82730 TEXT COPROCESSOR

- **High Quality Display for Text and Graphics Applications**
- **Provides Proportional Spacing, Simultaneous Superscript/Subscript, Soft Font Support and Bit Map Graphics**
- **High Performance Manipulation of Text/Graphics Strings**
- **Programmable Bus Interface Handles 8 or 16 Bit Data and 16 or 32 Bit Addressing; iAPX 86/88/186/188 Compatible**
- **On-Chip Processing Unit Simplifies Software Design by Executing High Level Commands and Supporting Linked List Data Structures**
- **Extremely Flexible; Programmable Features Include Screen and Row Formats, Two Cursors, Character and Field Attributes and Smooth Scrolling**
- **Supports Multiple Windows**
- **High Resolution Display; Up to 200 Characters/Graphics Cells per Row and 2048 Scan Lines per Frame**
- **Separate Bus and Video Clocks Allow Optimization of Overall System Performance**
- **Provides a Complete LSI Solution for Display Control when Used in Conjunction with the 82731 Video Interface Controller**
- **68 Pin JEDEC Package: (See Intel packaging: Order Number: 210931-004)**

The 82730 Text Coprocessor is a high performance VLSI solution for raster scan text and graphics displays. The 82730 works as a coprocessor and has processing capabilities specifically tailored to execute data manipulation and display tasks. It provides the designer the ability to functionally partition his system thereby offloading the system CPU and achieving maximum performance through concurrent processing. The 82730 supports the generation of high quality text displays through features like proportional spacing, simultaneous superscript/subscript, dynamically reloadable fonts and user programmable field and character attributes. It supports high quality graphics with fast manipulation and display of bit map strings. An intelligent system interface and efficient software capabilities makes 82730 based systems easy to design.

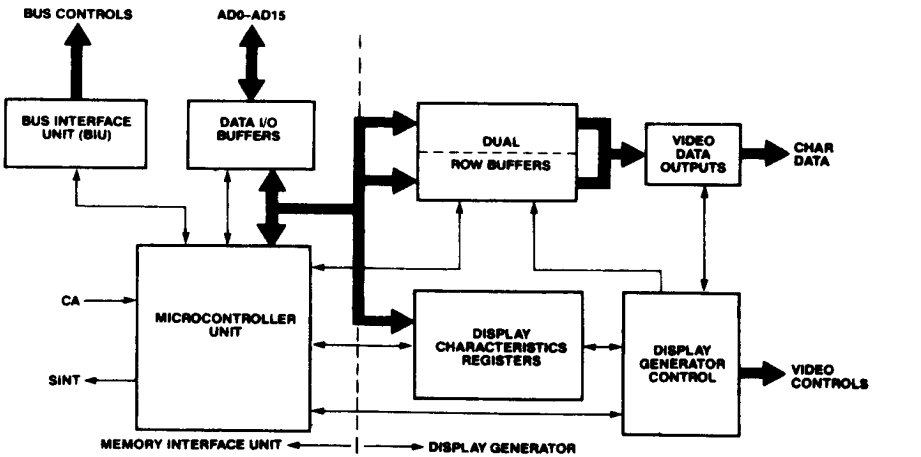


Figure 1. 82730 Block Diagram

November 1986
Order Number: 210931-004

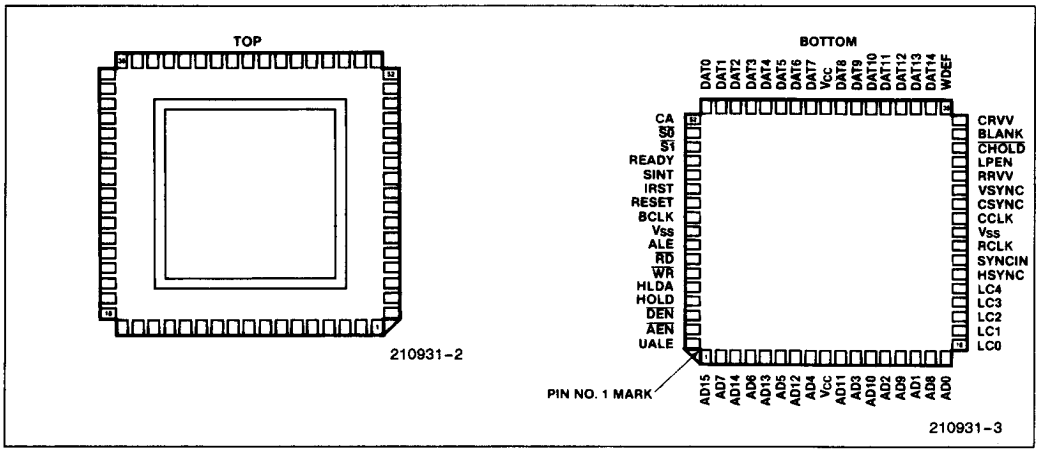


Figure 2. 82730 Pinout Diagram

Table 1. 82730 Pin Description

The 82730 is packaged in a 68 pin JEDEC Type A ceramic package.

Symbol	Pin Number	Type	Name and Function
AD15–AD0	1–8 10–17	I/O	**ADDRESS DATA BUS:** these lines output the time multiplexed address (TU, T1 states) and data (T2, T3, T4 and TW) bus. The bus is active HIGH and floats to 3-state OFF when the 82730 is not driving the bus (i.e., HOLD is not active or when HOLD is active but not acknowledged, or when RESET is active).
BCLK	59	I	**BUS CLOCK:** provides the basic timing for the Memory Interface Unit.
$\overline{RD}$	62	O	**READ STROBE:** indicates that the 82730 is performing a memory read cycle on the bus. $\overline{RD}$ is active low for T2, T3 and TW of any read cycle and is guaranteed to remain high in T2 until the address is removed from the bus. $\overline{RD}$ is active low and floats to 3-state OFF when 82730 is not driving the bus. $\overline{RD}$ will return high before entering the float state and will not glitch low when entering or leaving float.
$\overline{WR}$	63	O	**WRITE STROBE:** indicates that the data on the bus is to be written in a memory device. $\overline{WR}$ is active for T2, T3 and TW of any write cycle. It is active LOW and floats when 82730 is not driving the bus. $\overline{WR}$ will return high before entering the float state and will not glitch low when entering or leaving float.
ALE	61	O	**LOWER ADDRESS LATCH ENABLE:** provided by the 82730 to latch the address into an external address latch such as 8282/8283 (active HIGH). Addresses are guaranteed to be valid on the trailing edge of ALE.
UALE	68	O	**UPPER ADDRESS LATCH ENABLE:** it is similar to ALE except that it occurs in upper address output cycle (TU).
$\overline{AEN}$	67	O	**ADDRESS ENABLE:** $\overline{AEN}$ is active LOW during the entire period when 82730 is driving the bus. It can be used to unfloat the outputs of the Upper and Lower Address latches.

Table 1. 82730 Pin Description (Continued)

Symbol	Pin Number	Type	Name and Function
$\overline{DEN}$	66	O	**DATA ENABLE:** provided as a data bus transceiver output enable for transceivers like the 8286/8287. $\overline{DEN}$ is active LOW during each bus cycle and floats when 82730 is not driving the bus. $\overline{DEN}$ will not glitch when entering or leaving the float state.
$\overline{S0}$, $\overline{S1}$	53, 54	O	**STATUS PINS:** encoded to provide bus-transaction information: $\overline{S1}$ $\overline{S0}$ — Bus Cycle Initiated 0 0 — - - - (Reserved) 0 1 — Memory Read 1 0 — Memory Write 1 1 — Passive (No Bus Cycle) These pins are directly compatible with iAPX 86, 186 status outputs $\overline{S1}$ and $\overline{S0}$. The status pins are floated when 82730 is not driving the bus. They will not glitch when entering or leaving the 3-state condition.
READY	55	I	**READY:** signal to inform the 82730 that the data transfer can be completed. Immediately after RESET, READY is asynchronous (internally synchronized) but can be programmed during initialization to bus synchronous.
HOLD	65	O	**HOLD:** indicates that the 82730 wants bus access. HOLD stays active HIGH during the entire period when 82730 is driving the bus.
HLDA	64	I	**HOLD ACKNOWLEDGE:** indicates to 82730 that it is granted the bus access as requested. HLDA may be asynchronous to 82730 clock. If HLDA goes inactive (LOW) in the middle of an 82730 bus cycle, the 82730 will complete the current bus cycle first, then it will drop HOLD and float address and bus control outputs.
CA	52	I	**CHANNEL ATTENTION:** used to notify 82730 that a command in the command block is waiting to be processed. CA is latched on its falling edge.
SINT	56	O	**STATUS INTERRUPT:** used to inform the processor that an unmasked interrupt has been generated in the 82730 status register.
IRST	57	I	**INTERRUPT RESET:** SINT is cleared by activating the IRST pin.
RESET	58	I	**RESET:** causes 82730 to immediately terminate its present activity and enter a dormant state. The signal must be active HIGH for at least 4 BCLK cycles and is internally synchronized to the bus clock.
CCLK	27	I	**CHARACTER CLOCK:** input used to clock row buffer data, attribute, cursor and line count out of 82730. When more than one 82730 is connected in cluster mode, CCLK is used to synchronize output from both master and slave chips. A character data word will be output at every rising edge of CCLK.
RCLK	25	I	**REFERENCE CLOCK:** input used to generate timings for the screen layout and to define screen columns for data formatting. All raster output signals are specified relative to the rising edge of RCLK.
DAT0–DAT14	36–42 44–51	O	**VIDEO DATA BUS OUTPUT:** the least significant 15 bits of the character data words are passed through the 82730 row buffer and made available on the pins DAT0–DAT14. The user has the flexibility to partition the data word into character and attribute bits per his requirements. The bits that are assigned for internally generated attributes may also be available at pin DAT0–DAT14. New character data will be shifted to these output pins at every rising edge of the CCLK. Together with LC0–LC4, they may be used to address the character generator or as attribute controls.

Table 1: 82730 Pin Description (Continued)

Symbol	Pin Number	Type	Name and Function
WDEF	35	O	**WIDTH DEFEAT:** is used to indicate when the character is allowed to be a variable width or must be of fixed width. WDEF is LOW if the character being output is normal, but is HIGH if it is a superscript/ subscript character or visible attribute (TAB or GPA). Optionally, WDEF can be held high by user command.
LC0–LC4	18–22	O	**LINE COUNT OUTPUTS:** used to address the character generator for the line positions in a row. The line number output is a function of the display mode and character attributes programmed by the user.
CSYNC	28	O	**CCLK SYNCHRONIZATION OUTPUT:** used to synchronize external character clock generator to reference clock timing. This output is active (high) outside the display field.
$\overline{\text{CHOLD}}$	32	O	**CCLK INHIBIT OUTPUT:** used by external logic to inhibit CCLK generation. This output is active (low) during the tab and end-of-row function.
SYNCIN	24	I	**SYNCHRONIZATION INPUT:** used to synchronize the vertical timing counters to an externally generated VSYNC signal. Used by slave mode 82730 to synchronize to a master mode 82730 and by the master 82730 to lock the frame to an external source such as the power line frequency.
HSYNC	23	O (MASTER) I (SLAVE)	**HORIZONTAL SYNC:** in master mode, it is used to generate the CRT monitor's horizontal sync signal. It is active HIGH during the programmed horizontal sync interval. In interlace slave mode it is used in conjunction with SYNCIN to indicate the start of the even field for timing counter reset. At RESET, pin is set as an output in the LOW state.
VSYNC	29	O	**VERTICAL SYNC:** active HIGH during the programmed vertical sync interval and used to generate the CRT monitor's vertical sync signal.
BLANK	33	O	**BLANKING OUTPUT:** used to suppress the video signal to the CRT. BLANK is clocked by CCLK.
CRVV	34	O	**CHARACTER REVERSE VIDEO (CCLK OUTPUT):** used to externally invert video data output. CRVV is clocked by CCLK.
RRVV	30	O	**REFERENCE REVERSE VIDEO (RCLK OUTPUT):** to externally invert video in the field and border area if so programmed by user. It is LOW outside the border area, RRVV is clocked by RCLK.
LPEN	31	I	**LIGHT PEN INPUT:** used to latch the position of a light pen. At the rising edge of this input, the column position and the row position of the 82730 will be loaded into the LPENROW and LPENCOL locations in the Command block.
V_{CC}	9, 43		**POWER:** +5V nominal potential.
V_{SS}	26, 60		**POWER:** ground potential.

FUNCTIONAL DESCRIPTION

Figure 1 shows a basic block diagram of the 82730 Text Coprocessor. The chip is divided into two main sections, the Memory Interface Unit and the Display Generator. The Memory Interface Unit controls fetching of the data and commands and handles interrupts and status. The Display Generator takes the data fetched by the Memory Interface Unit and presents it to the Video Interface logic which in turn drives the CRT monitor.

Memory Interface Unit

The Memory Interface Unit is divided into two sections: The Bus Interface Unit and the Microcontroller Unit. The Bus Interface Unit does the acutal interfacing to the memory bus. It fetches or writes data under the control of the Microcontroller Unit. The Microcontroller Unit is a microprogrammed controller which is designed to efficiently fetch data from memory (up to 4 Mbytes/sec), and decode and execute various control and data handling commands. The Bus Interface Unit may be configured for 8 or 16 bit bus operation. With 8 bit bus selection, the user may specify either 8 or 16 bit character data. It also handles address manipulation automatically after being loaded from the Microcontroller Unit.

Display Generator

The Display Generator takes the data fetched from memory plus the modes programmed into it at initialization and produces all the video timing and the data transfers to support the CRT monitor at the character level. The 82730 works with an external character generator and the 82731 Video Interface Controller. The data is passed to the Display Generator from the Memory Interface Unit through the dual row buffers (similar in operation to the one in the 8275 CRT controller). The row buffers allow the user to use cheaper and slower main memory for display needs, provide on-chip attribute and display function generation, and avoid the conflict of access to the display memory (that would otherwise take place) by using an ordinary DMA access mechanism.

SYSTEM BUS INTERFACE

The Memory Interface Unit provides communication with system processor as well as memory interactions. Communication between the processor and the 82730 is performed via messages placed in communication blocks in shared memory. The processor can issue commands by preparing message blocks and directing the 82730's attention to them by asserting a hardware channel attention. The 82730 can cause interrupts on certain conditions, if enabled by the processor by activating its System Interrupt output, with status and error reporting taking place through the communication block in memory.

BUS INTERFACE UNIT

The 82730 Bus Interface Unit provides an 8086 compatible bus interface which consists of:

- a 16/32 bit multiplexed Address/Data Bus: AD_0–AD_{15}
- A complete set of local bus control signals compatible with 8086 min mode: $\overline{RD}$, $\overline{WR}$, ALE, $\overline{DEN}$, and READY
- Two status signals $\overline{S0}$ and $\overline{S1}$, compatible with 8086 max mode so that a bus controller (8288) can be shared for Multibus® access.
- Local bus arbitration through HOLD/HLDA
- Two upper Address Latch controls: UALE and $\overline{AEN}$

The BUS INTERFACE UNIT (BIU) utilizes the same Bus structure as the 80186 or basically the same bus structure as the 8086 in both Min. and Max. mode, (with the exception of RQ/GT) and it performs a bus cycle only on demand (e.g., to fetch a command from the command block, or fetch a character from display data memory). The same set of T-states (T1, T2, T3, T4 and TW) of 8086 are used to handle the time multiplexed address/data bus. However, adaptations are made to handle 32 bit addresses as explained in the following sections where specific details of the BIU operation are described. Those details not mentioned can be assumed to be the same as those of the 80186.

