A Versatile I/O System

For a

Real Time Image Processor.

by

Sanjay Adkar

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

APPROVED:

M. Nadler, Chairman

J. R. Armstrong

J. G. Tront

August, 1986

Blacksburg, Virginia

A Versatile I/O System

For a

Real Time Image Processor.

by

Sanjay Adkar

M. Nadler, Chairman

Electrical Engineering

(ABSTRACT)

A versatile I/O system for a real time image processor and a complex clocking circuit for the I/O system and the image processor have been designed. The I/O system receives data from an arbitrary video source. These data are digitized and conditioned to be compatible with the image processor. The image processor output is conditioned such that these data can be displayed on a standard RS 170 2:1 video monitor. Variable frame rate reduction circuits and bit reduction techniques such as line, column and dot interlace are incorporated during output conditioning. Experiments on reducing the frame rate and bit rate of a processed image can be carried out using this I/O system.

ACKNOWLEDGEMENTS

This work is dedicated to my father-in-law ,
and to my wife who are a constant source of encouragement and support.

I would like to thank Dr. M. Nadler, my committee chairman and advisor for his invaluable guidance and encouragement throughout this project. This work has greatly benefited from our many thought provoking discussions.

Finally, I would like to thank Dr. Armstrong and Dr. Tront for kindly consenting to be on my committee.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1.0 INTRODUCTION

1.1 PROBLEM DEFINITION

There is a need for a flexible real time I/O system which would interface between a real time Image Processor and arbitrary real time I/O devices (e.g. T.V cameras, CCD cameras, Monitor etc). This is the general problem from which this research has evolved.

The Telesign project undertaken at Virginia Tech has precisely the above requirements. The Telesign system is described in [1], [2], [3]. A conceptual block diagram in its present state of evolution is described in Figure 1 on page 2. This thesis describes the design of the input buffer, output buffer, and system clock for the Telesign system. These systems used together would indeed constitute a variable frame rate I/O system with real-time post-processing capabilities.

The IPB is the real-time reconfigurable image processor. The IPB initialization and selection of the reconfigurable parameters is controlled by the system PC. The LINK is the hardware interface between the Telesign system and the PC.

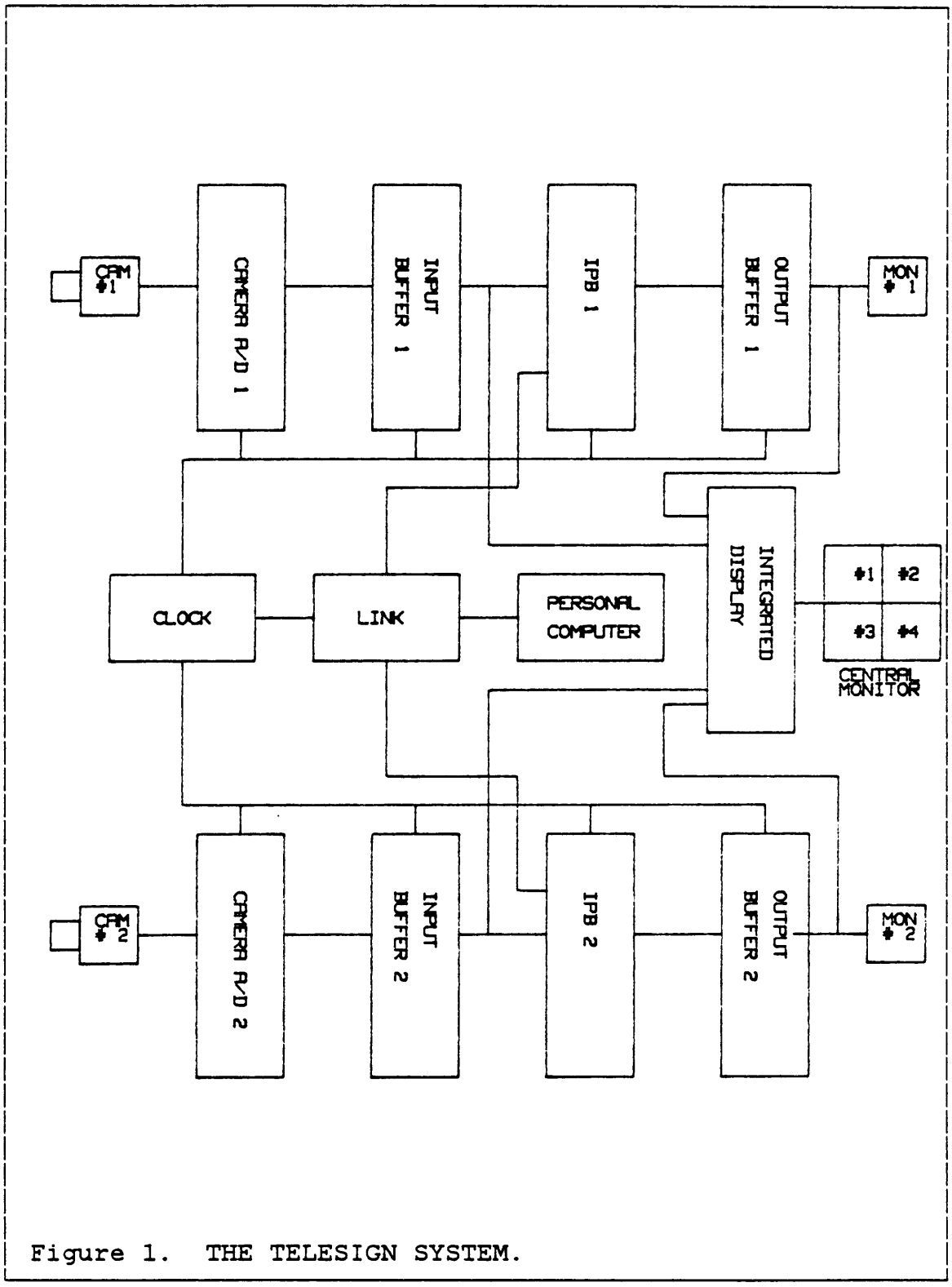


Figure 1. THE TELESIGN SYSTEM.

1.2 SCOPE & OBJECTIVE OF THESIS

1.2.1 INPUT REQUIREMENTS

The input requirements are obviously the input requirements of the Telesign system. Since the final objective of the Telesign system is to provide effective low cost visual communication for the deaf, it is highly desirable that the following input requirements be strictly met.

1. Ability of the input buffer to accept data in real time from a standard low cost camera operating in the RS 170 2:1 interlaced mode.

2. Any other RS 170 source, such as VCR or a video disc player, etc.

3. CCD camera source operating at a random frame rate in a non-interlaced mode.

In effect the input buffer should be able to accept almost all commercially available input sources.

It is assumed here that the input buffer receives a digitized information stream from these sources with 'start of frame' (SOF) and 'start of line' (SOL) pulses to allow a spatial

reference for the entire system. A front-end A/D converter is being separately fabricated for this purpose, and is not included within the scope of this thesis.

An additional requirement, rather than a limitation, is the fact that for proper working of the image processor the input frame rate should not be less than the effective frame rate of the image processor. The need for this will be apparent in Chapter 2.1.1. In general the input buffer should effectively lower the frame rate to match the speed of the image processor.

1.2.2 OUTPUT REQUIREMENTS

It is extremely desirable for any image processor system to have its output displayed on a standard T.V monitor and at the same time ensure good quality pictures.

This implies that the output buffer should be able to effectively increase the frame rate to meet the RS 170 2:1 standards as required by the monitor. (Note: Image refresh rate cannot increase beyond the effective frame rate of the image processor.)

One of the major objectives of the Telesign system is to experiment with the refresh rates and image resolution, and

experimentally derive a minimum effective data transfer rate for an acceptable Telesign image sequence. Hence it follows that output buffer (input buffer as well) should have the ability to regulate the refresh rate. Additionally the output buffer should able to postprocess the Image Processor Board (IPB) output image to allow row, column, and dot interlace formats that are compatible with the RS 170 2:1 interlace format.

Finally, since most image processors (IPB is no exception) operate on a square image, in the case of the Telesign 256*256 image (i.e., 256 pixels & 256 lines) the output buffer should be able to compose the final video image for a distortion-free image on a 4:3 video monitor.

Thus the output buffer has the following requirements

1. Should increase frame rate to that required by RS 170 standards (30 frames/sec.)

2. Should be able to vary the refresh rate from IPB frame rate downwards.

3. Should be able to post-process image in real time to allow row, column, & dot interlaced formats that are compatible with RS 170 2:1 intelace format.

4. Should be able to display a square image on a standard video screen (T.V) with a 4:3 aspect ratio.

1.2.3 CLOCKING REQUIREMENTS

The IPB, input buffer, output buffer, and in general all other subsystems of the Telesign project require complex clocking pulses. It is also the objective of this work to design the system clock that will satisfy the clocking requirements of all subsystems.

2.0 INPUT SIGNAL CONDITIONING.

2.1 PURPOSE AND OBJECTIVE OF INPUT BUFFER.

The objective of the input buffer has already been briefly described in section 1.2.1. Here we attempt to explain more fully this objective and also to highlight the importance of this requirement. The entire purpose of the input buffer will become clearer if we keep in mind the current objective of the Telesign System.

1. To reduce the effective data rate of the system to a minimum, but at the same time produce images acceptable to the sign language users of the system.

2. Given the complex nature of the Image Processor Board (IPB) it would be desirable that it be operated at a constant data rate.

It is apparent that the above Telesign objectives appear contradictory; not only that, it is also obvious that as a system it would be absolutely essential to be able to experiment with the effective data rate of the system to enable us to find the minimum acceptable data rate. The above contradiction has to be resolved by the input and output buff-

ers. Figure 2 on page 9 indicates the changes in the data rate and the frame rate from the camera through to the IPB input.

Thus to summarise, the objectives of the input buffer are the following;

1. To condition digitized video into video compatible to the slower but real time image processor.

2. To regulate the effective frame rate (refresh rate) seen by the image processor and thus eventually by the output video monitor to evaluate minimum frame rate for an acceptable processed image. The refresh rate should be variable and should be controlled in real time from an external device such as the PC used in the Telesign system. The refresh rate is the rate at which new data is available to the IPB. It is also called the 'effective frame rate'.

The image processor used in the Telesign system can process 256 * 256 pixel images at the frame rate of about 25 f/s. With this fact in mind, we generated the IPB pixel clock of 2.165 MHz (refer to Chap.4 for choice of frequency, etc.). The IPB is a pipeline processor which requires 8 clock pulses to process a pixel [2],[3]. Thus the IPB clock is 2.0165 * 8

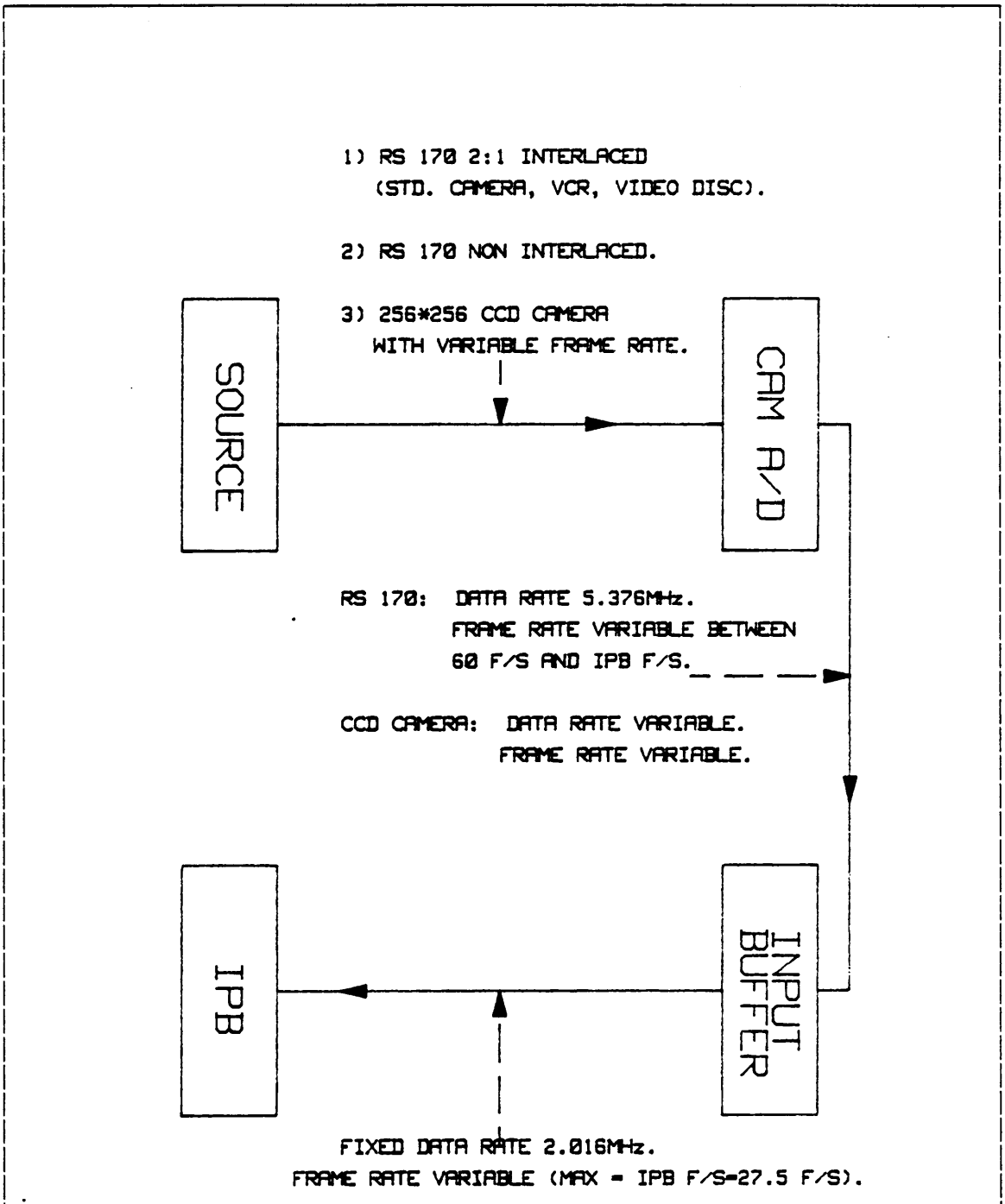


Figure 2. DATA AND FRAME RATES THROUGH INPUT BUFFER.

MHz = 16.132 MHz. Further, if we assume a dead time between lines of 16 pixels and a dead time between frames of 16 lines we can derive the exact frame rate of the IPB as follows:

Frame Rate = pixel clock/(no. of pixels per line * no. of lines perframe) (Note: These include the pixels in the dead time betweenn lines and frames).

Frame Rate = 2.0165 MHz / (272 * 272) = 27.255 f/s.

As explained later in chap. 4 it is possible to change the dead time between lines and frames and hence change the effective frame rate of the IPB without actually changing the pixel clock. However, it may be noted that only minor changes in the frame rate can be accomplished by using this technique and hence should be used only for fine tuning rather than for coarse changes in the frame rate.

For example, in the Telesign system using a standard video camera, we have an input frame rate of 60f/s(fields per sec.) and an A/D converter with data rate of 5.736 MHz (cf. Chap.4 for choice of frequencies). It is important to note here that this choice of frequency for the A/D converter provides us with more than 256 pixels per line(only 256 pixels will be stored per line). The input buffer should now be able to output data to the IPB at a data rate of 2.0165 MHz and, al-

lowing for dead times described above, a frame rate of 27.255 f/s. will be presented to the IPB. The conceptual realization of this transformation is explained in the next section.

2.2 CONCEPTUAL REALIZATION OF THE OBJECTIVE.

Before we get into the actual details of how we can solve the above problem, we must realise that the Telesign requirement of a fixed data clock for the IPB automatically restricts us to a constant frame rate for the IPB (the dead time between lines and frames is fixed at some predetermined value). The question that arises as an outcome of this: is how do we change the effective frame rate of the system? The answer, though not so obvious, is quite simple; we can determine the effective frame rate of the system by refreshing an IPB frame at that rate. This refresh is absolutely independent of the data rate to the IPB. This updated frame would manifest itself on the TV screen and the effective frame rate of the system would be the refresh rate of the frame to the IPB.

From the above it is clear to us that if we are to achieve the objectives of the input buffer we need to be able to store an entire image frame in real time at the incoming clock rate from the camera A/D converter (5.736 MHz in case of standard RS 170 video). Also we should be able to transmit the previous frame simultaneously to the IPB at a slower rate (2.0165

MHz in case of the Telesign system). In addition to this we should be able to update the frame tramsmitted to the IPB at the desired effective frame rate.

Since a standard image procesor normally handles 256 pixels per line and 256 lines per frame, assuming that each pixel is made up of 8 bits to allow full grey scale resolution a standard frame would require 256 *256 *8 (64K*8) bits of memory. A 256*256 organisation would be desirable since each line can be distinctly identified by a row of such a memory. Eight such memories would be required to store an entire frame, allowing for 256 different shades of grey. Texas Instrument's TMS 4161 dual port video DRAM's 64k(256 *256) are ideal for this application. The memory access time is 150ns and thus is able to handle the 5.7367MHz clock without problems [4]. The serial I/O port of this memory is of particular interest to us, as shall be clear in the next section.
Figure 3 on page 13 shows the structure of the TMS 4161 dual port memory. Note that for our application the data would be shifted in and out of the memory through the 256 bit serial shift register. Thus a line by line transfer of data is achieved between the actual memory and the shift register. The shift register data will be transferred to the appropriate row during the dead time between the lines. Figure 4 on page 14 outlines the way rows of a video frame are shifted

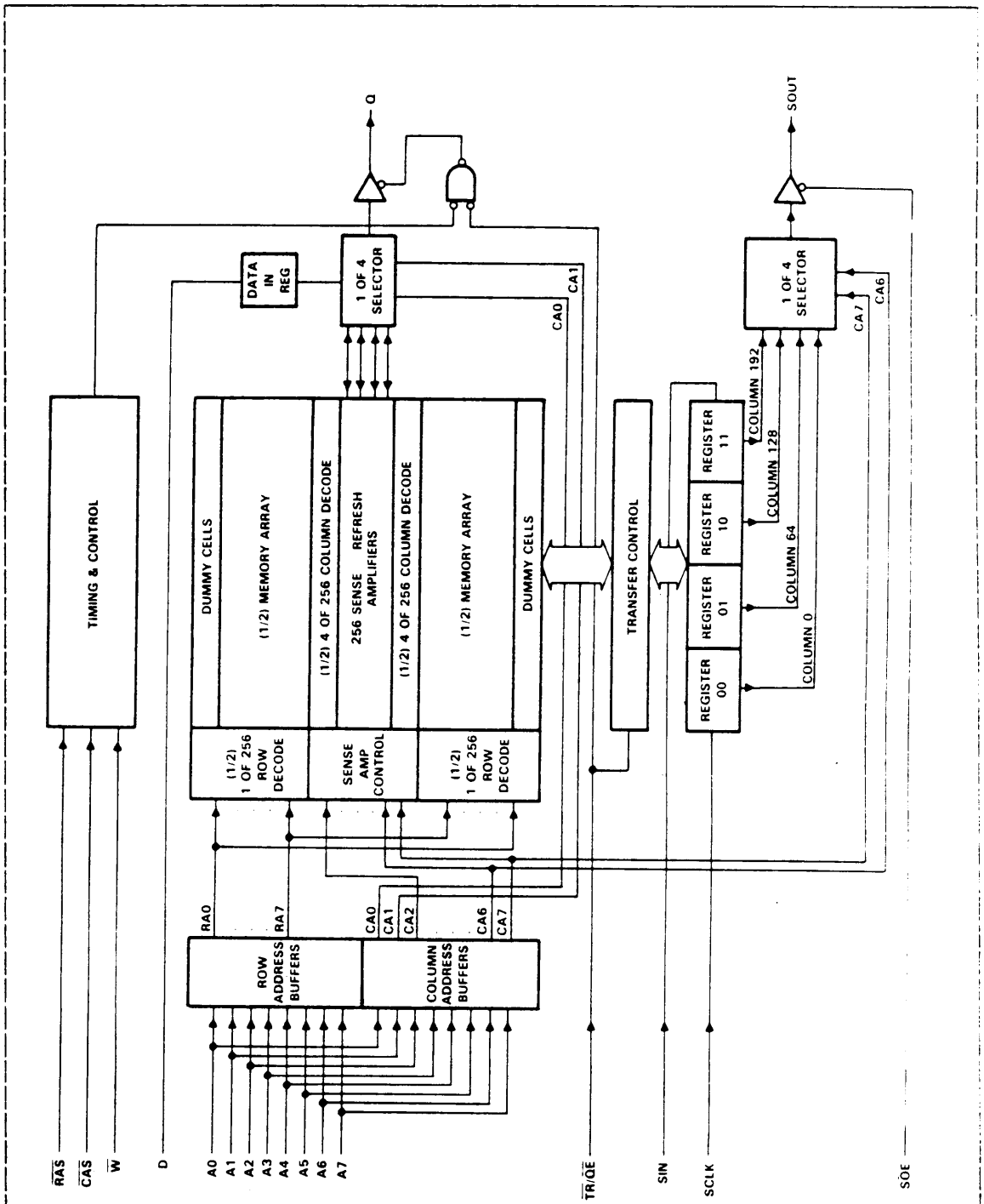


Figure 3. STUCTURE OF TMS 4161 DUAL PORT MEMORY.

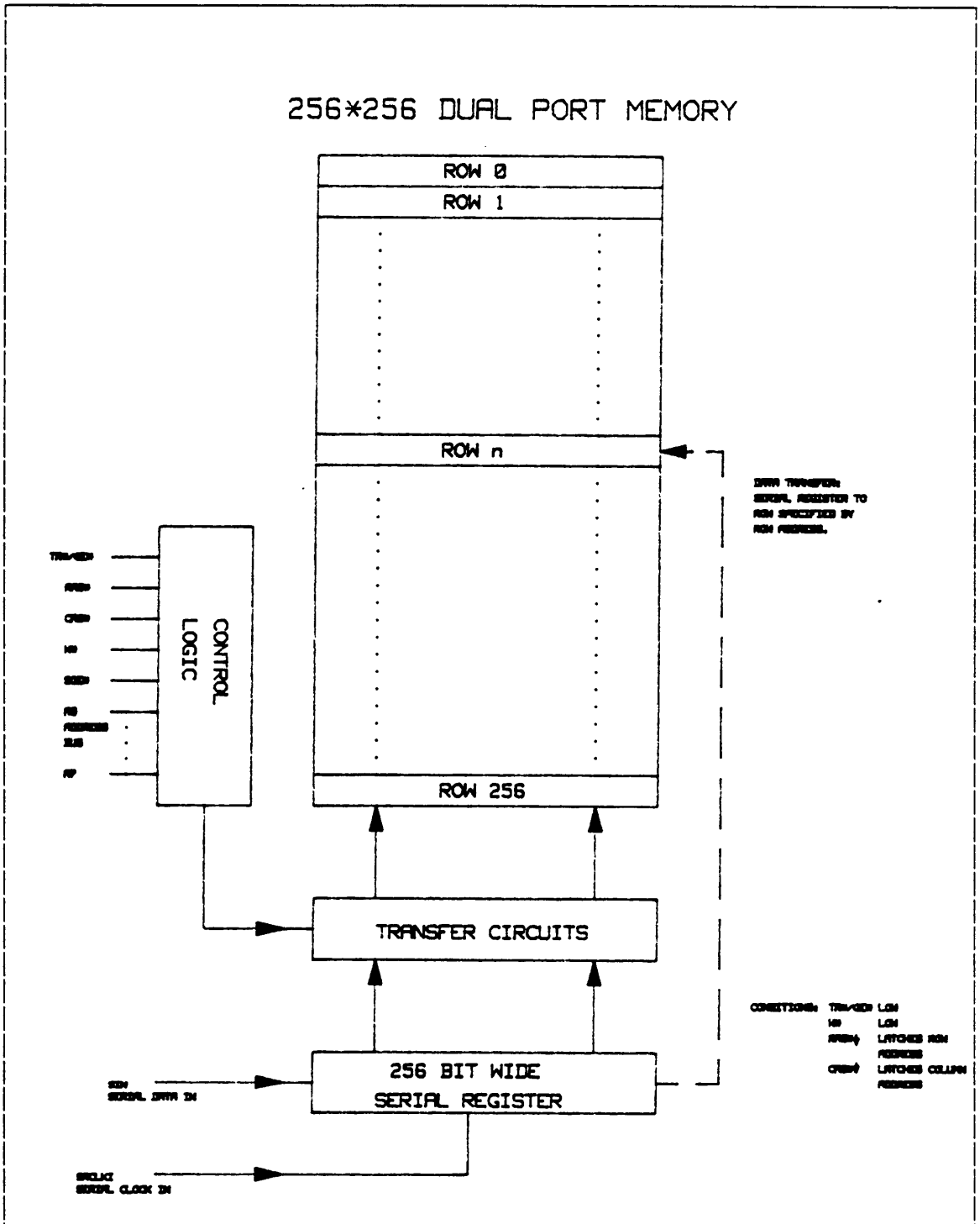


Figure 4. SHIFT REGISTER TO MEMORY TRANSFER.

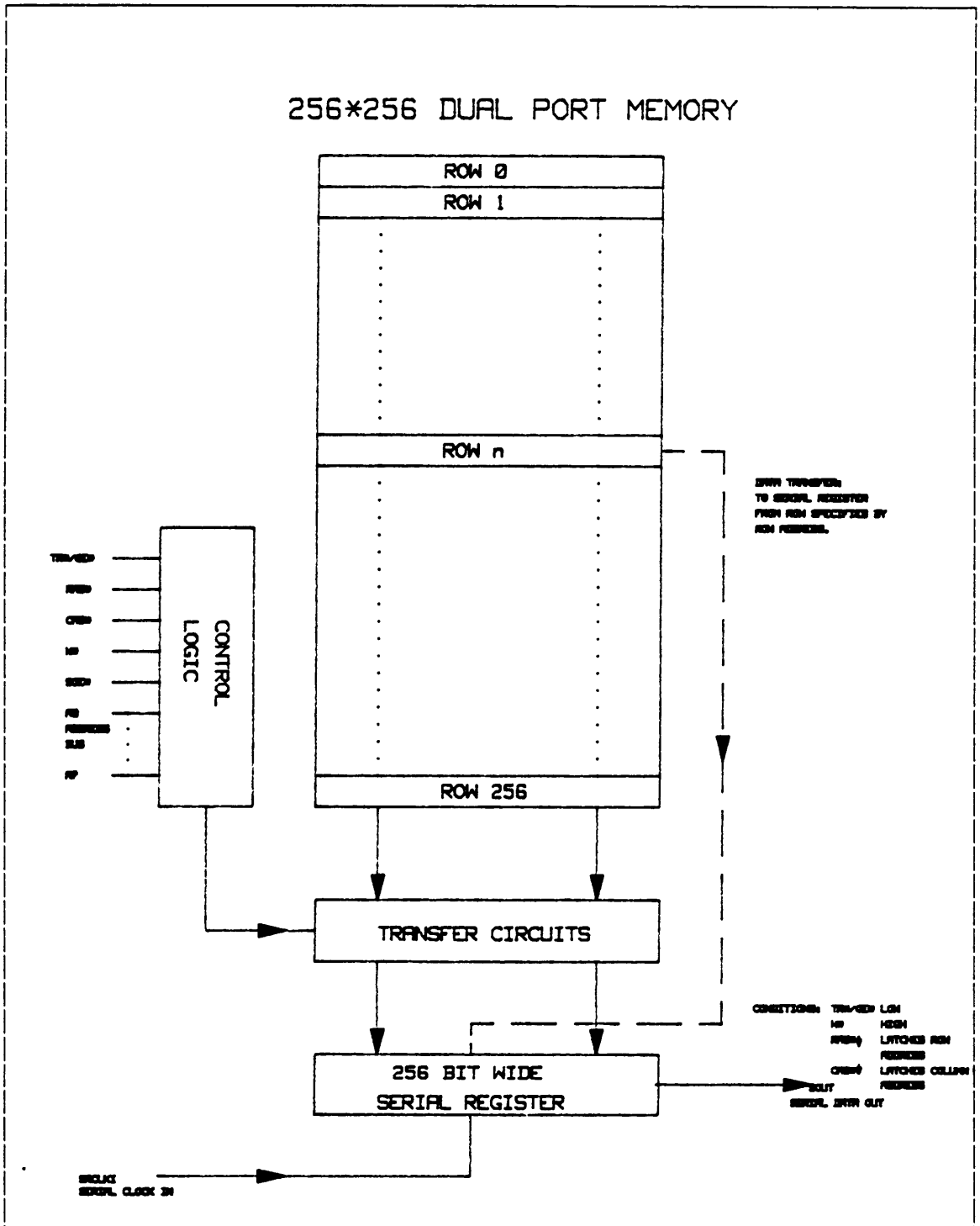


Figure 5. MEMORY TO SHIFT REGISTER TRANSFER.

into memory. Figure 5 on page 15 outlines the way rows of a video frame are shifted out from the memory.

The next step will be to use the above memories, so as to realise the objective of the input buffer. For convenience, let us consider just one bit resolution per pixel in all our future discussions. Thus we need only consider one TMS 4161 per frame.

Consider two such memories and consider also that at any given time one of these is configured in the input mode while the other is configured in the output mode. The memory in the input mode continuously receives fresh video data from the camera at the standard video rate. If at the end of one input frame the memory is still configured in the input mode, fresh data from the next input frame will overwrite the present frame data. This insures that at any given time the memory configured in the input mode has the most current data. Simultaneously the memory configured in the output mode will be writing its data out to the IPB at a slower rate compatible with the IPB. This status of the input buffer will be maintained until a REFRESH FRAME signal is given to the input buffer from an external timing device (e.g. the clock board). At this time the two memories would exchange modes; the input memory will become the output memory and vice-versa. The conceptual timing diagrams of the input buffer memory con-

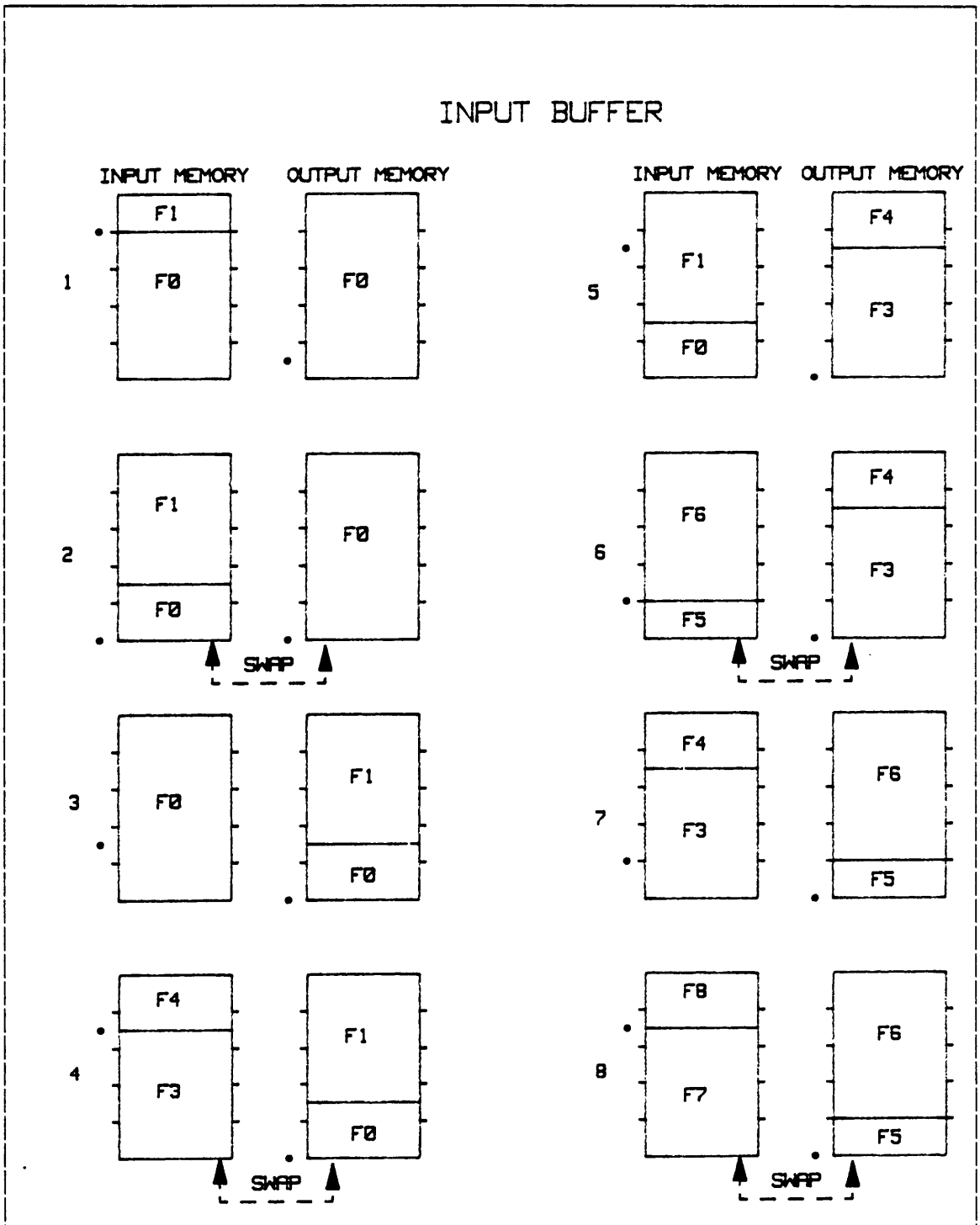


Figure 6. MEMORY CONTENTS -- TIMING DIAGRAMS.

tents are shown in Figure 6 on page 17. From these diagrams we see that in the worst case the IPB is presented with an image composed of 2 successive frames. We anticipate that the tearing effect observed because of this will be negligible. This switch should be completely transparent to the input source as well as the IPB. The only way this can be achieved is by instantaneously swapping all the control and address lines of the two memories. Figure 7 on page 19 shows the conceptual block schematic of such a system.

It should be clear from the above discussion that to be assured of fresh data at every refresh, the refresh rate should be less than the incoming frame rate of the system. Furthermore, if we wish to present to the IPB a complete frame of informatiom without a memory swap between frames, we have to wait until the end of frame of the output memory before we swap memories. Thus we must have some way of remembering the REFRESH FRAME signal until it can be used. This fact would automatically set the upper limit on the refresh rate, limiting it to the IPB frame rate. On the other hand there is no lower limit and we can freeze a frame by simply disabling the REFRESH FRAME signal.

The hardware implementation of the above concept is a rather complex task which has beeen somewhat simplied by the use of Programmable Array Logic chips(PALs) to generate all the

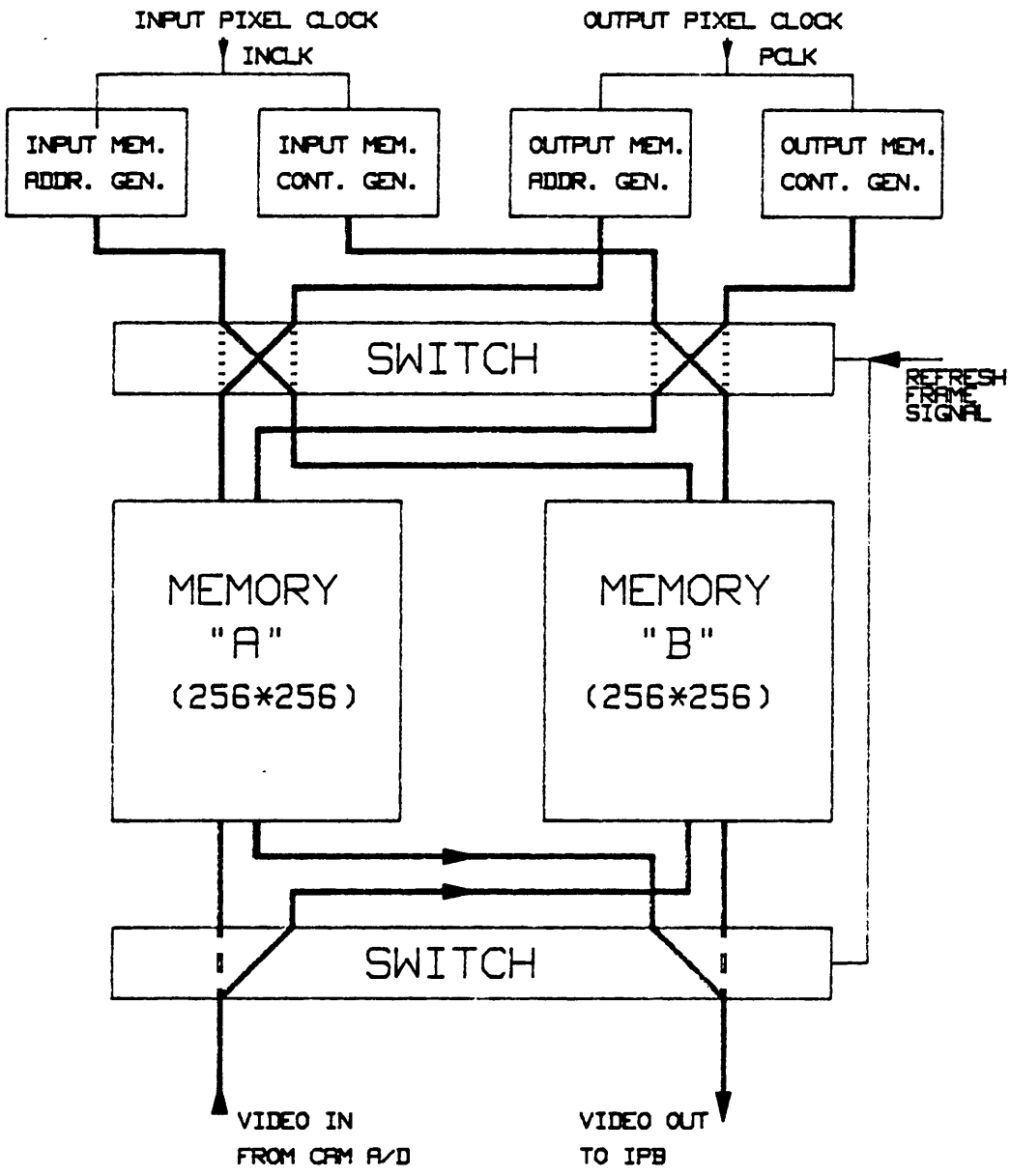


Figure 7. CONCEPTUAL BLOCK SCHEMATIC - INPUT BUFFER

controls and clocks required. The next section details the general hardware implementation of the input buffer.

2.3 HARDWARE IMPLEMENTION OF THE INPUT BUFFER

2.3.1 SYSTEM BLOCK SCHEMATIC EXPLANATION

The conceptual schematic in Figure 7 on page 19 can now be used to develop a hardware block schematic of the input buffer. The following blocks can immediately be identified.

1. Input Address Control Block

2. Output Address Control Block

3. Input Memory Control Block

4. Output Memory Control Block

5. Memory Block A

6. Memory Block B

7. "SWAP" Block

We now consider each of the above blocks.

INPUT ADDRESS CONTROL BLOCK

The function of this block is to generate the address required by the "Input Memory" so that the serial data that are being shifted into the serial shift register are stored in the appropriate memory row. It should be noted here that the video frame maps 1:1 into the TMS 4161 64K(256*256) memory. Since the above memory is dynamic we also need to generate the memory refresh address. The column address is not required because of the nature of the shift-register-to-Memory transfer [4]; however, the TMS 4161 requires A6,A7 address bits be 0 during CAS* strobe. To ensure this the column address of zero has to be generated.

OUTPUT ADDRESS CONTROL BLOCK

This block should generate the address required by the "output memory" so that the serial data that are being shifted out of the serial shift register are obtained from the appropriate row in the memory. It is essentially the same as the input address control block, but with different clocking rate and also different frame & line synchronization.

INPUT MEMORY CONTROL BLOCK

This block should generate all the control signals required to run the TMS 4161 memory in the input mode (as shown in Figure 4 on page 14). Thus the row address strobe (RAS*), column address strobe (CAS*), write (W*), register-to-memory transfer strobe (TR*/SE*) should be generated. Also all the necessary control signals required to refresh the memory should be generated.

OUTPUT MEMORY CONTROL BLOCK

This block shall generate all the control signals required to run the TMS4161 in the output mode (as shown in Figure 5 on page 15). All the signals described in Input Memory Control Block need to be generated here but with different clocking rates and also different line and frame synchronization.

MEMORY BLOCK A

The memory block will contain 8 TMS 4161 64K(256*256) memory chips. This will allow for 256*256*8 bits. Since each pixel is 8 bits wide, 8 TMS 4161 would store an entire frame with 8-bit resolution per pixel. These 8 bits of data will be fed in parallel to the memory block. These would be shifted into 8 256-bit shift registers and eventually into the appropriate row in each memory.

MEMORY BLOCK B

This block will be identical to Memory Block A in every respect.

SWAP BLOCK

This block will be a set of 2 to 1 multiplexers which will toggle control/address and the data between the two memories. The switch will be controlled by the "Refresh Rate Control Pulse".

These basic blocks are a logical realization of the conceptual block schematic. However, we can see from the above description that the hardware can be designed by combining the input-output address control along with its multiplexer circuits in a single block and, similarly the Output-Input Memory Control Block along with its multiplexers in one block. The two memories will form the other two blocks. As will be apparent later it would be more convienent to design the input buffer by combining all controls, including the controls required by address generation logic and its multiplexer circuit in one single block. The counters required to generate the addresses (row,column,refresh) will be combined in one block. Since the row, column and refresh address are to be multiplexed to the address lines of the memory, one ad-

dress bus with the necessary multiplexers and tristate buffers would be needed per memory. Thus two memory bus blocks would be required. Finally, two memory blocks will complete the input buffer. The following will be used to implement the hardware.

1. INPUT BUFFER CONTROL BLOCK (IBC)

2. INPUT BUFFER ADDRESS COUNTER BLOCK (IBAC)

3. INPUT BUFFER MEMORY A BUS (IBMAB)

4. INPUT BUFFER MEMORY B BUS (IBMBB)

5. INPUT BUFFER MEMORY A (IBMA)

6. INPUT BUFFER MEMORY B (IBMB)

Figure 8 on page 25 shows all the above blocks interconnected to each other to form the input buffer. The input-output timing diagrams are presented in Appendix A. The next section takes a closer look at each of the above blocks in greater detail.

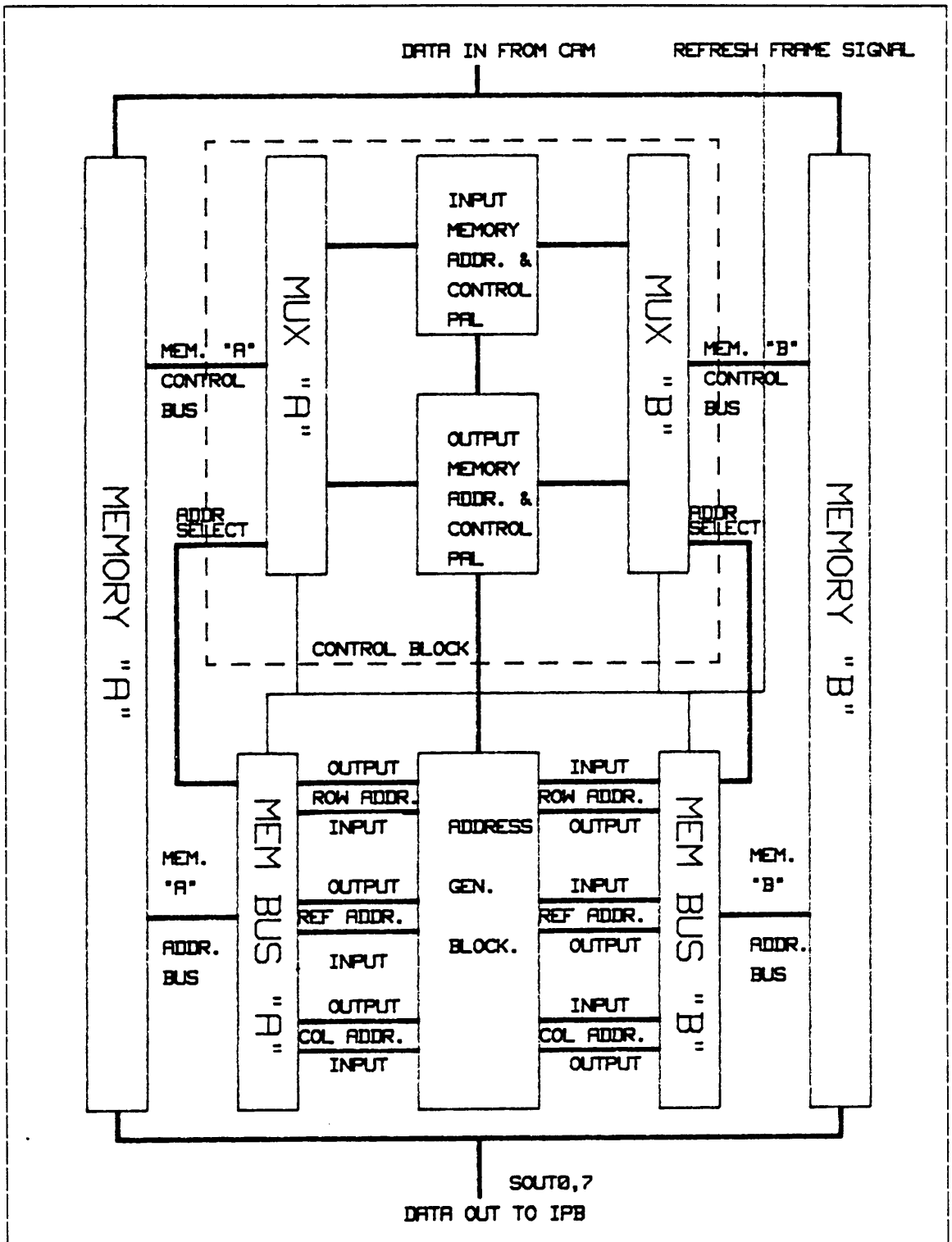


Figure 8. BLOCK INTERCONNECTION - INPUT BUFFER

2.3.2 INPUT BUFFER CONTROL BLOCK.

The control block must generate all the signals required to control the memories as well as the address counter block. The detailed hardware design is explained in Appendix A. However the basic design concepts used and all the different control signals generated are highlighted here. Figure 9 on page 27 shows the block schematic of the IBC block.
The IBC generates all the signals required to control the memories. They are as follows:

- IRASI*, IRASO* - Row Address Strobe.
- ICASI*, ICASO* - Coloumn Address Strobe.
- IWI*, IWO* - Write Enable.
- ITRI*, ITRO* - Refresh Control.
- ISOEI*, ISOEO* - Serial Output Enable Strobe.

(Note: A prefix I indicates that the signal belongs to the input buffer, similarly O indicates that the signal belongs to the output buffer. A suffix of I or O indicates that the signal belongs to the input memory or to the output memory respectively. Also a sufix of A or B indicates that the sig-

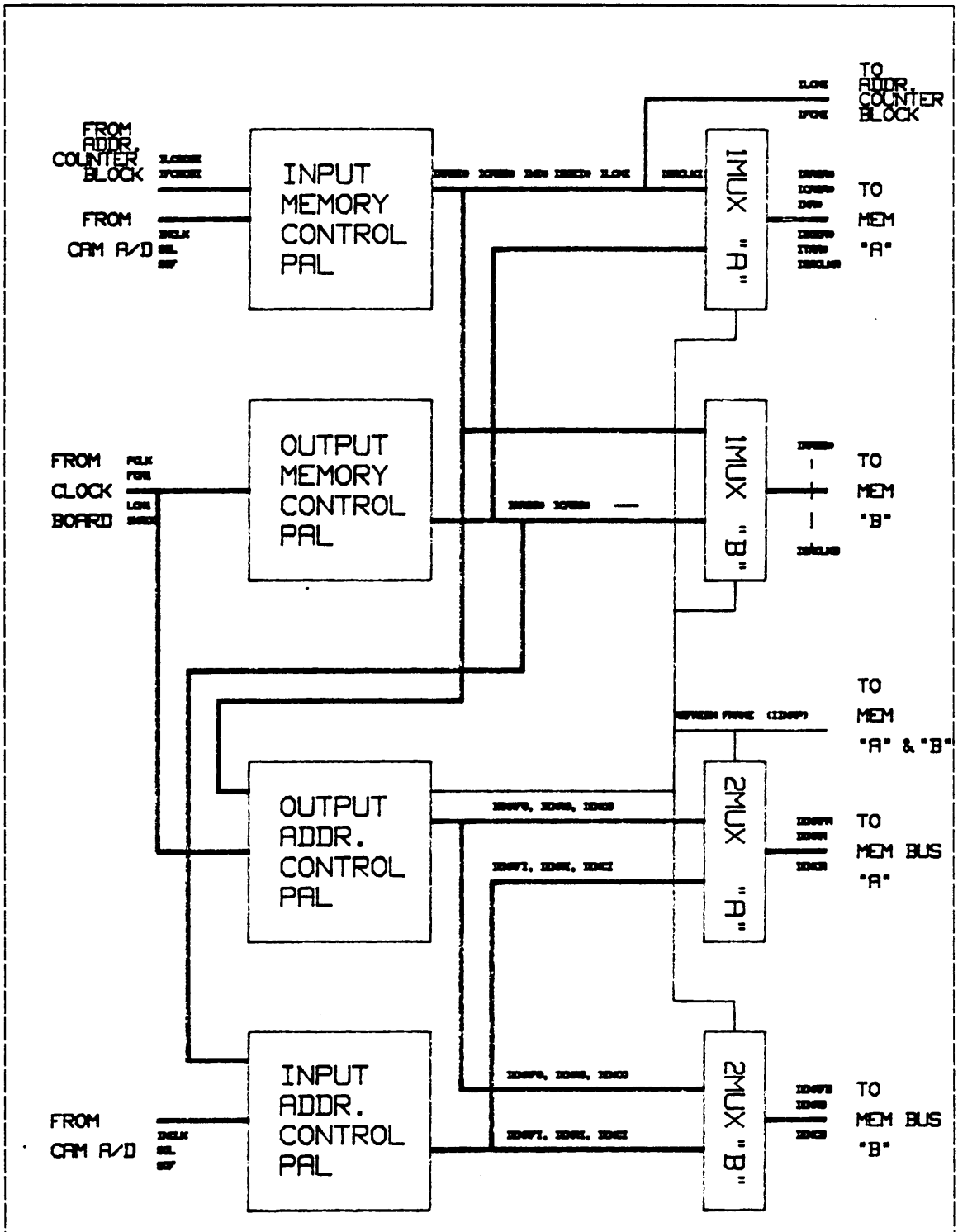


Figure 9. BLOCK SCHEMATIC - IBC BLOCK.

nal belongs to memory A or B respectively. This is irrespective of its current mode. e.g. IRASA* - Input Buffer Row Address Strobe for memory A). Further note that * , ' , / all indicate complement of the signal.)

