



October 1986

# **82786 Hardware Configuration**

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office to obtain the latest specifications before placing your order.

The following are trademarks of Intel Corporation and may only be used to identify Intel Products:

Above, BITBUS, COMMputer, CREDIT, Data Pipeline, GENIUS, i, i<sup>+</sup>, ICE, iCEL, iCS, iDBP, iDIS, i<sup>2</sup>ICE, iLBX, i<sub>m</sub>, iMDDX, iMMX, Insite, Intel, int<sub>e</sub>l, int<sub>e</sub>IBOS, Intelelevision, int<sub>e</sub>lligent Identifier, int<sub>e</sub>lligent Programming, Inteltec, Intellink, iOSP, iPDS, iPSC, iRMX, iSBC, iSBX, iSDM, iSXM, KEPROM, Library Manager, MAP-NET, MCS, Megachassis, MICROMAINFRAME, MULTIBUS, MULTICHANNEL, MULTIMODULE, ONCE, OpenNET, PC-BUBBLE, Plug-A-Bubble, PROMPT, Promware, QUEST, QueX, Quick-Pulse Programming, Ripplemode, RMX/80, RUPI, Seamless, SLD, UPI, and VLSiCEL, and the combination of ICE, iCS, iRMX, iSBC, iSBX, iSXM, MCS, or UPI and a numerical suffix 4-SITE.

Ethernet is a trademark of Xerox.

MDS is an ordering code only and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corporation.

\*MULTIBUS is a patented Intel bus.

Additional copies of this manual or other Intel literature may be obtained from:

Intel Literature Inquiries  
SC6-58  
P.O. Box 58065  
Santa Clara, CA 95052-8065

---

# 82786 HARDWARE CONFIGURATION

## CONTENTS

PAGE

<b>1.0 INTRODUCTION</b> .....	1
<b>2.0 OVERVIEW</b> .....	1
2.1 Dedicated Graphics Memory .....	1
2.2 System Bus Interface .....	1
2.3 Video Interface .....	2
2.4 82786 Internal Registers .....	3
<b>3.0 DEDICATED GRAPHICS MEMORY INTERFACE</b> .....	3
3.1 DRAM Configurations .....	5
3.2 DRAM Timing Parameters .....	9
3.3 Initializing the DRAM Controller .....	11
<b>4.0 SYSTEM BUS INTERFACE</b> .....	13
4.1 Memory Map .....	14
4.2 BIU Registers .....	18
4.3 80286 Synchronous Interface .....	23
4.4 80186 Synchronous Interface .....	27
4.5 Asynchronous Interface .....	27
4.6 Multiple 82786 Interface .....	30
<b>5.0 VIDEO INTERFACE</b> .....	30
5.1 Various CRT Interfaces .....	30
5.2 CRTs with TTL-level Inputs .....	33
5.3 CRTs with Analog Inputs .....	33
5.4 Using a Color Palette RAM .....	35
5.5 Using the Window Status Signals .....	37
5.6 Higher Resolutions .....	39
5.7 Multiple 82786s .....	42
5.8 Video DRAMs .....	43
5.9 External Character ROM .....	44
5.10 Combining the 82786 With Other Video Sources .....	45
5.11 Other Types of Displays and Printers .....	47
5.12 Calculating the Video Parameters .....	48
5.13 A Spreadsheet for Calculating Video Parameters .....	53
<b>APPENDIX A—SAMPLE INITIALIZATION CODE</b> .....	A-1



## 1.0 INTRODUCTION

The 82786 is an intelligent co-processor capable of creating and displaying high performance graphics. Both drawing and display functions are integrated into a single VLSI chip to provide an inexpensive solution for bit-mapped graphics subsystems.

This application note is intended to show, through examples, use of the 82786 and the hardware interfaces between the 82786 and the rest of the system. Because the 82786 integrates many functions onto one chip, the hardware design of a graphics system is greatly simplified.

## 2.0 OVERVIEW

Internally, the 82786 consists of two independent processors.

- Graphics Processor: executes high-level line drawing, character drawing and bit-block-transfer commands to create and modify bit-maps in memory
- Display Processor: displays portions of bit-maps in regions on the CRT called windows

Figure 1 illustrates these processors and their hardware interfaces.

- Graphics Memory Interface: connects dedicated graphics memory to the 82786
- System Bus Interface: connects CPU and system memory to the 82786
- Video Interface: connects the 82786 to CRT or other display

The video interface is controlled directly by the Display Processor. The other interfaces are controlled by the

82786 Bus Interface Unit (BIU). The BIU connects the internal Graphics and Display Processors to the CPU and system memory as well as to the graphics memory through the internal DRAM/VRAM controller.

## 2.1 Dedicated Graphics Memory

The dedicated graphics memory provides the 82786 with very fast access to memory without contention with the CPU and system memory. Typically, the bit-maps to be drawn and displayed, the character fonts, and the command lists for the 82786 processors are all stored in this memory. In some instances it is desirable to have the Graphics Processor command lists stored in system memory.

The 82786 contains a complete DRAM/VRAM controller on-chip which interfaces directly with a wide variety of DRAMs without external logic. This direct connection not only reduces chip count but also allows the 82786 to perform very fast burst accesses to the DRAMs. The DRAM/VRAM controller can take advantage of the quick burst-mode sequential accesses made possible by page-mode, fast-page-mode (sometimes called Ripplemode™), and Static Column DRAMs. In addition, interleaved DRAM/VRAM arrays are fully supported by the on-chip DRAM/VRAM controller allowing even faster burst access.

## 2.2 System Bus Interface

The system bus interface connects the CPU and its system memory to the 82786 and its graphics memory.

The most common 82786 configuration (shown in Figure 1) allows the CPU to access the system memory while the 82786 accesses its dedicated graphics memory simultaneously. It also allows the CPU to access the graphics memory and for the 82786 to access the system memory (but not simultaneously). The system bus

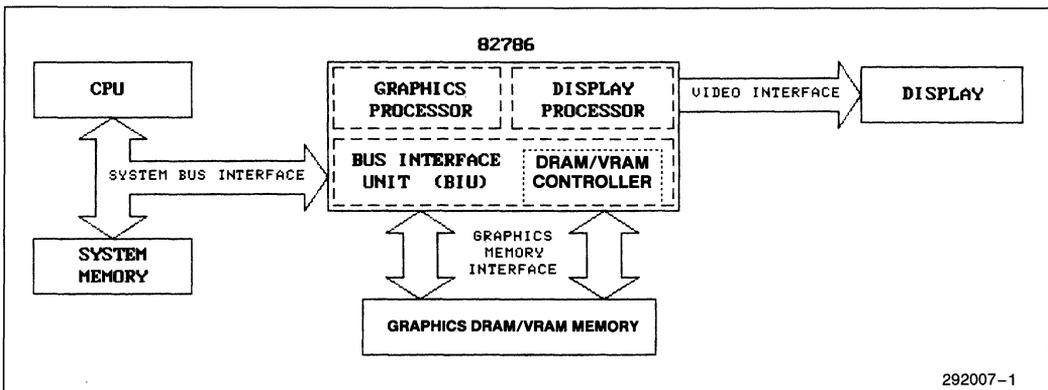


Figure 1. 82786 System Block Diagram

connects the 82786 graphics subsystem to the system CPU and memory. If DMA capability is also provided in the system, it interfaces to the 82786 exactly as the CPU does. The interface allows accesses in two directions.

- Slave Mode: CPU or DMA read or write access of the 82786 internal registers or dedicated graphics memory through the 82786
- Master Mode: 82786 read or write access to system memory

Therefore, any processor (CPU, DMA, Graphics and Display Processors) can access both the system memory and the graphics memory. The 82786 BIU arbitrates between both of the internal 82786 processors as well as the external processor (CPU and DMA) to decide which processor gets access of the bus.

The CPU software accesses both system and graphics memory in an identical manner (except that the specific memory addresses are different). Therefore the actual location of the memory (whether in system or graphics memory) is transparent to the software. However, the CPU can access the system memory faster than the graphics memory because there is less contention with the Graphics and Display Processors. When the CPU accesses the 82786, the 82786 BIU is said to be running in slave mode.

In slave mode, the 82786 looks like an intelligent DRAM/VRAM controller to the CPU (Figure 2). The CPU can chip-select the 82786 and the 82786 will acknowledge when the cycle is complete by generating a READY signal for the CPU.

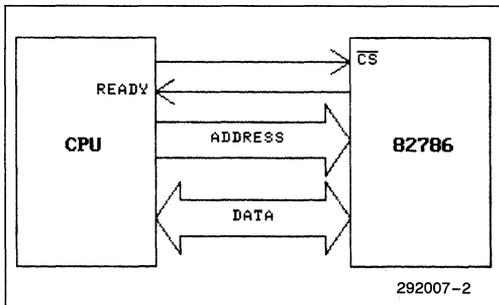


Figure 2. Slave Bus Cycle

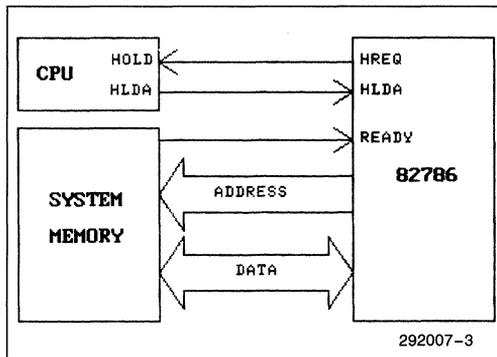


Figure 3. Master Bus Cycle

Conversely, the 82786 Graphics and Display Processors access both system memory and graphics memory in an identical manner. However, they can access the graphics memory faster than the system memory because there is less contention with the CPU. When the 82786 accesses the system memory, the 82786 BIU is said to be running in master mode.

In master mode, the 82786 looks like a second CPU controlling the local bus (Figure 3). The 82786 activates HOLD to request control of the system bus. When the CPU acknowledges the HLDA line, then the 82786 will take over the bus. When the 82786 is through with the bus, it will release HOLD and the CPU can remove HLDA to regain control of the bus.

The 82786 system bus interface is optimized to interface to an 80286 synchronously (using the same bus clock). As a synchronous slave it interprets the 80286 status lines directly and performs the requested bus accesses. As a master it generates 80286 style bus signals.

The 82786 system bus may alternatively be set up to interface asynchronously to virtually any processor. In this mode, read and write signals are used when slave accesses are performed.

### 2.3 Video Interface

The 82786 supports two different video interfaces in order to support both standard DRAMs and dual port video DRAMs/(VRAMs). When using standard

DRAMs the 82786 reads the video data from memory and internally serializes the video data to generate the serial video data stream up to 25 MHz. When using VRAMs the 82786 loads the VRAM shift register periodically and the internal system generates the serial video data stream.

With standard DRAMs displays up to 640 by 480 by 8 resolution can be generated at 60 Hz non-interlaced refresh. With VRAMs displays up to 2048 by 1936 by 8 can be generated at 60 Hz without interlacing.

In addition, horizontal and vertical sync signals and a blanking signal are provided and may be programmed to satisfy the requirements of nearly any CRT.

In the standard DRAM mode all of the logic to support the advanced capabilities of the Display Processor such as panning, zooming, windowing, and switching between various bits/pixel in various windows is contained internally in the 82786. Provision is also made for the addition of up to four external color look-up tables.

Higher resolution displays (dot clock rates greater than 25 MHz) can also be implemented by using external logic to trade-off bits/pixel for dot clock rates. Also, multiple 82786s can be used together for even greater performance.

### 2.4 82786 Internal Registers

A 64 word (128 byte) direct-mapped register block is contained internally in the 82786 (Figure 4). Software may locate this register block to the beginning of any 128 byte boundary anywhere in the 82786 I/O or memory address space. No matter where these registers are mapped, they are only accessible by the external CPU. The Graphics and Display Processors can not access these registers.

Registers, located at specified offsets within this block, allow programming of the BIU and Graphics and Display Processors. The Graphics and Display Processors also have other registers which are only accessible through commands to these processors. These commands are initiated by writing into the corresponding opcode and address registers within this 128 byte register block.

All of these registers are described in detail in the 82786 data sheet. Be careful when using "reserved" registers. When these reserved registers are read, the data returned is indeterminate. When these reserved registers are written, they should only be written as zeros to ensure compatibility with future products.