ADDRESS BUS

The 82730 can be programmed during initialization to operate on either 16 bit or 32 bit (including any length between 17 and 32) physical addresses. Note that the 82730 does not use memory segmentation. The programmer must calculate physical addresses from segment and offset values to manipulate data structures.

To support 32 bit physical addresses with a 16 bit physical bus, multiplexing is again used. An upper address output cycle, TU, is inserted between T4 and T1 to output the upper 16 bits of address. The upper address latch enable, UALE, is used to latch the upper addresses during TU. Figure 3 shows the configuration of a 32 bit address bus.

TU occurs only when the 32 bit mode is specified and the upper address register of BIU is reloaded by MCU. This may result from:

i) Initialization

ii) Manipulation of display data or command pointers, for example, when a new string pointer is loaded during the execution of the END OF STRING command.

iii) DMA address incrementing across a 64K byte segment boundary.

iv) Regaining the bus after losing it to a higher priority master.

Timing of UALE is identical to that of ALE. $\overline{AEN}$ is equivalent to the active period of 82730 driving the bus.

If 16 bit address mode is programmed, TU will never occur in any bus cycle since the MIU treats all display pointers as 16 bit quantities and loading of internal upper address register is bypassed during address calculation. UALE always stays inactive, but $\overline{AEN}$ still goes active to indicate the 82730 has control of the bus.

DATA BUS

The 82730 is capable of operating on either an 8 bit or a 16 bit Data bus, as programmed during initialization on the SYSBUS byte.

When an 8 bit data bus is specified, the address present on AD15 to AD8 Address/Data lines is maintained for the complete bus cycle. Therefore, compatibility with 80188, 8088, 8089 and 8085 multiplexed address peripherals is maintained. Since the internal processing of the 82730 generally operates on 16 bit data quantities, two Bus fetch cycles are performed for each 16 bit data item. The first cycle fetches the low order byte, the second cycle the high order byte. These 2 fetch cycles are always executed back to back. If HLDA drops during the first cycles, the 82730 will not respond until the second cycle is completed. An 8 bit data mode can be selected in an 8 bit bus system that requires only 8 bit character data be fetched.

In 16 bit bus system, the 82730 requires all 16 bit quantities to start on even address boundary. Word transfer to or from odd boundary is not allowed since this type of transfer not only doubles the use of bus bandwidth but also can be easily avoided in application software. All that is required is to make sure all address pointers be an even number (A0 = 0).

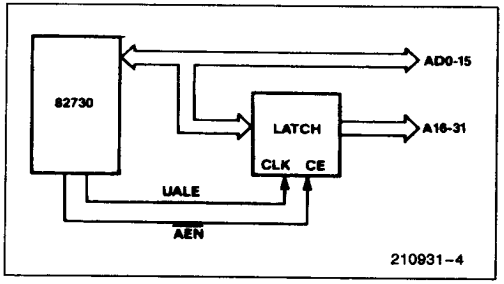


Figure 3. Address Extension up to 32 Bits

BUS CONTROLS

The 82730 BIU provides both the 8086 MIN. Mode (Local Bus Control) and MAX. mode bus control signals simultaneously in any bus cycle. By providing a complete set of Local Bus control signals, the component count of the Local processing module is minimized.

Because only two types of Bus operations, Memory Read and Memory Write, are executed in the 82730 BIU, the 8086's $\overline{S2}$ status signal is omitted from the Max. mode controls. S2 could be set to "1" during any 82730 Bus cycle. $\overline{AEN}$ can be used to produce $\overline{S2}$ since it stays active whenever 82730 is driving the bus. The status signals become valid at the middle of the cycle before T1 which could be either T4 or TU.

$\overline{BHE}$ is not provided on the 82730 because, the 82730 only writes words to even address boundaries and bytes to the upper byte position. For these writes $\overline{BHE}$ is always high. A pullup resistor or a three-state buffer controlled by $\overline{AEN}$, can provide this signal.

DT/R is also not provided on the 82730 because its function can be replaced with $\overline{S1}$, latched by ALE.

After RESET is applied, READY is set to be an asynchronous input. An on-board synchronization circuit provides reliable operation for any type of system. During initialization, READY may be programmed to be bus synchronous. For those systems that can meet the set-up time specifications, this mode provides more efficient bus utilization.

LOCAL BUS ARBITRATION

The 82730 BIU is designed to function as a bus master in a multimaster Local bus environment using the HOLD/HLDA protocol for Bus arbitration.

In the Self Contained Arbitration scheme, one processor and one 82730 share access to the local bus. The 82730 raises its HOLD request whenever it needs bus access. After HLDA is granted from the processor, the 82730 will not start driving the bus until 2 clock cycles later. This latency allows sufficient time for the 8086 or 80186 processor to get off the bus. When 82730 completes it bus accesses, it will first float its output drivers before dropping the hold request.

In a Local bus configuration with three or more bus masters, a higher priority DMA Peripheral device can preempt the HLDA from a 82730 which is the current bus master. The 82730 will complete its current bus cycle, then float its output drivers and drop the HOLD request. However, the 82730 may raise the HOLD request again 2 clock cycles later if it still needs the bus to complete the interrupted burst DMA activities.

DMA BURST AND SPACE

Some system configurations using the 82730 would be adversely affected by the long burst data transfers which the Memory Interface Unit (MIU) may occasionally desire. Since the 82730 will normally be configured as one of the higher priority bus masters, burst lengths must be limited for these systems. For this reason, the length of a burst transfer and the number of memory cycles between burst transfers are both programmable via the mode registers:

	15	14	8	7	6	0
MPTR	—	BRSTLEN		—	BRSTPAC	

BRSTLEN—Burst Length. Determines the number of contiguous word-fetch cycles which may be requested. Programmable from 1 to 127. Note that in an 8 bit bus, 16 bit data system, the burst counter only increments once for the 2 bus cycles required to complete a word fetch. (Note: burst length = 0 is not defined and should not be programmed with a non-zero burst space.)

BRSTSPAC—Burst Space. Determines a minimum number of bus clocks to occur between burst accesses. Programmable from 0–511 in increments of four. Zero space selects an infinite burst length.

A DMA burst could be terminated before the programmed burst length is reached in the following circumstances:

i) The MIU does not need any more bus accesses, for example, when the row buffer if filled.

ii) A datastream command is encountered and the MIU must execute the command first before it resumes data accessing.

iii) The bus is taken away by a higher priority device in multi-master bus configuration.

In these cases, the burst counter is cleared. The BIU must complete a full burst before it waits through the SPACE cycles. DMA Burst/Space will be set to zero space until the completion of the first MODESET command.

INITIALIZATION OF BIU

Upon activation of the RESET input, the 82730 BIU will stop all operations in progress and deactivate all outputs. It will stay in this quiescent state until memory access is requested by the MCU after MCU receives its first channel attention after RESET. The following table shows the state of all MIU outputs during and after reset.

Table 2. 82730 Bus During and After Reset

Signals	Condition
AD15–0	Three-state
$\overline{RD}$, $\overline{WR}$, $\overline{DEN}$	Driven to '1' then three-state
$\overline{S0}$, $\overline{S1}$	Driven to '1' then three-state
ALE, UALE	Low
$\overline{AEN}$	High
HOLD	Low
SINT	Low

82730 COMPATIBILITY ISSUES

82730 Bus Clock Compatibility

The 82730 uses the 50% duty cycle output of the iAPX-186 at 8 MHz or that generated by a clock generator such as the 82285. A different duty cycle clock may be used at lower frequencies, so the 82730 is also usable with the iAPX-86, 88 family.

82730 Bus Interface Compatibility

The bus interface compatibility between the 82730 and another bus master has four main issues: data bus width, addressability, control bus structure and local bus mastership arbitration.

Data Bus

Data Bus width compatibility with all 85/86 family processors (8085, 8086, 8088, 80188, 80186, and 80286) is being supported by the 8/16 data bit pro-

grammability already discussed. This allows interfacing to the above processors either directly or through a Multibus-like interface.

Address Bus

The 82730 uses real 32-bit addresses. The user's software must calculate real addresses; this general addressing scheme allows the 82730 to be used with any microprocessor.

Control Bus

The 82730 implements both 8086 minimum and maximum mode bus control structures. This was done to maximize compatibility with the 80186 which has the same structure. This allows the 82730 to be run locally (minimum mode) with a 8085, 8086, 8088, 80188, or 80186. The 80186/188 and 82730 can run together at 8 MHz because of clock duty cycle considerations. The 82730 can only communicate to an 80286 via a system bus (such as MULTIBUS), bus interface, or dual-port RAM.

INITIALIZATION SEQUENCE

The first CA (Channel Attention) after Reset causes an Initialization Sequence to be executed. The system processor must set up the appropriate initialization information in memory and set the BUSY flag in the Intermediate Block to a non-zero value prior to issuing this CA.

Initially, 32-bit addressing and 8-bit bus width are assumed until the corresponding information is fetched during the initialization. First the SYSBUS byte is fetched from memory location FFFF FFF6. (When the address bus is less than 32 bits wide, the higher order bits are unused.) The format for SYSBUS byte is shown in Figure 4 and is the same as that used for 8089. The data bus width is specified by the least significant bit w, with w = 0 indicating an 8-bit bus and w = 1 signifying a 16-bit bus.

A 32-bit real address pointer is then fetched from memory locations FFFF FFFC through FFFF FFFF, with lower bytes of the pointer residing in lower addresses. This pointer is used as an Intermediate Block Pointer (IBP).

The Intermediate Block Pointer (IPB) is incremented by two and is used to locate the Command Block Pointer (CBP). Four bytes are fetched irrespective of whether a 16-bit or 32-bit addressing option is used. The System Configuration byte (SCB) is then fetched from location (IBP + 6).

The least significant, (U of the SCB) specifies 16 or 32-bit addressing option, with U = 0 indicating 16 bit addressing and U = 1 specifying 32-bit addressing. The SCB also contains information about cluster operation. Since up to four 82730's can be connected in a cluster with their respective data interleaved in memory, cluster information is needed for the data access task. The SCB specifies Cluster Number (CL NO), which is the number of 82730's connected in a cluster and Cluster Position (CL POS) which is the position of this particular 82730 within the cluster. CL NO = 0, 1, 2 or 3 indicates a cluster containing 1, 2, 3 or 4 82730's respectively. Simlarly, CL POS = 0, 1, 2 or 3 indicates 1st, 2nd, 3rd or 4th position respectively. Each 82730 adds an offset equal to 2 * CLPOS to the SPTR fetched from memory and increments the pointer by 2 * (CL NO + 1). The programming of CL NO and CL POS is independent. No checking is done for CL POS greater than CL NO on the 82730. Note that at least one 82730, in a cluster (even if it is a cluster of one), must be assigned as cluster position zero (CL POS = 0) for Virtual Display mode to work properly.

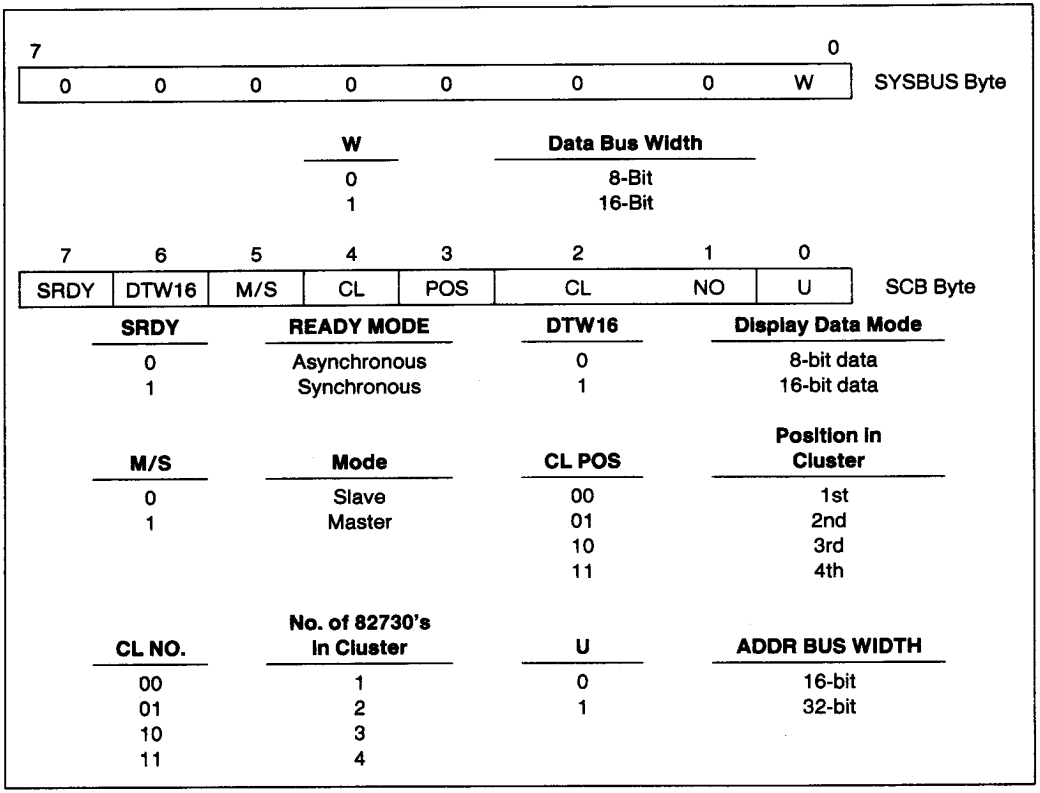


Figure 4. SYSBUS and SCB Encoding

The SCB also contains an M/$\overline{S}$ bit which specifies a master or slave mode. The M/$\overline{S}$ bit is stored internally for use by the Display Generator (DG). M/$\overline{S}$ = 1 indicates a master mode and M/$\overline{S}$ = 0 specifies a slave mode. The format for the System Configuration Byte (SCB) is shown in Figure 4. Following these actions, the BUSY flag in the Intermediate Block at address IBP is cleared and a normal Channel Attention sequence is then executed.

The last two bits in the SCB are DTW16 and SRDY. DTW16 specifies whether the display data in 8 bit bus mode (W = 0) is 8 or 16 bit. If a 16 bit system is specified (W = 1) then DTW16 is ignored and forced internally to a "one". SRDY specifies whether the clock synchronization circuit for the READY pin is internal (SRDY = 0) or external (SRDY = 1).

The Initialization Control Blocks in memory are illustrated in Figure 5a. How these fit into the control structure of the 82730 is shown in Figure 5b.

Channel Attention Sequence

When the processor activates CA, an internal latch in 82730 is set on the falling edge of CA input and this latch is sampled by the MCU. The first CA activation after reset causes the 82730 to execute an initialization sequence. Any subsequent activation will cause the MCU to start processing the command block by fetching a channel command.

If a display is in progress, the MCU will sample CA at each end of frame, otherwise it will sample CA every cycle until it is found active. When CA is found active, the MCU will fetch the command byte from "COMMAND" location in the command block, execute the command and clear the BUSY flag upon completion. The internal CA latch is also cleared by the MCU. An invalid command code has the effect of NOP and the BUSY flag is cleared. It will also cause the Reserved Channel Command (RCC) status bit to be set.