The control block should also generate the serial clocks for both the input and the output memories. In addition to these signals the control block must generate the line and the frame markers to indicate the valid periods of the lines and the frame. These signals have to be generated from the Start of Line (SOL) and Start of Frame (SOF) signals received from the camera. Thus the following additional signals need to be generated.

- ISRCLKI, ISRCLKO - Serial Clock.
- ILCMI - Line Marker For Input Memory.
- IFCMI - Frame Marker for Input Memory.

The TMS 4161 memory has only 8-bit address lines. Thus the row and the column address have to be multiplexed. Also, since rowwise refresh needs to be accomplished, the refresh address must also be multiplexed. This requires the creation of a memory address bus for each memory; thus control signals are required by each of the address counters to ac-

cess the bus. Thus the following signals need also to be generated.

- IENRI, IENRO Enable Row Address.

- IENRFI, IENRFO - Enable Refresh Address.

- IENCI, IENCO - Enable Column Address.

Finally on receipt of the Frame rate control pulse the control circuit must generate the bus multiplexer control signal (REFRESH FRAME). This signal will swap all the signals to the two memories. These signals are the following:

- IINAB - REFRESH FRAME Signal.

- IAINB - /REFRESH FRAME Signal.

Appendix A shows all the circuit details of the IBC block. Four Programmable Array Logic chips (PAL) are used to generate the above control signals. PALASM files are included in Appendix A.3. They describe all the sequential and combinational logic equations that correspond to the signals. The equations were derived from the timing diagrams also in Appendix A.

Four IC 74AS157 2 to 1 quad multiplexers [5] are used to multiplex the input-output control signals to the A or B memory. The important points to note here are the following:

1. The frame switch pulse is remembered by the PAL and the actual switching is done in the frame dead time that follows this pulse. This is to avoid any loss of pixels during switching.

2. The SOL and the SOF pulses are not necessary for the output memory because the line and the frame markers are generated on the clock board since they are required to generate the IPB clock(refer Chap. 4 for details).

2.3.3 INPUT BUFFER ADDRESS COUNTER BLOCK.

The schematic of this block is shown in Figure 10 on page 31 Appendix A. has the circuit diagram of this block. The IBAC consists of three subblocks for generating the input-output row, column and refresh addresses respectively. IC 74AS869 8-bit counters [5] are used. The counters are controlled by signals generated in the IBC block. The clocks are provided by the clock board. The column addresses of both the input and the output memory are always zero and hence need not be generated in the IBAC block. The row addresses are generated

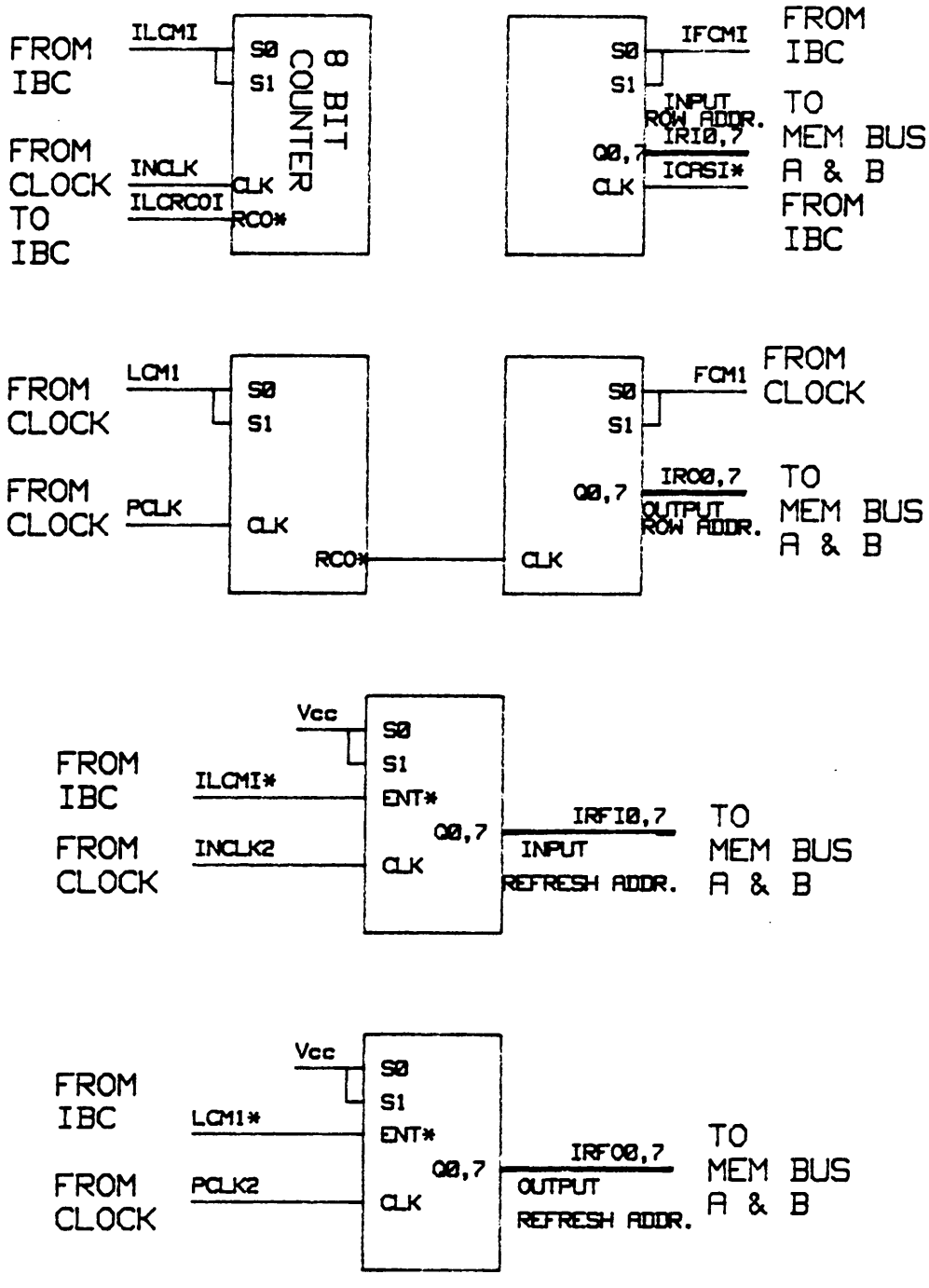


Figure 10. SCHEMATIC - IBAC BLOCK

by using 8-bit counters; these counters are clocked by the line markers of the respective memories. The row refresh addresses are similarly generated, the counters are clocked by half the pixel clock of the respective memory.

The important points to note here are:-

1. The refresh counter is inhibited when refresh is not being performed using the ENT input of the refresh counter.

2. The row counters are cleared for the entire dead time between the lines and frames.

3. The input row counter is clocked by the CAS* strobe; this ensures that the row address remains stable during shift register to memory cycle.

2.3.4 INPUT BUFFER MEMORY BUS (A OR B)

The two memory buses are identical and consist of three 2 to 1 octal multiplexers with latched as well as tristated outputs. Thus the clocks to these multiplexers are the control signals which enable the latches, while the multiplexing is controlled by the REFRESH FRAME control signal. Figure 11 on page 33 has the details of the input buffer memory bus A(B).

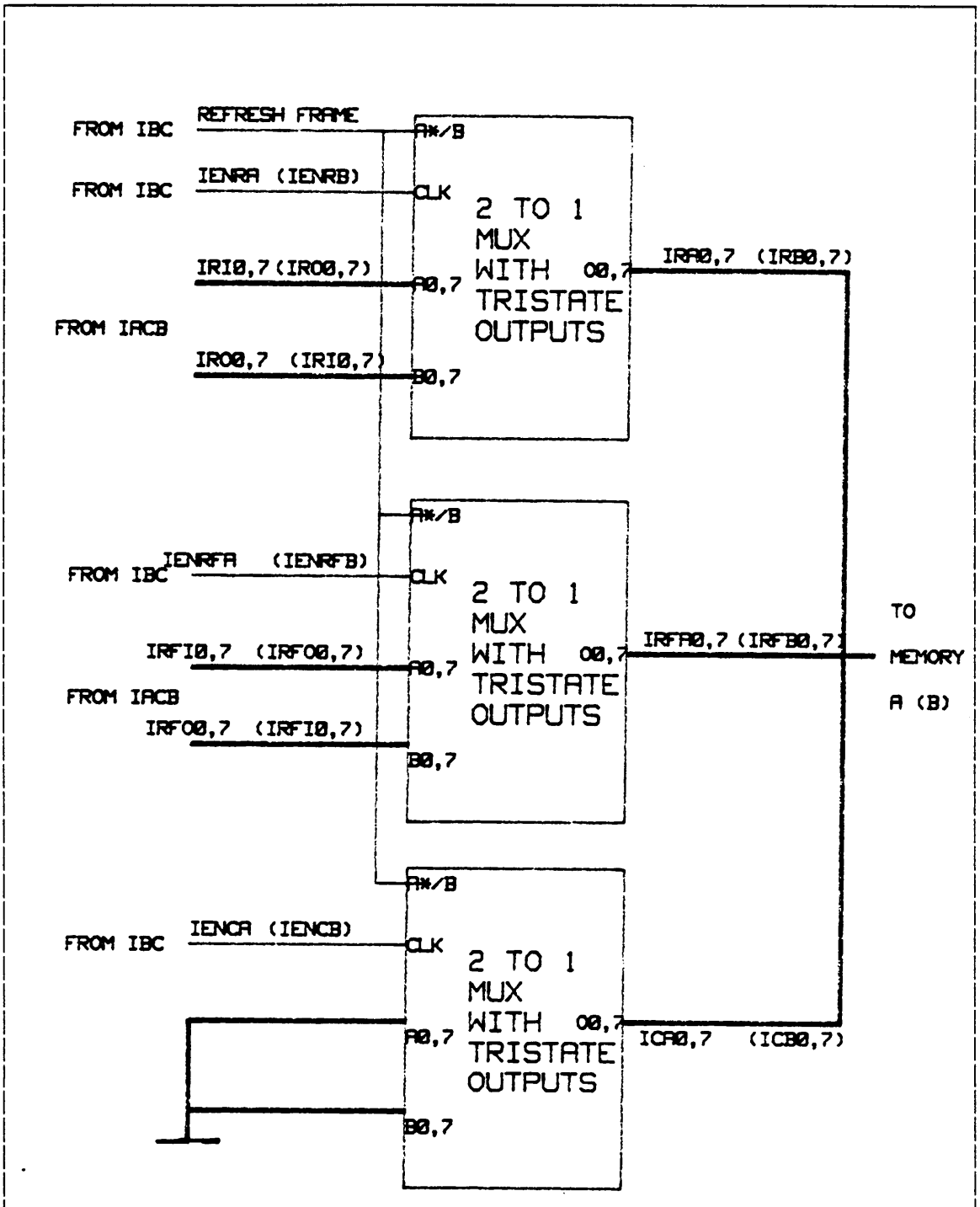


Figure 11. SCHEMATIC - INPUT BUFFER MEMORY BUS A (B).

Appendix A contains all the details of the circuit developed to perform this operation.Also the timing diagram indicating the multiplexer clocks (IENR,.... etc) are shown in Appendix A.2

Important points to note here are :

1. The memory is refreshed while the serial shift register is being filled with a video line. It is clear from the timing diagrams that exactly half the rows of memory are refreshed during serial operation. As soon as the serial register is filled the refresh operation is inhibited, i.e., the last address is frozen until the next line serial read has started, during which the other half of the buffer is refreshed. Immediately after the line dead period starts, a serial register to memory transfer cycle is initiated and the row and the column address are presented respectively to the address bus. This is for the input memory.

2. For the output memory a very similar operation is performed; the difference is as follows: The memory is refreshed during the serial shift out of the video line instead of the serial shift in. Secondly, instead of the serial register to memory transfer, memory to serial register cycle is initiated.

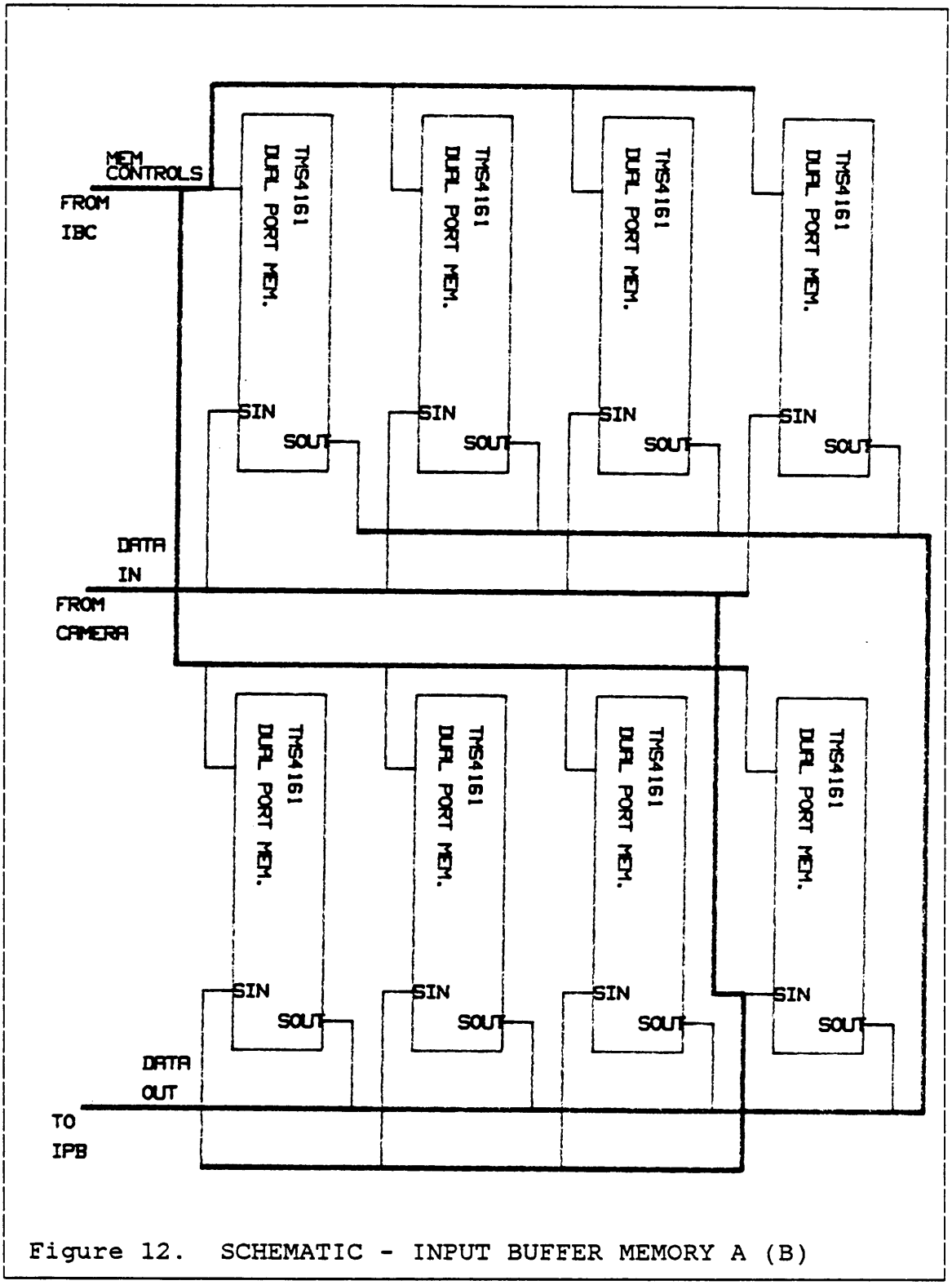


Figure 12. SCHEMATIC - INPUT BUFFER MEMORY A (B)

2.3.5 INPUT MEMORY BLOCKS A & B

This is the same as mentioned in section 2.2.1. Figure 12 on page 35 contains the block schematic of the memory block. Appendix A has the detailed circuit interconnection of this block.

2.3.6 BRIEF DISCUSSION ON DESIGN OF CONTROL CIRCUITS & USE OF PAL'S.

All the control circuits required for the input buffer are in the IBC block. The controls can be clasified as follows:

1. Address bus multipler control (IENRI,IENRFI... etc)

2. Memory control signals (IRAS*,ICASA*...etc)

3. Control signals required to control the IBAC board.

4. Serial clocks for the input and output memory serial registers.

The above control signals are derived from the available inputs. PALs are used for this purpose. The combinational and sequential equations defining these equations are derived from the timing diagrams of Appendix A.2. The verification

of these equations is done by the PALASM program. The program simulates the truth table of these equations and compares it with the function table supplied to the program from the timing diagrams. The PALASM files, with the simulation results and PALASM generated fuse plot of the PAL, are included in Appendix A.3. The JEDEC file used by the PAL programmer to program the PAL is also included.

PALs were prefered over standard LSI/MSI devices or PROMs for the following reasons [8], [9]:

1. The operation is real time, requiring very fast operation. (The input clock rate is over 5 MHz and the output clock rate is just over 2MHz.) PALs are ideal for such fast operation with delay typically between 10 to 15 ns.

2. Only 4 PALs were required to generate all the control signals, compared to at least 30-40 MSI chips that would have to be used.

3. Speed and ease of programming of PALs make them preferable to PROMs.

This concludes the input buffer. The next chapter describes the output buffer in detail.

3.0 OUTPUT BUFFER.

3.1 OBJECTIVE OF OUTPUT BUFFER.

The function of the output buffer has been briefly described in Chapter 1. In this section we shall attempt to fully explain this function. To gain perspective on the output buffer we need once again, to look at the current requirements of the Telesign system. These are outlined clearly in section 2.2.1 of the input buffer. In summary, the output buffer has the following functions:

1. To condition digitised video from the IPB into a video signal that can be converted to RS 170 standard video format ready for display on a standard TV monitor.

2. To regulate the frame rate (refresh rate) seen by the monitor, so as to evaluate the minimum acceptable frame rate of the processed image.

3. To post-process IPB image into various formats in order to reduce the effective data rate of the system before supplying it to an interpolator circuit and eventually to the Video Digital to Analog Converter (VDAC).

4. To generate appropriate blanking pulses associated with each of the above formats and also to generate all the I/O control siganls to the interpolator.

The specification is basically familiar as it says that the output buffer should be able to do a reverse data rate transformation which will be able to convert the slower data rate out of the IPB to the faster data rate required by the RS 170 2:1 interlaced standard video at the output.

For the Telesign system this will mean converting a 2.016 MHz input data rate to a 5.376 MHz data rate at the output. As discussed in Chapter 2,the IPB frame rate is fixed at 27.255 f/s; it will be required to change this to 60 f/s (fields/sec). This requirement is automatically taken care of by the faster output data rate. The frames are synchronised with the video display by the start of line (SOLV) and the start of frame (SOFV) generated in the VADC circuit. This information is used to clock the serial register of the output memory of the output buffer. The details are given in the next section. The block schematic of the output buffer is indicated in Figure 13 on page 40.

The second function is the same as the second objective of the input buffer. Here we are interested in just propagating the refresh rate selected at the input buffer. In the Tele-

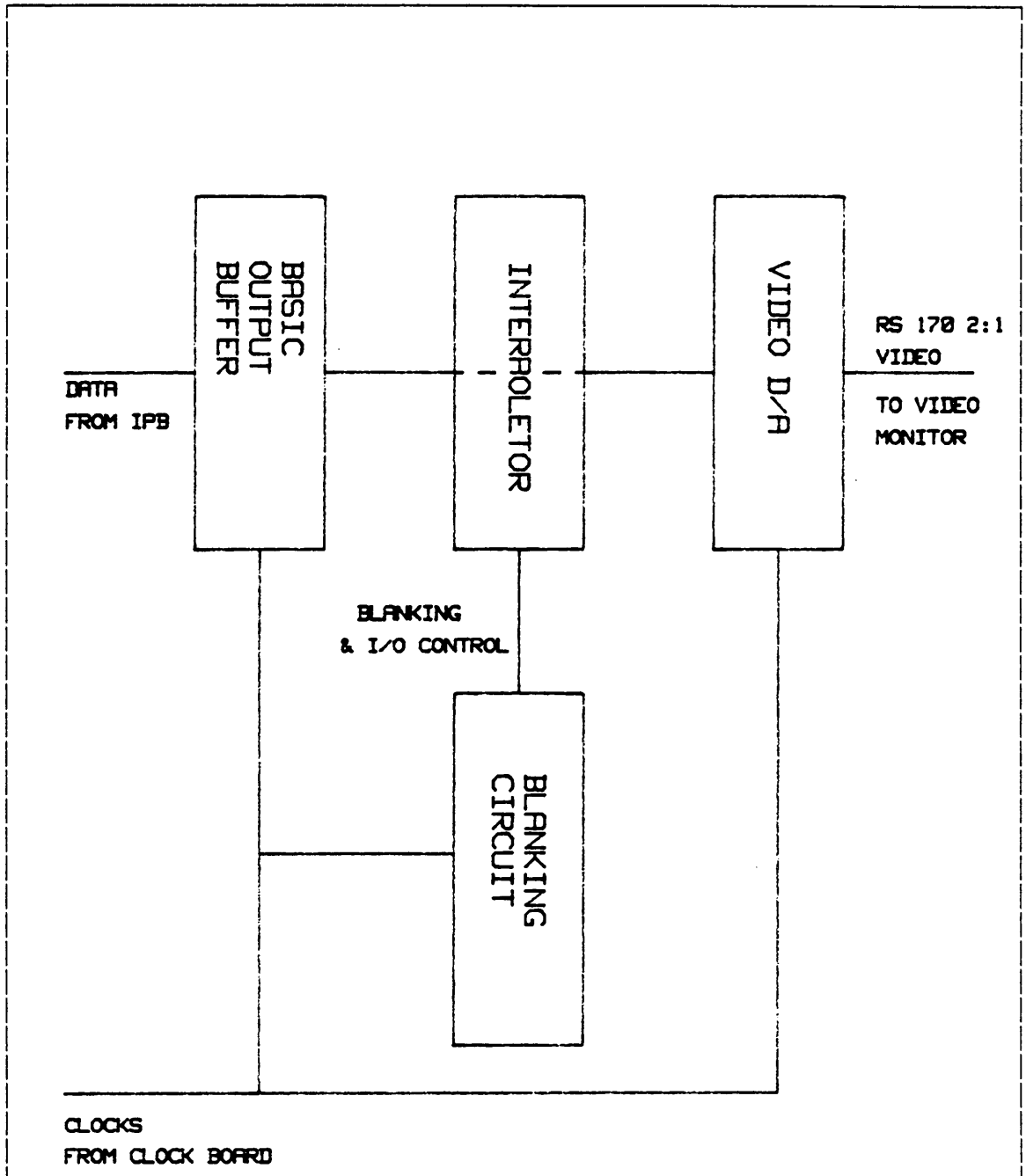


Figure 13. OUTPUT SIGNAL CONDITIONING

sign system the refresh rates of the input and the output buffers are identical. They could be made to differ; however this will serve no useful purpose since the lower rate will always be the final refresh rate as seen by the monitor.

The third function is an exercise in finding out how a change in the effective data rate would affect the quality of the picture. Various data reduction formats are proposed in the next section. One of main functions of the Telesign system is to study these various formats in conjunction with the variable frame rate and find a minimal and optimal data transmission rate.

The final function is actually linked to the third, as shall be obvious later. In order to reduce the data rate we must be able to generate blanking pulses corresponding to the locations at which we wish to eliminate the pixels in a particular frame. These blanking pulses will inform the interpolater circuit to interpolate a pixel value at that location.

The next section explains all the above functions of the output buffer in detail and also puts forth a conceptual design of the output buffer.

3.2 CONCEPTUAL REALISATION OF THE OUTPUT BUFFER OBJECTIVE.

We have seen in the previous section that there is a fourfold function of the output buffer; however we must realise that the first two functions are identical to the input buffer objectives. It is therefore logical to assume that the conceptual design as well the hardware design of this part of the output buffer will be identical to that of the input buffer. Figure 14 on page 43 indicates this conceptual circuit. It should be noted here that we are now converting from a slower data rate to a faster data rate(2.0165 MHz to 5.376 MHz). The input frame rate is 27.255 F/s while the output frame rate is 60 F/s. It is important to note that the refresh rate, as in the input buffer, is independent of the input and output frame rates of the output buffer. The only restriction on the refresh frame rate is that it cannot exceed the input frame rate.

It would be appropriate at this time to introduce the following concept. This concept is unique to the Telesign system. We recall that we have mentioned in several instances before that the IPB processes 256*256 pixels per frame; the clock to enable this operation of the IPB is generated on the clock board. However, for the Telesign system the IPB should also be able to process 128*128 pixels per frame in the same

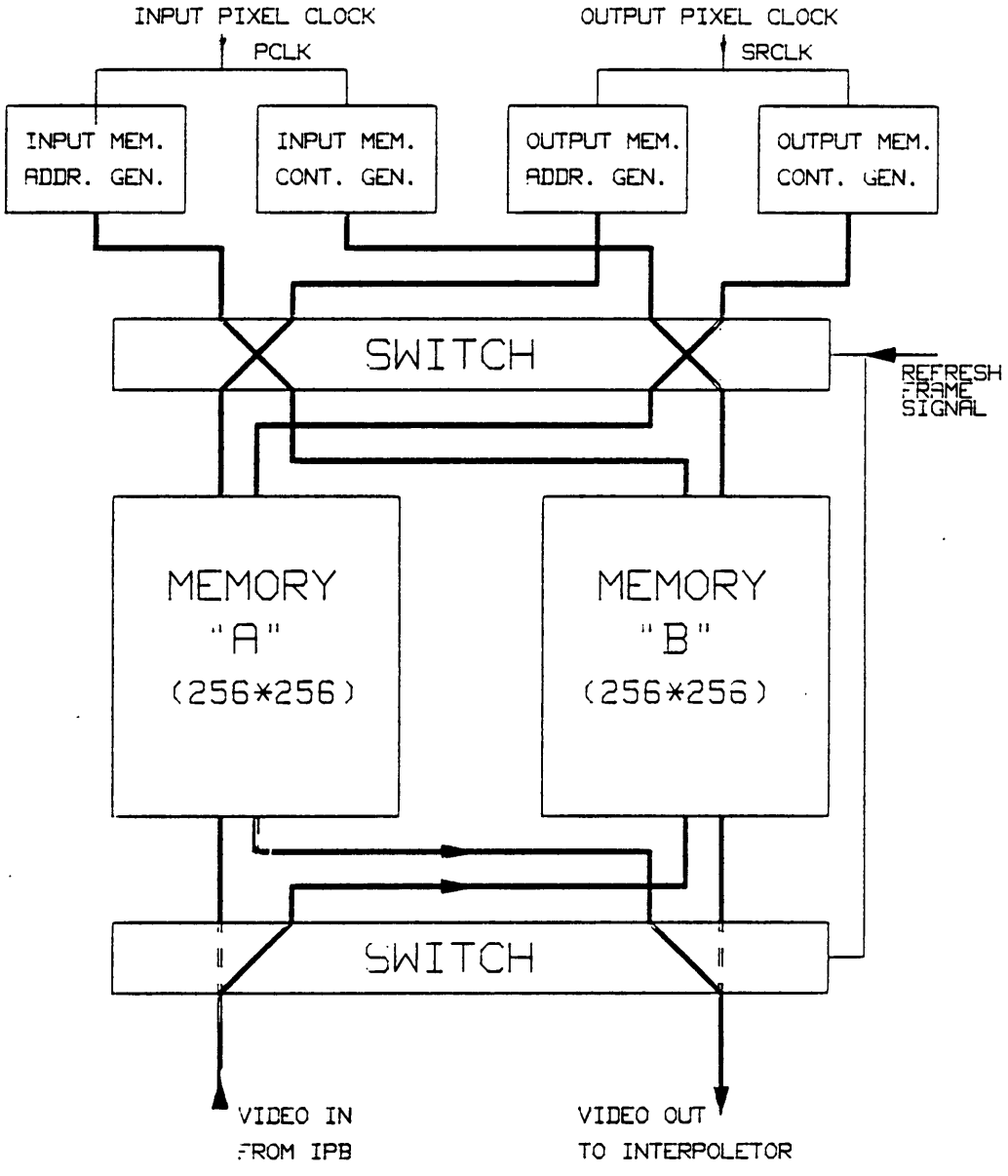


Figure 14. CONCEPTUAL BLOCK SCHEMATIC-OUTPUT BUFFER.

time frame as the 256*256 Image. This is achieved by selecting alternate pixels in a line and also selecting alternate lines for processing through the IPB. In effect this is implemented by stopping the IPB during every alternate pixel and every alternate line. Effectively each frame is made up of 128*128 unique pixels; this reduces the effective data rate by half, along with the picture resolution. The IPB can thus be identified with two basic modes of operation, the 256*256 and the 128*128.

With the above in mind, we must now use the output buffer to further reduce the data rate. The purpose is to reduce either temporal or spatial resolution or both. We must be able to achieve this reduction such that it is compatible with the inherent line interlace of the RS 170 2:1 standard video format

3.2.1 WHAT IS POST-PROCESSING?

The basic concept of post-processing should be quite clear. It is in essence a way of reducing the effective data rate by effectively blanking out selected pixels in a line, or selected lines in a frame, or both. The selections of pixels and lines to be blanked out will vary in general from frame to frame. This has the effect of reducing the effective data rate of the system, but because of a frame by frame selective

blanking an interlaced pattern can be realised. Recognising the fact that the T.V. monitor as well as the human eye has persistance of vision, this post processed image will have a much better resolution as compared to an unprocessed image at the same average data rate. The main objective of the Telesign system is to evaluate the improvement in the image quality after post processing.

In addition to the 256*256 and 128*128 non-interlaced formats, we propose here various formats to achieve the above mentioned interlacing with effective data rate reduction per frame:

1. Line Interlace.

2. Column Interlace.

3. Dot interlace.

LINE INTERLACE.

Figure 15 on page 46 shows the line interlaced format for a 4*4 image; this can be easily extended for the 256*256 image. We have made a further simplifying assumption for the purpose of demonstrating this concept. We assume that the Telesign refresh rate is half that of the output video rate. This is not true but this adequately explains the interlacing con-

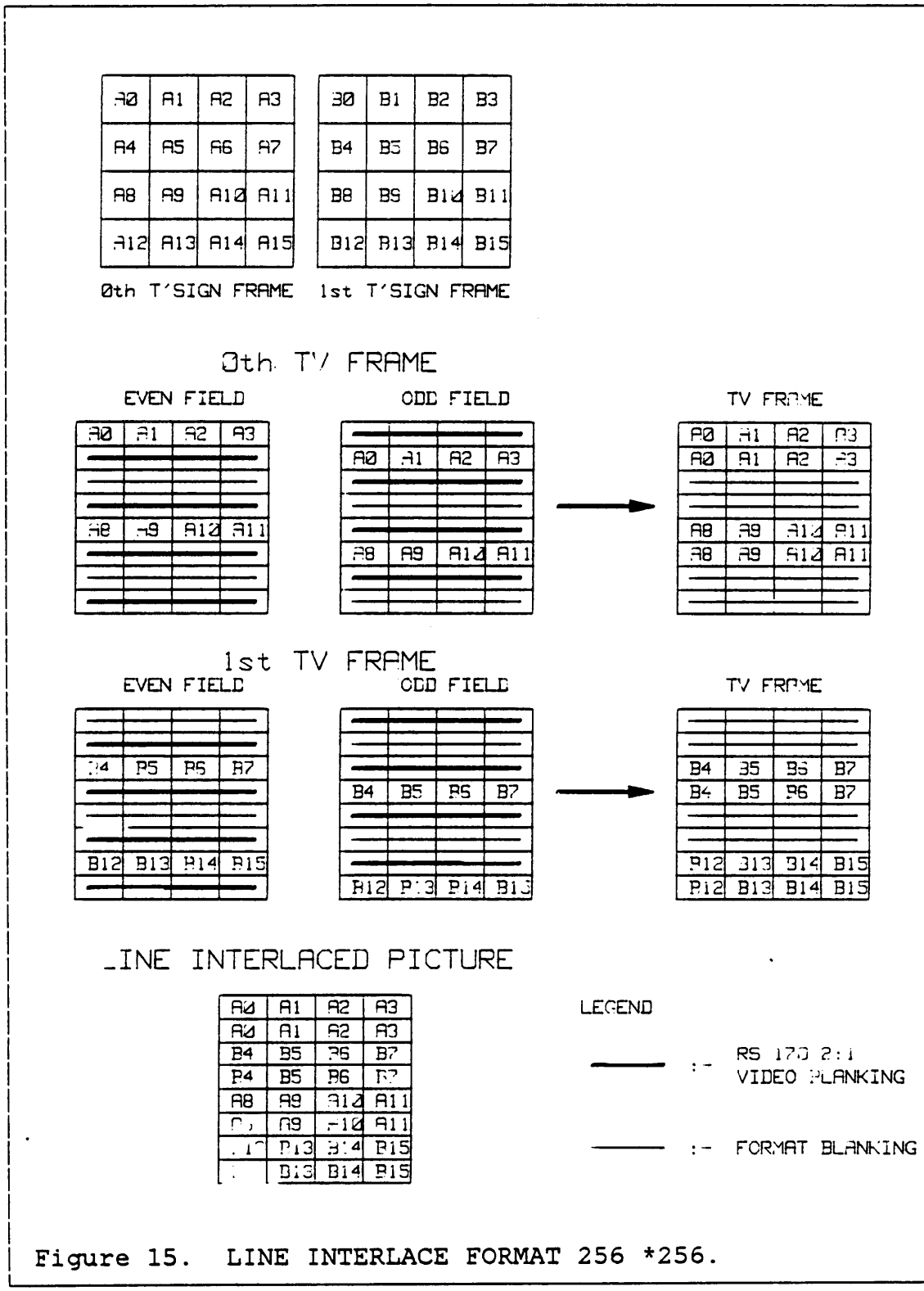


Figure 15. LINE INTERLACE FORMAT 256 *256.

cept. Also in Figure 15 on page 46 we see how the image would appear on the T.V. screen after two video frames. Since two complete video frames are required to fill up the video image we can look upon this as a 128*256*2 format because each video frame contains 128*256 pixels of information. The bold line in the figure indicates the inherant video blanking due to the RS 170 interlace, the broken line indicates the blanked pixels in that frame. From Figure 15 on page 46we can deduce a general rule for the line interlaced format. The following sequence defines the line interlace format for each T.V. frame (even and odd video fields have the same blanking sequence).

Zeroth T.V. frame (both even and odd fields).

- Blank every odd line.
- Display every even line.

First T.V. Frame (both even and odd fields)

- Blank every even line.
- Display every odd line.

Repeat the above procedure to maintain this format.

OUTPUT BUFFER.

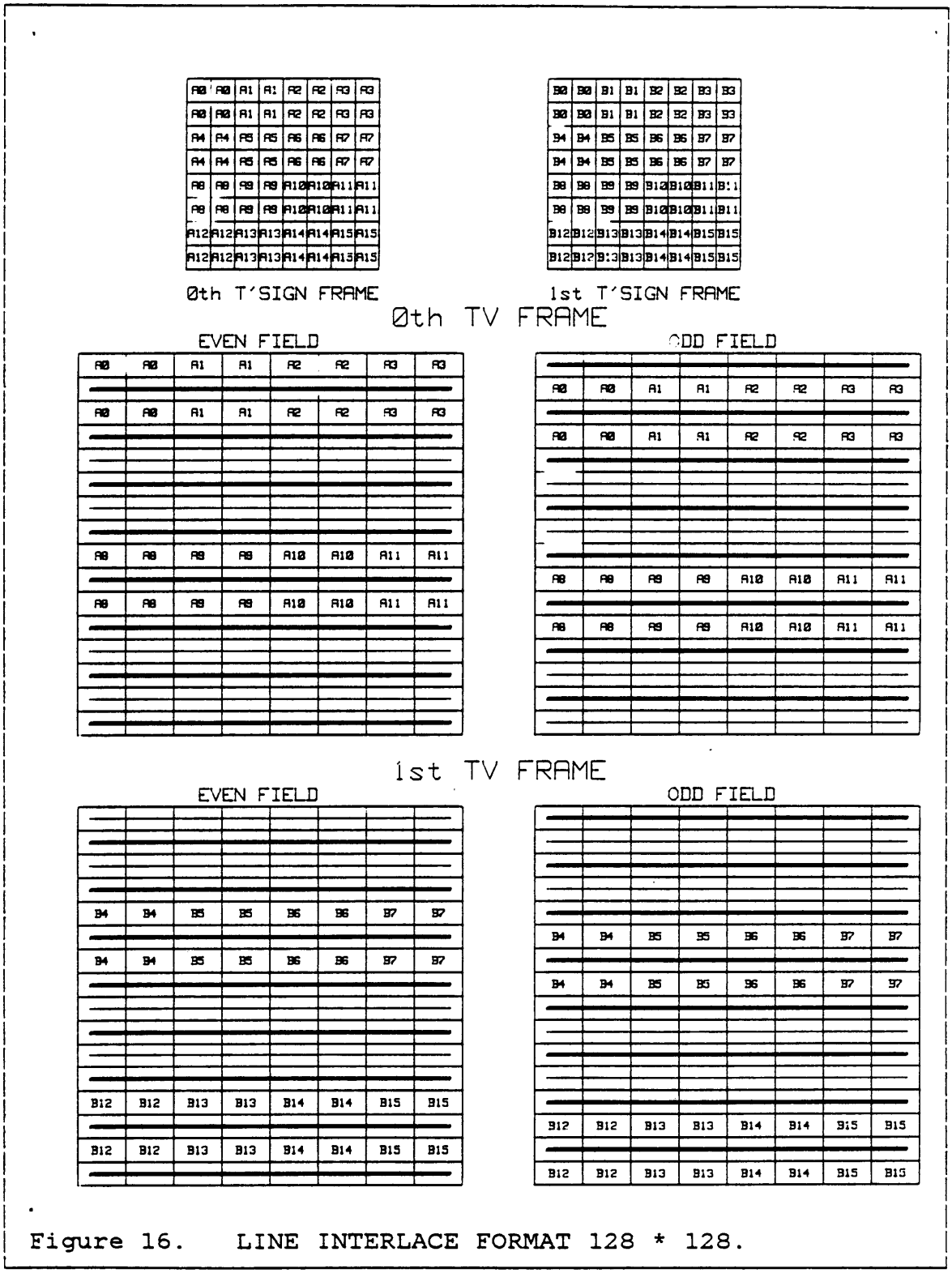


Figure 16. LINE INTERLACE FORMAT 128 * 128.

Figure 16 on page 48 indicates the line interlace for a 128*128 image from the IPB. An 8*8 image is used here as an example. It is important to note here that the output buffer is of a fixed size; hence the 128*128 image would be stored in the output buffer by repeating each IPB pixel twice and each IPB line twice. Since it takes 2 video frames to completely fill the video image and since each video frame has 128 lines of information, we require two video frames to completely fill up the video image. Hence we can look upon this as a 64*128*2 format. This is because each frame of video has only 64 unique lines and 128 unique pixels. The blanking sequence is as follows:

Zeroth T.V frame

- Display even pairs of lines.

- Blank odd pairs of lines.

First T.V frame

- Blank even pairs of lines.

- Display odd pairs of lines.

we must repeat the above sequence as long as we desire to maintain this format.

The differance between 128*256*2 and 64*128*2 format is that each pixel in the later format would be twice as large as in the former format. The interpolator would substitute an appropriate black or white pixel in place of the blanked pixel according to an algorithm using a 3*3 window. It should be noted here that in a 128*128 image the interpolator will form a window comprised of only unique pixels from unique lines. The details of the interpolator are outside the scope of this research and hence are not described here.

Column interlace

Figure 17 on page 51 shows the column interlaced format for a 4*4 image. This can easily be extended for a 256*256 image. All the simplifying assumptions indicated above for line interlaced format apply to this format. The basic difference between column interlace and dot interlace is that in each video frame we display every other pixel in each line and blank out the other pixels. Thus in every video frame we display 256 line * 128 pixels instead of the 256*256 image. Thus we would require 2 video frames to fill up one video image. Hence the format can also be called 256*128*2 format.

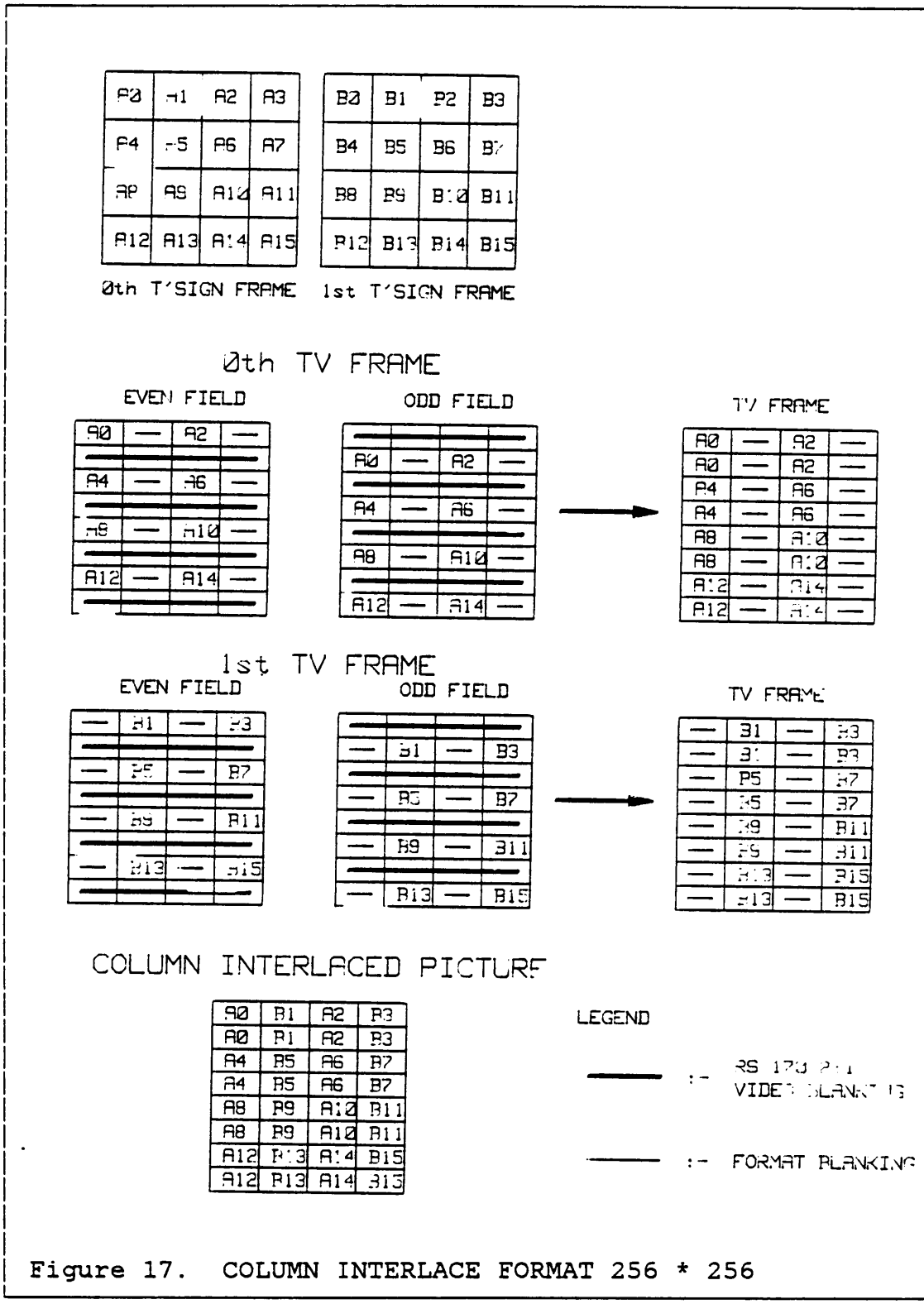


Figure 17. COLUMN INTERLACE FORMAT 256 * 256

The following defines the blanking sequence for the column interlace format:

Zeroth T.V frame:

- Display every even pixel in each line.
- Blank every odd pixel in each line.

First T.V frame:

- Blank every even pixel in each line.
- Display every odd pixel in each line.

we must repeat the above sequence as long as we desire to maintain this format.

Figure 18 on page 53 is used to demonstrate column interlace for the 128*128 mode of the IPB. An 8*8 image is used as an example. The output buffer, because of its fixed size, presents the monitor with a 256*256 image by repeating each IPB pixel twice and each IPB line twice. Since each video image contains 128 unique lines and 64 unique pixels per line we require two column interlaced frames to complete one video

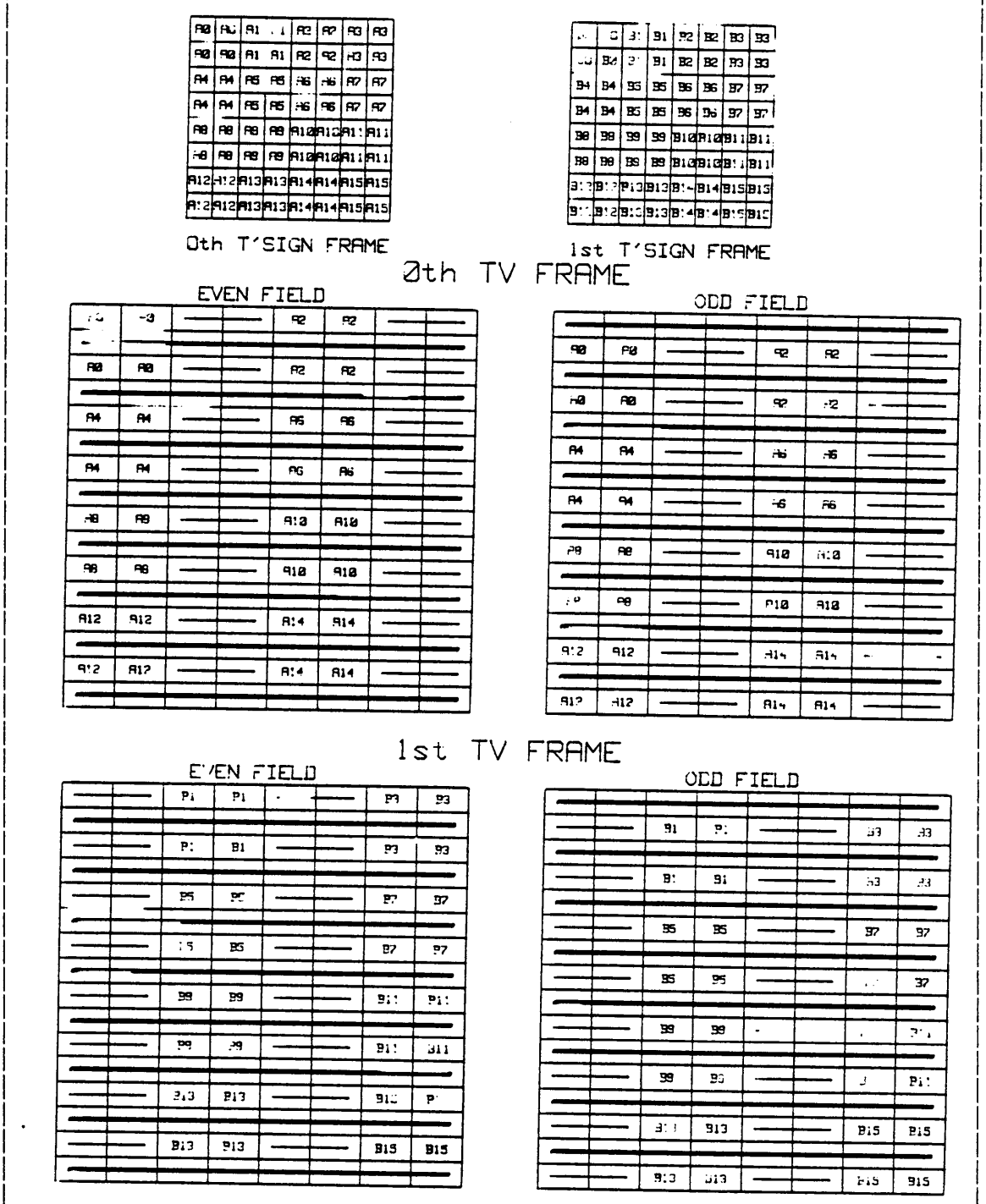


Figure 18. COLUMN INTERLACE FORMAT 128 * 128.

image; we can name this as 128*64*2 format. The blanking sequence for this format is as follows:

Zeroth T.V frame

- Display every even pair of pixels in every line.
- Blank every odd pair of pixels in every line.

First T.V frame

- Blank every even pair of pixels in every line.
- Display every odd pair of pixels in every line.

we must repeat this format as long as we desire to maintain this mode.

