

JULY 1982

SY6545
CRTC Design and
Applications Manual
Microprocessor Products

Synertek

CONTENTS

VIDEO RAM ACCESS

- 1.0 GENERAL CONSIDERATIONS
- 2.0 COMMON CHARACTERISTICS
- 3.0 MPU PRIORITY
- 4.0 ϕ_1/ϕ_2 INTERLEAVE
- 5.0 TRANSPARENT ADDRESSING — BLANKING UPDATES
- 6.0 TRANSPARENT ADDRESSING — ϕ_1/ϕ_2 INTERLEAVE
- 7.0 PROGRAM EXAMPLES TO LOAD A COMPLETE SCREEN OF DISPLAYED TEXT
- 8.0 SUMMARY

MISCELLANEOUS APPLICATIONS

- 9.0 SPLIT-SCREEN DISPLAYS
- 10.0 OPERATING AT VERY FAST CHARACTER CLOCK RATES
- 11.0 DYNAMICALLY CHANGING SCREEN DISPLAY FORMATS
- 12.0 USING LPEN AS KEYBOARD SCAN STROBE

The SY6545 CRT Controller is capable of operating in a wide variety of modes. In particular, several different schemes for sharing the Video Display Memory with the system MPU may be accommodated. Each of these schemes has its own relative advantages and drawbacks. This note presents some detailed system implementations (including schematics and listings) for each memory sharing technique.

1.0 General Considerations

Common to any CRT control circuit is the requirement that the Video Display RAM (used to store the currently displayed text) be accessible by both the system MPU (to change the display) and the CRT display logic (to keep the screen active). This is illustrated by the diagram in Figure 1, which shows the dual access nature of the Video Display RAM.

The SY6545 CRT Controller permits several methods of memory sharing to be implemented:

- MPU Priority — With this technique, any MPU request

immediately has access to the Video Display RAM, overriding any current access being made by the CRT logic.

- Vertical Blanking — All MPU accesses are made during the vertical blanking time, when the screen is inactive.
- ϕ_1/ϕ_2 Interleave — This method “time-shares” the memory. Each MPU clock cycle is split so that both the MPU and the CRT logic have immediate access.
- Transparent Addressing - Blanking — This technique permits MPU access to the Video Display RAM to occur only during non-display times (both vertical and horizontal).
- Transparent Addressing — ϕ_1/ϕ_2 Interleave — In this method, the MPU accesses the RAM through the CRT Controller with ϕ_1/ϕ_2 “time-sharing”.

Each of these methods requires different circuit implementations and variations in how the MPU programs are constructed.

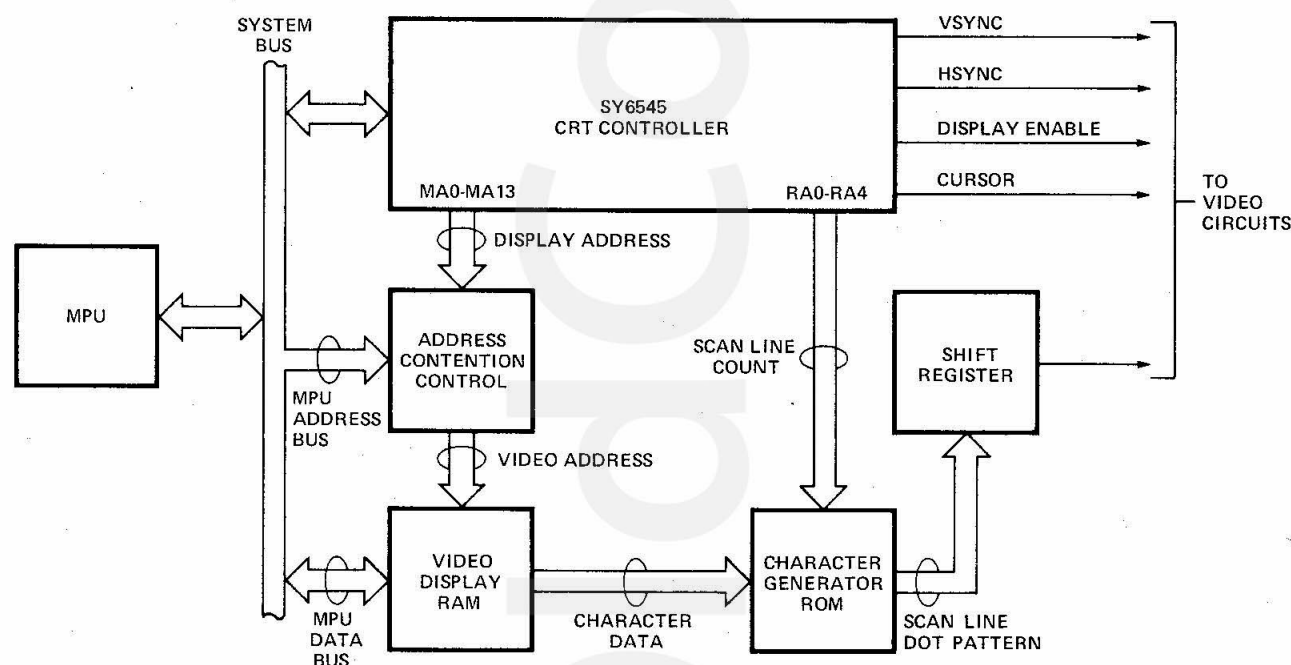


Figure 1.

2.0 Common Characteristics

Some common assumptions are made for these implementations:

- The display format is chosen to be 80 columns x 24 rows of characters, except for the last mode which is 40 x 24.
- Each character block is an 8 x 8 dot size with the actual character size 5 x 7.
- The character generator (to generate dot patterns for each character scan line is a 2k x 8 ROM, in this case a Synertek SY2316B.
- A 4k-byte Video Display RAM is utilized to permit slightly more than two complete pages of text data to be stored.

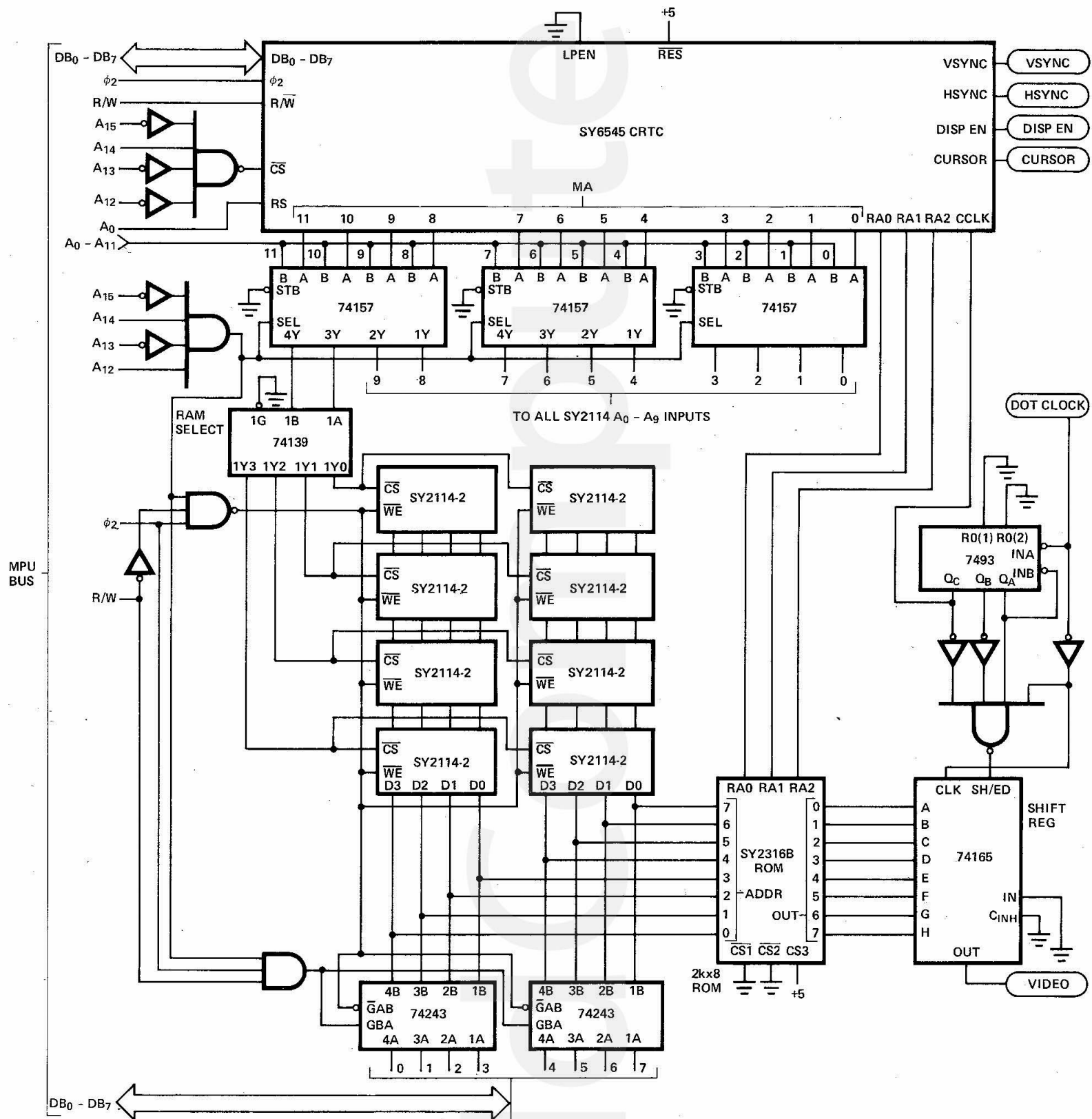


Figure 2. MPU Priority System Schematic

Table 1.

SELECT	ADDRESS																PROGRAM ADDRESS
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CRTC ADDR/STATUS	0	1	0	0	—	—	—	—	—	—	—	—	—	—	—	0	4000
CRTC DATA	0	1	0	0	—	—	—	—	—	—	—	—	—	—	—	1	4001
RAM SELECT	0	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X	5XXX

— DENOTES "DON'T CARE"
X DENOTES ADDRESS BIT

3.0 MPU Priority

Figure 2 is a detailed schematic of the CRT control logic for this scheme and Figure 3 shows the associated timings. Several points are noteworthy:

- In Figure 2, the 74157 multiplex switches are used to select either the CRTC or the MPU as the addressing inputs to the Video Display RAM. Whenever the MPU address bus selects a Video Display RAM address, the multiplex switches immediately gate the MPU address to the RAM, de-selecting the CRTC addresses. Table 1 summarizes the address decoding for Figure 2.
- The 74243 bus transceivers are selected in conjunction with the multiplex switches to connect the MPU data bus to the Video Display RAM whenever it is selected by the MPU. The direction of the data through the transceivers is determined by the level of the R/W line on the MPU. Further, they are active only during ϕ_2 time of the MPU cycle, since data transfers in the MPU are only done at that time. The controls for the transceivers GBA and $\bar{G}AB$ are determined by Table 2.

Table 2.

INPUTS			OUTPUTS		FUNCTION
ϕ_2	R/W	RAM SEL	GBA	$\bar{G}AB$	
—	—	L	L	H	RAM ISOLATED FROM MPU
L	—	—	L	H	RAM ISOLATED FROM MPU
H	H	H	H	H	MPU READ FROM RAM
H	L	H	L	L	MPU WRITE INTO RAM

- In Figure 3, note that the dot time is 80 nsec and the character time is 640 nsec (8 dots/character). The address propagation delays (6545) and the RAM (2114) and ROM (2316B) access times determine when the dot pattern data is available to be loaded into the shift register. The load strobe may occur any time when the dot pattern is valid. In this example, the load strobe is generated early in the character clock cycle, approximately coincident with the leading edge of the time when a delayed Cursor signal would be available. In this way, video data from the shift register may be modified (high-

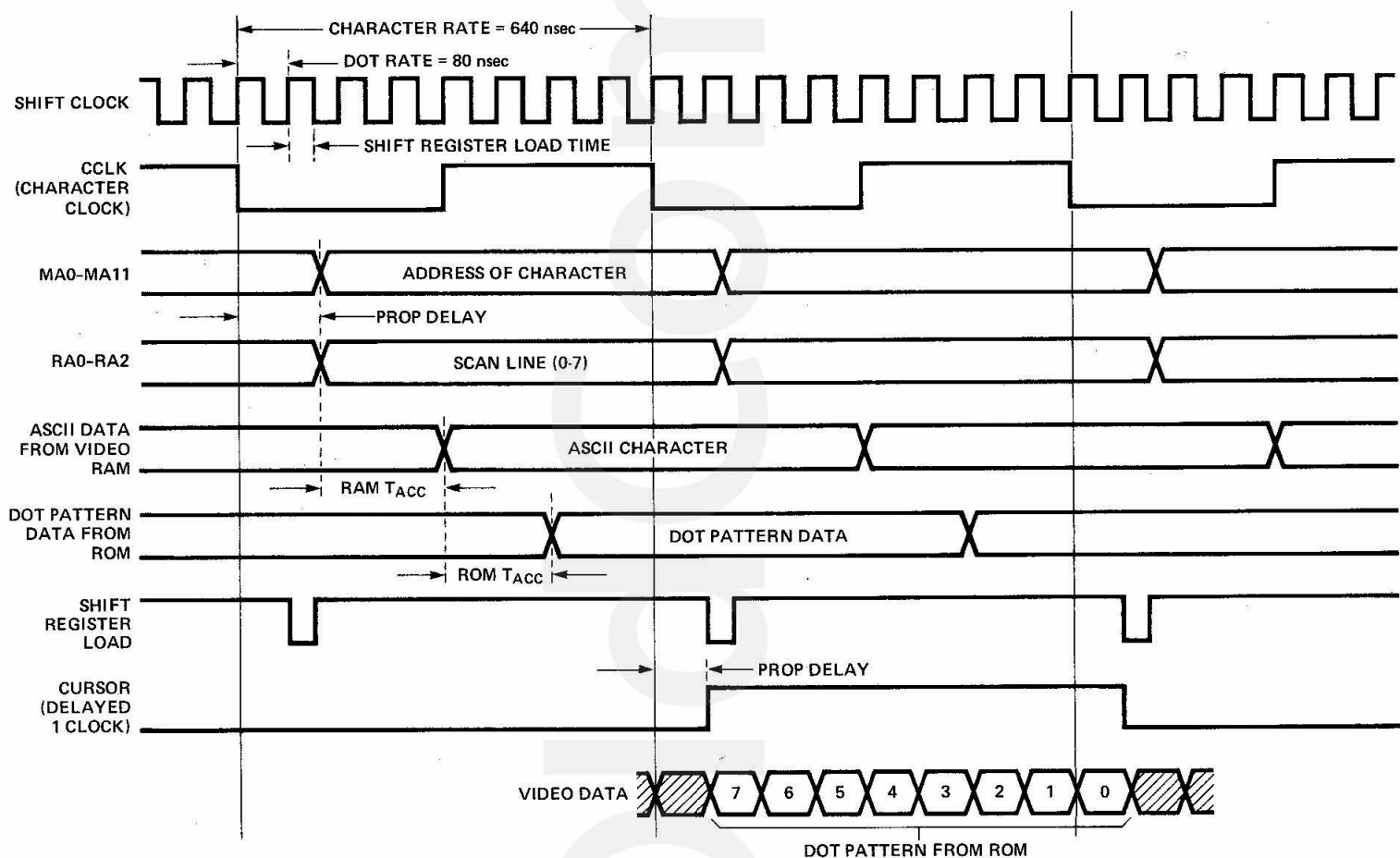


Figure 3. MPU Priority Timing Diagram

lighted, reversed, etc) by the Cursor signal, directly. Note that, since the load strobe for a character occurs in the CCLK cycle following when the address was generated, it is necessary to program the Cursor to be delayed one cycle (a programmed function in the SY6545). For the same reason, Display Enable must be delayed one cycle, as well.

- The effect of MPU updates to the Video Display RAM may be understood with Figures 2 and 3. The only way any MPU operation can disturb the video appearance is if the MPU write or read occurs simultaneously with the shift register load signal. If this happens the shift register will be loaded with wrong data and some spurious flashing will be visible on the screen. Updates, during blanking times have no visible effects.

3.1 Vertical Blanking

Utilization of the vertical blanking time of raster-scan displays overcomes disadvantages of the previous method, but introduces its own drawback. To accommodate this technique, a facility must exist for the MPU to identify time intervals when the display is in its vertical blanking (retrace plus non-display margins) time. This can be of the form of a status bit (as in the SY6545) or an externally-generated MPU interrupt. In either case, the basic premise is for the MPU to perform all memory accesses during this vertical blanking time when the display is inactive and no flashing can occur. The address selection employs the simple MPU priority scheme described previously. As such, no special high-speed memory is required.

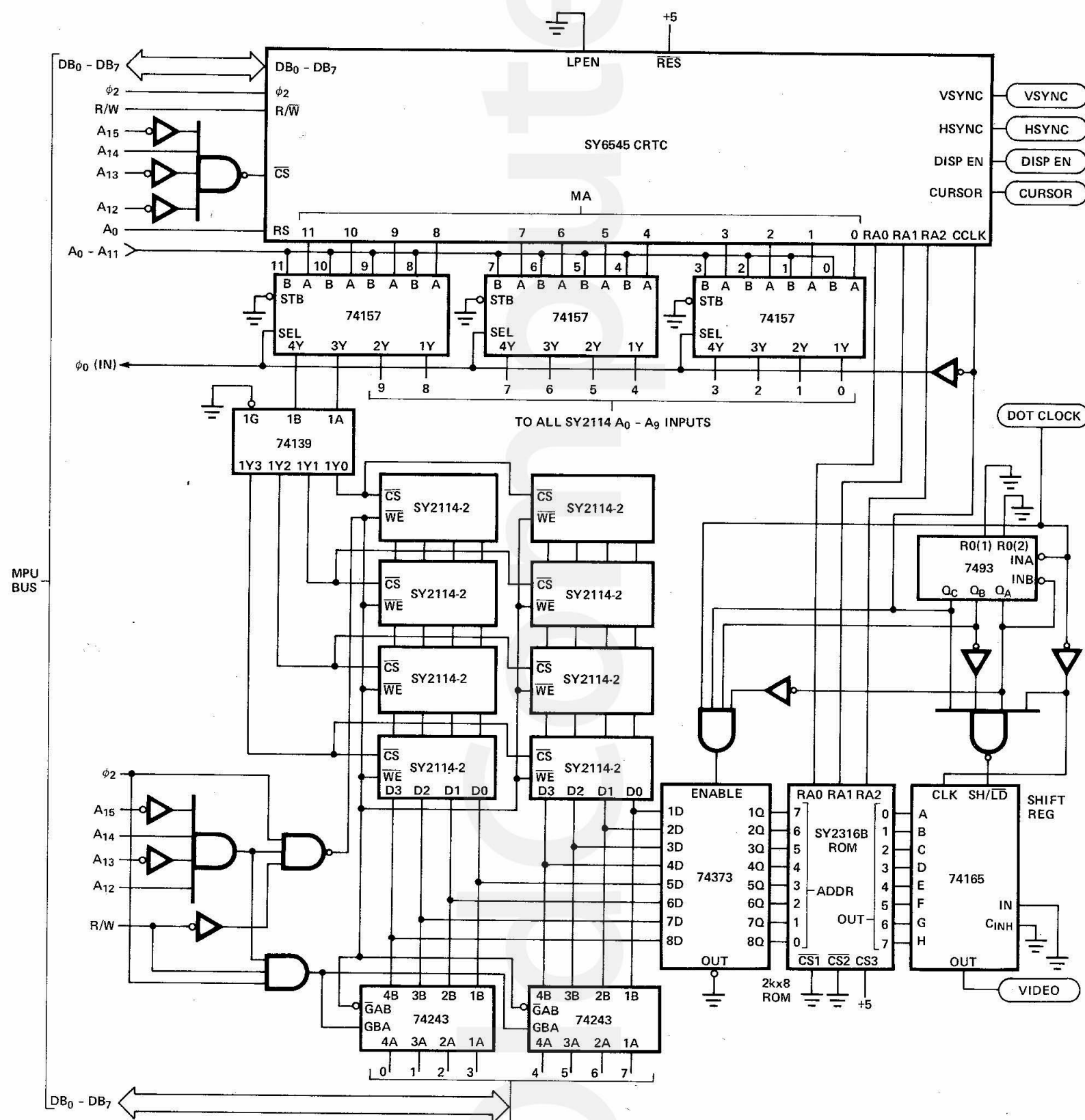


Figure 4. ϕ_1/ϕ_2 Interleave Schematic

Table 3.