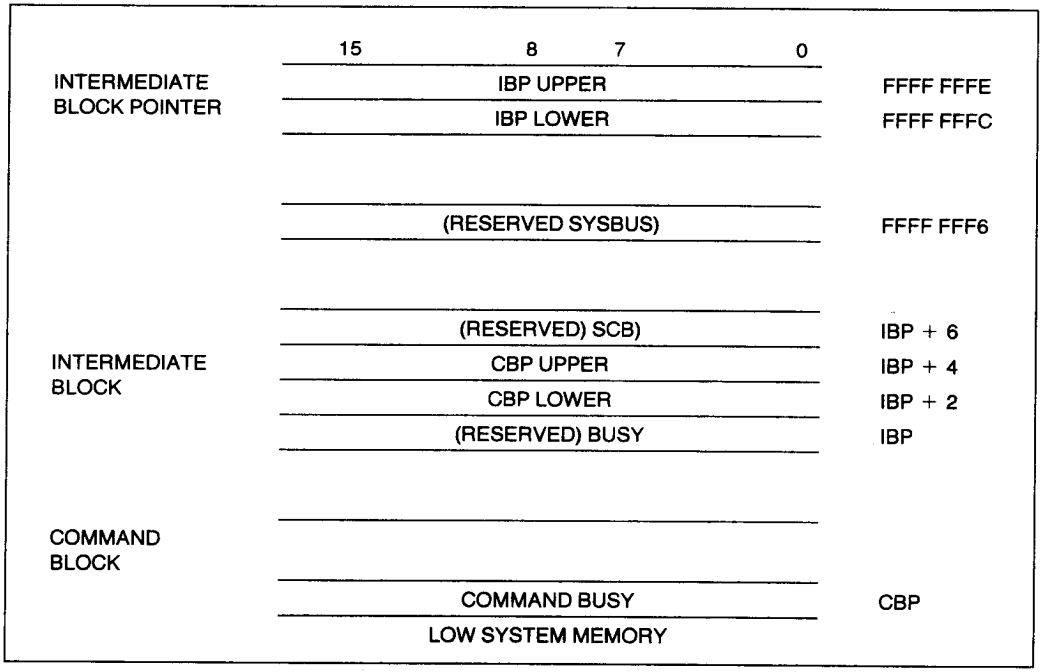


Figure 5a. Initialization Control Blocks

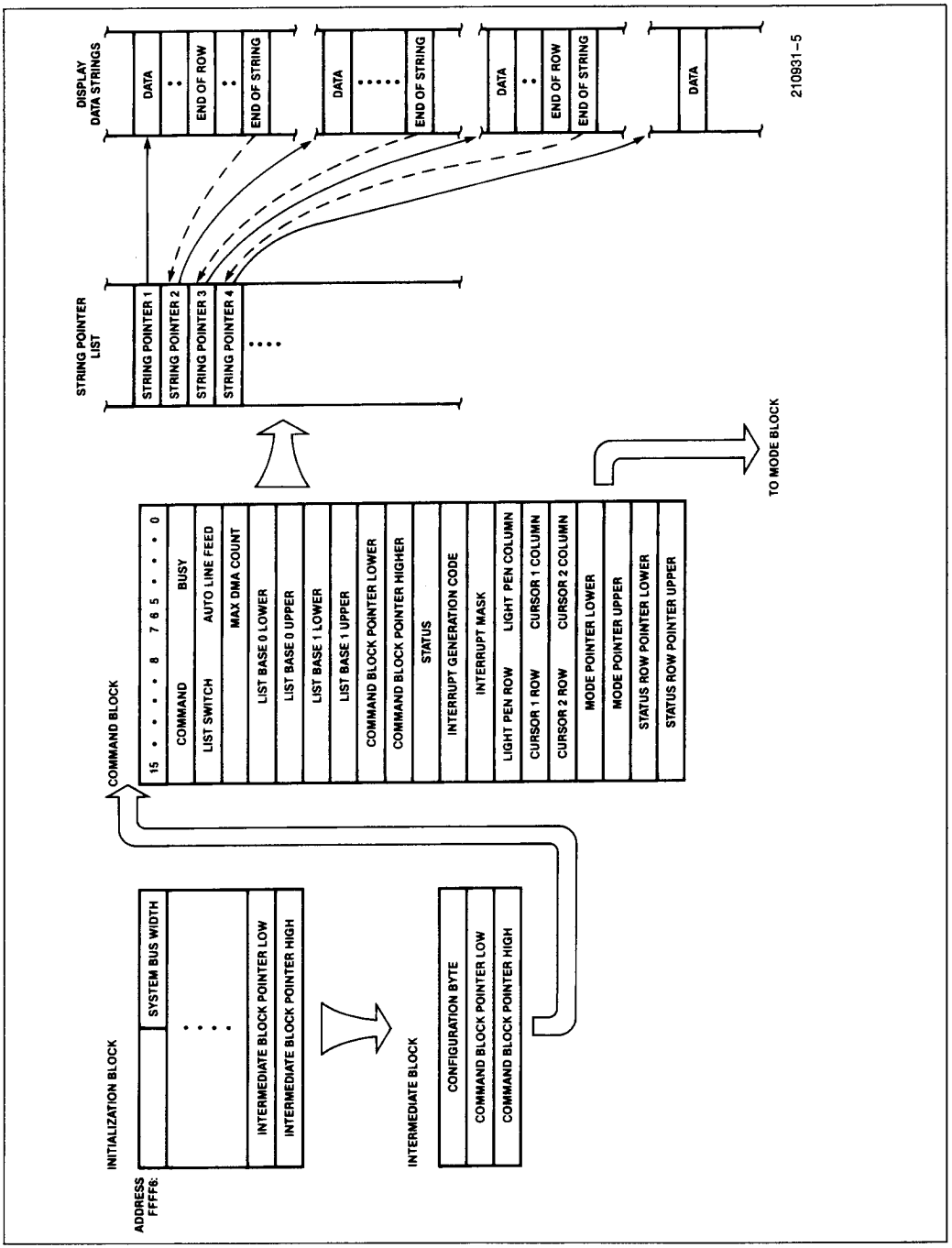


Figure 5b. Control Structure of the 82730

82730 TEST FEATURES

The 82730 has built in Self-Test features that provide testability at the component or at the board level. These features include the test commands and the output pin force capability and are described below.

Output Pin Force Capability

A capability to force logic state (high, low, high impedance) on all output pins is provided in the 82730 Text Co-Processor. This is accomplished by providing a stimulus on pins LC0–LC2 during chip reset. This feature is used for dc parametric tests on the output pins.

The state of pins LC0–LC2 is monitored during chip reset. The state of these pins is latched internally on the falling edge of chip reset. If no external inputs are applied during reset, the state observed will be all 1's and no action will be taken by the 82730. If any external inputs are applied to pins LC0–2 during reset, the resulting action will depend upon the state latched on the falling edge of reset. The 82730 maintains pins LC0–LC2 in high impedance state for the duration of chip reset to avoid contention with external inputs. Also internal pull-ups ensure that a state of all 1's will be detected if no external inputs are applied.

The actions corresponding to each of the observed states of pins LC0–LC2 are summarized in Figure 6a.

Stand-Alone "Self Test"

The built-in Self Test capability of the 82730 can be invoked in a stand-alone mode by applying an external stimulus through pins LC0–LC2 during chip reset. This is the same mechanism as the one used for forcing logic states on output pins. Figure 6a.

If pin LC2 is pulled low during chip reset, the 82730 executes a built-in self test. Upon completion of the self-test, a 16-bit signature, generated internally as the test result, is output via pins WDEF, DAT14-DAT0. The completion is signalled by providing a logic "O" output on pin LC3 as a completion flag. The signature will remain on the output pins until the next chip reset. The 82730 will enter an idle state awaiting chip reset and will not respond to any external inputs until a reset signal is applied. During the process of presenting the signature onto WDEF, DAT14-DATO, the signature will also appear briefly on the AD bus in the form of a bus cycle with two 8-bit accesses to addresses, AAAAH, AAABH. However, this phenomenon is only incidental. Pins WDEF, DAT14-DATO should be used for observing the signature.

The stand-alone self test includes the testing of internal address pointer registers. These registers are not tested when the self test is invoked by issuing a "Self Test" command. (See under Channel Commands below). Therefore, the signature generated during stand-alone self test will be different from that generated by the "Self Test" command.

State of Pins LC0–LC2 During Chip Reset			Action
LC2	LC1	LC0	
0	X	X	Invoke Stand-Alone Self Test
1	0	0	Force all Outputs to High Impedance State
1	0	1	Force all Outputs to Logic High State
1	1	0	Force all Outputs to Logic Low State
1	1	1	NOP

Figure 6a. Output Pin Forcing and Stand-Alone Self Test Invocation

82730 CHANNEL COMMANDS

Table 3. Channel Commands

Command	OP CODE		
1 START DISPLAY	0000	0001	01 H
2 START VIRTUAL DISPLAY	0000	0010	02 H
3 STOP DISPLAY	0000	0011	03 H
4 MODE SET	0000	0100	04 H
5 LOAD CBP	0000	0101	05 H
6 LOAD INTMASK	0000	0110	06 H
7 LPEN ENABLE	0000	0111	07 H
8 READ STATUS	0000	1000	08 H
9 LD CUR POS	0000	1001	09 H
10 SELF TEST	0000	1010	0A H
11 TEST ROW BUFFER	0000	1011	0B H
12 NOP	0000	0000	00 H
13 (RESERVED)	From: 0000	1100	0C H
	To: 1111	1111	FF H

The system processor issues channel commands to 82730 via the Command Block. The processor first checks if the BUSY flag in the command block has been cleared. It should wait for the BUSY flag to be cleared before proceeding with the issuing of a command. When the BUSY flag is cleared, the processor places a command byte in the "COMMAND" location in command block, sets the BUSY flag to a non-zero value and asserts Channel Attention (CA), by activating the CA input to 82730. A Channel Attention should not be issued, if the BUSY flag has not been cleared.

START DISPLAY

0000	0001	CMD Byte

LISTSWITCH, Auto Linefeed, Max DMA Count and Cursor Position values are fetched from the Command Block and stored internally after this command is received. The BUSY flag is cleared and the normal display process is activated.

The MCU fetches strings of data from the memory, using the parameters LISTSWITCH, LBASE0 and LBASE1. The data fetched is interpreted as datastream commands or character data to be displayed by the Display Generator. The MCU loads the data into one of the two Row Buffers in the CRT controller, while the Display Generator displays the data from the other buffer, the buffers being swapped at the end of the row. Any datastream commands encountered during data fetch are immediately executed.

The display process is continued until it is deactivated by a STOP DISPLAY command or a Reset. Other channel commands can be issued while a display is in progress and they will be executed when CA is found active at one of the periodic samplings at each end of frame.

The DIP (Display in Progress) status bit is set and the VDIP (Virtual Display in Progress) is cleared upon receiving a START DISPLAY command. Both bits are reset upon receiving a STOP DISPLAY command or a Reset.

It is necessary to load in proper mode information through a MODESET command before activating the display. Following Reset, START DISPLAY command will not be executed, i.e., will result in a NOP until a MODESET command has been issued.

START VIRTUAL DISPLAY

0000	0010	CMD Byte

LISTSWITCH, Auto Linefeed, Max DMA Count and Cursor Positions are fetched from the Command Block and stored internally upon receiving this command. The BUSY flag is cleared and the Virtual Screen display process is activated.

The operation of Virtual Screen display process is similar to that of a regular display process, except for following a different data access mechanism. The parameters LISTSWITCH, LBASE0 and LBASE1 in the command block represent ACCESS SWITCH, ACCESS BASE0 and ACCESS BASE1 respectively, in virtual screen display.

The VDIP (Virtual Display in Progress) status bit is set and the DIP status bit is cleared upon receiving a START VIRTUAL DISP command: Both DIP and VDIP are reset upon receiving a STOP DISPLAY command or a Reset.

START VIRTUAL DISPLAY command will not activate a display and results in a NOP until a MODESET command is issued after a Reset.

STOP DISPLAY

0000	0011	CMD Byte

The display process is deactivated upon receiving this command. The DIP and VDIP status bit are reset and the BUSY flag is cleared.

This command blanks the display. HSYNC and VSYNC are **not** affected.

MODESET

0000	0100	CMD Byte

The Mode Pointer contained in command block location (CBP + 30) is used to access the Mode Block and the modes are fetched sequentially and loaded into the corresponding internal registers in 82730. LISTSWITCH, Auto Linefeed, Max DMA Count and Cursor Positions are fetched from the Command Block and stored internally upon completion and the BUSY flag is cleared.

The organization of mode words in the mode block and the parameters supplied by them are shown below (See Figure 10). Some of these parameters which are critical to the operation of a text coprocessor are required to remain unchanged over most of normal operation. No provision is made to prevent MODESET from changing these parameters and it is left to the designer to insure that they are not changed.

The modes provide horizontal and vertical mode display parameters, interlace information, DMA burst and spacing specifications, cursor characteristics as well as attribute enables and bit-selects. Typically, this would be the first command issued after initialization. The Mode Block provides all the parameters needed for a complete initialization of the 82730 for display. Thus a single Modeset command can fully initialize the chip. Note that until the first Modeset command is sent, certain functions such as VSYNC and HSYNC are not enabled.

It is necessary to set up proper mode information, before activating a display. Therefore, a display activating command should not be issued unless proper mode information has been loaded through a MODESET command. START DISPLAY and START VIRTUAL DISPLAY commands will result in a NOP if a MODESET command has not been issued since the most recent Reset.

LOAD CBP

0000	0101	CMD Byte

The address pointer "NEW CBP" contained in the command block is fetched and stored in the CBP register in the text coprocessor, replacing the old CBP. This effectively moves the command block in the memory. The Command byte from the new Command Block is fetched and the specified channel command is executed. The BUSY flag in the new Command Block is cleared upon completion.

LOAD INTMASK

0000	0110	CMD Byte

The interrupt mask contained in location "INTMASK" in the command block is fetched and stored internally in the CRT controller. When a particular mask bit is set, the interrupt is disabled for a status bit in the corresponding bit position. An interrupt is generated by the text coprocessor by activating the SINT pin, if a status bit is 1 and the corresponding bit in the interrupt mask is 0. The BUSY flag is cleared upon completion.

Interrupts can be enabled for the following status bits.

7	6	5	4	3	2	1	0	BIT
—	RDC	RCC	FDE	EOF	DBOR	LPU	DUR	STATUS WORD

RDC: Reserved Datastream Command Encountered

RCC: Reserved Channel Command Executed

FDE: Frame Data Error (Fetching characters past physical End of Frame)

EOF: End of "n" frames (Logical end of nth frame)

DBOR: Data Buffer Overrun (Row Buffer filled completely without encountering END OF ROW command)

LPU: Light Pen Update

DUR: Data Underrun (Buffer swap initiated before finishing Row Buf loading)

READ STATUS

0000	1000	CMD Byte

The internal status register is written to "STATUS" location in the command block. The status register is then cleared, however DIP and VDIP status bits are not cleared. LISTSWITCH, AUTO LINEFEED, Max DMA Count and Cursor Positions are fetched from the Command Block and stored internally. The BUSY flag is then cleared.

STATUS WORD

15-9	8	7	6	5	4	3	2	1	0
—	VDIP	DIP	RDC	RCC	FDE	EOF	DBOR	LPU	DUR

LPEN ENABLE

0000	0111	CMD Byte

The Light Pen detection process is enabled to search for a rising edge on the LPEN pin. The BUSY flag is then cleared.

If the display process is active and a rising edge is detected on the LPEN input, the corresponding row and column position on the screen is stored internally. At the next end of frame, the LPEN position is written to locations "LPENROW" and "LPENCOL" in the command block and the LPU (Light Pen Update) status bit is set.