Dot interlace

Figure 19 on page 55 and Figure 20 on page 56 show the dot interlace format for a 4*4 image. This can be easily extended for the 256*256 format. This interlace format is actually a combination of both the column and line interlace formats implemented in a particular sequence over four video frames. As can be seen from the above figures only 128 unique lines

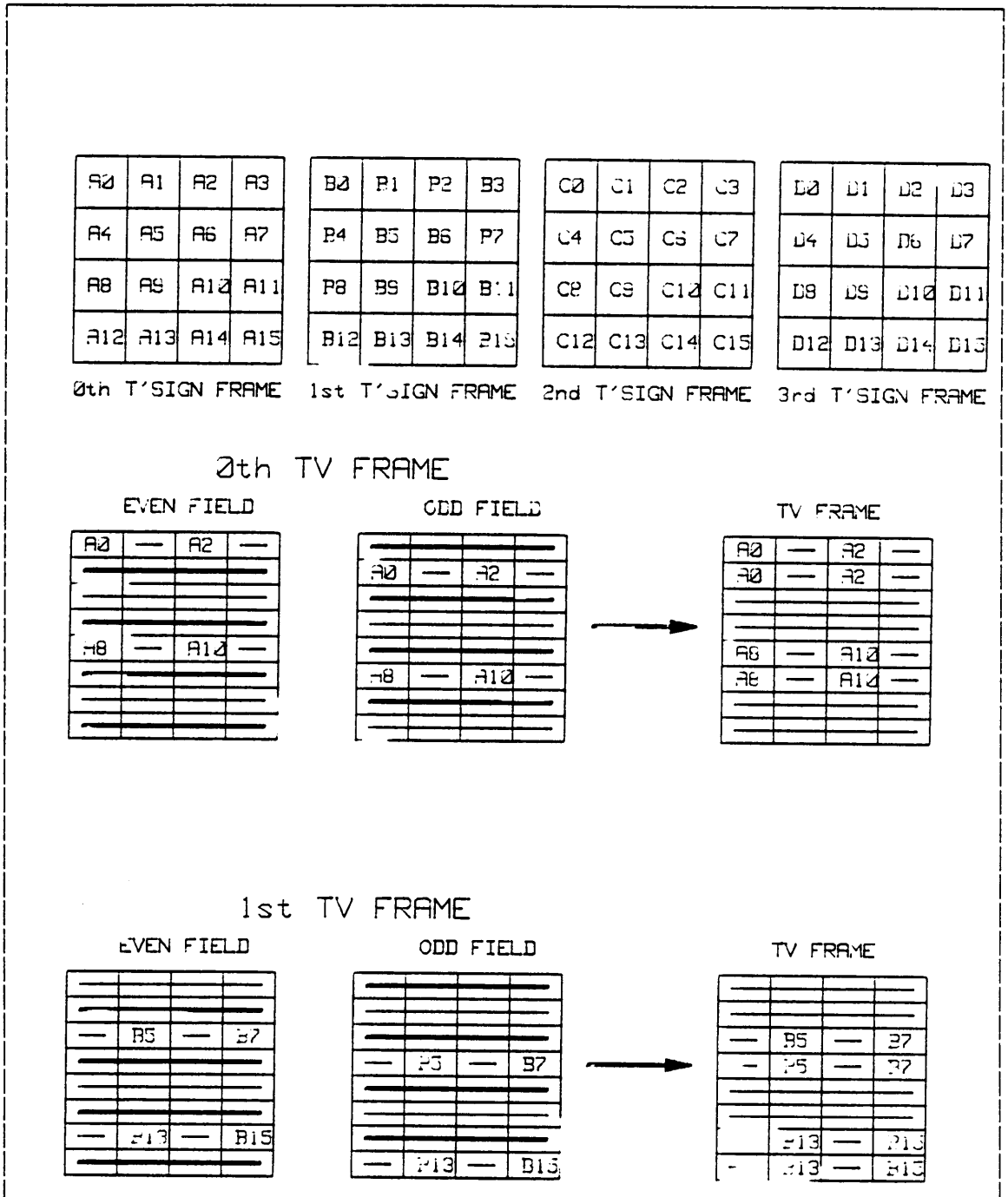


Figure 19. DOT INTERLACE FORMAT 256 * 256.

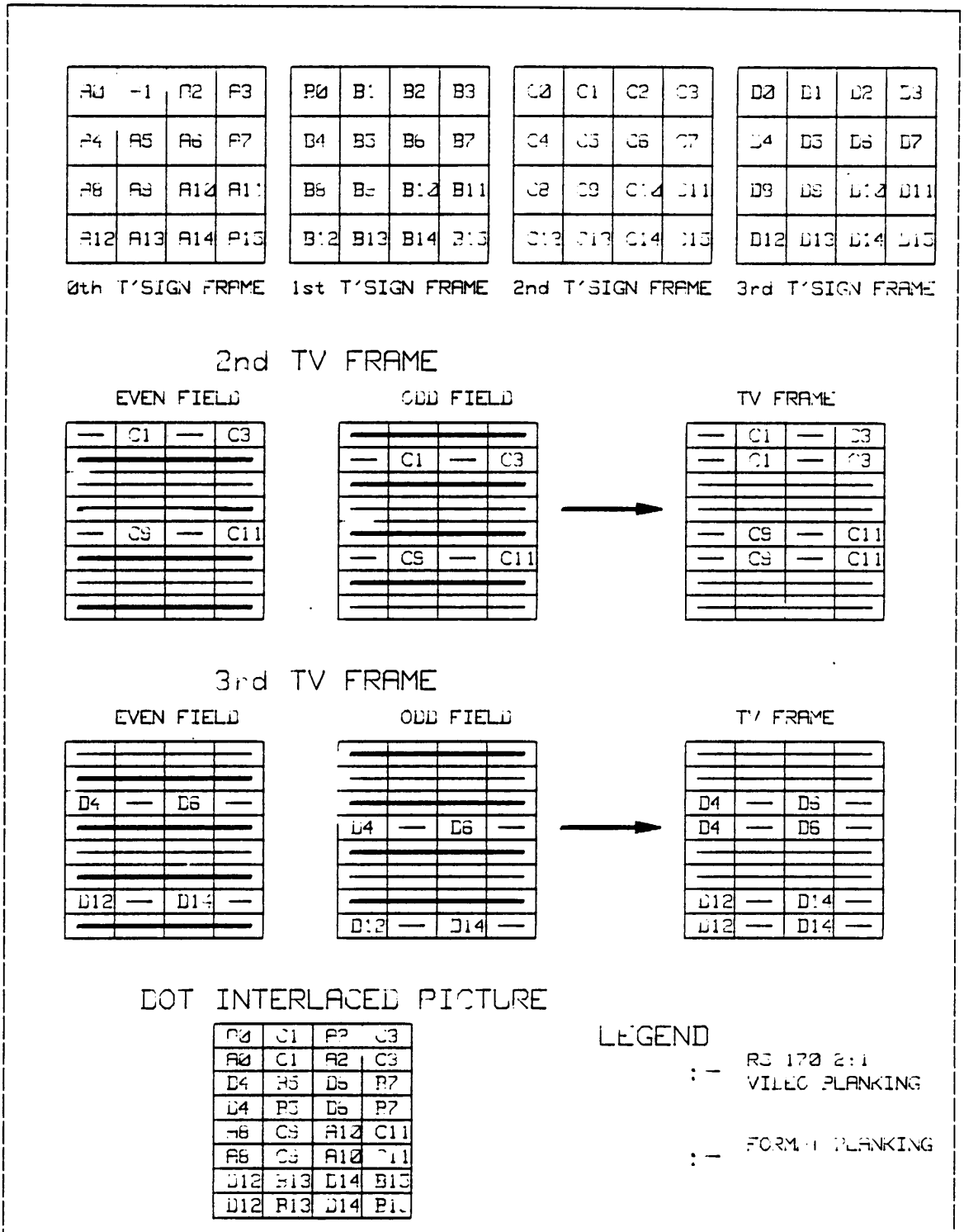


Figure 20. DOT INTERLACE FORMAT 256 * 256 (cont'd).

and 128 unique pixels are displayed in one video frame. Hence it would require 4 video frames to complete one image. Thus this format can also be labeled as 128*128*4 dot interlaced format. The blanking sequence for this format can be generalised as follows:

Zeroth T.V frame(both even and odd fields)

- Display even pixels on even lines.
- Blank odd pixels on even lines.
- Blank odd lines.

First T.V frame (both even and odd fields)

- Blank even lines.
- Blank even pixels on odd lines.
- Display odd pixels on odd lines.

Second T.V frame (both even and odd fields)

- Blank even pixels on even lines.

- Display odd pixels on even lines.

- Blank odd lines.

Third T.V frame (both even and odd fields)

- Blank even lines.

- Display even pixels on odd lines.

- Blank odd pixels on odd lines.

we must repeat the above sequence frame by frame to retain this format.

Figure 21 on page 59 and Figure 22 on page 60 are used to demonstrate the dot-interlaced format for a 128*128 image. As explained earlier the repeated lines and pixels are due to the fact that the output buffer is of a fixed size. Because of this the 128*128 IPB image is converted to a 256*256 image in the output buffer by padding alternate pixels and lines. The 8*8 structure is used to demonstrate this format. We can easily see from the above mentioned figures that we can have only 64 unique pixels per line and 64 unique lines per video frame. Thus four interlaced video frames are required to

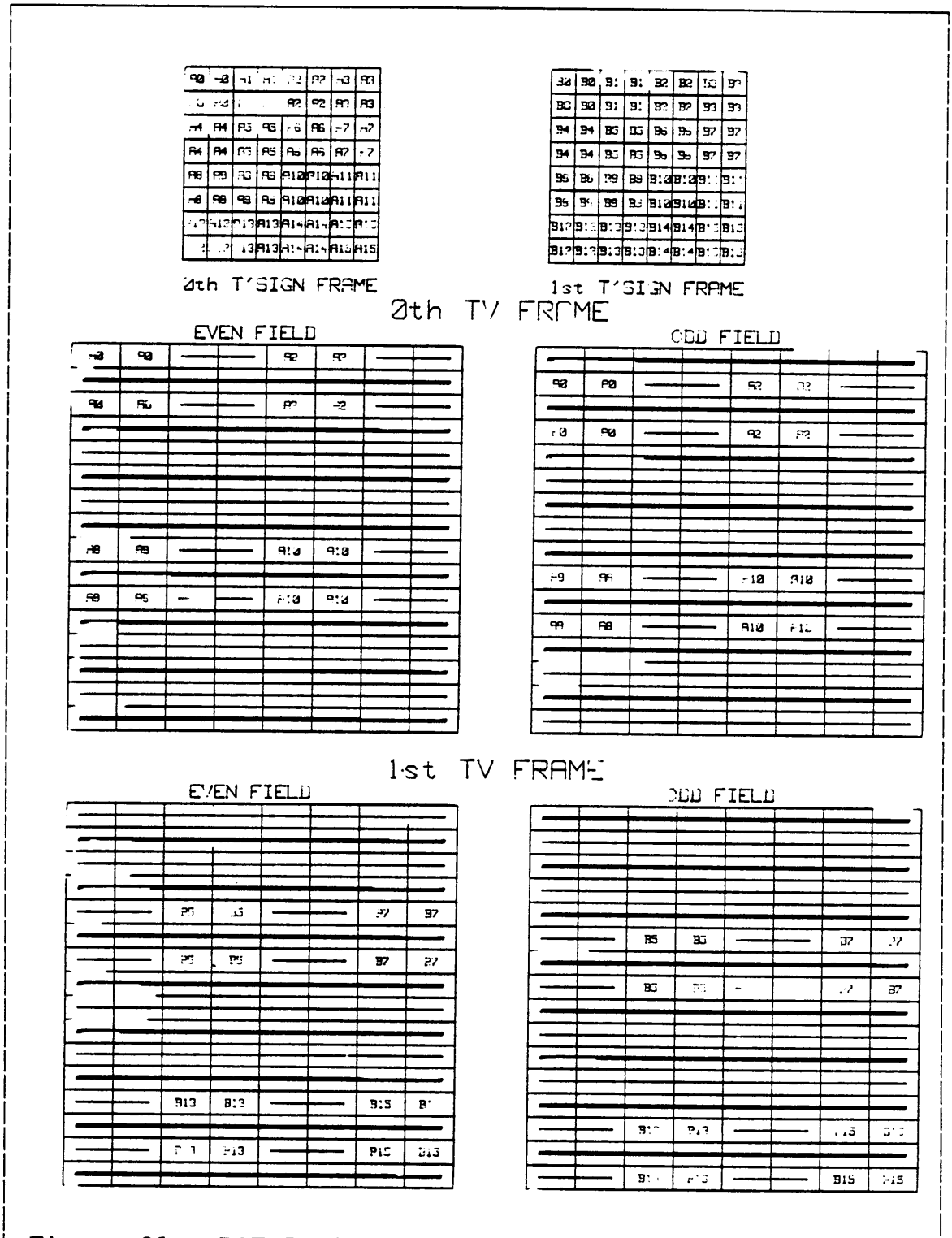


Figure 21. DOT INTERLACE FORMAT 128*128.

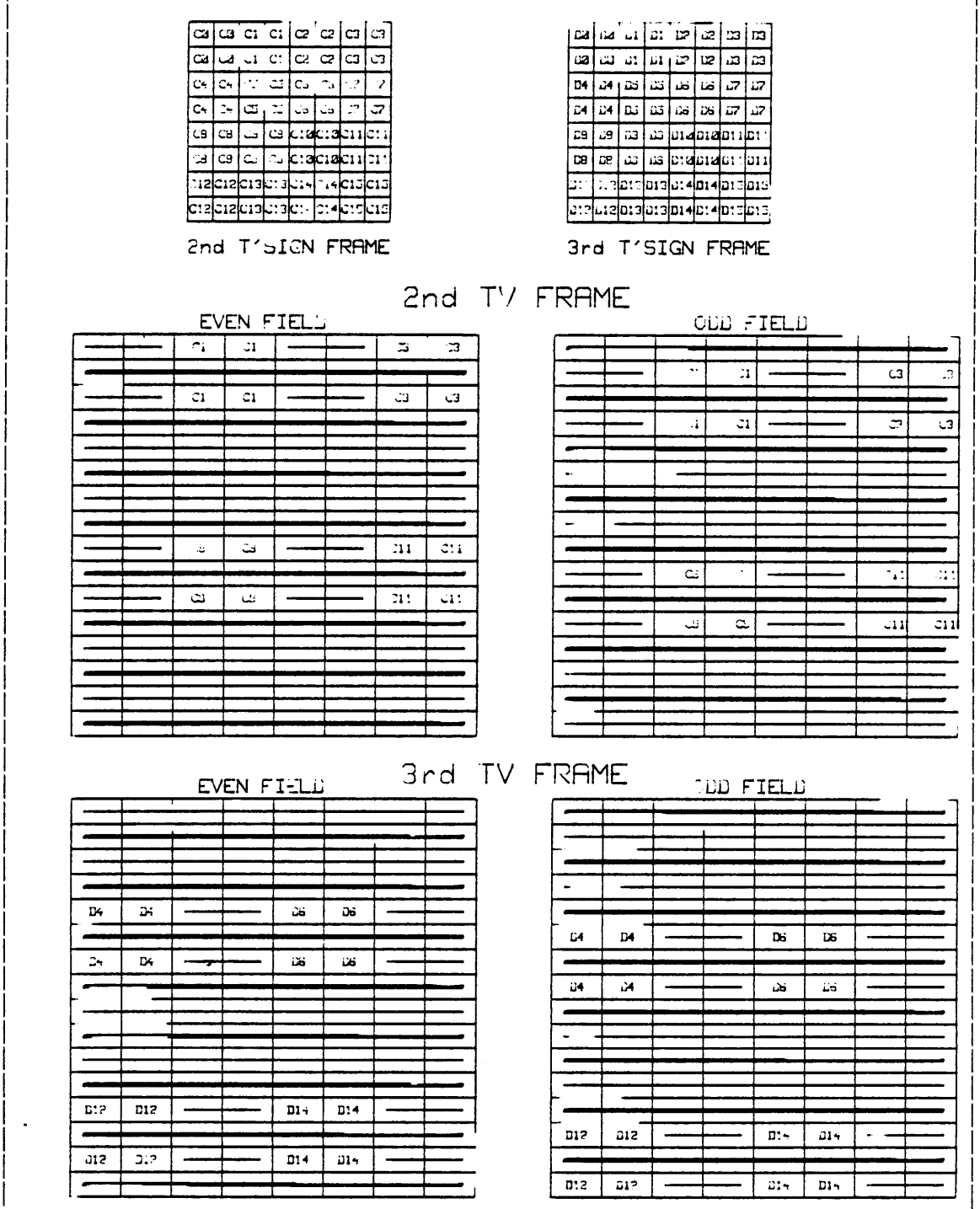


Figure 22. DOT INTERLACE FORMAT 128*128 (cont'd)

complete a full video image. We can thus look upon this format as a 64*64*4 dot-interlaced format.

The following blanking sequence must be used to generate this format.

Zeroth T.V. Frame(both even and odd fields).

- Display even pair of pixels in even pair of lines.
- Blank odd pair of pixels in even pair of lines.
- Blank odd pair of lines.

First T.V. Frame(both even and odd fields).

- Blank even pair of lines.
- Blank even pair of pixels in odd pair of lines.
- Display odd pair of pixels in odd pair of lines.

Second T.V. Frame(both even and odd frames).

- Blank even pair of pixels in even pair of lines.

- Display odd pair of pixels in even pair of lines.

- Blank odd pair of lines.

Third T.V. Frame(both even and odd frames).

- Blank even pair of lines.

- Display even pair of pixels in even pair of lines.

- Blank odd pair of pixels in even pair of lines.

This sequence must be repeated as long as we wish to maintain this format.

Please note the following point regarding the basic output buffer design. As stated earlier the basic output buffer is identical to the input buffer, the difference being only in the configuration and routing of the signals. However, when the IPB is configured in the 128*128 mode the output memory row addressing has to be altered to enable us to reconfigure this image to 256*256.This is done by repeating each IPB pixel and line. This process must be transparent to the basic output buffer. The address lines are conditioned in the blanking circuit PAL of the output buffer. The details of how this is done are indicated in the next section.

3.2.2 WHY POST-PROCESSING?

The following points adequately explain the need for post-processing and its conceptual realization.

- Actual blanking of the pixels or lines as required by the formats is not done at all. Instead, the output buffer transmits a 256*256 image to the interpolator during all formats. However, the blanking circuit provides the required blanking pulses to the interpolator in synchronism with the data. The blanking circuit also generates the input and the output clocks of the interpolator. This ensures that only the unique lines and pixels go through the interpolator during 128*128 configuration of the IPB. The output clock of the interpolator is same as the serial output clock of the output buffer. Thus the interpolator is transparent to the VDAC and, in fact, we can bypass the interpolator and provide the VDAC with blanking pulse and the output memory data of the output buffer. The VDAC will blank out the appropriate pixels while displaying the image on the monitor. If the interpolator is incorporated into the circuit, an interpolated pixel or line will replace the pixel or line that was previously blanked out.

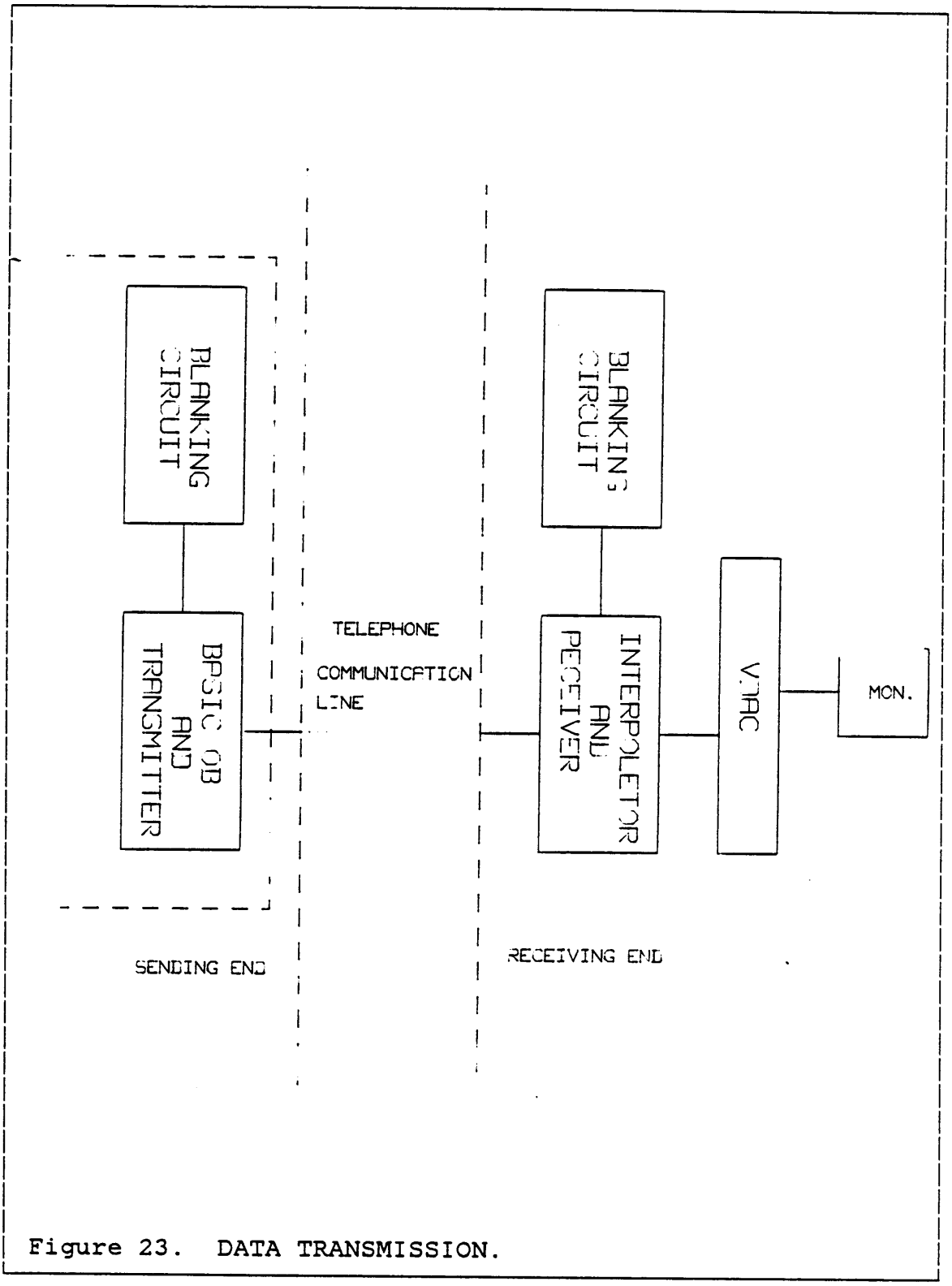


Figure 23. DATA TRANSMISSION.

OUTPUT BUFFER.

- The important point to note here is that the IPB outputs only one bit per pixel. Thus the interpolator will guess the value of the blanked pixel by looking at the neighbouring pixels in a 3*3 window. This would have the effect of smoothing edges of the Telesign images as well as reducing the artifacts caused by simply blanking the pixels or lines.

- The blanking and interpolating is necesary because in the final Telesign system the output buffer data will be transmitted over a communication line and a low data rate would be a requisite. The interpolator will be at the receiving end along with the VDAC. All we need to do is to transmit the unique pixels and lines of data from the output buffer, omitting the blanked data. The interpolator at the other end would receive this data and introduce the interpolated pixels into the data stream according to the format chosen. It is, of course, assumed that frame and line sync signals will have to be multiplexed on the data line for synchronization. Also the format type information should be multiplexed on the same line. Both receiving and sending ends would have similar blanking circuits. Figure 23 on page 64 indicates such a system.

The next section attempts to describe briefly the hardware implementation of the concepts evolved in this section.

3.3 HARDWARE IMPLEMENTATION OF THE OUTPUT BUFFER.

The block schematic of the output buffer is shown in Figure 24 on page 67. As already explained in the previous section, the output buffer consists of the following four blocks.

- Basic Output Buffer Block (OB).
- Blanking Block (OBB).
- Interpolater Block (OIB).
- Video D/A Converter Block (VDAC).

The interpolator block is outside the scope of this work, hence this block is not explained here in detail. The circuits of all the other blocks are included in Appendix B.2. The timing diagrams are included in Appendix B.3.

Basic Output Buffer Block. This block is identical to the input buffer and therefore the hardware design principles are not repeated here. Chapter 2.2 describes the hardware design of

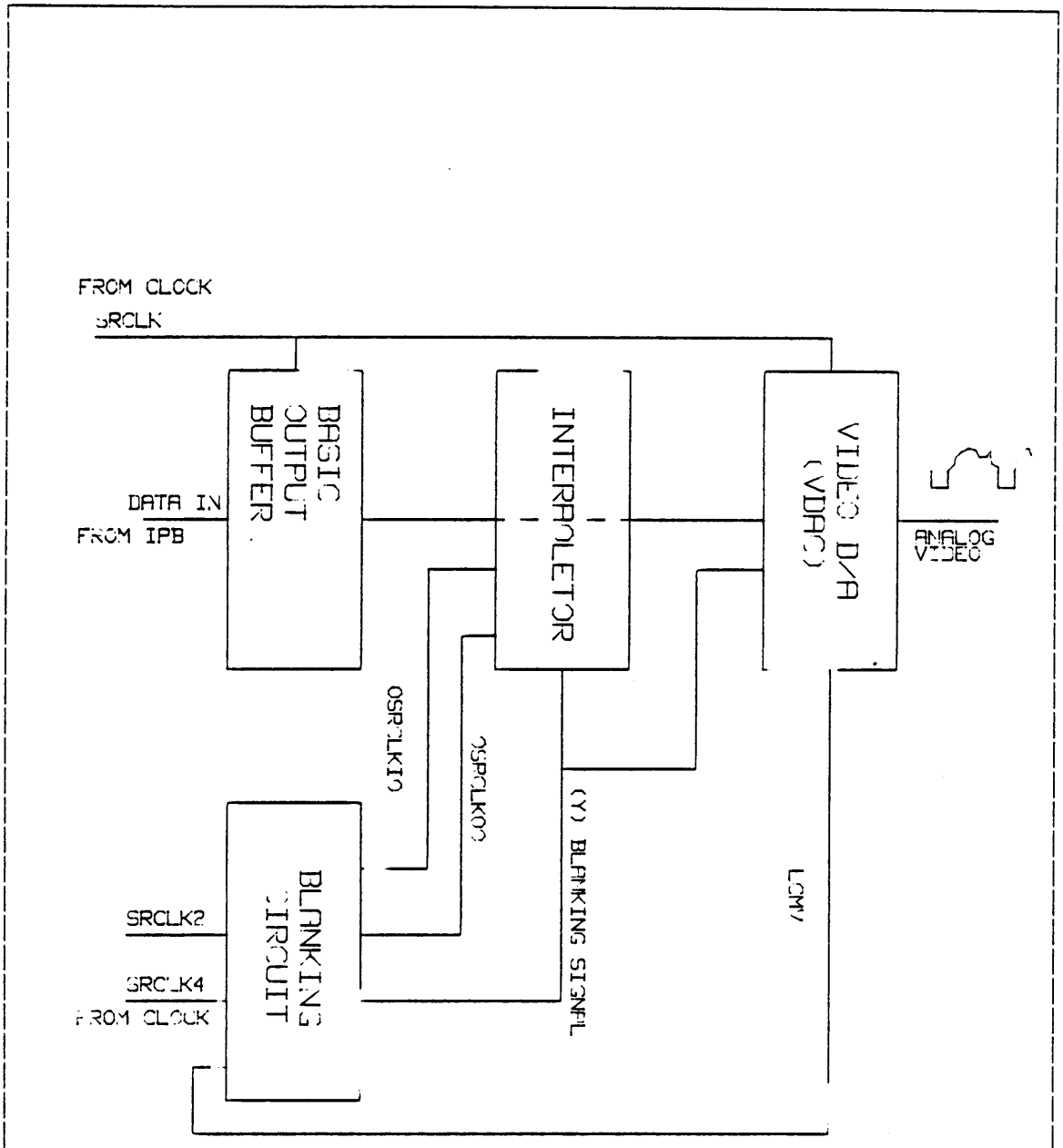


Figure 24. BLOCK SCHEMATIC OF THE OUTPUT BUFFER.

the input buffer in complete detail, as is true for the output buffer. The signal nomenclature is different for the output buffer, as indicated in the circuit diagrams. The PAL programs are similar; Appendix 2.3 has all the PAL programming details.

The output row addressing has to be somewhat altered when the OB is configured in the 128*128 mode. The data are shifted into the OB through the serial port at the pixel clock rate (2.016MHz). The IPB configured in the 128*128 mode sends out data on alternate pixel clocks pulses and on alternate lines. Since the output of the IPB is latched, every pixel gets clocked into the serial shift register of the OB twice, thus filling up the entire 256 bit shift register with 128 bit IPB data. Also, since the IPB sends data every alternate line, the OB serial register will be filled up completely with the value of the last pixel sent by the IPB during a valid line. This incorrect row of data is stored in every alternate line in the OB memory. This can be corrected by taking care that we do not output the incorrect lines at all; instead we output the correct line twice. This can be achieved simply by controlling the LSB of the OB row address. We have assumed in our design that even lines are valid. Hence we force the LSB of the output memory row address to remain low, thus ensuring that only even rows are transferred to the shift register and hence to the monitor. Each even row is also

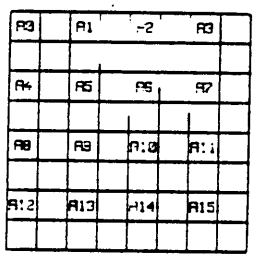
automatically transfered twice. Thus effectively 256 rows are transfered out to complete an entire frame. This conditioning of the row address is done in the blanking circuit PAL. Figure 25 on page 70 demonstrates this concept by looking at an 8*8 image at the output of the IPB, inside the OB memory and at the output of the OB.

3.3.1 OUTPUT BUFFER BLANKING BLOCK.

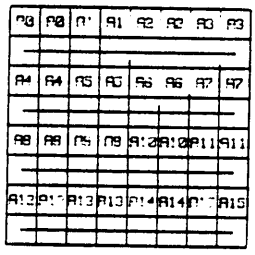
In the last section we have defined three different interlacing formats. They differ for the 256*256 and the 128*128 mode. We also have the normal 256*256 and 128*128 modes where no interlacing is done. Thus in all we have eight possible modes of operation. The modes can be selected by the system PC and thus the control can be coded on a 3 bit bus (C2, C1, C0). The following are the codes assigned to each format.

256*256 MODE.

- 000 256*256*1 non-interlaced.
- 001 128*256*2 line-interlaced.
- 010 128*128*4 dot-interlaced.
- 011 256*128*2 column-interlaced.



IPB OUTPUT



OUTPUT MEMORY CONTENTS

OUTPUT MEMORY AS SEEN BY
VDPC OR INTERPOLETOR

Figure 25. OB MEMORY ORGANIZATION (128*128 MODE).

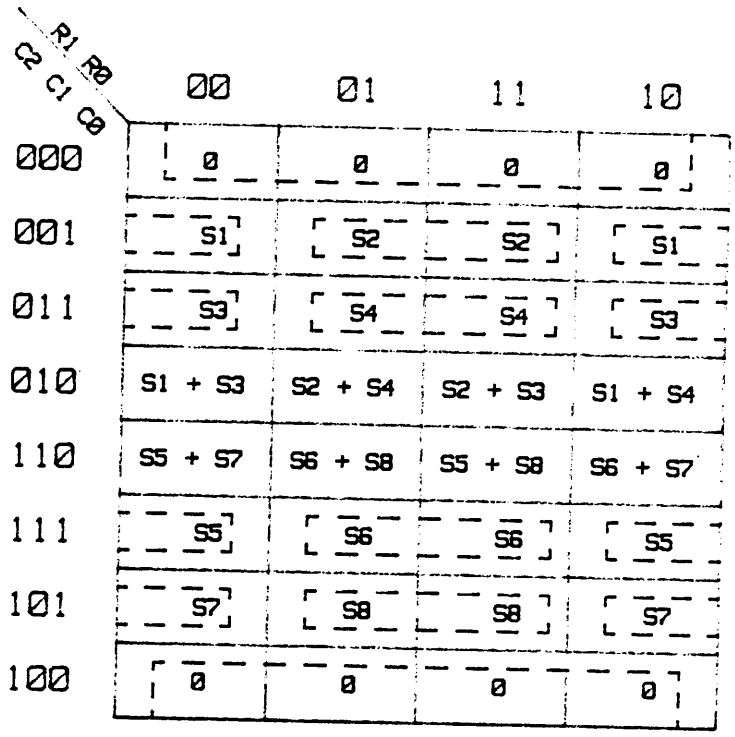
128*128 MODE.

- 100 128*128*1 non-interlaced.
- 101 64 *128*2 line-interlaced.
- 110 64 * 64*4 dot-interlaced.
- 111 128* 64*2 column-interlaced.

As seen in the last section, we need to keep track of up to four frames to be able to generate the different blanking sequences. For this purpose a 2 bit counter is created in one of the blanking circuit PALs. This counter is clocked by a video frame pulse (FCMV2) from the VDAC circuit. The counter output (R0,R1) along with the format control signals C0, C1 and C2, also line marker (LCMV), Serial output clock divided by two (SRCLK2), serial output clock divided by four(SRCLK4) and internally generated states Q1 and Q2 are needed to generate the blanking signal to the interpolator.

Figure 26 on page 72 shows the Karnaugh map used to minimize the expression for the blanking signal (Y). The minimized expression is as follows:

$$y = LCMV^*/C2^*/C1^*C0^*[R0 \oplus Q1] + LCMV^*/C2^*C1^*C0^*[R0 \oplus SRCLK2]$$



LEGEND:

S1 = LCMV * Q1

S2 = LCMV * /Q1

S3 = LCMV * SRCLK2

S4 = LCMV * /SRCLK2

S5 = LCMV * SRCLK4

S6 = LCMV * /SRCLK4

S7 = LCMV * Q2

S8 = LCMV * /Q2

Figure 26. BLANKING SIGNAL KARNAUGH MAP.

+ LCMV*C2*C1*C0*[R0 ⊕ SRCLK4]

+ LCMV*C2*/C1*C0[R0 ⊕ Q2]

+ LCMV*/C2*C1*/C0*/R1[(R0 ⊕ SRCLK2) + (R0 ⊕ Q1)]

+ LCMV*/C2*C1*/C0*R1[/(R0 ⊕ SRCLK2) + (R0 ⊕ Q1)]

+ LCMV*C2*C1*/C0*/R1[(R0 ⊕ SRCLK4) + (R0 ⊕ Q2)]

+ LCMV*C2*C1*/C0*R1[/(R0 ⊕ SRCLK4) + (R0 ⊕ Q2)]

This equation is implemented by two 16R4 PALS. For convenience of PAL programing the above equation is split up into four terms: y1, y2, y3, y4. Appendix B.3 has the details of these equations. These are generated in the Format 2 PAL and are added together in the Format 3 PAL. The Format 1 PAL generates the video frame counter signal R0 R1. It also controls the low order bit of the row address of the basic output buffer, allowing it to pass through in the 256*256 mode and clamping it low in the 128*128 mode. All three PALs are used for the general inversion of the various signals. Figure 27 on page 74 shows the block schematic of the blanking circuit.

3.3.2 VIDEO D/A CONVERTER CIRCUIT.

The video D/A converter takes in digital data from the interpolator or the basic output buffer and converts it to RS 170 2:1 interlaced analog video. This conversion is done by the Telmos (TML 1840)[10] video D/A converter. The sync and composite blanking signals are generated by the RCA CD22402 [11] sync generator. This chip also provides the

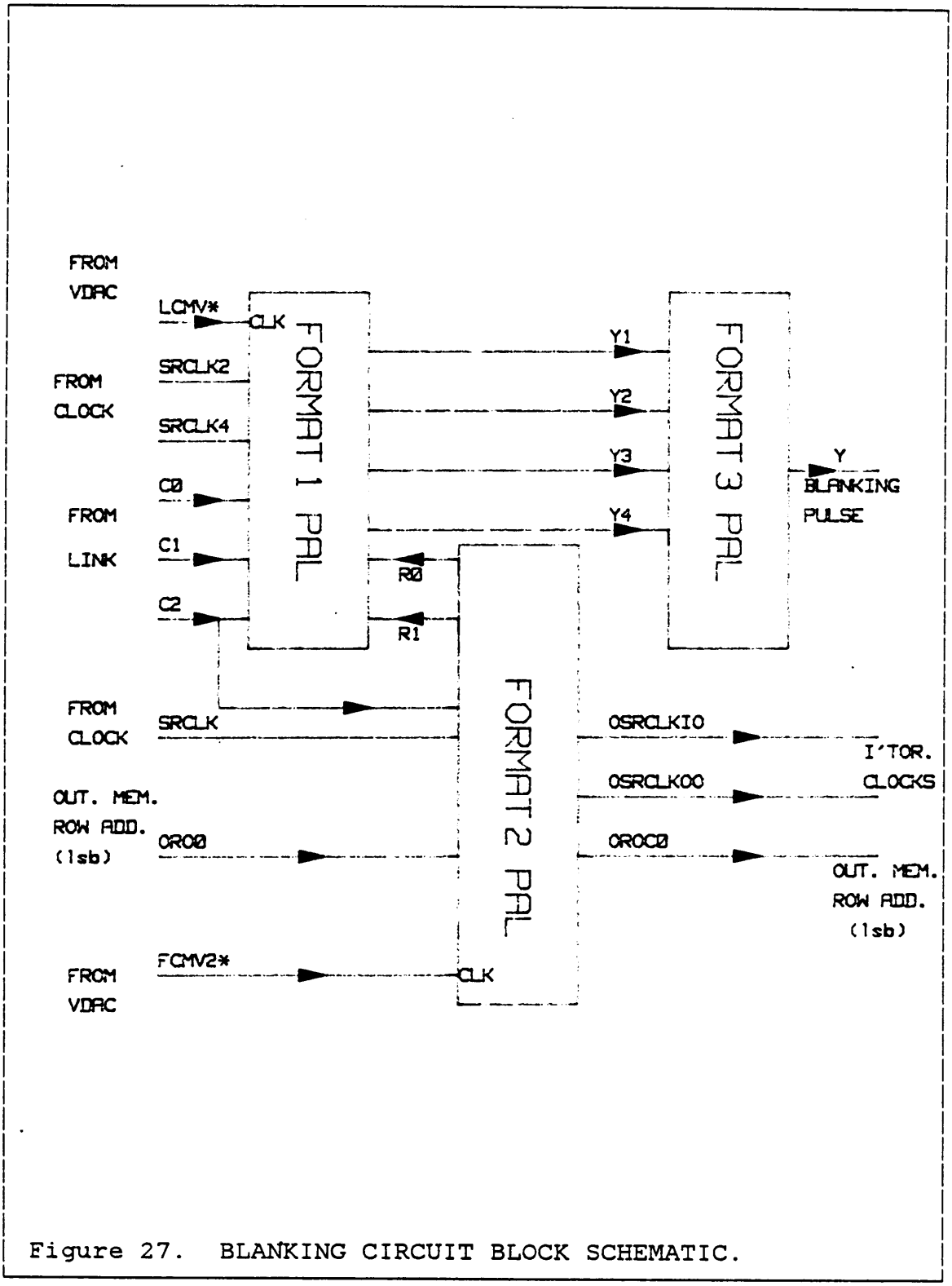


Figure 27. BLANKING CIRCUIT BLOCK SCHEMATIC.

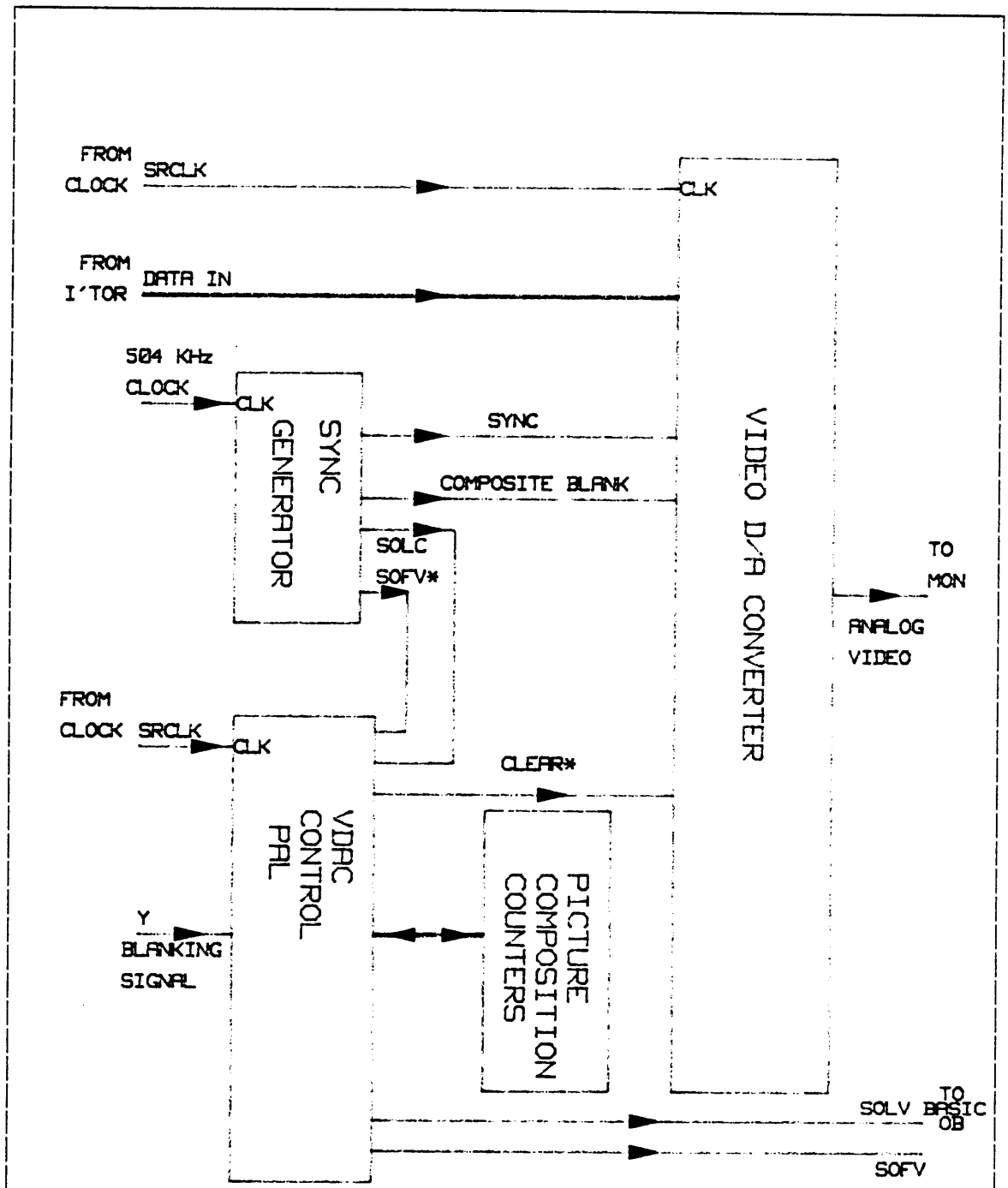


Figure 28. VIDEO D/A BLOCK SCHEMATIC.

start of line (SOLV), the start of field pulse (FCMV) and also the start of frame pulse on every alternate field (FCMV2). These signals are supplied to the output buffer control circuits to generate the necessary line and frame markers. The frame pulse is supplied to Format 2 control PAL in the blanking circuit. This clocks the two bit counter to generate the R0 R1 signals described in the last section. Figure 28 on page 75 shows the block schematic of the video D/A converter. A 16R4 video PAL is used to generate control signals to control the three counters shown. Their function is described below.

Counter 1 is used to generate a delay (LCMC) between the start of active video line and the start of display on the monitor. The delay is controlled by the DIP switch bank 1 which loads the counter with the value corresponding to the desired delay. The screen is blanked out during this delay by the clear line to the VDAC chip. In addition to this delay the counter 2 is used to generate a blocking pulse (LCML) after the actual display has started. This allows us to blank out the initial pixels if desired. During this blocking pulse the clear line to VDAC is held low. The number of pixels to be blocked out is controlled by the DIP switch bank 2 (e.g. if first 10 pixels are to be blanked out the DIP switch should be set to a value (255 - 10) = 245 and so on). The third counter controls the clear line of the VDAC and is valid only

when LCMC and LCML are low. It is set by the DIP switch 3. Since we need to display a maximum of 256 pixels and since we desire to block a few inital and final pixels in each line because of edge effects in the IPB toward the begining and end, we must always set the DIP switch such that /clear remains high (valid) for less than 256 output pixel clocks. Hence only the desired pixels will be displayed. Note here that when /clear goes low the video monitor will be blanked. Appendix B.2 has the detailed timing diagram for the above described control.

The need to delay the start of display from the actual start of active video line is to allow the display of the 256*256 square image on a rectangular monitor which has an aspect ratio of 4 to 3. The fast pixel clock ensures that the 256 pixels are displayed in 75% of the active line time. Thus if the delay is set correctly we should be able to have equal front and rear black bands, and a square undisturbed image would result. Figure 29 on page 78 shows the above video composition pictorially in relation to the various composition pulses (LCMC LCML and /clear).

The TML 1840 accepts 8 bit resolution per pixel; however, since the IPB generates a single bit output all the input bits of TML 1840 are tied together. Hence only 0 or 255 (i.e. black or white) are the only two levels received by the VDAC.

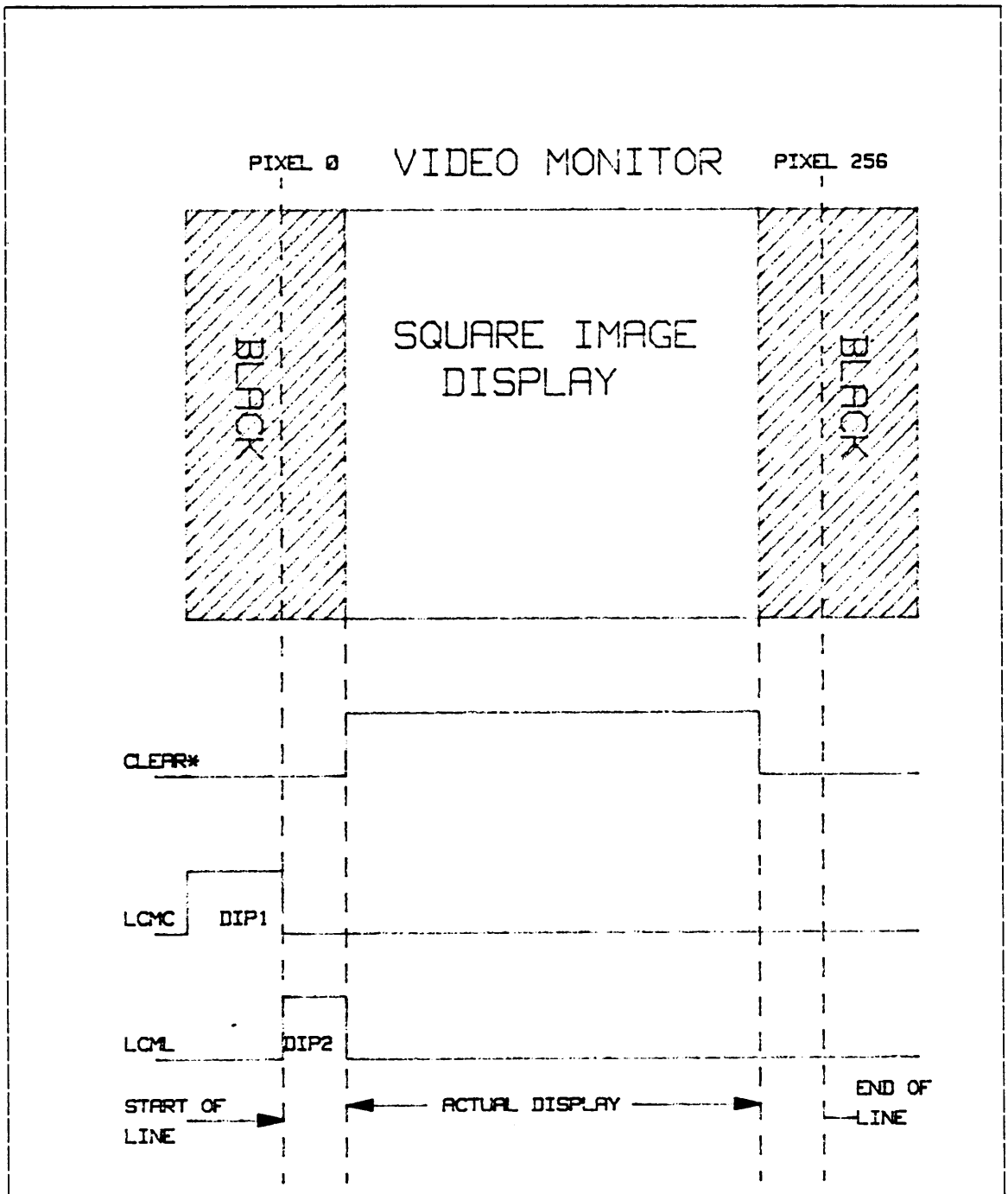


Figure 29. VIDEO COMPOSITION.

The VDAC converts the digital signals to analog with the analog voltage levels compatible for direct input to a 75 ohm RS170 2:1 interlaced video monitor. The sync and the composite blank pulse required are provided by the sync generator.

This completes the description of the output buffer along with the blanking and the VDAC circuits. Appendix B has all the circuit details, timing diagrams, and the PAL programs for these circuits. The next chapter covers the generation of the system clock.

4.0 SYSTEM CLOCK

4.1 GENERAL REQUIREMENTS OF THE SYSTEM CLOCK

It is obvious from the last two chapters that there is a great need for a sophisticated clock for the Telesign system. The purpose of this chapter is to define these requirements clearly and to indicate how these requirements will be met in hardware. Appendices C.1,C.2 and C.3 contain the detailed timing diagrams, circuit design and the PAL programming details for the system clock.

The system clock must perform the following functions.

- Generate the IPB clocks for the Telesign system.
- Generate the input/output buffer clocks.
- Generate the camera clock.
- Generate the link clocks.