SELECT	ADDRESS												PROGRAM ADDRESS				
	15	14	13	12	11	10	9	8	7	6	5	4		3	2	1	0
CRTC ADDR/STATUS	0	1	0	0	—	—	—	—	—	—	—	—	—	—	—	0	4000
CRTC DATA	0	1	0	0	—	—	—	—	—	—	—	—	—	—	—	1	4001
RAM SELECT	0	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X	5XXX

— DENOTES "DON'T CARE"
X DENOTES ADDRESS BIT

The only drawback is fairly significant. Since MPU access may only be done during vertical blanking times and since these interrupts only occur once for each frame (about 2-3 msec every 17 msec), significant time overhead may be encountered if the MPU is required to do a large number of accesses.

4.0 ϕ_1/ϕ_2 Interleave

Figure 4 is a detailed schematic and Figure 5 illustrates the timing for this method.

- In Figure 4, note that the 74157 multiplex switches are controlled by the MPU clock ϕ_0 (IN). Further, ϕ_0 (IN) is nothing more than the CCLK, so that the system MPU and the CRTC are exactly synchronized. In this way, the address inputs to the Video Display RAM repetitively switch between the MPU address lines and the CRTC address outputs. Thus, MPU operations may be done

directly, since the MPU address is always on the Video Display RAM during ϕ_2 time (time when ϕ_0 is high). As a result, MPU program operations on the RAM are identical for this mode as for the MPU Priority scheme; only the hardware is different. The address decoding is repeated in Table 3 for clarity.

- The 74243 bus transceivers are decoded identically, as well:

Table 4.

INPUTS			OUTPUTS		FUNCTION
ϕ_2	R/W	RAM SEL	GBA	$\bar{G}AB$	
—	—	L	L	H	RAM ISOLATED FROM MPU
L	—	—	L	H	RAM ISOLATED FROM MPU
H	H	H	H	H	MPU READ FROM RAM
H	L	H	L	L	MPU WRITE INTO RAM

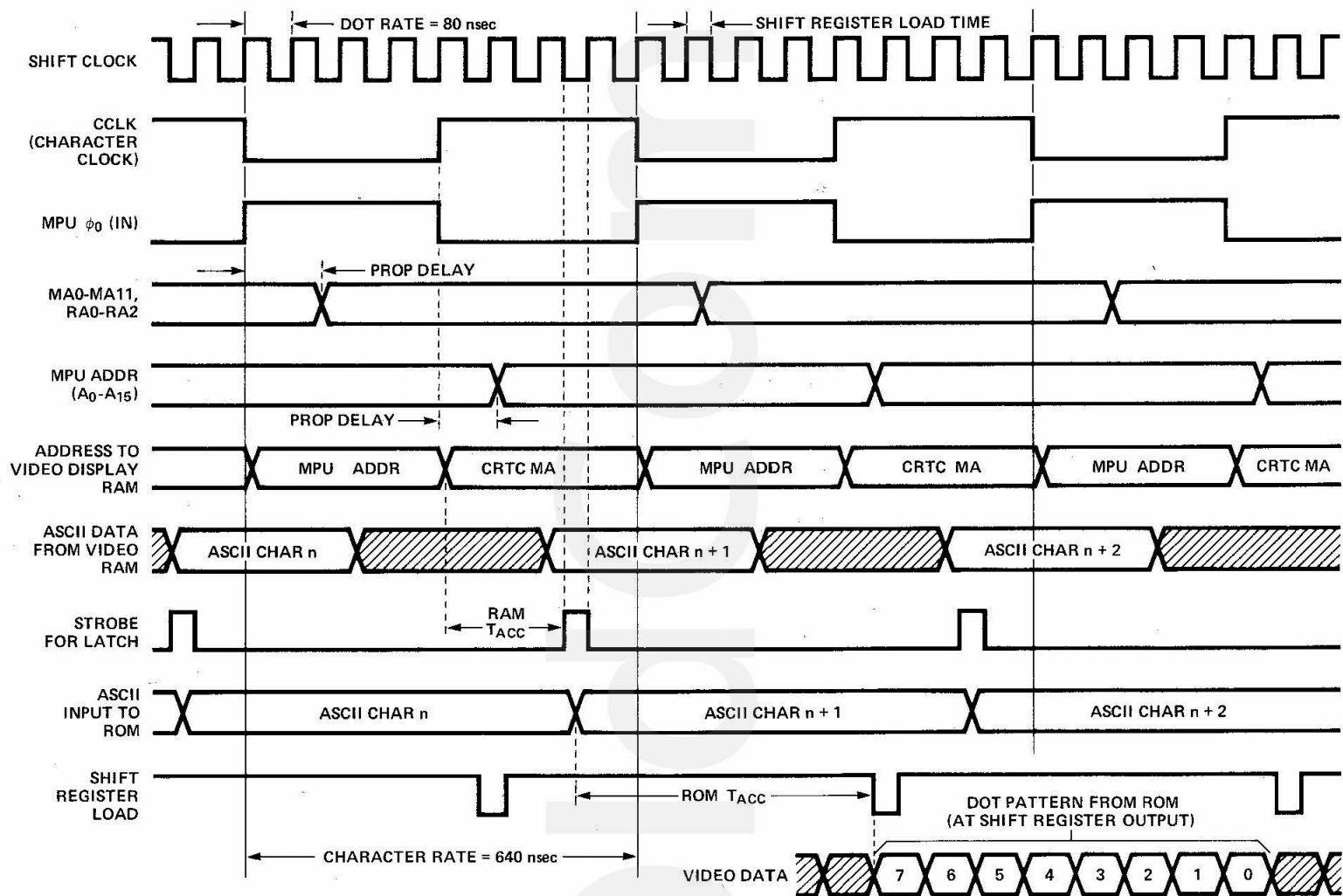


Figure 5. ϕ_1/ϕ_2 Interleave Timing Diagram

- In Figure 5, the same dot and character rates are used but the MPU clock rate is no longer independent. Instead, the MPU ϕ_0 (IN) is controlled directly from the character clock.
- Note that the CRTC propagation delays (MA and RA outputs) and the MPU propagation delays ($A_0 - A_{15}$) both occur during times when they are not selected by the multiplex switches. Thus, the RAM address inputs are clean and are only affected by the switching time of the 74157 units (typically 10-15 nsec).
- An additional 8-bit latch (the 74373 in Figure 4) is needed to hold the ASCII character from the Video Display RAM until the ROM output is loaded into the Shift Register. This is required because the cumulative access times are too slow to respond to the rapidly changing addresses. Alternative to the latch, faster memories (RAM and ROM) could be used, but this is strictly an economic issue.
- Because MPU operations take place during time intervals where the CRTC is not utilizing the memory, updates cause no visible screen effects.

5.0 Transparent Addressing — Blanking Updates

5.1 Theory

Using the retrace intervals for reads and/or updates simplifies the timing and control circuits required but may impact MPU efficiency slightly, since updates do not occur immediately and require some additional program steps to check status bits.

Another requirement is the use of the Update Strobe from the CRTC to synchronize read/updates with the update address on the RAM address lines.

The timings in Figure 6 indicate the relationship of the Update Strobe, the Update Address, the Display-Enable, and the Character Clock. The update address is gated on the RAM address lines one clock after the display is blanked and stays for three character clock cycles. The Update Strobe is one character clock in duration and is centered within the update address. Its function is to signal the Data Bus Latch/Buffers to perform the required read/update operation.

Several noteworthy points should be understood before advancing to more detail on this update scheme.

- The Display Enable signal is high during CRT display times and low during horizontal or vertical blanking periods. Typically, the low portion of this signal might be about 20 μ sec (horizontal blank) and the high portion, about 40 μ sec for each line. The vertical blank might be about 5 msec every frame time (17 msec/frame total).
- The Ready bit in the Status Register goes low (not-ready) whenever the dummy register is first selected and whenever the RAM Latches are selected with the read or write commands. The Ready bit goes high whenever an automatic read (AR) of the RAM occurs.
- The RAM Latches are used:
 - a. To store the 8 data bits to be written when an MPU-write occurs.
 - b. To store the 8 bits on an automatic read (AR) in anticipation of an MPU-read operation.
- The Update Address Counter (UAC) in the 6545 is incremented whenever an Update Strobe occurs in response to a RAM write and whenever a RAM read occurs.

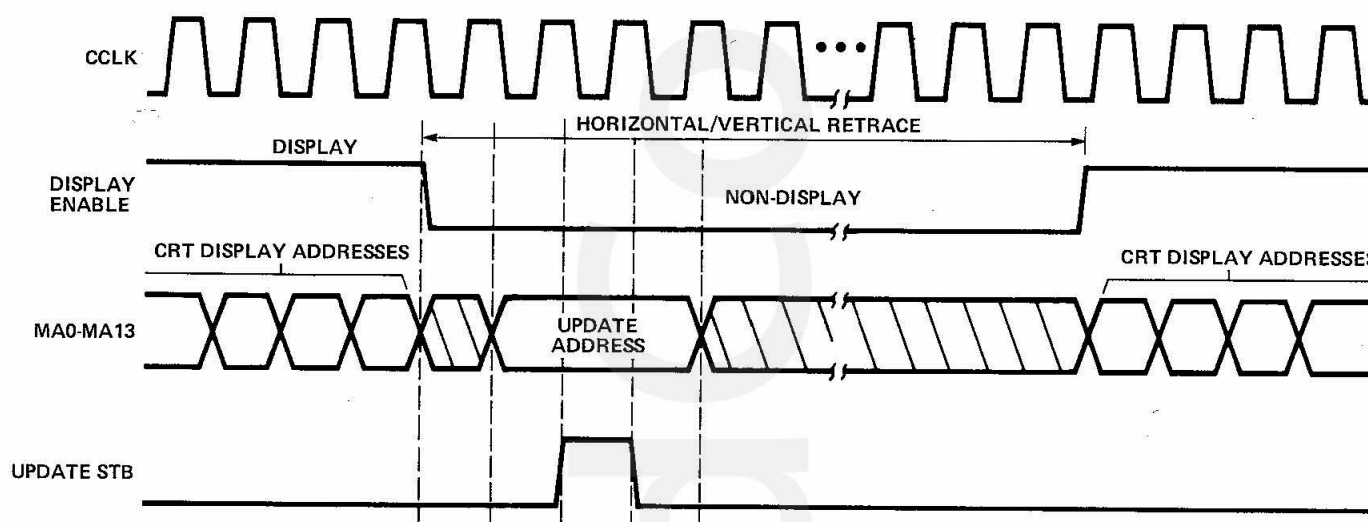


Figure 6. Retrace Update Timings

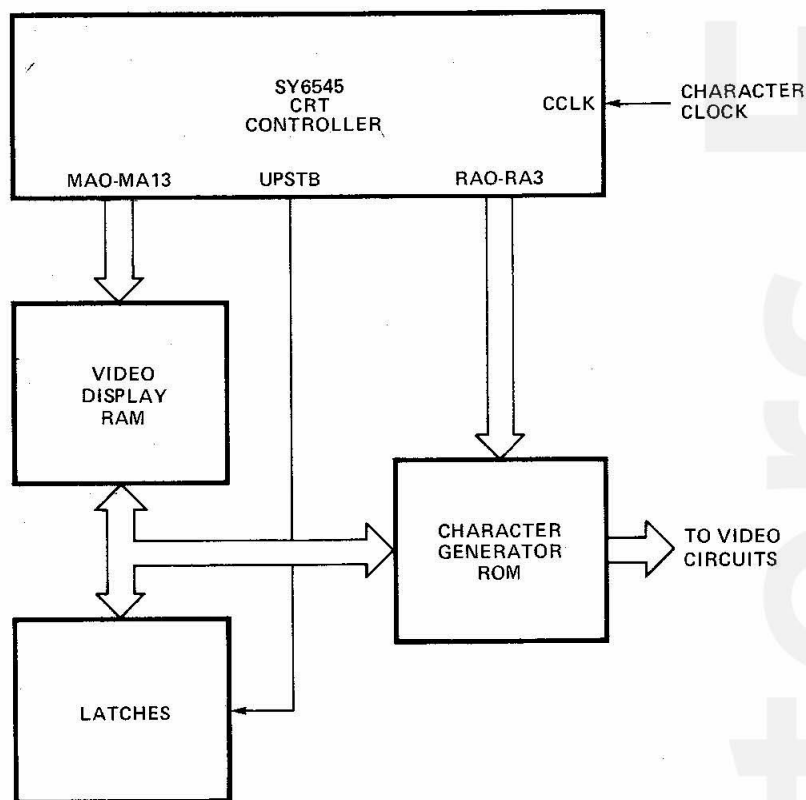


Figure 7. Simplified Diagram for Retrace Updates

Figure 7 illustrates the use of the UPSTB signal to control the data-bus latches.

The relationship of the CCLK, Display Enable, MA lines, UPSTB, and Ready bit in the 6545 status register is shown in Figure 8, an example of timings for successive read operations. The key items to note are:

- The use of the UPSTB to key all hardware operations
- The use of the Ready bit to key all software operations

The function of the Ready bit must be clearly understood:

"1" (READY)

Signifies that the RAM latches contain the data stored in the RAM address of the UAC. Thus, the RAM latches may be read by the MPU or they may be written if it is desired to store new data in the RAM. In the latter case, the data is not stored immediately, but is retained until the next UPSTB or the next available retrace period.

"0" (NOT-READY)

Signifies that the 6545 is either waiting to store data into the RAM or is waiting for the next automatic read from the RAM. In either case, the MPU may not write into or read from the RAM latches during the "not-ready" time.

Figure 9 shows another timing example; this one for the case of successive RAM write operations (updates).

Of particular interest is that memory writes always cause a double update cycle to occur, whereas reads cause only single updates. This is apparent in Figure 9 and is required because the 6545 must assume the processor will do a read operation and therefore must pre-load the latches with the RAM data. If the processor should do a write operation, this is no problem, since the data in the latches is ignored.

For example, in Figures 8 and 9, the scan is assumed to be entirely within the non-display (retrace) time. In a real situation, updates may be desired which take more time than is available during a retrace period. This conflict is resolved by the Ready bit in the 6545, which simply stays "not ready" until the next available retrace time, so no special processor considerations must be made.