If the display process is not active, the command has no immediate effect. However, the LPEN detection process remains enabled and will take effect if a display is activated subsequently.

LD CUR POS

0000	1001	CMD Byte

The display row and column positions of cursors 1 & 2 as set in locations "CUR1 ROW," "CUR1 COL," "CUR2 ROW" and "CUR2 COL" in the command block are loaded into internal registers in the CRT controller. Also LISTSWITCH Auto Linefeed and Max DMA Count are loaded from the Command Block and the BUSY flag is cleared. Ths command is used to change the cursors only. Note that the cursor positions are also updated with the execution of other channel commands.

The cursor characteristics for display are defined by the mode. During the display process, a cursor will be displayed accordingly at the position specified above.

TEST COMMANDS

The test commands for the 82730 are issued in the same manner as the normal channel commands. However, the parameters used by test commands are different from those used by the channel commands in normal operation. Therefore, a Test Block which is similar in format to the Command Block is defined. Switching between Command Block and Test Block is accomplished using the "Load CBP" command. The Test Block differs only in the parameter locations associated with the command. The locations for New CPB, command byte and busy flag are the same for both Command Block and Test Block. The "Test Result" location in Test Block corresponds to the "Status" location in Command Block.

The test commands can be executed, following chip reset, only until the first Modeset command is issued. Once a Modeset command has been executed following chip reset, any subsequent test commands will not be executed and will result in a NOP.

"Self Test" Command

0000	0010	CMD Byte

A built-in Self test is performed using an internal test pattern. The signature generated during the test is written to the Test Result location (TBP + 18) in the Test Block. The Busy Flag in the Test Block is then cleared. The Self Test command must be immediately preceded by a chip reset in order to ensure a consistent signature.

The Test Block format for issuing the Self Test command is shown in Figure 6b.

"Row Buffer Test" Command

0000	1011	CMD Byte

The Load Pointer in Test Block is fetched. It points to the system memory area storing the test pattern to be used for testing the on-chip RAM (i.e. - the Row Buffers). The Store Pointer, which points to memory area where the data read back from the RAM will be written, is also fetched from Test Block.

Successive words are fetched from memory and written to the Row Buffer, until it is completely filled. Note that three extra words beyond the maximum Row Buffer capacity will be fetched. If N = Max Row Buffer capacity, (N + 3) words will be fetched from memory. The extra words fetched will be ignored. The Row Buffer contents are then read back and are written to successive locations in memory area pointed to by the Store Pointer. The test is then repeated on the second Row Buffer. Note that the (N + 4)th word in the pattern stored in memory constitutes the first word written to the second Row Buffer. The data storage for the Row Buffer test patterns is illustrated in Figure 6c.

Internally, the Row buffers are 17-bits wide, while the data path is 16-bits wide. During the writing of data to Row Buffers, a complement of bit 15 is written to bit 16 of the Row Buffer in order to test all 17 bits. During the read back, two data words are stored in system memory for each location in the Row Buffer. The first word will consist of bits 0–15 read from the Row Buffer, while the second word will consist of bits 0–14 and bit 16 from the Row Buffer. Thus a total of 4*N words will be stored back in system memory as a result of the Row Buffer Test (2*N for each Row Buffer).

	BIT 15 8	7 0	TEST BLOCK POINTER (TBP)
	COMMAND	BUSY	
PARAMETER			TBP + 2
LOCATIONS			TBP + 4
NOT USED			TBP + 6
FOR SELF TEST			TBP + 8
COMMAND			TBP + 10
			TBP + 12
	NEW CBP LOWER		TBP + 14
	NEW CBP UPPER		TBP + 16
	TEST RESULT		TBP + 18

Figure 6b. Test Block Format for "Self Test" Command (For both 16-bit and 32-bit addressing modes)

A signature is generated during the test and is written to Test Result location in Test Block upon completion. The BUSY flag in the Test Block is then cleared.

The Test Block format for issuing the Row Buffer Test command is illustrated in Figures 6d.1 and 6d.2. Note that the locations for Load Pointer and Store Pointer parameters are different for 16-bit and 32-bit addressing modes.

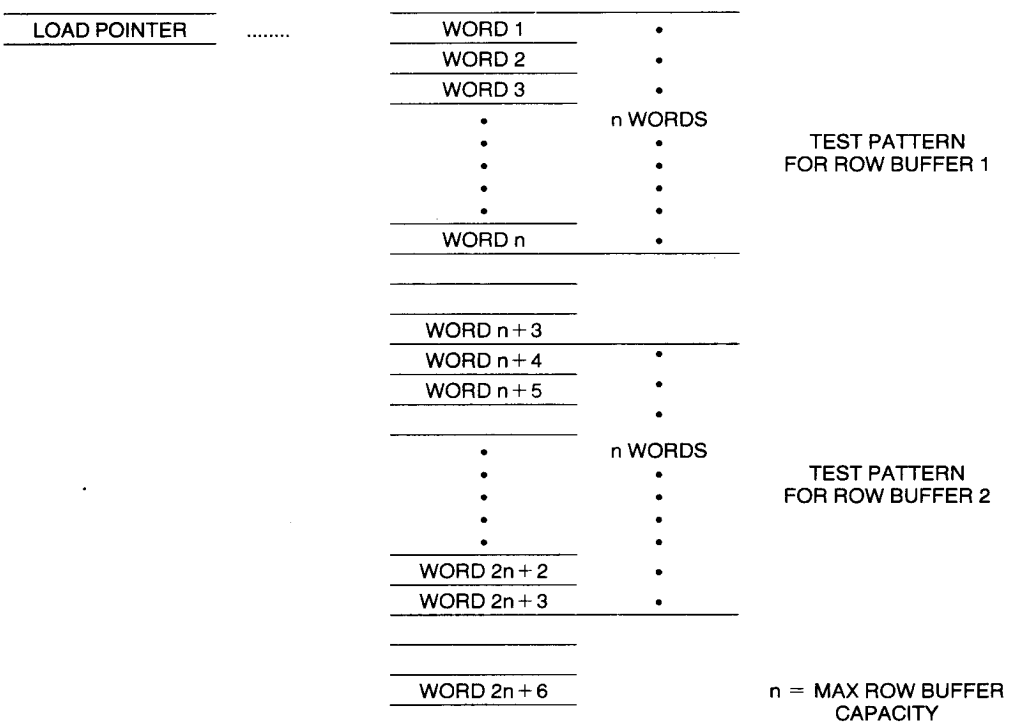


Figure 6c. Data Storage for Row Buffer Test Command

15	8	7	BIT 0	TEST BLOCK
COMMAND		BUSY		POINTER (TBP)
(RESERVED)				TBP + 2
LOAD POINTER LOWER				TBP + 4
LOAD POINTER UPPER				TBP + 6
STORE POINTER LOWER				TBP + 8
STORE POINTER UPPER				TBP + 10
(RESERVED)				TBP + 12
NEW CBP LOWER				TBP + 14
NEW CBP UPPER				TBP + 16
TEST RESULT				TBP + 18

Figure 6d.1 Test Block Format for "Row Buffer Test" Command (32-bit addressing mode)

15	8	7	BIT 0	TEST BLOCK
COMMAND		BUSY		POINTER (TBP)
(RESERVED)				TBP + 2
(RESERVED)				TBP + 4
LOAD POINTER				TBP + 6
(RESERVED)				TBP + 8
STORE POINTER				TBP + 10
(RESERVED)				TBP + 12
NEW CBP LOWER				TBP + 14
NEW CBP UPPER				TBP + 16
TEST RESULT				TBP + 18

Figure 6d.2 Test Block Format for "Row Buffer Test" Command (16-bit addressing mode)

NOP

0000	0000	CMD Byte

LISTSWITCH, Auto Linefeed, Max DMA Count, and Cursor Positions are fetched from the command block and stored internally as in all other channel commands. The Busy flag is then cleared.

82730 DATASTREAM COMMANDS

Datastream Commands

Datastream commands are commands embedded in the data fetched from memory by the data access task. These commands are differentiated from character data by the command bit. The most significant bit (MSB) of each data word is designated as the command bit. If the command bit is "1", the lower 15 bits of the data word are interpreted as a datastream command, while if the command bit is "0" the lower 15 bits (or 7 bits if DTW16 = 0) are interpreted as character data.

Datastream Command Operation

During the data access task, the Micro Controller Unit (MCU) examines the command bit of each data word fetched. If the command bit is 1, it executes the datastream command specified in the data word. Otherwise, it stores the lower 15 bits of the data word in the Row Buffer as character data. This process is repeated for each data word fetched.

Datastream Command List

Table 4. 82730 Datastream Commands

	Command	Command Code				OP Code
		Op Code		Parameters		
1	ENDROW	1000	0000	XXXX	XXXX	80
2	EOF	1000	0001	XXXX	XXXX	81
3	END OF STRING & END OF ROW	1000	0010	XXXX	XXXX	82
4	FULROWDESCRPT	1000	0011	"n"		83
5	SL SCROLL STRT	1000	0100	XXX SCR LINE		84
6	SL SCROLL END	1000	0101	XXX END LINE		85
7	TAB TO n	1000	0110	"n"		86
8	LD MAX DMA COUNT	1000	0111	COUNT		87
9	ENDSTRG	1000	1000	XXXX	XXXX	88
10	SKIP n	1000	1001	"n"		89
11	REPEAT n	1000	1010	"n"		8A
12	SUB SUP n	1000	1011	"n"		8B
13	RPT SUB SUP n	1000	1100	"n"		8C
14	SET GEN PUR ATTRIB	1000	1101	GPA OP		8D
15	SET FIELD ATTRIB	1000	1110	XXXX	XXXX	8E
16	INIT NEXT PROCESS (Command process command)	1000	1111	XXXX	XXXX	8F
17	(RESERVED)	10XX	XXXX	XXXX	XXXX	90-BF
18	NOP	11XX	XXXX	XXXX	XXXX	C0-FF

Datastream commands can be used for changing Row Characteristics on a row by row basis, for carrying out editing functions and for formatting data into rows and frames. These commands are executed by the MCU immediately after they are encountered. As a convenience for the user, the set of all possible command codes starting with "1" in the two most significant bits has been designated as NOP commands. The user can use these command codes for any desired purpose. All other command codes which are not presently defined, are reserved for future expansion and should not be used by the user. The currently undefined codes cause the RDC (Reserved Datastream Command) status bit to be set and also generate an interrupt, if enabled. Reserved command codes should not be used.

The preceding commands are recognized as valid datastream commands. The corresponding command codes are also indicated. It should be noted that the most significant bit of the command bit is always 1, in order for the word to be interpreted as command.

The "Init Next Process" command can be issued only through a command process in Virtual Screen Display. It is included in this list because its operation is analogous to a datastream command in a virtual screen access environment. Also, in virtual screen display certain datastream commands are interpreted differently, depending upon whether they are encountered in a process datastream or as command process commands. When a command is ignored (becomes a NO-OP) in a virtual display, any parameters that are associated with it are also ignored. The command process command operation is discussed separately. The operation of all other datastream commands is described below.

ENDROW

15	14	8	7	0
1	000	0000	XXXX	XXXX

This command signifies that no more characters will be loaded in the Row Buffer for this row and an End of Row indicator is stored accordingly. When the row currently being loaded is displayed, the Display Generator (DG) will blank the screen from the end of row character position until the physical end of row.

The Micro Controller Unit (MCU) stops fetching data and waits for DG to swap the Row Buffers. The data access task is resumed following the buffer swap. If a physical end of frame is reached while the MCU is waiting for a buffer swap the MCU ceases to wait and executes an EOF (End of Frame) command.

In virtual display, this command is interpreted as a VEOR (Virtual End of Row) if encountered in a virtual process datastream.

VEOR

ENDROW command in a virtual process datastream is interpreted as VEOR (Virtual End of Row) and it terminates a virtual row. The current LPTR is stored in the process header addressed by the "Process Addr" register. The Max Count register is also stored in the Max DMA Count location in the process header. Similarly, the Field Attribute Mask is also stored in the header. In addition, in auto linefeed mode (ALF = 1) other parameters characterizing the process state are also saved in the header. The "Process Addr" register is loaded with the address of the header of the next process fetched from the Access table. The "Access Tab Addr" register is post-incremented by two if a 16-bit addressing option is used and by four if 32-bit addressing is used. The data access task is then resumed for the next process.

EOF

15	14	8	7	0
1	000	0001	XXXX	XXXX

This command (End of Frame) signifies that no more characters will be loaded in the Row Buffers for this frame. The Micro Controller Unit (MCU) stops fetching data words and waits for the physical end of frame. If a virtual display is in progress, this command is interpreted as VEOS (Virtual End of Frame), if encountered in a virtual process datastream.

The Display Generator (DG) swaps the row buffers at the end of the current display row and starts displaying the row containing the EOF command. When the character preceding the EOF command is displayed, the DG blanks the screen until the physical end of frame. The MCU fetches the Status Row data then waits until its display is completed. It then performs the actions described below.

If LPEN has been enabled and a rising edge on the LPEN input has been detected, the LPENROW and LPENCOL positions in the command block are updated and the LPU status bit is set. If a Channel Attention has occurred, i.e., if CA has been activated, the command byte is fetched from command block and the specified channel command is executed. If the command issued is a "Stop Display" command, the MCU will terminate the display process and wait for the next channel attention. Otherwise, the MCU resumes the data access task be reinitializing pointers for the new frame and continues to fill the Row Buffers.

VEOF

EOF command in a virtual process datastream is interpreted as VEOF (Virtual End of Frame). It provides for reinitialization of LPTR using LISTSWITCH, LBASE0 AND LBASE1 for each process, analogous to the automatic reinitialization of LPTR at each end of frame in a Normal Display.

LPTR for the current process is reinitialized using LISTSWITCH, LBASE0 and LBASE1 contained in the process header. The End of Display (EOD) bit in the header is set to 1. The current process is terminated as in a VEOR and the next process in Access Table is accessed.

EOL

15	14	8	7	0
1	000	0010	XXXX	XXXX

The EOL (End of Line) command has a combined effect of NXTROW and NXTSTRG commands. All the actions performed in a END OF ROW command are carried out. In addition a END OF STRING command is executed before resuming the data access task. Thus, following the end of row, the data access is continued with the next data string. In virtual process datastream, this command has the combined effect of VEOR and END OF STRING.

FULROWDESCRPT

15	14	8	7	0
1	000	0011	N	

The next "n" words fetched from memory are loaded into the Row Characteristics holding registers. "n" is specified by the lower order byte of the command word and should be between 0 and 7.

The parameters loaded by this command will be used to define the row characteristics at the time the row currently being loaded is displayed. The data words defining these characteristics which follow the FULROWDESCRPT command must be ordered and organized in memory in a specific format. The format for FULROWDESCRPT parameters is shown below in Figure 6e starting with "Lines Per Row" as the first parameter loaded.

This command will be ignored if encountered in a virtual process datastream. The MSB of all the parameters must be zero for proper operation in virtual display.

	Upper Byte											Lower Byte			
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					RVV	BLK	DBL	W								
Lines Per Row	—	—	—		ROW	ROW	HGT	DEF	—	—	—			LPR		
Normal Start/Stop	—	—	—			NRMSTRT			—	—	—			NRMSTOP		
Superscript Start/Stop	—	—	—			SUPSTRT			—	—	—			SUPSTOP		
Subscript Start/Stop	—	—	—			SUBSTRT			—	—	—			SUBSTOP		
Cursor 1 Start/Stop	—	—	—			CUR1 STRT			—	—	—			CUR1 STOP		
Cursor 2 Start/Stop	—	—	—			CUR2 STRT			—	—	—			CUR2 STOP		
Underline Line Selects	—	—	—			UL2 LINE SEL			—	—	—			UL1 LINE SEL		

RVV ROW, when this bit is set the CRVV pin will be inverted for the next full row.