The following sections take a closer look at each of the above requirements. Figure 30 on page 81 shows the inter-

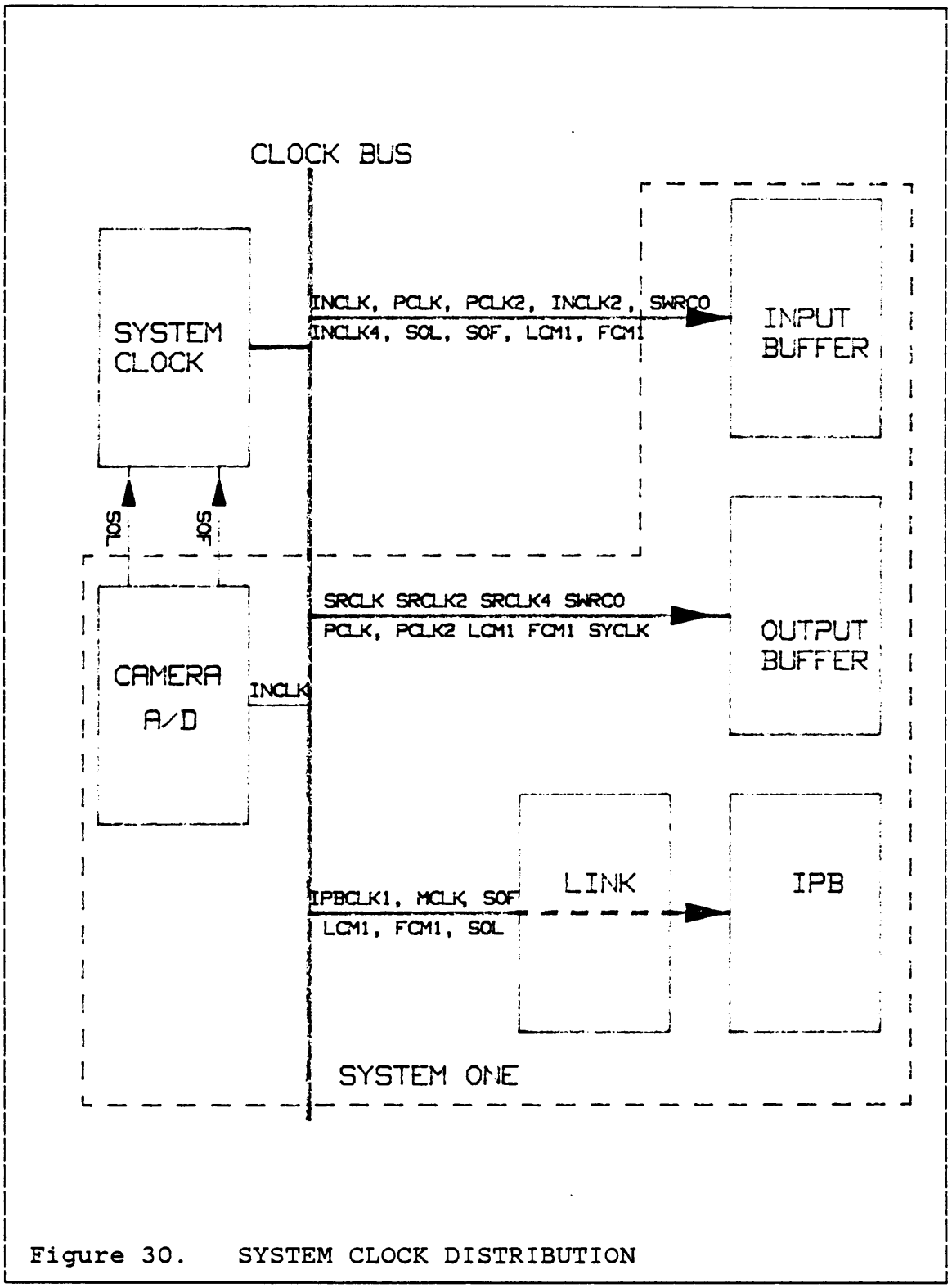


Figure 30. SYSTEM CLOCK DISTRIBUTION

connections between the clock board and the rest of the Telesign system.

4.1.1 IPB CLOCK

The IPB, as explained in Chapter 2 must be clocked 8 times faster than the pixel clock. Also the IPB should be able to stop processing during the dead times between lines and frames. Thus the IPB clock should be on during a valid line and valid frame. The valid line will have exactly 256*8 clock pulses, and the valid frame will have exactly 256 lines of the above clock pulses. The start of a valid line (SOL) and a valid frame (SOF) are provided to the clock board by the camera in the asynchronous mode of operation, while the clock board itself generates the SOL and SOF in the synchronous mode. These modes are explained in detail in section 4.3.

Since the complete Telesign system is a two-way communication system we will have have two IPB boards running simultaneously. Each will have a camera associated with it. In general the SOL & SOF pulses of each camera will be asynchronous with each other. Hence it is obvious that the line markers (LCM) and frame markers (FCM) generated for each IPB will be different. (Note: The line and frame markers are high level signals to indicate a valid line or frame respectively.) Since the two IPB clocks are generated from their respective

LCMs and FCMs they will in general be different. The IPB clock is basically an 'and' function of the LMC, FMC, and the MASTER CLOCK (MCLK) pixel clock*8 .

The above describes the IPB clock for the processing of 256 pixels and 256 lines per frame (256*256); however, we have indicated in the previous two chapters that the IPB will also operate in the 128 pixel and 128 lines per frame mode (128*128). These two modes must be transparent to the IPB. Hence the best way to operate the IPB in the 128*128 mode is to modify the IPB clock. This can be done by suppressing the IPB clock completely for every alternate line and every alternate pixel in the valid line. This will force the IPB to process every alternate pixel per line and every alternate line per frame. Thus in a 256*256 frame only 128*128 pixels will be processed by the IPB. Since the IPB clock is 8 times faster than the pixel clock, we must supress alternate bursts of 8 clock pulses in a valid line of the 128*128 mode. Also every alternate line must be completely blanked out. The above can be achieved quite simply by 'ANDing' the pixel clock divided by 2 signal (PCLK2), the line marker divided by 2 signal (LCM2:1), and FCM with the master clock (MCLK). The high order bit (C2) of the 3-bit format control signals from the link board determines the mode of operation for the IPB. The appropriate IPB clocks are generated on the clock board by 'ANDing' the above mentioned signals in accordance

with the mode signal. Appendix C.1 contains the timing diagrams for the IPB clock generation in both modes of operation.

4.1.2 INPUT AND OUTPUT BUFFER CLOCK REQUIREMENTS.

4.1.2.1 INPUT BUFFER REQUIREMENTS.

The input buffer needs to load the incoming video data into the serial register before transferring it to the input memory. This clock must be the same as the camera clock (INCLK). The input buffer must also know the start of a valid line and start of a valid frame for this incoming data from the camera. These SOL and SOF pulses are supplied by the camera or the camera A/D converter circuit. The input buffer output serial clock must be in sync with the IPB pixel input rate, thus the pixel clock (PCLK=2.016MHz) is required. The LCM and FCM signals are 'ANDed' to this pixel clock so that the data clocked out of the input buffer is synchronized with the data clocked in to the IPB. The INCLK signal divided by 2 (ICLK2) and the PCLK signal divided by 2 (PCLK2) are also required by the input buffer to clock the refresh address counters of the input and output memory respectively. The REFRESH FRAME pulse (SWRCO) is also required to swap the input and the output memory. The Telesign system requires that SWRCO pulse be variable from 25Hz to 10Hz to enable effective refresh

rate control. This pulse is generated by clocking an 8-bit counter with a 2625 Hz signal. The counter is used in the count-up mode. The SWRCO pulse is generated when the count reaches the maximum value of 255. The counter starts counting up from the 8-bit value loaded into it either via the 8-bit data bus from the link board or from a set of eight DIP switches on the clock board. The SWRCO pulse rate can thus be varied by loading different values to the counter. The same SWRCO signal is required by the output buffer.

4.1.3 OUTPUT BUFFER REQUIREMENTS.

Similarly to the input buffer, the output buffer requires the LCM and the FCM signals as well as the PCLK signal to generate the serial shift register clock in order that the IPB output data be correctly latched into the serial register of the output buffer's input memory. The INCLK signal is required to generate the output memory serial clock (SRCLK). Also PCLK2 and INCLK2 signals are required to clock the input and output memories refresh address counters.

The VDAC circuit of the output buffer requires a 504.00 Hz clock to generate all the necessary output video control signals, i.e., sync, composite blanking, etc. This clock has to be supplied by the clock board.

4.1.4 CAMERA CLOCK REQUIREMENT.

The camera A/D converter circuit requires input clock signal to digitize the analog signal. Since an active line duration for a standard video camera is 63 μs., a clock is required that will sample at least 256 pixels in that time. In the Telesign system the INCLK signal 5.367 MHz is used as the camera A/D clock. This is similar to the serial input clock of the input buffer.

For a C.C.D. camera the clocking rate is variable. In the Telesign system the IPB pixel clock is chosen to be this clock. It should be noted here that the serial input clock will be the same as the camera clock to enable synchronous loading of the data into the input buffer.

4.1.5 LINK CLOCK REQUIREMENTS.

The link board in general does not require a clock signal to operate it. The link derives all the necessary clocks and control signals from the PC. However, the IPB clock, SOL and SOF signals are made available to the link for routing to the IPB. The link can thus interrupt these signals to the IPB during IPB initialisation and also during debugging.

4.2 CRYSTAL FREQUENCY SELECTION AND MODES OF OPERATION.

4.2.1 FREQUENCY SELECTION.

The choice of crystal frequency is based on the following considerations:

- The RS 170 requires a line frequency of 15750 Hz.
- The IPB is required to operate at a frame rate of about 25 Hz.
- Aspect ratio compensation to fit a 256 * 256 square image on a monitor with a 4:3 width to height ratio.

The 15750 Hz signal and its multiples of two are required to clock the various circuits of the A/D converter. The output buffer output data rate is such that we are able to clock out 256 pixels, i.e., one whole video line in 75% of the time available to display a line. This counteracts the 4:3 aspect ratio of the monitor. The serial clock frequency is calculated as follows.

SRCLK = (15750 Hz * 256) / 0.75= 5.376MHz.

The condition that the IPB clock should operate at about 25 F/s should be met as closely as possible. As explained earlier, 15750 Hz clock is definitely required; we therefore look at powers of two of this frequency for the IPB pixel clock. A frequency of 2.016 MHz is one frequency which meets our requirement for the IPB frame rate. The IPB frame rate can be calculated as follows.

IPB F/s.= Pixel clock / (lines per frame* pixels per line)

(Note: The dead time between lines and pixels is included in the above calculations. A reasonable assumption would be 16 dead pixels per line and 16 dead lines between frames for a 256 * 256 image.)

IPB F/s. = 2.016 MHz / {(256 + 16) * (256 + 16)}

= 2.016 MHz / (272 * 272)

= 27.249 F/s.

The choice of 2.016 MHz as the pixel clock meets the IPB frame rate requirement. If desired the IPB frame rate can be reduced to exactly 25 F/s by suitably increasing the dead time between lines and frames.

This choice of the pixel clock automatically decides the IPB clock to be 16.128 MHz because the IPB clock is required to be eight times faster than the pixel clock.

Finally, the choice of the crystal frequency can be made from the two high frequency clocks needed, i.e., 16.128 MHz (IPB Clock) and 5.376 MHz (SRCLK). If we divide the IPB clock by SRCLK we see that the IPB clock is exactly three times faster than SRCLK. It was thus decided to use a crystal of 10.752 MHz (twice the serial clock) operating it in the third harmonic mode [12]. Thus the maximum available frequency is 32.256 MHz, from which 16.128 MHz and all other submultiples of two can be derived using a cascade of two 8-bit counters. From the 10.752 MHz clock, the 5.376 MHz clock and all other required frequencies of two can be derived by cascading two 8-bit counters. The frequency of 2625 Hz is one such clock. This is used to generate the refresh frame signal. Figure 31 on page 90, indicates some of the signals generated on the clock board.

4.2.2 MODES OF OPERATION

Apart from generating all the above clocks, the clock board must generate the line and frame markers (a valid line and a valid frame) for the incoming video signal. There are two possiblities:

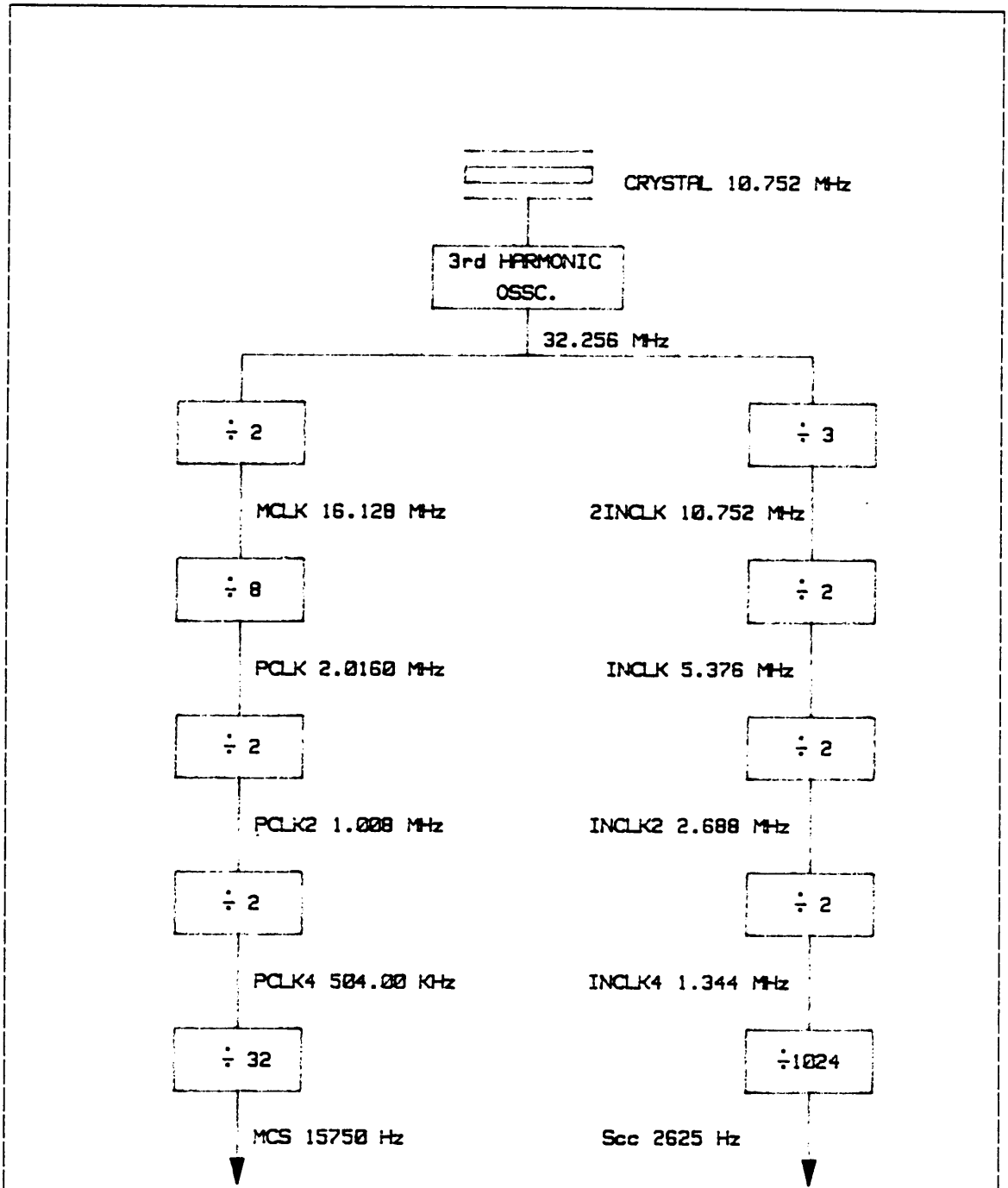


Figure 31. FREQUENCY CHAIN.

- Synchronous mode: The clock board receives the start of line and start of frame from the camera. If the camera is a standard video camera the clock board cannot use these SOL and SOF pulses to generate the LCM and FCM respectively for the IPB and the buffers because these will be too fast for the slow data rate of the IPB. Thus the clock board must generate its own line and frame markers (LCM andFCM) which are compatible with the slower clock of the IPB (pixel clock). Since the Telesign system consist of two IPB's, two input and output buffers, the same SOL and SOF pulses are used to generate the LCM and FCM for both the systems. The two systems will have lines and frames synchronized with respect to the IPB. Thus this can be termed a synchronous mode of operation.

- Asynchronous mode: When the input camera runs off the pixel clock (e.g. CCD camera with external clock input), then the input video data rate and hence the SOL and SOF pulses from such cameras can be used to generate the line and frame markers of the IPB. However, we will no longer have synchronization between the two IPB's of the system. Since the SOL and SOF generated by their respective cameras will in general be out of phase, we call this mode asynchronous.

In the synchronous mode the clock board generates its own SOL and SOF pulse by using two 8-bit counters. These counters are clocked by pixel clock divided by 2 and line clock divided by 2. The dead time between lines and frames can be set by using the dip switches which load these counters. Appendix C.1 has the timing diagrams. Appendix C.2 shows the circuit used to implement this idea.

In the asynchronous mode the incoming SOL and SOF from the camera are used to generate the IPB line and frame marker. The choice of using the internally generated or externally generate SOL and SOF is made by the SYNC signal on a PAL which multiplexes the internal and external SOL and SOF signals.

4.2.3 HARDWARE IMPLEMENTATION OF THE CLOCK BOARD

The clock board can be partitioned into the following blocks:

- The Frequency Generation Block
- Internal SOL and SOF Generation Block
- LCM and FCM Generation Block
- Refresh Frame Pulse Generation Block

- Control Signal and IPB Clock Generation Block

Figure 32 on page 94 shows the interconnection of these blocks to form the clock board. The operation of each of these blocks is rather self-explanatory and clear from the timing diagrams and the circuit diagrams in Appendix C, and will not be explained in detail here. However, the following points highlight some of the important features incorporated into the clock board.

- As explained earlier, a 10.752 MHz crystal is used. However, the third harmonic of this is used as the basic clock. The fundamental is filtered out [12].

- Three PALs are used. Registered PALs are used to generate the line and the frame markers. An unregistered PAL is used to generate the IPB clocks as well as to control the Refresh Frame Pulse generation block. This PAL is also used to choose the synchronous or the asynchronous mode of operation.

- 74AS869 8-bit counters are used to generate the various clock signals which are a submultiple of either 32.256 MHz or 5.376 MHz.

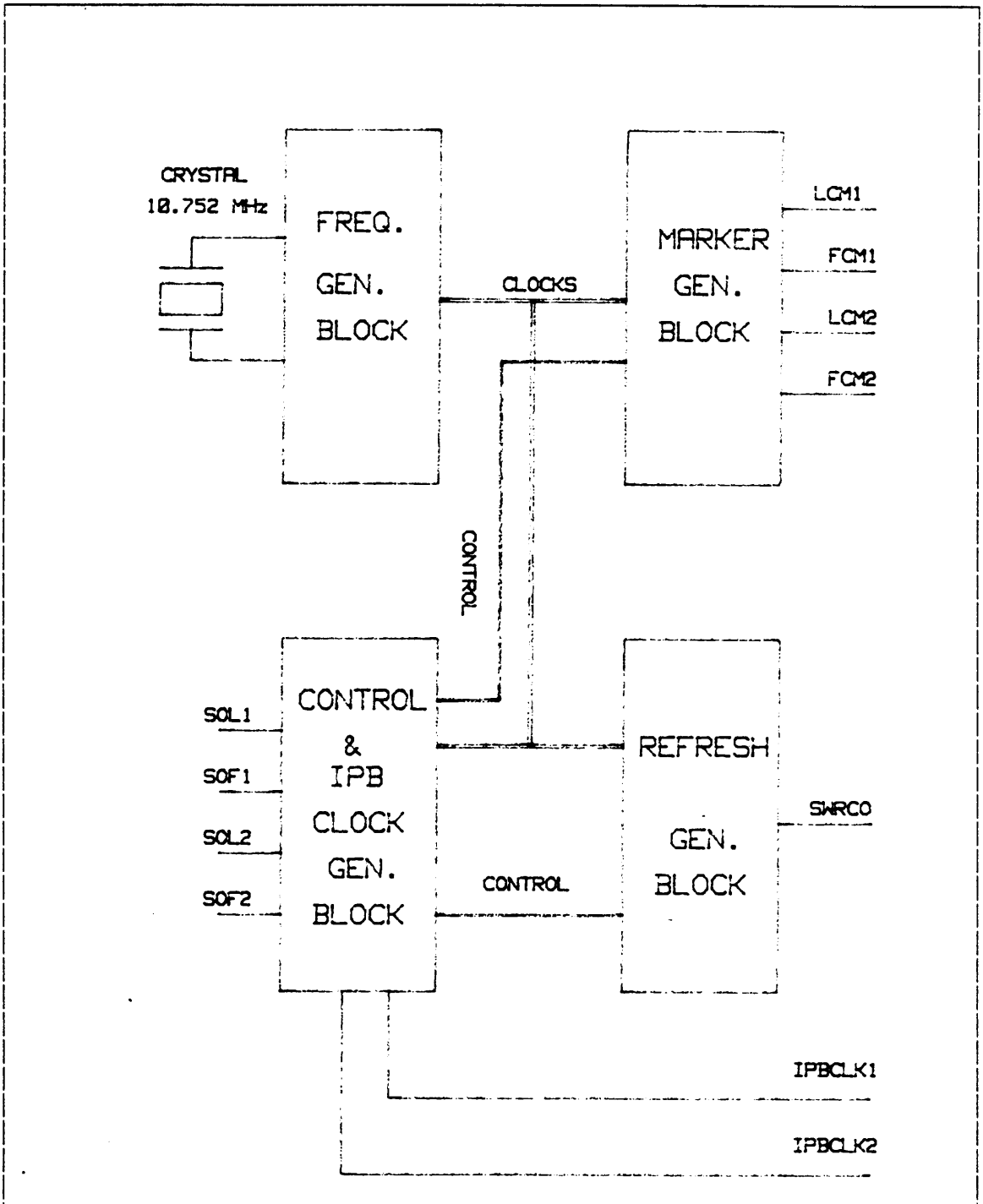


Figure 32. CLOCK BOARD SCHEMATIC

- The dead time between the lines and frames is controlled by a set of DIP switches on the clock board.

- The Refresh Frame rate can be controlled by an 8-bit data bus which loads the appropriate counter or by an 8-bit wide DIP switch which performs the same function. In the Telesign system the data bus is controlled by the PC.

This concludes the salient features of the clock board design. In the next chapter we conclude this work and also attempt to highlight a few of problems faced in designing and implementing these circuits.

5.0 PROBLEMS ENCOUNTERED AND CONCLUSIONS.

5.1 PROBLEMS ENCOUNTERED.

This project involved the design of the three basic subsystems of the telesign system which have already been described in the preceding chapters. As with any design project the problems involved were numerous. In this section an attempt is made to list these problems as well as the solutions that had to be developed to solve them.

After the complete system was incorporated we discovered a very disagreeable phenomenon. The subject on the screen had multiple tears whenever it moved. The tearing was more noticeable at higher frame rates. The tearing increased whenever the speed of motion increased. This was totally unexpected since only one tear was anticipated from the analysis in Chapter 2.

Analysis of the propagation of data through the input buffer the maximum is presented in Figure 33 on page 97, Figure 34 on page 98 and Figure 35 on page 99. The frame data rate from the camera A/D is 60f/s while the IPB frame rate is 25f/s. The output data rate is again 60f/s. In the analysis presented in the above diagrams we start by assuming that a

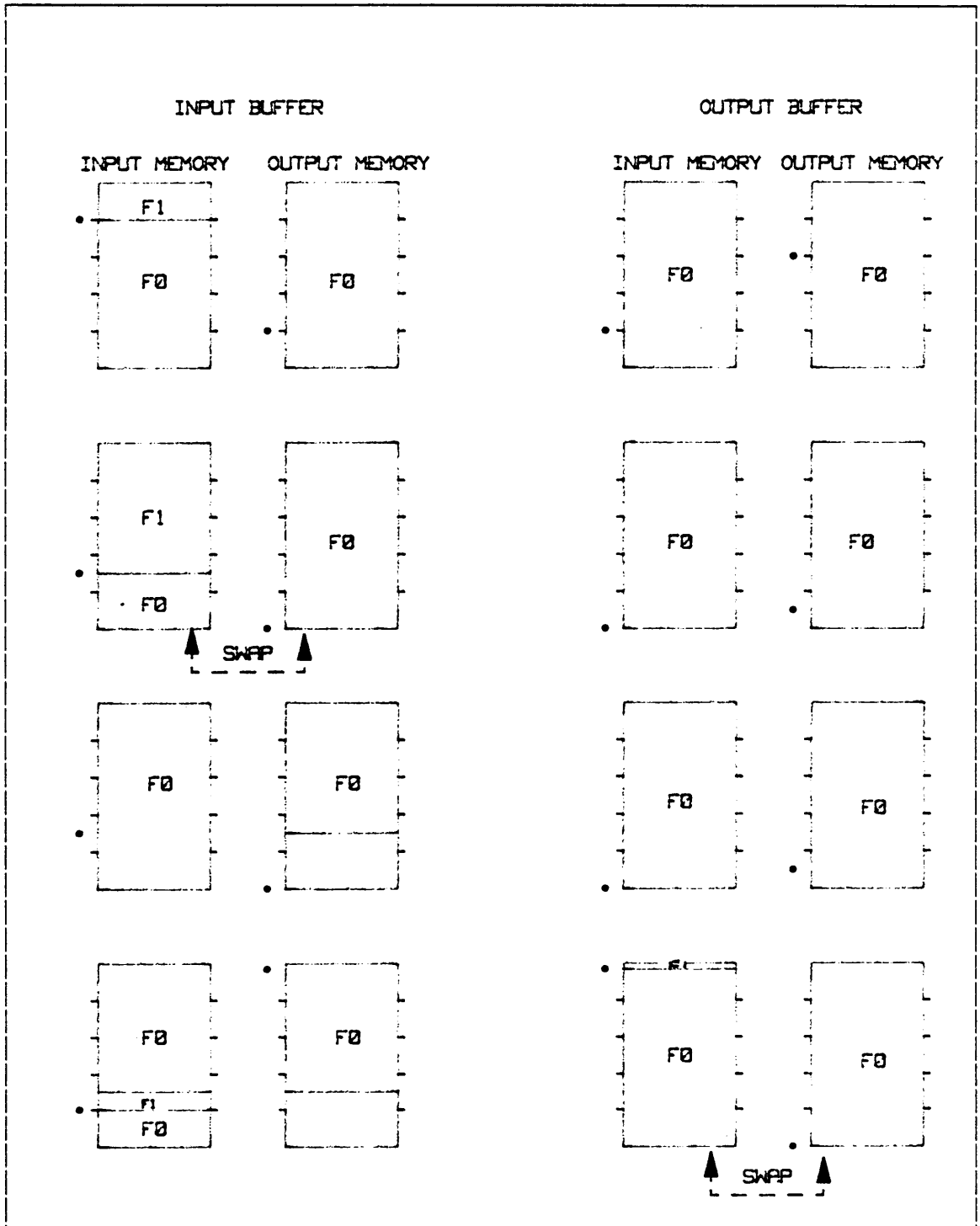


Figure 33. DATA FLOW I/O BUFFERS

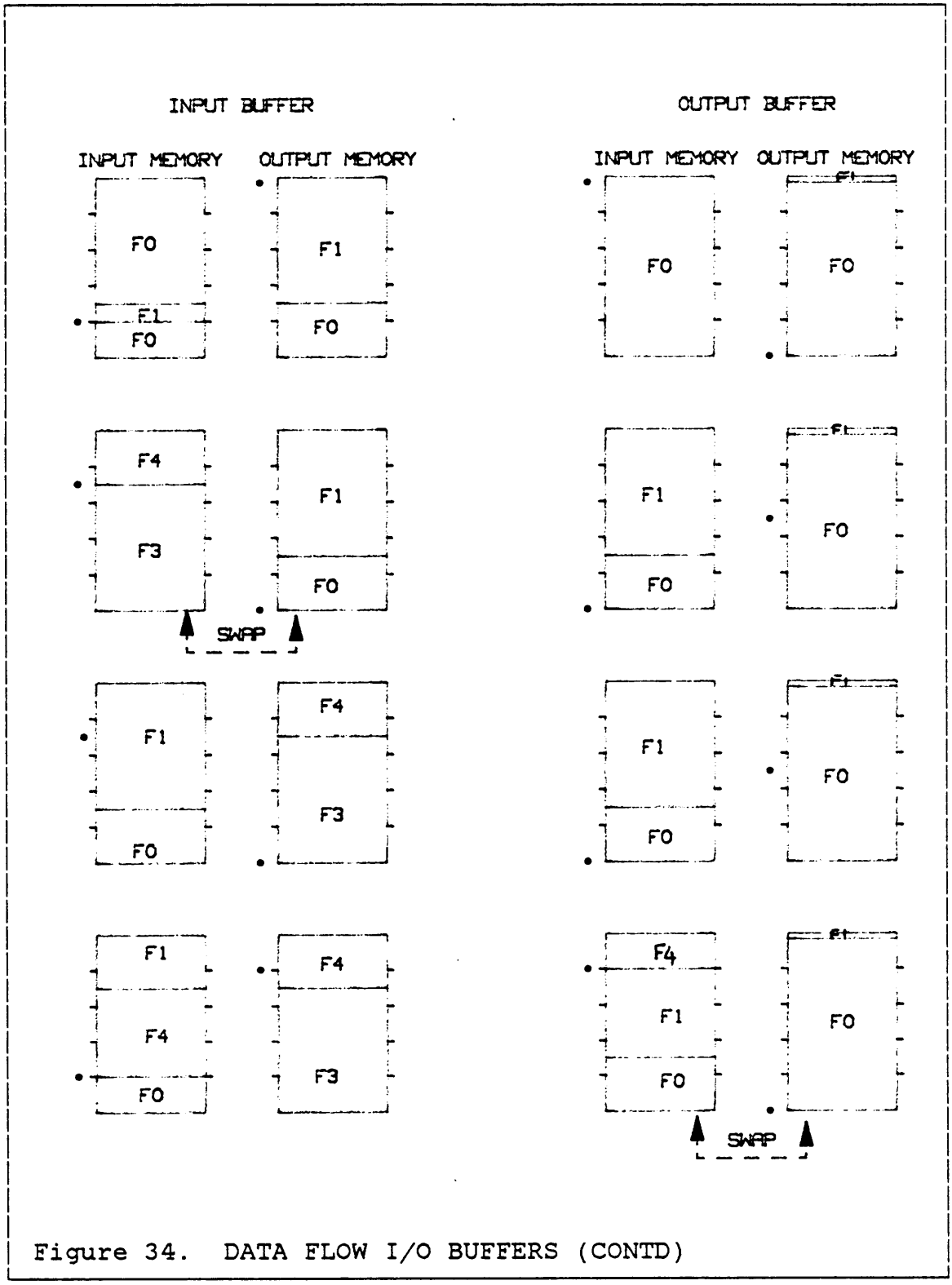


Figure 34. DATA FLOW I/O BUFFERS (CONTD)

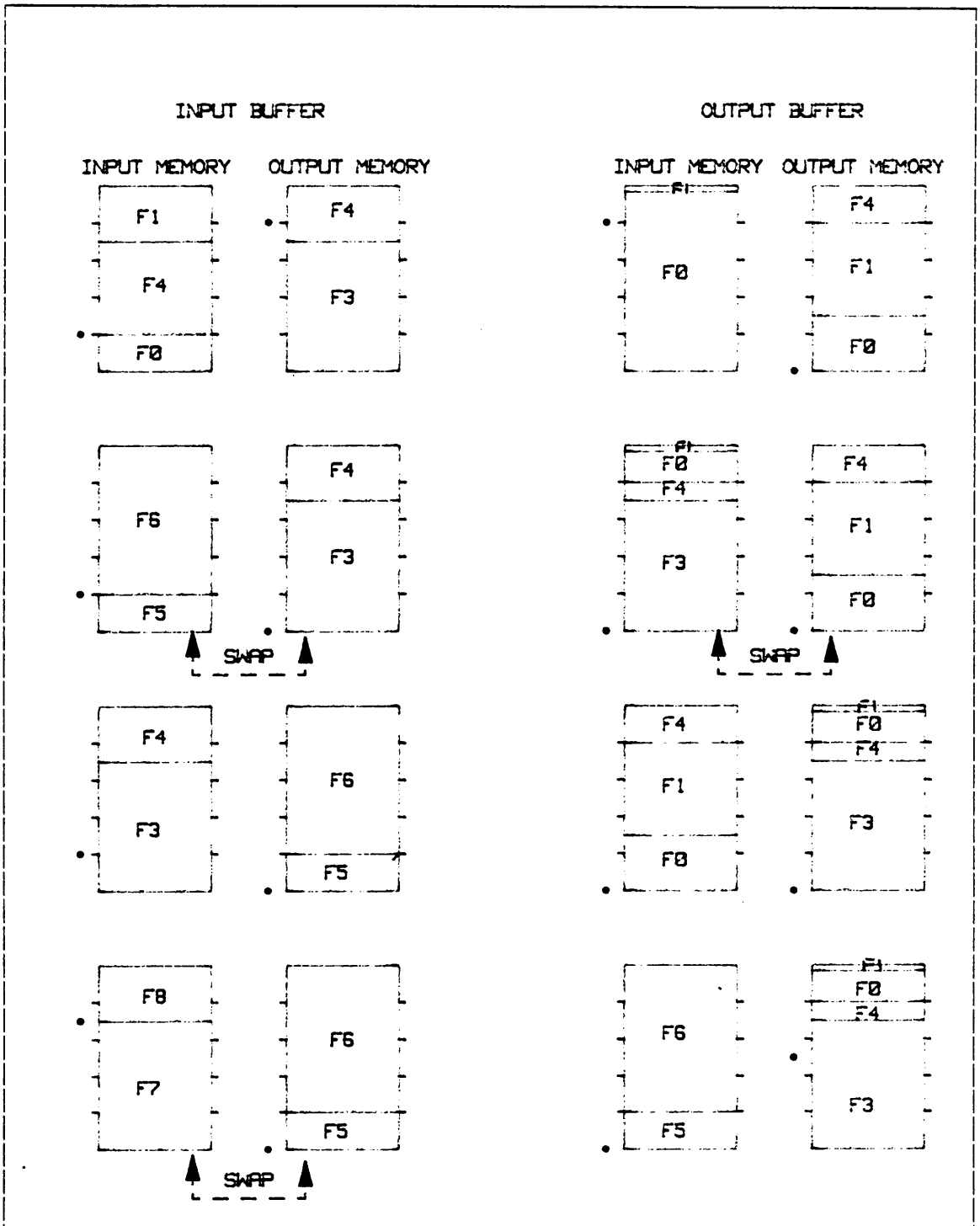


Figure 35. DATA FLOW I/O BUFFERS (CONTD)

still image has propagated through the input and output buffers. The markers next to each memory indicate the row address of that memory. We see from the diagrams that parts of many frames are presented to the output monitor, which are responsible for the tearing of the subject. The conclusion that can be drawn from this analysis is that the output buffer compounds the tearing problem. At worst the IPB is presented with an image consisting of consecutive frames. We still maintain that the tearing effect due to this phenomenon will be negligible. We must therefore modify the output buffer such that the IPB image is presented to the monitor without any further tearing.

A triple memory scheme instead of the present dual memory scheme for the output buffer will solve the problem. To prevent tearing of image at the input buffer we suspend reading data into the input memory after it has grabbed a frame. We start reading a new frame on the SOF pulse after the memory switch has taken place. This will ensure that whole images are presented to the IPB. Figure 36 on page 101 is the event schematic for the input buffer. The triple memory at the output is configured such that an additional memory isolates the input and output memory. The working of triple memory scheme is indicated in Figure 37 on page 102, Figure 38 on page 103 with the help of timing diagrams. The assumptions are the same as in the analysis of the dual memory presented

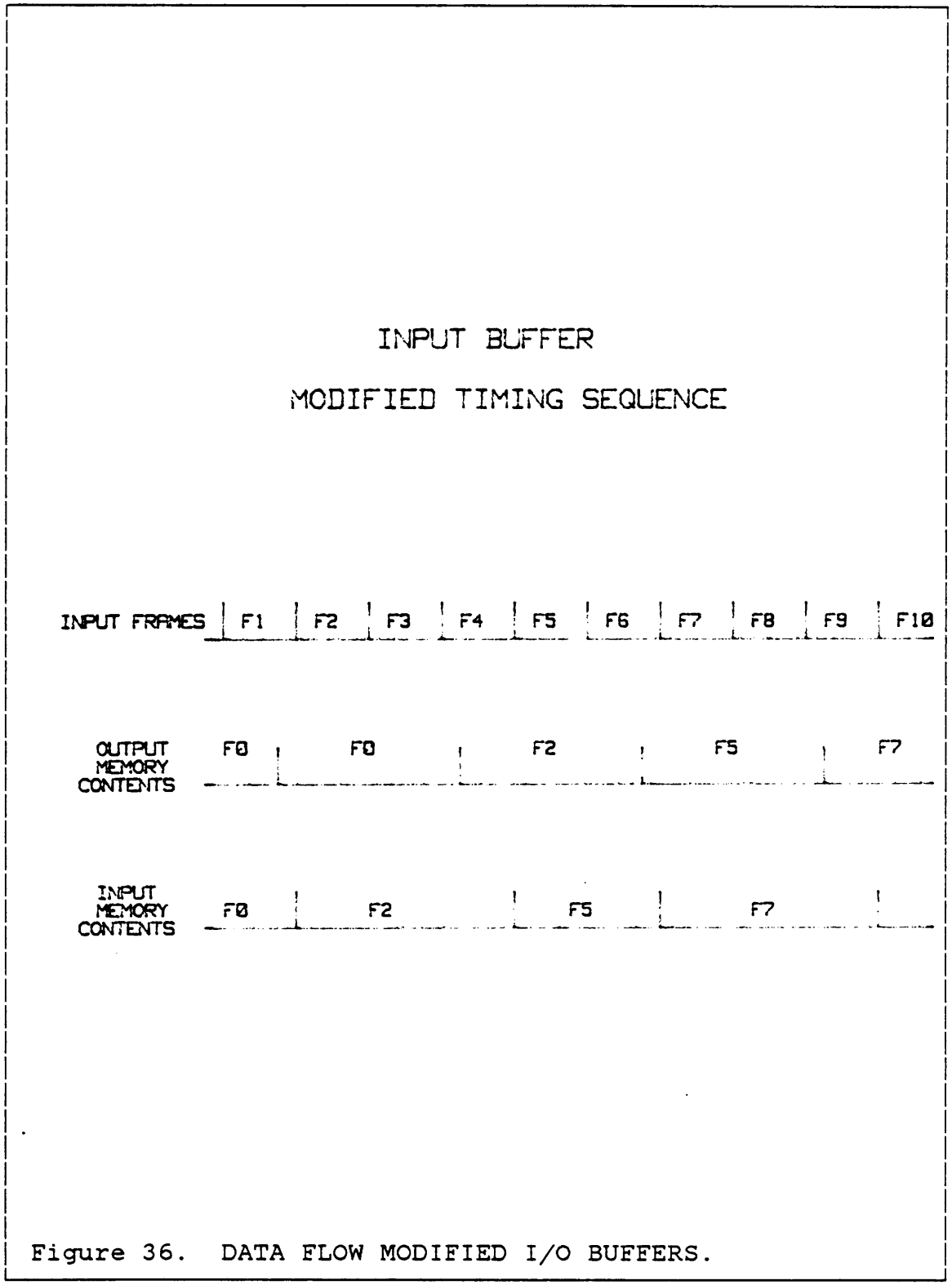


Figure 36. DATA FLOW MODIFIED I/O BUFFERS.

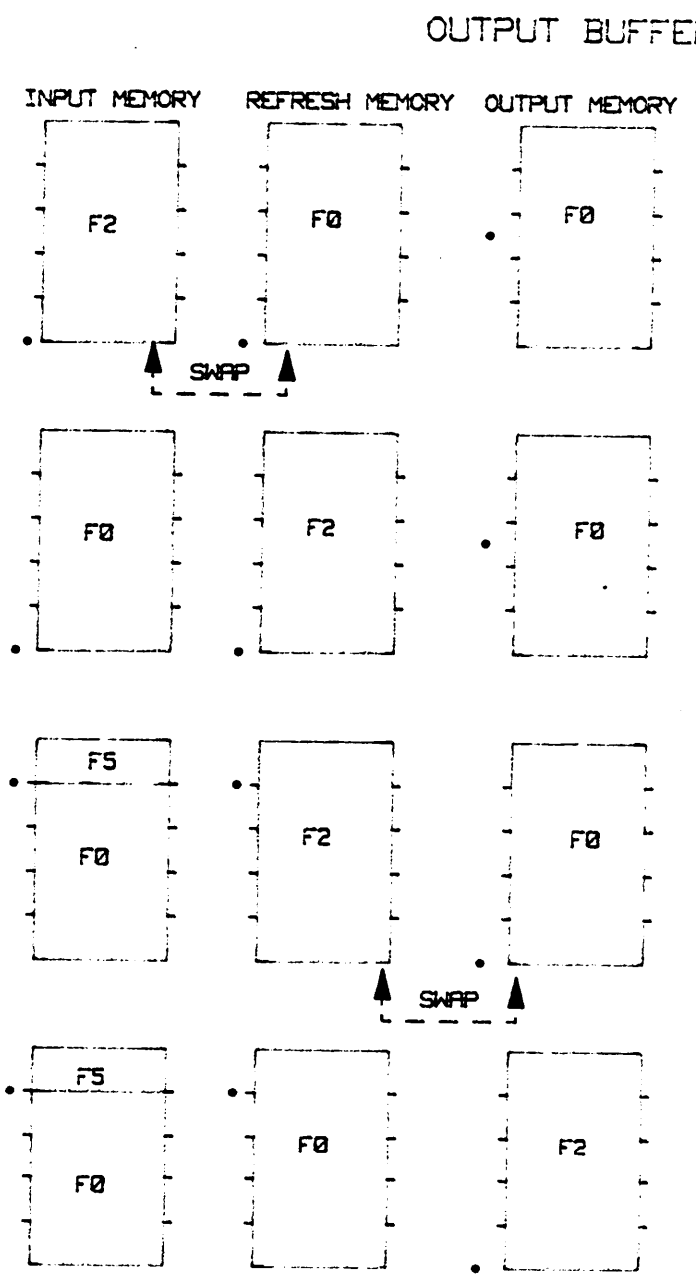


Figure 37. DATA FLOW MODIFIED I/O BUFFERS (CONTD)

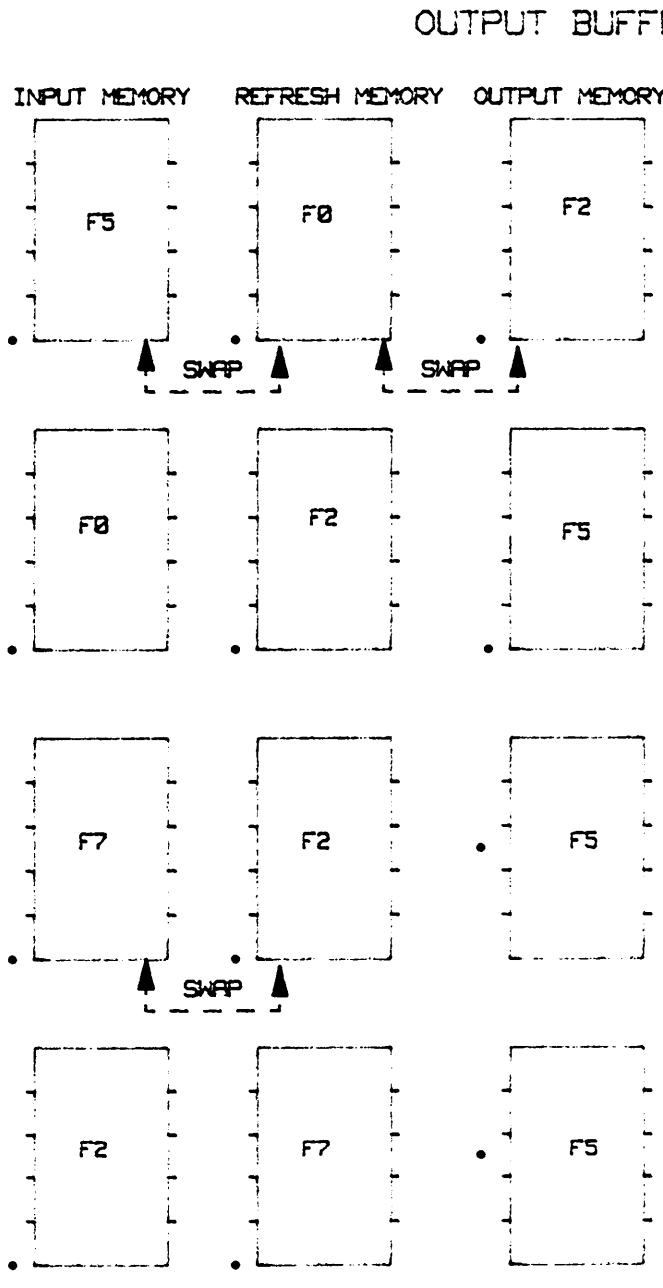


Figure 38. DATA FLOW MODIFIED I/O BUFFERS (CONTD)

earlier in this section. We can conclude from these figures that the monitor will alway be presented with a whole image. This should get rid of the tearing.

A one pixel jitter is visible on every line of the image because the sync pulses on the camera and the digitizing clock (INCLK) generated on clock board are not synchronized. This problem can be solved by generating INCLK on the camera A/D board. The crystal oscilations are inhibited on the falling edge of the sync pulses and start on the rising edge of the sync pulses, thus ensuring synchronization.

5.2 OTHER MINOR PROBLEMS

The following is the list of the problems that we faced and their solutions. In some cases the only solution was time!

This work began with a very vague problem definition, we had a a very general feeling that the IPB (which had been previously designed) needed some kind of I/O conditioning for it to be used with a standard camera and a T.V. monitor. There was also a need to run the IPB at variable frame rates without it (IPB) knowing it. Finally there was a need to experiment with the data rate at the output of the IPB so that an acceptable mimimum data rate could be arrived at as well an interlacing format most suitable

for the ultimate Telesign system. The solution to this of course is this work; however it took us about six months to define the problem.

- The clock board was the first design undertaken. Two initial designs had to be discarded because of changes in the Telesign system. The first design had to be discarded because the new Telesign requirement was that the IPB be able to operate in 128*128 mode; this mode change was to be implemented by changing the IPB clock. This design was later completely revised since it was extemely complex and involved the use of a large number of MSI control chips. The final design evolved from this and three PALs were used to replace almost all MSI logic chips.

- The designs of the input buffer and the output buffer were undertaken only after all the concepts involved were fully understood. Thus only changes had to be made after the design was complete. However the conceptual realisation of both the input and the output buffers took a considerable amount of time.

- Finding the correct ICs was a problem; also poor availability of the required chips and long delivery times

delayed the project considerably. There is of course no solution to this.

- The non-availibility of a PAL progammer was a major handicap. However this was resolved after the purchase of a PC-compatible PAL programmer with PALASM software. Due to lack of experience in PAL progamming, the PAL programs had to be written after the PALASM language was learned. This fact delayed the design to a large extent.

- The problems encountered in the actual implementation of the hardware were the normal problems any hardware designer would encounter. We attempt to list here a few of the problems encountered.

    1. The circuits had to be designed to operate at relatively high frequencies; this posed a major problem, especially since all the boards were to be wire-wrapped. This was solved by using high frequency wire-wrap boards with Vcc and GND planes. All the I.C.s used are of the Advanced Schottkey (AS) series with typical delay times of 15 to 25 ns. PALs also have typical delays of about 10 to 15 ns, hence were ideal.

2. Interboard communication posed a major problems introducing crosstalk between signals, giving rise to glitches. This problem was resolved by using current drivers and receivers and terminating the backplane to form low impedence transmission lines between boards.

3. Wiring mistakes took up a considerable amount of debugging time and at times were difficult to locate.

This describes the problems encountered in implementing this project.

5.3 CONCLUSIONS.

A variable frame rate I/O system with postprocessing options has been designed and fabricated. After the incorporation of the major modification to the I/O buffers discussed in section 5.1 it will be possible to run the IPB or any other real time image processor at a variable frame rate; the image processor data can be further processed in the output buffer using the various formats proposed. This will allow a dymanic variation of the effective frame and data rate of the image processor and at the same time adapt any real time processor to receive data from a wide range of input sources and display the processed data on a standard video monitor.

BIBLIOGRAPHY.

1. Nadler M., et al : "The Telesign Project." IEEE Special Issue on Visual Communication Systems, April 1985.

2. Pearson D. E. : "Visual Communication at Very Low Data Rates." IEEE Special Issue on Visual communication Systems, April 1985.