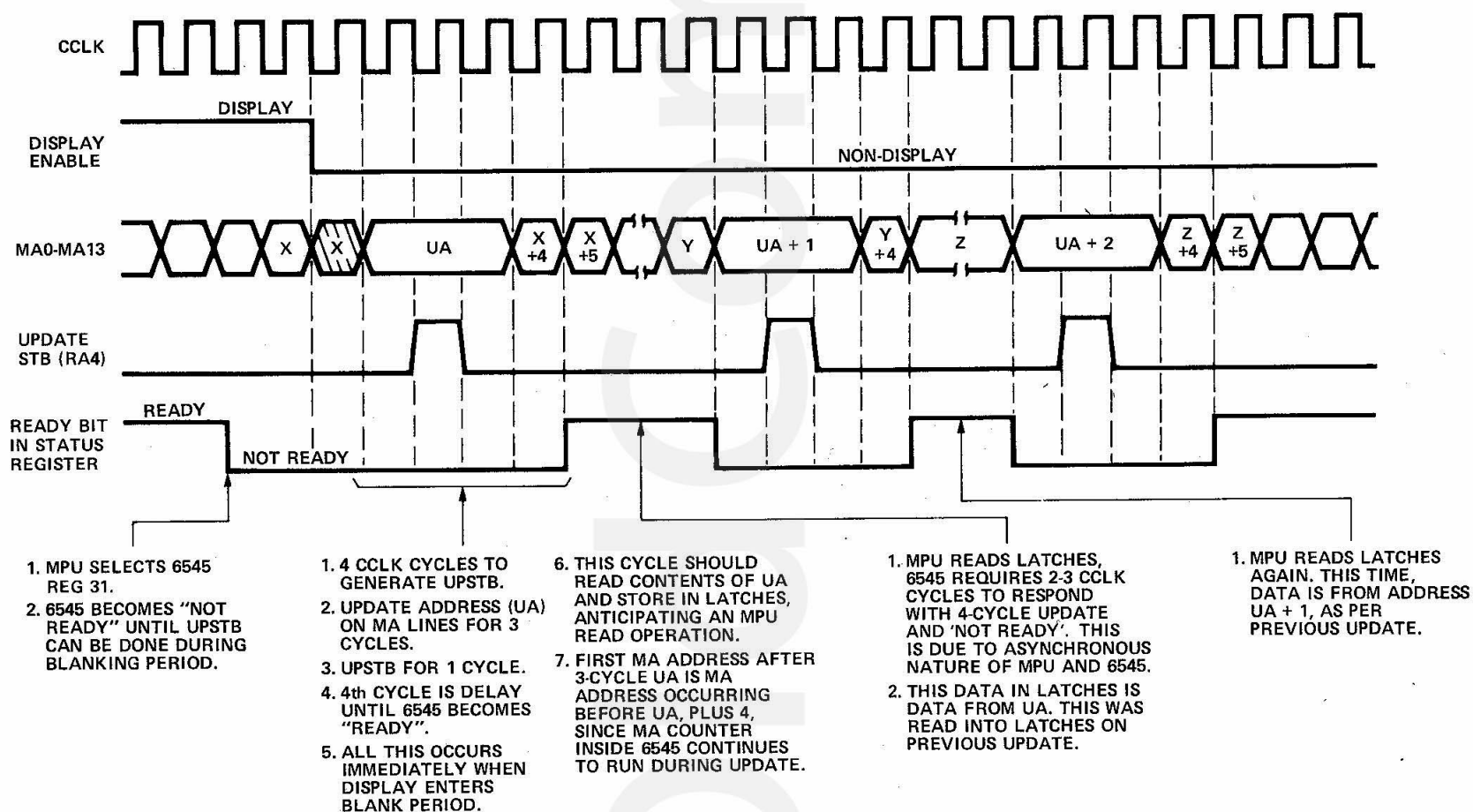


Figure 8. Timing for Successive Memory Reads as Scan Starts Non-Display Period

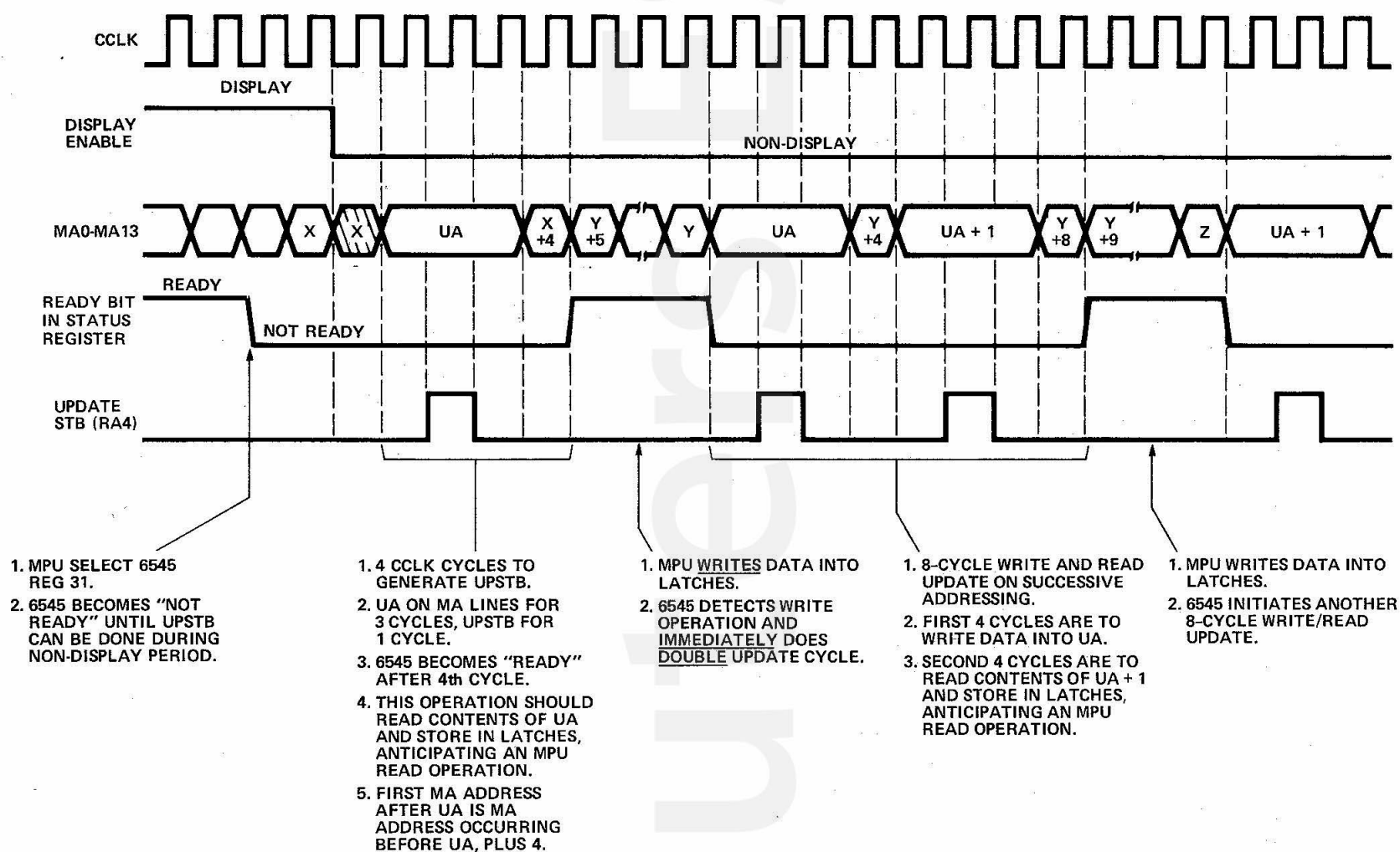


Figure 9. Timing for Successive Memory Writes as Scan Starts Non-Display Period

Figures 10 through 12 illustrate the timings which result when update cycles occur very near the end of a non-displayed time interval. In Figures 10 and 11, the update is performed before the display becomes active (although in

Figure 11, the cycle is truncated) and in Figure 12, the update is aborted and delayed until the next occurring non-display time.

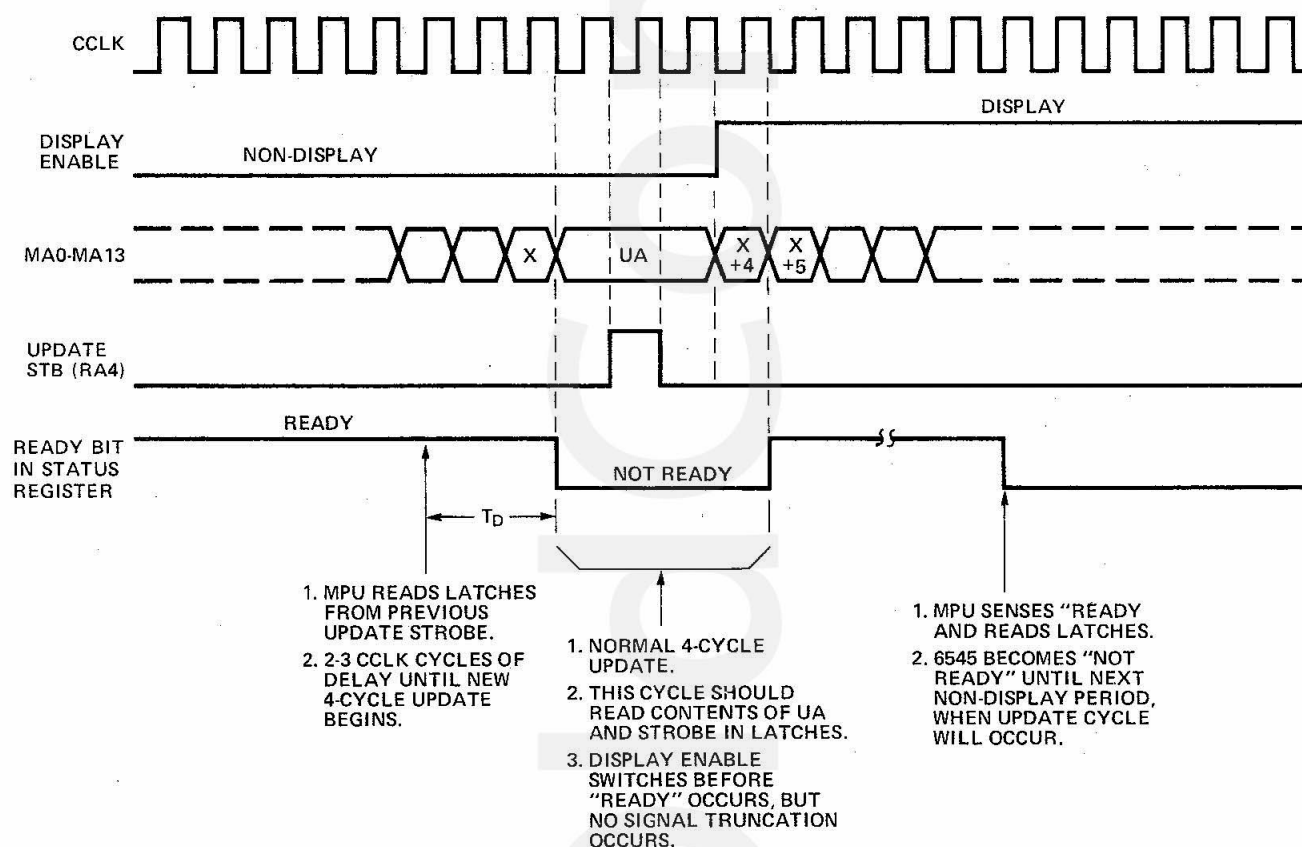


Figure 10. Memory Read at End of Non-Display Period (No Truncation)

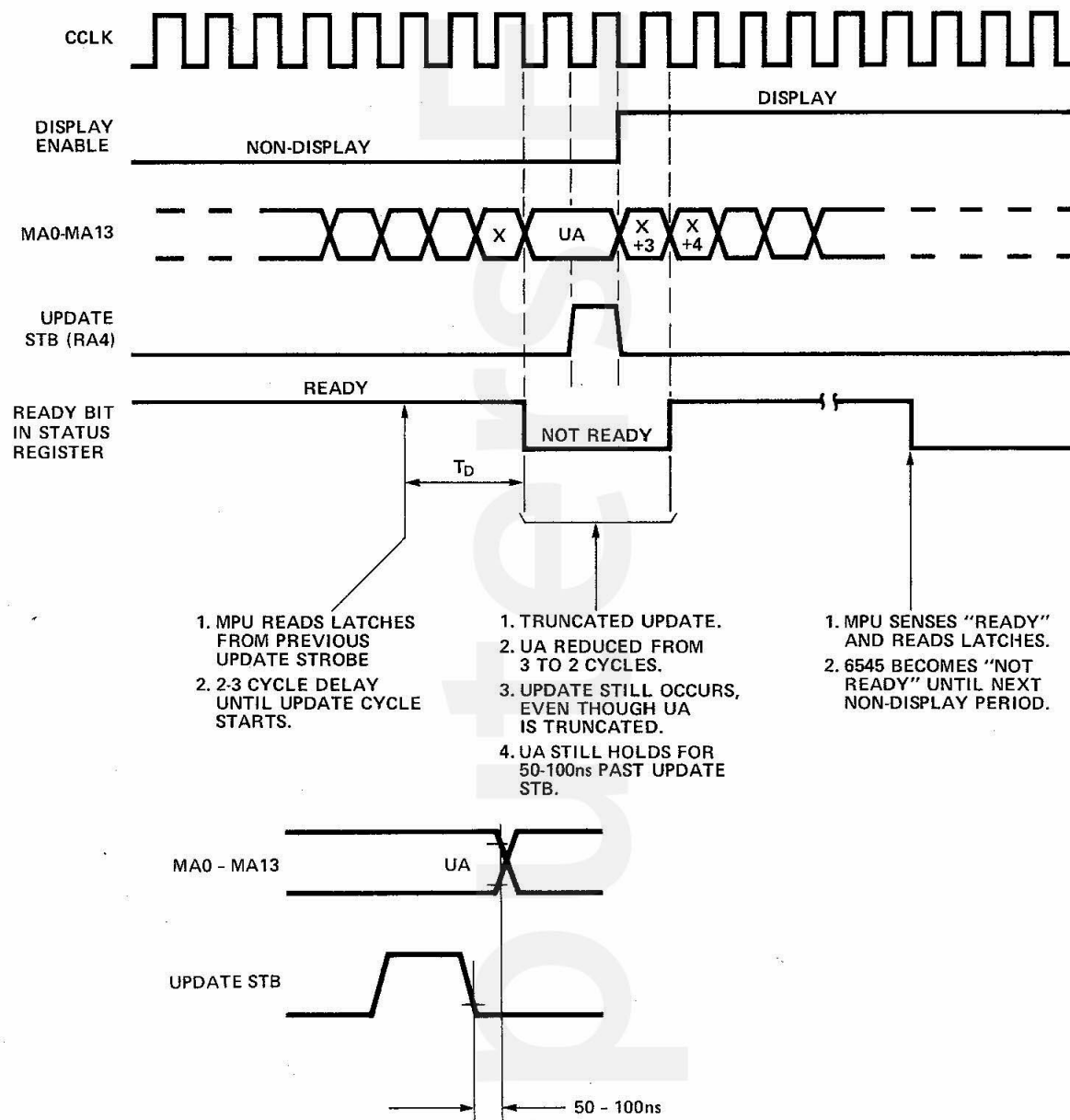


Figure 11. Truncated Memory Read at End of Non-Display Period

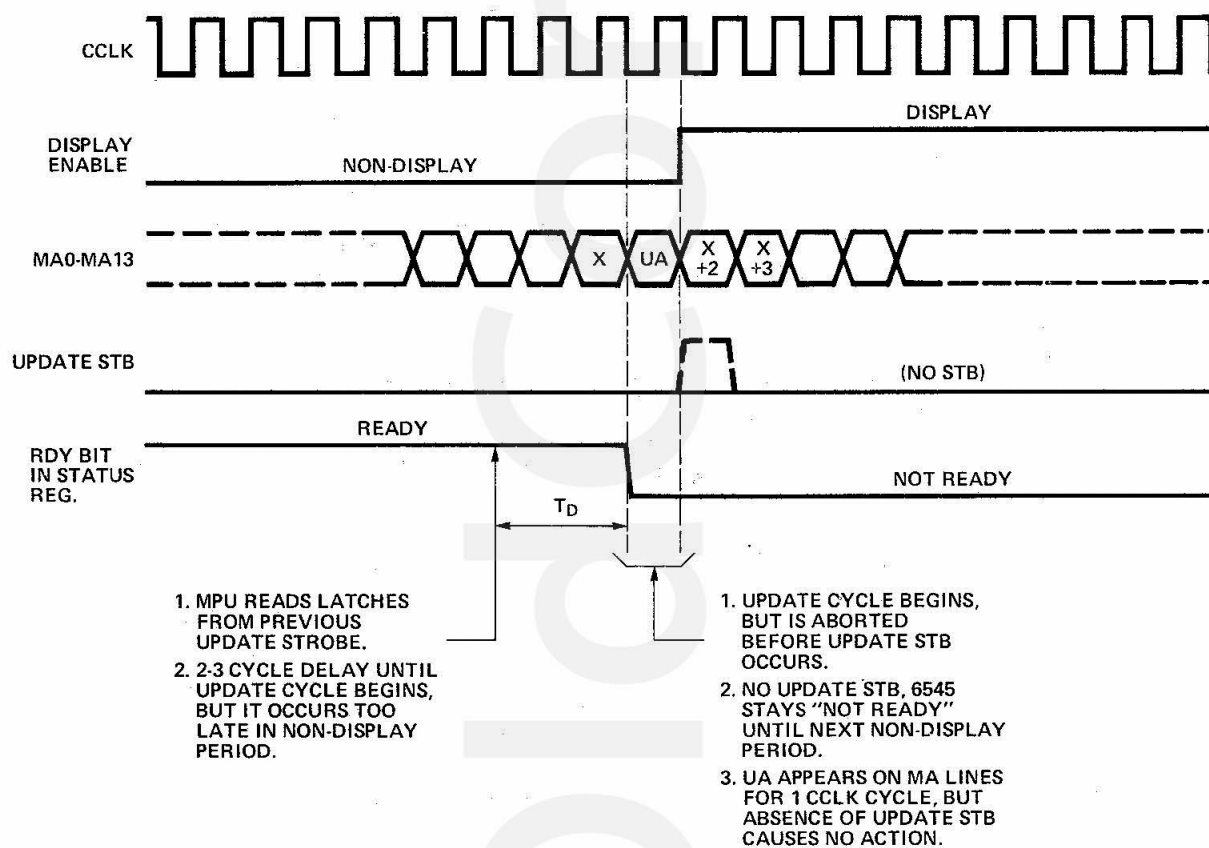


Figure 12. Aborted Memory Read at End of Non-Display Period

5.2 Implementation

Figure 13 shows the detailed circuit schematic and Figure 14, the related timings.

- Note the absence of the 74157 multiplex switches in Figure 13. Multiplexing is done internal to the SY6545 in this mode. When memory updates are required, the SY6545 generates the required memory address on the

MA output lines and a strobe on RA4. This causes the required update to occur.

- The data bus interface for the MPU is implemented with bi-directional latches, in this case, two 74373 latch units. Bi-directional operation permits both writes and reads to be done by the MPU.

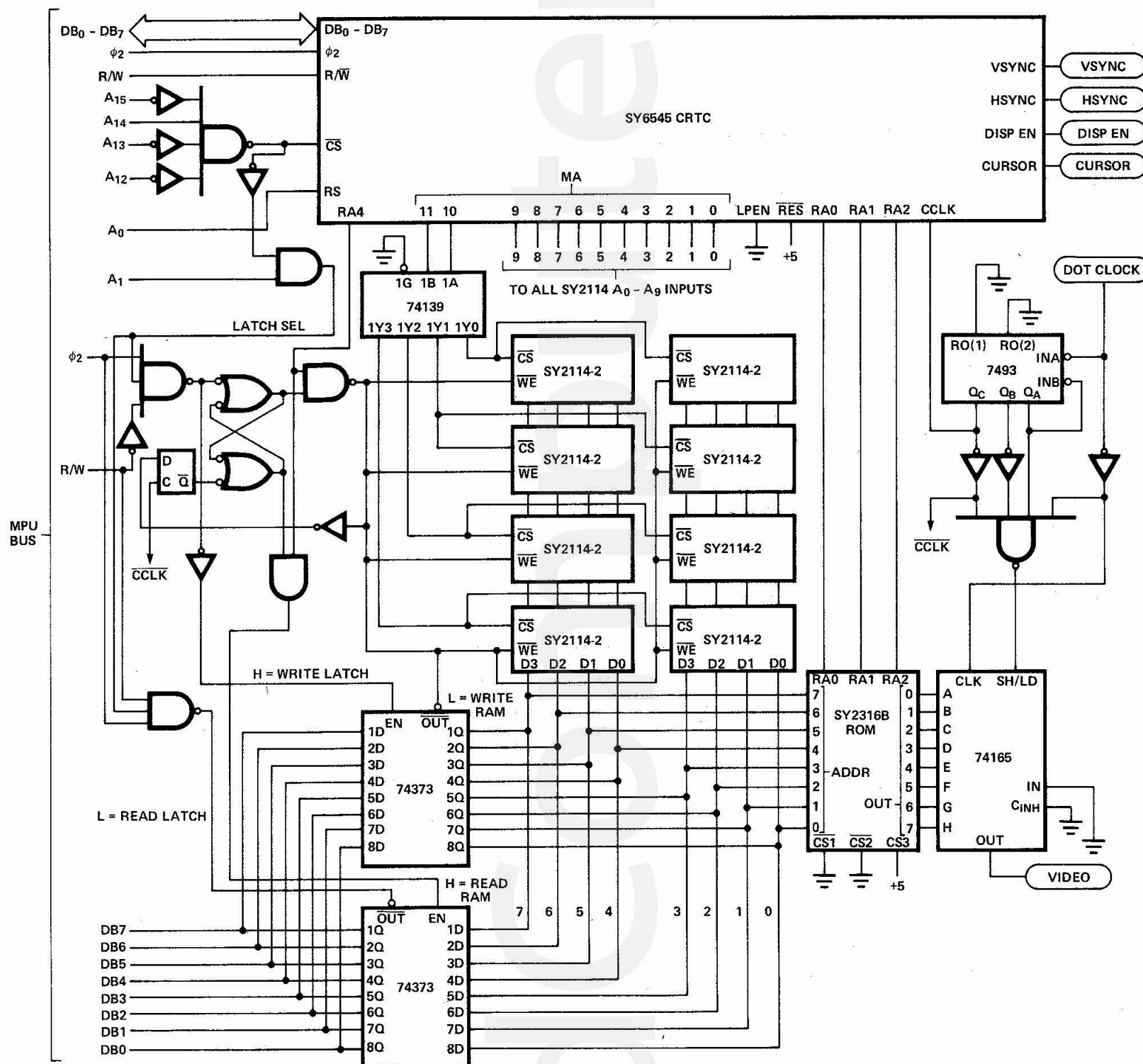


Figure 13. Transparent Addressing (Blanking Update) Schematic

Table 5.

