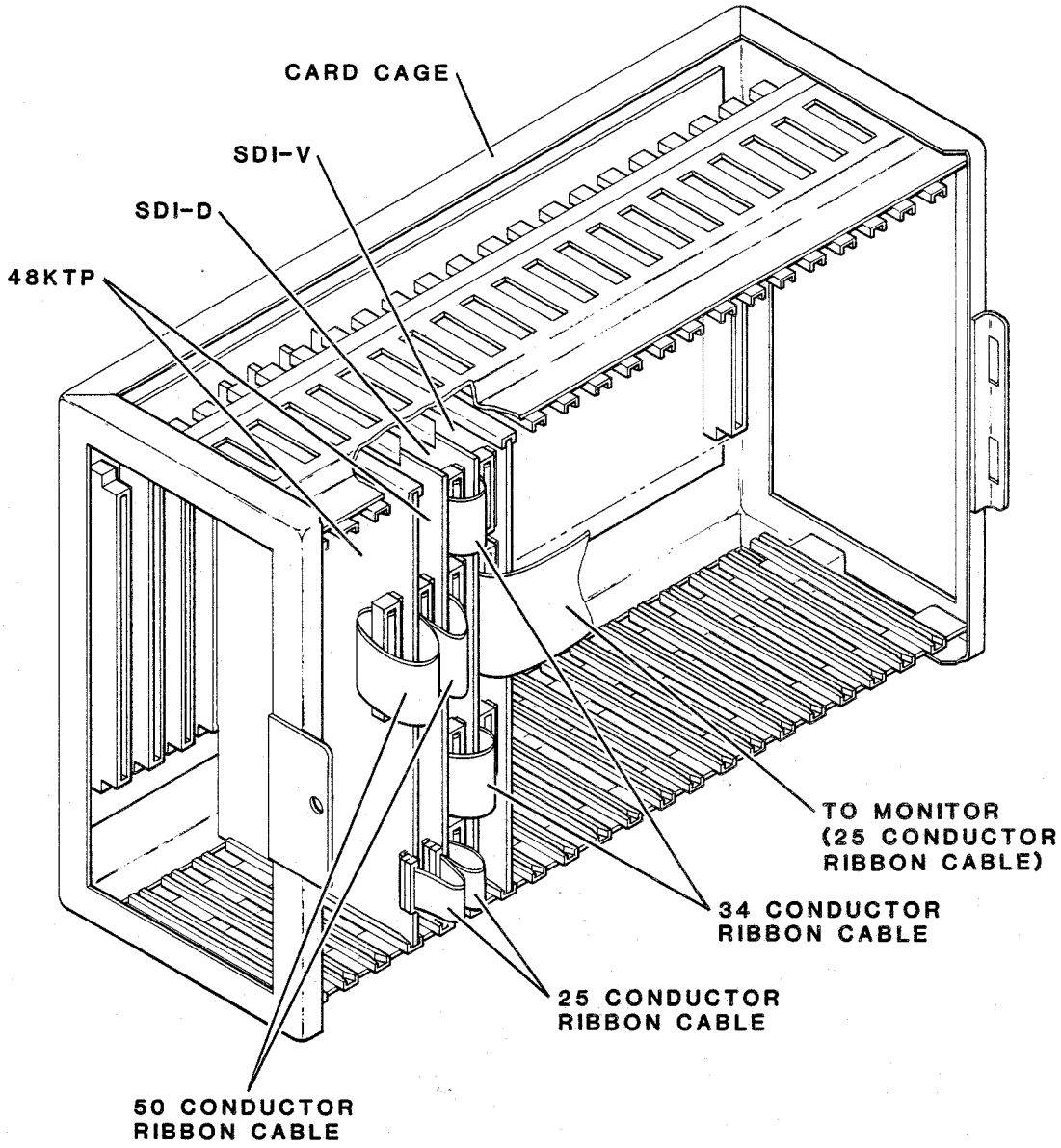
Using the SDI on External Sync



SDI DMA Board External Connections

The SDI DMA board has provisions for 4 separate cable connections on the card edge. Pin groups labelled J-1 and J-3 are for internal SDI communication and connect directly to the SDI board 2 (video board). Connector J-4 is used to communicate with other (optional) SDI boards.

Connector J-2 is used to connect the SDI and the second port of the special Two Port RAMs made by Cromemco for use with the SDI. When the SDI is reading from the second port of these RAMs the address and data signals are transported via a 50 conductor ribbon cable which is connected to this port. With 16KTP RAMs no other connection to the SDI is necessary. With 48KTP RAM an additional connection via a 25 conductor ribbon cable is necessary. This cable runs from connector J4 of the SDI DMA board to connector J2 of the 48KTP. See illustration and installation directions.