3. Nadler M. : "Telesign Project Report Submitted to NSF." Interim report of the Telesign project, 1984-1985.

4. MOS Memory Data Book. Texas Instruments, 1986.

5. TTL Data Book. Vol. 3. Texas Instruments,1986.

6. High Speed CMOS Logic Data Book. Texas Instruments, 1986.

7. LSI Logic Data Book. Texas Instruments, 1986.

8. Monolithie Memories. Pal Handbook, 1986.

9. Programmable Array Logic. Advanced Micro Devices, 1983.

10. TML 1840 Video Dac Data Sheet. Telemos Inc, 1985.

11. CD 22402D CMOS LSI Sync Generator. RCA Solidstate, 1985.

12. Bipolar Microprocessor Logic and Interface Data Book. Advanced Micro Devices, 1985.

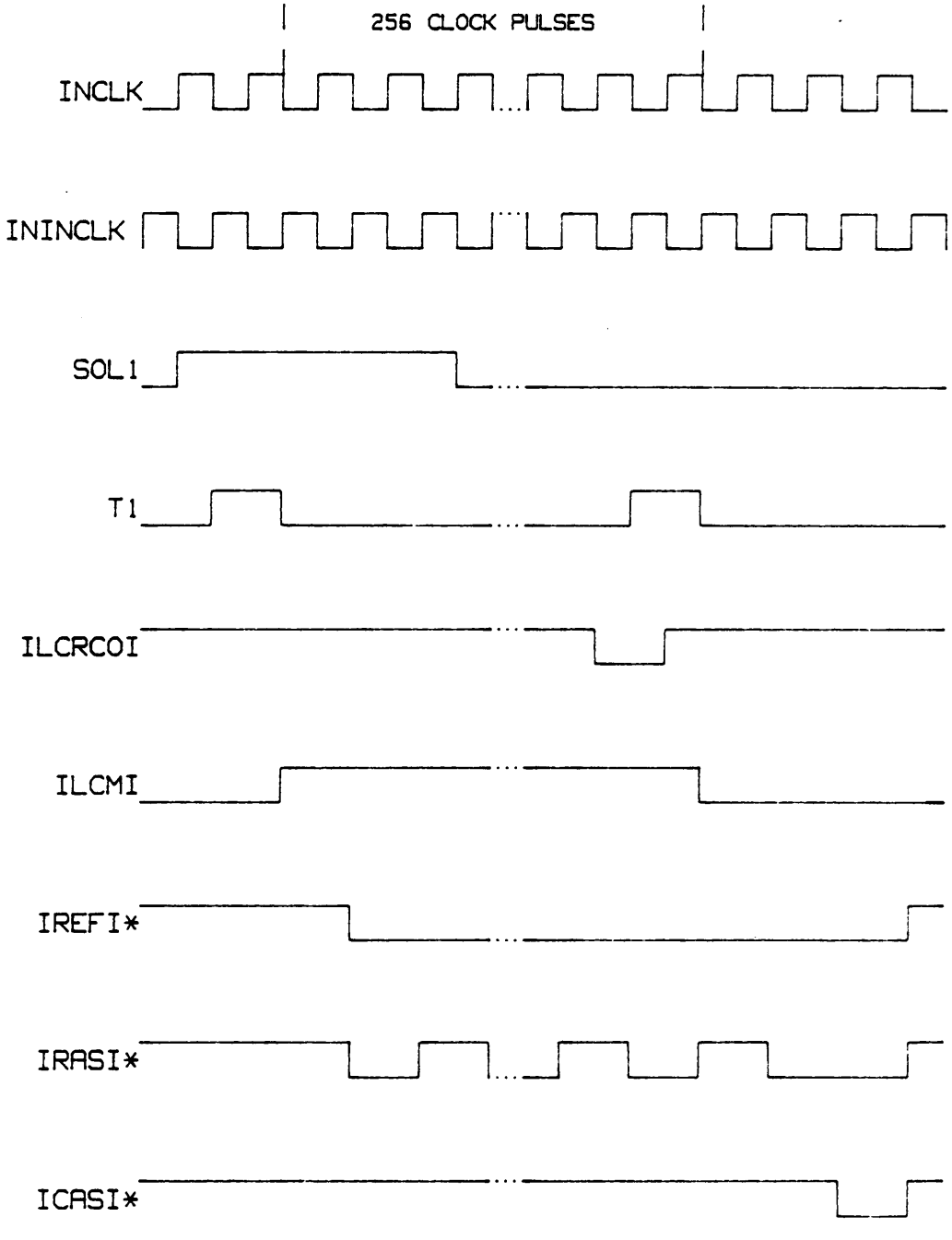
APPENDIX A. INPUT BUFFER DESIGN DOCUMENTATION.

The principles involved and the basic hardware design concepts have already been explained in Chapter 2 of this thesis. The following sections contain all the design documentation for the input buffer.

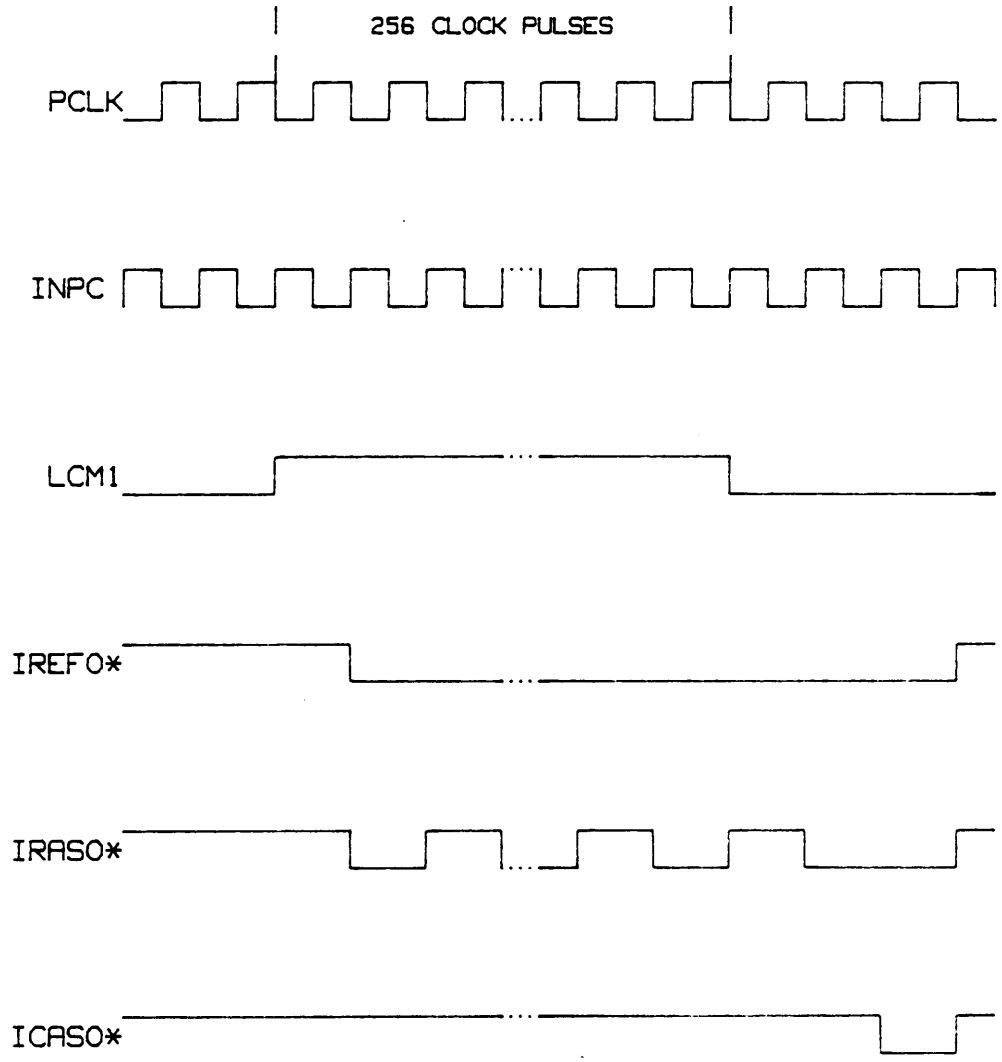
A.1 INPUT BUFFER TIMING DIAGRAMS.

The following set of drawings are the timing diagrams of the input buffer. They should be read in reference to the circuit diagrams and the PAL programs in the next two sections. Chapter 2 contains all the necessary explanation of the various signals. The signal generation is actually performed in the PALs from the available incoming signals.

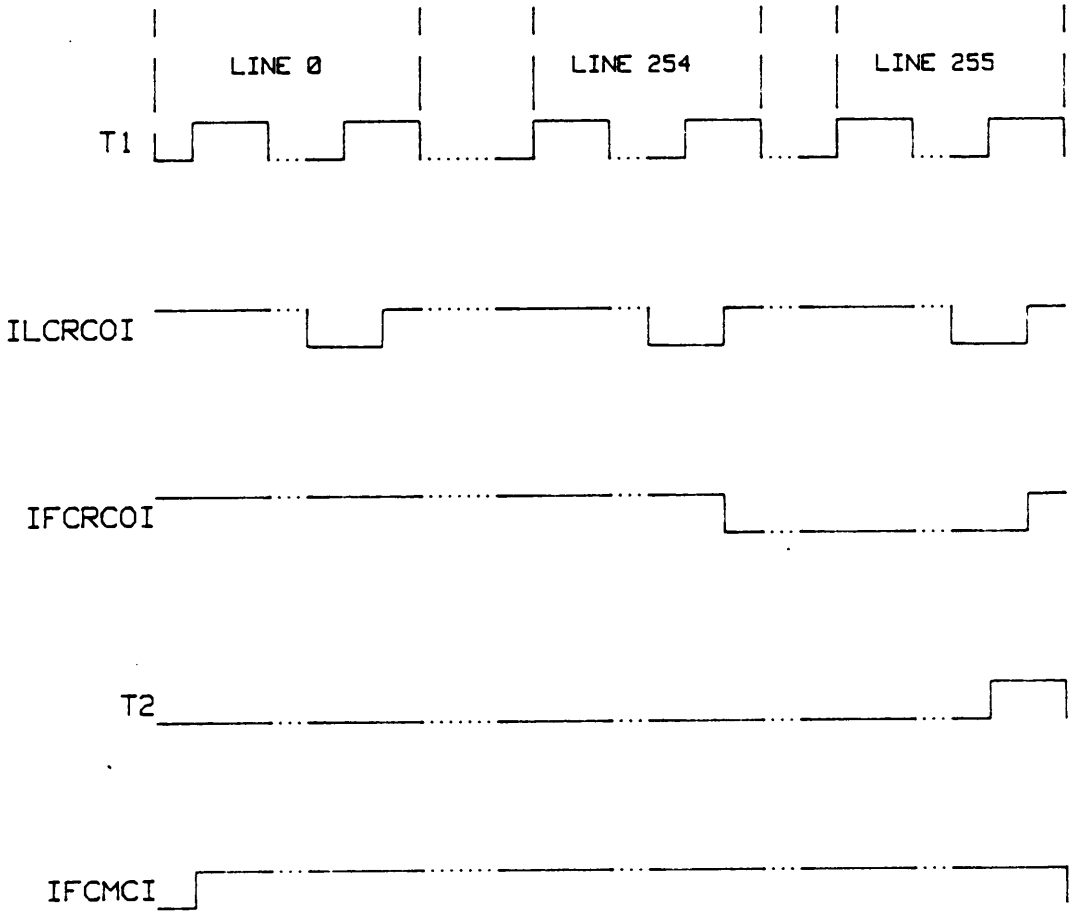
INPUT BUFFER: MEMORY CONTROL SIGNALS.



INPUT BUFFER: MEMORY CONTROL SIGNALS.
OUTPUT MEMORY

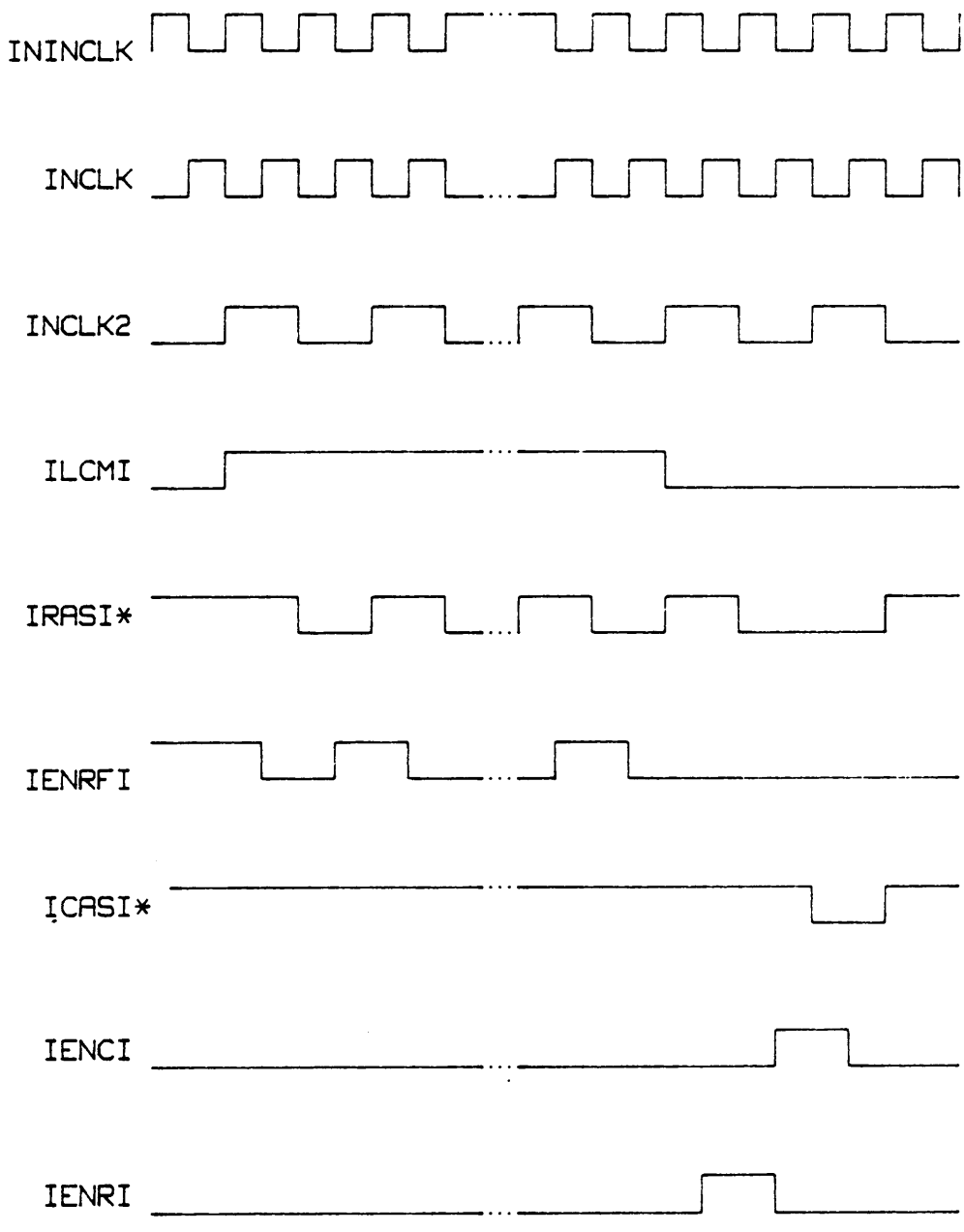


INPUT BUFFER: INPUT MEMORY FRAME MARKER.



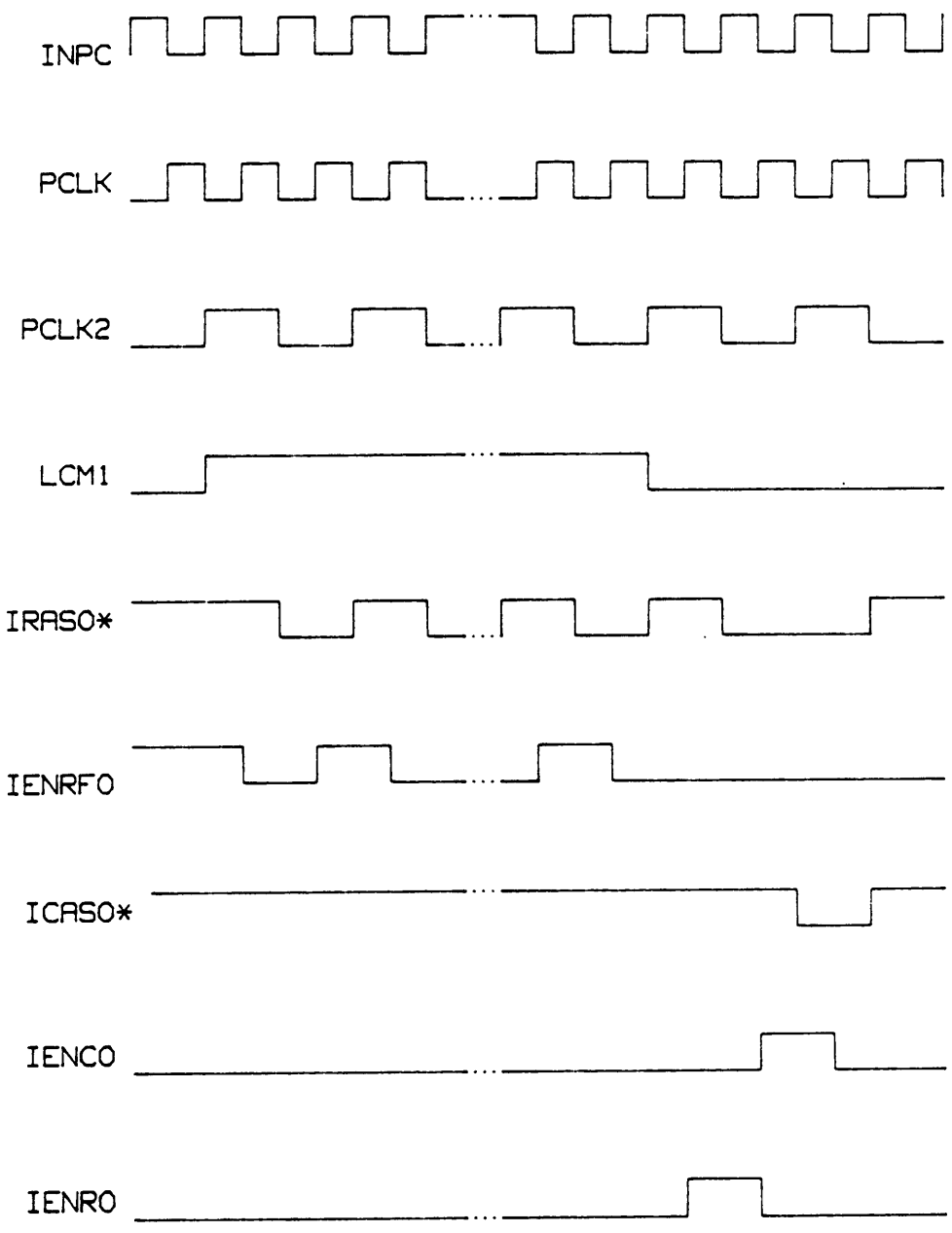
INPUT BUFFER

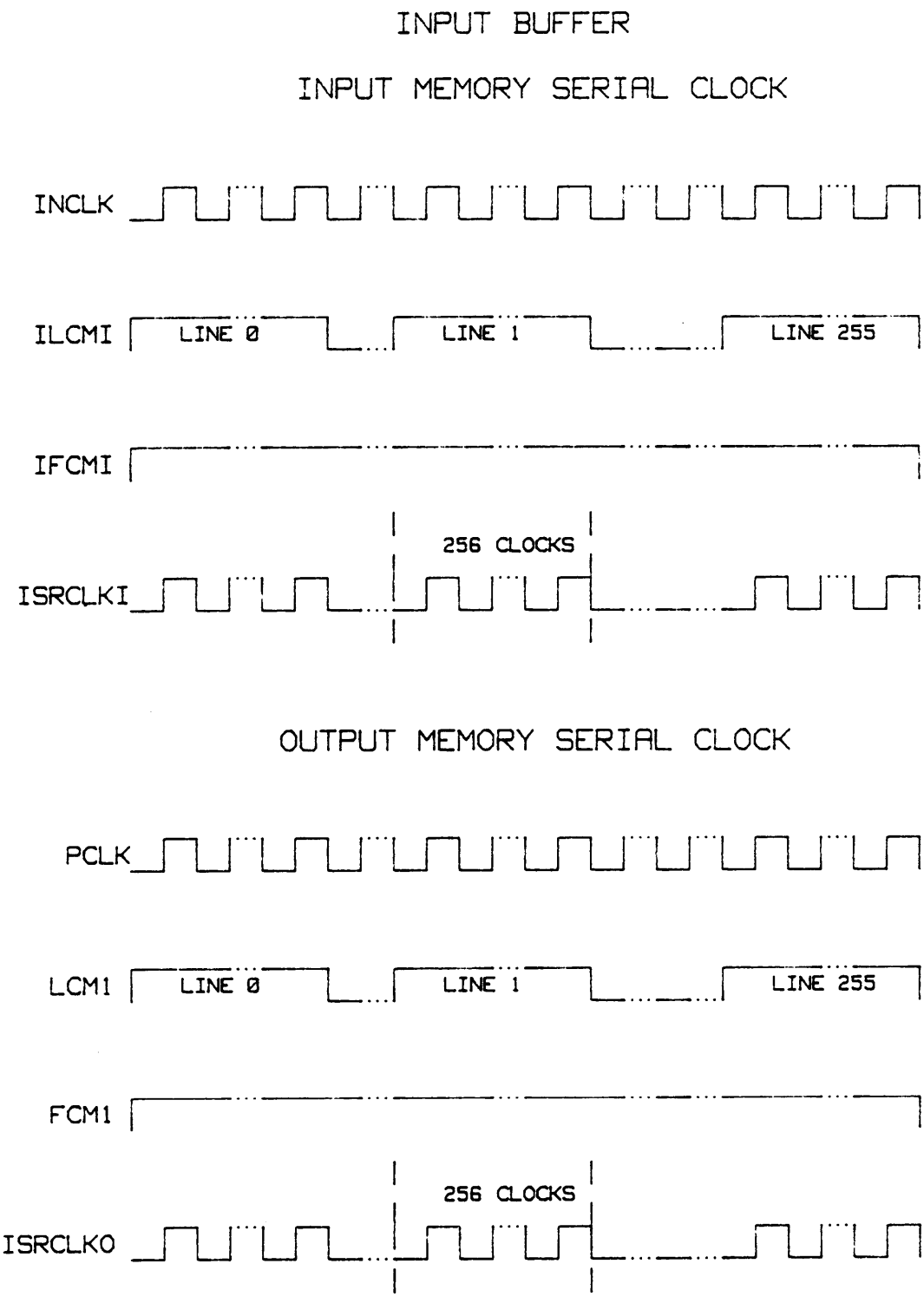
INPUT MEMORY ADDRESS CONTROL.



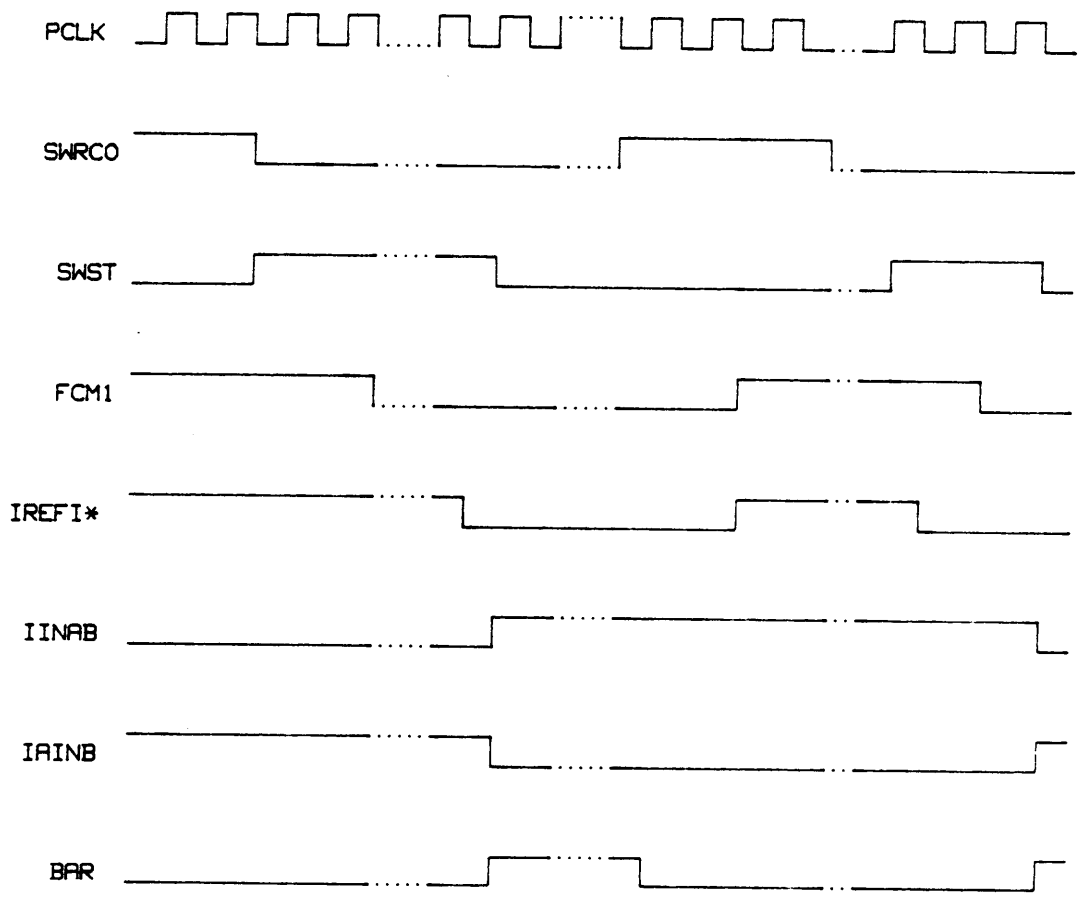
INPUT BUFFER

OUTPUT MEMORY ADDRESS CONTROL.



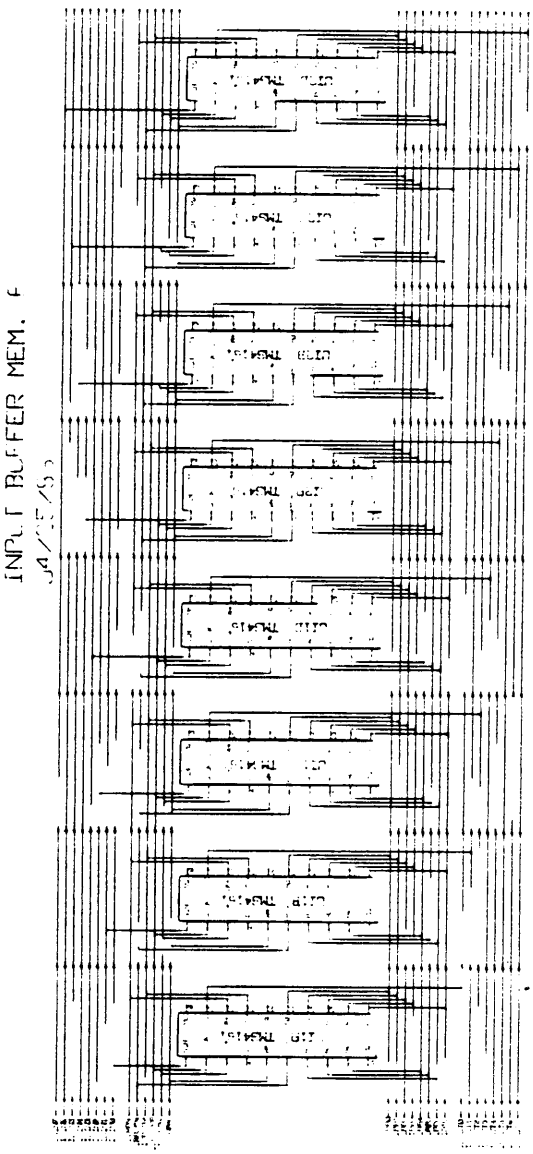

INPUT BUFFER
INPUT MEMORY SERIAL CLOCK
INCLK
ILCMI
LINE 0
LINE 1
LINE 255
IFCMI
256 CLOCKS
ISRCLKI
OUTPUT MEMORY SERIAL CLOCK
PCLK
LCM1
LINE 0
LINE 1
LINE 255
FCM1
256 CLOCKS
ISRCLKO

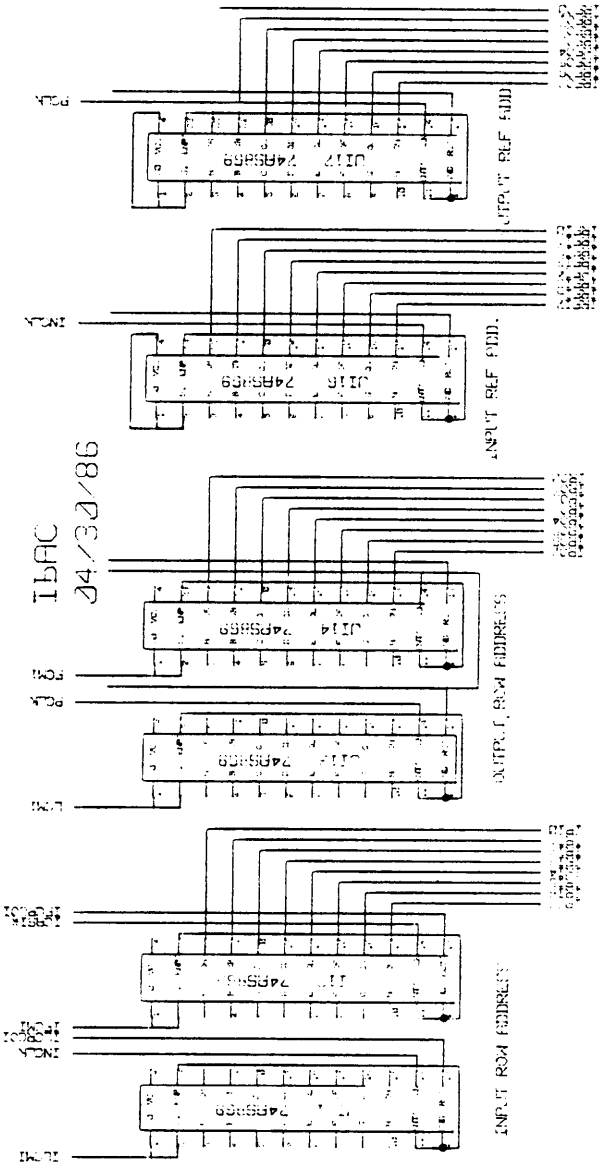
INPUT BUFFER: FRAME REFRESH SIGNAL.

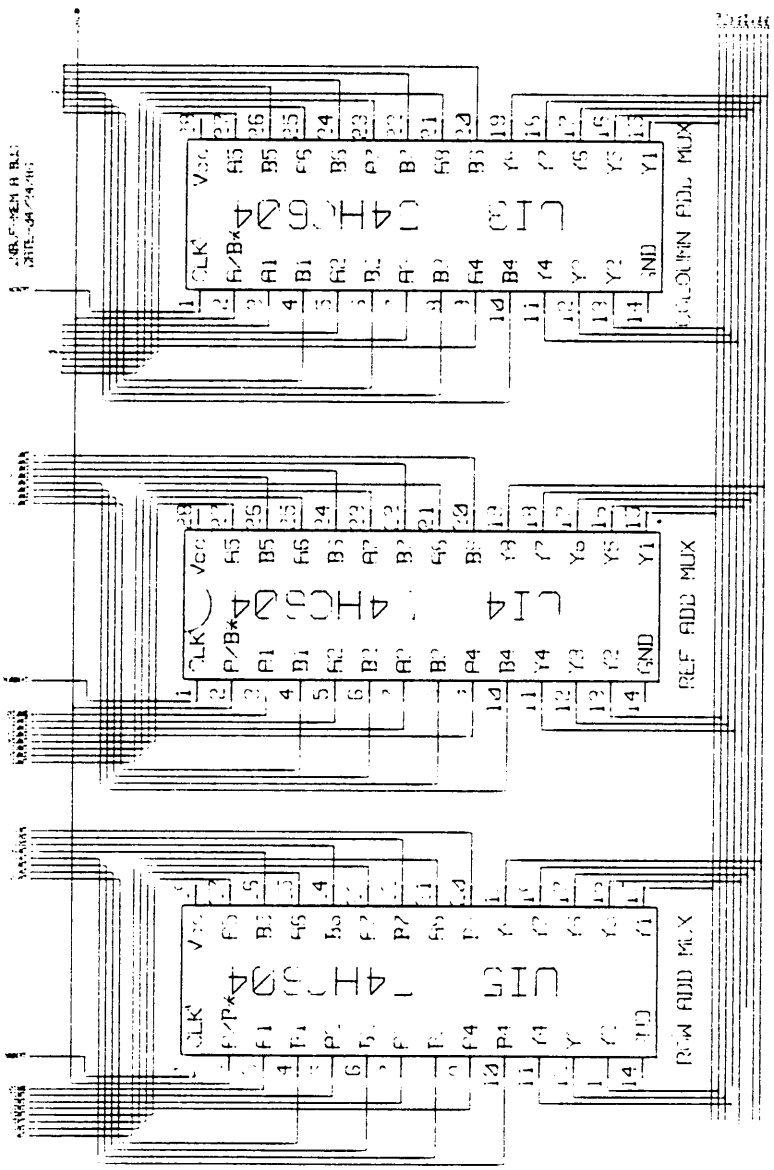


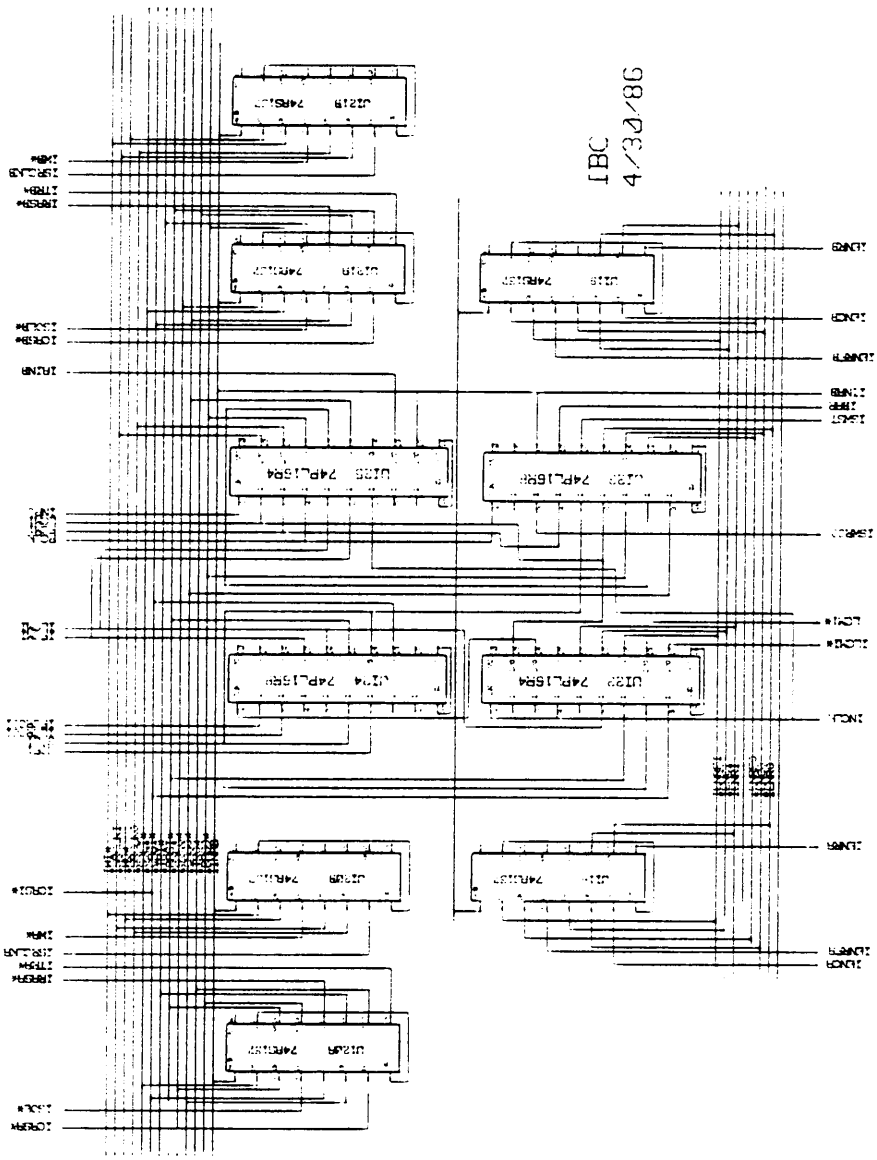
A.2 INPUT BUFFER CIRCUIT DIAGRAMS.

The following are the circuit diagrams for the input buffer. The details of all the input buffer blocks described in Chapter 2 are shown here.




TBAC
04/30/86




IBC
4/30/86

A.3 INPUT BUFFER PAL PROGRAMS AND FUSE PLOTS.

This section contains the PAL programs in the PALASM language for all the PALS used in the input buffer. Also the program generated simulation results, the fuse plots and the listing of the JEDEC files used by the PAL programmer to program the PAL are included in this section.

```
PAL16R4                    PAL DESIGN SPECIFICATIONS
PAT1001                    SANJAY ADKAR 04-13-1986
INPUT BUFFER OUTPUT MEMORY CONTROL PAL  UI 25
VIRGINIA TECH, BLACKSBURG, VA 24061

INPC LCM1 FCM1 PCLK ILCMI IFCMI INCLK NC NC GND
OE IINAB IAINB NC /ICASO IREFO /IRASO  ISRCLKO ISRCLKI VCC
;THE ABOVE DEFINES THE PIN LIST.

;EQUATIONS

/IREFO := /ICASO * LCM1                +
          /IREFO * /ICASO

IRASO  := /LCM1 * /IREFO * /ICASO      +
          /IREFO * /IRASO  * /ICASO    +
          /IRASO * /ICASO * LCM1       +
          /LCM1 * /ICASO * IRASO

ICASO  := IRASO * /IREFO * /ICASO * /LCM1

IF (VCC) /ISRCLKO = /PCLK              +
                    /LCM1              +
                    /FCM1

IF (VCC) /ISRCLKI = /INCLK             +
                    /ILCMI             +
                    /IFCMI


IF (VCC) /IAINB  = IINAB
```

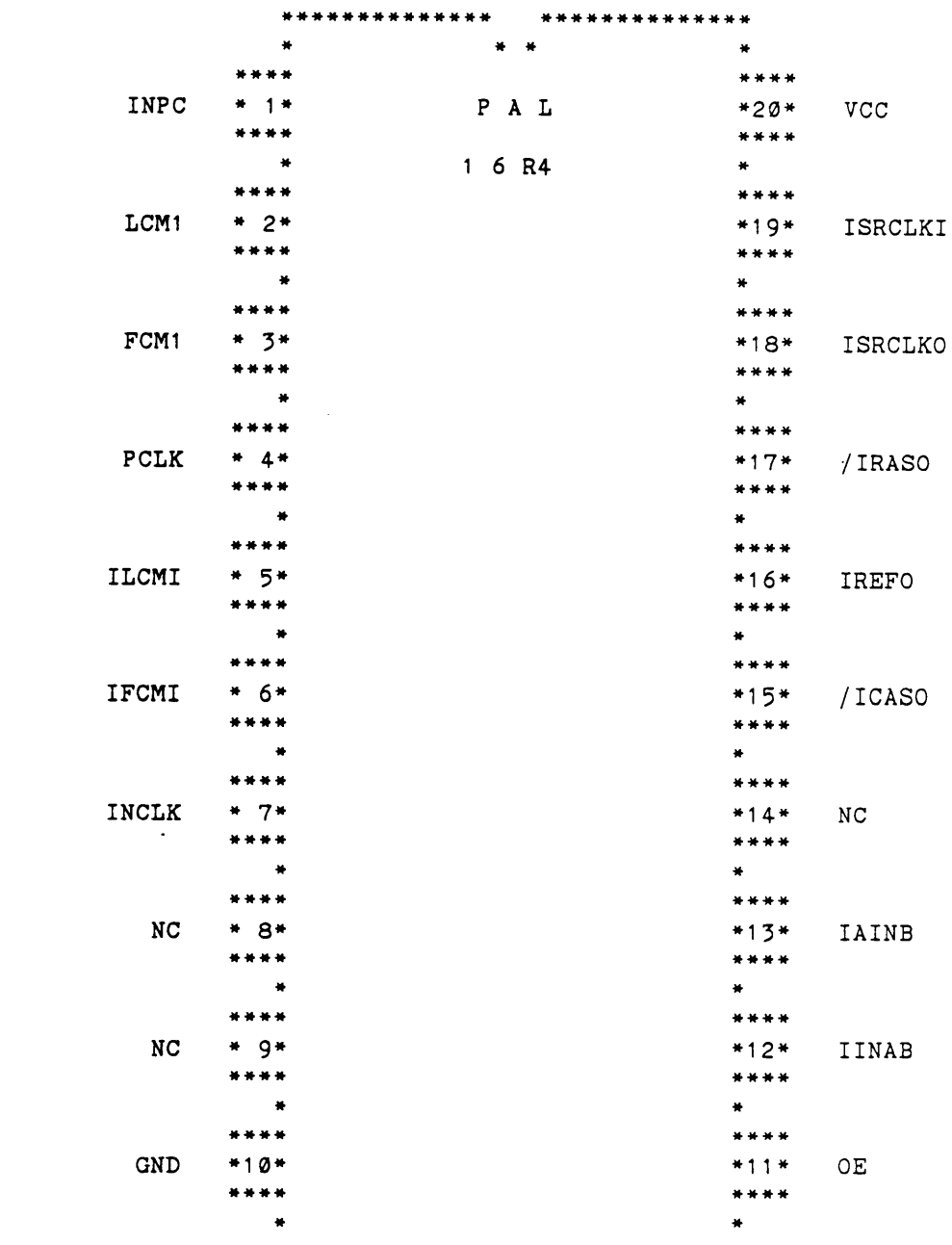
FUNCTION TABLE

INPC LCM1 IREFO /IRASO /ICASO
PCLK FCM1 ISRCLKO INCLK ILCMI IFCMI ISRCLKI IINAB IAINB

```
;                                I           I
;              /   /            S           S
;         I    I   I            R   I   I   I   R   I   I
; I   L   R    R   C   P   F    C   N   L   F   C   I   A
; N   C   E    A   A   C   C    L   C   C   C   L   N   I
; P   M   F    S   S   L   M    K   L   M   M   K   A   N
; C   1   O    O   O   K   1    O   K   I   I   I   B   B
-------------------------------------------------------
  C   L   H    H   H   H   L    L   H   H   H   H   L   H
  C   H   L    L   H   L   H    L   L   H   H   L   H   L
  C   H   L    H   H   H   H    H   H   H   H   H   L   H
  C   H   L    L   H   L   H    L   L   H   H   L   H   L
  C   H   L    H   H   H   H    H   H   H   H   H   L   H
  C   L   L    L   H   H   H    L   H   L   H   L   L   H
  C   L   L    L   L   L   H    L   L   L   H   L   H   L
  C   L   H    H   H   H   H    L   H   H   H   H   L   H
-------------------------------------------------------
```

DESCRIPTION: THIS PAL PERFORMS THE FOLLOWING FUNCTIONS:
A) GENERATES THE MEMORY CONTROL SIGNALS FOR MEMORY TO SHIFT REGISTER TRANSFER.
B) GENERATES THE CONTROL SIGNALS TO REFRESH THE OUTPUT MEMORY.
C) GENERATES THE SERIAL SHIFT REGISTER CLOCKS FOR BOTH THE INPUT AND THE OUTPUT MEMORY.
D) INVERTS THE REFRESH SIGNAL.