SELECT	ADDRESS																PROGRAM ADDRESS
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CRTC ADDR/STATUS	0	1	0	0	—	—	—	—	—	—	—	—	—	—	—	0	4000
CRTC DATA	0	1	0	0	—	—	—	—	—	—	—	—	—	—	—	1	4001
LATCHES (RAM)	0	1	0	0	—	—	—	—	—	—	—	—	—	—	1	1	4003

— DENOTES "DON'T CARE"

- The RAM is not addressed as a memory block, as in the previous cases, but only the latch need be selected. In effect, then, the memory appears as a port to the MPU. Table 5 summarizes this.
 - When address 4003 is generated to select the latches, however, the CRTC DATA function is simultaneously selected. This is required for correct transparent Addressing operation.
 - The RA4 pin of the SY6545 functions as a special strobe for transparent addressing. Since memory accesses only occur during screen blanking times (horizontal or vertical), some signal must be used to trigger these operations. RA4 does this on the SY6545.
- Notice the special nature of the addressing:
- When addresses 4000 or 4001 are generated by the MPU, only the respective CRTC function is selected; the latches are unaffected.

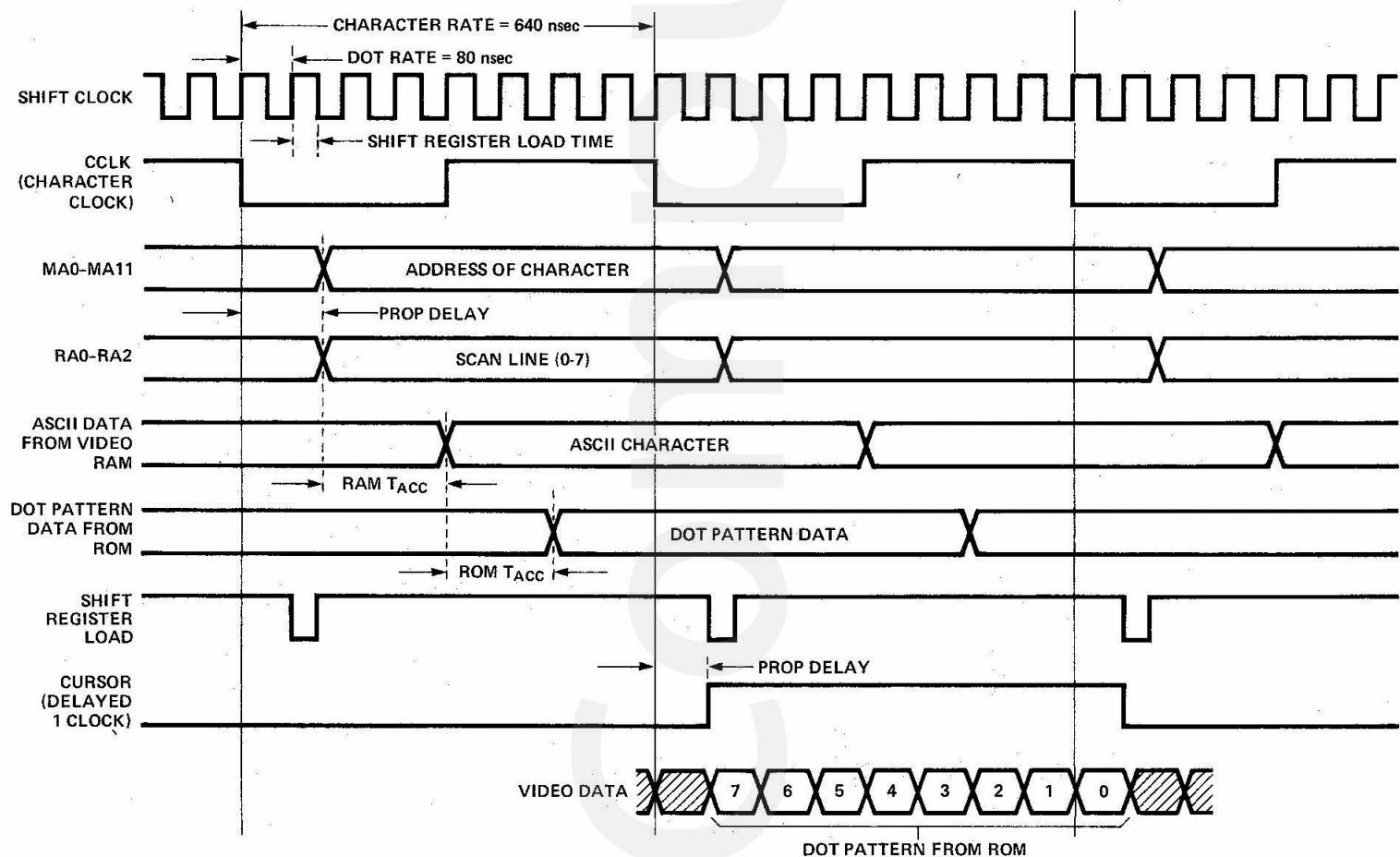


Figure 14. Transparent Addressing (Blanking Update) Timing

- The operation of the control logic is crucial to understanding this particular implementation. Figure 15 illustrates the logic and Figure 19, timing details for the control of memory operations:
- The timings in Figure 15 are for the case of successive MPU writes into the RAM. The operation is initiated when the MPU selects REG 31 in the SY6545. This has no effect on the control logic, but causes the CRTC (SY6545) to begin its transparent update sequence.

When the display timing enters a blanking interval (horizontal or vertical), the CRTC puts the internal Update Address (UA) on the MA lines and generates a strobe on RA4. Since the WRITE PENDING line is low, no memory write occurs but, instead a read is activated and the memory data is stored in the lower of the two latches. This step is required, since the CRTC does not yet know whether a memory write or a read is to be done. Hence, it assumes a read and fetches the data in advance.

Next in the sequence, the MPU either writes data into or reads from the latch. In Figure 16, write cycles are assumed, since this is the more complex example. Writing into the latches sets the cross-coupled latch and the WRITE PENDING signal becomes high. This operation also causes a select on the CRTC, so it knows the data has been stored in the latch. At the next available blank

time (may be immediate), the UA is again put on the MA lines and another RA4 strobe is generated. Since the WRITE PENDING line is high, this causes the latch $\overline{\text{OUT}}$ and the memory $\overline{\text{WE}}$ to simultaneously go low, in turn generating the appropriate memory write operation. At the next negative edge of the CCLK, WRITE PENDING returns low, causing $\overline{\text{OUT}}$ and $\overline{\text{WE}}$ to terminate. Internal to the CRTC, the UA is incremented to UA + 1 and another update is generated. As before, since WRITE PENDING is low, an automatic memory read occurs in anticipation of the next MPU operation, which may be a write or a read.

The sequence may continue indefinitely and will end up with an automatic read cycle after the last MPU write has been done. The CRTC may then be de-selected by programming its internal register select to some register other than REG 31, which is only used for transparent addressing.

- In the case wherein the MPU is to read successive locations, the timing of Figure 16 changes slightly. WRITE PENDING never goes high, so no memory write cycles are generated. Instead, only a series of automatic reads occurs. Figure 17 illustrates this.
- Note that Figure 14, the routine timing for this mode, is identical to that for the MPU Priority scheme.

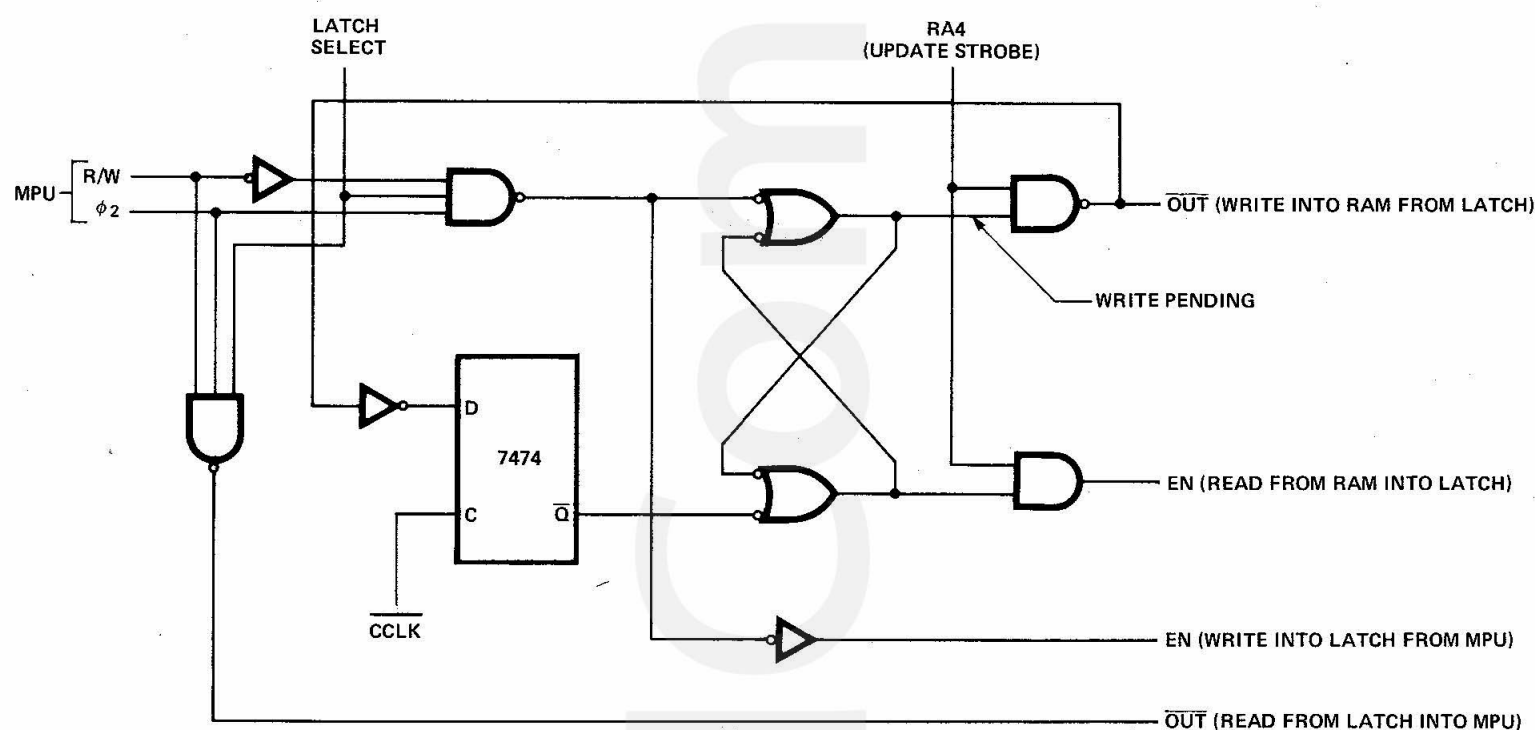


Figure 15. Control Logic for Transparent Addressing (Blanking Updates)

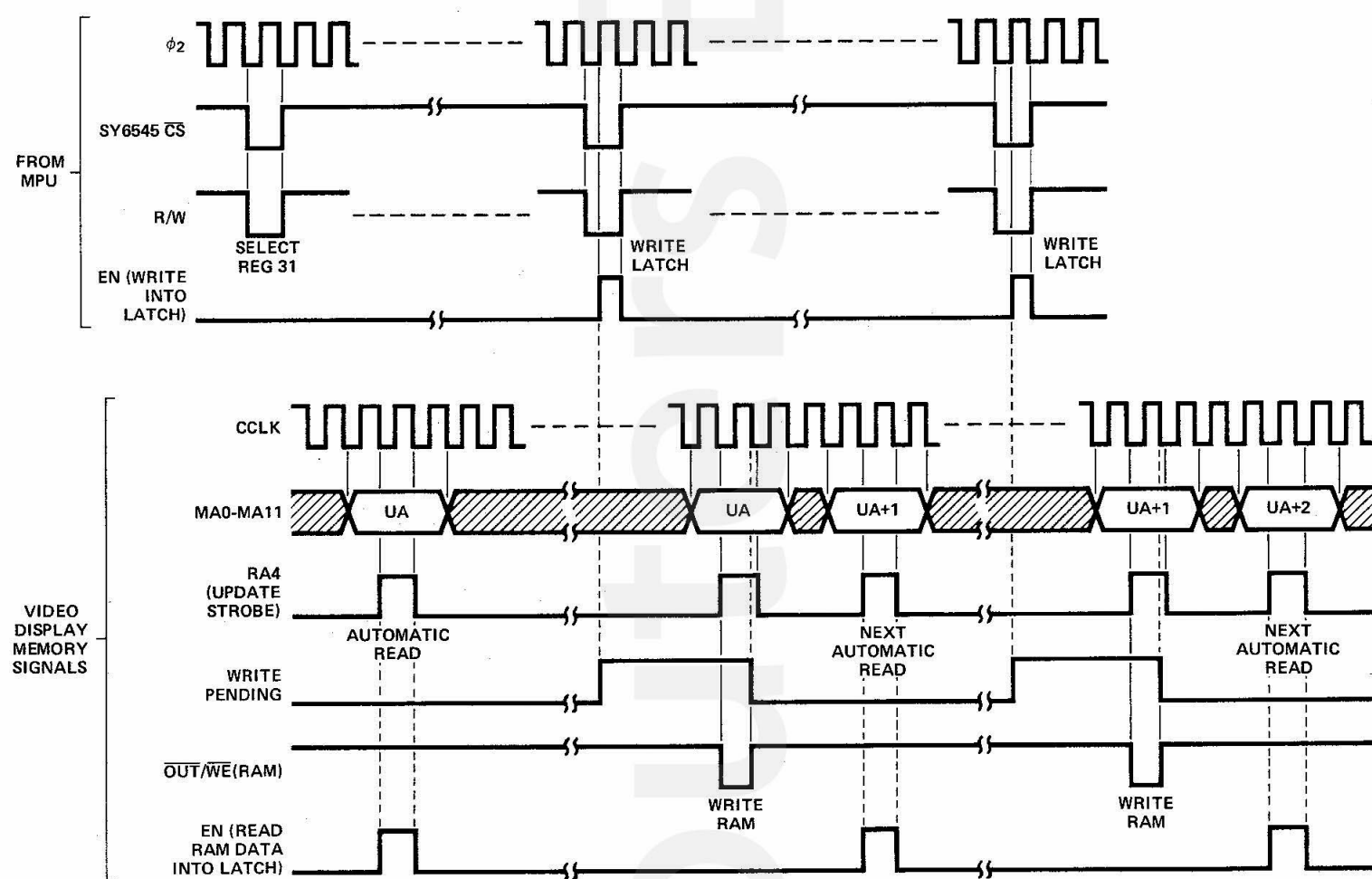


Figure 16. Timing Details for Transparent Addressing (Blanking Updates)

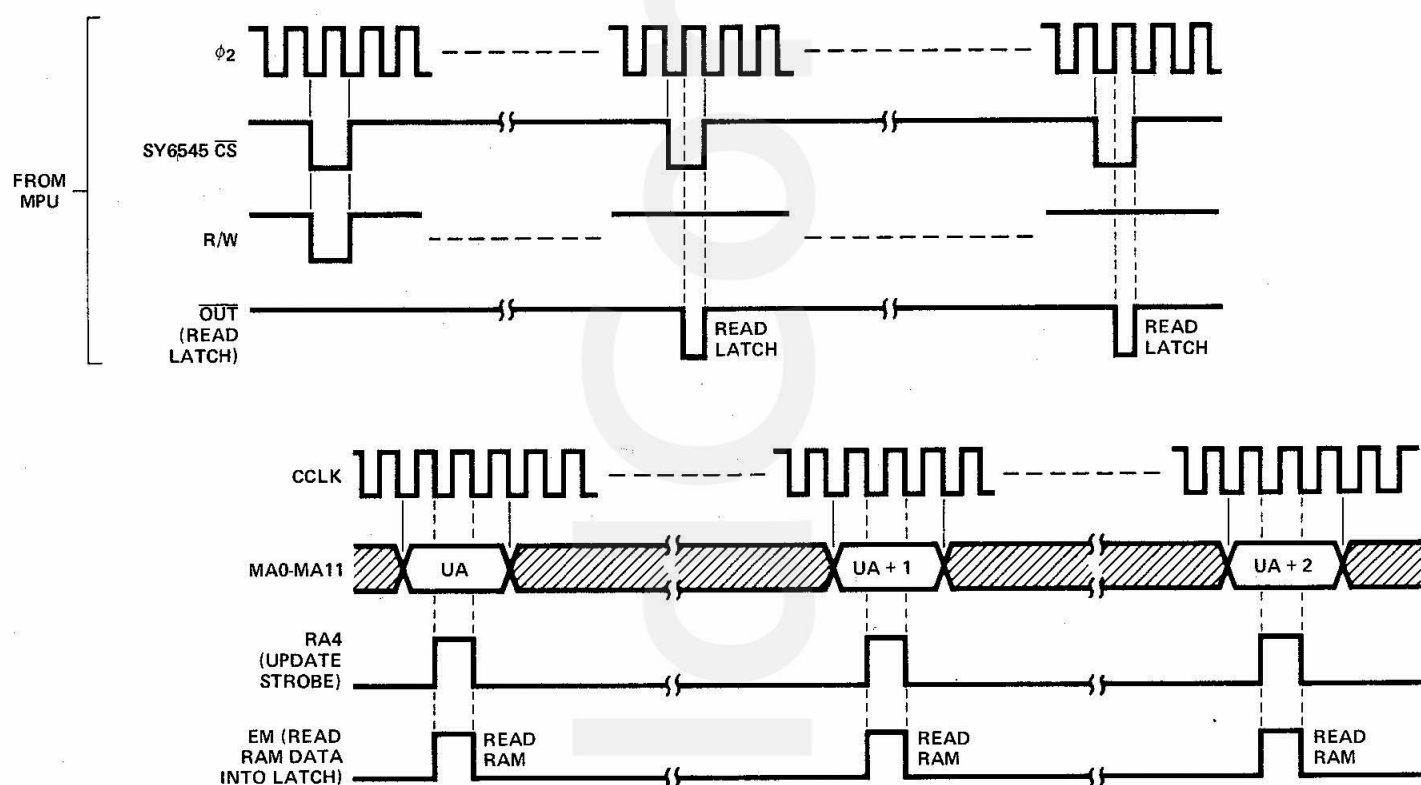


Figure 17. Successive Read Cycles

6.0 Transparent Addressing — ϕ_1/ϕ_2 Interleave

6.1 ϕ_1/ϕ_2 Clock Cycle Interleaving for Updates

The fundamental basis for interleaved RAM accessing is that the 6502 (and 6800) clock cycle consists of two equal portions, ϕ_1 and ϕ_2 , and that ϕ_2 is normally used for MPU data bus operations and ϕ_1 is optionally available for special functions.