Connections Between the SDI and 48KTP

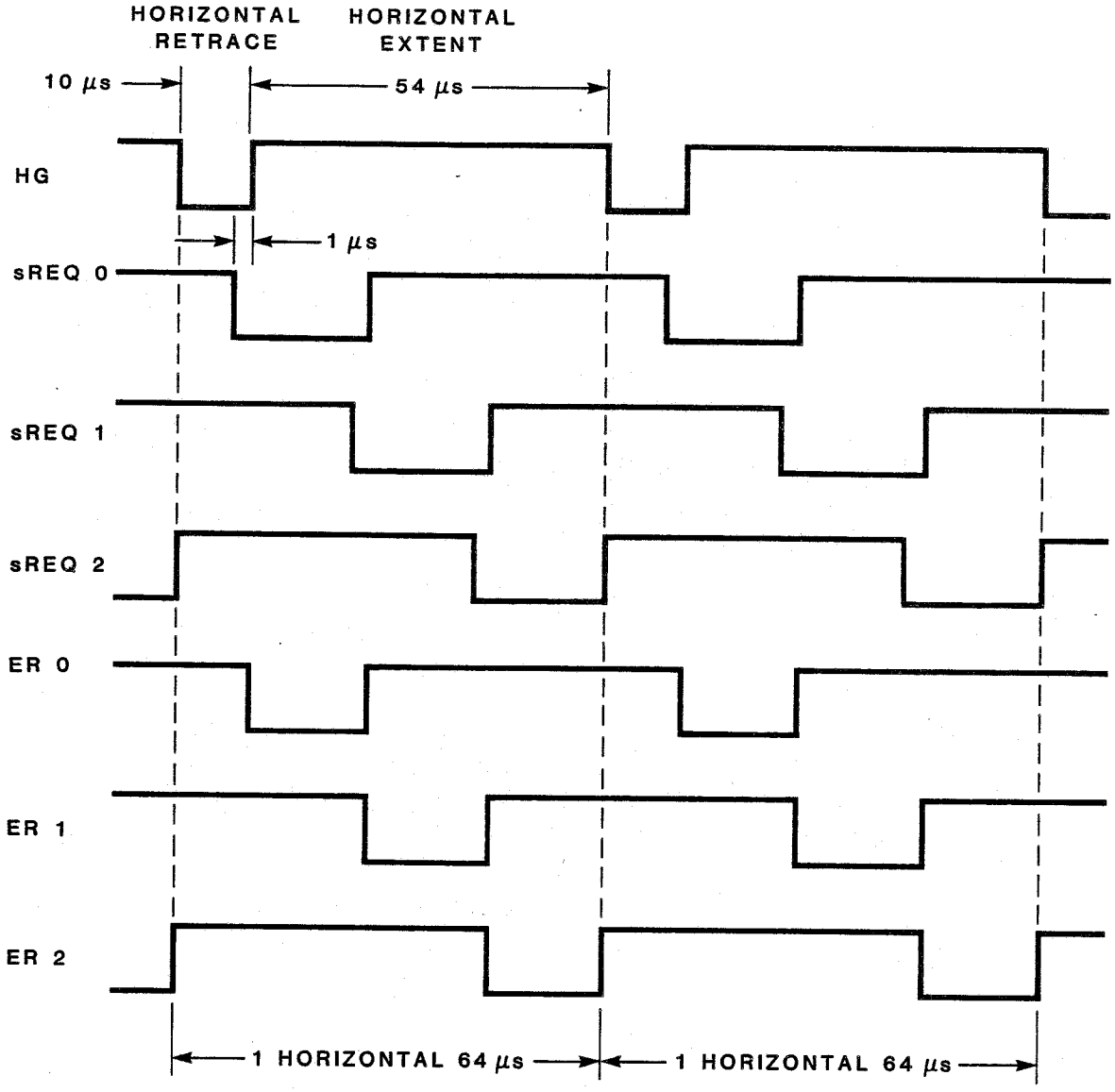
1.	GND	2.	
3.		4.	NO
5.	NO	6.	CONNECTION
7.	CONNECTION	8.	
9.		10.	GND
11.	EDI0	12.	EDI1
13.	EDI2	14.	EDI3
15.	EDI4	16.	EDI5
17.	EDI6	18.	EDI7
19.	GND	20.	EA0
21.	EA1	22.	EA2
23.	EA3	24.	EA4
25.	EA5	26.	EA6
27.	EA7	28.	EA8
29.	EA9	30.	EA10
31.	EA11	32.	EA12
33.	EA13	34.	GND
35.	SREQ0	36.	SREQ1
37.	SREQ2	38.	NO
39.	NO CONNECTION	40.	CONNECTION
41.	ER0	42.	ER1
43.	ER2	44.	ORRQ
45.	PG1	46.	ORACK
47.	NO CONNECTION	48.	NO CONNECTION
49.		50.	GND