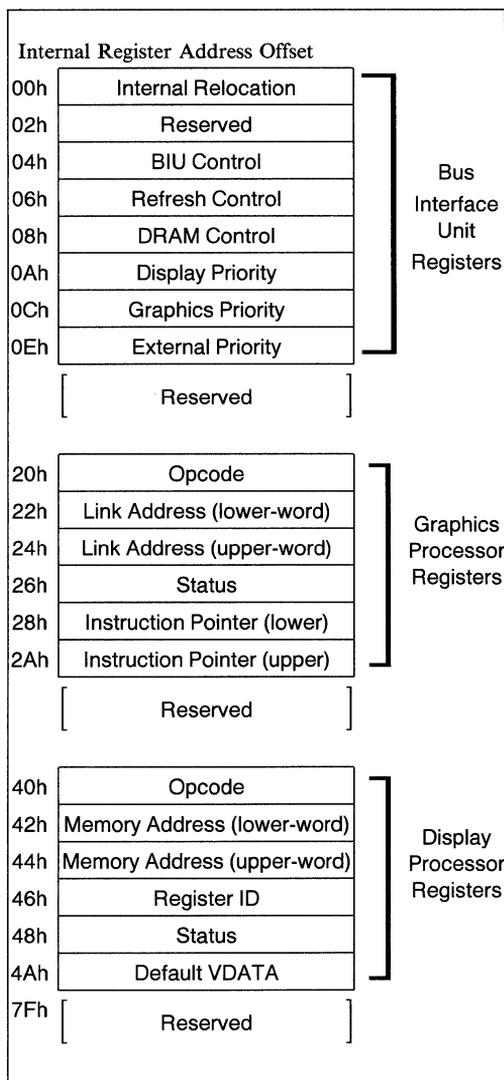
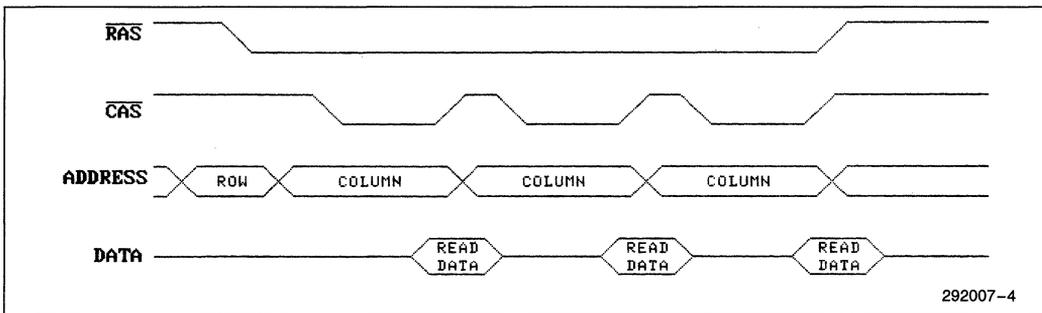


Figure 4. 82786 Internal Registers

### 3.0 DEDICATED GRAPHICS MEMORY INTERFACE

The 82786 contains a full DRAM/VRAM controller on-chip which allows it to be connected directly to arrays of DRAMs without external logic.

A wide range of DRAM configurations are possible for x 1, x 4 and x 8 bit wide DRAMs. Both Page mode and


**Figure 5. Fast-Page-Mode Burst-Access Read Cycle**

Fast-page-mode burst accesses for block transfers are supported directly by the 82786 to take advantage of the fast sequential addressing capability of DRAMs (see Figure 5). Once the DRAM is set-up with the row address, the column addresses can be quickly scanned in for several burst-accesses to the same page. With the 82786, fast-page-mode bursts for block transfers run at twice the speed of page mode.

Interleaving of two banks of DRAMs is also supported directly by the 82786. For a sequential burst access, DRAM cycles for both banks can be initiated. Then, during the burst access, the 82786 can alternate accesses between the two banks, thus cutting the effective DRAM access time in half (see Figure 6).

Static Column DRAMs can also be used to get the same performance as fast-page-mode. The only difference between the two types is that Static Column DRAMs do not latch the column address, whereas, fast-page-mode DRAMs do latch the column address on the falling edge of CAS. In non-interleaved configurations, Static Column DRAMs can directly replace fast-page-mode. However, in an interleaved configuration, the column address must be latched externally for Static Column DRAMs.

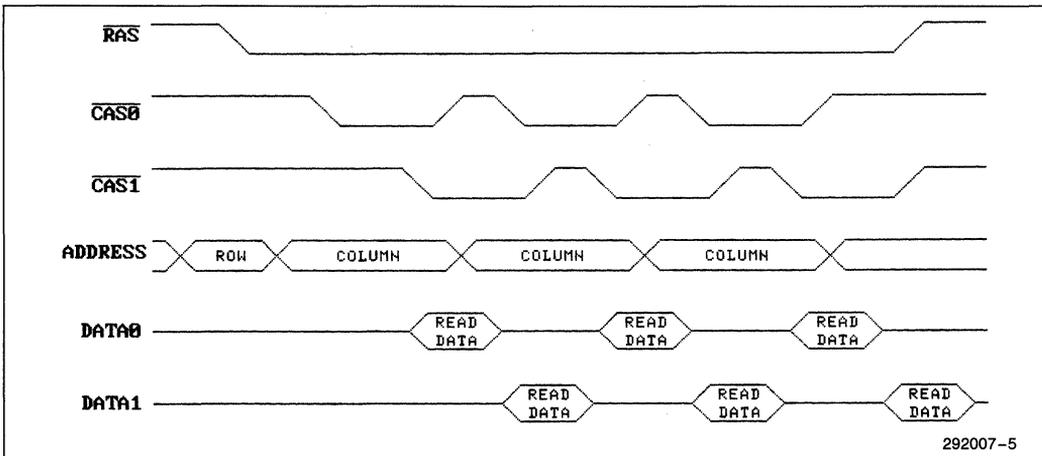
The following table shows the burst-access rate of these various configurations for a 10 MHz 82786.

	Page Mode	Fast-page-mode and Static Column
Non-interleaved:	10 Mbyte/sec (2 cycles)	20 Mbyte/sec (1 cycle)
Interleaved:	20 Mbyte/sec (1 cycle)	30 Mbyte/sec (0.5 cycle)

The other cycle times, and speeds at 10 MHz, are the same for all DRAM configurations:

Single Reads	3 cycles	300 ns
Single Writes	3 cycles	300 ns
Read-Modify-Writes	4 cycles	400 ns
Burst-Access Set-Up	2 cycles	200 ns
Refresh	3 cycles	300 ns

All burst-accesses for block transfers perform an even number of 16-bit word accesses.


**Figure 6. Interleaved Fast-Page-Mode Burst-Access Read Cycle**

Burst-accesses for block transfers are used by all Display Processor memory accesses except the operand for LD\_REG and DMP\_REG operands. Block-read accesses are used by the Graphics Processor for command-block fetching and to fetch the character fonts. The Graphics Processor uses a block-read followed by a block-write for the read-modify-write operations of BitBlt, Scan\_Line, and Character drawing. All other pixel drawing uses single read-modify-write cycles.

### 3.1 DRAM Configurations

Up to 4 rows per bank, and 1 non-interleaved or 2 interleaved banks are supported (see Figure 7). Each bank must always be 16 bits wide. If only one non-interleaved bank is used, it must be bank 0 (using CAS0 and BEN0). If interleaving is used, both banks must have the same number of rows. In either case, if only one row is used, it must be row 0 (using RAS0 and RAS1). Similarly, three rows use row 0, 1, and 2.

The 82786 can directly drive up to 32 DRAM/VRAM chips. One 82786 pin shares two DRAM functions DRA9/RAS3. These functions are never both used in the same configuration. DRA9 is only used by 1M x 1 DRAMs, which limit the number of rows to only two due to both addressing (4 Megabytes) and drive (32 chips) limitations.

Figure 8 shows a full connection diagram for thirty-two 64K x 4 DRAMs. Two interleaved banks of four rows each are used. Unlike most DRAM/VRAM controllers, no impedance-matching resistors are usually needed between the 82786 chip and the DRAM/VRAM chips. The impedance-matching for most configura-

tions is handled internally by the 82786. This is also the connections required for x4 VRAMs which use the BEN signal to control their DT/OE input which is used to determine when to load their internal shift register (Figure 9).

If Static Column DRAMs are used in an interleaved configuration, an external latch is required to latch the column address for the second bank (Figure 8a). The 82786 can directly drive up to thirty-two DRAM devices. For configurations requiring more than thirty-two devices, external buffering must be used.

DRAMs with separate data-in and data-out pins (such as the x 1 DRAMs) require a tristate buffer for the data-out lines of each bank. (All of the rows within each bank may share the same tristate buffer). Figure 10 shows a full connection diagram for thirty-two 256K x 1 DRAMs including the tristate buffers. Two interleaved banks of one row each are used. This is a special case for the RAS lines. Normally RAS0 would drive all of the DRAMs in both banks for the one row as in Figure 7. However, because the RAS lines have drive capability for only 16 DRAMs, both RAS0 and RAS1 are used. The 82786 recognizes this special case and automatically drives RAS1 identically to RAS0.

The other special DRAM case is using two rows of x 1 DRAMs in a non-interleaved configuration. This configuration has the advantage that only one bank of transceivers is required, but burst access time is reduced by half from the previous example. Normally, CAS0 would be used to drive all 32 DRAMs, but because of drive limitations, both CAS0 and CAS1 are used, (one for each bank). Again the 82786 recognizes this special case and automatically drives CAS1 identically to CAS0.

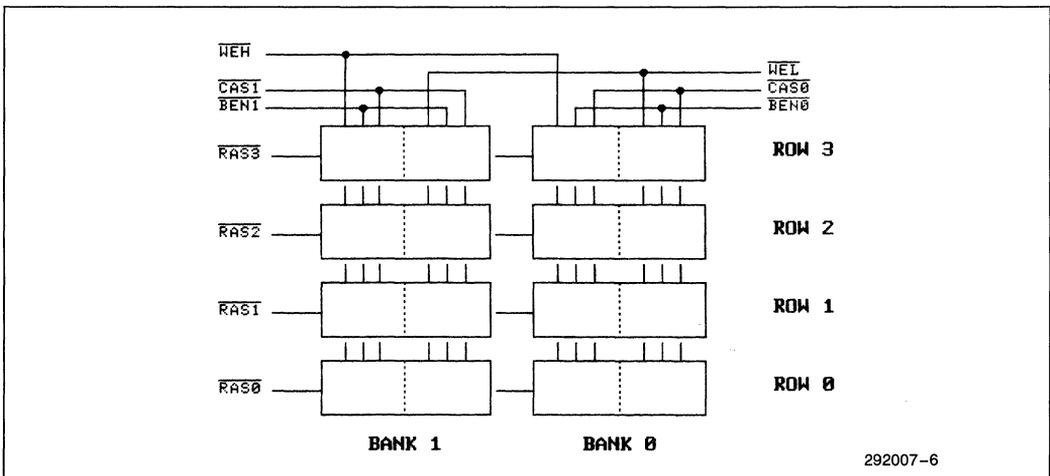
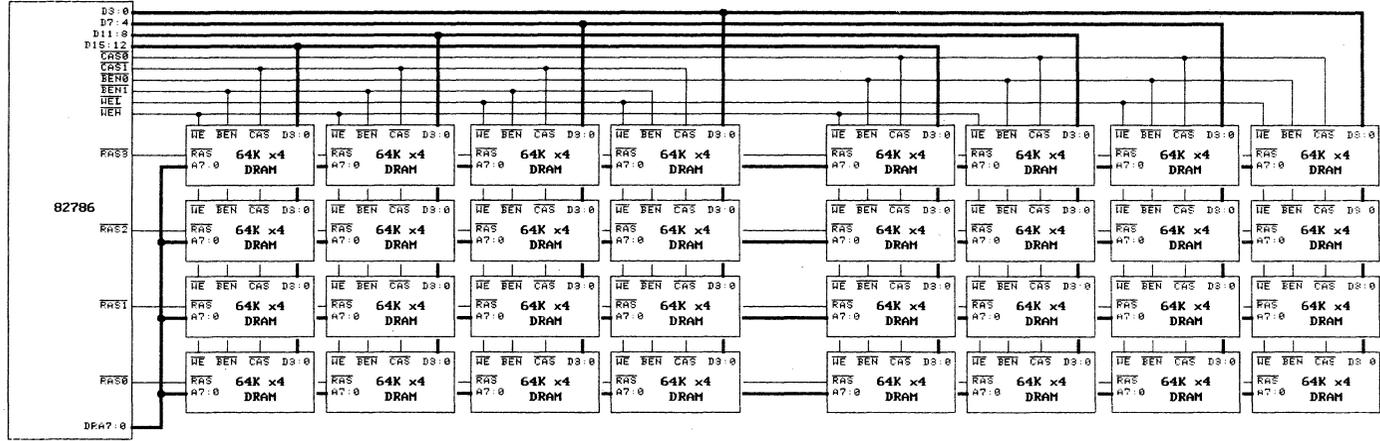


Figure 7. 82786 Supports up to 4 Rows of 2 Interleaved Banks of DRAMs  
64K x 4 Video RAMs with 82786 1 Row, 2 Banks, 4 Bits/Pixel