INPUT BUFFER OUTPUT MEMORY CONTROL PAL UI 25



```
PAL20 V1.7K - PAL16R4  - INPUT BUFFER OUTPUT MEMORY CONTROL PAL  UI 25

                 11 1111 1111 2222 2222 2233
   0123 4567 8901 2345 6789 0123 4567 8901

 0 ---- ---- ---- ---- ---- ---- ---- ----
 1 ---- ---- ---- ---- ---- -X-- ---- ---- /INCLK
 2 ---- ---- ---- -X-- ---- ---- ---- ---- /ILCMI
 3 ---- ---- ---- ---- -X-- ---- ---- ---- /IFCMI
 4 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 5 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 6 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
 7 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

 8 ---- ---- ---- ---- ---- ---- ---- ----
 9 ---- ---- -X-- ---- ---- ---- ---- ---- /PCLK
10 -X-- ---- ---- ---- ---- ---- ---- ---- /LCM1
11 ---- -X-- ---- ---- ---- ---- ---- ---- /FCM1
12 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
13 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
14 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
15 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

16 -X-- ---- ---- ---X --X- ---- ---- ---- /LCM1*/IREFO*/ICASO
17 ---- ---- --X- ---X --X- ---- ---- ---- /IREFO*/IRASO*/ICASO
18 X--- ---- --X- ---- --X- ---- ---- ---- /IRASO*/ICASO*LCM1
19 -X-- ---- ---X ---- --X- ---- ---- ---- /LCM1*/ICASO*IRASO
20 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
21 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
22 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
23 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

24 X--- ---- ---- ---- --X- ---- ---- ---- /ICASO*LCM1
25 ---- ---- ---- ---X --X- ---- ---- ---- /IREFO*/ICASO
26 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
27 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
28 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
29 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
30 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
31 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

32 -X-- ---- ---X ---X --X- ---- ---- ---- IRASO*/IREFO*/ICASO*/LCM1
33 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
34 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
35 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
36 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
37 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
38 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
```

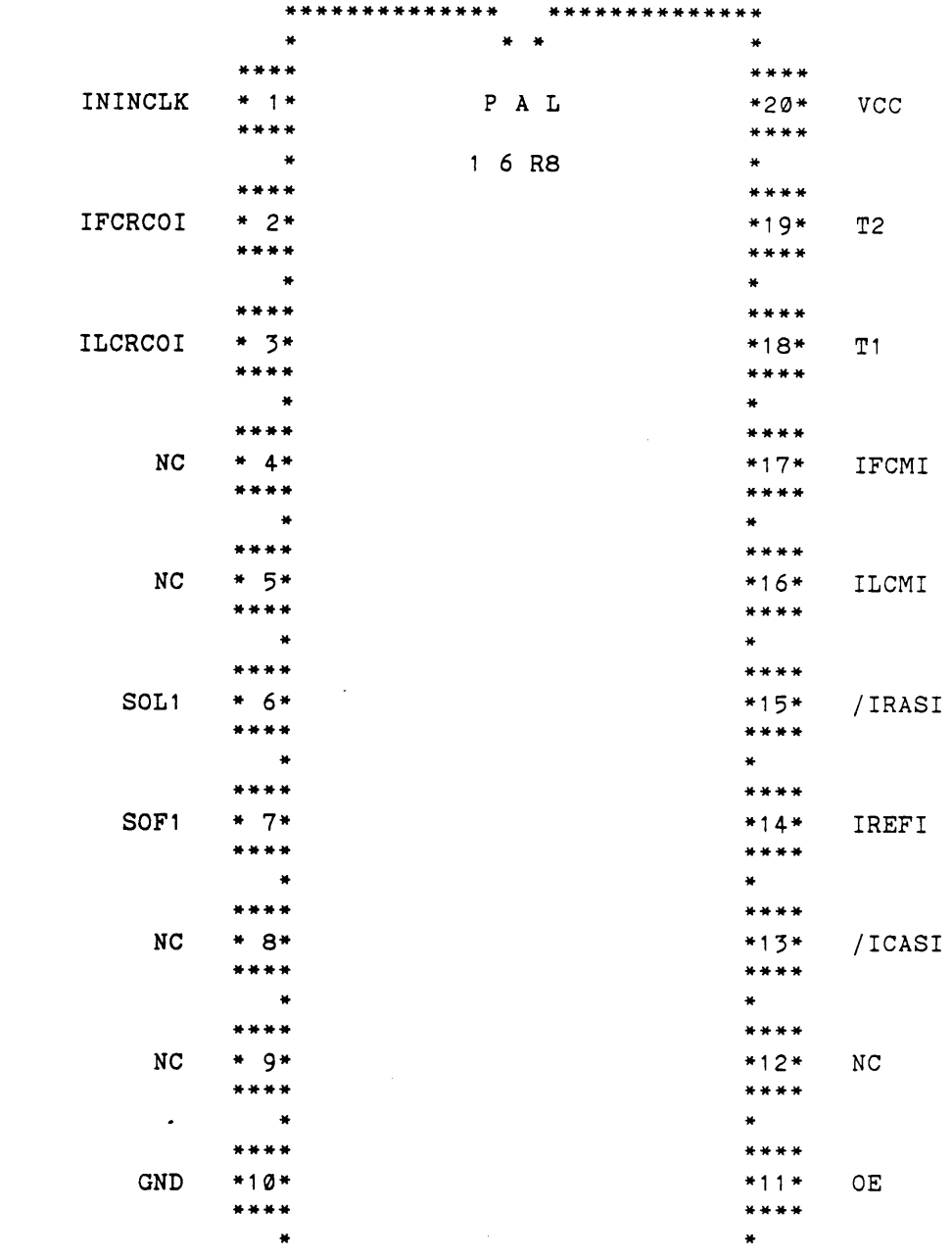
```
39 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

40 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
41 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
42 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
43 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
44 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
45 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
46 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
47 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

48 ---- ---- ---- ---- ---- ---- ---- ----
49 ---- ---- ---- ---- ---- ---- ---- --X- IINAB
50 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
51 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
52 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
53 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
54 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
55 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

56 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
57 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
58 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
59 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
60 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
61 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
62 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
63 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX


LEGEND:  X : FUSE NOT BLOWN (L,N,0)   - : FUSE BLOWN   (H,P,1)

NUMBER OF FUSES BLOWN =  517

SECURITY FUSE XX
```

```
INPUT BUFFER OUTPUT MEMORY CONTROL PAL  UI 25

   1 C001111XXXX0HXHHHLH1
   2 C110110XXXX1LXHLLLL1
   3 C111111XXXX0HXHLHHH1
   4 C110110XXXX1LXHLLLL1
   5 C111111XXXX0HXHLHHH1
   6 C011011XXXX0HXHLLLL1
   7 C010010XXXX1LXLLLLL1
   8 C011111XXXX0HXHHHLH1

PASS SIMULATION


PAL16R4                    PAL DESIGN SPECIFICATIONS
PAT1001                    SANJAY ADKAR 04-13-1986
INPUT BUFFER OUTPUT MEMORY CONTROL PAL  UI 25
VIRGINIA TECH, BLACKSBURG, VA 24061
*D2224*
G0*F0*
L0000 11111111111111111111111111111111*
L0032 11111111111111111111101111111111*
L0064 11111111111110111111111111111111*
L0096 11111111111111111011111111111111*
L0256 11111111111111111111111111111111*
L0288 11111111101111111111111111111111*
L0320 10111111111111111111111111111111*
L0352 11111011111111111111111111111111*
L0512 10111111111111101101111111111111*
L0544 11111111110111101101111111111111*
L0576 01111111110111111101111111111111*
L0608 10111111111011111101111111111111*
L0768 01111111111111111101111111111111*
L0800 11111111111111101101111111111111*
L1024 10111111111011101101111111111111*
L1536 11111111111111111111111111111111*
L1568 11111111111111111111111111111101*
V0001 C001111XXNX0HXHHHLHN*
V0002 C110110XXNX1LXHLLLLN*
V0003 C111111XXNX0HXHLHHHN*
V0004 C110110XXNX1LXHLLLLN*
V0005 C111111XXNX0HXHLHHHN*
V0006 C011011XXNX0HXHLLLLN*
V0007 C010010XXNX1LXLLLLLN*
V0008 C011111XXNX0HXHHHLHN*
C40DA*
F649
```

```
PAL16R8   PAL DESIGN SPECIFICATIONS
PAT1007                  SANJAY ADKAR 04-14-86
INPUT MEMORY CONTROL PAL - INPUT BUFFER UI 24
VIRGINIA TECH BLACKSBURG VA 24061

ININCLK IFCRCOI ILCRCOI NC NC SOL1 SOF1 NC NC  GND
OE NC /ICASI IREFI /IRASI ILCMI IFCMI T1 T2 VCC

;THE ABOVE DESCRIBES THE PIN LIST OF THE PAL.

;EQUATIONS.
/T1   :=      T1                    +
              /SOL1 *ILCRCOI         +
              /SOL1 * /ILCMI         +
              ILCMI *ILCRCOI

/T2   :=      ILCRCOI               +
              IFCRCOI               +
              T2


/IFCMI  :=     /SOF1 * /IFCMI        +
              /SOF1 * T1 * T2      +
              /T1 * /IFCMI          +
              IFCMI * T1 * T2


/ILCMI  :=     ILCMI *T1             +
              /ILCMI * /T1


ICASI   :=     IRASI * /IREFI * /ICASI */ILCMI


IRASI   :=    /ILCMI * /IREFI */ICASI +
             /IREFI * /IRASI * /ICASI +
             /IRASI * /ICASI * ILCMI  +
             /ILCMI * /ICASI * IRASI


/IREFI  :=     /ICASI * ILCMI        +
              /IREFI * /ICASI
```

```
FUNCTION TABLE

ININCLK SOF1 SOL1 ILCRCOI IFCRCOI
IFCMI ILCMI T1 T2 /IRASI /ICASI IREFI

; I        I  I                    /  /
; N        L  F                    I  I  I
; I        C  C  I  I              I  I  I
; N  S  S  R  R  F  L              R  C  R
; C  O  O  C  C  C  C              A  A  E
; L  F  L  O  O  M  M  T  T        S  S  F
; K  1  1  I  I  I  I  1  2        I  I  I

-----------------------------------------------
  C  L  L  H  H  L  L  L  L  H  H  H
  C  H  L  H  H  L  L  L  L  H  H  H
  C  H  H  H  H  L  L  H  L  H  H  H
  C  H  X  H  H  H  H  L  L  H  H  H
  C  X  X  H  H  H  H  L  L  L  H  L
  C  X  X  H  H  H  H  L  L  H  H  L
  C  X  X  L  H  H  H  H  L  L  H  L
  C  X  X  H  H  H  L  L  L  H  H  L
  C  X  L  H  H  H  L  L  L  L  H  L
  C  X  L  H  H  H  L  L  L  L  L  L
  C  X  L  H  H  H  L  L  L  H  H  H
  C  X  H  H  L  H  L  H  L  H  H  H
  C  L  X  H  L  H  H  L  L  H  H  H
  C  L  X  H  L  H  H  L  L  L  H  L
  C  L  X  H  L  H  H  L  L  H  H  L
  C  L  X  L  L  H  H  H  H  L  H  L
  C  L  X  H  H  L  L  L  L  H  H  L
-----------------------------------------------
;

DESCRIPTION: THIS PAL GENERATES THE INPUT MEMORY LINE AND FRAME
MARKERS FROM THE SOL AND SOF SIGNALS.
IT ALSO GENERATES ALL THE INPUT MEMORY CONTROL SIGNALS  REQUIRED
TO CONTROL THE MEMORY.
```

INPUT MEMORY CONTROL PAL - INPUT BUFFER UI 24



```
PAL16R4                 PAL DESIGN SPECIFICATIONS
PAT1005                 SANJAY ADKAR  04-13-1986
OUTPUT ADDRESS CONTROL PAL - OUTPUT BUFFER UI22
VIRGINIA TECH , BLACKSBURG, VA.


INCLK NC NC ICLK NC ILCMI /IRASI IREFI /ICASI GND
OE ILCMI' LCM1' IENRFI IENCI IENRI NC ININCLK LCM1 VCC

;THE ABOVE DESCRIBES THE PIN LIST OF THE PAL.

;EQUATIONS

/IENRFI :=      IENRFI
          +     /ILCMI

/IENCI  :=      /IENRI
          +     IENCI

/IENRI  :=    IENRI
          +   IREFI
          +   IENCI
          +   ILCMI

IF (VCC) /ININCLK =      ICLK

IF (VCC) /LCM1'   =      LCM1

IF (VCC) /ILCMI'  =      ILCMI
```

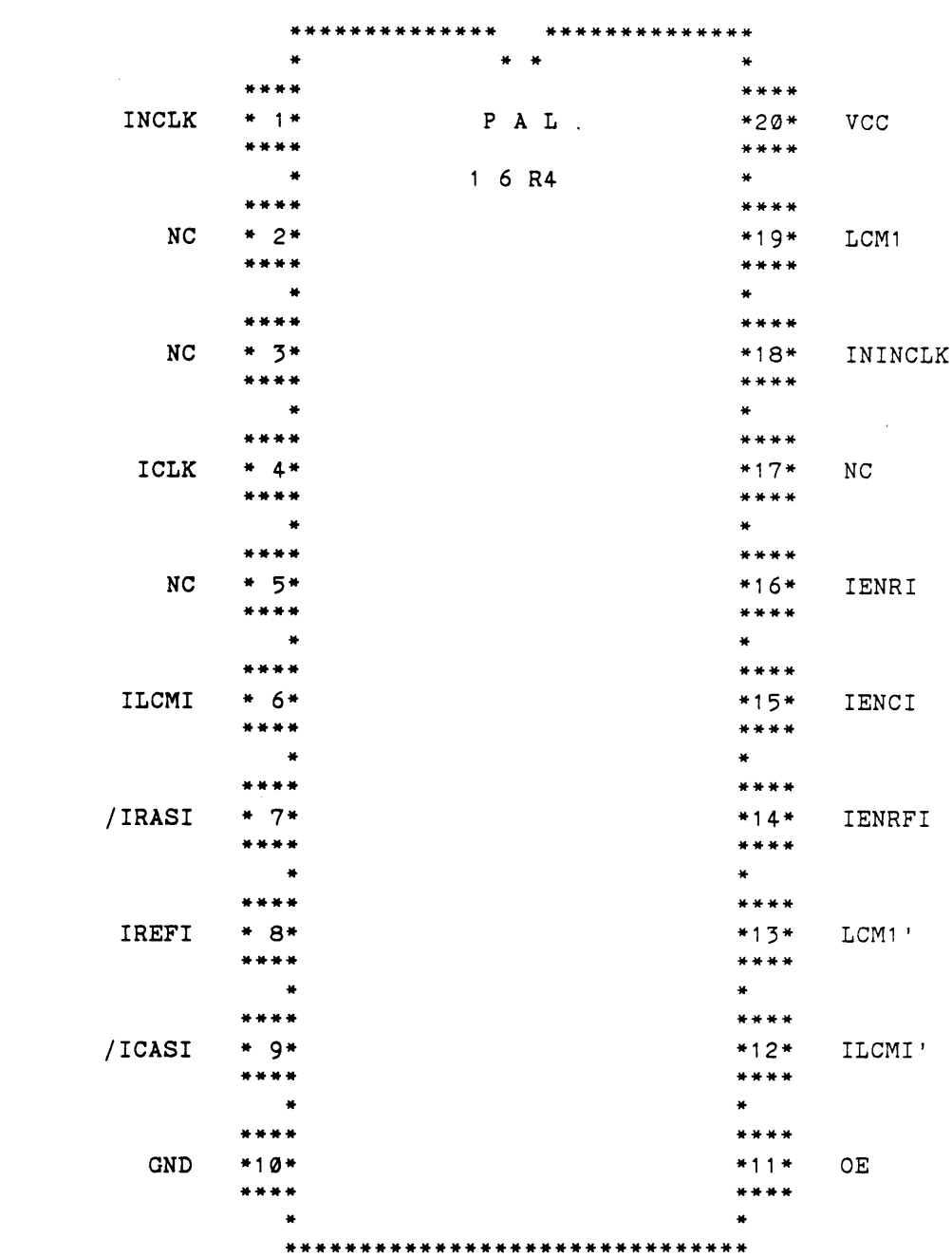
FUNCTION TABLE

INCLK /IRASI /ICASI IREFI ILCMI ICLK
IENRFI IENCI IENRI ININCLK LCM1 LCM1' ILCMI'

```
;                       I         I
;     /   /             E         N           I
; I   I   I   I   I     N   I   I   I     L   L
; N   R   C   R   L   I R   E   E   N   L C   C
; C   A   A   E   C   C E   N   N   C   C M   M
; L   S   S   F   M   L F   C   R   L   M 1   I
; K   I   I   I   I   K I   I   I   K   1 '   '
-----------------------------------------------
  C   H   H   H   H   H   H   L   L   L   H   L   L
  C   L   H   L   H   L   L   L   L   H   L   H   L
  C   H   H   L   H   H   H   L   L   L   H   L   L
  C   L   H   L   H   L   L   L   L   H   L   H   L
  C   H   H   L   L   H   L   L   H   L   H   L   H
  C   L   H   L   L   L   L   H   L   H   L   H   H
  C   L   L   L   L   H   L   L   L   L   H   L   H
  C   H   H   H   L   L   L   L   L   H   L   H   H
-----------------------------------------------
```

DESCRIPTION: THIS PAL GENERATES ALL THE CONTROLS REQUIRED TO MULTIPLEX THE INPUT ROW, COLOUMN AND THE REFRESH ADDRESS ON THE ADDRESS BUS. IT ALSO GENERATES THE COMPLEMENTS OF THE INPUT AND OUTPUT LINE MARKERS, THESE SIGNALS ARE USED TO CONTROL THE REFRESH AND THE MEMORY TRANSFER CYCLES.

OUTPUT ADDRESS CONTROL PAL - OUTPUT BUFFER UI22



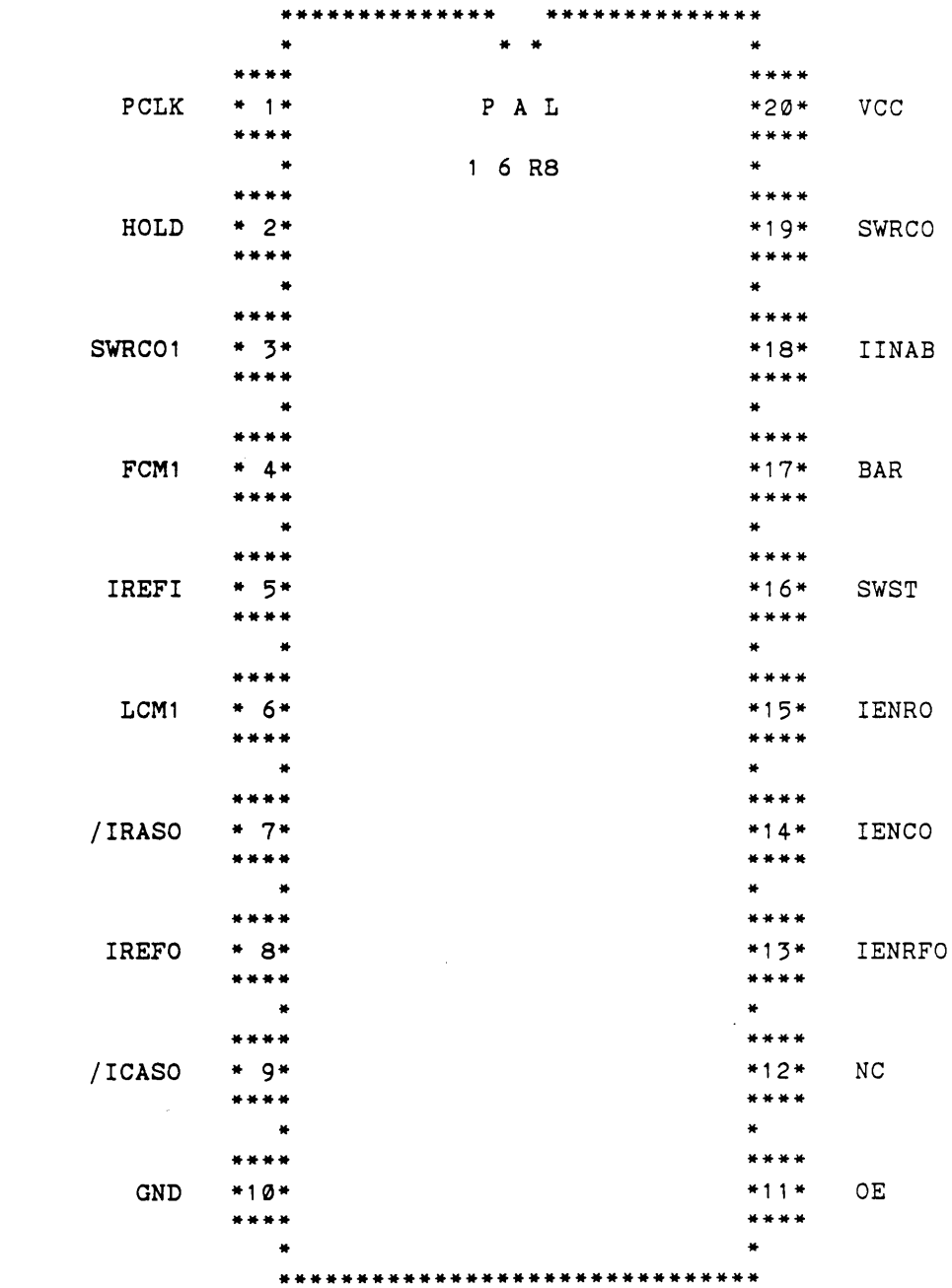
```
PAL16R8                    PAL DESIGN SPECIFICATION
PAT1003                    SANJAY ADKAR 04-14-1986
OUTPUT ADDRESS CONTROL PAL.-INPUT BUFFER UI23
VIRGINIA TECH BLACKSBURG VA 24060


PCLK HOLD SWRCO1 FCM1 IREFI LCM1 /IRASO IREFO /ICASO GND
OE NC IENRFO IENCO IENRO SWST BAR IINAB SWRCO VCC

;THE ABOVE DESCRIBES THE PIN LIST OF THE PAL.

;EQUATIONS

/IENRFO :=     IENRFO              +
               /LCM1


/IENCO  :=      IENCO              +
                /IENRO


/IENRO :=      LCM1                +
               IREFO               +
               IENCO               +
               IENRO

/SWRCO :=      /SWRCO1            +
               /HOLD


/SWST :=       SWRCO * /SWST       +
               SWRCO */FCM1 *IREFI * IREFO +
               BAR * /SWST         +
               BAR */FCM1 * IREFI * IREFO +
               SWST * /FCM1 * IREFO *IREFI

/BAR :=       /SWST * /BAR         +
               /IREFI * /BAR        +
               FCM1 * /BAR         +
               /SWST * SWRCO       +
               /IREFI * SWRCO       +
               FCM1 * SWRCO        +
               /IREFO * /BAR        +
               /IREFO * SWRCO
```

```
/IINAB :=      IINAB * SWST * /FCM1 * IREFI * IREFO +
              /IINAB * /SWST        +
              /IINAB *FCM1          +
              /IINAB * /IREFI        +
              /IINAB * /IREFO
```

FUNCTION TABLE

```
PCLK /IRASO /ICASO IREFO LCM1 IENRFO IENRO IENCO
SWRCO FCM1 IREFI SWST BAR IINAB HOLD SWRCO1

;    /  /       I                              S
;    I  I  I    E  I  I  S    I        I      W
; P  R  C  R  L  N  E  E  W  F  R  S     I  H  R
; C  A  A  E  C  R  N  N  R  C  E  W  B  N  O  C
; L  S  S  F  M  F  R  C  C  M  F  S  A  A  L  O
; K  O  O  O  1  O  O  O  O  1  I  T  R  B  D  1

-------------------------------------------------------
  C  H  H  H  H  H  L  L  L  H  L  H  L  L  H  L
  C  L  H  L  H  L  L  L  L  H  L  H  L  L  H  L
  C  H  H  L  H  H  L  L  L  L  L  H  L  L  H  L
  C  L  H  L  H  L  L  L  L  L  H  H  L  L  H  L
  C  H  H  L  L  L  H  L  L  L  H  H  L  L  H  L
  C  L  H  L  L  L  L  H  L  L  H  H  L  L  H  L
  C  L  L  L  L  L  L  L  L  L  H  H  L  L  H  L
  C  H  H  H  L  L  L  L  L  L  H  L  H  H  H  L
  C  H  H  H  L  L  L  L  H  L  H  L  H  H  H  H
  C  H  H  H  L  L  L  L  H  L  H  L  L  H  H  H
-------------------------------------------------------
;
DESCRIPTION:THIS PAL DOES THE FOLLOWING FUNCTIONS:
A) IT GENERATES ALL THE ADDRESS CONTROL SIGNALS REQUIRED FOR THE
OUTPUT MEMORY ADDRESS BUS.  THE TRI-STATE MULTIPLEXER BUFFERS
ARE CLOCKED USING THESE SIGNALS.
B) IT GENERATES THE REFRESH FRAME SIGNAL (IINAB).
```

OUTPUT ADDRESS CONTROL PAL.-INPUT BUFFER UI23



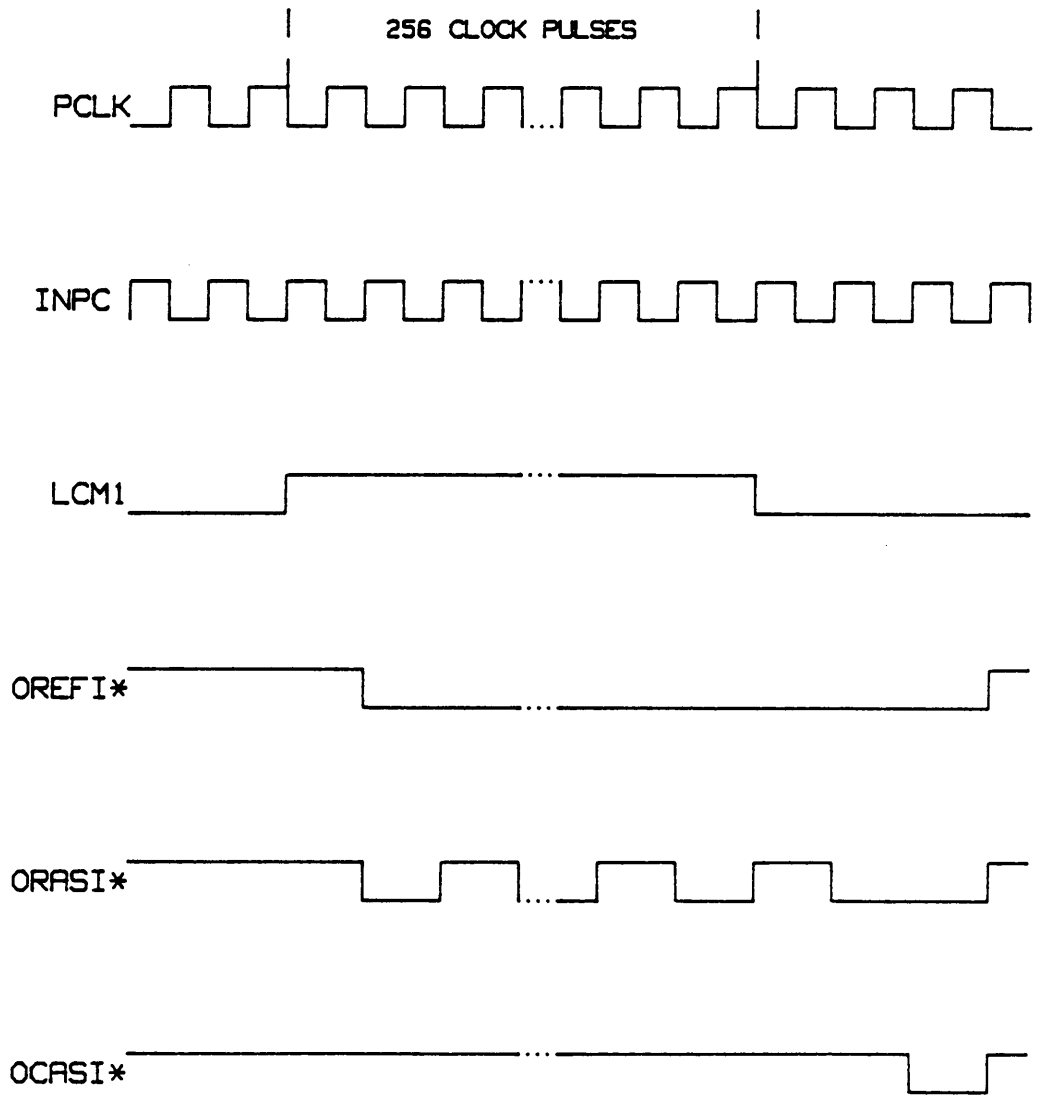
APPENDIX B. OUTPUT BUFFER DESIGN DOCUMENTATION.

The principles involved and the basic hardware design concepts have already been explained in Chapter 3 of this thesis. The following sections contain all the design documentation for the output buffer.

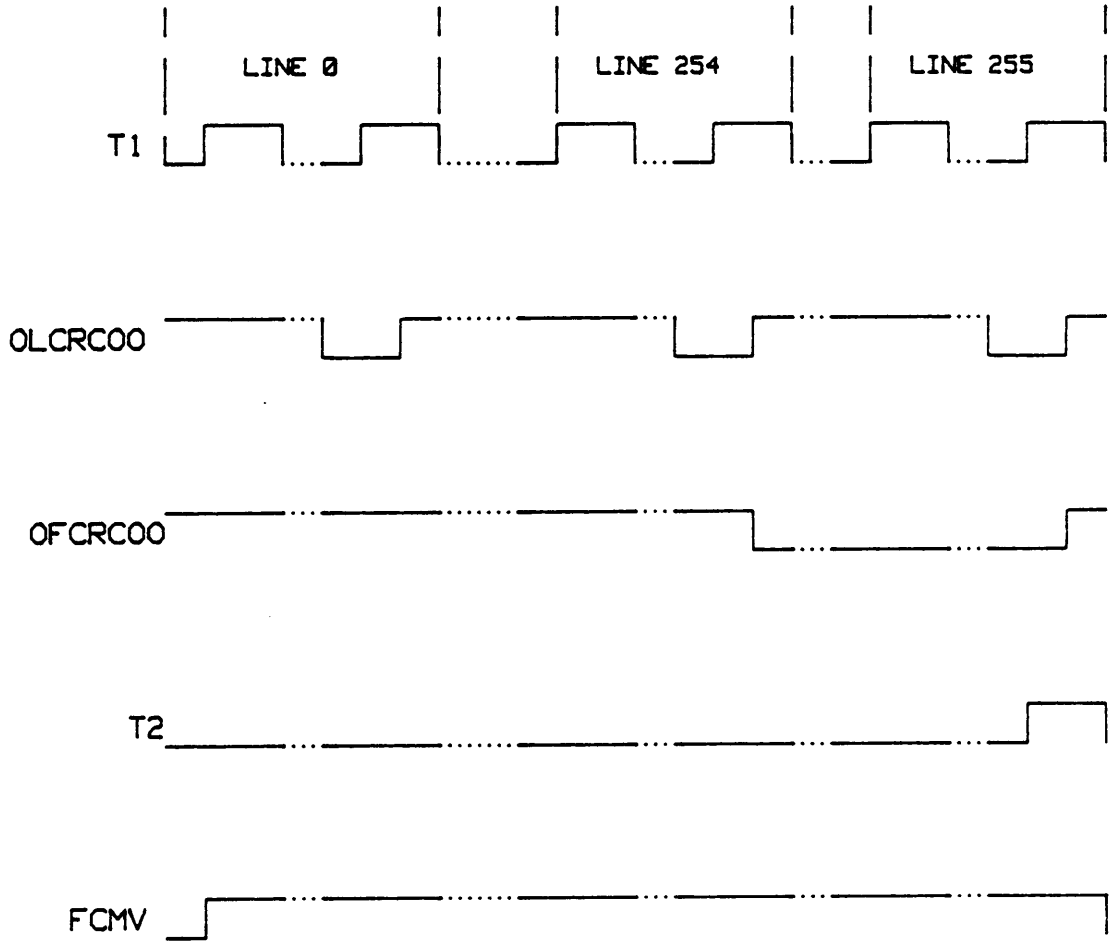
B.1 OUTPUT BUFFER TIMING DIAGRAMS.

The following set of drawings are the timing diagrams of the output buffer. They should be read in reference to the circuit diagrams and the PAL programs in the next two sections. Chapter 3 contains all the necessary explanation of the various signals. The signal generation is actually performed in the PALs from the available incoming signals.

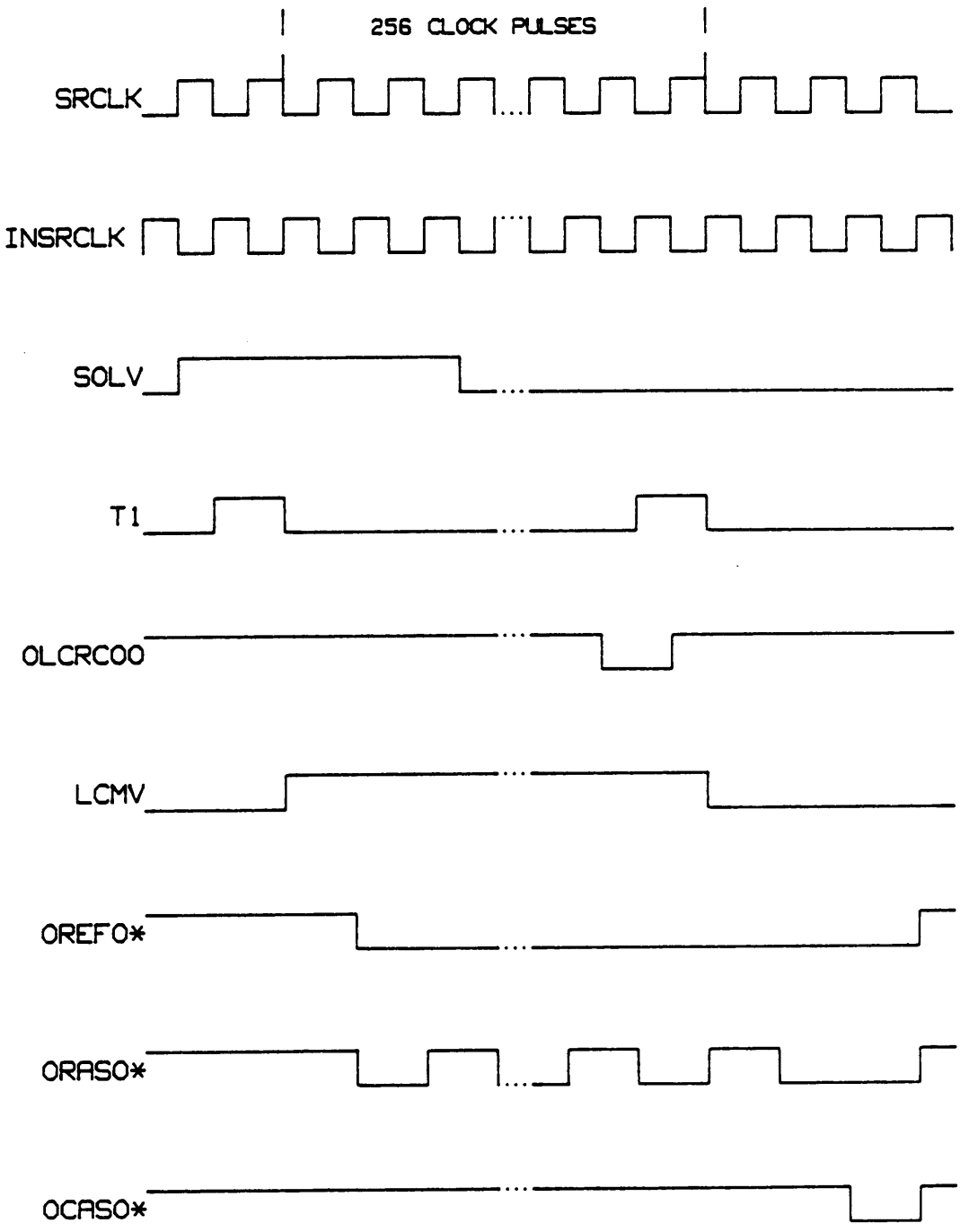
OUTPUT BUFFER: MEMORY CONTROL SIGNALS.
INPUT MEMORY



OUTPUT BUFFER: OUTPUT MEMORY FRAME MARKER.

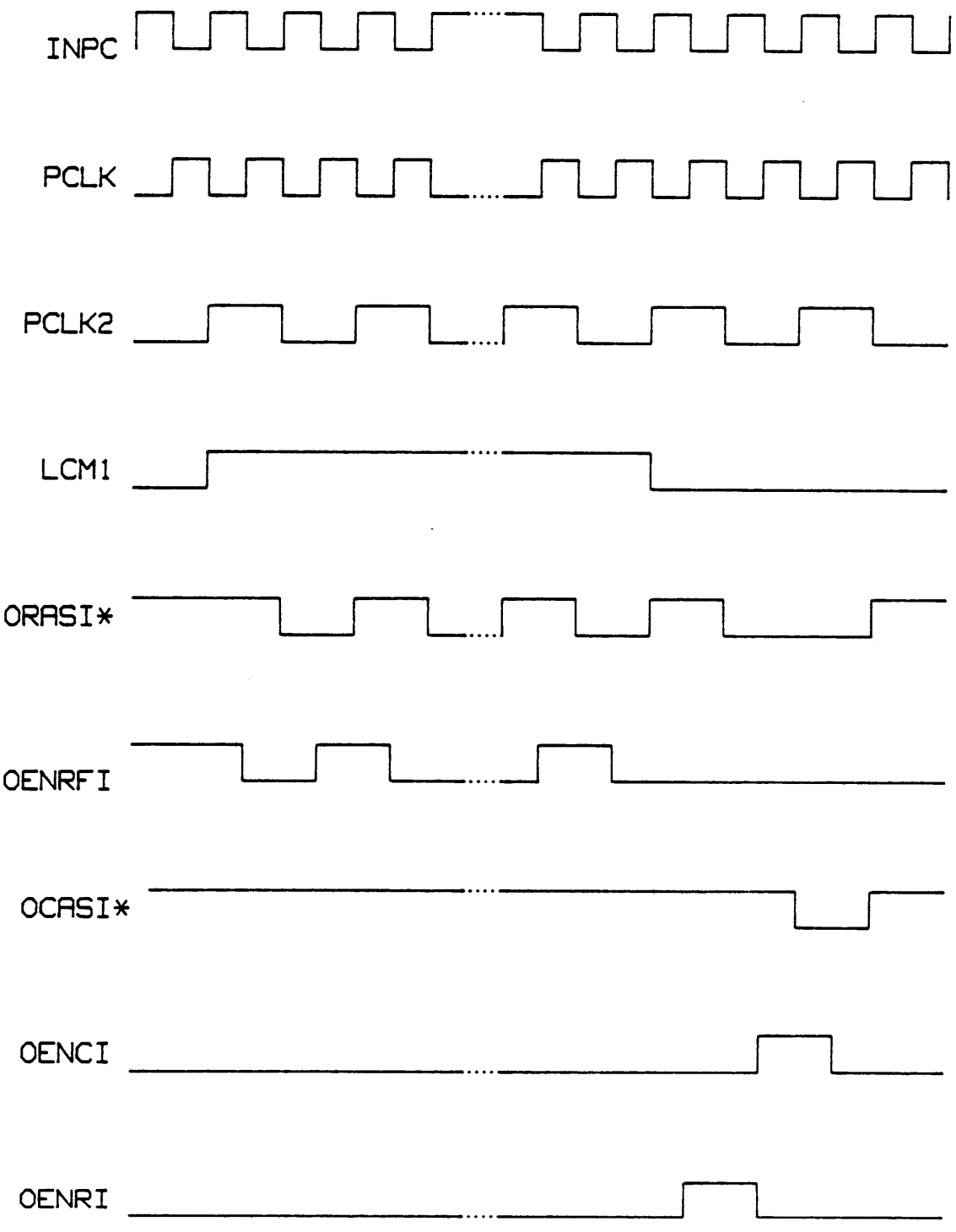


OUTPUT BUFFER: MEMORY CONTROL SIGNALS



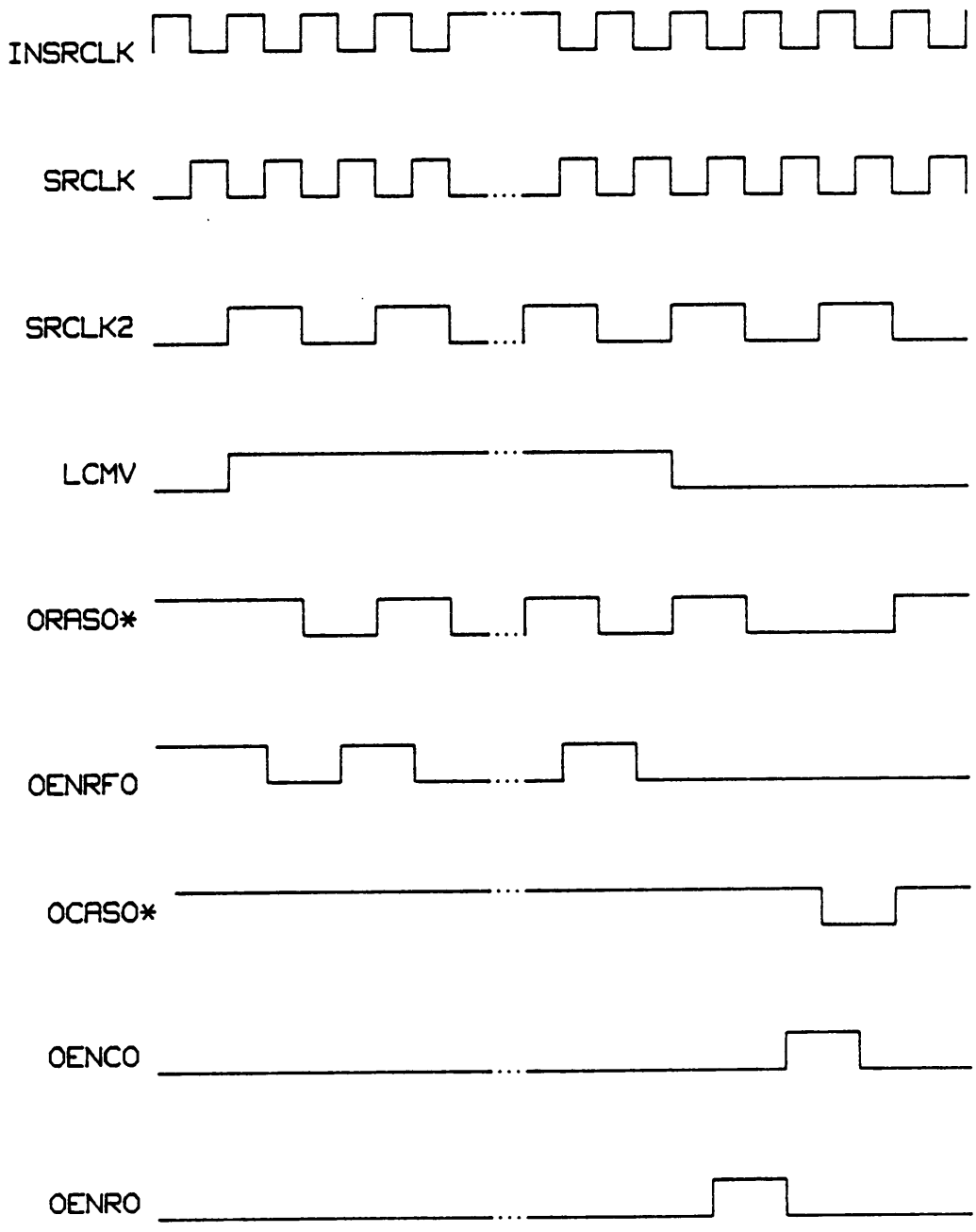
OUTPUT BUFFER

INPUT MEMORY ADDRESS CONTROL.



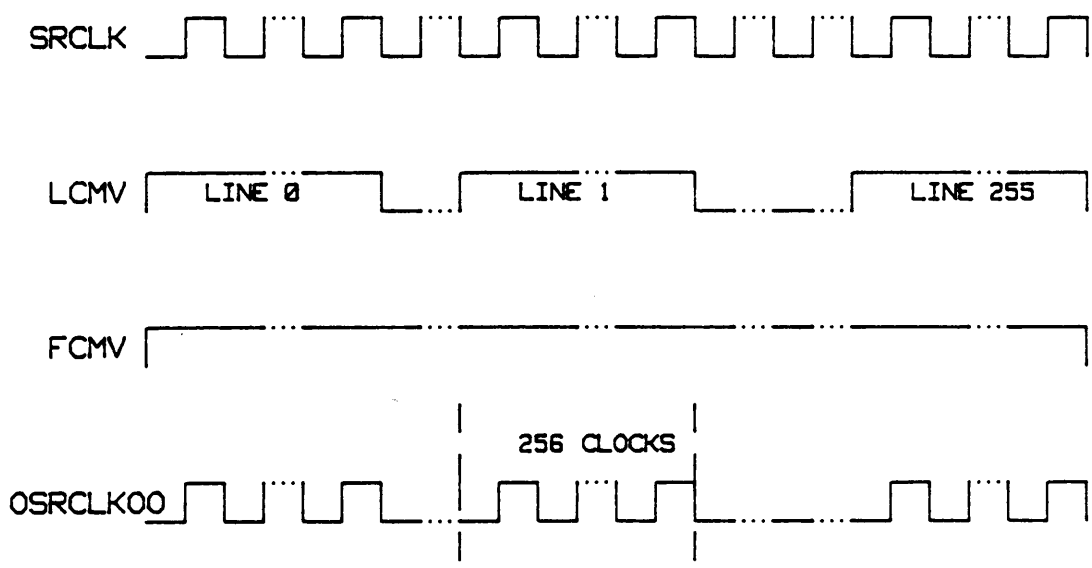
OUTPUT BUFFER

OUTPUT MEMORY ADDRESS CONTROL.

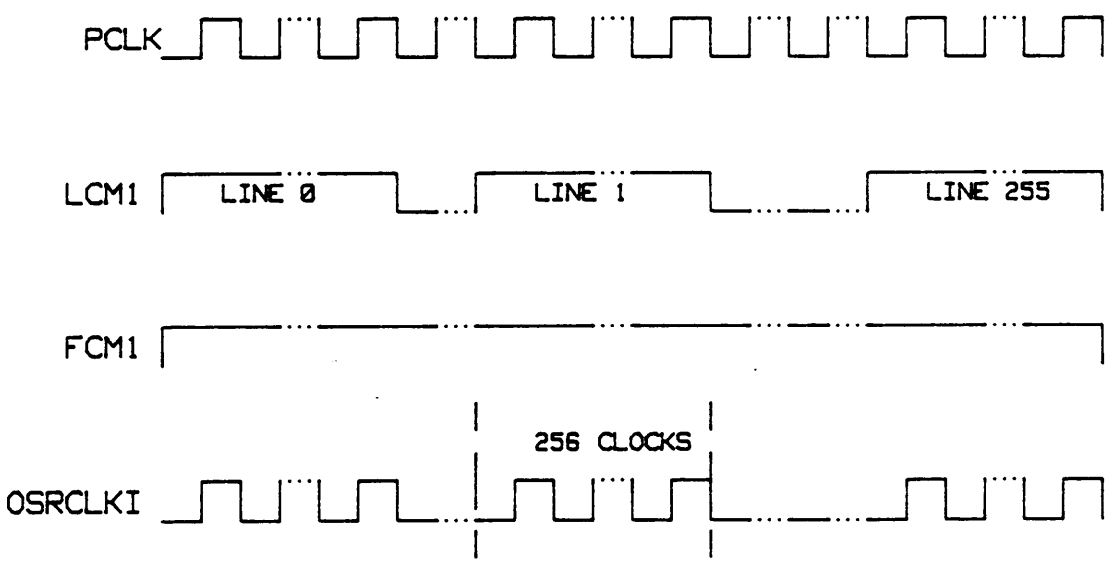


OUTPUT BUFFER

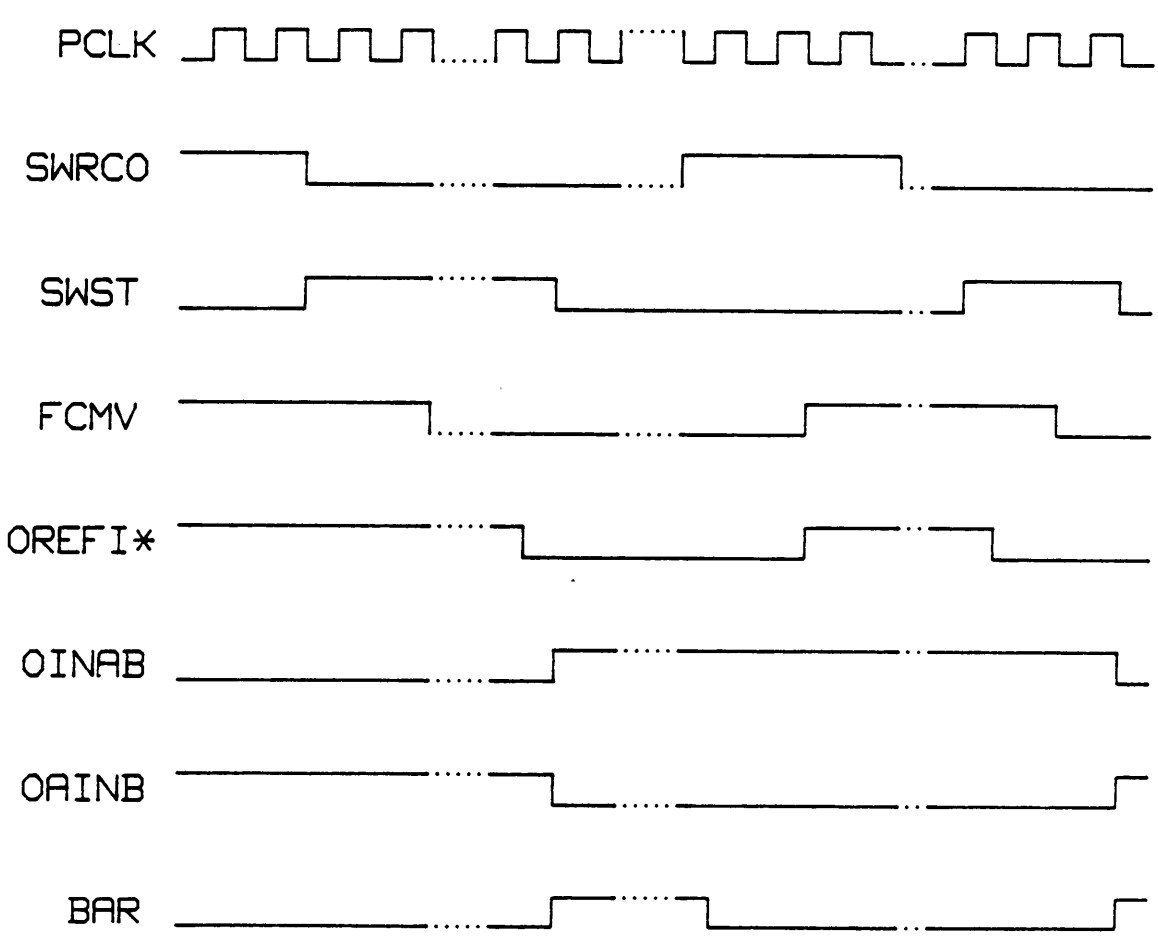
OUTPUT MEMORY SERIAL CLOCK



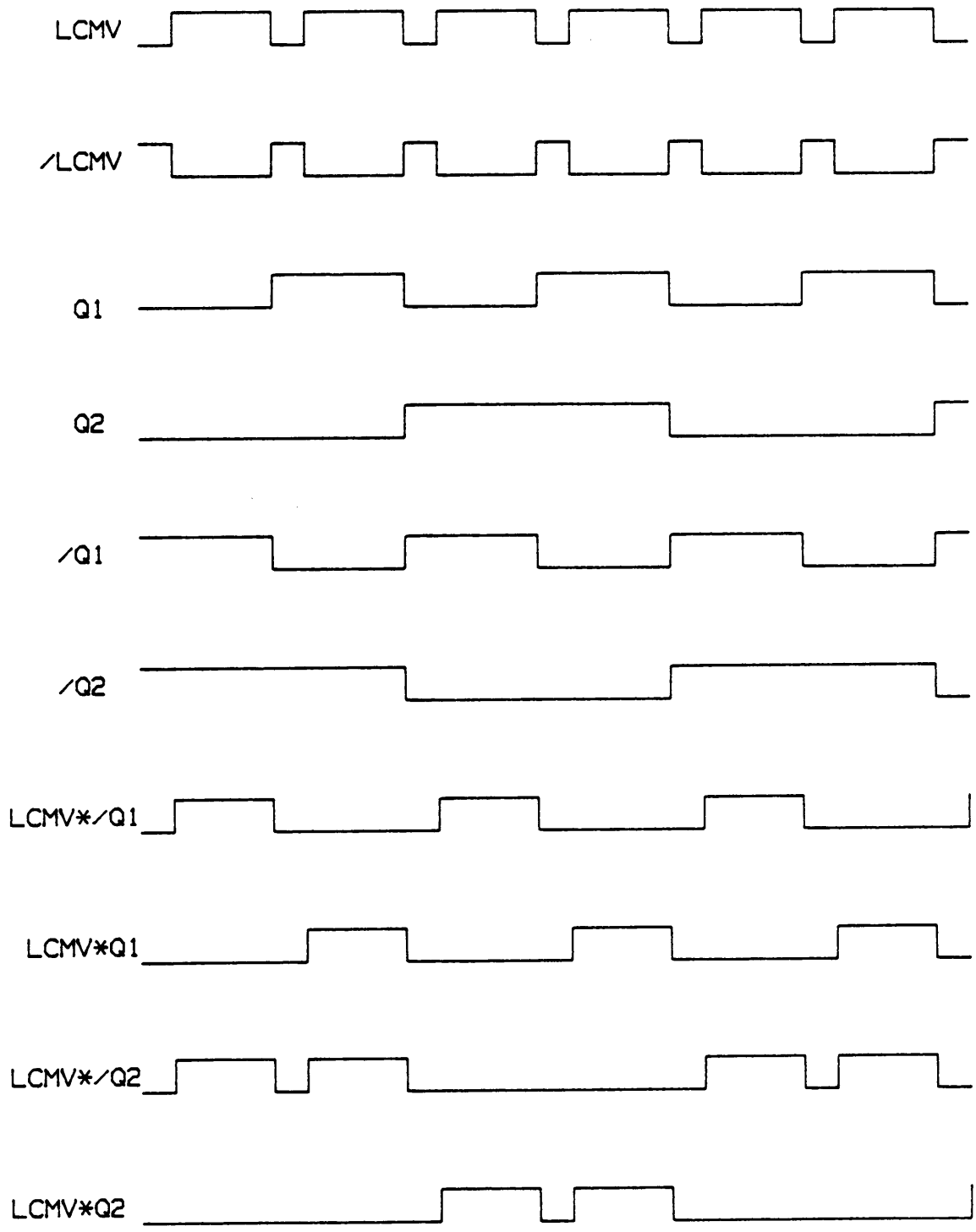
INPUT MEMORY SERIAL CLOCK



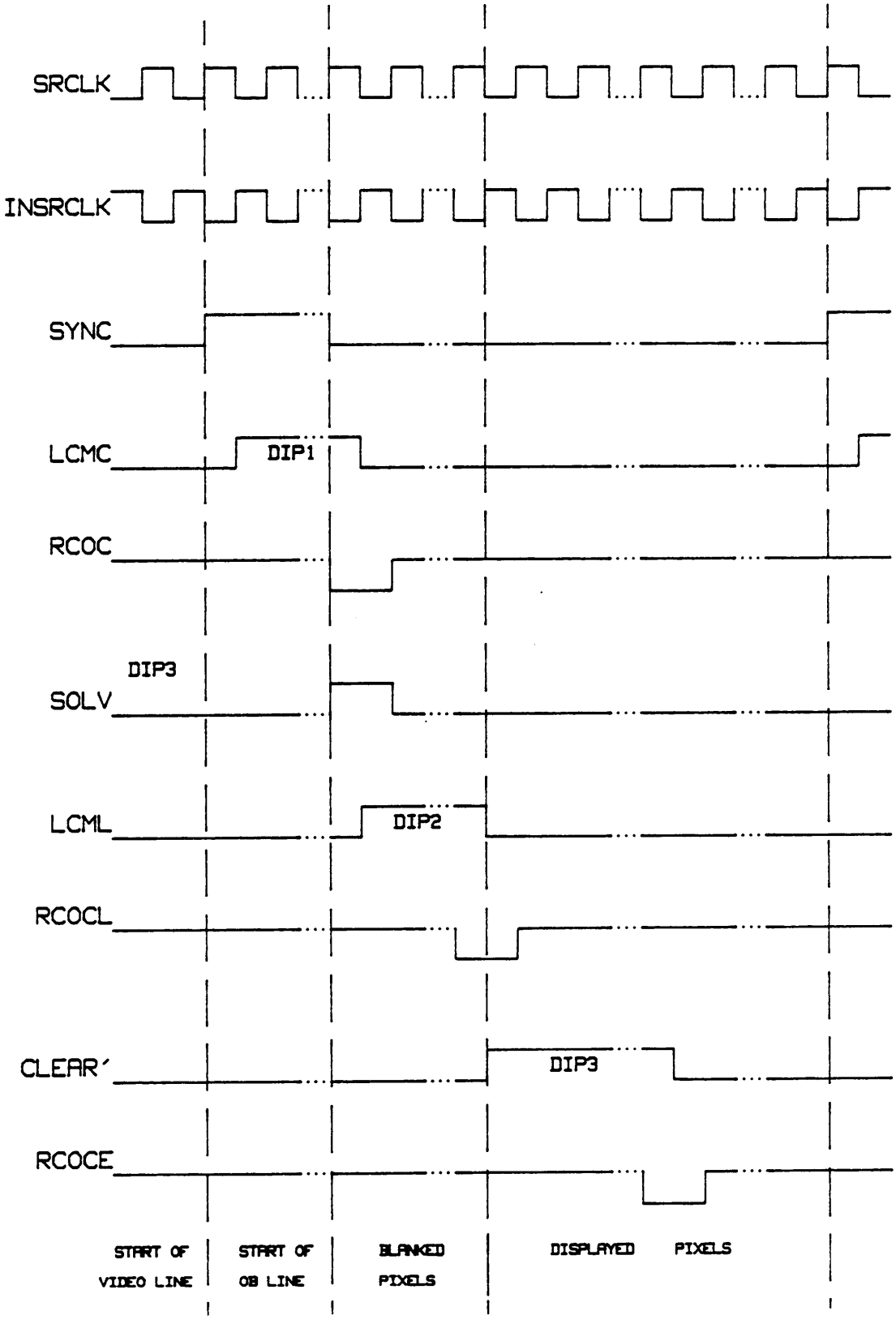
OUTPUT BUFFER: FRAME REFRESH SIGNAL.