The ϕ_2 portion of each clock cycle is always made available to the system MPU, while ϕ_1 is dedicated to normal CRT screen refresh functions. Thus each ϕ_1 cycle is used by the CRTC to fetch character data from the Video Display RAM to keep the display active. The ϕ_2 portion is used for all MPU operations, among them being access to the Video Display RAM to perform read and update cycles.

A unique requirement of this mode is that the MPU clock (ϕ_2) must be synchronous with the screen character clock (CCLK). No degree of asynchronism is allowed. This means, then, that the system MPU must not be permitted to free run, but must be slaved to the character clock.

Since the dot rate of the display is fixed by the screen display parameters, this determines the CCLK rate, also. The MPU must not run slower than the CCLK rate, but must be either equal to it or an even multiple faster rate. In most cases, the CCLK rate will be around 1-2 MHz. Under these conditions, equal CCLK and MPU clock rates work quite well.

One additional requirement of the CRTC for this mode is that the Video Display RAM address lines (MA0-MA13) switch between the display scan address (for screen data) and the read/update address (for MPU operations). This is automatically done by the SY6545 when this mode is selected and results in the ϕ_1 portion of the clock cycle with the scan address on the MA lines and the ϕ_2 portion, with the contents of the Update Address Register of the CRTC on the MA lines. Thus, any MPU operation on the Video Display RAM will be done on the programmed update address and will be independent of the display scanning function. Figure 19 illustrates the timing for this mode, when the CCLK and MPU clock rates are equal.

Several points are noteworthy, regarding Figure 19.

- ϕ_2 causes the multiplexing between the scan address and the update address.
- CCLK advances the scan address on each occurrence of a CCLK pulse.
- Video Display RAM select (either read or write) causes the update address to increment. This simplifies operations which require updating or reading a long sequence of screen characters.
- Because of the propagation delay from ϕ_2 clock edges to the MA lines switching, care must be taken to ensure fast enough RAMs to respond to the relatively short access time requirements which result.

6.2 ϕ_1/ϕ_2 Implementation

- Address multiplex switches are not required, since this is accomplished internal to the SY6545. In this way, this mode resembles the previously mentioned Transparent Addressing Mode.

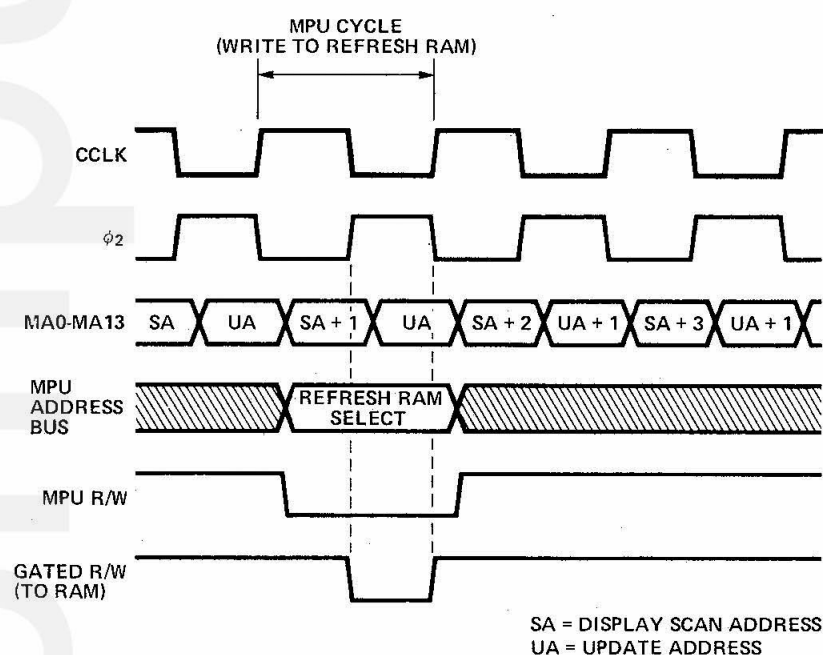


Figure 19. ϕ_1/ϕ_2 Clock Interleaving — Update Timing (Note: UA is update address, SA is scan address)

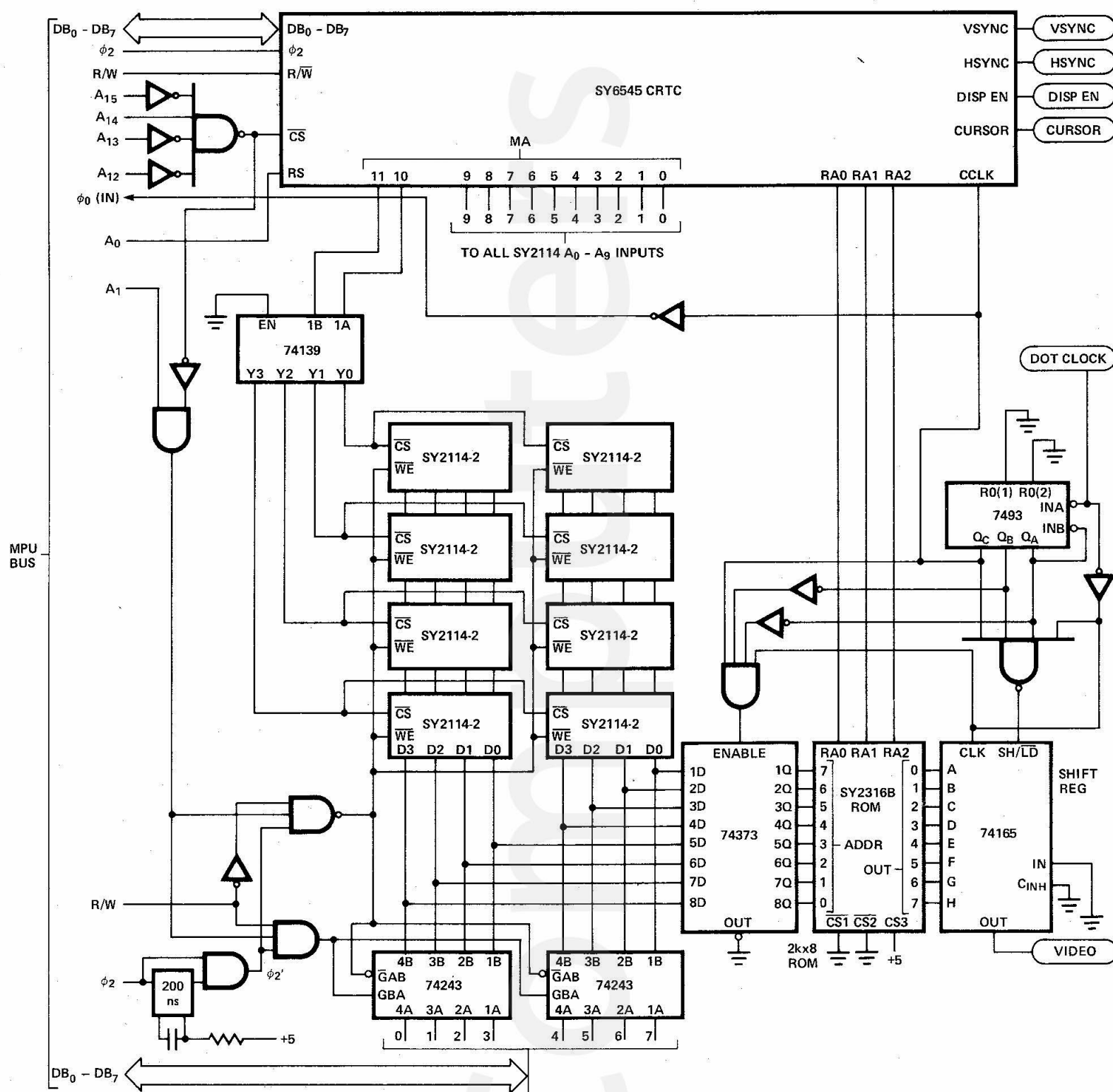


Figure 20. Transparent Addressing (ϕ_1/ϕ_2 Interleave) Schematic

Table 6.

SELECT	ADDRESS												PROGRAM ADDRESS				
	15	14	13	12	11	10	9	8	7	6	5	4		3	2	1	0
CRTC ADDR/STATUS	0	1	0	0	—	—	—	—	—	—	—	—	—	—	—	0	4000
CRTC DATA	0	1	0	0	—	—	—	—	—	—	—	—	—	—	—	1	4001
RAM	0	1	0	0	—	—	—	—	—	—	—	—	—	—	1	1	4003

— DENOTES "DON'T CARE"

- The RAM is not addressed as a memory block, but as a single address. This derives from the fact that the address selected for updates or reads was previously stored in the CRTC. The table above summarizes the address decoding:

Note that this address decoding is identical to the other transparent mode. No data bus latches are required here, however, since the Update Address is on the MA lines each ϕ_2 cycle and the operation does not have to wait for blanking times.

- A significant difference between this mode and the normal ϕ_1/ϕ_2 Interleave mode is that the address multiplexing in this mode is much slower. Figure 21 shows this. The result of this is that the MPU ϕ_2 clock, normally used to gate the MPU R/W signal to generate $\overline{WE}$ for the RAM, may not be used for this purpose. If it is, a RAM write operation could inadvertently affect an undesired address location. This would happen when $\overline{WE}$ went low

(write) before the Update Address was driven onto the MA lines. To resolve this, it is necessary to generate a special gating clock (ϕ_2') for RAM operations. ϕ_2' is identical to ϕ_2 , except the leading edge is delayed about 150-200 nsec to compensate for the slow switching time of the MA lines.

- The propagation delay of the MA lines, relative to ϕ_2 , results in a requirement for faster RAMs, since the write cycle (and read cycle) is shortened by 150-200 nsec. Alternately, slower system timing may be used. In the example in Figures 20 and 21 all the times are doubled. In effect then, for this case, the display size was reduced from 80 x 24 to 40 x 24.
- Figure 21 also illustrates the use of a clocked Cursor signal (clocked by the Shift Register $\overline{LD}$). This is a simple technique to align the Cursor output from the CRTC with the Video dot pattern emanating from the Shift Register. This same technique is applicable to the Display Enable output.

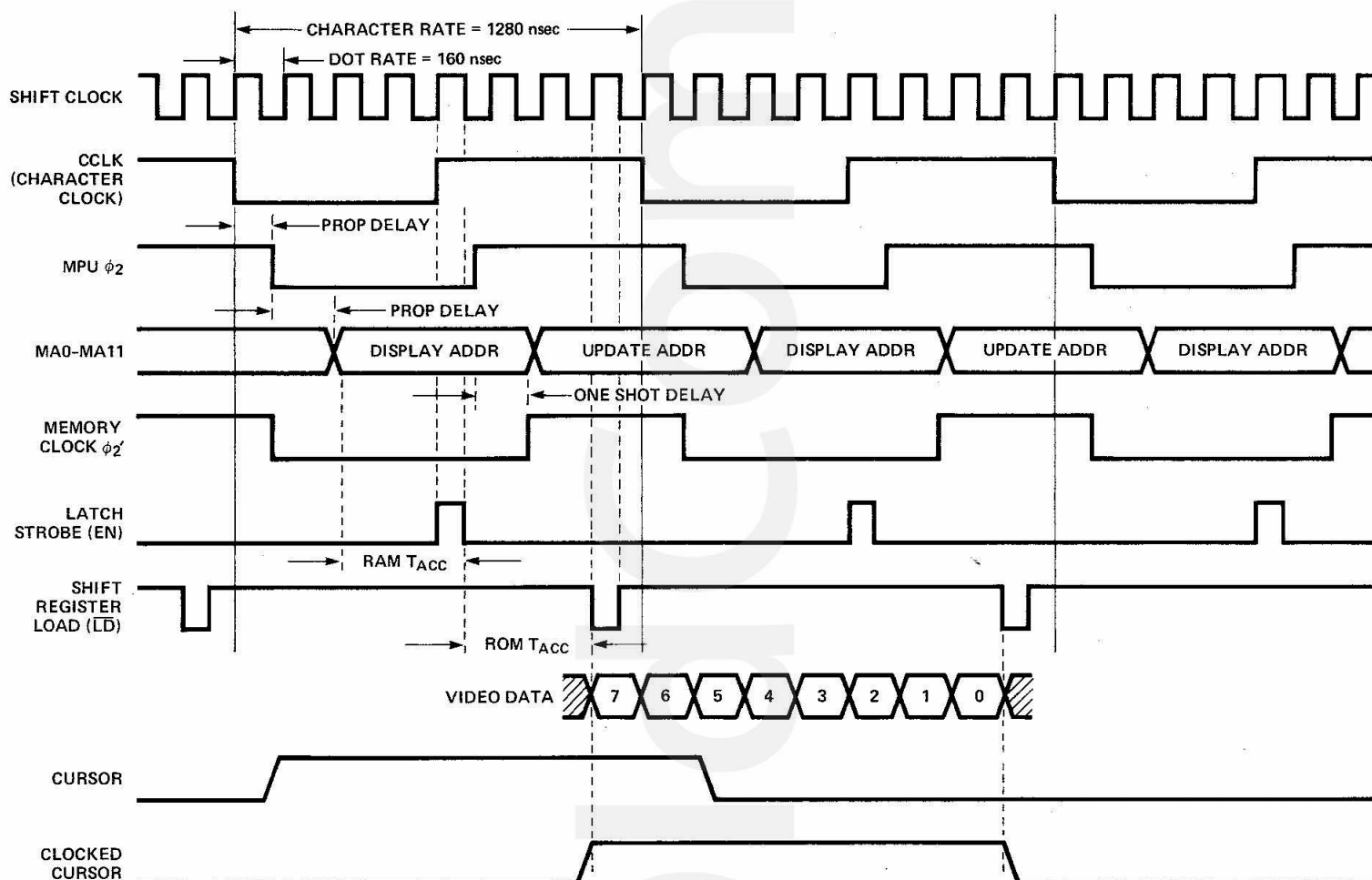


Figure 21. Transparent Addressing (ϕ_1/ϕ_2 Interleave) Timing

6.2 Program Example to Load All SY6545 Internal Registers

A simple SY6500 family program to load all internal registers of the SY6545 is given in Figure 22. Typically, this routine is executed on power-up or System reset and rarely, if ever, otherwise. All methods outlined in this note may utilize the same routine, since the CRTC addressing is identical for all.

7.0 Program Examples to Load a Complete Screen of Displayed Text

Programming examples for loading the Video Display RAM with text to be displayed are shown in Figures 24-26 (for the various memory addressing modes). Each routine uses the

```

0000                                     *=$200
0200                                :REGISTER DATA TABLE
0200 31                                .BYT 49                                :HOR TOT(N-1)
0201 28                                .BYT 40                                :HOR DISP
0202 2D                                .BYT 45                                :HOR SYNC POS
0203 11                                .BYT $11                                :HOR/VER SYNC WIDTHS
0204 1F                                .BYT 31                                :VER TOT (N-1)
0205 0B                                .BYT 11                                :VER TOT ADJ
0206 18                                .BYT 24                                :VER DISP
0207 1C                                .BYT 28                                :VER SYNC POS
0208 B8                                .BYT $B8                                :MODE CONTROL
0209 07                                .BYT 07                                :SCAN LINES (N-1)
020A 00                                .BYT $00                                :CUR START
020B 07                                .BYT $07                                :CUR END
020C 00                                .BYT $00                                :START - HI
020D 00                                .BYT $00                                :START - LO
020E 00                                .BYT $00                                :CUR - HI
020F 00                                .BYT $00                                :CUR - LO
0210 00                                .BYT $00                                :LPEN - HI
0211 00                                .BYT $00                                :LPEN - LO
0212 00                                .BYT $00                                :UA - HI
0213 00                                .BYT $00                                :UA - LO
0214                                AD6545=$4000
0214                                :
0214                                :-----LOAD ALL 6545 REGISTERS-----
0214                                :
0214                                *=$300
0300 A0 14                                START1 LDY #$14
0302 A2 00                                LDX #0
0304 BA                                BACK1 TXA
0305 BD 00 40                                STA AD6545
0308 BD 00 02                                LDA $200,X
030B BD 01 40                                STA AD6545+1
030E E8                                INX
030F B8                                DEY
0310 D0 F2                                BNE BACK1

```

Figure 22. Load Internal Registers

same algorithm (for demonstration purposes):

1. Load all blank characters (hex 20) to initialize screen (all blank).
2. Load 16 x 16 matrix of characters in center of screen. This will comprise entire set of 256 characters available in SY2316B Character Generator ROM. Figure 23 illustrates the display.

The following points highlight the differences in the programs of Figures 24-26:

- The routine in Figure 24 is very simple and consists of only loading a memory block directly.
- The routine in Figure 25 (Transparent Addressing — Blanking Updates) illustrates the use of the Update Address Register, the “dummy” register (Register 31), the Data Holding Latches, and the Ready bit in the Status Register.
- The routine in Figure 26 (Transparent Addressing — ϕ_1/ϕ_2 Interleave) uses the Update Address Register and

the “dummy” register, but does not require latches or the Ready bit.