292007-7

Figure 8. 82786 Driving 4 Rows of Two Interleaved Banks of 64K x 4 DRAMs

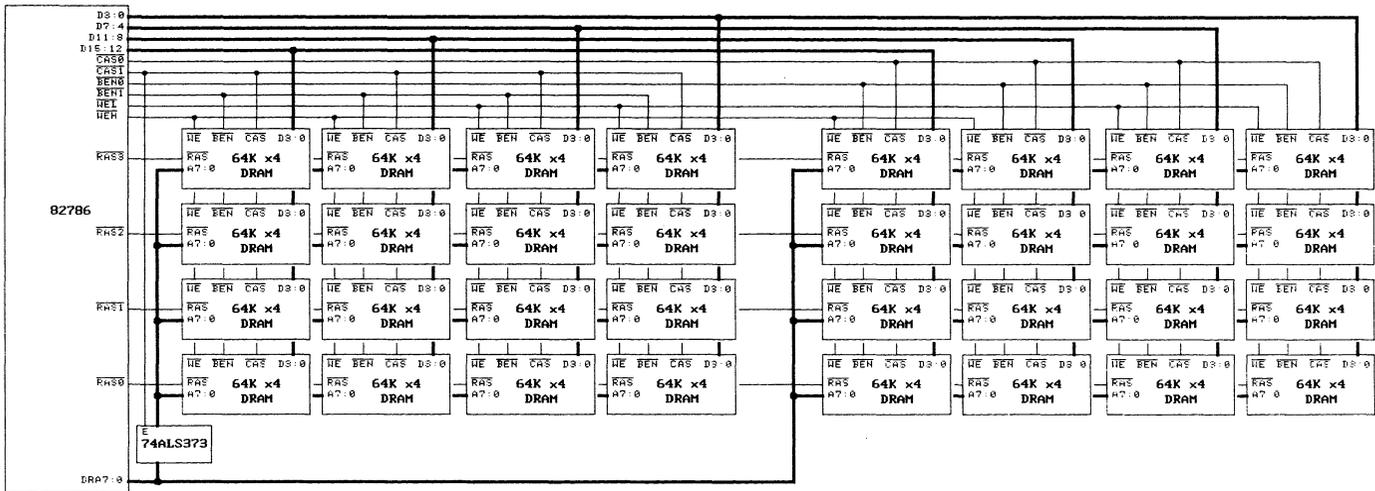
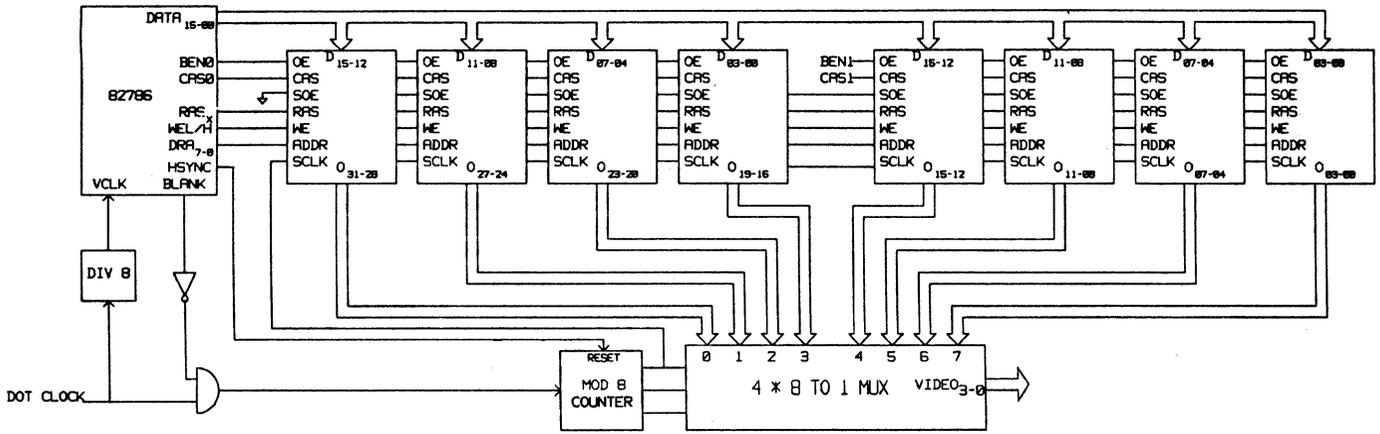


Figure 8a. 82786 Driving 4 Rows of Two Interleaved Banks of 64K x 4 Static Column DRAMs



292007-9

Figure 8b. 64k x 4 Video Rams with 82786 1 Row, 2 Banks, 4 Bits/Pixel

The table in Figure 11 shows all the possible configurations for 64K bit, 256K bit and 1 Mbit DRAMs.

### 3.2 DRAM Timing Parameters

Care should be taken to ensure that all of the timings of the DRAMs used, fit with those in the 82786 data sheet. To make the comparisons easier, the names of the parameter in the 82786 data sheet exactly correspond to the names in most DRAM data sheets. In addition, the parameters have been broken into the same four groups used by most DRAM data sheets.

The critical parameters for page mode DRAMs are generally:

Single rd/wrt/RMW	Single wrt	RMW	Page rd/wrt
Tcac Trp Trcd Trah Tasc Ton	Trwl Tcwl	Tds(rw) Toff	Tds(i)

Some of the 82786 parameters may not be found in all page-mode data sheets. If no corresponding DRAM parameters for Tcaa or Tcar is specified, then the 82786 spec may be ignored. The reason is that, if no such DRAM parameters exists, then the resulting minimum values for these parameters are at most:

$$Tcaa = Tasc + Tcac$$

$$Tcar = Tasc + Trsh$$

Then as long as the Tasc, Tcac, and Trsh specs fit, the 82786 timings guarantee Tcaa and Tcar to fit.

A third parameter that may not be found in all page-mode data sheets is Ton. If x 1 DRAMs are used, the external data transceiver is responsible for meeting this and the DRAM is not required to meet this spec. If, however, x 4 or x 8 DRAMs are used, without the data transceiver, care must be taken to ensure that this spec is met.

The critical parameters for Fast-page-mode and Static Column DRAMs are generally:

Single rd/wrt/RMW	Single wrt	RMW	Fast-page-mode rd/wrt
Trp Trah Tasc	Trwl Tcwl	Tds(rw) Toff	Tcp Tcaa Tcap Tds(n) Tcah(i) Tds(i) Tdh(i) Ton(ri)

For interleaved Static Column DRAMs, the address latch delay must be added to the DRAM parameters corresponding to the row and column addresses. These parameters are:

- Tasr
- Tasc
- Tcaa

For all types of x1 DRAMs, page-mode, Fast-page-mode and Static Column, the transceiver delay must be added to the DRAM parameters which correspond to read-data. These parameters are:

- Trac
- Tcac
- Tcaa

Notice that all of the 82786 DRAM timings are specified relative to the bus clock (CLK). This has two implications. First, a slower bus clock can be used to allow the 82786 to use slower DRAMs. Secondly, many of the parameters are determined by the duty cycle of the bus clock (as their specification is dependent on clock high or low time). A slightly non-symmetric clock, such as the clock for the 80286, can be used for the 82786 CLK, but care should be taken to examine the effects on the DRAM timing. In some circumstances, it may be advantageous to use a slightly non-symmetric clock.

Some of the specifications are relative to the 82786 clock period (Tc), while others are relative to a specific phase (THigh, TLow).

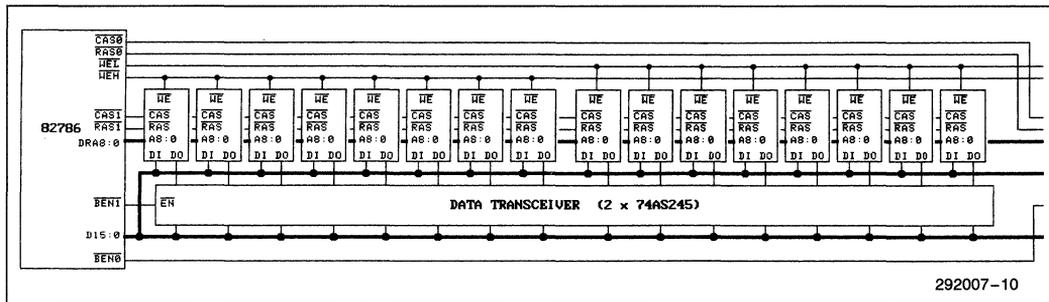


Figure 10. Two Interleaved Banks of 256K x 1 DRAMs

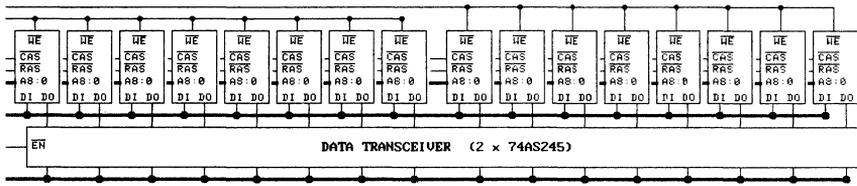
	Non-Interleaved				Interleaved			
	1-row	2-rows	3-rows	4-rows	1-row	2-rows	3-rows	4-rows
64K x 1	128K 16	256K 32	384K 48*	512K 64*	256K 32	512K 64*	768K 96*	1024K 128*
16K x 4	32K 4	64K 8	96K 12	128K 16	64K 8	128K 16	192K 24	256K 32
8K x 8	16K 2	32K 4	48K 6	64K 8	32K 4	64K 8	96K 12	128K 16
256K x 1	512K 16	1024K 32	1536K 48*	2048K 64*	1024K 32	2048K 64*	3072K 96*	4096K 128*
64K x 4	128K 4	256K 8	384K 12	512K 16	256K 8	512K 16	768K 24	1M 32
32K x 8	64K 2	128K 4	192K 6	256K 8	128K 4	256K 8	384K 12	512K 16
1M x 1	2M 16	4M 32	—	—	4M 32	—	—	—
256K x 4	512K 4	1M 8	1.5M 12	2M 16	1M 8	2M 16	3M 24	4M 32
128K x 8	256K 2	512K 4	768K 6	1M 8	512K 4	1M 8	1.5M 12	2M 16

\*Requires external buffering

Figure 11. Possible DRAM configurations for 64K, 256K and 1 Mbit DRAMs. The top number in each box is total memory size in bytes, the bottom is the number of DRAM chips required.

Look at this example. Suppose you use 51C256H Fast-page-mode DRAMs with the 82786 as in Figure 10. First, look at the critical parameters shown above. Since it is not possible to create a precisely 50% duty cycle clock, you must consider clocks with a few percent tolerance. The table compares the 82786 using several clock frequencies and duty cycle tolerances with two versions of the 51C256H. The table is ordered with the tightest timings first.

From the table, you can see that the fast 120 ns access DRAMs can be used with the 82786 with a 10 MHz clock with as much as a 40%–60% duty cycle skew. The slower DRAMs can be used at 9 MHz with a tighter 45%–55% duty cycle skew or at 8 MHz with a 40%–60% skew.



292007-49

**Figure 10. Two Interleaved Banks of 256K x 1 DRAMs (Continued)**

Parameter	82786 Specifications						51C256H DRAM Specs	
			10 MHz 45-55%	10 MHz 40-60%	9 MHz 45-55%	8 MHz 40-60%	- 12 120 ns	- 15 150 ns
Tdh	Min	Tph	22.5	20	25	25	20	25
Toff	Max	T1 + 3	25.5	23	28	28	20	25
Tcah	Min	Tch + 2	26.5	22	24.5	27	15	20
Tcp	Min	Tcl - 5	17.5	15	20	20	10	10
Tds	Min	Tcl - 8	14.5	12	17	17	0	0
Tcaa	Max	2Tc - 27	73	73	83	98	55	70
Tcap	Max	2Tc - 21	79	79	89	104	60	75
Tasc	Min	Tcl - 5	17.5	15	17.5	17.5	5	5
Trp	Min	2Tc - 5	95	95	105	120	70	85
Trwl	Min	Tc - 9	41	41	46	53.5	25	30
Tcwl	Min	Tc - 12	38	38	43	50.5	25	30
Trah	Min	Tc + 3	53	53	58	65.5	15	20
Ton	Max	Tc - 24	26	26	31	38.5	25	30

Because these x 1 DRAMs require transceivers between their data outputs and the 82786, the transceiver delays must also be considered. The two parameters in the table above, that are affected are Tcaa and Tcap. The transceiver delay must be added to the DRAM access time for these parameters. This implies that the data-in to data-out time of the transceivers must be 18 ns or less for the 10 MHz-120 ns case and the 8 MHz-150 ns case. The delay must be 28 ns or less for the 9 MHz-150 ns case and the 8 MHz-150 ns case.

### 3.3 Initializing the DRAM Controller

Two of the 82786 Internal Registers are used to configure the DRAM/VRAM Controller. Both of the regis-

ters are typically set once during initialization and then never changed. The DRAM/VRAM Control Register is set to indicate the configuration of the DRAMs/VRAMs used. The DRAM/VRAM Refresh Control Register is set to indicate the frequency of refresh cycles. Once programmed, the settings can be write-protected using the write-protect bits discussed in Section 4.2.

It is recommended that all fields of the DRAM/VRAM Control Register be written simultaneously to avoid illegal combinations. Also, no DRAM accesses should be attempted until the DRAM/VRAM Control Register has been set. For the configuration in Figure 10 using one row of 256K Fast-page-mode DRAMs in two interleaved banks:



$$\begin{aligned} \text{Request\_time} &= \frac{16 \times (\text{RefreshCount} + 1)}{\text{CLK}} \\ &= \frac{16 \times (18 + 1)}{20 \text{ MHz}} = 15.2 \mu\text{s} \end{aligned}$$

The amount of latency that the DRAMs will tolerate for each row is:

$$\begin{aligned} \text{Allowed\_Latency} &= \text{Tref} - (\text{RequestTime} \times \text{Refresh\_} \\ &\quad \text{Rows}) \\ &= 4 \text{ ms} - (15.2 \mu\text{s} \times 256) = 108.8 \mu\text{s} \end{aligned}$$

But the real latency limit is that the 82786 allows only three requests to be queued:

$$\begin{aligned} \text{Maximum\_Latency} &= \text{Queue\_Size} \times \text{Refresh\_Time} \\ &= 3 \times 15.2 = 45.6 \mu\text{s} \end{aligned}$$

Therefore, the maximum number of wait-states allowed for a 82786 master mode transfer is:

$$\begin{aligned} \text{Wait\_States} &= ((\text{Maximum\_Latency} \times \text{PCLK}) - \\ &\quad \text{overhead}) / \text{bus-cycles} \\ &= (45.6 \mu\text{s} \times 10 \text{ MHz}) - 7 \text{ cycles} / 2 = 224 \end{aligned}$$

Clearly, in this situation, refresh latency is not a problem. If the system memory caused the 82786 to delay over 224 wait-states for a master-mode access, not only would DRAM/VRAM refresh be missed, but the display refresh would also be lost.