BLANKING FORMAT SIGNALS



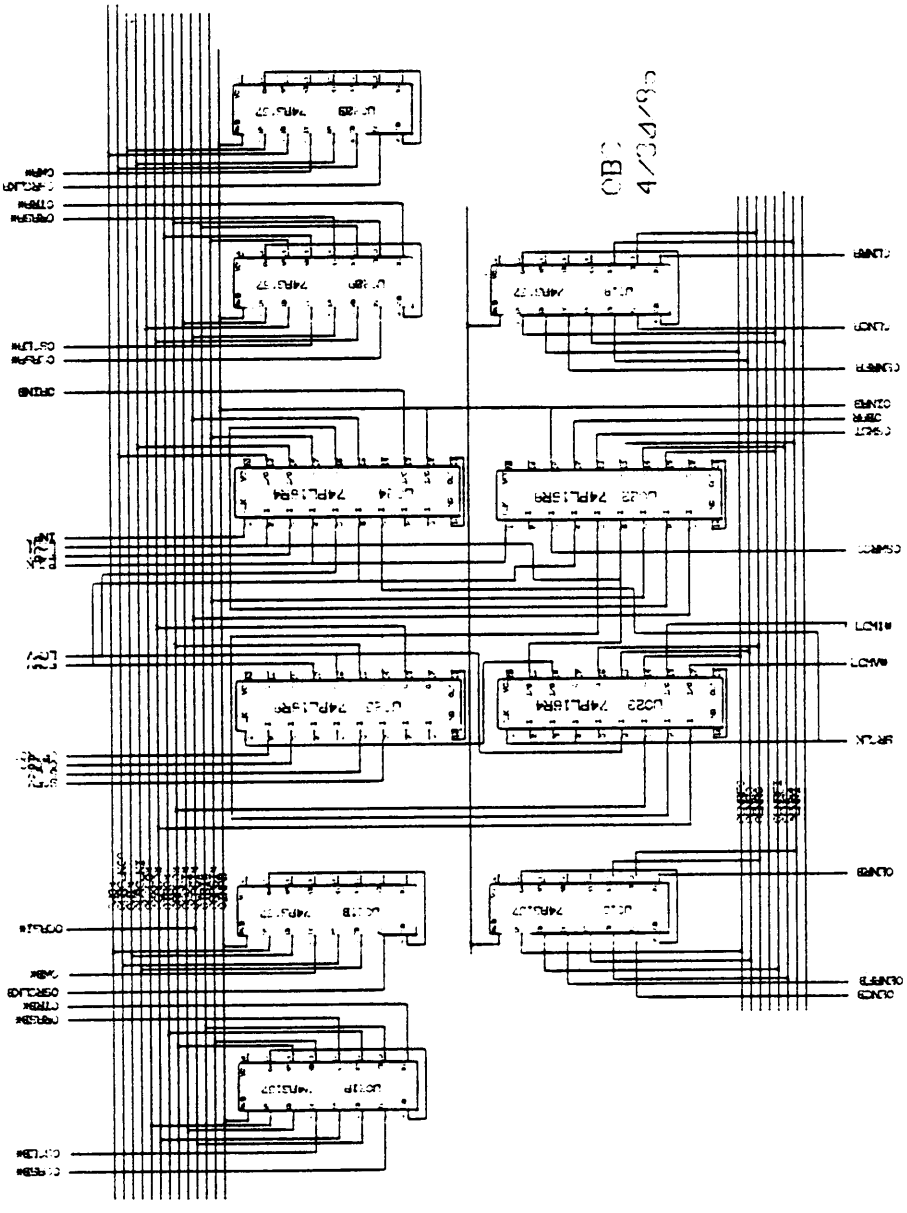
VDAC: PICTURE COMPOSITION.

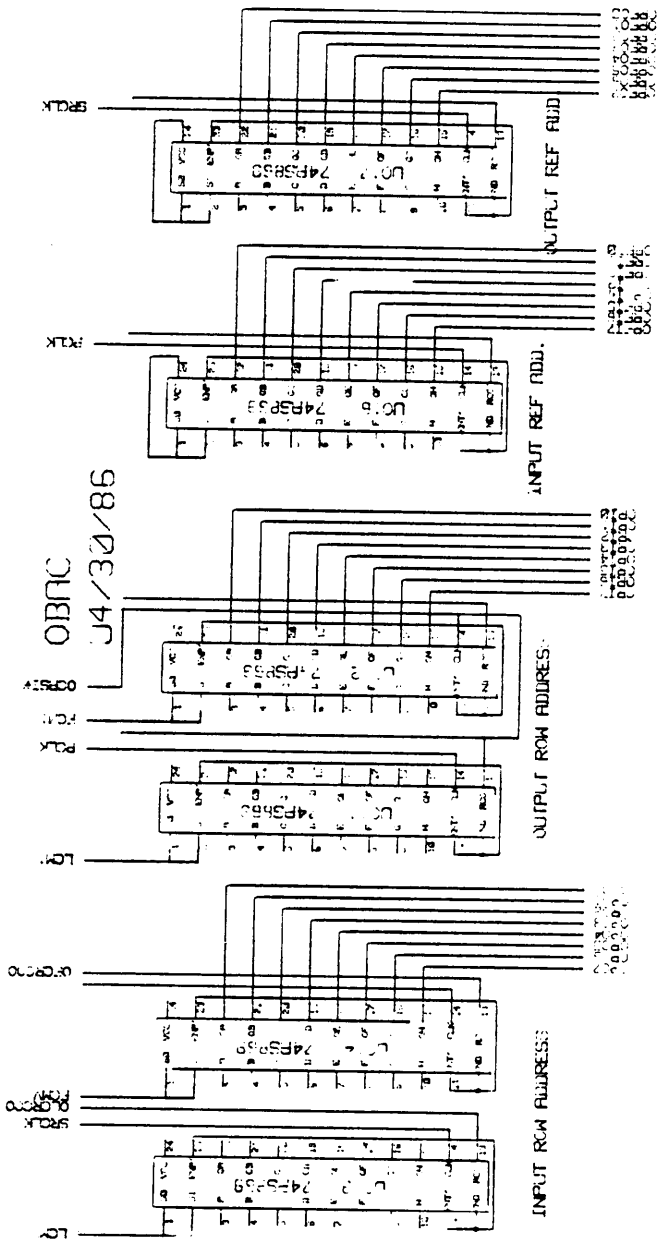


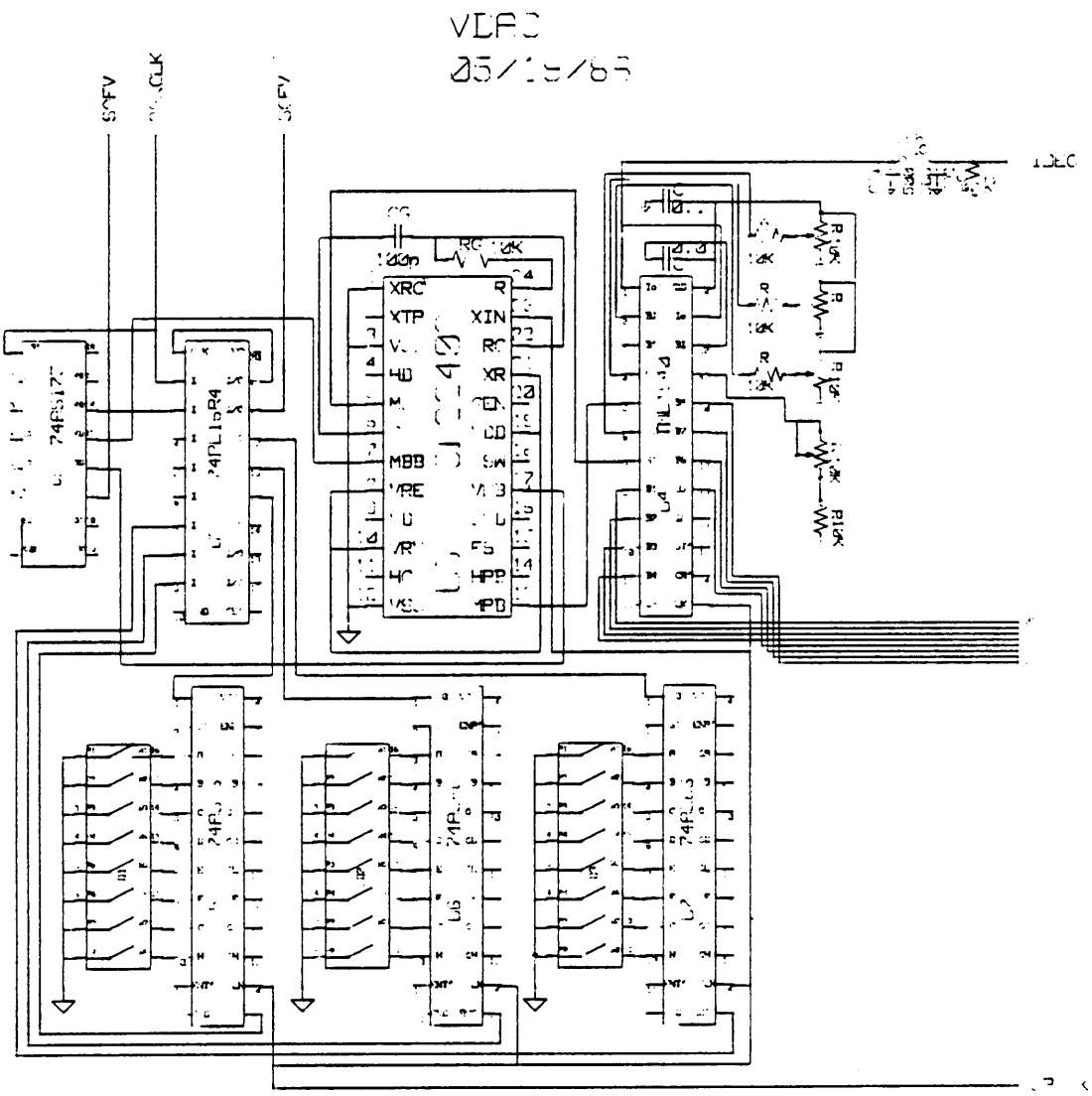
&rl.

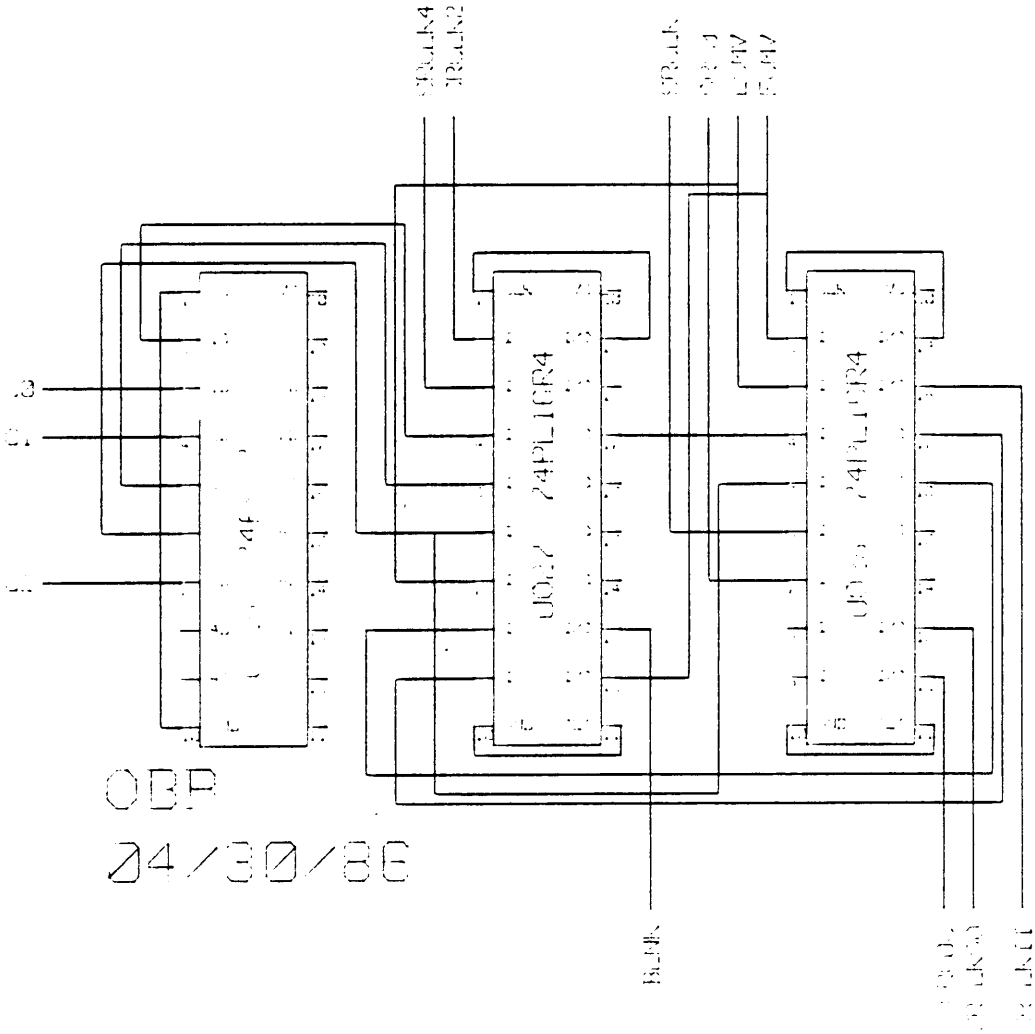
B.2 OUTPUT BUFFER CIRCUIT DIAGRAMS.

The following are the circuit diagrams for the output buffer. The details of all the output buffer blocks described in Chapter 3 are shown here.




OUTPUT REF ADD.
INPUT REF ADD.
OUTPUT ROW ADDRES
INPUT ROW ADDRESS




OBP
04/30/86

&rl.

B.3 OUTPUT BUFFER PAL PROGRAMS

This section contains the PAL programs in the PALASM language for all the PALS used in the output buffer.The program generated simulation results. The fuse plots and the listing of the JEDEC files used by the PAL programmer to program the PAL are not included in this section but are available on floppy disk.

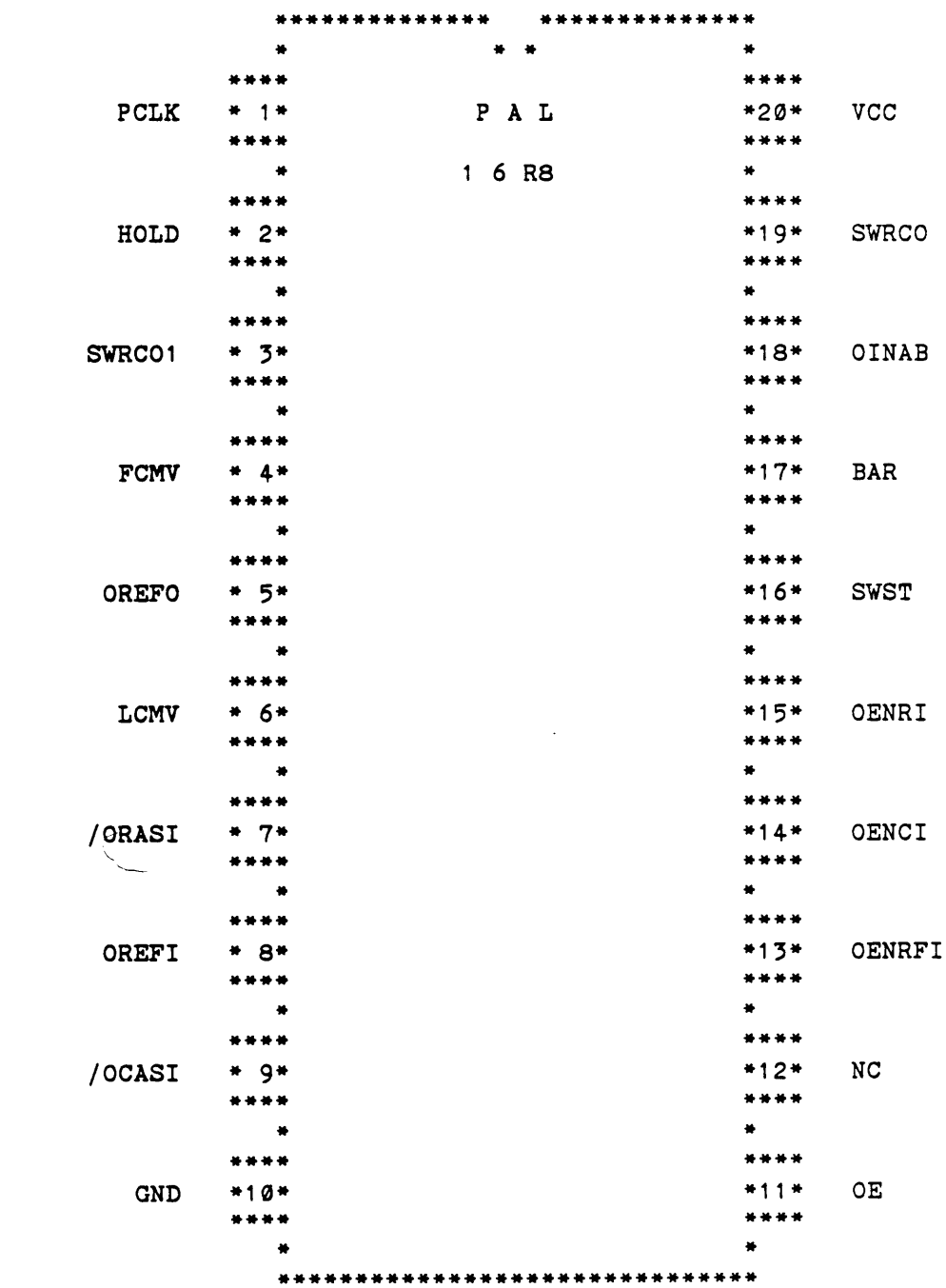
```
PAL16R8                    PAL DESIGN SPECIFICATION
PAT1004                    SANJAY ADKAR 04-14-1986
INPUT ADDRESS CONTROL PAL.-OUTPUT BUFFER U022
VIRGINIA TECH BLACKSBURG VA 24060


PCLK HOLD SWRCO1 FCMV OREFO LCMV /ORASI OREFI /OCASI GND
OE NC OENRFI OENCI OENRI SWST BAR OINAB SWRCO VCC

;THE ABOVE DESCRIBES THE PIN LIST OF THE PAL.

;EQUATIONS

/OENRFI :=     OENRFI              +
               /LCMV


/OENCI  :=      OENCI              +
                /OENRI


/OENRI :=      LCMV                +
               OREFI               +
               OENCI               +
               OENRI

/SWRCO :=      /SWRCO1            +
               /HOLD



/SWST :=       SWRCO * /SWST       +
               SWRCO */FCMV *OREFI * OREFO +
               BAR * /SWST         +
               BAR */FCMV * OREFI * OREFO +
               SWST * /FCMV * OREFO *OREFI

/BAR :=       /SWST * /BAR         +
               /OREFO * /BAR        +
               FCMV * /BAR         +
               /SWST * SWRCO       +
               /OREFO * SWRCO       +
               FCMV * SWRCO        +
               /OREFI * /BAR        +
               /OREFI * SWRCO
```

```
/OINAB :=      OINAB * SWST * /FCMV * OREFO * OREFI +
              /OINAB * /SWST       +
              /OINAB *FCMV         +
              /OINAB * /OREFO       +
              /OINAB * /OREFI
```

FUNCTION TABLE

```
PCLK /ORASI /OCASI OREFI LCMV OENRFI OENRI OENCI
SWRCO FCMV OREFO SWST BAR OINAB HOLD SWRCO1

;    /  /       O                              S
;    O  O  O    E  O  O  S     O        O     W
; P  R  C  R  L  N  E  E  W  F  R  S     I  H  R
; C  A  A  E  C  R  N  N  R  C  E  W  B  N  O  C
; L  S  S  F  M  F  R  C  C  M  F  S  A  A  L  O
; K  I  I  I  V  I  I  I  O  V  O  T  R  B  D  1

-------------------------------------------------------
  C  H  H  H  H  H  L  L  L  H  L  H  L  L  H  L
  C  L  H  L  H  L  L  L  L  H  L  H  L  L  H  L
  C  H  H  L  H  H  L  L  L  L  L  H  L  L  H  L
  C  L  H  L  H  L  L  L  L  L  H  H  L  L  H  L
  C  H  H  L  L  L  H  L  L  L  H  H  L  L  H  L
  C  L  H  L  L  L  L  H  L  L  H  H  L  L  H  L
  C  L  L  L  L  L  L  L  L  L  H  H  L  L  H  L
  C  H  H  H  L  L  L  L  L  L  H  L  H  H  H  L
  C  H  H  H  L  L  L  L  H  L  H  L  H  H  H  H
  C  H  H  H  L  L  L  L  H  L  H  L  L  H  H  H
-------------------------------------------------------
;
DESCRIPTION:THIS PAL DOES THE FOLLOWING FUNCTIONS:
A) IT GENERATES ALL THE ADDRESS CONTROL SIGNALS REQUIRED FOR THE
INPUT MEMORY ADDRESS BUS.  THE TRI-STATE MULTIPLEXER BUFFERS
ARE CLOCKED USING THESE SIGNALS.
B) IT GENERATES THE REFRESH FRAME SIGNAL (IINAB).
```

INPUT ADDRESS CONTROL PAL.-OUTPUT BUFFER U022



```
PAL16R4                    PAL DESIGN SPECIFICATIONS
PAT1006                    SANJAY ADKAR  04-13-1986
OUTPUT ADDRESS CONTROL PAL - OUTPUT BUFFER U023
VIRGINIA TECH , BLACKSBURG, VA.


SRCLK NC NC SCLK NC LCMV /ORASO OREFO /OCASO GND
OE LCMV' LCM1' OENRFO OENCO OENRO NC INSRCLK LCM1 VCC

;THE ABOVE DESCRIBES THE PIN LIST OF THE PAL

;EQUATIONS

/OENRFO :=     OENRFO
          +     /LCMV

/OENCO  :=     /OENRO
          +     OENCO

/OENRO  :=    OENRO
          +   OREFO
          +   OENCO
          +   LCMV

IF (VCC) /INSRCLK =      SCLK

IF (VCC) /LCM1'   =      LCM1

IF (VCC) /LCMV'   =      LCMV
```

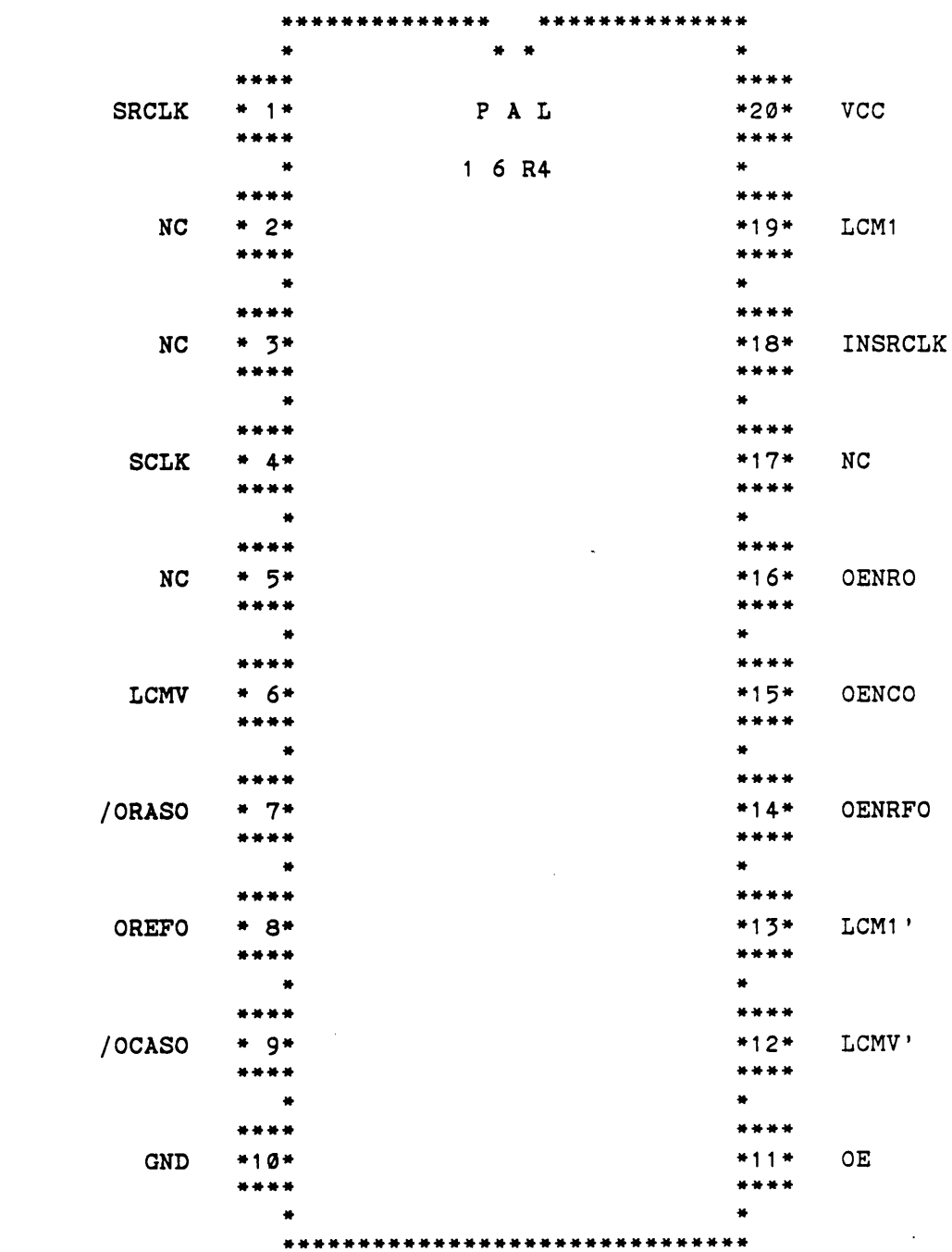
FUNCTION TABLE

SRCLK /ORASO /OCASO OREFO LCMV SCLK
OENRFO OENCO OENRO INSRCLK LCM1 LCM1' LCMV'

```
;                       O         I
;     /   /             E         N
; S   O   O   O         N   O   O   S       L   L
; R   R   C   R   L   S R   E   E   R   L   C   C
; C   A   A   E   C   C E   N   N   C   C   M   M
; L   S   S   F   M   L F   C   R   L   M   1   V
; K   O   O   O   V   K O   O   O   K   1   '   '
-------------------------------------------------
  C   H   H   H   H   H H   L   L   L   H   L   L
  C   L   H   L   H   L L   L   L   H   L   H   L
  C   H   H   L   H   H H   L   L   L   H   L   L
  C   L   H   L   H   L L   L   L   H   L   H   L
  C   H   H   L   L   H L   L   H   L   H   L   H
  C   L   H   L   L   L L   H   L   H   L   H   H
  C   L   L   L   L   H L   L   L   L   H   L   H
  C   H   H   H   L   L L   L   L   H   L   H   H
-------------------------------------------------
```

DESCRIPTION: THIS PAL GENERATES ALL THE CONTROLS REQUIRED TO MULTIPLEX THE INPUT ROW, COLOUMN AND THE REFRESH ADDRESS ON THE ADDRESS BUS. IT ALSO GENERATES THE COMPLEMENTS OF THE INPUT AND OUTPUT LINE MARKERS, THESE SIGNALS ARE USED TO CONTROL THE REFRESH AND THE MEMORY TRANSFER CYCLES.

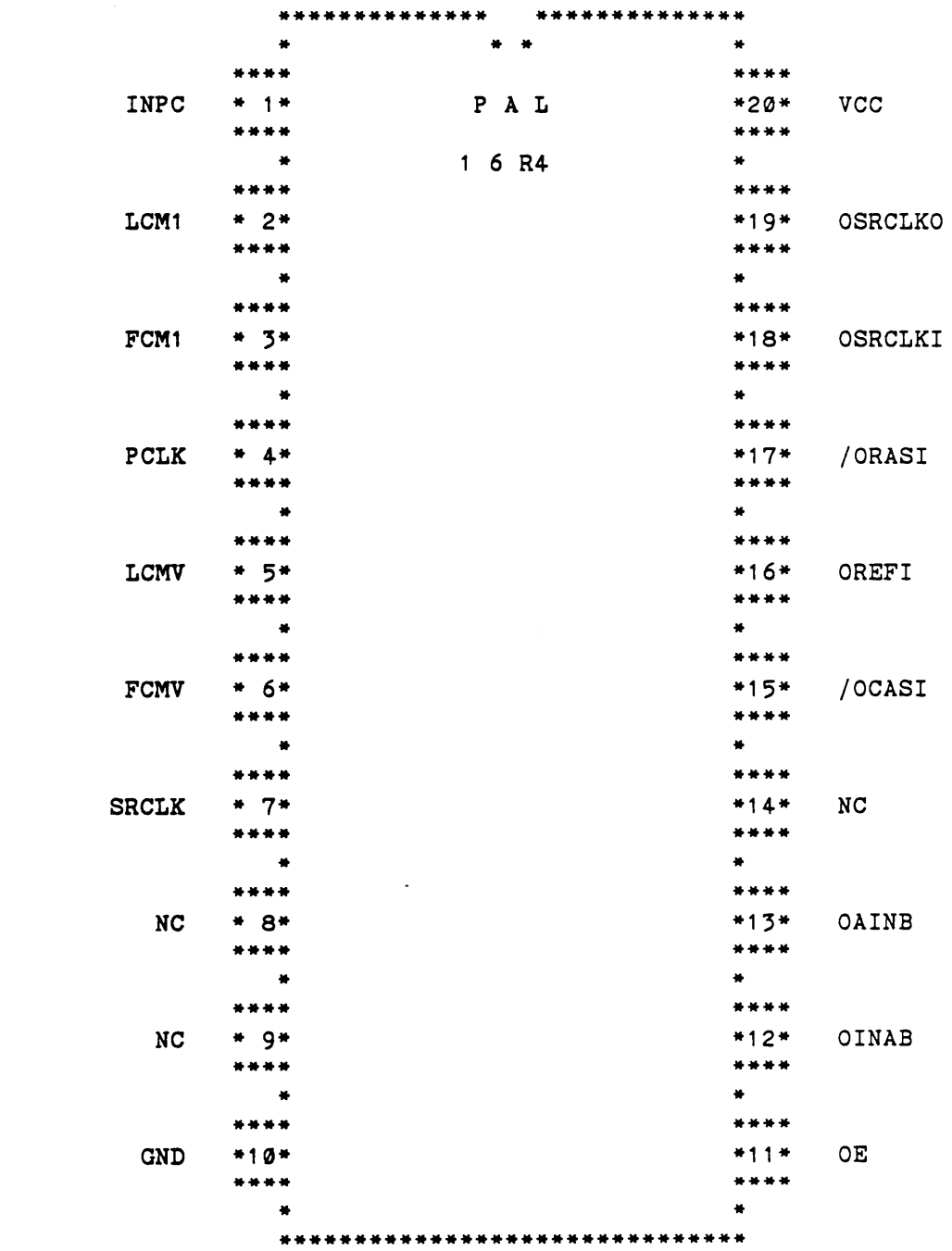
OUTPUT ADDRESS CONTROL PAL - OUTPUT BUFFER U023



```
PAL16R4                    PAL DESIGN SPECIFICATIONS
PAT1002                    SANJAY ADKAR 04-13-1986
OUTPUT BUFFER INPUT MEMORY CONTROL PAL  UI 24
VIRGINIA TECH, BLACKSBURG, VA 24061


INPC LCM1 FCM1 PCLK LCMV FCMV SRCLK NC NC GND
OE OINAB OAINB NC /OCASI OREFI /ORASI  OSRCLKI OSRCLKO VCC

;THE ABOVE DESCRIBES THE PIN LIST FOR THE PAL.

;EQUATIONS

/OREFI := /OCASI * LCM1                +
          /OREFI * /OCASI

ORASI  := /LCM1 * /OREFI * /OCASI      +
          /OREFI * /ORASI  * /OCASI    +
          /ORASI * /OCASI * LCM1       +
          /LCM1 * /OCASI * ORASI

OCASI  := ORASI * /OREFI * /OCASI * /LCM1

IF (VCC) /OSRCLKI = /PCLK              +
                    /LCM1              +
                    /FCM1

IF (VCC) /OSRCLKO = /SRCLK             +
                     /LCMV              +
                     /FCMV


IF (VCC) /OAINB  = OINAB
```

```
FUNCTION TABLE

INPC LCM1 OREFI /ORASI /OCASI
PCLK FCM1 OSRCLKI SRCLK LCMV FCMV OSRCLKO OINAB OAINB

;                          O           O
;                 /  /     S           S
;          O  O  O        R  S        R  O  O
; I  L  R  R  C  P  F  C  R  L  F  C  I  A
; N  C  E  A  A  C  C  L  C  C  C  L  N  I
; P  M  F  S  S  L  M  K  L  M  M  K  A  N
; C  1  I  I  I  K  1  I  K  V  V  O  B  B
-------------------------------------------------
  C  L  H  H  H  H  L  L  H  H  H  H  L  H
  C  H  L  L  H  L  H  L  L  H  H  L  H  L
  C  H  L  H  H  H  H  H  H  H  H  H  L  H
  C  H  L  L  H  L  H  L  L  H  H  L  H  L
  C  H  L  H  H  H  H  H  H  H  H  H  L  H
  C  L  L  L  H  H  H  L  H  L  H  L  L  H
  C  L  L  L  L  L  H  L  L  L  H  L  H  L
  C  L  H  H  H  H  H  L  H  H  H  H  L  H
-------------------------------------------------

DESCRIPTION: THIS PAL PERFORMS THE FOLLOWING FUNCTIONS:
A) GENERATES THE MEMORY CONTROL SIGNALS FOR SHIFT REGISTER TO
   MEMORY TRANSFER.
B) GENERATES THE CONTROL SIGNALS TO REFRESH THE INPUT MEMORY.
C) GENERATES THE SERIAL SHIFT REGISTER CLOCKS FOR BOTH THE INPUT
   AND THE OUTPUT MEMORY.
D) INVERTS THE REFRESH SIGNAL.
```

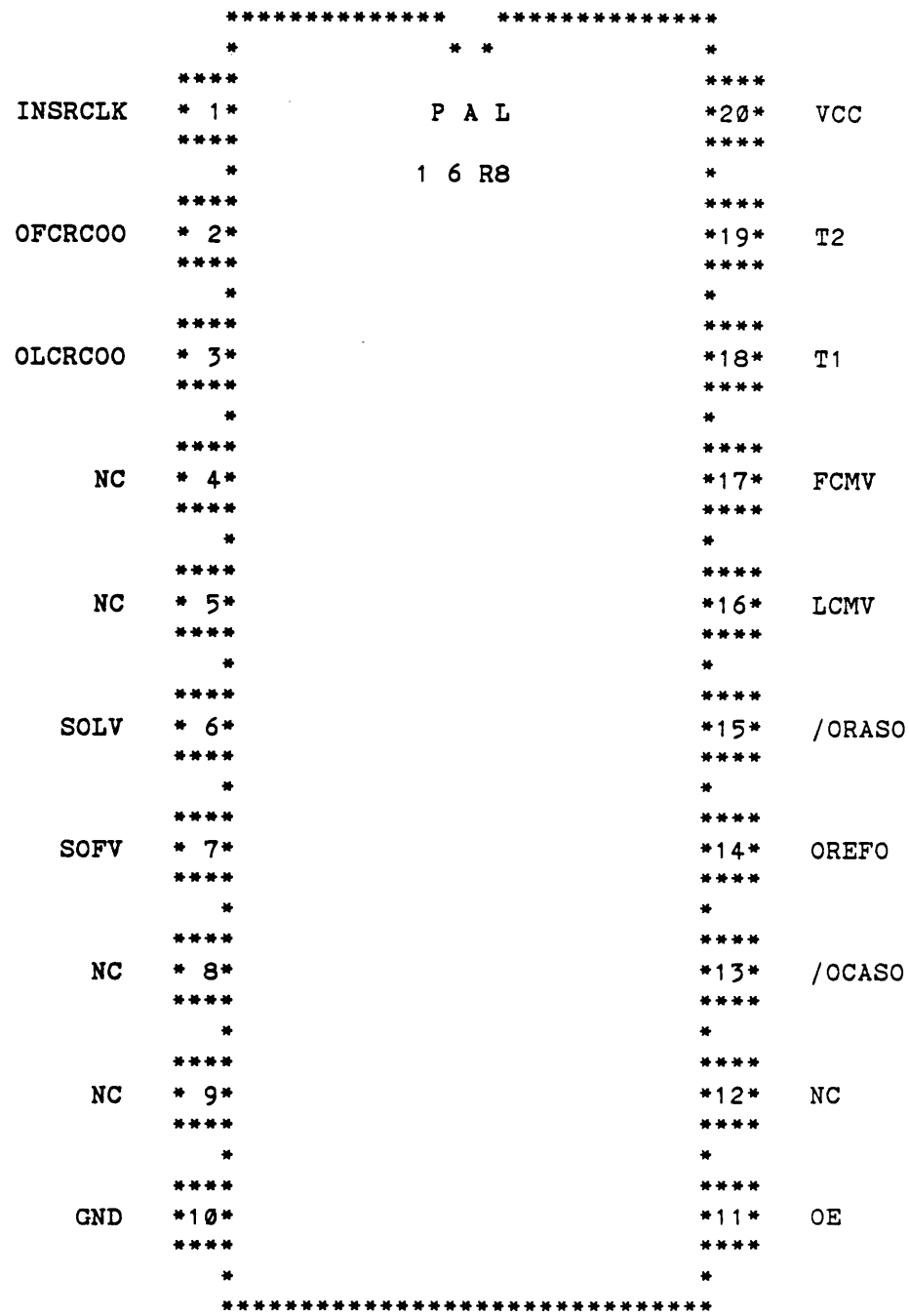
OUTPUT BUFFER INPUT MEMORY CONTROL PAL UI 24



```
PAL16R8   PAL DESIGN SPECIFICATIONS
PAT1008                  SANJAY ADKAR 04-14-86
OUTPUT MEMORY CONTROL PAL - OUTPUT BUFFER UO 25
VIRGINIA TECH BLACKSBURG VA 24061

INSRCLK OFCRCOO OLCRCOO NC NC SOLV SOFV NC NC  GND
OE NC /OCASO OREFO /ORASO LCMV FCMV T1 T2 VCC

;THE ABOVE DESCRIBES THE PIN LIST FOR THE PAL.

;EQUATIONS

/T1    :=      T1                    +
               /SOLV *OLCRCOO         +
               /SOLV * /LCMV          +
               LCMV *OLCRCOO

/T2    :=      OLCRCOO               +
               OFCRCOO               +
               T2


/FCMV  :=      /SOFV * /FCMV         +
               /SOFV * T1 * T2       +
               /T1 * /FCMV           +
               FCMV * T1 * T2


/LCMV  :=      LCMV *T1              +
               /LCMV * /T1


OCASO  :=       ORASO * /OREFO * /OCASO */LCMV


ORASO  :=     /LCMV * /OREFO */OCASO   +
              /OREFO * /ORASO * /OCASO +
              /ORASO * /OCASO * LCMV   +
              /LCMV * /OCASO * ORASO


/OREFO :=      /OCASO * LCMV         +
              /OREFO * /OCASO
```

```
FUNCTION TABLE

INSRCLK SOFV SOLV OLCRCOO OFCRCOO
FCMV LCMV T1 T2 /ORASO /OCASO OREFO

; I        O  O                        /  /
; N        L  F                        /  /
; S        C  C  O                     O  O  O
; R  S  S  R  R  F  L                  R  C  R
; C  O  O  C  C  C  C                  A  A  E
; L  F  L  O  O  M  M  T  T  S  S  F
; K  V  V  O  O  O  V  1  2  O  O  O

-----------------------------------------------
  C  L  L  H  H  L  L  L  L  H  H  H
  C  H  L  H  H  L  L  L  L  H  H  H
  C  H  H  H  H  L  L  H  L  H  H  H
  C  H  X  H  H  H  H  L  L  H  H  H
  C  X  X  H  H  H  H  L  L  L  H  L
  C  X  X  H  H  H  H  L  L  H  H  L
  C  X  X  L  H  H  H  H  L  L  H  L
  C  X  X  H  H  H  L  L  L  H  H  L
  C  X  L  H  H  H  L  L  L  L  H  L
  C  X  L  H  H  H  L  L  L  L  L  L
  C  X  L  H  H  H  L  L  L  H  H  H
  C  X  H  H  L  H  L  H  L  H  H  H
  C  L  X  H  L  H  H  L  L  H  H  H
  C  L  X  H  L  H  H  L  L  L  H  L
  C  L  X  H  L  H  H  L  L  H  H  L
  C  L  X  L  L  H  H  H  H  L  H  L
  C  L  X  H  H  L  L  L  L  H  H  L
-----------------------------------------------
;

DESCRIPTION: THIS PAL GENERATES THE OUTPUT MEMORY LINE AND FRAME
MARKERS FROM THE SOL AND SOF SIGNALS.
IT ALSO GENERATES ALL THE OUTPUT MEMORY CONTROL SIGNALS  REQUIRED
TO CONTROL THE MEMORY.
```

OUTPUT MEMORY CONTROL PAL - OUTPUT BUFFER UO 25

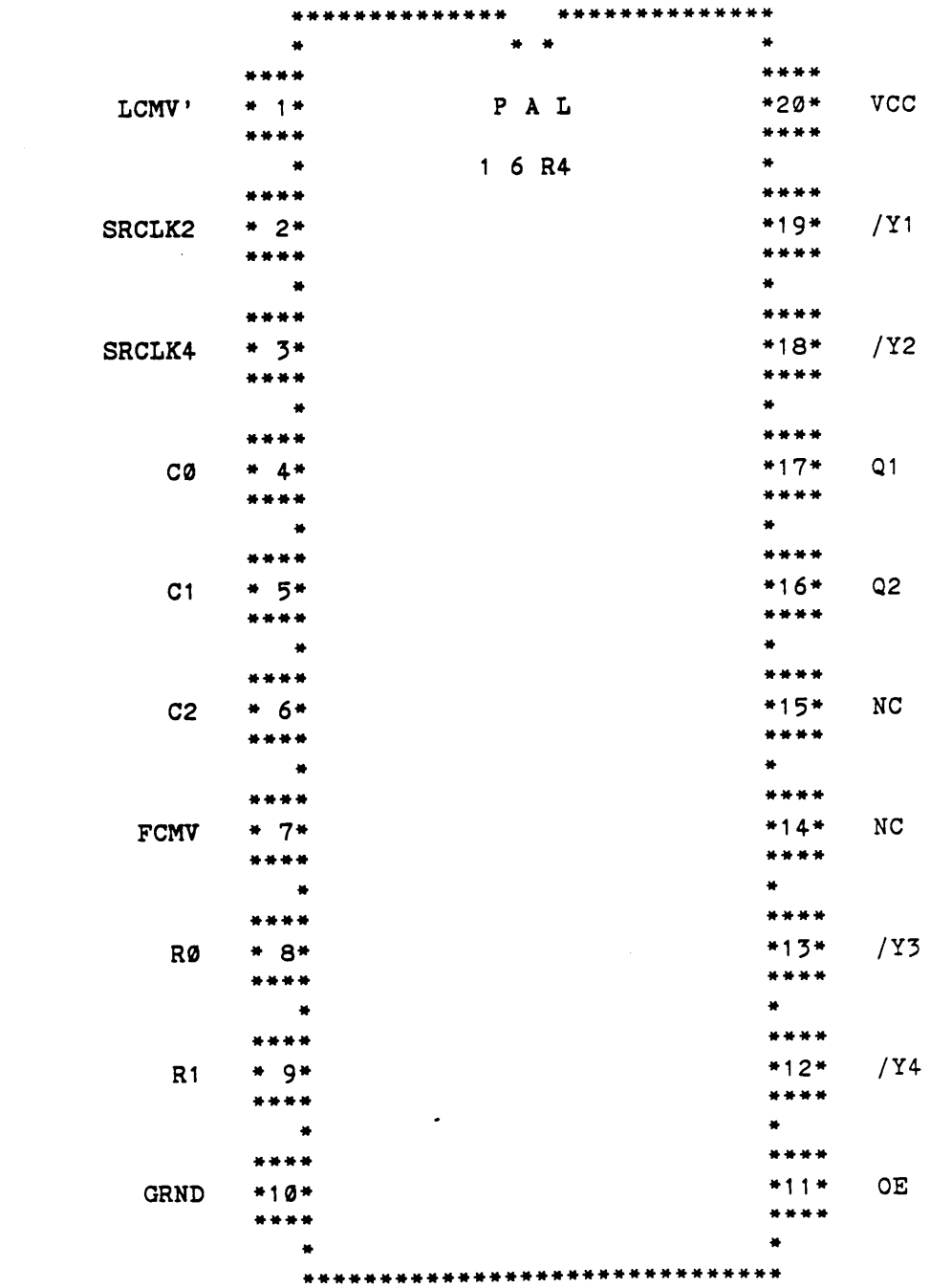


```
PAL16R4                    PAL DESIGN SPECIFICATIONS
PAT1009                    SANJAY ADKAR 05-21-86
FORMAT ONE PAL --- OUTPUT BUFFER.UO 27A
VIRGINIA TECH BLACKSBURG VA 24060.


LCMV' SRCLK2 SRCLK4 C0 C1 C2 FCMV R0 R1 GRND
OE /Y4 /Y3 NC NC Q2 Q1 /Y2 /Y1 VCC

;EQUATIONS.

/Q1    :=  Q1              +
           /FCMV


/Q2    :=  Q1   *  Q2      +
           /Q1   *  /Q2    +
           /FCMV

Y1      = /C2 * /C1 * C0 * /R0 * Q1              +
          /C2 * /C1 * C0 * R0 * /Q1              +
          /C2 * C1 * C0 * /R0 * SRCLK2           +
          /C2 * C1 * C0 * R0 * /SRCLK2           +
          C2 * C1  * C0 * /R0 * SRCLK4           +
          C2 * C1  * C0 * R0 * /SRCLK4           +
          C2 * /C1 * C0 * /R0 * Q2


Y2      = /C2 * C1 * /C0 * /R0 * /R1 * SRCLK2 +
          /C2 * C1 * /C0 * /R0 * /R1 * Q1      +
          /C2 * C1 * /C0 * R0 * /R1 * /SRCLK2 +
          /C2 * C1 * /C0 * R0 * /R1 * /Q1      +
          /C2 * C1 * /C0 * R0 * R1 * SRCLK2    +
          /C2 * C1 * /C0 * R0 * R1 * /Q1       +
          /C2 * C1 * /C0 * /R0 * R1 * /SRCLK2


Y3      = C2 * C1 * /C0 * /R0 * /R1 * SRCLK4  +
          C2 * C1 * /C0 * /R0 * /R1 * Q2       +
          C2 * C1 * /C0 * R0 * /R1 * /SRCLK4   +
          C2 * C1 * /C0 * R0 * /R1 * /Q2       +
          C2 * C1 * /C0 * R0 * R1 * SRCLK4     +
          C2 * C1 * /C0 * R0 * R1 * /Q2        +
          C2 * C1 * /C0 * /R0 * R1 * /SRCLK4
```

```
Y4      = C2 * /C1 * C0 * R0 * /Q2              +
          /C2 * C1 * /C0 * /R0 * R1 * Q1        +
          C2 * C1 * /C0 * /R0 * R1 * Q2
```

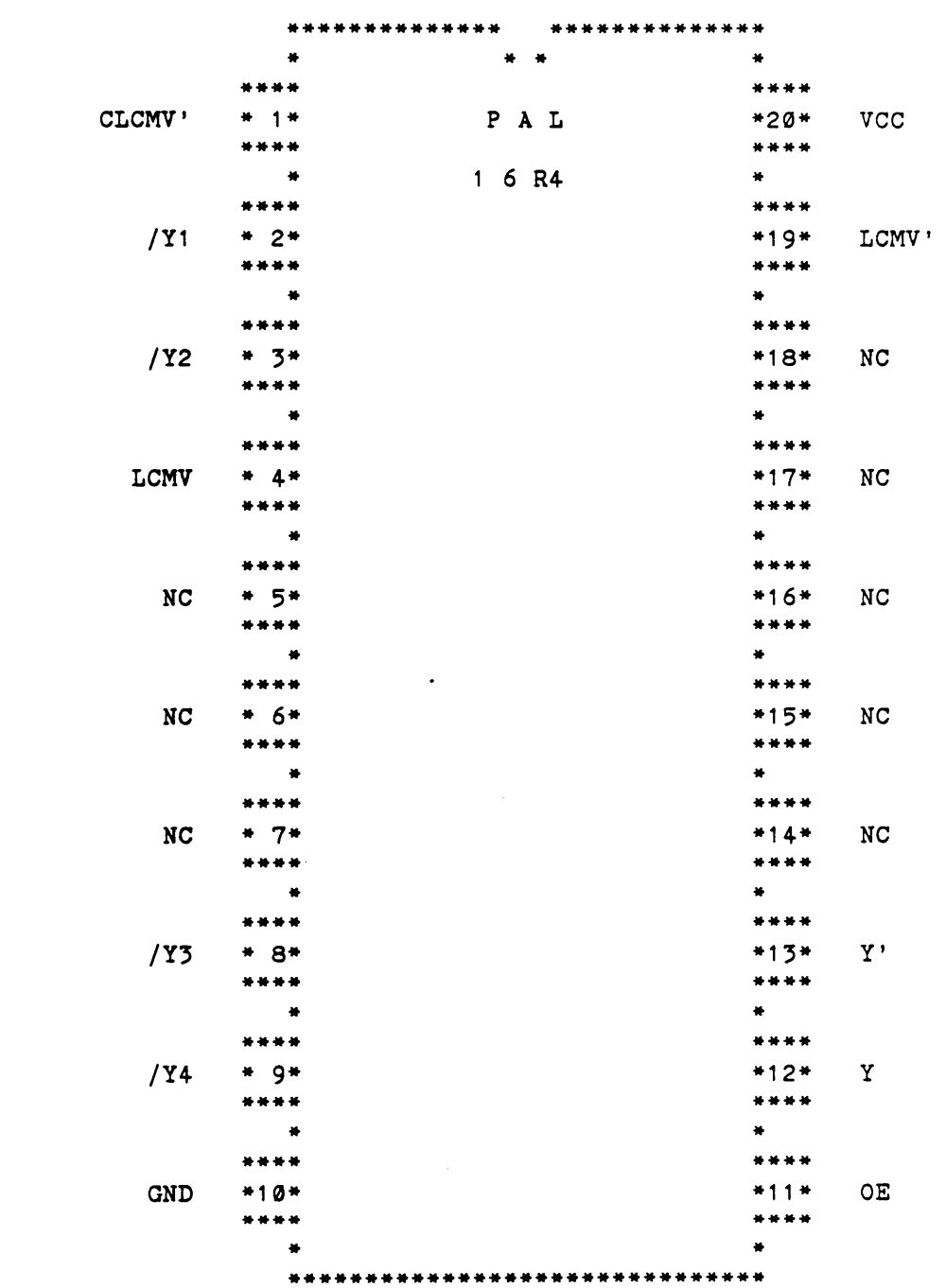
DESCRIPTION: THIS PAL GENERATES THE FORMAT BLANKING PULSES /Y1 /Y2 /Y3 & /Y4 . THESE ARE SUMMED IN FORMAT2 PAL TO GET THE BLANKING PULSE Y AND ITS COMPLEMENT /Y.