BLK ROW, when this bit is set the row will be blanked (BLANK high).

DBLHGT, when the double height bit is set, all character are displayed with twice the scan lines per row.

WDEF, when the width defeat bit is set, the WDEF pin is activated for the entire row.

The following can be programmed from 0 to 31 yielding a range of 1 to 32 lines.

LPR specifies number of lines per row.

NRMSTRT, SUPSTRT, SUBSTRT specify line numbers in a display row which mark the start of normal, superscript and subscript characters respectively.

NRMSTOP, SUBSTOP, SUBSTOP specify line numbers in a row where normal, super script and subscript characters end respectively.

CUR1 STRT, CUR2 STRT specify the starting line numbers in a row for cursor 1 and cursor 2 respectively.

ULINE1 SEL, ULINE2 SEL specify the line numbers in a row where underline 2 will appear respectively.

All FULROWDESCRPT parameters affect the row in which they are programmed and stay in effect until changed by another FULROWDESCRPT command.

Figure 6e. Format for FULROWDESCRPT

SL SCROLL STRT

15	14	8	7	5	4	0
1	000	0100	XXX	SCR LINE		

The Slow Scan register in 82C3 is loaded with the scroll line specified by the five least significant bits of the command word. When the row currently being loaded is displayed, the line count for that row will start with the value specified by the Slow Scan register. A "Margin" (MGN) parameter, loaded by MODESET, specifies the number of blank lines plus one to be added at the top of the slow scroll field on the screen. This ensures the availability of sufficient DMA time for fetching the next row, when only a small number of scan lines are displayed in the top row of slow scroll window. This command is used for starting a slow scroll. (Note: MGN = 0 results in no margin buffer lines.)

This command will be ignored if encountered in a virtual process datastream or if a SL SCROLL END command is encountered later on the same row.

SL SCROLL END

15	14	8	7	5	4	0
1	000	0101	XXX	END LINE		

The scroll location in row characteristics holding registers is loaded with the number of lines specified by the five least significant bits of the command word. This number specifies the number of lines to be displayed when the row currently being loaded is displayed. This is used instead of the regular LPR (Lines Per Row) characteristics, for this particular row. This command is used in the last row of a slow scroll for terminating a slow scroll. The Margin (MGN) parameter, loaded by MODESET, is used in the same way as in slow scroll start except that the specified number of blank lines are inserted at the bottom of the slow scroll in this case. This command will be ignored if encountered in a virtual process datastream or if followed by a SL SCROLL STRT on the same row.

TAB TO n

15	14	8	7	0
1	000	0110	"n"	

The lower byte of the command word specifies the column (RCLK count) after SYNCSTRT at which a Tab should occur. At display time, after the character preceding the Tab command is displayed, the screen is blanked until the RCLK count specified by the command ("n") is reached. After reaching the specified count, display is resumed by displaying the character following the TAB command.

If the RCLK count specified by the Tab command has already occurred before beginning the blanking for Tab, the display will be blanked until the end of the row.

This command is ignored, if encountered in a virtual display process datastream.

LD MAX DMA COUNT

15	14	8	7	0
1	000	0111		MAX COUNT

The Max Count register in 82730 is loaded with the Max DMA Count specified by the lower byte of the command word. The DMA Counter is also reinitialized with the Max Count value in the Command Block after all channel commands.

MAX DMA Count is programmable in the range of 1 to 256 (MAX COUNT value 0 equals 256). However, counts greater than the row buffer capacity will cause row buffer overruns if the data strings depend on MAX DMA to terminate the fetching.

The DMA counter is decremented for each data word as the Row Buffer is being loaded. Datastream commands and words supplying parameters for datastream commands as in FULROWDESCRPT, are not counted. Superscript/Subscript characters are counted in pairs, i.e., a pair of characters causes only one count.

In virtual screen display, every time a new process is accessed, the DMA counter is initialized with the Max DMA Count contained in the process header. This value is also stored in a Max Counter register.

At virtual end of row (VEOR) the Max Count register is written to the process header. The "LD Max DMA Count" command is ignored if encountered in a virtual process datastream.

ENDSTRG

15	14	8	7	0
1	000	1000	XXXX	XXXX

The SPTR register in the 82730 is loaded with a new String Pointer (SPTR) value fetched from the memory location indexed by the List Pointer (LPTR), which is stored in the LPTR register. The LPTR register is incremented by two if a 16-bit addressing option is used and by four if 32-bit addressing is used. When more than one 82730 is connected in a cluster, each of them adds an offset, determined by its position in the cluster, to the pointer fetched from memory, before storing it in its SPTR register.

This command directs the data access to the next data string in the list of strings indexed by LPTR. The operation of this command is identical for a Virtual or Normal Display. In virtual display, the next data string within the current display process is accessed.

SKIP n

15	14	8	7	0
1	000	1001	n	

The next "n" data words fetched from memory are ignored. "n" is specified by the lower byte of the command word and is programmable from 0 to 255. If n equal to 0 is specified, no words are skipped. Any datastream commands encountered in the data fetch are not counted towards these n words. Also parameters following the datastream command as in FULROWDESCRPT are not counted. All embedded datastream commands are executed with the following exceptions.

If a Tab To N data stream command is encountered during the execution of a Skip N command, the Tab command will result in a NOP, i.e. a Tab embedded in the data to be skipped will be ignored.

If an EOL (End Of Line) data stream command is encountered during the execution of a Skip N command, it will be executed with the following effect. In non-auto line feed mode, (ALF = 0) the EOL command has the combined effect of End Of Row and End Of String commands. In auto line feed mode, (ALF = 1) the EOL command has the effect of an End Of String command only.

If the data words skipped include any superscript-subscript characters, they are skipped in pairs and a pair of characters is counted as only one count in "n". If another skip command is encountered its value of "n" is added to the present skip count and skipping continues.

REPEAT n

15	14	8	7	0
1	000	1010	n	

The next data word (byte, if DTW16 = 0) fetched from memory is stored in the Row Buffer "n" times, where "n" is specified by the lower byte of the command word, "n" is programmable from 0 to 255. If n equal to 0 is specified no repetitions will occur, and the word following the Repeat n command will be ignored. This character will eventually be displayed n times. The DMA counter is also made to count n times. In non-auto linefeed mode (ALF = 0), reaching Max DMA Count before the n repetitions are completed will result in a termination of the Repeat n command. This command will also be terminated if the Row Buffer gets filled completely before the n repetitions are completed.

It should be noted that the data word immediately following the Repeat n command is treated as character data, irrespective of the value of its command bit.

SUP/SUB n

15	14	8	7	0
1	000	1011	n	

The next "n" pairs of data words (bytes, if DTW16 = 0) fetched from memory are treated as superscripts or subscript characters. "n" is specified by the lower byte of the command word. These n pairs are assumed to be ordered with the superscript preceding the subscript.

No datastream commands are permitted in the 2n words following this command. All of these words are interpreted as superscript-subscript pairs. The DMA counter is made to count only once for each pair of characters. In non-auto linefeed mode (ALF = 0), reaching the Max DMA Count will result in a termination of this command. If n equal to zero is specified, no action will result.

RPT SUB/SUP n

15	14	8	7	0
1	000	1100	n	

The operation of this command is similar to that of the "Repeat n" command except that the pair of characters following the "RPT SUB/SUP n" command is repeated n times. "n" is specified by the lower byte of the command word and is programmable from 0 to 255. If n equal to zero is specified , no repetitions will occur, and the two data words following the "RPT Sub/Sup n" command will be ignored. The two data words (bytes, if DTW16 = 0) immediately following the command word are interpreted as a superscript-subscript pair and are repeated. The DMA counter is made to count only once for each repetition of the pair. In non-auto linefeed mode (ALF = 0), reaching Max DMA Count prior to completion of n repetitions will cause a termination of this command.

SET GEN PUR ATTRIB

15	14	8	7	0
1	000	1101		GPA OPERAND

This command provides control over the output pins assigned to General Purpose Attributes, GPA 1 through GPA4.

The GPA in the Process Header is updated each time a SET GPA command is executed. Thus the GPA state in the header is updated to reflect any changes caused by the "Set Gen Pur Attrib" command. The GPA command occupies a character space on the screen. Consequently, a GPA command is counted as a character towards MAX DMA count. However, a GPA command nested in a Skip N or a TAB to N command is skipped, i.e., it has no effect.

The encoding of the operand, specifying GPA operation, is shown below.

SET FIELD ATTRIB

15	14	8	7	0
1	000	1110	XXXX	XXXX
0	FIELD ATTRIBUTE MASK			

The word following this command is fetched. This word is used as a Field Attribute Mask in storing all subsequent display data words in row buffer. The bits in the data words fetched from memory corresponding to the bit positions containing a "1" in Field Attribute Mask are all set to 1 before storing the data word in row buffer. The Field Attribute Mask is used on all display data words fetched from memory. The mask register will contain all 0's upon reset and is cleared at the beginning of each frame.

NOP

15	14	8	7	0
1	1XX	XXXX	XXXX	XXXX

No action is taken. The data access task is resumed by fetching the next data word.

	7	6	5	4	3	2	1	0
GPA OPERAND	GPA4 DATA	GPA4 EN	GPA3 DATA	GPA3 EN	GPA2 DATA	GPA2 EN	GPA1 DATA	GPA1 EN

ENCODING

GPAx DATA	GPAx EN	FUNCTION
0	0	ROW BUFFER DATA
1	0	ROW BUFFER DATA
0	1	GPA DATA = 0
1	1	GPA DATA = 1

Datastream Command Conventions

The reaching of Max DMA Count, encountering of terminating commands such as ENDROW, EOF, etc. and occurrences of these while executing a "skip n" command give rise to various possible combinations of events. The behavior of 82730 under these circumstances is described below:

i) When Max DMA Count is reached, it has the effect of a VEOR command if a Virtual Display is in progress or a ENDROW command if a Normal Display is in progress. It also causes an automatic end of string i.e., the effect of a NXTSTRG command in non-auto linefeed mode (ALF = 0).

ii) In non-auto linefeed mode, "Repeat n", "Sub/Sup n" and "RPT Sub/Sup n" commands are terminated upon reaching a max DMA count, even if "n" is not reached.

iii) "Skip n" command is terminated if EOF command is encountered. It is also terminated upon encountering a ENDROW command in non-auto linefeed mode (ALF = 0).

iv) "Repeat n" "Sub/Sup n" and "RPT Sub/Sup n" commands can be nested within a "Skip n" command. If superscript-subscript characters are skipped, each pair of characters counts as one skipped character. If the above commands are encountered during a "Skip n" and if the specified count (n) in these commands is not reached by the end of execution of the "Skip n" command, the execution of the nested command is continued beyond the termination of "Skip n" command until the remaining portion of the count specified in the nested command is completed.

VIRTUAL SCREEN MODE

Command Process Commands

In Virtual Screen Display, 82730 accesses display processes and command processes through the Access table. The command processes enable the I/O Driver process to direct 82730 to execute certain datastream commands by inserting an appropriate command process address in the Access table. This capability enables the preservation of uniformity and consistency of operation between normal and virtual environments, by assigning different interpretations to the command according to the access environment. It is especially useful for termination and initialization commands. The operation of command process commands is analogous to that of datastream commands except for a different access environment.

Command Process of Command List

The commands allowed in command processes can be divided into two subsets. The first subset consists of commands that can be issued only through a command process, while the second one consists of normal datastream commands that can also be issued through a command process. The command code for a datastream command issued through a command process is the same as that for the normal datastream command embedded in the data. However, certain datastream commands are interpreted differently when they are issued through a command process as opposed to embedding in the datastream of a virtual display process. The most significant bit (MSB) of the command word must be a "1". In the datastream, this bit distinguishes a command word from character data. In the process environment, this bit distinguishes a command process from a display process. The commands permitted in command processes are listed below. No other commands will be recognized if encountered in a command process and will result in a NOP. All undefined command codes apart from those designated as NOP are reserved and should not be used. Encountering an illegal command code causes the RDC (Reserved Datastream Command) status bit to be set and will generate an interrupt, if enabled.

Table 5. Command Process Command List

Command	Interpretation In Virtual Process Datastream	Command Code				OP Code
		Op Code		Parameters		
Command Process Only Command:						
1 INIT NEXT PROCESS	NOP	1000	1111	XXXX	XXXX	8F
Command Process or Datastream Commands:						
2 ENDROW	VEOR	1000	1000	XXXX	XXXX	80
3 EOF	VEOR	1000	0001	XXXX	XXXX	81
4 EOL	VEOR + NXTSTRG	1000	0010	XXXX	XXXX	82
5 FULROWDESCRPT	NOP	1000	0011	"n"		83
6 SL SCROLL STRT	NOP	1000	0100	XXX	"SCR LINE"	84
7 SL SCROLL END	NOP	1000	0101	XXX	"END LINE"	85
8 TAB TO n	NOP	1000	0110	"n"		86
9 LD MAX DMA COUNT	NOP	1000	0111	"COUNT"		87
10 (RESERVED)	RESERVED	10XX	XXXX	XXXX	XXXX	90-BF
11 NOP	NOP	11XX	XXXX	XXXX	XXXX	C0-FF

INIT NEXT PROCESS

15	14	8	7	0
1	000	1111	XXXX	XXXX

This command can be used only in a command process to initiate a virtual display "window".

Upon receiving this command, the command process is terminated and the next process in Access Table is accessed by fetching the new process address. However, the LPTR register is not directly loaded from the LPTR location in the process header. Instead, LISTSWITCH in the process header is examined and LPTR is initialized with the value LBASE 0 or LBASE 1 depending upon whether LISTSWITCH is 0 or 1 respectively. Both LBASE0 and LBASE1 are contained in the header.

The process header format is shown in Figure 7. Also the End of Display Bit (EOD) in the header is reset.

The data access task for a virtual display is then resumed, with this value of LPTR.

	15	14	13	8	7	6	0	LOCATION
LS: LISTSWITCH ALF: AUTO LINE FEED	0		—		EOD	—		PROCESS ADDR
			—		LS ALF			PROC ADDR + 2
			—		MAX DMA COUNT			PROC ADDR + 4
					LBASE0 LOWER			PROC ADDR + 6
					LBASE0 UPPER			PROC ADDR + 8
					LBASE1 LOWER			PROC ADDR + 10
					LBASE1 UPPER			PROC ADDR + 12
	1		—		GPA			PROC ADDR + 14
	1				FIELD ATTRIBUTE MASK			PROC ADDR + 16
					LPTR LOWER			PROC ADDR + 18
					LPTR UPPER			PROC ADDR + 20
SAVE AREA					SPTR LOWER			PROC ADDR + 22
					SPTR UPPER			PROC ADDR + 24
	RPT S/S	S/S	RPT	— —	REPT COUNT			PROC ADDR + 26
	1				REPT CHAR			PROC ADDR + 28
	1				REPT CHAR 2			PROC ADDR + 30

PROCESS ADDR

15	14	8	7	0
1			COMMAND	

C/$\overline{D}$

Figure 7. Process Header for Display and Command Process

ENDROW

15	14	8	7	0
1	000	0000	XXXX	XXXX

The actions performed by a ENDROW datasteam command in a Normal Display are carried out. The next process in Access Table is accessed and the data access task is resumed, after the next Row Buffer swap.