The 82786 always issues three refresh cycles following a RESET. Besides these first three refresh cycles, the 82786 does not perform any other DRAM/VRAM warm-up after cold or warm-reset. If the DRAMs/VRAMs require other warm-up cycles, the CPU should either perform dummy cycles to the DRAM/VRAM or wait until the refresh counter has requested enough refresh cycles to occur.

If the DRAM/VRAM Refresh Control Register is set to all ones, refresh cycles are disabled.

## 4.0 SYSTEM BUS INTERFACE

The 82786 system bus structure allows the 82786 to be easily connected to a variety of CPUs. The 82786 can act as both a slave and a master to the CPU's bus. As a slave, the CPU or DMA can perform read and write cycles to the 82786 internal registers or to the 82786 DRAM/VRAM. As a master, the 82786 Graphics and Display Processors can perform read and write cycles to the CPU's system memory.

The 82786 bus can operate in three different modes to handle various CPU interfaces. The 82786 determines which mode to use by sampling the  $\overline{\text{BHE}}$  and MIO pins during RESET:

	$\overline{\text{BHE}}$	MIO
Synchronous 80286 bus	1	0
Synchronous 80186 bus	1	1
Asynchronous bus	0	X

For synchronous 80286 interfaces, the Reset and Clock inputs into the 80286 and 82786 must be common. For synchronous interfaces to 80186, the 80186 CLKIN must be the same as the 82786 CLK (so external clock source must be used). The  $\overline{\text{RES}}$  input into the 80186 must meet a set up and hold time with respect to the CLKIN. The RESET for the 82786 should be generated from the  $\overline{\text{RES}}$  (for 80186) by delaying  $\overline{\text{RES}}$  by one CLKIN cycle and inverting it. This ensures that the 82786  $\text{ph1}$  is coincident with 80186 CLKOUT low.

These pin states are easy to achieve for the synchronous modes. During RESET, the 80286 always drives  $\overline{\text{BHE}}$  high and MIO low.

CPUs with timings different from the 80286 must use asynchronous mode (however, CPUs such as the 80386 can easily generate 80286 style timings). Care should be taken in this case to ensure  $\overline{\text{BHE}}$  is low during RESET.

In each of these three modes it is possible to configure the 82786 to allow both master and slave accesses or to simplify the logic to allow only slave access. In the master mode, the 82786 always generates 80286 style bus signals.

If the 82786 is used as a master, it will activate its HREQ line when it needs to become the system master to access system memory. It waits until HLDA is received and then begins driving the system bus. Once HLDA is received, a 10 MHz 82786 can perform system bus accesses at the following rate (assuming 0 wait-states).

single reads/writes	4 cycles	5 Mbyte/sec
read-modify-writes	6 cycles	3.3 Mbyte/sec
burst-access read/write	2 cycles	10 Mbyte/sec

The 82786 will begin the first master-mode bus access on the cycle after HLDA is activated. The only delay is the time between when the 82786 activates HREQ and the system can release the bus and return HLDA. Most synchronous CPUs require a minimum of three cycles between the time HOLD is activated until they can return HLDA. The 82786 will keep HREQ activated until it no longer has more accesses to perform to system memory. (Until either the next 82786 access is to the dedicated graphics DRAM/VRAM or until neither the Graphics or Display Processors require the bus.) Once the 82786 is done using the system bus, it will remove HREQ and is able to immediately access its Graphics DRAM/VRAM on the next cycle.

It is potentially possible for the 82786 to require the system bus for a lengthy period of time. For example, if the 82786 has been programmed to give the Graphics Processor high priority, and the Graphics Processor executes a command that requires a lot of access to system memory, then the system bus could potentially be held by the 82786 for several consecutive accesses. Drawing a long vector into a bit-map residing in system memory is such a command. In this case, the maximum time the 82786 can potentially keep the system bus is determined by the frequency of DRAM/VRAM refresh cycles programmed into DRAM/VRAM Refresh Control Register.

If the CPU needs to regain control of the bus before the 82786 is done, it may remove HLDA early. The 82786 will then complete the current access and remove HREQ to indicate to the CPU that it may now take-over control of the bus. If the 82786 still requires more access to the system bus, it will re-activate HREQ two cycles after it had removed it and wait until the next HLDA. Since the 82786 removes HREQ for only two cycles, it is important that the CPU recognize it immediately. Otherwise a lock-out condition will occur in which the CPU is waiting for the 82786 to remove HREQ and the 82786 is waiting for the CPU to issue HLDA. This is not a problem for the synchronous interfaces. Extra logic may be required to prevent this situation if the 82786 is used as a master in an asynchronous interface and HLDA is ever removed prematurely, especially if the CPU clock is significantly slower than the 82786 clock.

### 4.1 Memory Map

Figure 12 shows the memory map as it appears to both the 82786 Graphics and Display Processors. These processors both use a 22-bit address which provides for up to 4 Megabytes of address space. They are only allowed to make memory accesses so no I/O map is applicable.

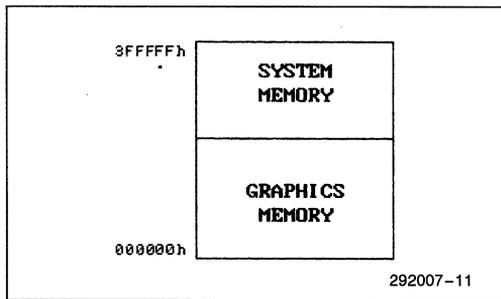


Figure 12. Memory Map for Graphics and Display Processors

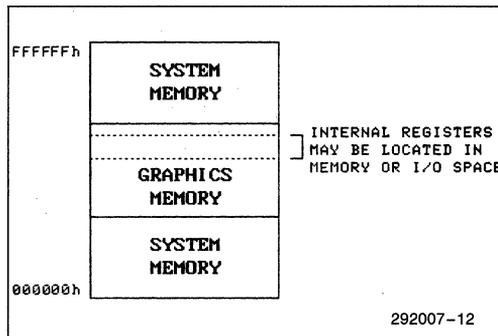


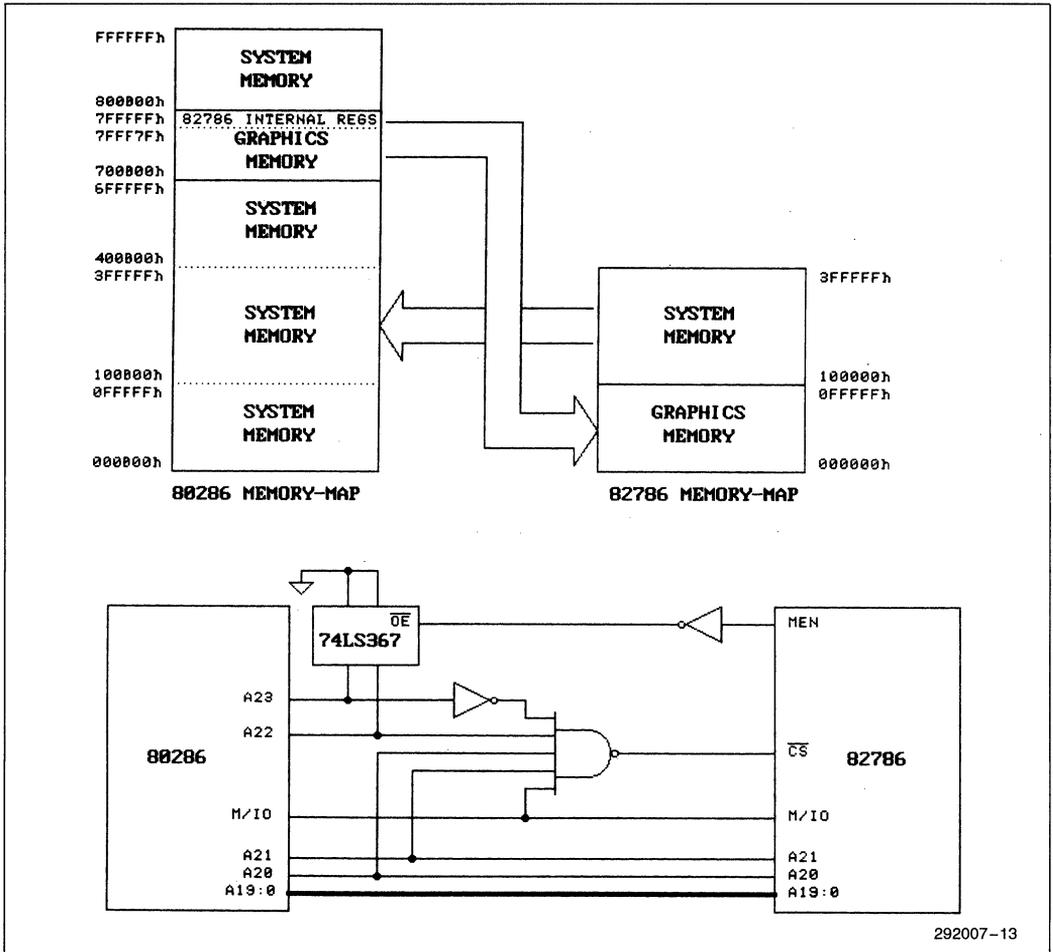
Figure 13. Memory Map for System CPU

The 82786 dedicated graphics DRAM/VRAM always starts at location 000000h and grows upwards. The upper address depends on the amount of DRAM/VRAM memory configured. The system bus memory begins where the DRAM/VRAM ends and continues to the highest addressable memory location 3FFFFFFh.

The memory map as it appears to the system CPU is shown in Figure 13. The area that the 82786 Graphics DRAM/VRAM is mapped into can be anywhere in the CPU address space and is completely defined by the address decode logic of the CPU system. Normally only the space for the configured graphics memory is mapped into CPU address space. If addresses above the configured graphics memory are mapped into the CPU address space, and the CPU writes to addresses above the configured 82786 memory, the write will be ignored. If it reads from these locations, the data returned is undefined.

The 82786 internal registers may be configured to reside in memory or I/O address space. If configured to reside in memory, then they will override a 128 byte area of the 82786 memory address space for external (CPU) accesses. The internal registers are only accessible by the external CPU and therefore are never found in the 82786 Graphics or Display Processor memory maps.

Suppose the 82786 is configured with 1 Megabyte of Graphics DRAM/VRAM and is used in an 80286 system. A possible memory map and connection diagram is shown in Figure 14. All of the 82786 memory is mapped into the 80286 address space. Also, a 3 Megabyte portion of the 80286 system memory is mapped into the 82786. Since the 80286 has two more address bits than the 82786, a tristate buffer is used to supply the top two address bits when 82786 enters master mode.

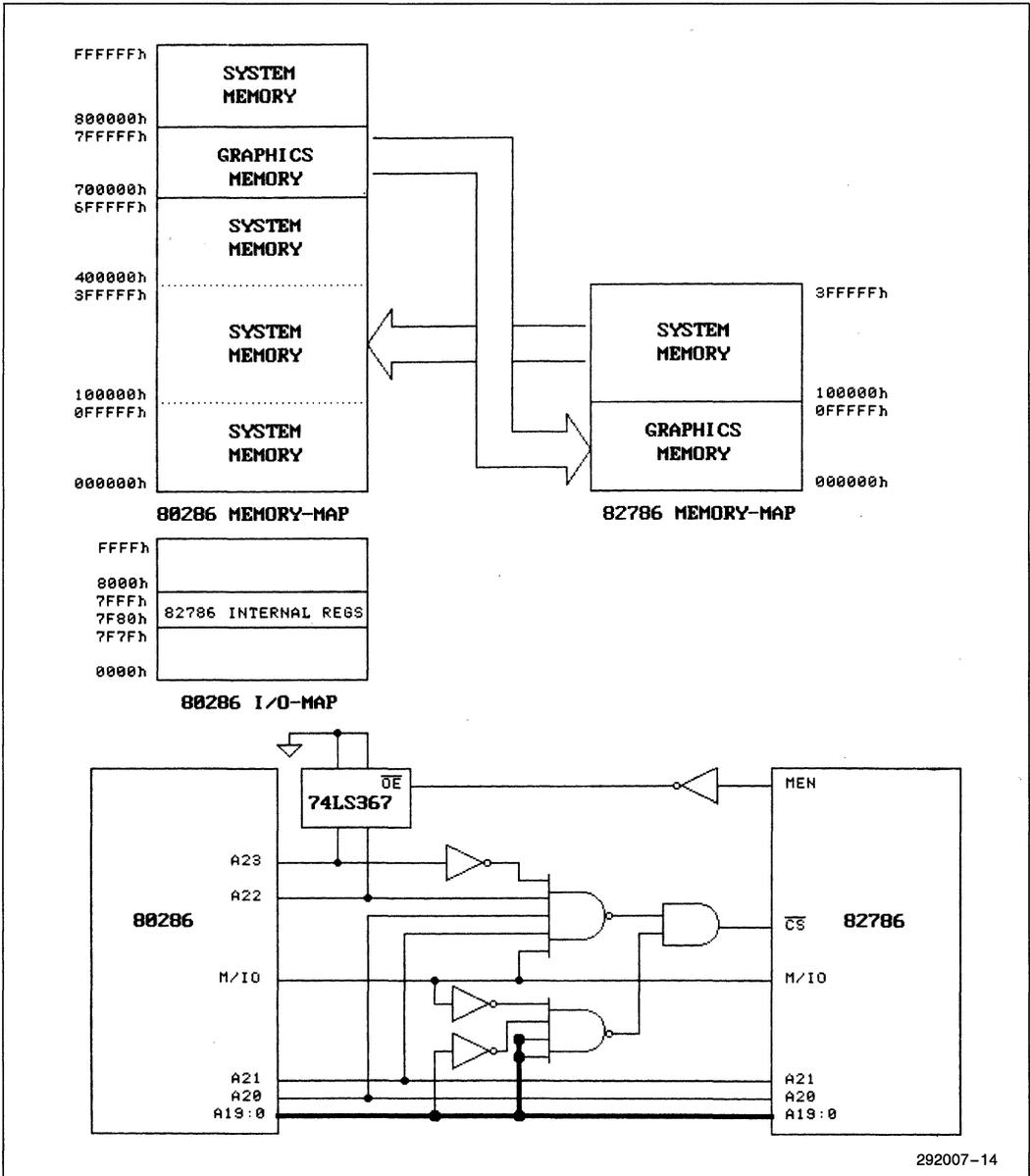


**Figure 14. Possible Memory Mapping for 80286/82786  
82786 Internal Registers are Memory Mapped**

Notice that the same memory corresponds to one set of memory addresses for the CPU and a different set of memory address for the 82786 Graphics and Display Processors. Although it is possible to make these addresses match, it is not necessary as long as the controlling CPU software understands the relationship and makes the simple conversion. Often it is not desirable to make the addresses match. For example, most CPUs use the lowest memory addresses for special purposes, such as for interrupt vectors. If the lowest CPU memory were 82786 memory rather than the faster (for CPU access) system memory, then these operations would execute significantly slower.