FORMAT ONE PAL --- OUTPUT BUFFER.UO 27A



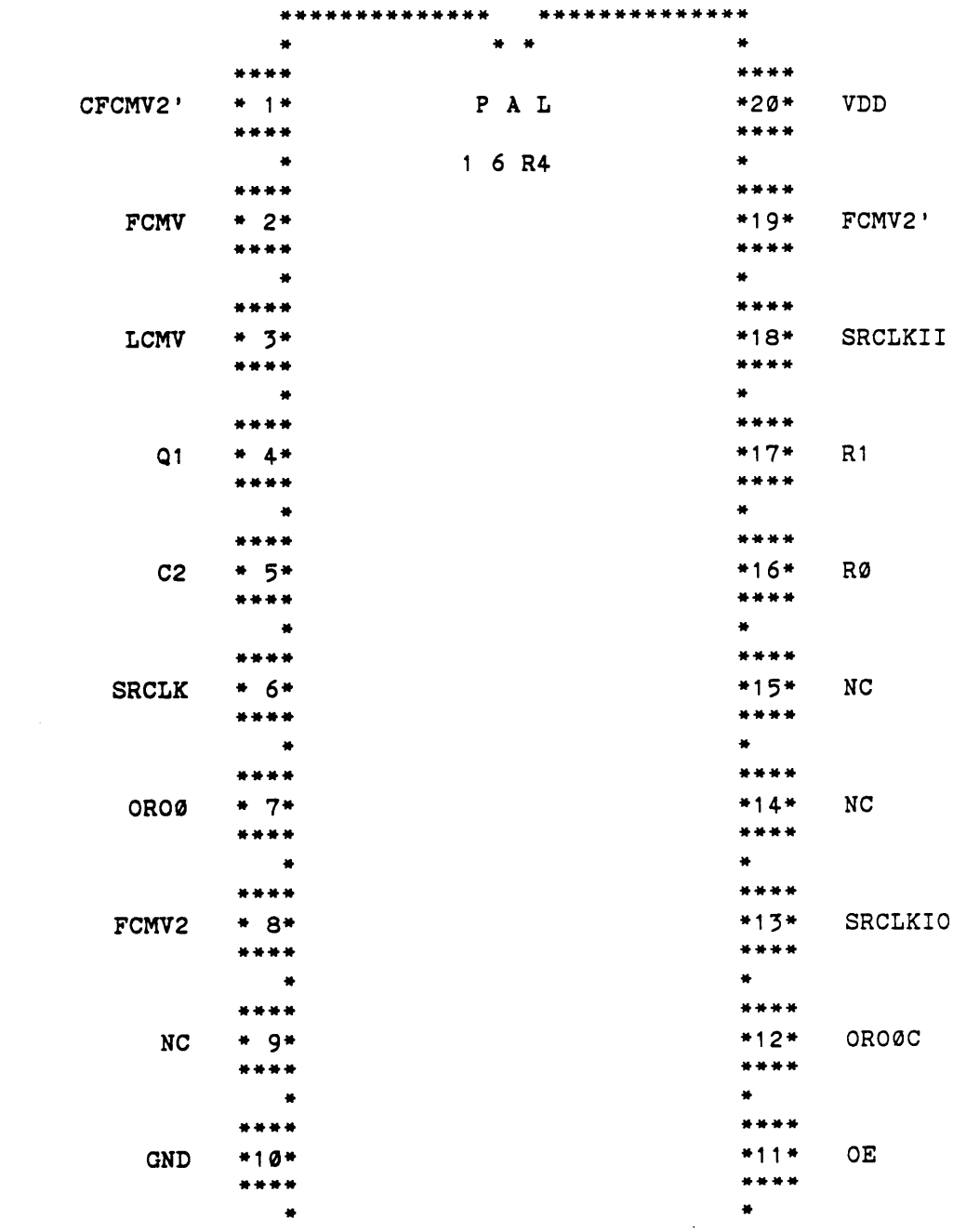
```
PAL16R4                    PAL DESIGN SPECIFICATIONS
PAT1010                    SANJAY ADKAR 05-22-86
FORMAT 2 PAL -- OUTPUT BUFFER. UO 27B
VIRGINIA TECH BLACKSBURG VA 24061

CLCMV' /Y1 /Y2  LCMV NC NC NC /Y3 /Y4 GND
OE Y Y' NC NC NC NC NC LCMV' VCC

;EQUATIONS.

/Y     = /LCMV                        +
         /Y1 * /Y2 * /Y3 * /Y4


/Y'    = Y


/LCMV'= LCMV


DESCRIPTION: THIS PAL SUMS ALL THE BLANKING PULSE EQUATIONS.
```

FORMAT 2 PAL -- OUTPUT BUFFER. UO 27B



```
PAL16R4                    PAL DESIGN SPECIFICATION
PAT1011                    SANJAY ADKAR 06-25-86
FORMAT THREE PAL --OUTPUT BUFFFER. UO 28

CFCMV2' FCMV  LCMV Q1 C2 SRCLK ORO0 FCMV2 NC GND
OE ORO0C SRCLKIO NC NC R0 R1 SRCLKII FCMV2' VDD

;EQUATIONS.


/R0            := R0


/R1            := R1 * R0                    +
                  /R1 * /R0


/FCMV2'         = FCMV2


/ORO0C          = /ORO0                      +
                  C2


/SRCLKIO        = /FCMV                      +
                  /LCMV                      +
                  /SRCLK

/SRCLKII        = /SRCLKIO                   +
                  C2 * Q1


DESCRIPTION: THIS PAL GENERATES THE I/O CLOCKS FOR THE
INTERPOLATER CIRCUIT
IT ALSO GENETATES A FRAME COUNT  WHICH IN TURN IS USED TO
GENERATE THE VARIOUS OUTPUT FORMATS.
```

FORMAT THREE PAL --OUTPUT BUFFFER. UO 28



```
PAL16R4                    PAL DESIGN SPECIFICATIONS
PAT1012                    SANJAY ADKAR 06-25-86
VDAC CONTROL PAL -- OUTPUT BUFFER UO 29
VIGINIA TECH BLACKSBURG VA 24061


CINSRCLK SRCLK SOCV Y NC NC RCOCE RCOCL RCOC GND
OE NC CR' NC LCMC LCML CLEAR' SOLV INSRCLK VCC

;EQUATIONS.

IF (VCC)  /SOLV          = RCOC

IF (VCC)  /INSRCLK       = SRCLK

IF (VCC)  /CR'           = Y                          +
                             /CLEAR'


/LCMC                    := LCMC * /RCOC              +
                            SOCV * /LCMC              +
                            SOCV * /RCOC

/LCML                    := LCML * /RCOCL             +
                            RCOC * /LCML              +
                            RCOC * /RCOCL

/CLEAR'                  := CLEAR' * /RCOCE           +
                            RCOCL  * /CLEAR'          +
                            RCOCL  * /RCOCE
```

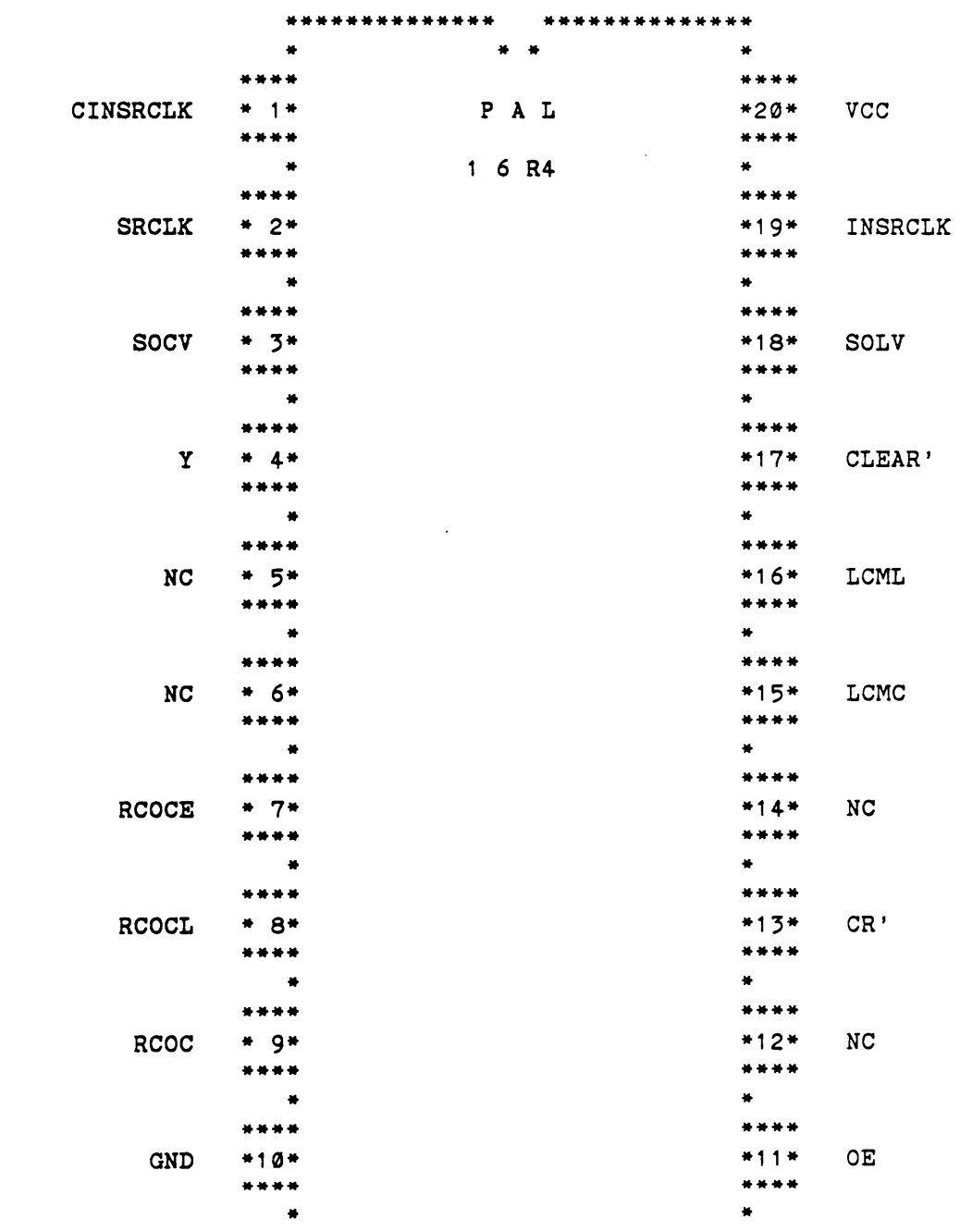
FUNCTION TABLE:

CINSRCLK SOCV LCMC RCOC SOLV LCML RCOCL
CLEAR' RCOCE Y CR' INSRCLK SRCLK

```
;C
;I                                        I
;N                        C               N
;S                    R   L   R           S   S
;R  S   L   R   S   L  C   E   C           R   R
;C  O   C   C   O   C  O   A   O       C   C   C
;L  C   M   O   L   M  C   R   C       R   L   L
;K  V   C   C   V   L  L   '   E   Y   '   K   K
-------------------------------------------------
 C  H   L   H   L   L  H   L   H   L   L   L   H
 C  L   H   H   L   L  H   L   H   H   L   H   L
 C  H   H   H   L   L  H   L   H   H   L   L   H
 C  H   L   L   H   H  H   L   H   L   L   H   L
 C  H   L   H   L   H  H   L   H   H   L   L   H
 C  H   L   H   L   L  L   H   H   L   H   H   L
 C  H   L   H   L   L  H   H   H   H   L   L   H
 C  H   L   H   L   L  H   L   L   L   L   H   L
 C  H   L   H   L   L  H   L   H   H   L   L   H
-------------------------------------------------
```

DESCRIPTION : THIS PAL GENERATES ALL THE CONTROL SIGNALS REQUIRED TO COMPOSE THE VIDEO FOR A SQUARE DISPLAY ON THE STANDARD 4:3 MONITOR.

VDAC CONTROL PAL -- OUTPUT BUFFER UO 29



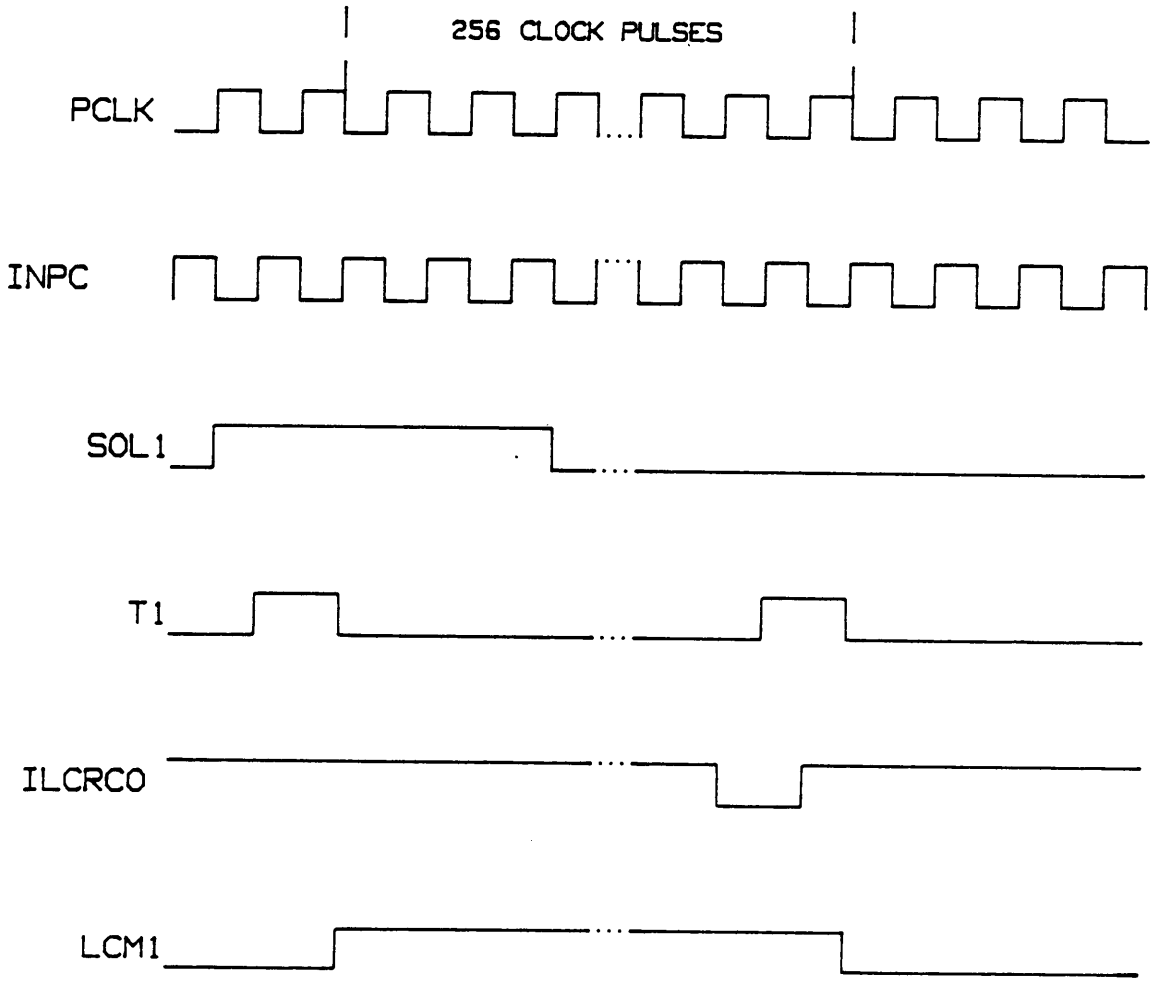
APPENDIX C. CLOCK BOARD DESIGN DOCUMENTATION.

The principles involved and the basic hardware design concepts have already been explained in Chapter 4 of this thesis. The following sections contain all the design documentation for the clock board.

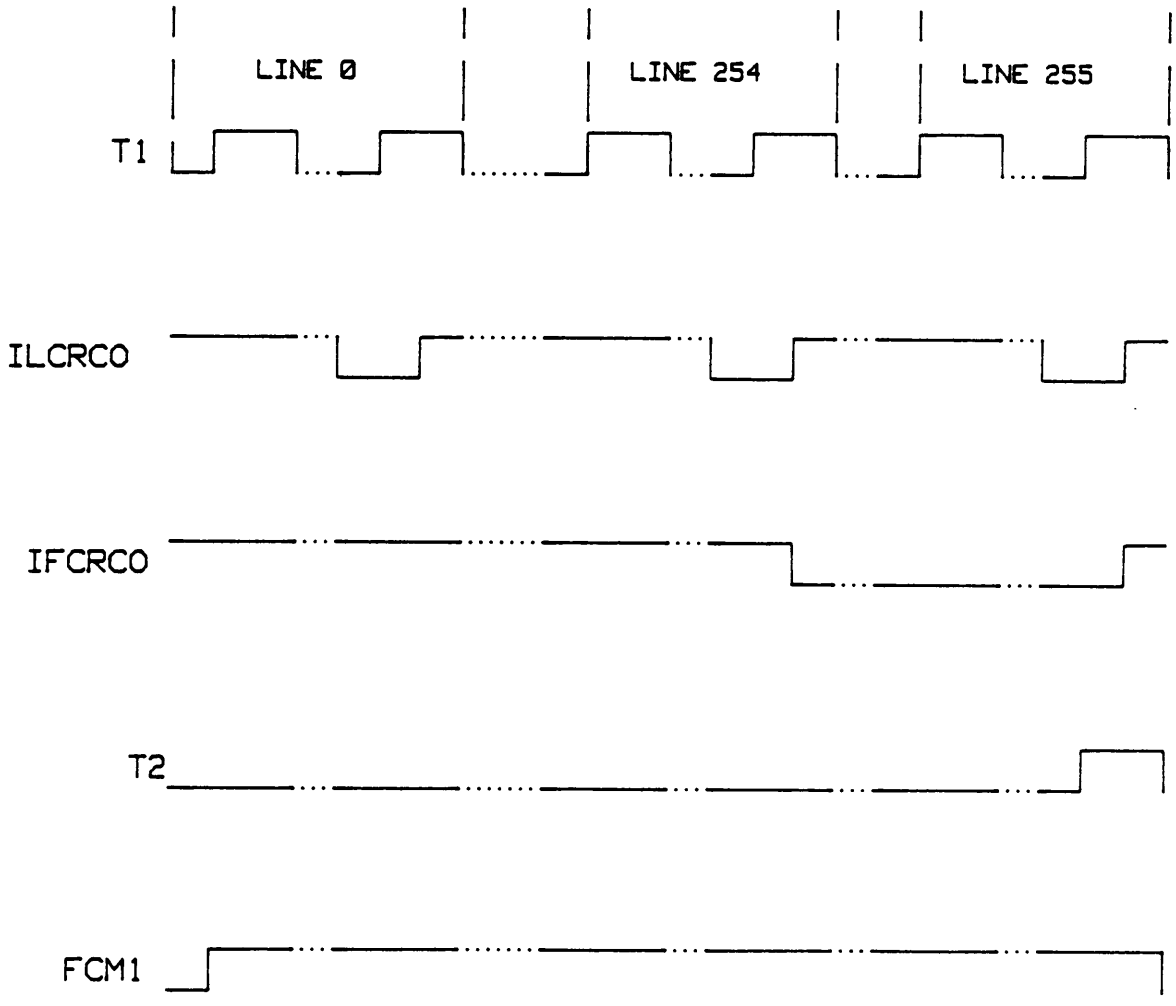
C.1 CLOCK BOARD TIMING DIAGRAMS.

The following set of drawings are the timing diagrams of the clock board. They should be read in reference to the circuit diagrams and the PAL programs in the next two sections. Chapter 3 contains all the necessary explanation of the various signals. The signal generation is actually performed in the PALs from the available incoming signals.

CLOCK BOARD: LINE MARKER SIGNAL.

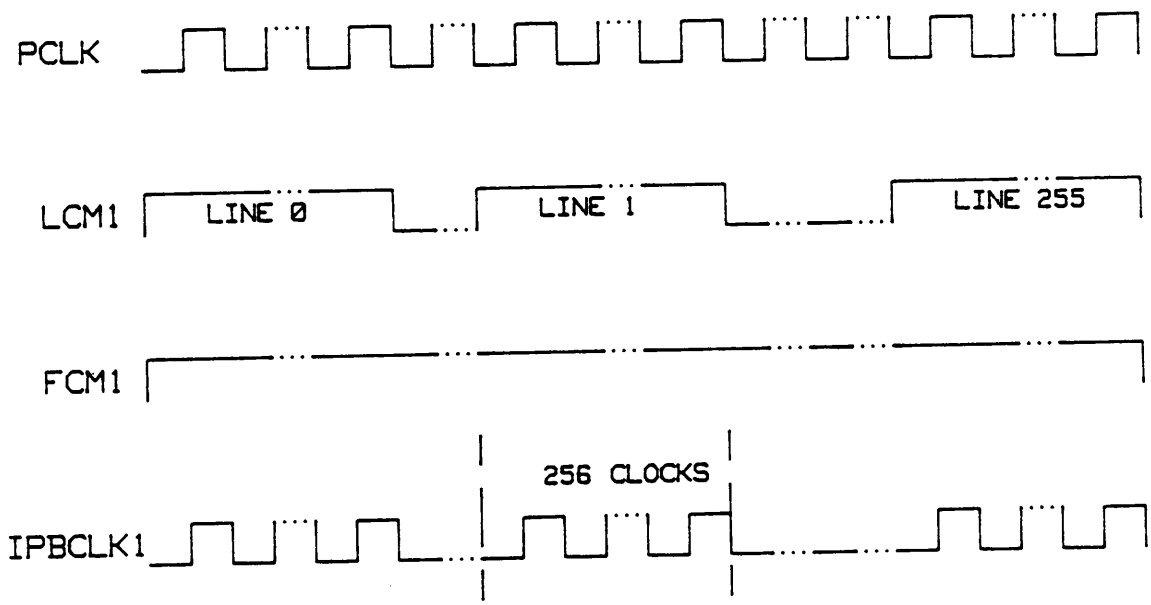


CLOCK BOARD: FRAME MARKER SIGNAL.

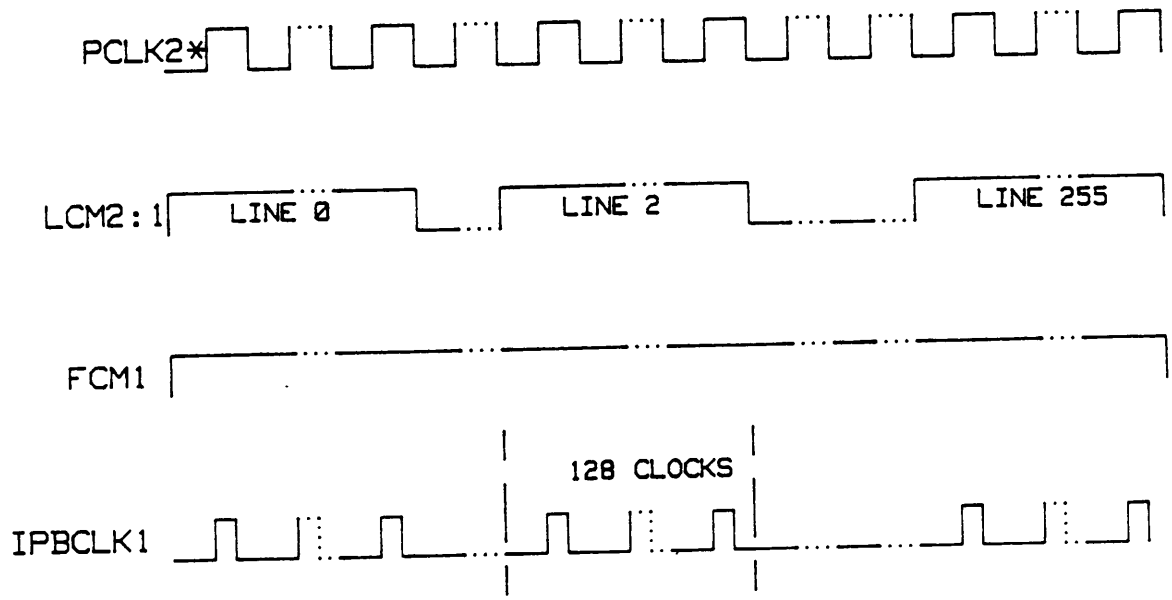


CLOCK BOARD

IPB CLOCK (256*256)

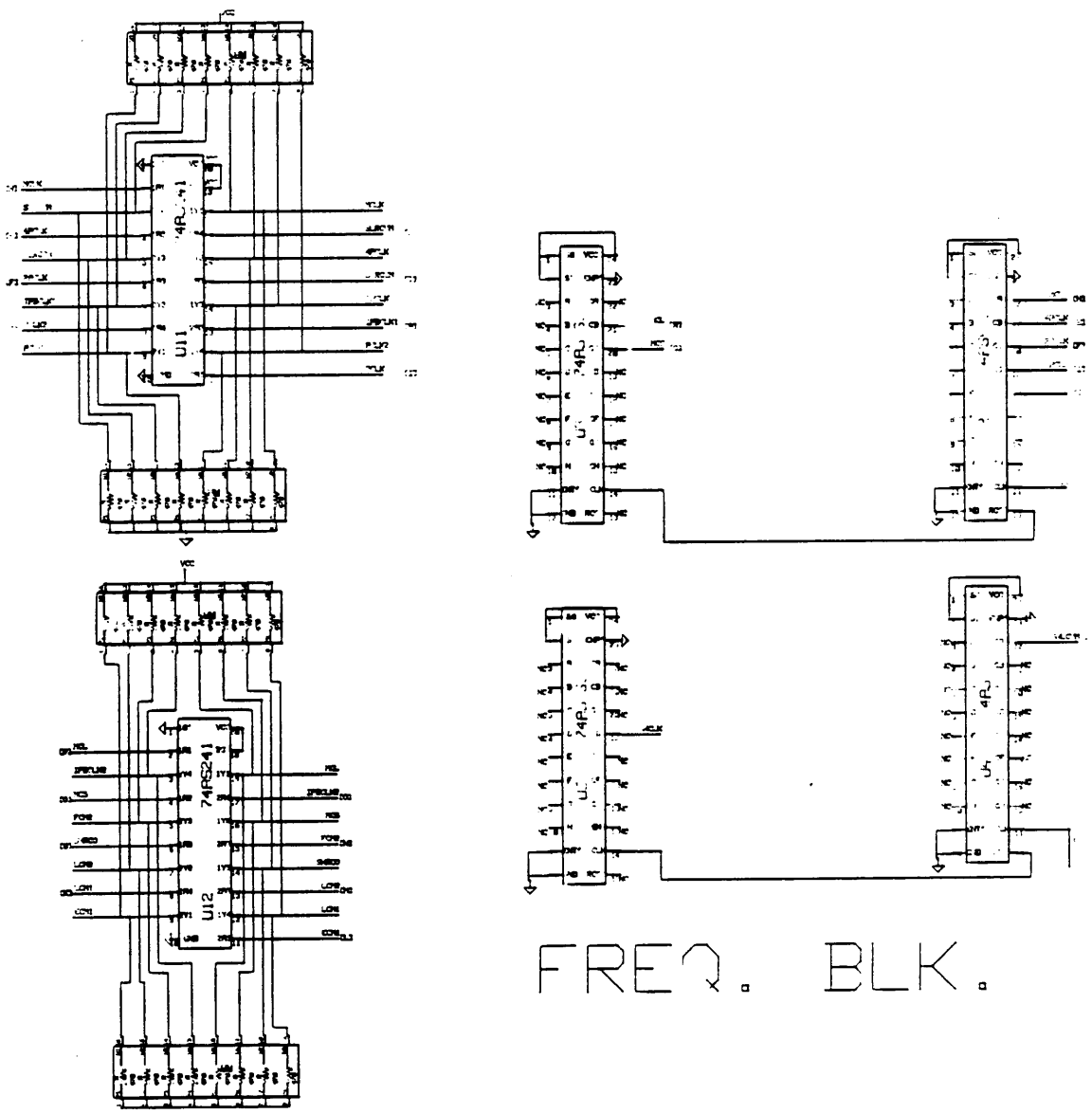


IPB CLOCK (128*128).



C.2 CLOCK BOARD CIRCUIT DIAGRAMS.

The following are the circuit diagrams for the clock board. The details of all the clock board blocks described in Chapter 3 are shown here.


U11
74RS241
U12
74RS241
FREQ. BLK.

C.3 CLOCK BOARD PAL PROGRAMS

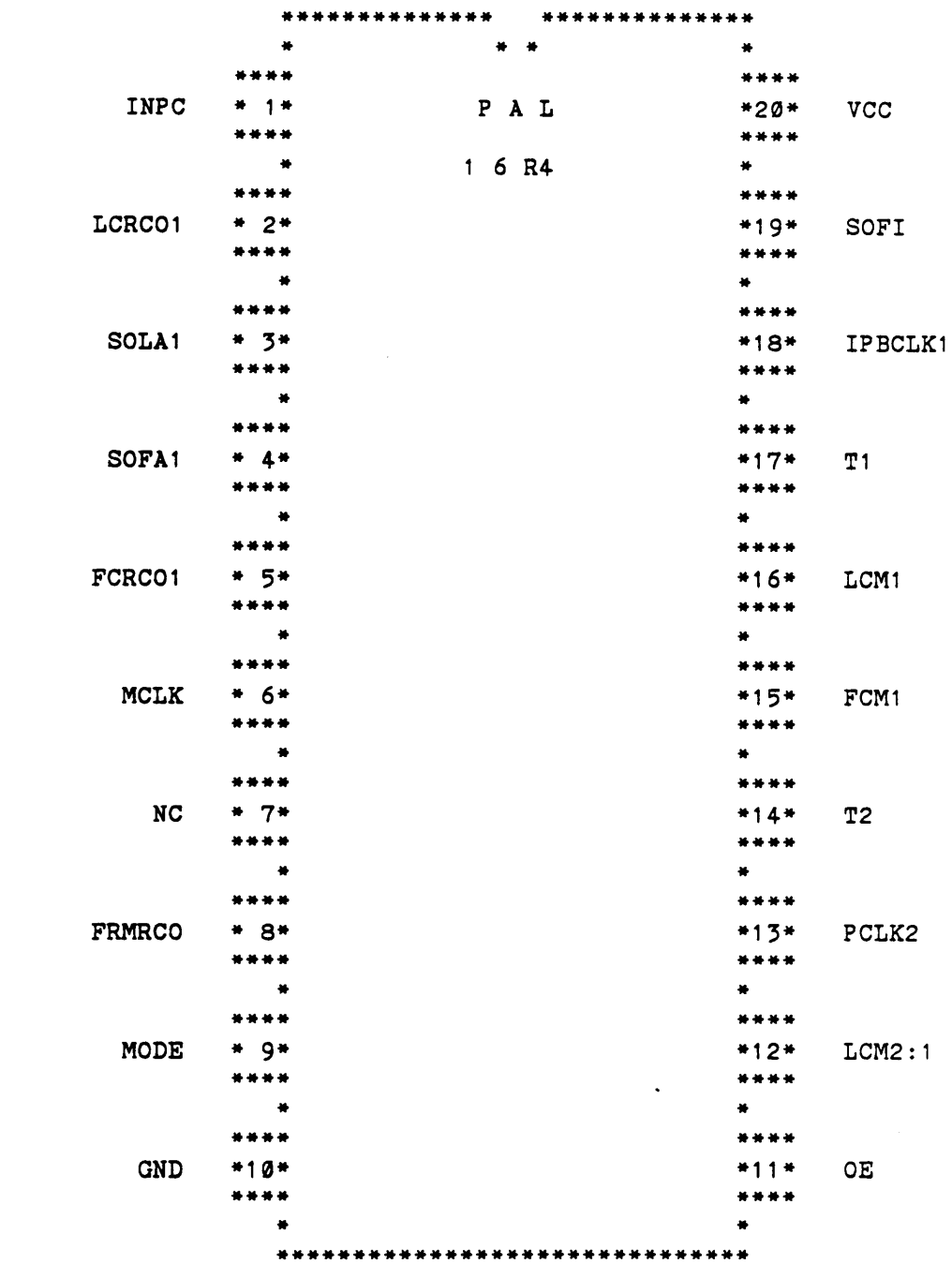
This section contains the PAL programs in the PALASM language for all the PALS used in the clock board. The program generated simulation results. The fuse plots and the listing of the JEDEC files used by the PAL programmer to program the PAL are not included in this section, however these are available on floppy disk.

```
PAL16R4                                  PAL DESIGN SPECIFICATIONS
PAT001                                   SANJAY ADKAR 3-31-1986
CLOCK BOARD PAL (SYSTEM ONE CONTROL PAL)
VIRGINA TECH BLACKSBURG VA 24061
INPC LCRCO1 SOLA1 SOFA1 FCRCO1 MCLK NC FRMRCO MODE GND
OE LCM2:1 PCLK2 T2 FCM1 LCM1 T1 IPBCLK1 SOFI VCC
;
;
;THE FOLLOWING DEFINE THE EQUATIONS USED IN THE PAL
;
/T1 := T1       +
       /SOLA1 *LCRCO1 +
       /SOLA1 * /LCM1 +
       LCM1 * LCRCO1

/T2 := LCRCO1         +
       FCRCO1         +
       T2


/LCM1 := /T1 * /LCM1     +
         T1 * LCM1


/FCM1 := /SOFA1 */FCM1      +
         /SOFA1 *T1 *T2    +
         /T1 * /FCM1        +
         T1 * T2 * FCM1



IF (VCC)   /IPBCLK1 = /MCLK          +
                      /LCM1          +
                      /FCM1          +
                      /MODE * PCLK2 +
                      /MODE *LCM2:1


IF (VCC)   /SOFI = FRMRCO
```

```
FUNCTION TABLE
OE INPC LCRCO1 SOLA1 SOFA1 FCRCO1 MCLK FRMRCO MODE  LCM2:1 PCLK2
FCM1 LCM1 T1 IPBCLK1 SOFI T2

;                                                  I
;     L      F    F    L                          P
;     C S S C    R    C P                        B
;  I R O O R M M M M C          F L    C S
;  N C L F C C R O 2 L          C C    L O
;O P O A A O L C D : K          M M T  K F T
;E C 1 1 1 1 K O E 1 2          1 1 1  1 I 2
-------------------------------------------------------------------
;FUNCTION TABLE FOR EVERYTHING
 L C H L L H L H H X X          L L L  L L L
 L C H L H H H L H X X          L L L  L H L
 L C H H H H H H H X X          L L H  L L L
 L C H H H H H H H X X          H H L  H L L
 L C H X X H L H H X X          H H L  L L L
 L X H X X H H H L H H          H H L  L L L
 L X H X X H H H L L H          H H L  L L L
 L X H X X H H H L L L          H H L  H L L
 L C L L L H H H H X X          H H H  H L L
 L C H L L H H H H X X          H L L  L L L
 L C L L L L H H H X X          H L L  L L H
-------------------------------------------------------------------
```

DESCRIPTION
THIS DESIGN IS USED TO CONTROL ALL THE COUNTER AND TO GENERATE
THE IPB CLOCK

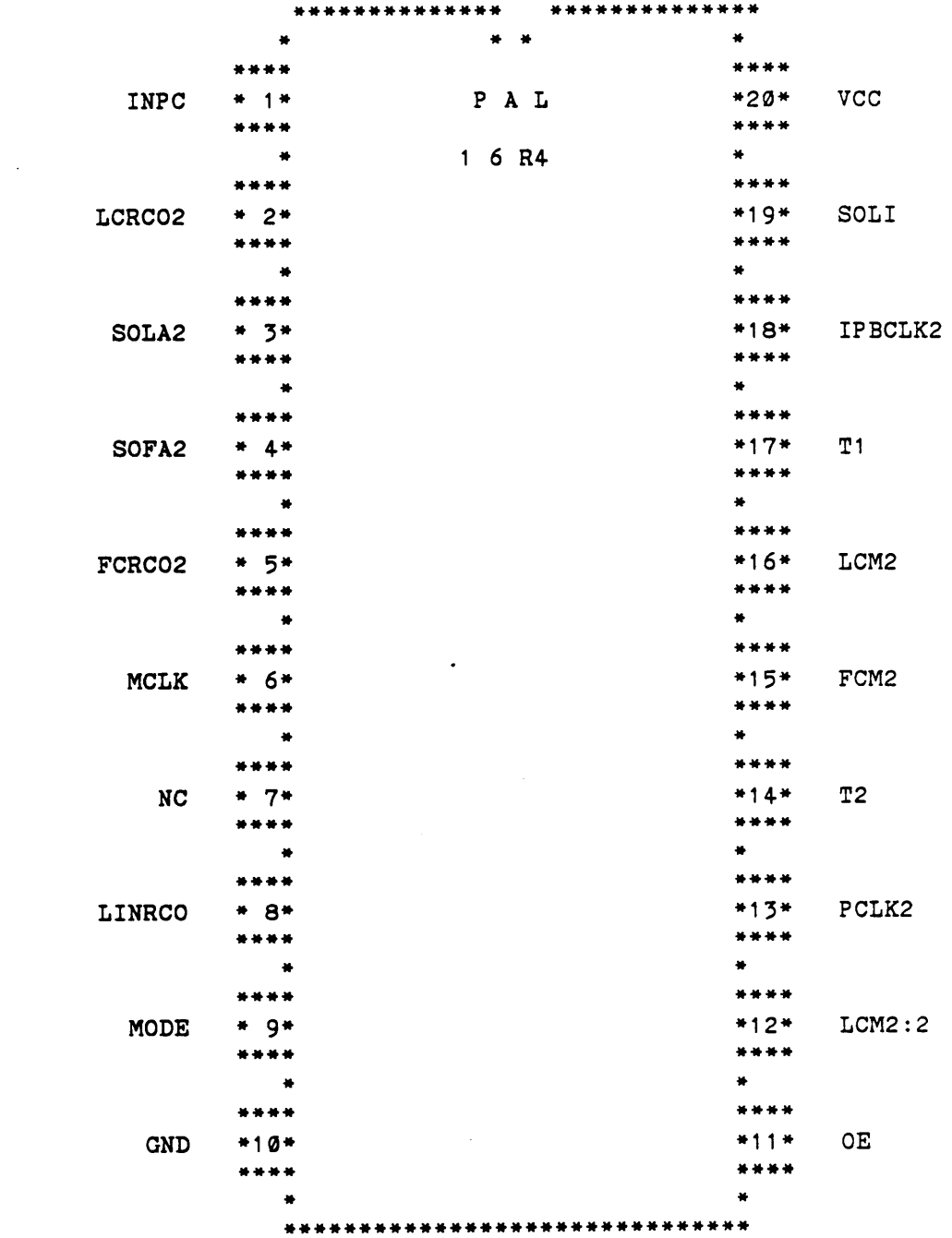
CLOCK BOARD PAL (SYSTEM ONE CONTROL PAL)



```
PAL16R4                                  PAL DESIGN SPECIFICATIONS
PAT002                                   SANJAY ADKAR 3-31-1986
CLOCK BOARD PAL (SYSTEM TWO CONTROL PAL)
VIRGINA TECH BLACKSBURG VA 24061
INPC LCRCO2 SOLA2 SOFA2 FCRCO2 MCLK NC LINRCO MODE GND
OE LCM2:2 PCLK2 T2 FCM2 LCM2 T1 IPBCLK2 SOLI VCC
;
;
;THE FOLLOWING DEFINE THE EQUATIONS USED IN THE PAL
;
/T1  := T1       +
        /SOLA2 * LCRCO2 +
        /SOLA2 * /LCM2  +
        LCM2 * LCRCO2

/T2  := T2               +
        LCRCO2           +
        FCRCO2


/LCM2  := /T1 * /LCM2     +
          LCM2 *  T1


/FCM2  := /SOFA2 */FCM2      +
          /SOFA2 *T1 *T2    +
          /T1 * /FCM2        +
          FCM2 *  T1 * T2



IF (VCC)   /IPBCLK2 = /MCLK          +
                      /LCM2          +
                      /FCM2          +
                      /MODE * PCLK2 +
                      /MODE *LCM2:2

IF (VCC)   /SOLI = LINRCO
```

```
FUNCTION TABLE
OE INPC LCRCO2 SOLA2 SOFA2 FCRCO2 MCLK LINRCO MODE  LCM2:2 PCLK2
FCM2 LCM2 T1 IPBCLK2 SOLI T2

;                                                  I
;     L       F   L   L                            P
;     C S S C     I   C P                          B
;  I R O O R M N M M C          F L   C S
;  N C L F C C R O 2 L          C C   L O
;O P O A A O L C D : K          M M T K L   T
;E C 2 2 2 2 K O E 2 2          2 2 1 2 I   2
-------------------------------------------------------------

 L C H L L H L H H X X          L L L L L   L
 L C H L H H H L H X X          L L L L H   L
 L C H H H H H H H X X          L L H L L   L
 L C H H H H H H H X X          H H L H L   L
 L C H X X H L H H X X          H H L L L   L
 L X H X X H H H L H H          H H L L L   L
 L X H X X H H H L L H          H H L L L   L
 L X H X X H H H L L L          H H L H L   L
 L C L L L L H H H X X          H H H H L   H
 L C H L L L H H H X X          L L L L L   L
```

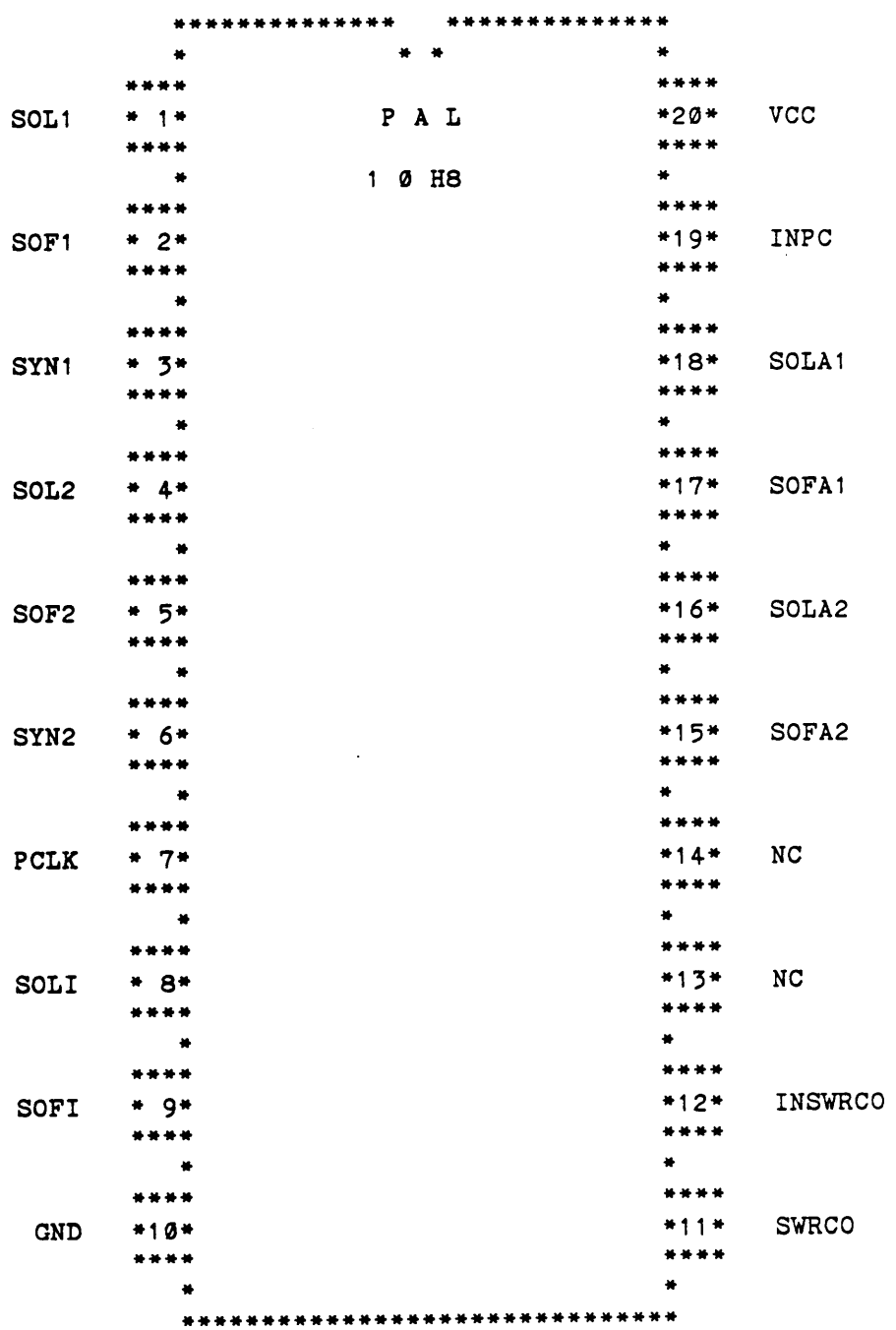
CLOCK BOARD PAL (SYSTEM TWO CONTROL PAL)



```
PAL10H8                          PAL DESIGN SPECIFICATION
PAT001                           SANJAY ADKAR
MULTIPLEXER/INVERTER PAL
VIRGINIA TECH BLACKSBURG
SOL1 SOF1 SYN1 SOL2 SOF2 SYN2 PCLK SOLI SOFI GND
SWRCO INSWRCO NC NC SOFA2 SOLA2 SOFA1 SOLA1 INPC VCC


;SOLA1 SOFA1 SOLA2 SOFA2 ARE THE OUTPUT SIGNALS
;SOL1 SOF1 SOL2 SOF2 SOLI SOFI ARE THE INPUT SIGNALS
;THESE SIGNALS SHOULD BE ROUTED ACCORDING TO THE FUNCTION TABLE
;

;EQUATIONS

SOLA1 = SOL1*SYN1 + SOLI*/SYN1

SOFA1 = SOF1*SYN1 + SOFI*/SYN1

SOLA2 = SOL2*SYN2 + SOLI*/SYN2

SOFA2 = SOF2*SYN2 + SOFI*/SYN2

INPC  =/PCLK

INSWRCO = /SWRCO
```

```
FUNCTION TABLE

PCLK SWRCO SOL1 SOF1 SOL2 SOF2 SOLI SOFI SYN1 SYN2
SOLA1 SOFA1 SOLA2 SOFA2 INPC  INSWRCO


-------------------------------------------------------------------------
------

;INVERTER  TABLE
H H X X X X X X X  X X X X L L
L H X X X X X X X  X X X X H L
H L X X X X X X X  X X X X L H
L L X X X X X X X  X X X X H H
;
;MULTIPLEXER TABLE
;
X X L H L H H L L L  H L H L X X
X X L H L H H L H L  L H H L X X
X X L H L H H L L H  H L L H X X
X X L H L H H L H H  L H L H X X
-------------------------------------------------------------------------
------

DESCRIPTION: THIS DESCRIBES THE PAL USED FOR MULTIPLEXING IN THE CLOCK.
```

MULTIPLEXER/INVERTER PAL



The vita has been removed from the scanned document