EOL

15	14	8	7	0
1	000	0010	XXXX	XXXX

This command is identical to ENDROW command in Virtual Display in Command Process environment. ENDSTRG, which is strictly a data operation within a display process is meaningless in the command process environment.

EOF

15	14	8	7	0
1	000	0001	XXXX	XXXX

The actions performed by an EOF (End of Frame) datastream command in a Normal Display are carried out.

FULROWDESCRPT

15	14	8	7	0
1	000	0011	"n"	

The actions performed by the FULROWDESCRPT datastream command are carried out. The data access task is resumed by accessing the next process in the Access Table.

SL SCROLL STRT

15	14	8	7 5	4 0
1	000	0100	XXX	"SCR LINE"

The same actions as the SL SCROLL STRT datastream command. The data access is resumed with the next process in Access Table.

SL SCROLL END

15	14	8	7 5	4 0
1	000	0101	XXX	"END LINE"

The actions performed by a SL SCROLL END datastream command, in a Normal display, are carried out. The data access task is resumed with the next process in Access Table.

TAB TO n

15	14	8	7 0
1	000	0110	"n"

The effect of this command process command is identical to that of the TAB TO n datastream command. The TAB can be used to establish the left edge of a virtual display "window".

LD MAX DMA COUNT

15	14	8	7 0
1	000	0111	MAX COUNT

The Max Count register on 82730 is loaded with the value specified by the lower byte of the command word. The DMA counter is also initialized with this Max Count Value.

The next process in the Access Table is accessed. However, the Max DMA Count value in the process header is not used for initializing the DMA counter. Instead, the DMA counter as initialized by the LD Max DMA Count command is used for this process. The virtual display data access task is then resumed normally. When the process is terminated, the new Max Count value is written to the process header. Thus the Max Count value in the header is updated as a result of this command.

NOP

15	14	8	7	0
1	1XX	XXXX	XXXX	XXXX

No action is taken. Data access task is resumed by fetching the next process address from Access Table.

ERROR AND STATUS HANDLING

Error Conditions

Since the MCU and DG function asynchronously with respect to each other, different relative timings in MCU and DG operation are possible, some of which result in error conditions. The lack of appropriate termination commands for row or frame data in the datastream also gives rise to certain error conditions. These types of situations occurring in display process operation are described below.

In normal operation, DG initiates a buffer swap at the physical end of a display row. If the MCU has not finished loading its row buffer by that time, a "Data Underrun" occurs. This results in blanking of the screen until physical end of frame by DG and execution of an EOF (End of Frame) command by MCU. Data underrun also occurs when the first row of the frame has not finished loading by the start of the character field. The entire frame will be blanked in this case.

If a physical end of frame is reached prior to encountering an EOF datastream command, a "Frame Data Error" occurs, which results in the execution of an EOF command by MCU. (Note that this does not disrupt the visible display action, and may not constitute an error for certain data structures. The error indication is included as a flag where knowledge of this condition is desired.) Similarly, when the MCU fills up a row buffer completely, without encountering a ENDROW comand, the "Data Buffer Overrun" flag is set.

All of the above conditions result in the setting of an appropriate status bit and generation of an interrupt if the corresponding interrupt has been enabled.

Status and Interrupt Handling

A status word is maintained in an internal register by 82730 and it is written to the "STATUS" location in command block when the "Read Status" channel command is executed. The processor can thus read status information by issuing this command. The processor can also enable interrupts for certain status bits by specifying an interrupt mask which is loaded in 82730 as a result of a "Load Int Mask" channel command. This establishes a communication mechanism between 82730 and the processor for error and status reporting.

Status Word

The format for the status word is shown below. The function of each of the status bits is described below.

The status bits get set under the conditions described above. Interrupts can be enabled for all status bits except DIP and VDIP bits. The interrupt status bits are cleared at the beginning of each new display field. DIP and VDIP bits are cleared only after receiving a "STOP DISPLAY" command or a Reset.

All status bits are cleared by a Reset.

15	9	8	7	6	5	4	3	2	1	0
(RESERVED)		VDIP	DIP	RDC	RCC	FDE	EOF	DBOR	LPU	DUR

VDIP: Virtual Display In Progress
DIP: Display In Progress
RCC: Reserved Channel Command
RDC: Reserved Datastream Command
FDE: Frame Data Error

DUR: Data Under Run

This status bit is set by Display Generator if the Microcontroller Unit (MCU) has not finished loading its Row Buffer when the DG intiates a buffer swap at the physical end of a display row. This condition is defined as data underrun and causes the MCU to execute an EOF command and the DG to blank the screen until the physical end of frame.

LPU: Light Pen Update

This status bit is set by the MCU after updating the LPENROW and LPENCOL locations in command block. The detection of LPEN input is enabled by the LPEN ENABLE channel command. The detection of a rising edge on the LPEN input causes the current row and column position to be stored internally. The MCU updates the LPEN ROW and LPEN COL locations in command block at the next end of frame and sets the LPU status bit. Further updates of these command block locations are inhibited until another LPEN ENABLE command is issued.

EOF: End of Frame

DBOR: End of Row

LPU: Light Pen Update

DUR: Data Under Run

DBOR: Data Buffer Over Run

This status bit is set when the MCU tries to fill a row buffer beyond its capacity. The MCU will stop fetching characters after this point and the display is blanked following the completion of the row currently being displayed.

EOF: End of Frame

This bit is set by the DG at the physical end of the nth frame, where 'n' is specified by the MODESET parameter FRAME INTERRUPT COUNT. This provides the means for timing frame related events such as slow scrolls.

FDE: Frame Data Error

This status bit is set by the DG at the physical end of frame if no EOS datastream command has been encountered until then. This also results in the execution of the EOS command by the MCU.

RCC: Reserved Channel Command

This bit is set by the MCU upon encountering an illegal datastream or command process command. This can be used to trap software errors during program development.

RDC: Reserved Datastream Command

This bit is set by the MCU upon encountering an illegal datastream or command process command. This can be used to trap software errors during program development.

DIP: Display In Progress

This bit is set by the MCU immediately after receiving a "Start Display" channel command. It remains set as long as the display process is active and is reset upon receiving a "Start Virtual Display" or "Stop Display" command or a Reset. Interrupts cannot be enabled for this status bit.

VDIP: Virtual Display In Progress

This bit is set by the MCU immediately after receiving a "Start Virtual Display" channel command and is reset upon receiving a "Start Display" or "Stop Display" command or a Reset. This bit remains active as long as the virtual display process is active. Interrupts cannot be enabled for this status bit.

Interrupt Processing

The system processor can enable interrupts on any of the status bits, with the exception of DIP and VDIP bits, by specifying an interrupt mask. A "1" in a bit position in the interrupt mask disables (masks out) interrupts on the status bit located in the corresponding bit position in the status word. The format for Interrupt Mask is shown below. The Int Mask can be loaded into 82730 from the INTMASK location in command block by a "Load Int Mask" channel command.

15 7	6	5	4	3	2	1	0
(RESERVED)	RDC INT MASK	RCC INT MASK	FDE INT MASK	EOF INT MASK	DBOR INT MASK	LPU INT MASK	DUR INT MASK

INT MASK = 0 Enables the corresponding interrupt.

INT MASK = 1 Masks or disables the corresponding interrupt.

Figure 8. Interrupt Mask

If the interrupt is enabled for a particular status bit by programming a "0" in the corresponding bit position in INTMASK and if the status bit gets set during the course of the display, an interrupt will be generated by 82730 at the next end of frame. At the end of frame, the 82730 will first perform the tasks of updating LPEN position (if required) and servicing the Channel Attention (if CA was activated). Then the status word in the internal register will be written to the INT GENERATION CODE location in the Command Block and the SINT output will be activated. The SINT pin is not deactivated until an interrupt reset signal is received at the IRST pin.

82730 continues to perform its normal display task activating the SINT pin. If no interrupt reset is received until the next end of frame then any new interrupts that might have been generated at that end of frame will be lost. Therefore, it is essential for the system processor to issue an interrupt reset within a frame time after an interrupt is generated.

When the display is not activated, the only interrupt that can occur is the Reserved Channel Command interrupt. Upon receiving an invalid channel command, 82730 will write the status word to INT Generation Code location in the Command Block and activate SINT output, if that interrupt is enabled.

The processor can use the interrupt capability to get status information from 82730. A possible interrupt service routine for the system processor is shown in flow chart form in Figure 9.

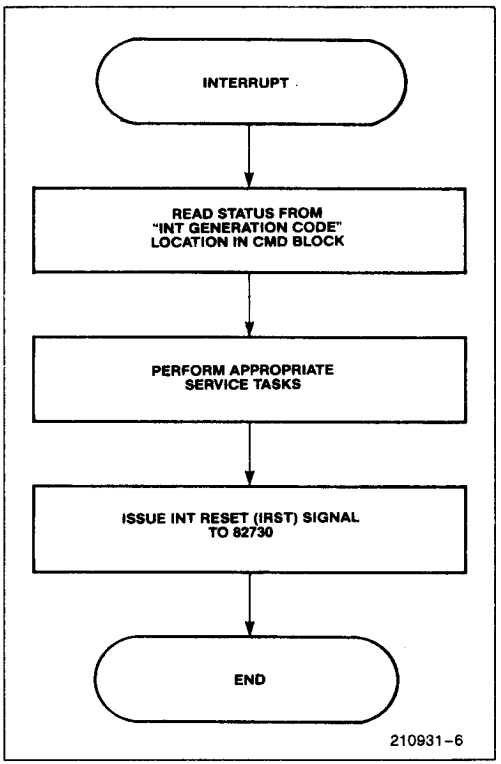


Figure 9. Interrupt Service Routine For System Processor

82730 VIDEO INTERFACE

The Mode Pointer in the Command Block points to a parameter block containing the Mode information required for the display. The organization of the mode words in the Mode Block is shown below.

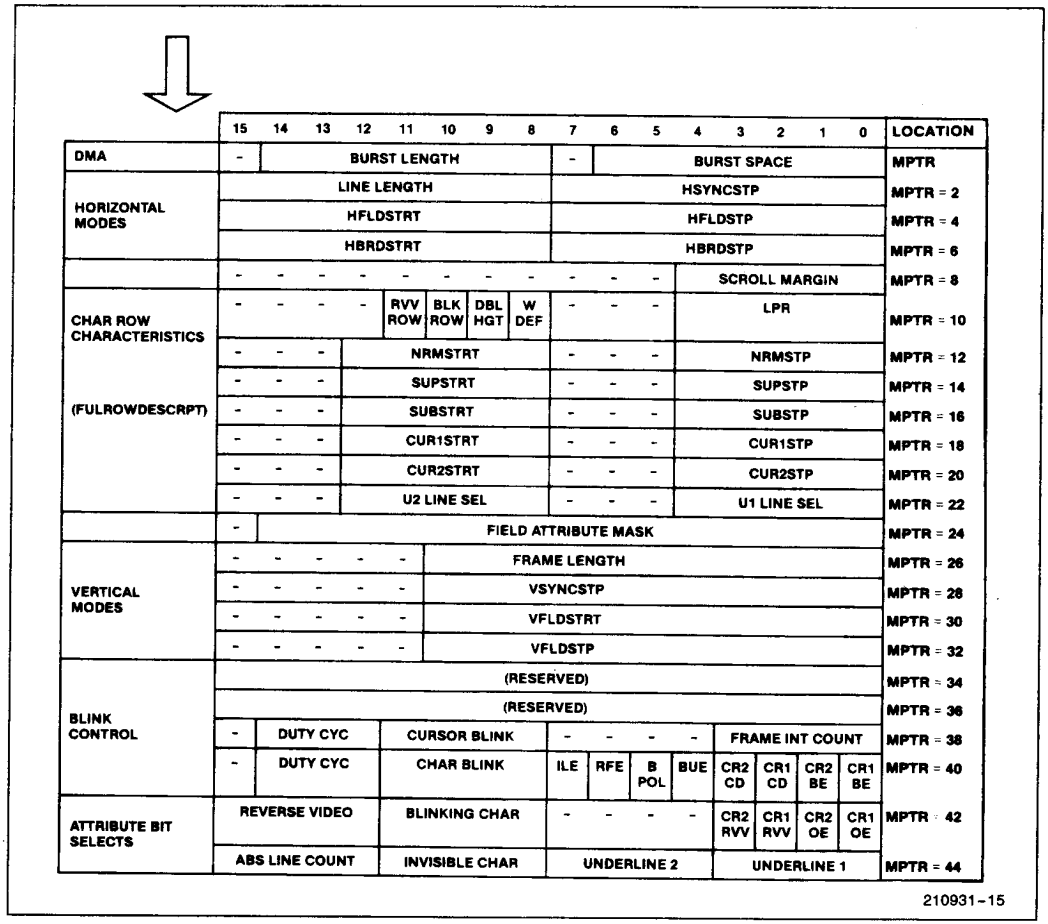


Figure 10. Mode Block Organization

CAM ARRAYS

Three Content Addressable Memory arrays are used for generating timing parameters to control the video display: the HORIZ MODE CAM, the VERT MODE CAM and the CHAR ROW CAM. The user has the flexibility to define his own timing parameters by loading them into the CAM arrays via the MIU. All of these parameters can be modified at the end of every frame. All the parameters in the CHAR ROW CAM, except MARGIN, are changeable on a row by row basis. Each of the three CAM arrays is described separately below:

Timing Sources

RCLK and CCLK inputs are provided by the external video logic to the 82730. The RCLK is used to increment the HORIZ COL CNTR and hence generates all horizontal timing parameters. CCLK is used to clock the character and attribute data output from the 82730 to the external display dot logic. Data changes on the positive going edge of RCLK or CCLK.

Initialization

Upon activation of the RESET input, the 82730 display generator will stop all operations in progress and deactivate all outputs. It will stay in this quiescent state until the MIU executes the MODESET command. The following table shows the states of all the Display Generator outputs during and after RESET.

Pin Name	Condition
DAT0–14	Low
WDEF	Low
LC0–4	High
BLANK	Low
CSYNC	High
$\overline{\text{CHOLD}}$	High
HSYNC	Low
VSYNC	Low
CRVV	Low
RRVV	Low

After reset of the 82730, the CAM arrays are in undetermined states. The CAM arrays are set upon the execution by the MIU of the MODESET command. The HORIZ and VERT MODE CAM contents are especially critical since they are used to generate timing control signals to the external video logic. Without the generation of the timing signals, no display process can take place. Hence, START DISPLAY command cannot be executed before the first MODESET command after the device reset. The START DISPLAY command will be ignored if it precedes the MODESET command.

The row buffers also contain unknown information after power up and reset. In executing the START DISPLAY command, the MIU would first load the two row buffers with the first two rows of character data to be displayed. Upon completion of loading of both buffers, it will signal the DG to begin the display process. In this way, only valid character data will be output to the external video logic.

Timing Parmeters

The timing parameters read from the MODESET Block and stored in the VERT MODE CAM and HORIZ MODE CAM are used to control the video display and they can be best illustrated in the following Map of Timing Parameters. All of these timings have to be defined after power up and reset and can be changed on a frame by frame basis during display.