Even though the real addresses don't match, the operating system for a CPU such as the 80286 could make the CPUs virtual addresses map easily to the 82786 real addresses.

The 82786 internal registers may either be memory or I/O mapped. If they are memory mapped over the Graphics DRAM/VRAM, the CPU will not be able to access the 128-bytes of DRAM/VRAM which they cover, (although the Graphics and Display Processors can). If they are memory mapped above the Graphics DRAM/VRAM (over non-configured memory), then they will not prevent the CPU from accessing any of the 82786 memory, but they must be included in the CPU memory space that the address decoder allocates for the 82786. The 82786 internal registers may be I/O mapped, so they do not overlap any memory, however the CPU chip select logic for the 82786 becomes slightly larger. Figure 15 shows a circuit similar to Figure 14, except the registers are I/O mapped. Memory mapping the internal registers allows the software slightly more flexibility in accessing the registers.



292007-14

Figure 15. Possible Memory Mapping for 80286/82786  
82786 Internal Registers are I/O Mapped

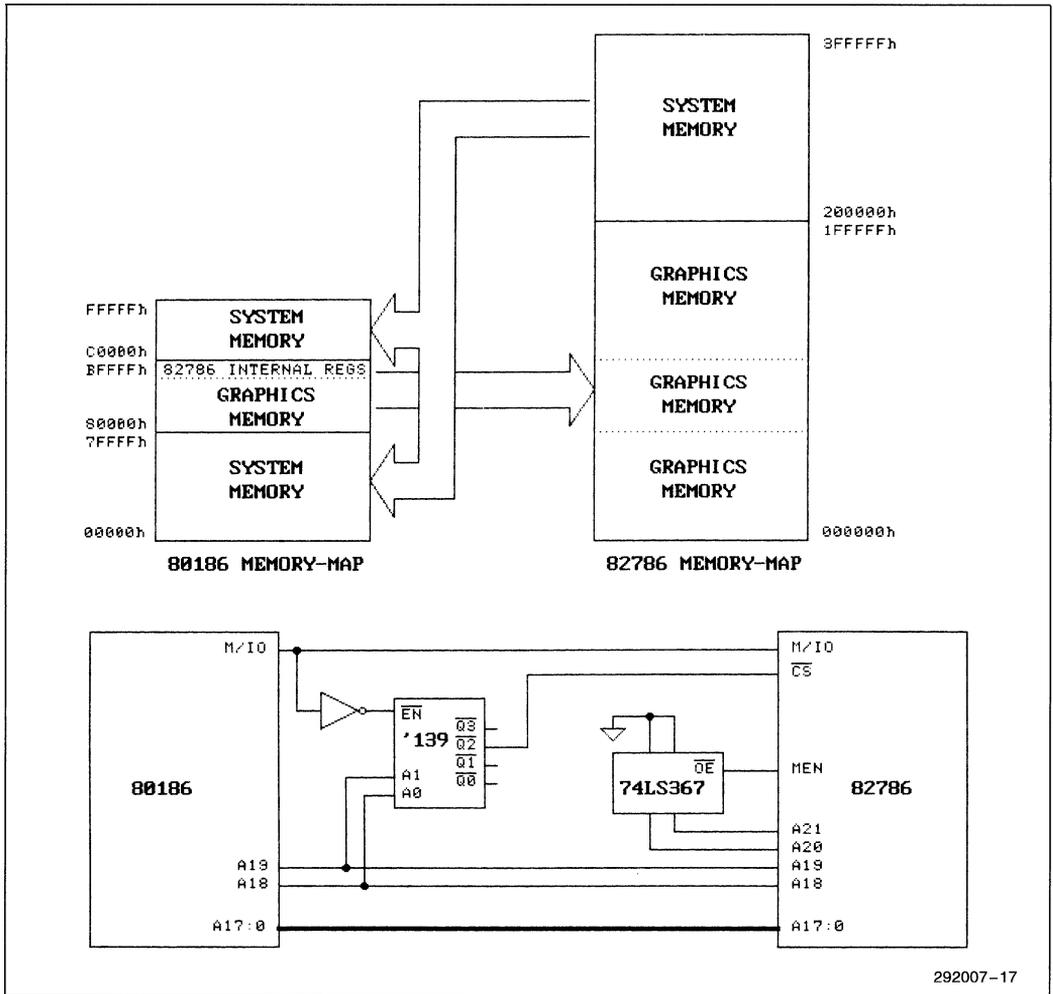


Figure 16. Possible Memory Mapping for 80186/82786

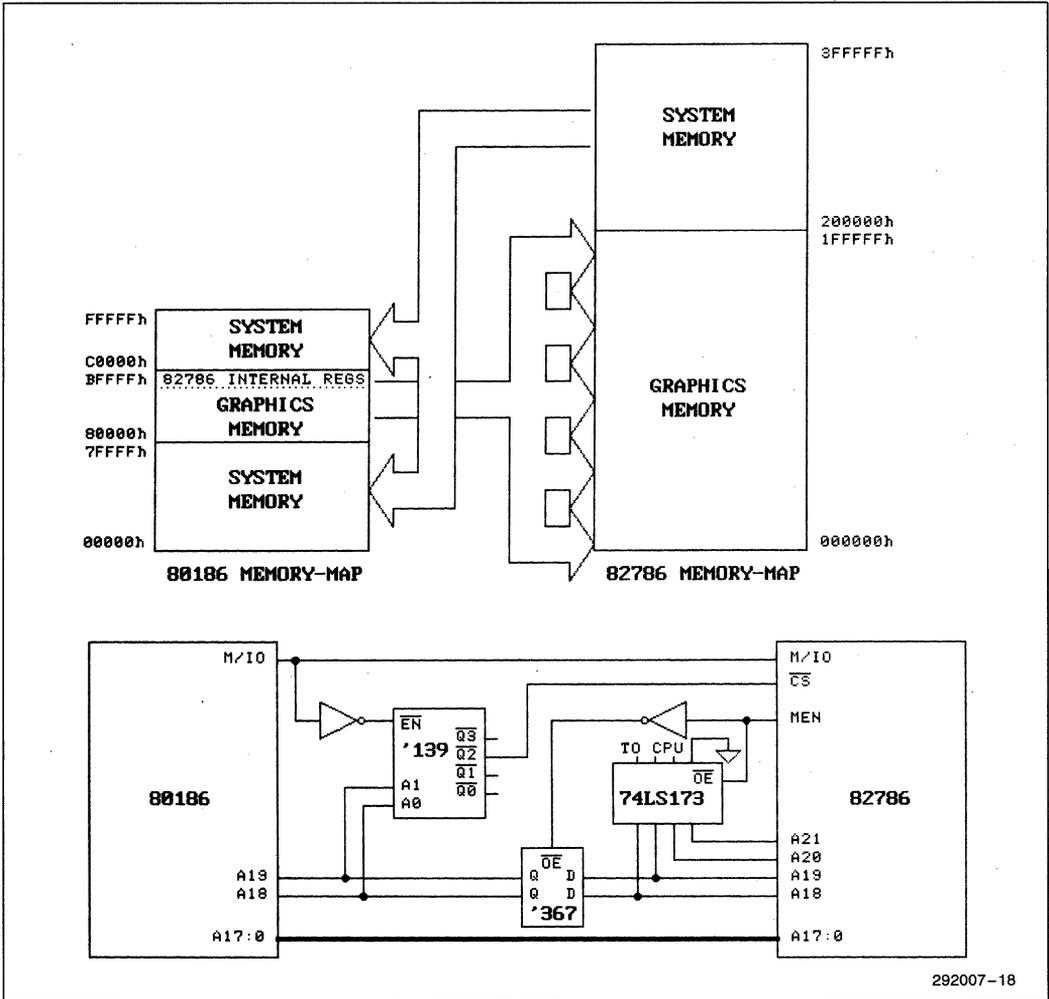
Because graphics memory can be quite large, some system designs might not allow all of the configured Graphics DRAM/VRAM to be directly mapped by the CPU. For example, if the 82786 has 2 Megabytes of Graphics DRAM/VRAM and is used with a 80186 processor, which can only address 1 Megabyte, then the 80186 can not directly access all of the 82786 memory. In this case the CPU can be permitted to only access a portion of the Graphics DRAM/VRAM. Figure 16 shows a memory map and connection diagram for such a system. Since the 82786 has two more address bits than the 80186, a tristate buffer is used to supply the two highest address bits when the 82786 is in slave mode.

In many cases the CPU does not require access to all of the graphics memory. For example, many situations

will not require the CPU to directly access the bit-maps. If the CPU must gain access to the Graphics memory which is not directly mapped to the CPU, the 82786 Graphics Processor can be instructed (using the BitBlt command) to move portions of the Graphics memory to and from the area accessible by the CPU.

Alternatively, the Graphics DRAM/VRAM areas can be bank switched to allow the CPU direct access at any portion of the graphics memory. Figure 17 shows the use of an I/O port (74LS173 latch) to which the CPU can write the highest 3 bits of the address for the 82786 slave accesses.

In both Figures 16 and 17, it is possible for the 82786 in master-mode to access the CPU memory addresses that



**Figure 17. Possible Memory Mapping for 80186/82786  
Bank-Switching Allows 80186 to Access All 82786 Memory**

correspond to the 82786 slave addresses. In this case, the circuit will generate a 82786 chip-select, but the 82786 will not respond to this chip-select while it remains in master-mode. As long as the READY logic goes high (it may not since the 82786 will not perform the slave-access) then the 82786 will complete the master-mode cycle. By the time the 82786 returns to slave-mode, the chip-select will have gone away.

## 4.2 BIU Registers

Within the 82786 internal register block, the registers at offsets 00h-0Fh are used by the Bus interface Unit to

control the system configuration (Figure 4). These registers are normally set once during power-up initialization and never changed.

Two of these registers, DRAM/VRAM Refresh Control and DRAM/VRAM Control have already been discussed in Section 3.3. The rest of the registers are discussed in this section.

The Internal Relocation Register is used to locate the 82786 internal registers anywhere in the 82786 memory or I/O address space.

Internal Relocation - Internal Register Offset 00h

15	1	0
Base Address		MIO

RESET Default: XX XXXX XXXX XXXX X 0

Base Address: determines bits 21:7 of internal register address (bits 6:0 of address are used as offset)

0 = I/O mapped  
1 = memory mapped

After RESET, any CPU slave I/O address to the 82786 (which activates the 82786 Chip-Select) will access the internal register block. During initialization, a write to the Internal Relocation Register should be performed to locate the register at the specific memory or I/O address desired. Once the write to the Internal Relocation Register occurs, the 82786 internal register block no longer occupies all of 82786 I/O space, rather it is restricted to just the 128 memory or I/O bytes specified. The internal registers can be located anywhere accessible by the CPU, however, if they are memory-mapped and located over configured graphics memory, they will take precedence over the memory for CPU accesses to those addresses. Graphics or Display Processor accesses to these addresses will still be directed to DRAM/VRAM. For example, writing the value of 03F8h locates the internal registers at I/O addresses FE00h – FE7Fh.

03F8h = 00 0000 1111 1110 0 0  
|  
I/O mapped  
Base Address 00FE00h(offsets 0–7Fh)

Note that the address written to the Internal Relocation Register determines the memory or I/O address that is required to be placed on the 82786 address pins during a CPU access to the 82786 internal registers. The actual CPU address used may be different, and is dependent on the chip select and memory mapping logic described in Section 4.1.

- There are four sources of requests for the 82786 bus:
- DRAM/VRAM refresh
  - Display Processor
  - Graphics Processor
  - External Processor (CPU or DMA slave accesses)

The DRAM/VRAM refresh requests are always top priority. That is, once the DRAM/VRAM refresh request is made, the 82786 bus will complete the current bus access and then perform the DRAM/VRAM refresh. Three BIU registers are used to set the priorities of the other three bus requests. Two priority values are used:

**FPL** - First Priority Level - priority used when processor first requests bus.

**SPL** - Subsequent Priority Level - priority used for processor to maintain bus during a block transfer. If a block transfer is interrupted, this is also the priority used for regain bus to complete the burst access.

When a processor first requests the 82786 bus, its FPL value is used. The processor with the highest priority gets access to the bus. Once the bus is granted, the first access occurs. If a multiple-word block transfer is performed the SPL value is then used as the priority to maintain the bus for subsequent cycles. As long as no other processor of higher priority requests the bus, the burst-access is allowed to continue to completion. If a higher priority request is made, the block transfer will be suspended and the bus granted to the new request. The suspended block transfer will not get the bus back until its SPL value is again the highest priority request.