DMA Connector J-2 Pin Description

DMA CONNECTOR J-2 PIN FUNCTIONS

Pin	Functions
11-18	EDI0-7 Second port data bus
20-33	EA0-13 Second port address bus. These lines are the same as address lines A0-A13. The memory page (i.e., the 16KTP being accessed) is determined through pins 35-37.
35-37	SREQ0-2 These low active lines select the 16KTP being accessed. They are decoded in that only one is active at any given time. (See figure.)
41-43	ER0-2 These low active lines enable second port read from the selected Two Port RAM. (See figure.)

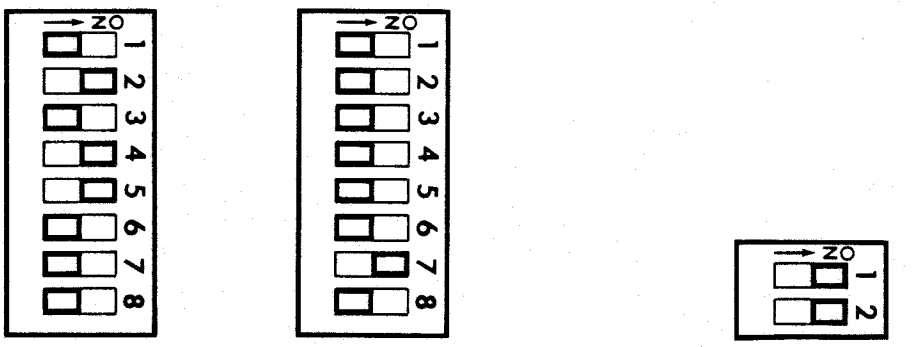
44,46 ORRQ, ORACK
These lines work in conjunction with SW-1 on the DMA board. If the override enable switch of SW-1 is on, then an override request on pin ORRQ will cause the SDI to temporarily stop driving the Two Port RAM.



Second Port Read Request Timing

4. INSTALLATION and SWITCH SETTINGS

In the majority of cases the switch positions for the SDI should be left unchanged from the factory settings. If they are inadvertantly moved, the correct settings can be found in the illustration below.



SW-1 SW-2 SW-1

DMA BOARD VIDEO BOARD

SDI Switch Settings

VIDEO BOARD SWITCH DESCRIPTION

SW-1 Bit 1 (setup)
- off: no boarder for active screen area.
- on: boarder for active screen area.

Bit 2 (composite sync)
- off: composite video sync appears on the composite sync output **only.**
- on: composite video sync appears on all 4 video outputs to the monitor.

DMA BOARD SWITCH DESCRIPTION

SW-1 Bit 1 (override enable)

- off: SDI ignores override requests to pin 44 of DMA connector J-2.
- on: override requests to pin 44 of DMA connector J-2 cause SDI to cease Two-Port drive.

Bits 2 and 3 (vertical timing): these bits can be used to synchronize the SDI to external broadcast equipment when long cables are used for connections. The lead/lag times for the various settings listed below are given in vertical units of 1-line/field.

	BIT 3	BIT 2
LAG 1	0	0
ON TIME	0	1
LEAD 1	1	0
LEAD 2	1	1

Bits 4,5,6,7 (horizontal timing): these bits are similar to the vertical timing bits above, but control lead/lag time in units of 270 nanoseconds each = 1/color frequency.

	BIT 7	BIT 6	BIT 5	BIT 4
LAG 3	0	0	0	0
LAG 2	0	0	0	1
LAG 1	0	0	1	0
ON TIME	0	0	1	1
LEAD 1	0	1	0	0
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
LEAD 12	1	1	1	1

Bit 8 No connection.

SW-2 Bit 1 (Memory):
off: for 64KZ host memory
on: for 16KZ host memory. Matching this switch to host memory is important only if S-100 DMA is used for picture scan. If second port read (DMA) is used, this bit should remain **off** at all times.

Bits 2-7 (Port Address): these bits determine the port location of the SDI. For normal operation they should remain at the factory settings.