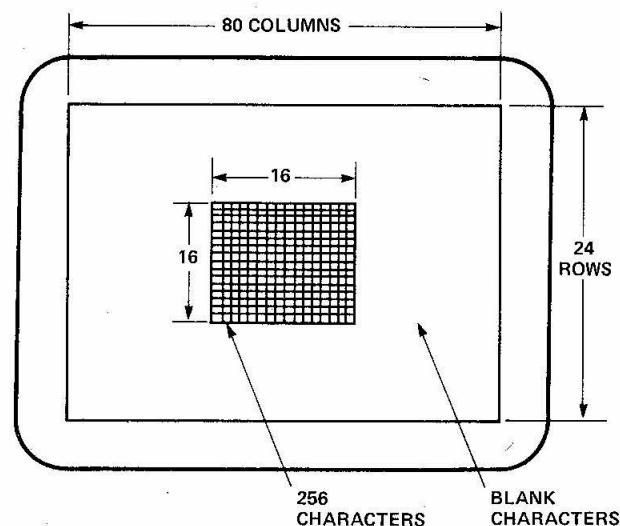


Figure 23. Screen Display Layout for Sample Programs

```

0312          ;
0312          ;----- LOAD ASCII BLANKS INTO ALL MEMORY -----
0312          ;
0312  A9 50          LDA #$50
0314  8D 1F 03      STA LOC+2
0317  A2 0F          LDX #$0F
0319  A9 20          LDA #$20          ;ASCII BLANK
031B  A0 00          LOOP1 LDY #0
031D  99 00 50      LOC STA RAM,Y
0320  C8            INY
0321  D0 FA          BNE LOC
0323  EE 1F 03      INC LOC+2
0326  CA            DEX
0327  10 F2          BPL LOOP1
0329          ;
0329          ;----- LOAD CHARACTER SET INTO DISPLAY MEMORY -----
0329          ;
0329  DISP = $5160          ;RAM ADDR OF TOP LEFT CHARACTER
0329  ADDR = $0040
0329  DATA = $0042
0329  INDEX = $0043
0329  A2 0F          LDX #$0F          ;HOR LINE COUNT
032B  A9 60          LDA #<DISP        ;START ADDR - L
032D  85 40          STA ADDR
032F  A9 51          LDA #>DISP        ;START ADDR - H
0331  85 41          STA ADDR+1
0333  A9 00          LDA #0
0335  85 42          STA DATA          ;ASCII DATA
0337  A9 0F          L1 LDA #$0F
0339  85 43          STA INDEX          ;ROW COUNT
033B  A0 00          LDY #0
033D  A5 42          L2 LDA DATA        ;GET ASCII CHARACTER
033F  91 40          STA (ADDR),Y       ;STORE IN RAM
0341  C8            INY
0342  E6 42          INC DATA          ;NEXT ASCII CHARACTER
0344  C6 43          DEC INDEX
0346  10 F5          BPL L2
0348  18            CLC
0349  A5 40          LDA ADDR
034B  69 50          ADC #80            ;NEXT LINE
034D  85 40          STA ADDR
034F  90 02          BCC L3
0351  E6 41          INC ADDR+1
0353  CA            L3 DEX
0354  10 E1          BPL L1

```

Figure 24. Load Display Memory (MPU Priority and ϕ_1/ϕ_2 Interleave Modes)

```

0312      :
0312      :-----LDA ASCII BLANKS INTO ALL MEMORY-----
0312      :
0312      A9 1F          LDA #31
0314      8D 00 40      STA AD6545          :R31 SELECT
0317      A2 0F          LDX #0F
0319      A0 00      LOOP1 LDY #0
031B      AD 00 40      LOC   LDA AD6545          :STATUS
031E      10 FB          BPL LOC
0320      A9 20          LDA #$20          :ASCII BLANK
0322      8D 03 40      STA LATCH          :WRITE DATA
0325      C8            INY
0326      D0 F3          BNE LOC
0328      CA            DEX
0329      10 EE          BPL LOOP1
032B      :
032B      :-----LOAD CHARACTER SET-----
032B      :
032B      DISP=$0160
032B      ADDR=$0040
032B      DATA=$0042
032B      INDEX=$0043
032B      A9 00          LDA #0
032D      85 42          STA DATA
032F      A2 0F          LDX #0F
0331      A9 01          LDA #>DISP
0333      85 40          STA ADDR          :UA - HI
0335      A9 60          LDA #<DISP
0337      85 41          STA ADDR+1        :UA - LO
0339      A9 12      L1   LDA #18
033B      8D 00 40      STA AD6545          :R18
033E      A5 40          LDA ADDR
0340      8D 01 40      STA AD6545+1        :UA-HI
0343      A9 13          LDA #19
0345      8D 00 40      STA AD6545          :R19
0348      A5 41          LDA ADDR+1
034A      8D 01 40      STA AD6545+1        :UA - LO
034D      A9 1F          LDA #31
034F      8D 00 40      STA AD6545          :R31 SELECT
0352      A9 0F          LDA #0F          :NO OF CHARS PER LINE
0354      85 43          STA INDEX
0356      AD 00 40      L2   LDA AD6545          :STATUS
0359      10 FB          BPL L2
035B      A5 42          LDA DATA          :GET CHARACTER
035D      8D 03 40      STA LATCH          :STORE IT IN LATCH
0360      E6 42          INC DATA          :NEXT CHARACTER
0362      C6 43          DEC INDEX
0364      10 F0          BPL L2
0366      18            CLC          :GET ADDRESS
0367      A5 41          LDA ADDR+1        :OF FIRST LOCATION
0369      69 50          ADC #$50          :OF NEXT LINE
036B      85 41          STA ADDR+1
036D      90 02          BCC L3
036F      E6 40          INC ADDR
0371      CA      L3     DEX
0372      10 C5          BPL L1

```

Figure 25. Load Display Memory (Transparent Addressing — Blanking Updates)

8.0 Summary

Detailed system implementations have been presented to demonstrate the full range of SY6545 operating modes. In addition to circuit schematics, sample program listings are shown to tie together both software and hardware. These examples are intended to provide the system designer with clear-cut comparisons of each mode of operation.

```

0312      ;-----LDA ASCII BLANKS INTO ALL MEMORY-----
0312      ;
0312      LDA #31
0314      STA AD6545      ;R31 SELECT
0317      LDX #0F
0319      LDY #0
031B      LOC      LDA #20      ;ASCII BLANK
031D      STA RAM      ;WRITE DATA
0320      INY
0321      BNE LOC
0323      DEX
0324      BPL LOOP1
0326      ;
0326      ;-----LOAD CHARACTER SET-----
0326      ;
0326      DISP=$0084      ;RAM ADDR OF TOP LEFT CHARACTER
0326      ADDR=$0040
0326      DATA=$0042
0326      INDEX=$0043
0326      A9 00      LDA #0
0328      B5 42      STA DATA      ;FIRST DATA CHARACTER
032A      A2 0F      LDX #0F      ;NO OF ROWS IN BLOCK
032C      A9 00      LDA #>DISP
032E      B5 40      STA ADDR      ;UA - HI
0330      A9 84      LDA #<DISP
0332      B5 41      STA ADDR+1      ;UA - LO
0334      A9 12      L1      LDA #18
0336      B0 00 40      STA AD6545      ;R18
0339      A5 40      LDA ADDR
033B      B0 01 40      STA AD6545+1      ;UA-HI
033E      A9 13      LDA #19
0340      B0 00 40      STA AD6545      ;R19
0343      A5 41      LDA ADDR+1
0345      B0 01 40      STA AD6545+1      ;UA - LO
0348      A9 1F      LDA #31
034A      B0 00 40      STA AD6545      ;R31 SELECT
034D      A9 0F      LDA #0F      ;NO OF CHARS PER LINE
034F      B5 43      STA INDEX
0351      A5 42      L2      LDA DATA      ;GET CHARACTER
0353      B0 03 40      STA RAM      ;STORE IT IN RAM
0356      E6 42      INC DATA      ;NEXT CHARACTER
0358      C6 43      DEC INDEX
035A      10 F5      BPL L2
035C      18      CLC      ;GET ADDRESS
035D      A5 41      LDA ADDR+1      ;OF FIRST LOCATION
035F      69 28      ADC #40      ;OF NEXT LINE
0361      B5 41      STA ADDR+1
0363      90 02      BCC L3
0365      E6 40      INC ADDR
0367      CA      L3      DEX
0368      10 CA      BPL L1

```

Figure 26. Load Display Memory (Transparent Addressing ϕ_1/ϕ_2 Interleave)

9.0 Split-Screen Displays

The SY6545 CRT Controller fulfills most requirements for raster-scan display system designs. Certain display applications, however, present special problems requiring novel solutions.

One such application is split-screen displays. The basic concept of a split-screen display is illustrated in Figure 27.

The diagram on the left illustrates a conventional display, characterized by a text area composed of rows and columns of alphanumeric characters. The split-screen display, however, consists of both a left and a right side text area. The real complication of implementation of a split-screen CRT control system arises when special requirements of such a display are tabulated:

- Left and right side texts must have independent scrolling
- Left and right side text memory must be separate and independent

The impact of these requirements is that conventional CRT Controllers, like the SY6545, do not have facilities to accommodate these needs. Special techniques may be employed with the SY6545, however, to achieve split-screen display.

9.1 Functional Requirements

The specific application presented here is assumed to have certain constraints:

- Left and right side texts are comprised of characters of the same size
- The number of columns in each text is variable and they need not be equal
- The space between texts is selectable
- The top row of each text is at the same position
- The number of displayed rows in each text may be different

These constraints are not very limiting and, in fact, permit a great deal of display flexibility. Figure 28 illustrates the CRT display timing in simplified form.

For a complete understanding of the display timing, the reader is referred to the SY6545 data sheet. Of special interest, regarding Figure 28 is the lower set of waveforms. Note especially, the timing of Display Enable. For each scan line, Display Enable, is true for two separate time intervals: once, for the left side display, and once for the right side. Memory addresses (MA lines) will be non-contiguous for each displayed time; that is the addresses for the left side will not be related to those of the right.