A separate register is used to program the priority for each of the three processors. Because the External Processor can not perform block transfers, no External SPL value is required for it.

Display Priority - Internal Register Offset 0Ah

15	6	5	4	3	2	1	0
Reserved		F P L			S P L		
RESET Default:		1	1	0	0	1	1

Graphics Priority - Internal Register Offset 0Ch

15	6	5	4	3	2	1	0
Reserved		F P L			S P L		
RESET Default:		1	0	1	0	1	0

External Priority - Internal Register Offset 0Eh

15	6	5	4	3	2	1	0
Reserved		F P L			Reserved		
RESET Default:		1	1	1			

All of the priorities are programmable values from 0 to 7 with 7 being the highest priority. If two processors

that are programmed with the same priority both request the bus, the priority in which the bus will be granted for the two will be (from highest to lowest):

- Display Processor
- Graphics Processor
- External Processor

There are two exceptions to these programmable priorities. If the CPU makes a slave request while one of the 82786 processors makes a master request, the CPU's request will always be handled first by the 82786 regardless of priority settings. This is necessary to prevent the lock-out situation where the CPU will not grant HLDA until it completes the bus access to the 82786 and the 82786 will not complete the CPU bus cycle until the higher priority master cycle completes. Refresh cycles also always will be handled while the 82786 is in a HLDA loop.

The values programmed into these priority registers should be selected carefully. There is a performance penalty whenever a block transfer is interrupted. However, if block transfers are not interrupted, then it is possible that one processor must wait a long time to get the bus while another is finishing. A balance between overall bus performance and maximum tolerable latency must be made.

For example, if the Display Processor is not given high enough priority, it may not always be able to fetch the bit-mapped display data fast enough to keep up with the CRT. When this happens, the Display Processor will not be able to send the correct video data to the CRT and will instead place the value in the Default-VDATA register on the VDATA pins. To prevent this "snow" on the display, the Display Processor can be programmed for the highest priority (after DRAM/VRAM refresh).

The Display Processor internally contains a FIFO which is used to buffer the bit-map data to be displayed. The FIFO consists of 32-double-words of 32 bits each. Each FIFO double-word contains the results of a 32-bit fetch from the bit-map memory. A double-word can therefore contain as many as 32 pixels, or as few as 1 pixel (such as at window borders).

Display Processor Register 2 (TripPt) controls when this FIFO is loaded. If the trip point is set at 16, the Display Processor waits until the FIFO is half empty (only 16 double-words left) before it requests a new block transfer to refill the FIFO. The block transfer request will not end until the FIFO is again full (although the block transfer may be interrupted by a higher priority request). If the trip point is set at 28, the Display Processor will begin requesting a new block transfer after only 4 FIFO double-words are emptied (28 left remaining). A low trip point generates fewer but longer block transfers and therefore the overall Display Processor bus efficiency is increased. However, a low trip point also requires that the bus latency be smaller. A low trip point means that there are less double-words left in the FIFO when the bus request is

made. If the FIFO drains completely before the bus has been granted, then the DefaultVDATA will be used from the current pixel through the end of the current scan line. The trip point may be programmed to 16, 20, 24, or 28 using the Display Processor LD\_REG or LD\_ALL commands.

The Display Processor also keeps busy during blank times. During Vertical Blank time it performs any command loaded into its Opcode Register. During Horizontal Blank time it loads a new Strip Descriptor if necessary and begins fetching the first pixels on the line. (Actually, the descriptor fetch begins as soon as the last pixel of the last line has been placed in the FIFO). If the Display Processor priority is not high enough to allow these fetches during blank time, then again part of the display can not be generated correctly and Field Color will be used. Two bits in the Display Processor Status Register can be used to determine if the Display Processor ever gets behind:

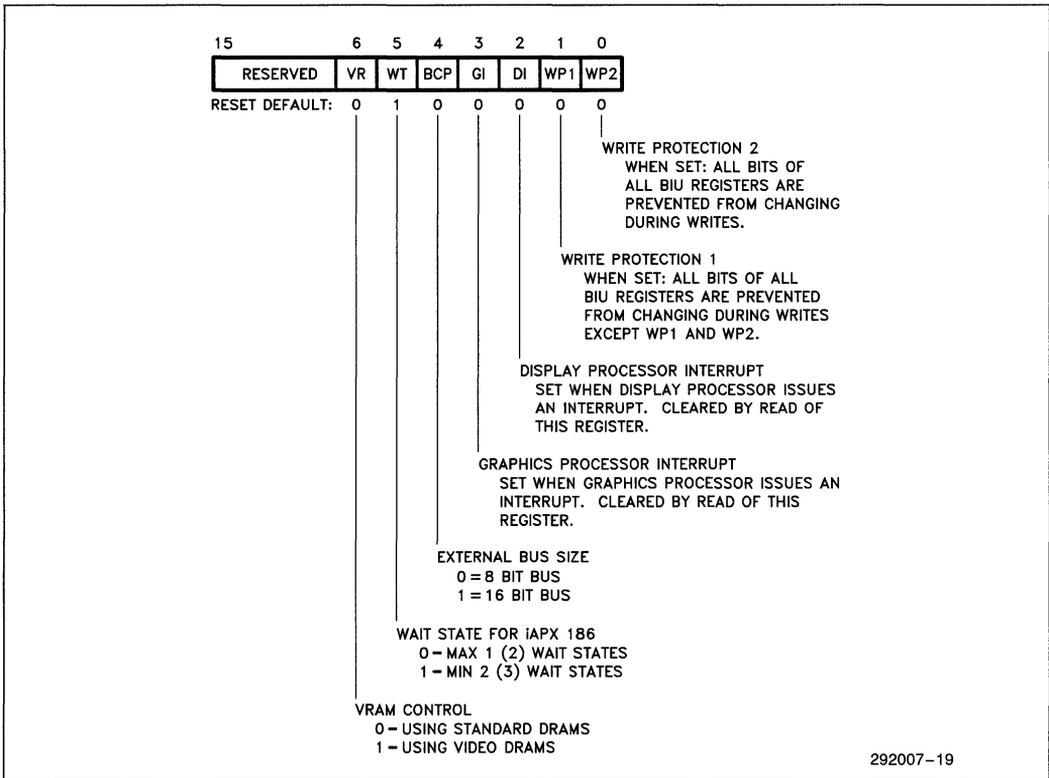
- bit-5 – DOV - Descriptor Overrun - set if strip descriptor fetch has ever not completed by the time horizontal blanking ends.
- bit-4 – FMT - FIFO Empty - set if the Display FIFO has ever completely drained.

Both bits are reset after reading the Status Register.

The setting of the External Priority register can greatly affect the performance of the external CPU when it performs an access to the 82786. Unless the External Priority is greater than the Graphics Processor, whenever the Graphics Processor is busy with a command stream that demands significant bus bandwidth, the CPU may have to wait a significant amount of time before it can complete an access to the 82786. The CPU waits for the 82786 in the middle of a bus access until the 82786 returns the READY signal. During this wait time, the CPU will not be able to process anything, including interrupts. Of course, if the application is very Graphics intensive and the CPU throughput is of lesser concern, then the Graphics Processor can be programmed with a higher priority.

Use the following priority values during your initial design. Once the system is working properly, you may wish to tweak the values for optimum performance. The optimum values are dependent on the CPU and video speeds as well as the CPU and graphics instruction mix and the window arrangement. In most cases, these registers will be initialized once and never changed. It may, however, be advantageous in some specialized applications to adjust these values when the application changes modes.

	FPL	SPL
Display Processor	6	6
Graphics Processor	2	2
External Processor	4	
Trip Point		24



One final BIU register contains a miscellany of bits.

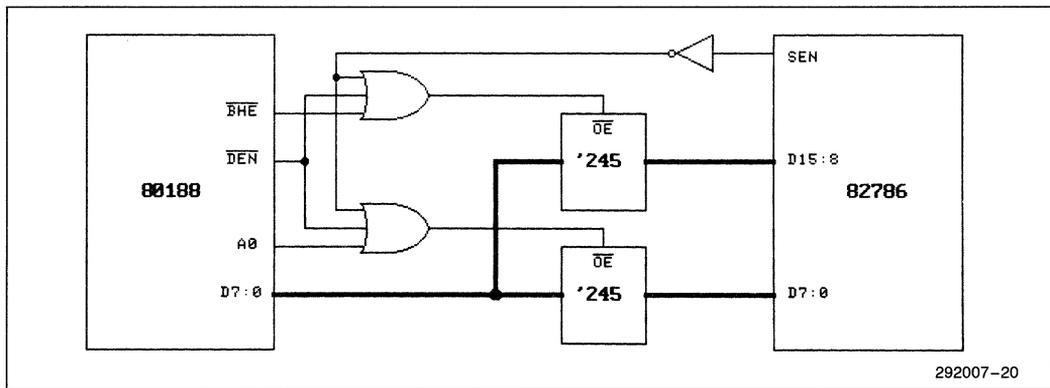
After the BIU registers have been initialized, the WP1 and WP2 bits can be used to protect all of the BIU registers (82786 internal register offsets 00h - 0Fh) from being rewritten. This will prevent faulty software from going wild and placing the 82786 into an unwanted state. Once WP1 is set, the only way to change the BIU registers is to reset WP1 first. Once WP2 is set, there is no way for the software to modify the BIU registers until a 82786 hardware RESET is performed.

After the 82786 causes an interrupt, the GI and DI interrupt bits are used to allow the software to determine whether the Graphics or Display Processor caused the interrupt. It is possible that both of these bits may be set if both processors have caused an interrupt by the time the interrupt handler reads this register. In this case, both interrupts should be handled by the interrupt handler.

Although it is not absolutely necessary to allow the 82786 to interrupt the CPU, it is very desirable. Graphics Processor interrupts can inform the software when it has completed all the commands as well as to report

error conditions. Display Processor interrupts can inform the software when a new display field has begun. A new command can then be loaded into the Display Processor to be executed before the next display field. This facilitates operations such as smooth scrolling and blinking. The only hardware requirement to permit 82786 interrupts is that the 82786 INTR pin is tied to one of the interrupt controller inputs.

Although the 82786 always uses 16 bits, the 82786 can be used with both 8 and 16 bit processors. For an 8-bit CPU, separate transceivers are required for the low and high bytes to the 82786 (Figure 18). In both 8 and 16 bit modes, graphics memory may be accessed a byte at a time. Although the 82786 internal registers may be read a byte at a time, they all are considered to be 16 bits (even if some of the bits aren't used) and must always be written in 2-byte even-word pairs. In 16-bit mode, they must be written as a 16-bit word. In 8-bit mode, first the lower (even-address) byte is written and then the upper (odd-address) byte is written. With an 8-bit processor such as the 8088, both of the following assembly routines may be used to load the 16-bit BIUControl Register with AX.



**Figure 18. 8-Bit CPU Uses Two Data Transceivers to Connect to 82786**

```

mov dx,BIUControl
out dx,al      ;write AL into low-byte of BIUControl
mov al,ah
inc dx
out dx,al      ;write AH into high-byte of BIUControl

or:

mov dx,BIUControl
out dx,ax      ;write AX into BIUControl word
    
```

In 8-bit mode, an even-byte write to a 82786 internal register does not change any of the 82786 internal registers, the data is simply saved until an odd-byte write to a 82786 internal register is performed. Then both the high and low bytes are written into that register. In effect, the even-byte address is ignored and an odd-byte write will write into the register both the odd-byte data and whatever even byte data was last written, into the register address specified by the odd-byte access. There is no limit to the amount of time allowed between the even-byte and corresponding odd-byte writes. An odd-byte write that is not preceded by an even-byte will be ignored.

The 82786 always comes up in 8-bit mode after RESET. This means that a 16-bit CPU should change the BCP bit to one. It must perform two byte-wide accesses to do this. The following initialization code can be used.

```

mov dx,BIUControl
mov al,30h
out dx,al      ;write 30h into low-byte of BIUControl

xor al,al
inc dx
out dx,al      ;write 00h into high-byte of BIUControl

mov dx,InternalRelocation
mov ax,03F8h
out dx,ax      ;write 03F8h into InternalRelocation word
    
```

The 82786 is first placed in 16-bit mode (using two 8-bit writes), then the 82786 internal registers are located at the desired address (which is done with a 16-bit write). Next, the DRAM/VRAM and priority registers should be initialized. Byte-wide writes into the 82786 internal registers can not be performed while BCP = 1.

All the 82786 master mode operations are 16 bits wide independent of the BCP bit. This means that system memory must be accessible 16-bits at a time if master mode is to be used. The WT bit is set to 1 on reset. The VR bit is reset to 0 at reset.

### 4.3 80286 Synchronous Interface

Since the 82786 has been optimized for the 80286, it is not surprising that the interface logic is very minimal. Figure 19 shows a 82786 connected synchronously to an 80286. Much of the logic, such as the 82288, chip-select, and ready, can be shared by the rest of the 80286 system.

This configuration allows both master and slave accesses. The data transceivers allow the 80286 to access the 82786 and graphics memory and the 82786 to access the 80286 system memory. They also provide the isolation required to allow the 80286 to access system memory while the 82786 accesses graphics memory simultaneously. The tristate buffer 74LS367 is used to pull the 80286 upper address lines,  $\overline{\text{COD}}/\overline{\text{INTA}}$ ,  $\overline{\text{LOCK}}$  and  $\overline{\text{PEACK}}$  to their proper states during master-mode. If any of these signals are not used by the rest of the system, they need not be driven by a tristate buffer.