Bit 8 No connection.

INSTALLATION

1. Main Memory
   Set bank select switches on main memory so that these cards appear in all banks. There is no memory conflict with the Two Port Memory boards as long as the phantom enable switch on the Two Port board is in the ON position.

2. SDI
   Copy switch settings as shown in illustration.

   Install the 2 SDI cards in adjacent slots and so that the video card, SDI-V, is within reach of the video cable leading to the monitor (or system back panel).

   Connect the 2 SDI cards with the 2 34-conductor ribbon cables supplied. These attach to the cards at connectors J-1 and J-3.

   Connect the video cable (from monitor or back panel) to connector J-2 of video card SDI-V. This should be a **25** conductor ribbon cable attached to a **34** pin edge connector.

   Note that the Cromemco RGB Monitor is provided with a video input cable terminated by a DB-25P connector to mate with the ribbon cable supplied with the SDI.

3. 48KTP (if present)
   Copy switch settings from illustration. (If only 1 48KTP is present, address it as page 0, bank 5.)

   Insert one (or both) 48KTP cards directly adjacent to the DMA board SDI-D.

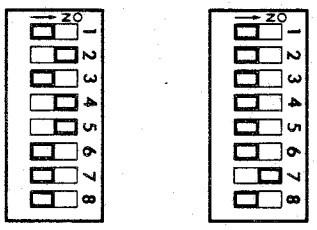
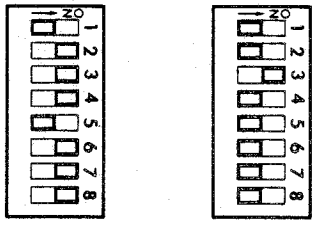
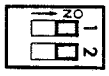
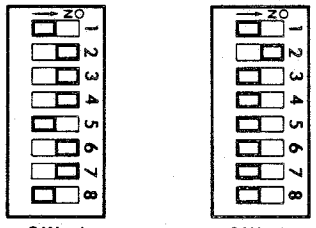
   Connect both 48KTP cards to the SDI-D with 50 conductor ribbon cable to the J-2 edge connector of the SDI.

Connect both 48KTP boards to the SDI-D with 26 conductor ribbon cable to the J-4 edge connector of the SDI.

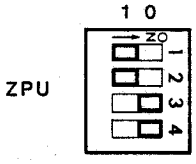
4. 16KTP (if present)
Copy switch settings from illustration (you will need 3 16KTP cards for each 48K page of memory).

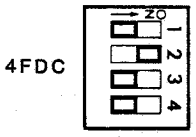
Insert cards adjacent to each other and the DMA card, SDI-D.

Connect all 16KTP cards to the SDI-D with the 50 conductor ribbon cable to the J-2 edge connector of the SDI.

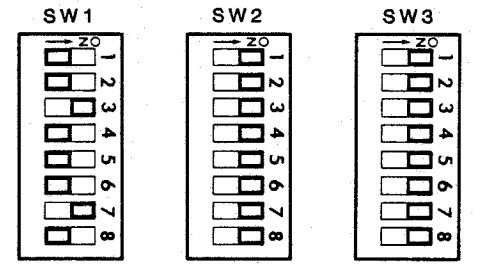
<table>
<tr><th colspan="4">SWITCH SETTINGS</th></tr>
<tr><th>64KZ OR 16KZ</th><th>SDI</th><th>48KTP</th><th>16KTP</th></tr>
<tr><td rowspan="2">(ALL BANKS ON)</td><td>DMA
 
SW-1 SW-2</td><td>PAGE 0 (BANK 5)
 
SW-1 SW-2</td><td rowspan="2">(SEE NEXT PAGE)</td></tr>
<tr><td>VIDEO
  </td><td>PAGE 1 (BANK 6)
 