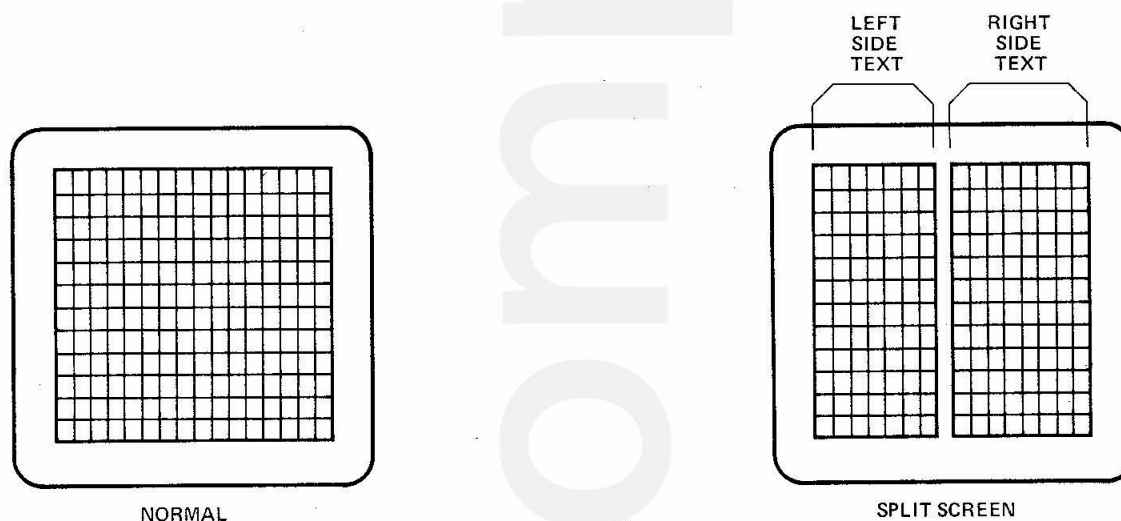


Figure 27. Normal vs. Split-Screen Display Illustration

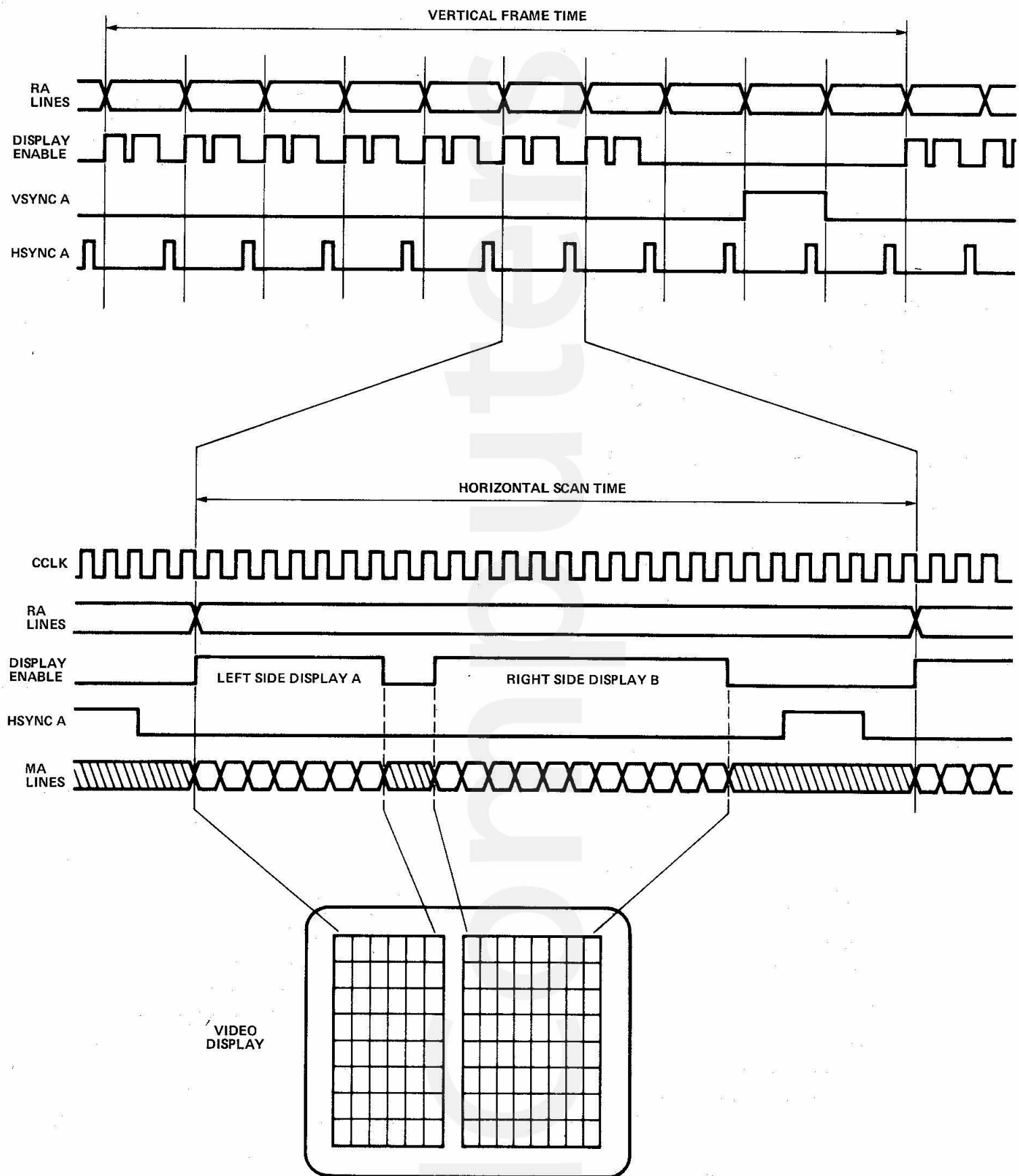


Figure 28. Simplified Split-Screen Timing

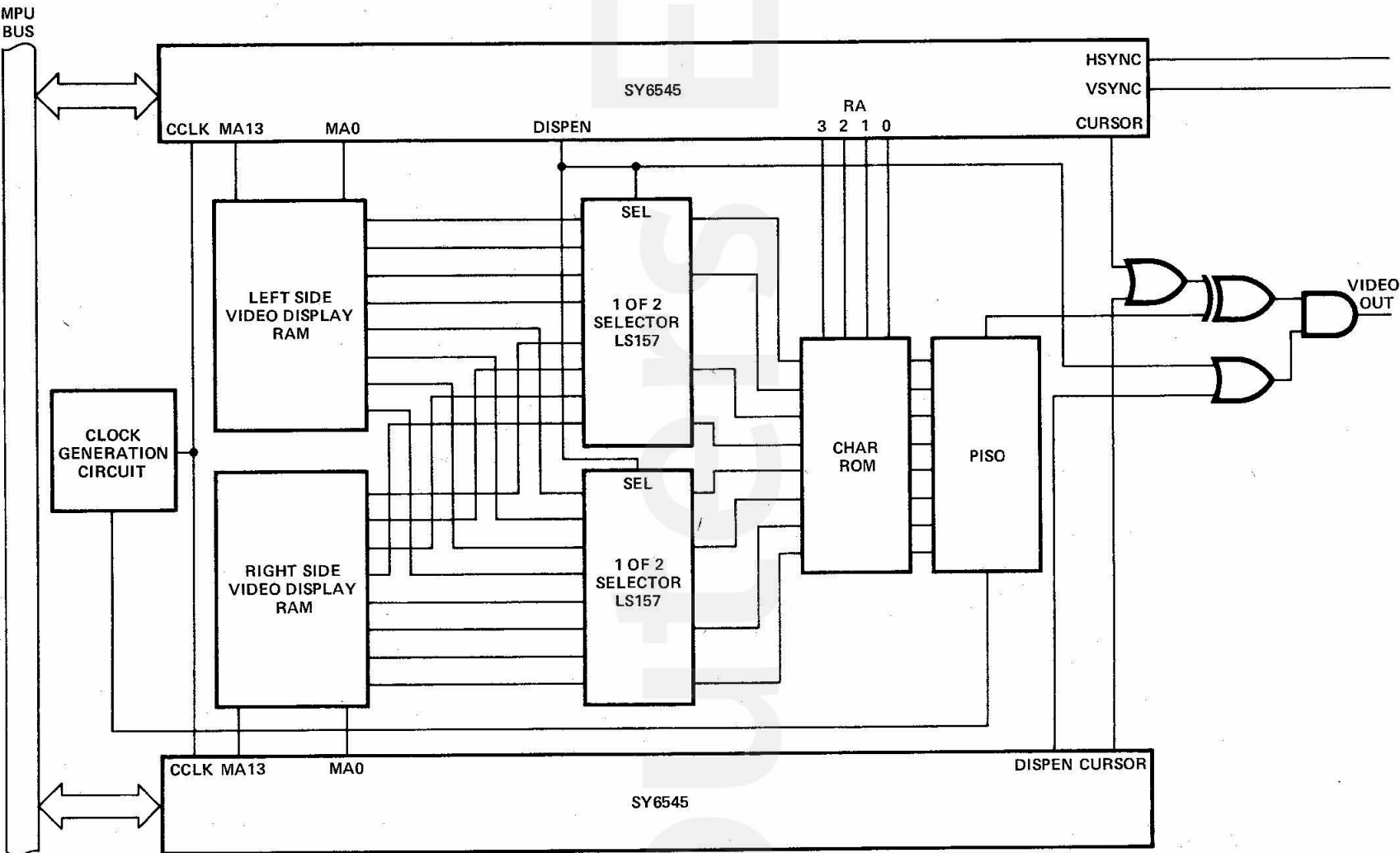


Figure 29. Logic Implementation

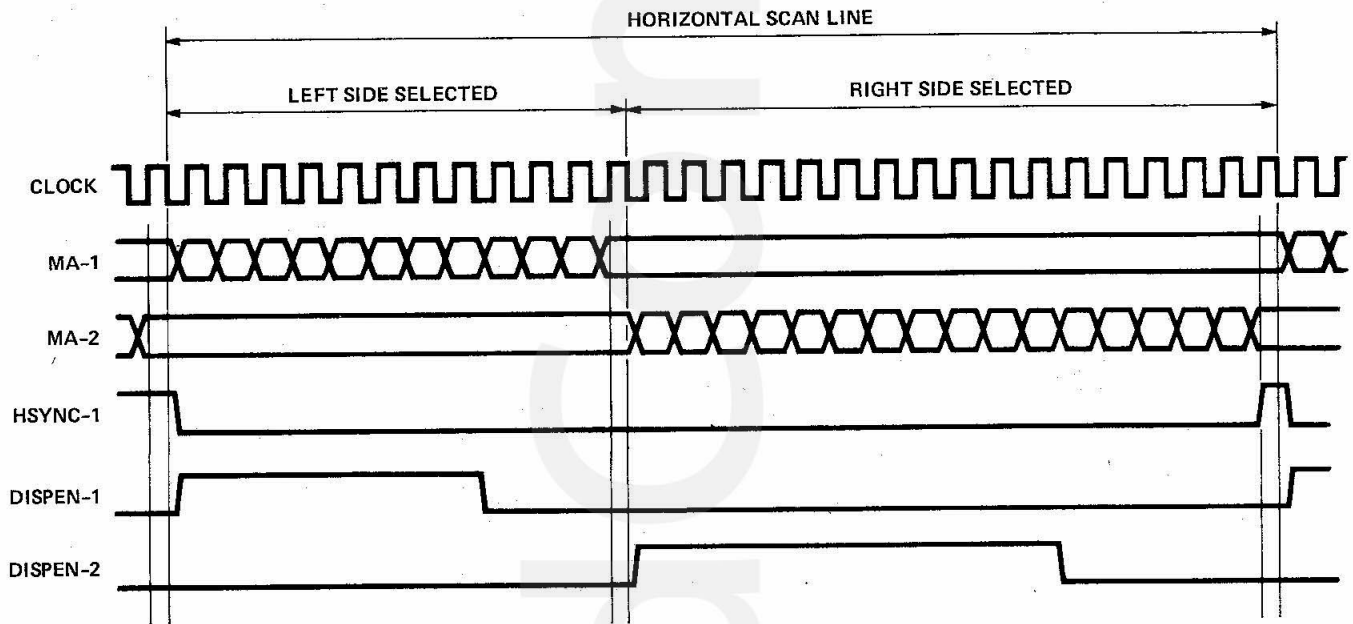


Figure 30. Dual-6545 Timing for Split-Screen Display

9.2 Logical Implementation

Figure 29 shows a simplified logic implementation for achieving split-screen capabilities.

Both 6545s should be identically programmed for basic screen parameters EXCEPT for R2 (horizontal sync position) and possibly for R1 (horizontal total displayed which would be a user option). This scheme depends on adjusting the horizontal sync position and synchronizing the two devices.

As a theoretical example let's assume the following requirements for a split-screen display.

1. 80 columns total split equally, 35 both left and right.
2. 10 blank columns separating the two sides.
3. A horizontal retrace pulse width of 10 CCLKs (columns).

These display parameters give us a horizontal total of 90. Therefore, R0 should program in both devices at 89. Now keeping in mind that adding to R2 shifts the display to the left of the screen and subtracting from R2 shifts it to the right, we can easily determine the R2 values for both devices. Device A in this example controls the left portion of the display, so we program its R2 with 80, forcing it all the way to the left. This means that device A's DISPEN output will be active for 35 columns starting at the left. 45 columns later HSYNC becomes active initiating the retrace of the beam. Device B's R2 should now be programmed for 35. This means that device B's DISPEN output will not be active for the first 45 columns of the horizontal sweep, providing the blanking between screens.

Once both devices' parameters have been loaded the devices must be synchronously reset. At this point they will be in sync and should need no further adjustments.

NOTES:

1. Device A DISPEN controls which memories datalines are switched to the character generator ROM.
2. To simplify implementation in this example the RA lines from device A only are used to generate scan line addresses.
3. Cursors for both screens are provided from their respective controllers.
4. Neither of device B's HSYNC or VSYNC outputs is required.
5. Memory contention circuitry was not detailed for this example only to reduce the complexity of the diagrams. However, the Transparent modes would be ideal for this complex task.

9.3 Summary

The scheme presented here may be used to accommodate general left-right split-screen displays. Details have been necessarily omitted in the interest of focusing on the fundamental concepts and techniques.

10.0 Operating at Very Fast Character Clock Rates

In some CRT system applications, it may be necessary to operate at character clock (CCLK) rates in excess of the maximum rate specified for the SY6545 CRT Controller. This section outlines a technique whereby the SY6545 is operated at a speed of one-half the system rate. Thus, the effective character clock rate can be doubled.

10.1 Basic Concept

The method used to achieve this is quite simple. It consists of a configuration wherein the SY6545 accesses

two characters in each cycle. The system timing must be arranged to select the two accessed characters one-at-a-time to be displayed. Figure 31 illustrates the system architecture necessary for this scheme.

10.2 Hardware Details

The video display memory must be organized in two separate blocks containing alternate, odd and even, character locations. In the example of Figure 31, several assumptions are made:

1. The size of the video display memory is 2k bytes. This is enough for a single page of 80x24 display (1920 locations), but is not limited to this size.
2. The memory contention scheme shown permits the MPU to have overriding priority in selecting the video display memory. Other schemes may be implemented as well, but this one is the simplest and allows attention to be focused on the more significant aspects of this method.
3. The video display memory is 8-bits wide. Normal ASCII coding only requires 7 bits, but 8-bits is used in this example.
4. Four raster code lines (RA0-RA3) are shown, indicating that more than 8 scan lines per character row (including row spacing) are used. This is also a system function and is not a limitation imposed on this scheme.
5. 8 horizontal bits per character (including spacing) are also assumed in the example.

A key point to note is the video display memory addressing arrangement. The SY6545 address lines, MA0-MA9, are routed through the address selector block to the memory addresses, A0-A9, of both even and odd memory blocks. In this way, as the SY6545 sequences through its count states, the two memories are accessed simultaneously. The MPU addresses, A1-A10, on the other hand, are routed to these same memory addresses, A0-A9. MPU address, A0, is used to select either the even memory (A0=0) or the odd (A0=1), but not both.

In this way, then, the video display memory is configured as a dual-port memory but, more significantly, each port sees the memory with a different organization. The SY6545 accesses the memory as a 1kx16 and the MPU, as a 2kx8.

10.3 Memory Control Logic

The video display memory control signals must function in the following fashion:

1. The chip selects, $\overline{CS}$, must always both be low (select) when the MPU is not addressing the memory. In this way, the SY6545 will be accessing both memory blocks.
2. When the MPU selects the memory, A11-A15 will correspond to the desired memory map decode combination. A0 will select either the even or the odd block. Thus, one $\overline{CS}$ will be low and the other high (de-select).
3. When A11-A15 correspond to the memory map decode, the address selector must also be switched to drive the memory addresses from the MPU address lines, instead of the SY6545 MA lines.

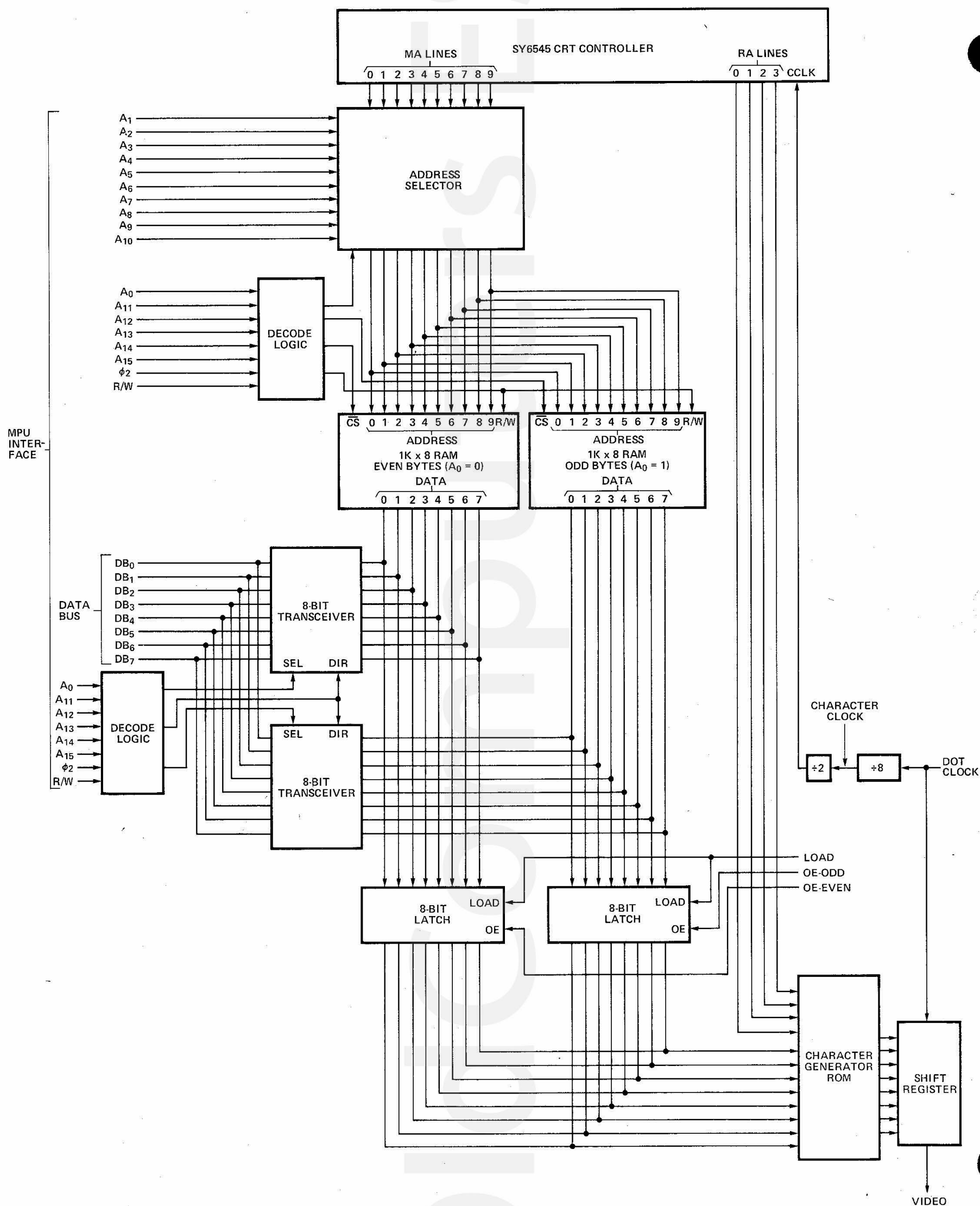


Figure 31. System Block Diagram

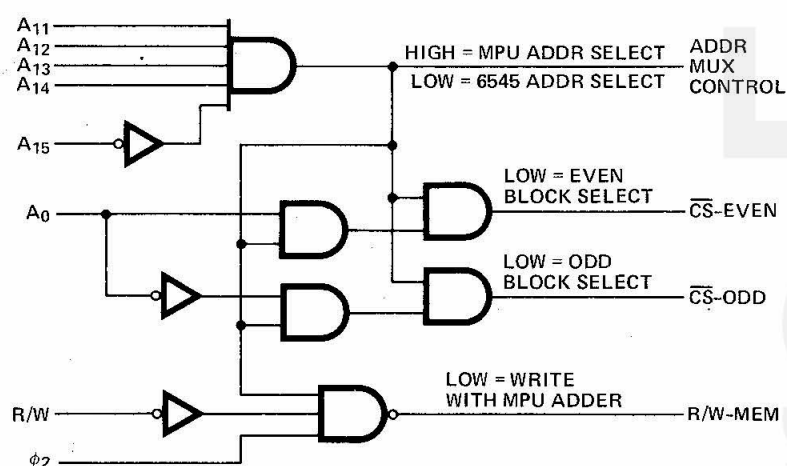


Figure 32. Video Display Memory Control Logic

4. Finally, the video display memory R/W line must be driven by the MPU R/W line, but only when the correct A11-A15 is present.

Figure 32 shows the memory control logic for the example of the video display memory block to be selected with A15 low and A11 through A14 high.

10.4 Data Bus Transceiver Control Logic

The control logic for the odd and even data bus transceivers has similar characteristics to that required for the video display memory.

1. The transceiver selects must also be energized when A11-A15 correspond to the memory map decode. Furthermore, A0 determines whether the odd or even memory block is selected.
2. When A11-A15 do not match the decode pattern, then the transceivers must be unselected and the MPU data bus must be isolated from the video display memory.
3. The MPU R/W line state determines the direction of the data through the transceivers.

Figure 33 illustrates the necessary logic for the example presented in Figure 31.

10.5 System Timing

Figure 34 shows illustrative system timing which further clarifies the detailed operation of this method. The dot clock is the timing signal which is used to drive the shift register for actual video generation. For this example, eight horizontal bits are required for each character. Thus, the dot clock must be divided down to derive the character clock signal. In a conventional system arrangement, this would then be used to drive the SY6545 CCLK pin. However, since this configuration calls for the SY6545 to fetch two characters at a time, the character clock must be halved to derive the CCLK input in the SY6545. In this way, two character clocks generate one CCLK which, in turn, causes two consecutive characters to be fetched, simultaneously, from the video display memory.

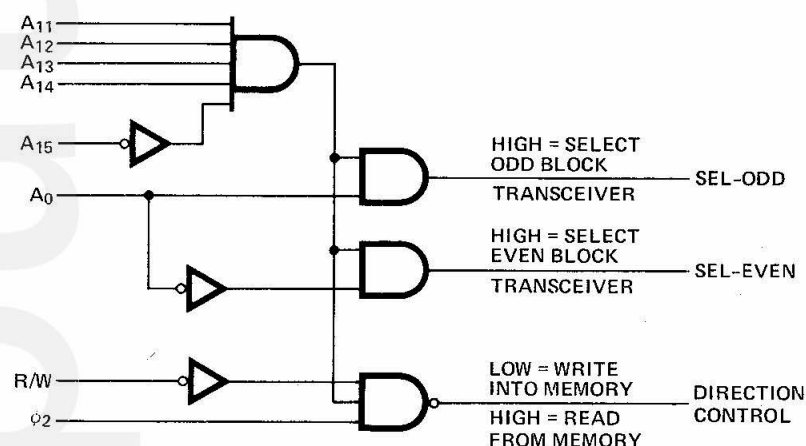


Figure 33. Data Bus Transceiver Control Logic

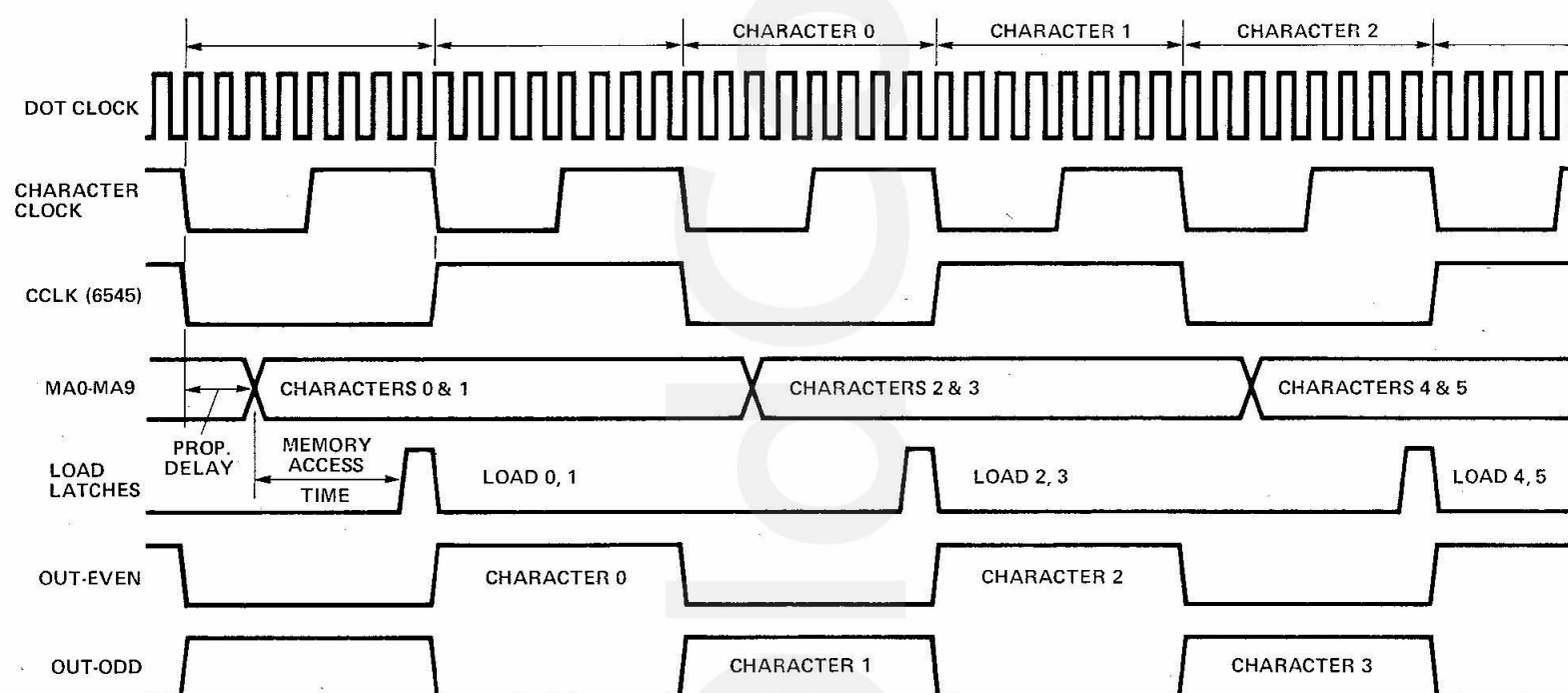


Figure 34. System Timings

The remaining signals shown in Figure 34 illustrate the approximate timing requirements for the load and output gating signals needed to operate the two 8-bit holding registers for the odd and even memory blocks. Note that the LOAD signal causes the video display memory outputs (odd and even characters) to be loaded into the latches and separate odd and even output controls alternately select the latches to drive the input lines of the character generator ROM.

Propagation delays through each functional section must be carefully accounted for. Figure 34 is primarily intended to demonstrate the alternating nature of the odd and even memory block timing control.