If master mode is not required,  $\overline{\text{MEN}}$  will stay low and three of the four gates driving the data transceivers can be eliminated. Also, the tristate buffer, which is only used in master-mode, may be eliminated.  $\overline{\text{HREQ}}$  should be left open and the 82786  $\overline{\text{HLDA}}$  pin should be tied to ground so that the 82786 will never enter master mode.

Both the 80286 and the 82786 internally divide-by-two the CLK input and use both phases. For the 82786 to run correctly with the 80286, these phases must be correlated correctly. This can easily be done by observing the setup and hold times for rising RESET for both

chips (see 80286 data sheet specifications 6 and 7 and 82786 data sheet specifications C6 and C7). The 82284 chip will meet this requirement.

Depending on the CLK speed and the type of DRAM/VRAM used, the 82786 may have very stringent CLK duty cycle requirements (see Section 3.2). It may not be possible to use the internal oscillator of the 82284 chip but it may be possible to use an external oscillator to drive the 82284 external clock (EFI) pin.

Clock skew between the 80286 and the 82786 should be kept to a minimum so the chips should be placed as close together as possible.

When the 82786 bus is free, the circuit in Figure 19 permits CPU slave accesses using 2 wait-states for writes and 3 wait-states for reads. Using DRAMs/VRAM with slightly faster access times, the circuit in Figure 20 permits both read and write slave accesses using 2 wait-states. The 82284 SRDY input is used instead of ARDY. The 82786 SEN timing is such that a minimum of 2-wait states are always generated for writes but a minimum of 2 or 3 wait-states are used for reads depending on the use of SRDY or ARDY. Notice that with 2 wait-state reads, the SEN signal must be qualified with  $\overline{\text{CS}}$  so that SEN does not extend into the cycle following the slave write. The most critical relationship to be satisfied in order for 2 wait-state writes is:

$$T_{\text{cac}} < T_{\text{c}} + T_{\text{ch}} - 15 - 45$$

For a 10 MHz 82786 the DRAM/VRAM column access time must be:

$$T_{\text{cac}} < 50 + 25 - 45 = 30 \text{ ns}$$

Note that x 1 DRAMs have two transceiver delays.

The critical timing calculations for slave mode are calculated as follows. The actual numbers calculated are for an 80286/82786 system running at 10 MHz.

chip-select-logic	=	path from 80286 address to 82786 $\overline{CS}$ pin				
	<	$2 \times$ clock period	—	address valid	—	setup
	<	$2 \times 286.T1$	—	286.T13	—	82786.Ts1
	<	$2 \times 50$ ns	—	60 ns	—	5 ns
	<	35 ns				
ready-logic	=	path from 82786 SEN to 82284 SRDY pin				
	<	clock period	—	SEN active	—	ARDY setup
(if ARDY is used as in Figure 19)	<	286.T1	—	82786.S18	—	82284.T13
	<	50 ns	—	25 ns	—	0 ns
	<	25 ns				
ready-logic	=	path from 82786 SEN to 82284 ARDY pin				
	<	clock period	—	SEN active	—	SRDY setup
(if SRDY is used as in Figure 20)	<	286.T1	—	82786.S18	—	82284.T11
	<	50 ns	—	25 ns	—	15 ns
	<	10 ns				
				from SEN active to read data valid		
read data valid $\geq$ 82786.Ts22 + transceiver delay				from SEN active to write data valid		
write data valid $\geq$ 82786.Ts20						

The master mode signals generated by the 82786 are all within the specification range guaranteed by the 80286. In other words, if the system memory is designed to function with the 80286, it will also be able to function with the 82786. The only signals that may not be within the range of the 80286 specifications are the data bus signals due to the transceiver delays. Care must be taken to ensure that the memory subsystems that the 82786 is to be able to access in master mode can meet these more stringent requirements:

		data valid to falling clock after Tc phase 2		
read data setup	>	82786 read data setup	+	transceiver-delay
	>	82786.T8	+	data in to data out
	>	5 ns	+	Tprop
		data valid delay from falling clock after Ts phase 1		
write data valid	<	82786 write data valid	—	transceiver delay
	<	82786.T14	—	data in to data out
	<	40 ns	—	Tprop

The clock skew between the 80286 and the 82786 must be considered in all these calculations.

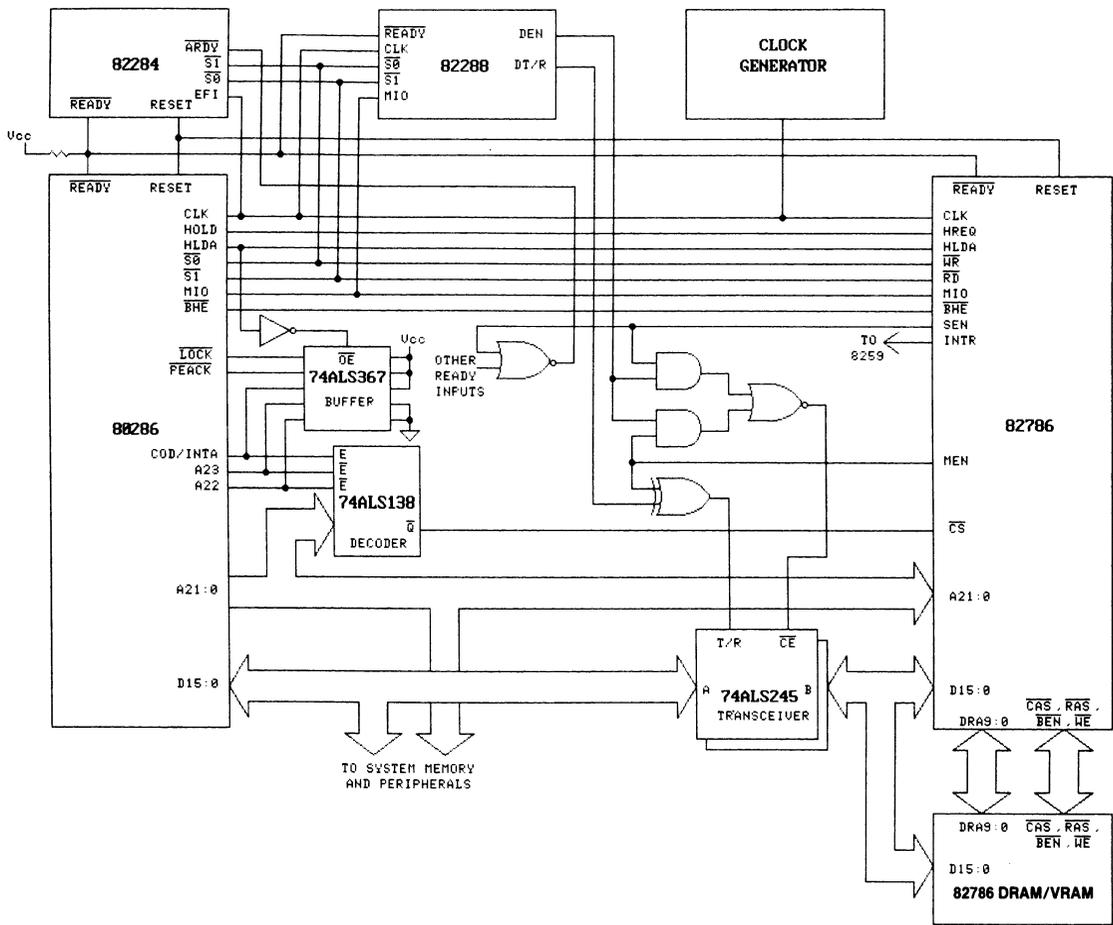


Figure 19. 286/82786 Synchronous Master/Slave Interface Permits Minimum of 2 Wait-State Write, 3 Wait-State Read

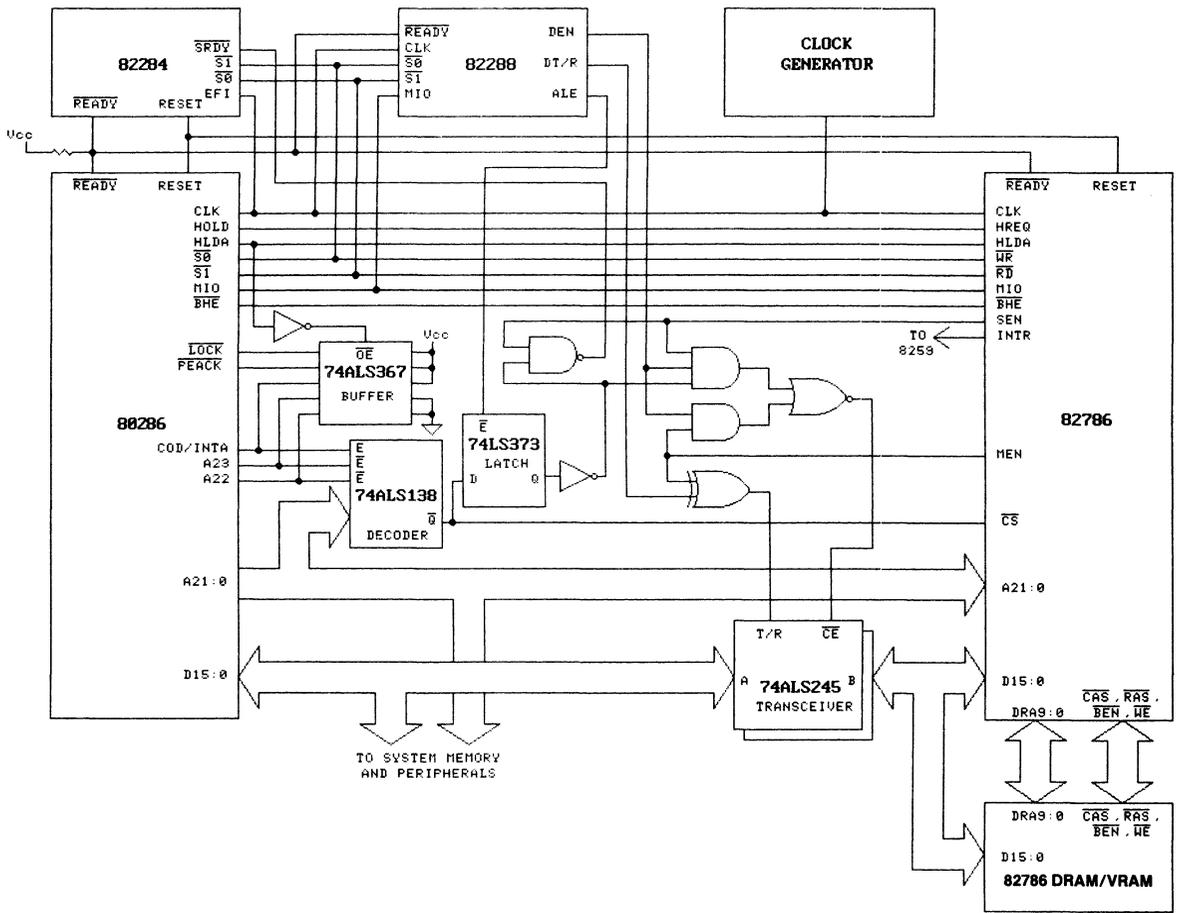


Figure 20. 286/82786 Synchronous Master/Slave Interface  
Fast DRAMs Permit Minimum of 2 Wait-State Read/Write

## 4.4 80186 Synchronous Interface

The 82786 supports a synchronous status interface to the 80186. The 82786 and the 80186 must be driven with the same external clock (EFI). The Reset inputs to the 82786 must be generated from the  $\overline{\text{RES}}$  for the 80186 by delaying it by one clock (input). This guarantees that the 82786 Clock phase 1 is coincident with 80186 CLKOUT low. A synchronous 80186 interface is selected if BHE is high and MIO is high prior to falling 82786 RESET.

Generally this configuration will be used with a minimum of 3 wait states for the 82786 slave read and write accesses. Therefore the WT bit in the 82786 BIU Control Register should be set. The 82786 slave accesses will then only be initiated when the 82786  $\overline{\text{CS}}$  is actually activated.

There is, however, a way to allow this interface to use a minimum of 2 wait states. (Set WT=0) Rather than wait for  $\overline{\text{CS}}$  to go active the 82786 can be allowed to request a slave access as soon as the 80186 status lines go active. If the 82786 is not in the midst of another bus cycle and the CPU request is the highest priority, the bus will immediately be granted to the CPU and a bus cycle started. If the  $\overline{\text{CS}}$  then goes active the 82786 can complete the access within 2 wait-states. If  $\overline{\text{CS}}$  does not go active (because the 80186 is not accessing the 82786 but rather its own memory or I/O) then the 82786 will abort the bus cycle by running a dummy 82786 bus cycle.

If there is other RAM or ROM in the system besides the 82786 Graphics DRAM/VRAM that the 80186 often accesses, then this 2 wait-state will probably hinder rather than help performance. Every time the 80186 fetches from its own system memory (such as an opcode fetch or operand access), and the 82786 bus is idle, the 82786 will waste time running a dummy cycle. Fortunately, the busier the 82786 bus is, the less likely it will be free when the 80186 initiates a bus cycle, and therefore the less likely the 82786 will waste time running a dummy cycle.

## 4.5 Asynchronous Interface

An asynchronous interface can be used to interface the 82786 with nearly any CPU. The CPU clock and the 82786 clock are independent and may run at different speeds. If the 80286 is connected asynchronously with the 82786 and both processors are run at approximately the same clock frequency, then the minimum possible wait-states is one more than for the corresponding synchronous mode.