SW-1 SW-2</td></tr>
</table>

Switch Settings 1

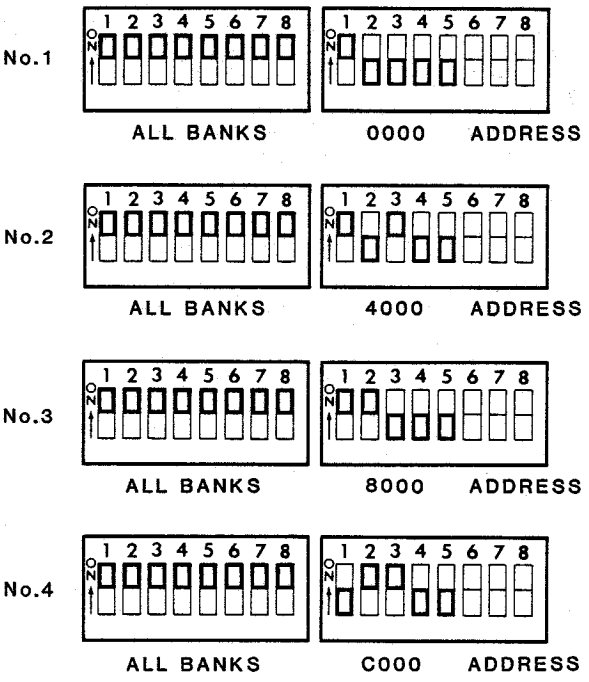




IF 64KZ PRESENT:



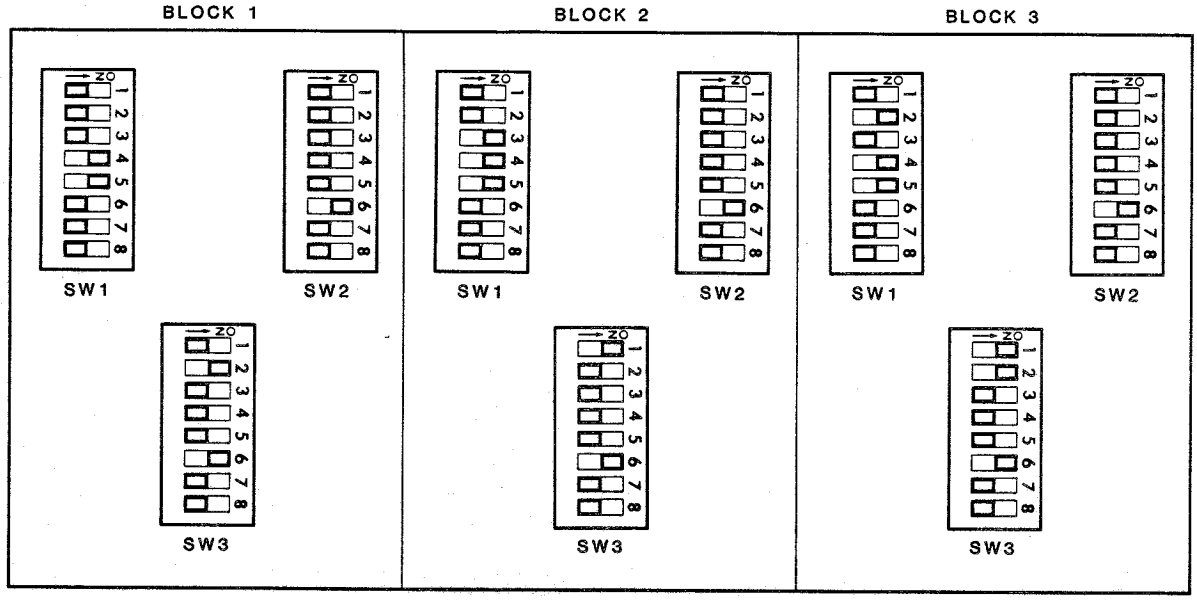
IF 16KZ'S ARE PRESENT:



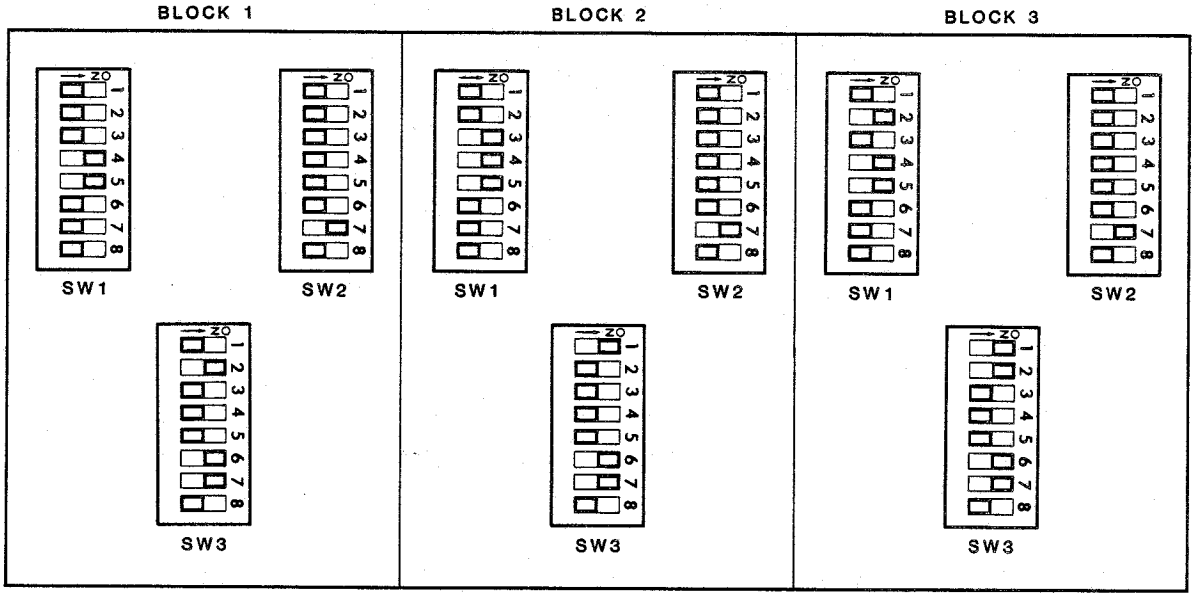
Switch Settings 2

SWITCH SETTINGS ON 16KTP BOARDS WHEN USED WITH CROMEMCO SDI

PAGE 0

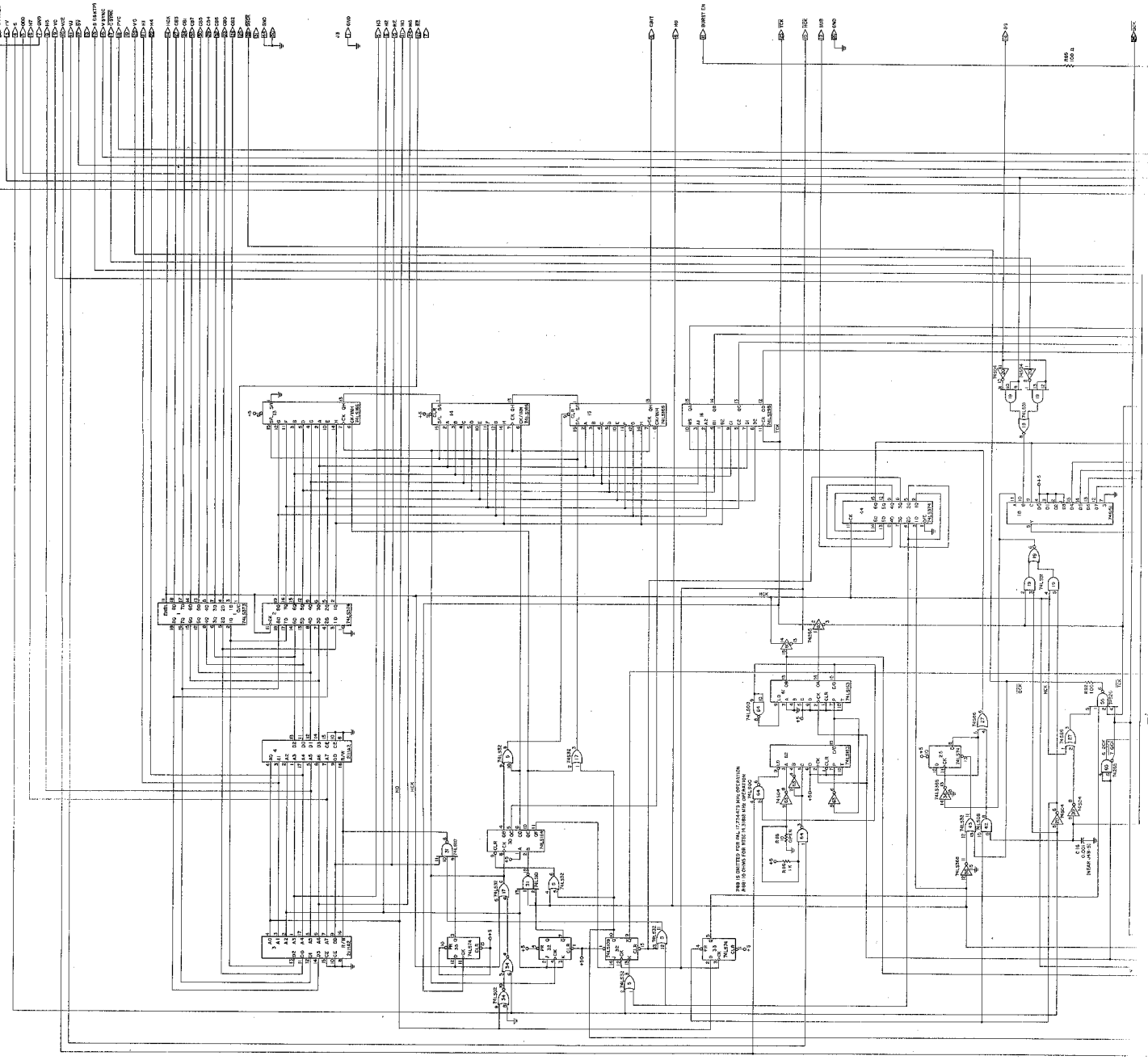


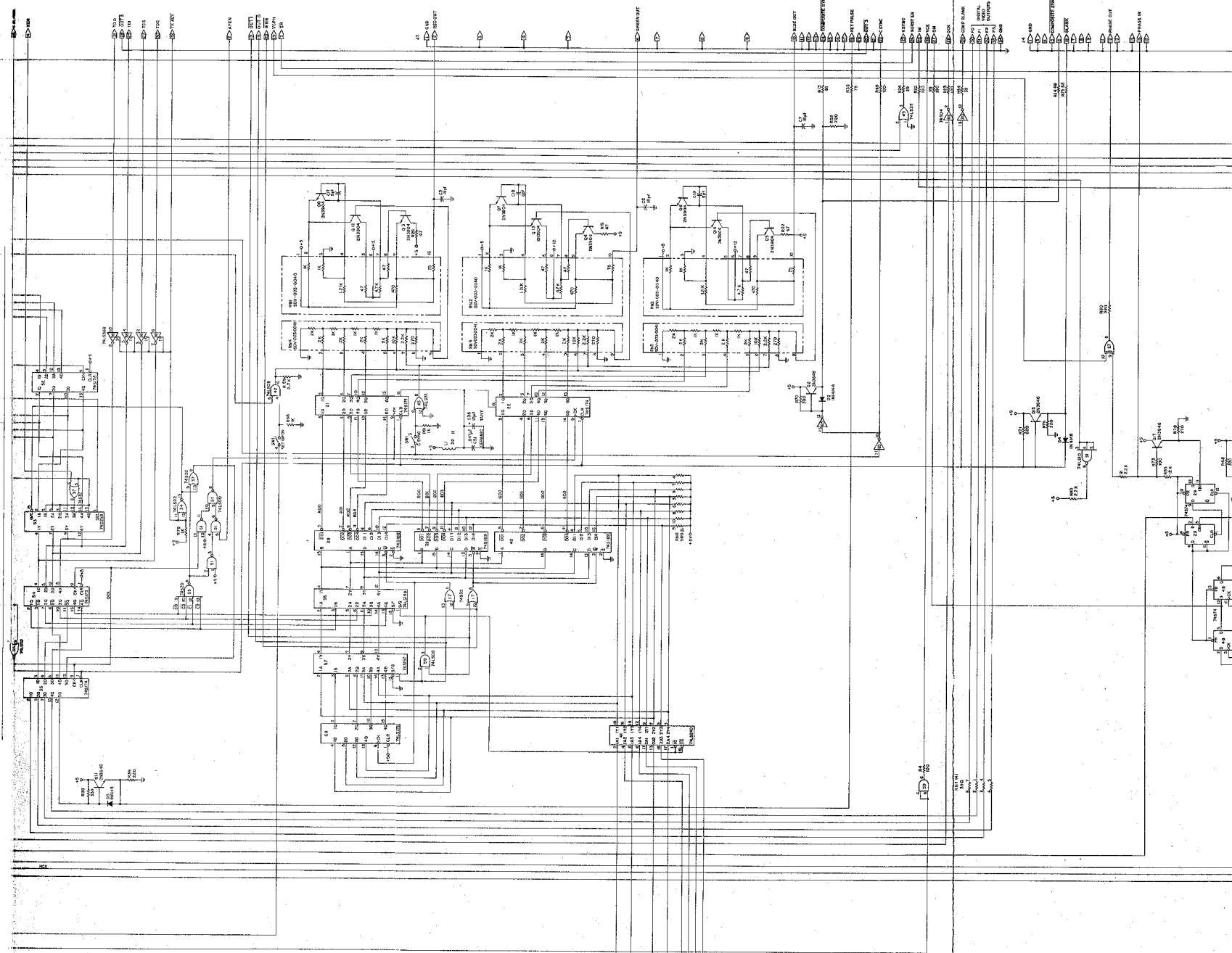