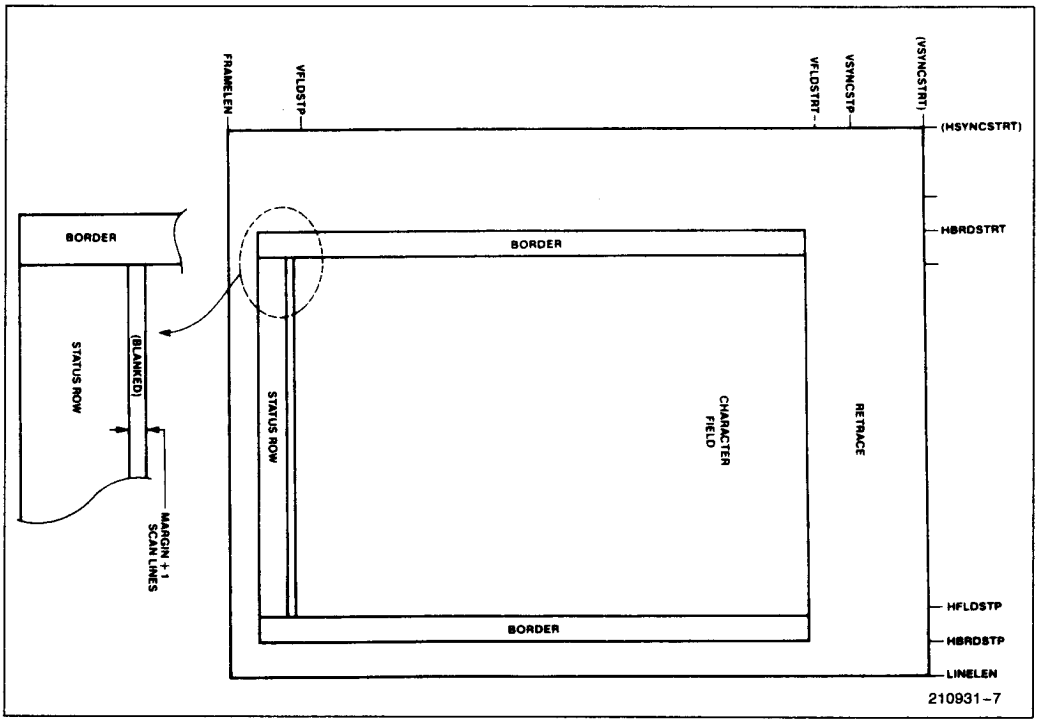


Figure 11. Timing Parameters

Row Timing Parameters

The row timing parameters are stored in HORIZ MODE CAM and are programmable from 0 to 255 RCLK times. These parameters are:

a) HSYNCSTRT—Horizontal Sync Start. The RCLK count on each scan line where HSYNC pin is activated. This parameter is not programmable. The RCLK period that follows the rising HSYNC edge is defined as column zero. It is used as the reference for all other horizontal timing parameters.

b) HSYNCSTP—Horizontal Sync Stop. The RCLK count on each scan line where the HSYNC pin is deactivated. The falling edge of HSYNC occurs at the leading edge of the programmed RCLK period.

c) LINELEN—Line Length. This parameter defines the total number of RCLK's in each scan line including display time, border and horizontal retrace time. There are LINELIN +1 RCLK periods per horizontal line scan.

d) HBDRSTRT—Horizontal border start. The RCLK count on a scan line where the border begins. The border begins at the leading edge of the programmed RCLK period.

e) HBDRSTP—Horizontal Border Stop. The RCLK count on a scan line where the border ends. The border terminates at the leading edge of the programmed RCLK period.

f) HFLDSTRT—Horizontal Field Start. The RCLK count on a scan line where the character display field begins. If the row buffer is ready to be displayed, the CSYN pin will be deactivated at this point. This field begins at the leading edge of the programmed RCLK period.

g) HFLDSTP—Horizontal Field Stop. The RCLK count on a line where the character display field stops. When this timing point is reached, CSYN will be activated. This field ends at the leading edge of the programmed RCLK period.

There is also one pseudo parameter, SYNCDLY. It is fixed at one half LINELEN and is used as the start and end timing for VSYNC in odd frames in interlaced displays. VSYNC starts at HSYNCSTRT in even frames for interlaced displays and all frames for non-interlaced displays.

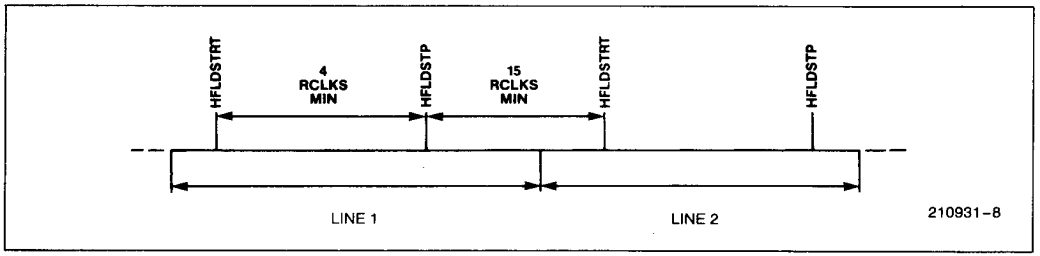


Figure 12. Horizontal Timing Restrictions

There are certain restrictions in the programming of HFLDSTRT and HFLDSTP and those restrictions are best illustrated below. There has to be at least 4 RCLKS in between HFLDSTRT and HFLDSTP of the same scan line and 15 RCLKS in between HFLDSTP of one line and HFLDSTRT of the next. The minimum delay of 15 RCLKS is for the charging of the pipeline from the row buffer to the character data output DAT0–DAT14 as well as the setting of the correct value for the scan line output LC0–LC4.

Frame Timing Parameters

Frame timing parameters are stored in the VERT MODE CAM and are programmable from 0–2047 scan lines. These parameters are:

a) VSYNCSTRT—Vertical Sync Start. The line count where the VSYNC is activated. This occurs at the end of a field automatically. This parameter is not programmable. The rising edge of VSYNC occurs with the rising edge of HSYNC for all non-interlace fields and for odd fields in the interlace mode.

b) VSYNCSTP—Vertical Sync Stop. The line count at which the VSYNC pin is normally deactivated. VSYNC changes at the rising edge of HSYNC normally. However it occurs at SYNCDLY at the beginning of odd fields of an interlaced display.

c) FRAMELEN—Frame Length. This parameter defines the total number of scan lines per frame. It is used to reset the FRAME LINE CNTR. In an interlaced display, FRAMELEN must be an even number. If an odd number is programmed, one additional line will occur automatically. There will be FRAMELEN + 1 scan lines per frame. (Note that interlace mode contains two fields per frame).

d) VFLDSTRT—Vertical Field Start. Programs the scan line count where the character display field begins.

e) VFLDSTP—Vertical Field Stop. Programs the scan line count where the regular character display field ends. VFLDSTP times the beginning of the Status Row. The channel attention sequences, interrupt handling, row buffer swap and initialization for the next frame are started after the display of the Status Row is completed. See * below.

***NOTE:**
(Character Field Boundary definition: The starting or ending event is defined to occur at HFLDSTP on the scan line following the programmed value. Thus the visible character field effectively begins two scan lines below the programmed start value and ends one scan line below the programmed stop value.)

Status Row

The Vertical Frame Timing Parameters have no border controls, unlike the Horizontal Row Timing Parameters. The top and bottom borders can be replaced with regular display rows that are video-reversed and contain no data. The top border is easily timed from VFLDSTRT. The bottom border is more difficult without help from the Vertical Timing generators. If there were no help, the user would have to keep track of the number of scan lines used in each row to know when to stop regular display and create the bottom border. This would also preclude his ending his regular display with an EOF command before the border.

The 82730 provides this help with the Status Row feature. The display of the Status Row is timed from VFLDSTP and allows the user to display a row in a fixed position at the bottom of the screen that is independent of the regular data and any display errors (display ended by an EOF command or the DURN, DBOR, or FDE errors). (There is one dependency on the regular display data: the row format. The last FULROWDESCPT (FRD) set in the regular data will be used on the Status Row unless a new command is issued for the row. It is recommended that the user include a new FRD command in the Status Row data to eliminate this dependency).

Status Row display starts SCROLL MARGIN plus one scan line after VFLDSTP. This margin is provided to insure enough DMA time if the regular display runs up to VFLDSTP. The user can create a bottom border or any end-of-display row that he chooses. A

display status or system status line, or special programmable key function definition line can be implemented with this feature.

CHARACTER ATTRIBUTES

The 15 bits of the character word can be partitioned into character address and attribute bits.

Some common attributes may be individually defined and enabled or disabled by fields in the attribute parameter registers. Each attribute has two means of being enabled. The enable bits defined below are set during the MODESET channel command and are used as a global enable. The user does not have to enable the provided attributes. He may free more data bits for his own use this way. The second enable bit is contained in each character loaded to the row buffer to enable the attribute on a character by character basis. They are individually described in detail in the following sections.

Reverse Video

When a character with the reverse video attribute is displayed, the CRVV pin will be inverted during the time the character is being displayed. The reverse video affects the entire height of the row for that character space. For superscript/subscript pairs, the reverse video effect is controlled by superscript until SUBSTRT when the subscript attribute bit takes control. The parameter for this attribute is:

RVBS—Reverse Video Bit Select. This parameter selects one of the 15 bits of a character data word. Values 0 through 14 select the corresponding bit. Value 15 disables the Reverse Video attribute.

Blinking Character

When a character with the blinking character attribute is displayed, the BLANK pin will be activated and deactivated during the character display time according to programmable rate and duty cycle. The parameters for this attribute are:

a) BCBS—Blinking Character Bit Select. Selects one of the 15 bits of a character data word as the blinking character attribute control. As with Reverse Video above, the value of the select determines the controlling bit or disables the attribute.

b) CHAR BLNK FREQ—Selects one of the 32 blinking frequencies available for the blinking character and blinking underline. The character blink rate is calculated as below:

$$\text{Blink Rate} = \frac{\text{Frame Refresh Rate}}{4 \times \text{CHAR BLNK FREQ}}$$

c) CHAR DUTY CYCLE—A 2-bit register to select 4 duty cycles available for blinking character and blinking underline. The selection logic is defined to be as follows:

00 = 100% always on
11 = 75% on
10 = 50% on
01 = 25% on

Underline #1

When a character with underline is displayed, the BLANK Pin will be activated and the CRVV pin will be inverted during the time the scan line specified by the underline select register is displayed. The parameters used to define underline #1 are:

a) ULS1—Underline Line Select 1. It determines which scan line of a character row will be used for the underline #1. This parameter is modifiable on a row by row basis by the FULROWDESCRPT command.

b) ULBS1—Underline Bit Select 1. This parameter can only be changed by MODESET. It selects one of the 15 bits of a character data word as the underline #1 attribute control. Again, a value of 15 in the select field disables this attribute.

Underline #2 (Blinking)

Underline #2 can be made to blink. When its blinking feature is deactivated, its visual effect is exactly the same as underline #1. When it is enabled to blink, its blink rate and blinking duty cycle are the same as those defined for blinking character. The parameters used to define this attribute are:

a) UL2SEL—Underline Line Select 2. This parameter determines which scan line of a character will be the 2nd underline. It is changeable on a row by row basis by the FULROWDESCRPT command.

The next two parameters can only be modified by the MODESET Command.

b) ULBS2—Underline Bit Select 2. Selects one of the 15 bits of a character data word or GPA1 as the second underline attribute control. A bit select value of 15 disables the second underline.

c) BUE—Blinking Underline Enable. Activation of this bit will cause the second underline attribute to start blinking.

Invisible

A character with this attribute will occupy its character position on the screen but will not be displayed (i.e. BLANK will be active). This attribute does not

affect the Reverse Video attribute if they are programmed together. The parameter that is used to implement this attribute is:

IBS—Invisible Bit Select. Selects one of the 15 bits of a character data word as the invisible attribute control. Value 15 disables the invisible attribute.

Absolute Line Cntr Attribute

This character attribute allows the display of special graphic characters, or may be used to upshift normal characters to implement displays with overlapping superscript and subscript fields. When a character with this character attribute enabled is being displayed, its LC0–LC4 pins will reflect the output from the CHAR ROW LNE CNTR which counts the absolute line count of a row. The activation of this attribute overrides the line count mode of both normal and subscript/superscript characters. The parameter used to select the attribute is:

ABS LINE BIT SEL. This four bit register selects one of the 15 bits of a character data word as the absolute line counter output attribute control. Select value 15 disables the ABS Line attribute.

Cursor Generation

The cursor characteristic parameters are changeable on a frame by frame basis by MODESET.

a) CUR FREQ—Cursor frequency. Selects the blinking frequency for both cursors. The selection logic is similar to CHAR BLNK FREQ

b) CUR DUTY CYCLE—Cursor duty cycle. Selects the blinking duty cycle for both cursors. Its selection logic is similar to CHAR DUTY CYCLE.

c) CR1RVV—Cursor 1 Reverse Video Enable selects a reverse video type cursor as opposed to a solid (blanking) cursor.

d) CR1BE—Cursor 1 Blink Enable changes the cursor 1 block or underline to a blinking block or underline. Enabling this bit also causes DAT 14 pin to "blink" as well, if the CR10E bit is set.

e) CR10E—Cursor 1 Output Enable reconfigures the DAT 14 pin to indicate when cursor 1 is active. CR20E enabled directs the cursor 2 signal to DAT 13 pin in a similar fashion.

f) CR1CD—Cursor 1 Light Pen Cursor Detect directs the CCLK cursor #1 position to be translated to its nearest equivalent RCLK position through the LPEN facility.

An identical set of parameters (c) through (f) is available for the generation of CURSOR 2. The two cursors share the same FREQ and DUTY CYCLE parameters.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 80°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with Respect to Ground ..... −1.0V to +7V
Power Dissipation ..... 3 Watts

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

NOTICE: Specifications contained within the following tables are subject to change.

D.C. CHARACTERISTICS T_A = 0°C TO 70°C, V_{CC} = 5V ±10%

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	+0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2 mA (1)
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
I_{CC}	Power Supply Current		400	mA	@ T_A = 0°C
I_{LI}	Input Leakage Current		10	μA	V_{IN} = 0 − V_{CC}
I_{LO}	Output Leakage Current		±10	μA	V_{OUT} = 0.45 − V_{CC}
I_{LCL}	LC0, LC1, LC2 Input Low Current	−125	−450	μA	V_{IN} = 0V Reset = "1" (2)
V_{BLI}	Bus Clock Input Low Voltage	−0.5	0.8	V	
V_{BHI}	Bus Clock Input High Voltage	2.0	V_{CC} + 1.0	V	
V_{CLI}	Character Clock Input Low Voltage	−0.5	0.8	V	
V_{CHI}	Character Clock Input High Voltage	2.2	V_{CC} + 0.5	V	
V_{RLI}	Reference Clock Input Low Voltage	−0.5	0.8	V	
V_{RHI}	Reference Clock Input High Voltage	2.2	V_{CC} + 0.5	V	

NOTES:
1. I_{OL} = 2.6 mA on the $\overline{S1}$ and $\overline{S0}$ pins.
2. Measured after at least 5 BCLK cycles after RESET = High.

A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5V ±10%. All timings in nanoseconds. CL = 50 pF.