10.6 Programming Considerations

Special considerations must be made to accommodate the effects of this technique on the register programming required by the SY6545. The following registers are affected:

1. Horizontal Total.
2. Horizontal Displayed.
3. Horizontal Sync Position.
4. Horizontal/Vertical Sync Width.

In each case, the programmed value must be half the desired result. For example, to achieve 80 horizontal displayed characters per line, 40 must be programmed. Thus, it is seen that only even numbers of horizontal characters may be selected. This limitation is normally not significant, however, since standard displays require even numbers of columns.

10.7 Cursor Effect

There is one other noteworthy effect of this method. That is, the cursor output signal from the SY6545 will be two characters in duration, the odd and even character of the address selected by the cursor registers in the SY6545. To achieve a single character cursor, external logic must be incorporated.

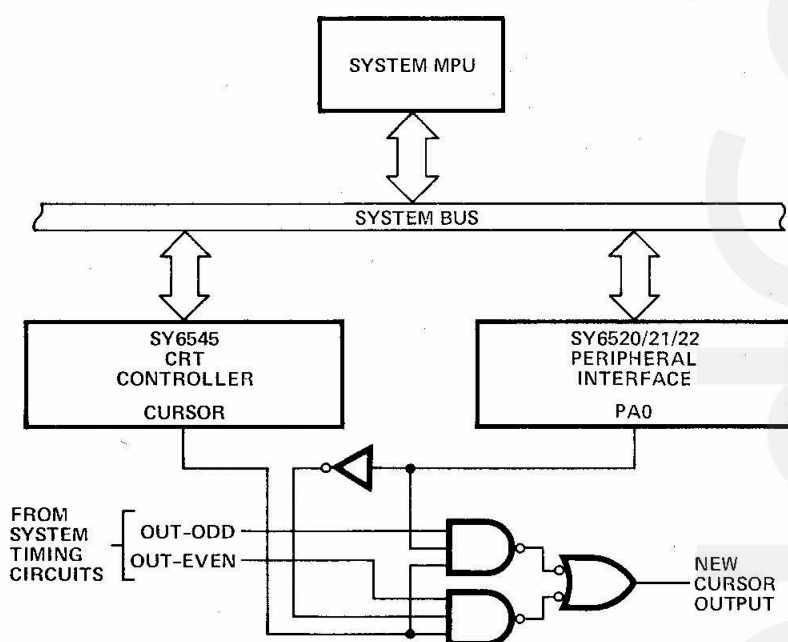


Figure 35. Cursor Gating Logic.

A simple technique is illustrated in Figure 35. In this approach, the cursor output from the SY6545 is gated with either the ODD or the EVEN latch select signal (shown in Figures 31 and 34, depending on the state of the peripheral pin, PA0). Thus, when the cursor address is set in the SY6545, the PA0 bit in the peripheral interface device must be set to either a "1", to select the odd character, or a "0", for the even. Other solutions are also possible, but this one is shown for demonstration purposes.

10.8 Summary

A technique has been presented which enables the SY6545 CRT Controller to be utilized in applications requiring higher character rates than are specified for the device. This particular method effectively doubles the character rate with relatively minimal hardware and software impact. Extensions to this approach may be utilized for even faster systems, using the same basic method.

11.0 Dynamically Changing Screen Display Formats

The SY6545 CRT Controller permits fully-programmable video screen display formatting. The screen parameters which are programmable are:

- Horizontal Sync Position and Width
 - This permits positioning the displayed text on the screen in the horizontal direction
- Vertical Sync Position and Width
 - This permits positioning the displayed text in the vertical direction
- Horizontal Total Characters and Displayed Characters
 - This permits selection of the number of displayed columns in each row
- Vertical Total Characters and Displayed Characters
 - This permits selection of the number of displayed rows on the screen
- Scan Lines per Row
 - This allows selection of the size of each displayed character — the height, including spacing

It is possible to alter these screen display parameters, within the confines of a specific hardware configuration. Care must be taken, however, to ensure that only those parameters whose alteration will still function with the hardware of the system, are allowed to change. For example, if the character clock (CCLK) rate of the system is fixed, it is important to change the display formats in a way which will not affect the overall frame rate (normally 60 Hz). Typically, for a fixed hardware design, the display formats which can be changed are:

- Horizontal Sync position
- Vertical Sync position
- Horizontal Displayed Characters

11.1 How to Change Parameters Dynamically

To change screen display formats dynamically means to re-program the SY6545 on-the-fly; that is, to re-program without disturbing the display. The SY6545 provides for

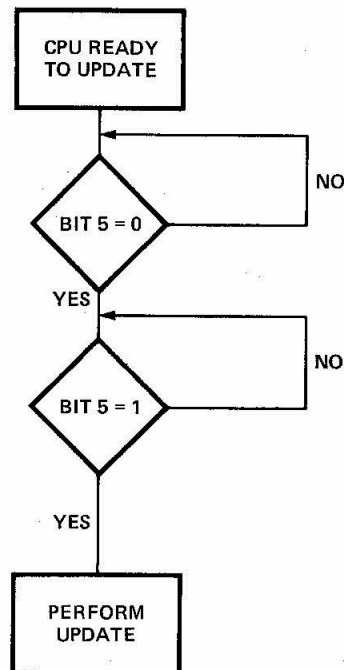


Figure 36.

dynamically changing screen formats without causing any disturbances on the screen.

The vertical blanking period is generally in the range of 2 to 3 msec, providing ample time to change all screen parameters. By monitoring bit 5 in the status register a user can take full advantage of this non-display period. The sequence should be as shown in Figure 36 so as to avoid starting an update at the end of the blanking period.

12.0 Using LPEN as Keyboard Scan Strobe

Much has been said and written about the impact of microprocessor technology on system design. It is clear that microprocessors not only permit existing systems to be implemented more cost-effectively, but that new systems, previously impractical, may now be very feasible and marketable. Two more recent developments extend this idea even further. They are single-chip microcomputers and dedicated controllers.

Single-chip microcomputers extend the impact of microprocessors toward the low-end of system applications. The characteristics which lend themselves more effectively to this technology are small size, low cost, relatively simple tasks, and high production volume.

The high-end of the spectrum is supported by new programmable dedicated controllers. Included in this category are devices such as floppy disk controllers, keyboard encoders, communications controllers, IEEE-488 bus controllers, DMA controllers, and interrupt controllers. In perspective, the main advantages to be gained by these devices are:

1. Low parts count for small system size.
2. Lower system cost.
3. High level of controller autonomy — microprocessor is relieved of detailed operational control
4. Flexible system design gained by programmability.
5. Inventory simplification by requiring only one device for several different system designs.

The area of CRT controllers is especially interesting. The evolution from teletype terminals to video terminals has been achieved in part by the utilization of microprocessor-based designs. The next steps in evolution will be towards even more sophistication and lower prices and will come about with the inclusion of dedicated MOS/LSI CRT controller devices into CRT system design, which already include microprocessor technology.

The wide variety of CRT controllers becoming available permits system design to be specifically tailored to the capabilities desired in the system. In some cases, clever use of the particular CRT controller can result in additional system features not foreseen when the controller was originally conceived. In this case, the CRT system can gain a competitive advantage if one or more of these capabilities is exploited. The remainder of this discussion focuses on just such a utilization, namely with the Synertek SY6545 device.

12.1 The SY6545 CRT Controller

The SY6545 is a microprocessor peripheral device intended to interface 6500/6800 microprocessors in CRT control applications. It is fully programmable, permitting a wide variety of screen display formats to be easily implemented.

Included in the SY6545 are counter outputs for memory addressing and a strobe input for light pen applications. The very nature of these pin functions allows these lines to be used as keyboard drive and sense signals, without affecting normal CRT operation. In other words, the SY6545 can be used simultaneously as a CRT Controller and a keyboard scanner. This note explains how this can be achieved.

12.2 Basic CRT Control Operation

Figure 37 shows a simplified block diagram of a microprocessor-based CRT Control system.

Note that the CRT Controller works in conjunction with a Video Display RAM (for screen display information storage), a character generator ROM (to generate dot patterns for each character) and video control circuits. All these components are required to keep the screen display "alive"; the data must be repetitively generated as video inputs to the CRT monitor or TV-type display. As a result, the address lines (MA0-MA13) are constantly counting from the start address (address of the first character to be displayed) to the end address (last character displayed). For a display size of 80 columns by 24 rows, this is 1920 characters, or a block of 1920 addresses, start to finish.

12.3 LPEN Input Strobe

The LPEN input pin on the SY6545 is intended as a strobe input for light pen applications. Internally, a signal on this pin gates the scanning counter state (MA0-MA13) into a light pen holding register. In this way, when the CRT electron beam passes across the light pen held at the surface of the video screen, a signal generated by the light pen and applied to the LPEN input to the SY6545 will effectively identify the location of the light pen, relative to the scanning addresses. In most CRT systems, there is no light pen capability required. Thus, there is very little system impact if this feature is used as a keyboard strobe, instead of its intended light pen function.

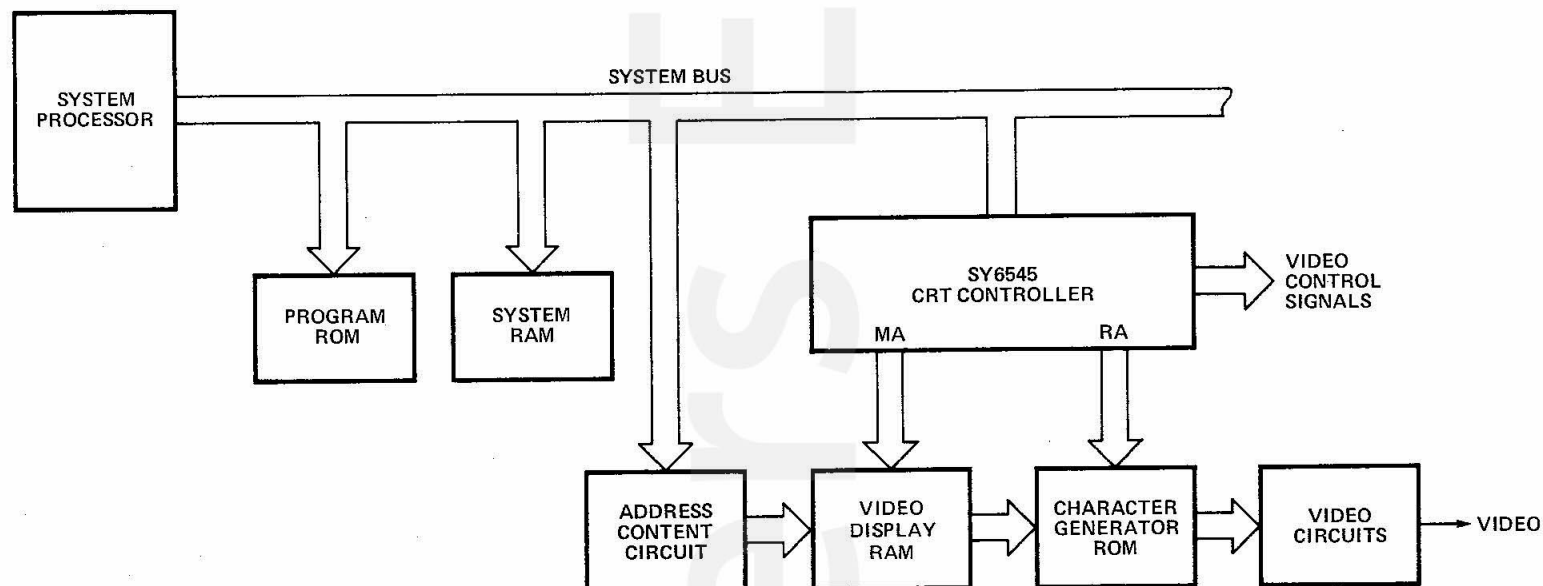


Figure 37. MPU-Based CRT System

12.4 A Simple Keyboard Scanning Configuration

Figure 38 illustrates the hardware configuration of the SY6545 with an 8x8 crosspoint keyboard matrix.

Note that the MA lines are utilized to scan the keyboard, but that their operation with the screen display RAM is totally unaffected. Using these lines to drive the keyboard requires no special programming steps, only a 1-of-8 decoder (active-low outputs) for driving the keyboard lines and 1-of-8 data selector (active-high output) to sense the keyboard and drive the LPEN strobe pin of the SY6545. However, in this case the LPEN line is not available for light pen applications, but as indicated previously, this is not a significant drawback.

12.5 Keyboard Strobe Timing

The MA0-MA13 addresses are generated as a result of the SY6545 CCLK (character clock) input. This signal occurs once every character time and automatically increments the scan counter for the MA lines. The closure of a key will have no effect until the specific drive line is energized and the specific sense line is gated to the LPEN input. At this time, the LPEN strobe occurs and the MA0-MA13 counter inside the SY6545 is loaded into the LPEN holding register, also inside the SY6545. The timing is shown in Figure 39.

All timing is very clean and not sensitive to race conditions or transients on any signals.

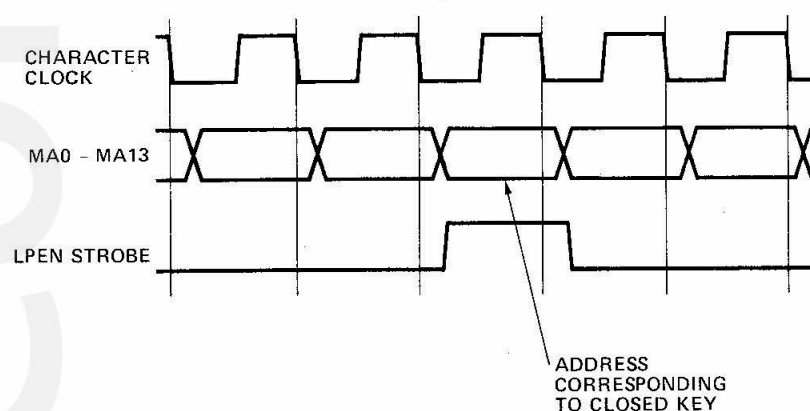
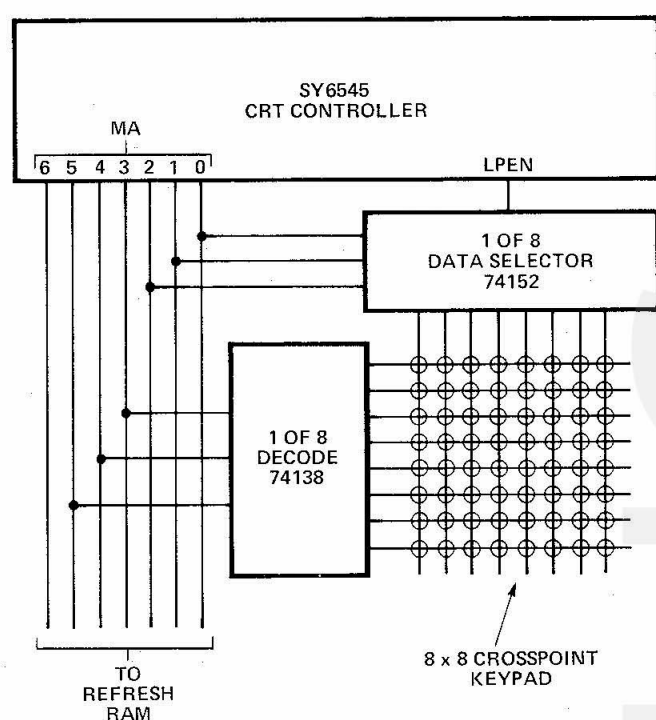


Figure 39. LPEN Strobe Timing for Keyboard Scan

Figure 38. Keyboard Scan Circuits for Use with SY6545 CRTC

12.6 Programming Considerations for Key Identification

For this example, six MA lines are used.

- MA0-MA2 identify the sense line for the key closed.
- MA3-MA5 identify the drive line of the key closed.

However, the SY6545 LPEN strobe will actually cause the next sequential address to be loaded into the internal holding register, due to internal timing skews in the SY6545. Thus, it is important to decrement the number stored into the holding register before decoding it to determine the key closed.

For example, consider the key closed as shown in Figure 40.

For this case, the MA lines which cause the LPEN strobe are,

7	6	5	4	3	2	1	0
—	—	1	1	0	1	0	1
DRIVE LINE 6				SENSE LINE 5			

However, the contents of the internal LPEN register will be,

7	6	5	4	3	2	1	0
—	—	1	1	0	1	1	0

This is easily corrected in the system program by decrementing after the register is read out.

12.7 Use of the LPEN Status Bit in the SY6545

A light pen status indication is available in the SY6545 to identify that a strobe has occurred. This is used by the system processor as a means of determining that a key has been pressed and the LPEN register may be interrogated. This status bit is not usable with the interrupt facility of the microprocessor, so the processor must periodically scan the status register to check for key closure. This can be done either by hardware timers to initiate the processor status check or simply by programming periodic checks to occur.

12.8 De-Bounce Considerations

Key de-bounce must be handled by software, as well. Once a key is determined to be down, several iterations of checking the LPEN status bit and holding register must be made to verify that it is indeed down, and not simply caused by system noise. Likewise, key release must be determined over several iterations, as well.

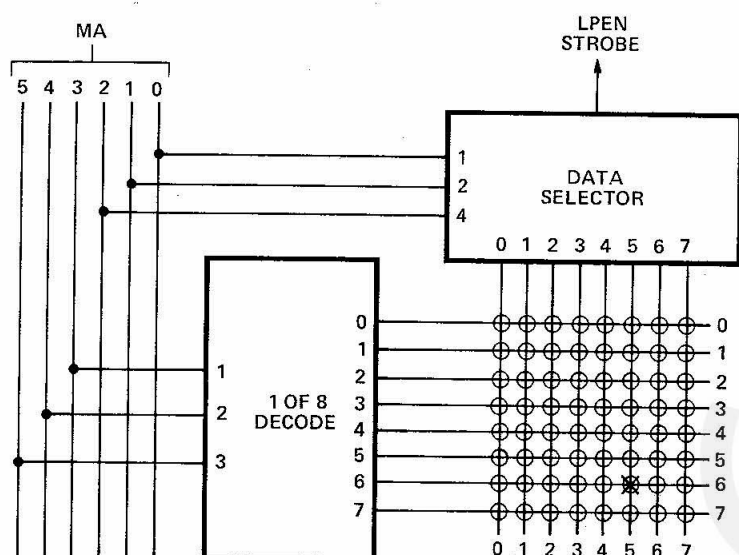


Figure 40. Key Closure Identification

ES s r s t e r s c o m p u t e r s

Synertek

COPYRIGHT 1982 BY SYNERTEK

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photo copying, recording, or otherwise, without the prior written permission of Synertek.

The information contained herein is subject to change without notice. Synertek assumes no responsibility for the use of any circuitry other than circuitry embodied in a Synertek product.

No other circuit patent licenses are implied.

SYNERTEK INC. • A SUBSIDIARY OF HONEYWELL, 3001 STENDER WAY, SANTA CLARA, CA 95054
TELEPHONE (408) 988-5600 • TWX: 910-388-0135

U1056 B-15K 7/82