PAGE 1

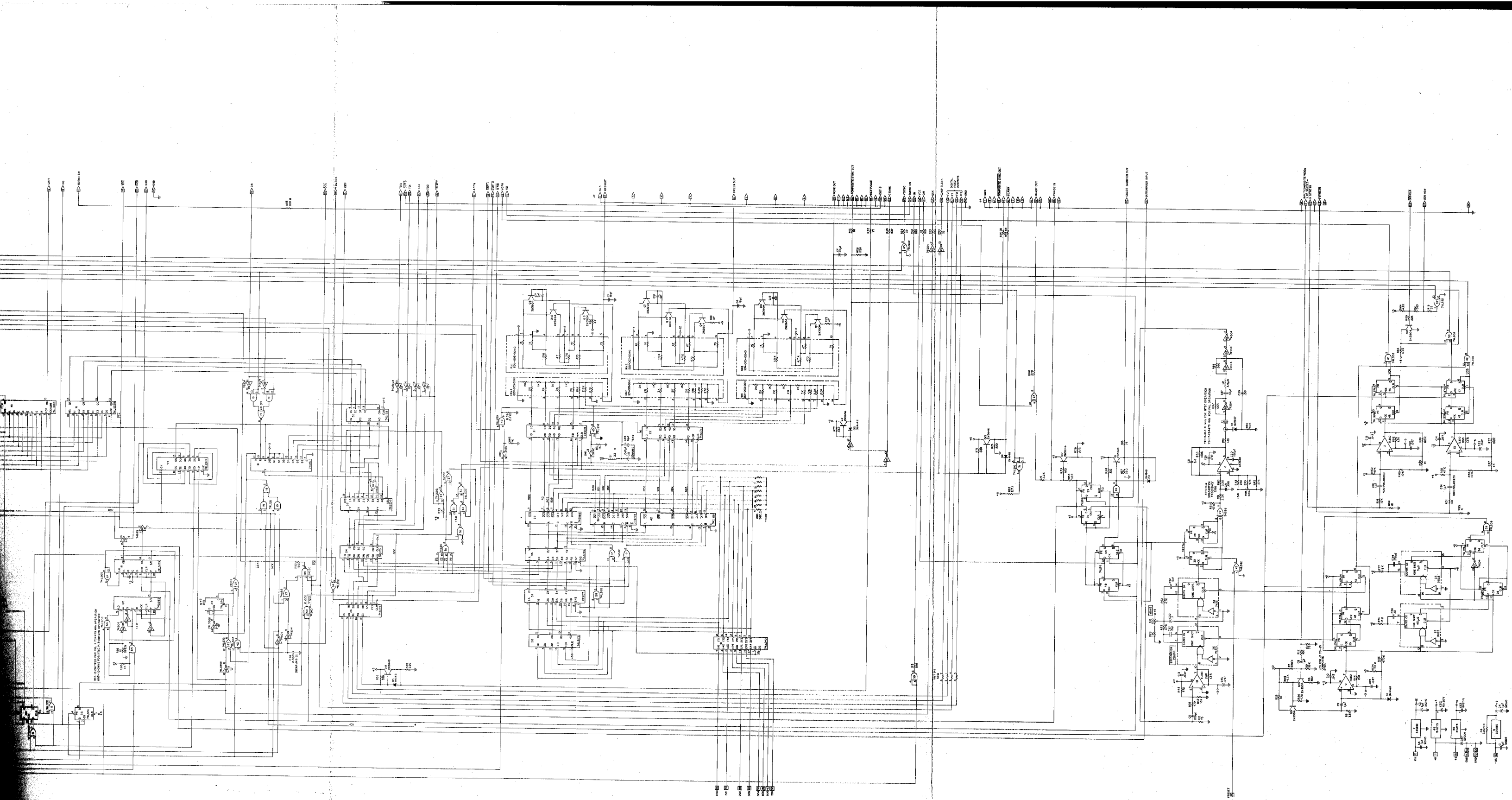


Cromemco SDI - Video PCB

BOARD REV.D
SCHEMATIC REV. 7, 6/20/80
TOLERANCE +/- 5%
THIS SCHEMATIC CONTAINS CIRCUITS COVERED BY PENDING U.S. PATENTS







Cromemco SDI™ -DMA Control

BOARD REV. D

SCHEMATIC REV. 4 1/8/80

SDI-D-040-0030

U.S. PATENT PENDING

REFERENCE SYMBOLS

ASSIGNED	NOT ASSIGNED
IC1-IC58	
SW1-SW2	
C1-C3, C23-C26, C32-C34	C4-C22, C27-C31
R1-R11, R13	R12
RN2-RN9	
Q1