82730 Bus Interface Input Timing Requirements

Symbol	Parameter	Min	Max	Units	Test Conditions
TCLCL	BCLK Cycle Period	125	2500	ns	
TCLCH	BCLK Low Time	52		ns	
TCHCL	BCLK High Time	52		ns	
TCH1CH2	BCLK Rise Time		30	ns	0.45C → 2.4V (1)
TCL1CL2	BCLK Fall Time		30	ns	2.4V → 0.45V (1)
TDVCL	Data In Set-Up Time	20		ns	
TCLDX	Data In Hold Time	5		ns	
TARYHCH	Async. READY Active Set-Up Time	35		ns	
TSRYHCL	Sync. READY Active Set-Up Time	20		ns	
TRYLCL	READY Inactive Set-Up Time	10		ns	
TCLRYX	READY Hold Time	20		ns	
TCTVCL	HLDA, RESET Set-Up Time	35		ns	
TCLCTX	HLDA, RESET Hold Time	10		ns	
TCAVCAX	CA Pulse Width	100		ns	
TRIVRIX	IRST Width	100		ns	
TRLLCH	LC_X Input Hold Time	5TCLCL		ns	(2)

NOTE:
2. Applies only to test mode invocation.

82730 Bus Interface Output Timing Response

Symbol	Parameter	Min	Max	Units	Test Conditions
TCLAV	Address Valid Delay	0	55	ns	
TCLAX	Address Hold Time	0		ns	
TAVAL	Address Valid to ALE/UALE Inactive	TCLCH − 30		ns	
TLLAX	Address Hold to ALE Inactive	TCHCL − 10		ns	
TCLAZ	Address Float Delay	TCLAX	45	ns	
TAZRL	Address Float to $\overline{RD}$ Active	0		ns	
TLHLL	ALE/UALE Width	TCLCH − 10		ns	
TCLLH	ALE/UALE Active Delay	0	45	ns	
TCHLL	ALE/UALE Inactive Delay	0	45	ns	
TCVCTV	Control Active Delay ($\overline{DEN}$, $\overline{WR}$, $\overline{AEN}$)	0	70	ns	
TCVCTXW	Control Inactive Delay ($\overline{WR}$, $\overline{AEN}$)	0	80	ns	
TCVCTXD	Control Inactive Delay ($\overline{DEN}$)	5	80	ns	
TCLDOV	Data Out Valid Delay	0	50	ns	
TCLDOX	Data Out Hold Time	0		ns	
TWHDX	Data Out Hold Time After $\overline{WR}$	TCLCL − 60		ns	
TCLHV	Hold Output Delay	0	85	ns	
TRLRH	$\overline{RD}$ Width	2TCLCL − 50		ns	
TCLRL	$\overline{RD}$ Active Delay	0	95	ns	
TCLRH	$\overline{RD}$ Inactive Delay	5	70	ns	
TRHAV	$\overline{RD}$ Inactive to Next Address Active	TCLCL − 40		ns	
TCLSIN	SINT Valid Delay	0	70	ns	
TRIHSIL	RINT Active to SINT Inactive		250	ns	
TCHSV	Status Active Delay	0	75	ns	
TCLSH	Status Inactive Delay	0	70	ns	
TWLWH	$\overline{WR}$ Width	2TCLCL − 40		ns	
TFLHL	Bus Float to HOLD Inactive	0		ns	

82730 Display Generator Input Timing Requirements

Symbol	Parameter	Min	Max	Units	Test Conditions
TRCHRCH	RCLK Cycle Period	100	2500	ns	
TRCHRCL	RCLK High Time	40		ns	
TRCLRCH	RCLK Low Time	40		ns	
TRRCK	RCLK Rise Time		30	ns	0.45V → 2.4V (1)
TFRCK	RCLK Fall Time		30	ns	2.4V → 0.45V (1)
TCCHCCH	CCLK Cycle Period	100	None	ns	
TCCHCCL	CCLK High Time	30		ns	
TCCLCCH	CCLK Low Time	40		ns	
TRCCK	CCLK Rise Time		30	ns	0.45V → 2.4V (1)
TFCCK	CCLK Fall Time		30	ns	2.4V → 0.45V (1)
TVCVCR	HSYNC, SYNCIN Set-Up Time	30		ns	
TCRVCX	HSYNC, SYNCIN Hold Time	10		ns	
TLPVCF	LPEN Set-Up Time	30		ns	
TCFLPX	LPEN Hold Time	10		ns	
TRCHCCH	CCLK/RCLK Skew During CSYNC	−10	10	ns	

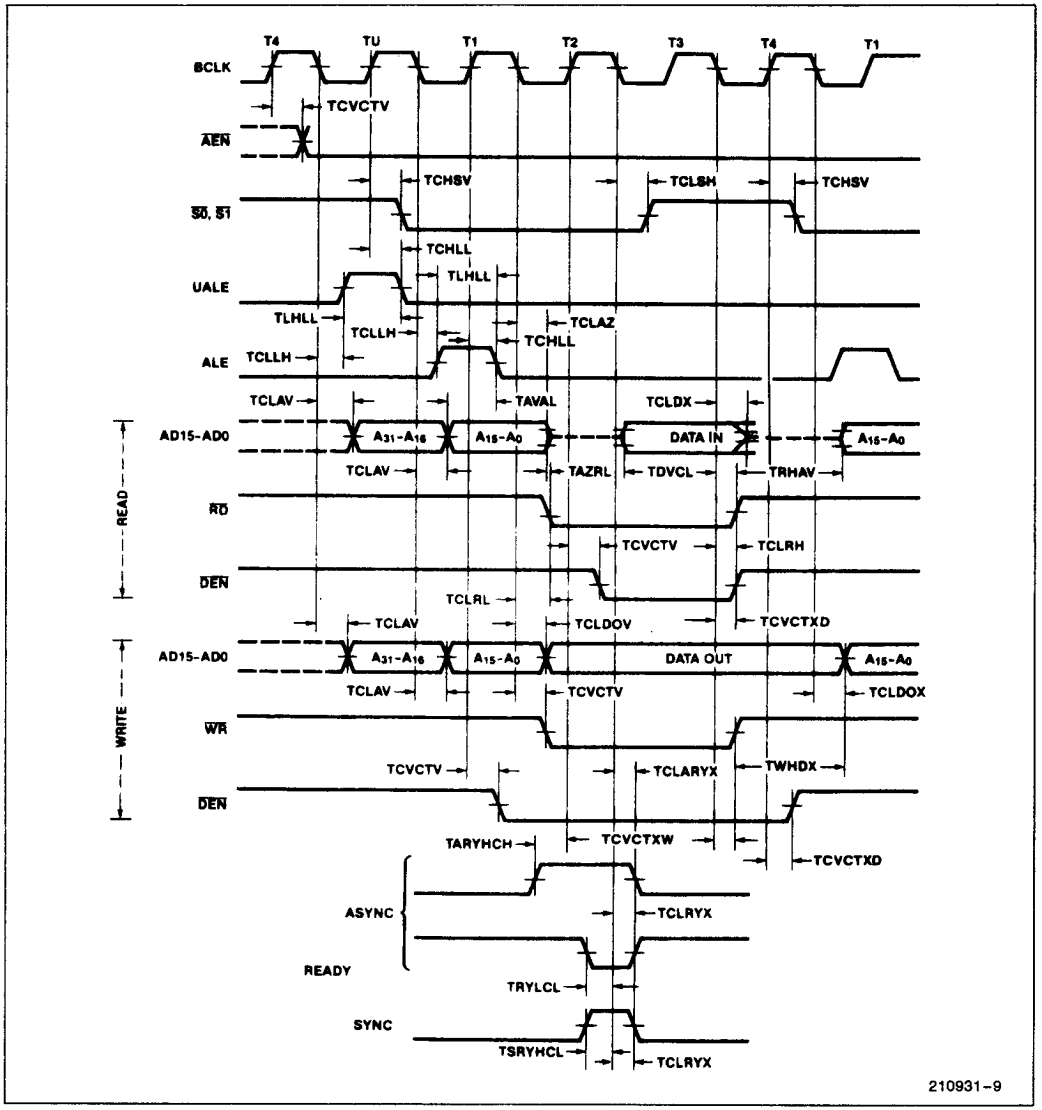
82730 Display Generator Output Timing Response

Symbol	Parameter	Min	Max	Units	Test Conditions
TCCHDV	Data, Line Count and Attribute and Output Valid Delay from the Rising Edge of CCLK		60	ns	C_L = 100 pF
TCCHDX	Data, Line Count and Attribute and Output Hold Time	5		ns	C_L = 100 pF
TRCHCV	Delay of Outputs CSYNC, VSYNC, HSYNC or RRVV from the Rising Edge of RCLK		70	ns	C_L = 100 pF
TCCHCL	CCLK Rising to $\overline{\text{CHOLD}}$ Low		75	ns	C_L = 50 pF
TRCLCH	RCLK Falling to $\overline{\text{CHOLD}}$ High		60	ns	C_L = 50 pF

NOTES:
1. Clock maximum rise and fall times are for functionality only. AC timings are not tested at this condition.
2. Applies only to test mode invocation.

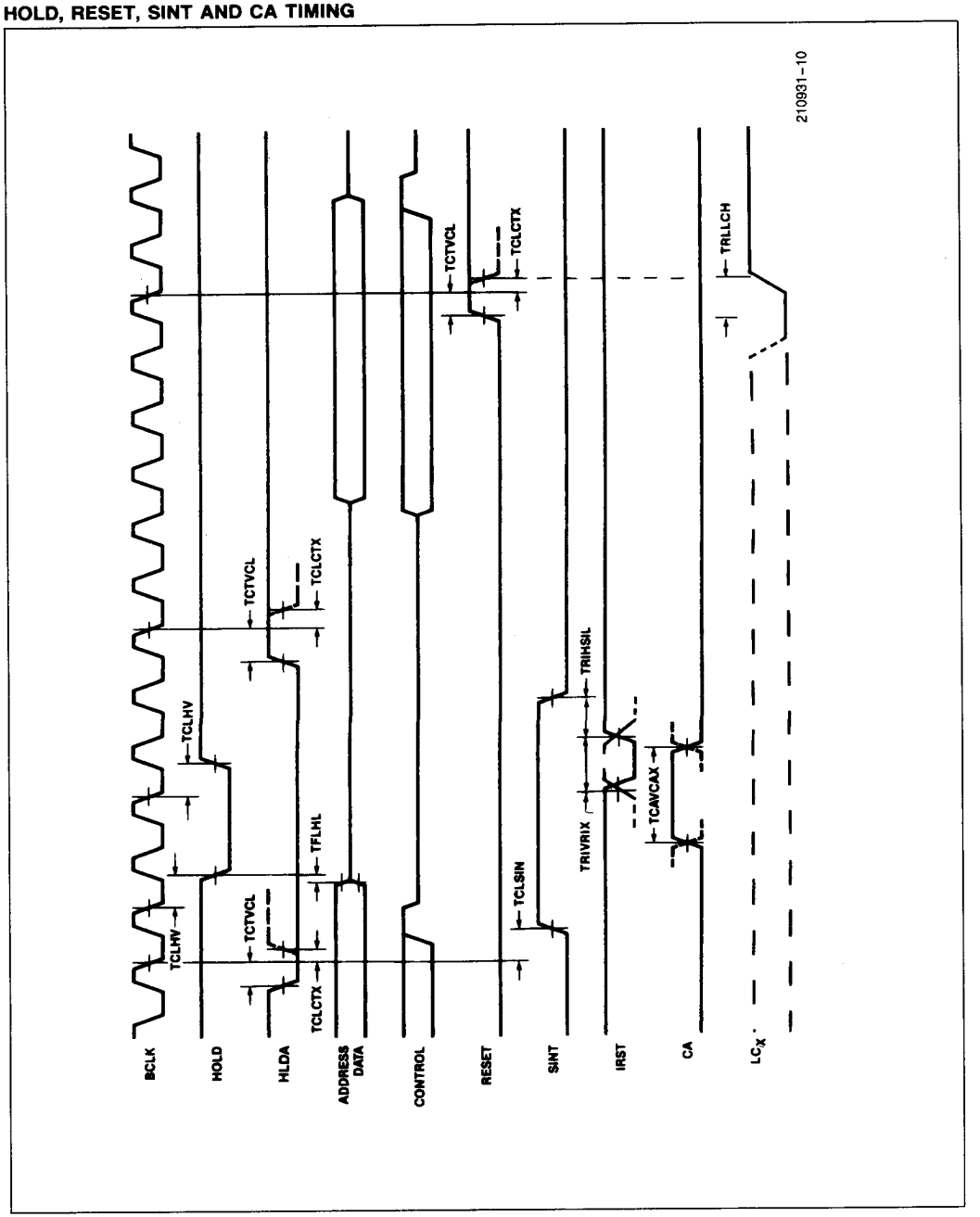
WAVEFORMS

BUS TIMING DIAGRAM



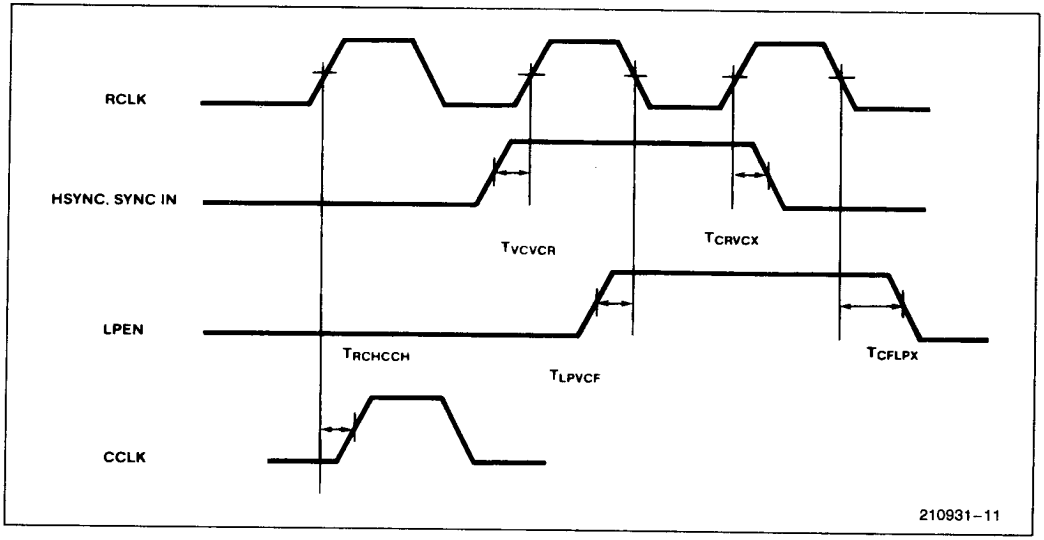
WAVEFORMS (Continued)

HOLD, RESET, SINT AND CA TIMING

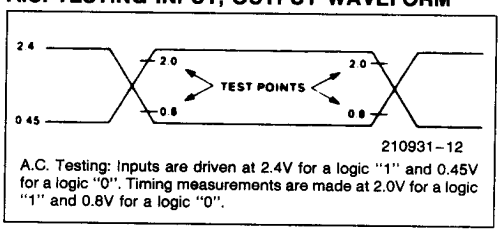


WAVEFORMS (Continued)

DISPLAY GENERATOR INTERFACE TIMING

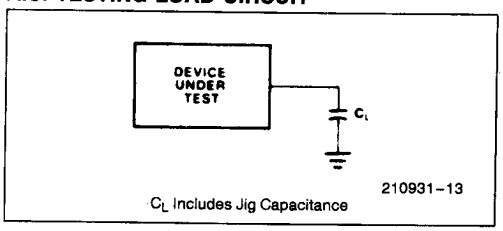


A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. Testing: Inputs are driven at 2.4V for a logic "1" and 0.45V for a logic "0". Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0".

A.C. TESTING LOAD CIRCUIT



C_L Includes Jig Capacitance

WAVEFORMS (Continued)

DISPLAY GENERATOR INTERFACE TIMING