Figure 21 shows a slave-only 10 MHz 82786 interface to an 8 MHz 80186. At 10 MHz, the 82786 requires that the address becomes valid S17 = 80 ns after  $\overline{\text{RD}}$  or  $\overline{\text{WR}}$  falls and remains valid for S16 = 130 ns. Because the 80186 address disappears the same cycle  $\overline{\text{RD}}$  and  $\overline{\text{WR}}$  fall, the address must be latched. This latched address can be shared by the other components on the 80186 bus.

Due to the indeterminate phase relationship between the CPU and 82786 clocks, care must be taken to ensure the read/write data timings have enough slack. When the read data is sampled, and when the write data is removed is determined by the CPU's ARDY input. The 82786 SEN signal is used to generate the ready signal which is responsible to ensure that the data is indeed available. D-flip-flops can be used to delay the SEN signal to delay the CPU ready signal. For a 10 MHz 82786:

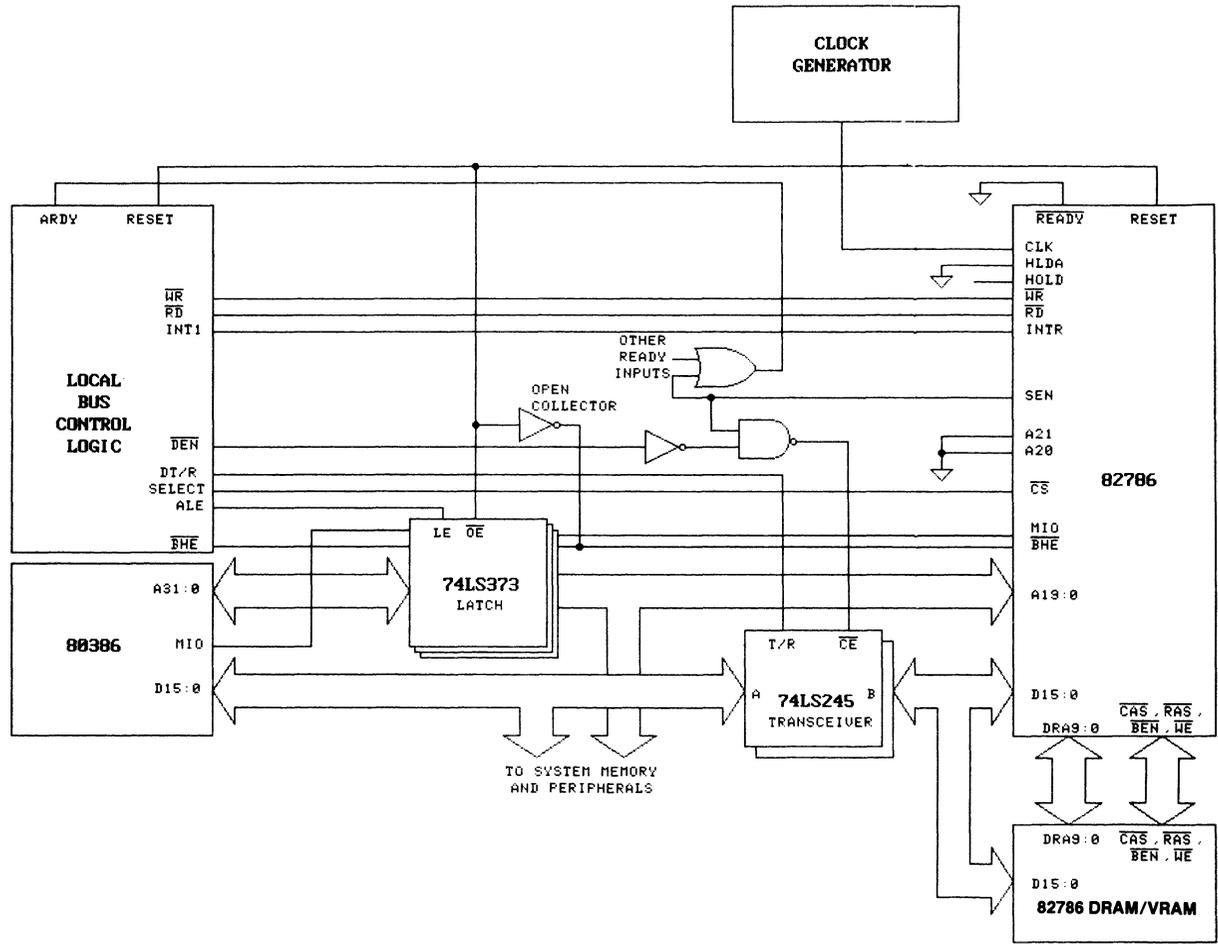
	from SEN active to read data valid
read data valid $\geq$	82786.Ts22 + Tprop
	from SEN active to write data valid
write data valid $\geq$	82786.Ts20

To initially place the 82786 into the asynchronous interface mode, the 82786  $\overline{\text{BHE}}$  pin must be low during the falling edge of RESET. To ensure this, the 74LS373 latch for  $\overline{\text{BHE}}$  is tristated and an open-collector inverter pulls down  $\overline{\text{BHE}}$  during RESET.

The 80386 processor can be interfaced to the 82786 either synchronously or asynchronously. For a synchronous interface, standard logic can be used around the 80386 to emulate a 80286 style bus for use with the interface described in Section 4.3. In this configuration the 82786 bus would run at half the clock rate of the 80386 (a 16 MHz 80386 would run with an 8 MHz 82786 bus). For an asynchronous interface, the standard local bus controller logic used by the 80386 to interface most peripherals can be used (Figure 22).

Although the actual bus transfers of a synchronous bus are faster than for an asynchronous bus, there are cases where an asynchronous interface provides the highest performance. For example, for a given display resolution, the Display Processor overhead of a 10 MHz 82786 is a lower percentage of the total bus throughput than for an 8 MHz 82786. If the 82786 is used with a 16 MHz 80386, then an asynchronous 10 MHz 82786 would have more bandwidth for the CPU and Graphics Processor than a synchronous 8 MHz 82786 and therefore CPU accesses, generally, will be completed faster with the asynchronous interface.





292007-B5

Figure 22. 80386/82786 Asynchronous Slave-Only Interface

## 4.6 Multiple 82786 Interface

For higher performance, it is possible to use several 82786 chips in the same system. Any of the above CPU/82786 interfaces can be used to attach multiple 82786s to one CPU in the system. Each 82786 will require its own separate DRAM/VRAM array.

The driving software for these multiple CPUs would most likely be sending nearly the same commands to all of the 82786s. Rather than forcing the software to write commands to each 82786 individually, it is possible to allow write commands to go to several or all the 82786s. One method of determining which 82786s should receive the write command would be to first write to an I/O port in which each bit corresponded to a different 82786. In Figure 23, the port bits set to 0 enable the corresponding 82786 for CPU writes. When a write to 82786 address-space occurs, all of the selected 82786s are chip-selected. The CPU will then wait for READY from all the selected 82786s before completing the bus cycle. In this manner, one, all, or any combination of 82786s can be written into at once.

Because it is impossible to read from several 82786s at once, a priority scheme is used on the I/O port to allow a read from only one of the selected 82786s. The circuit in Figure 23 only allows slave-accesses, the 82786s may not enter master-mode.

If master-mode operation of the multiple 82786s is desired, each 82786 must access the bus separately. A priority scheme is used to determine which 82786 is awarded the bus when the CPU issues HLDA. With only two possible 82786 masters, the random circuitry to hold one 82786 off the bus while the other is using it is straight-forward (Figure 24). With more 82786 masters, it is more feasible to use a state-machine (possibly implemented in PALs) to perform the arbiting.

## 5.0 VIDEO INTERFACE

The video interface connects the 82786 to the video display. The 82786 is optimized to drive CRT monitors but may also be used to drive other types of displays. Because CRTs provide an inexpensive method of generating moderate and high resolution, monochrome and color displays, this application note will concentrate on CRT interfaces. Section 5.10 briefly describes other display interfaces.

The video interface for a CRT is very dependent on the CRT requirements and the resolution and depth (bits/pixel) of the image desired. The 82786 can be programmed to directly generate all the CRT signals for up to 8 bits/pixel (256 color) displays at video rates up to 25 MHz. In addition, external hardware can be add-

ed to allow a color look-up table or to trade-off the number of bits/pixel for higher display resolutions, or to use VRAMs.

Some of the possible display configurations are shown below. The calculations assume a 60 Hz refresh rate. High resolution CRTs are often run at a slower rate, which permits the 82786 to generate significantly higher resolutions than those in the following table. All cases assume a CRT horizontal retrace time of 7  $\mu$ s, except the 512  $\times$  512  $\times$  8 (10  $\mu$ s) and 640  $\times$  400  $\times$  8 (13  $\mu$ s) cases.

### With Standard DRAMs

	Non-Interlaced	Interlaced
8 Bits/Pixel (256 colors)	512 $\times$ 512 640 $\times$ 400 640 $\times$ 480	900 $\times$ 675
4 Bits/Pixel (16 colors)	870 $\times$ 650	1290 $\times$ 968
2 Bits/Pixel (4 colors)	1144 $\times$ 860	1740 $\times$ 1302
1 Bit/Pixel (monochrome)	11472 $\times$ 1104	2288 $\times$ 1716

Multiple 82786s can be used together to generate even higher resolutions with more colors. For example, two 82786s allow a non-interlaced 1144  $\times$  860 sixteen color display.

### With Video DRAMs\*

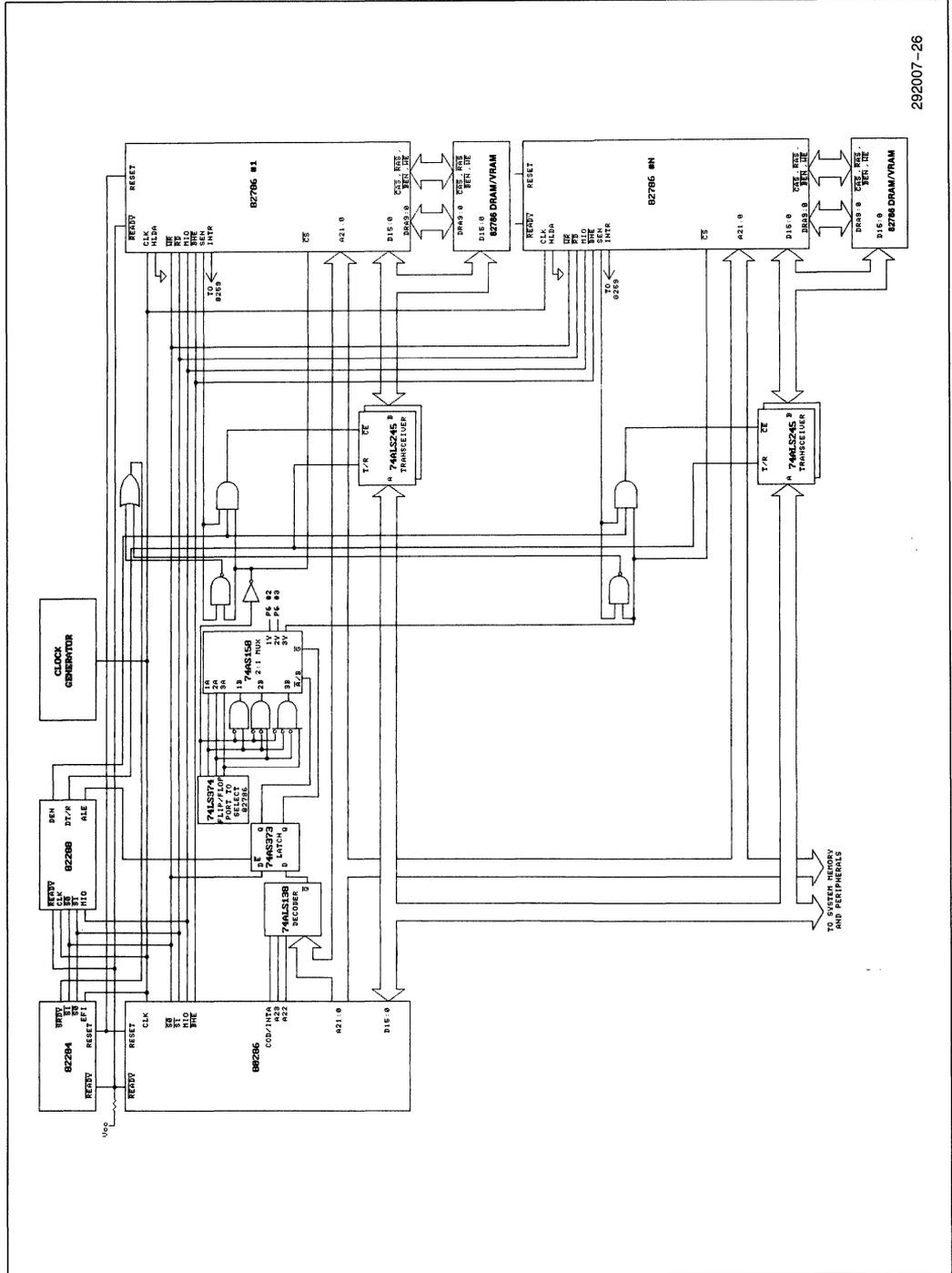
	Non-Interlaced
8 Bits/Pixel (256 colors)	1024 $\times$ 1024
4 Bits/Pixel (16 colors)	2048 $\times$ 1024
2 Bits/Pixel (4 colors)	2048 $\times$ 2048
1 Bit/Pixel (monochrome)	4096 $\times$ 2048

\*For 64K by 4 - with 256K by 4 higher resolutions are supported

## 5.1 Various CRT Interfaces

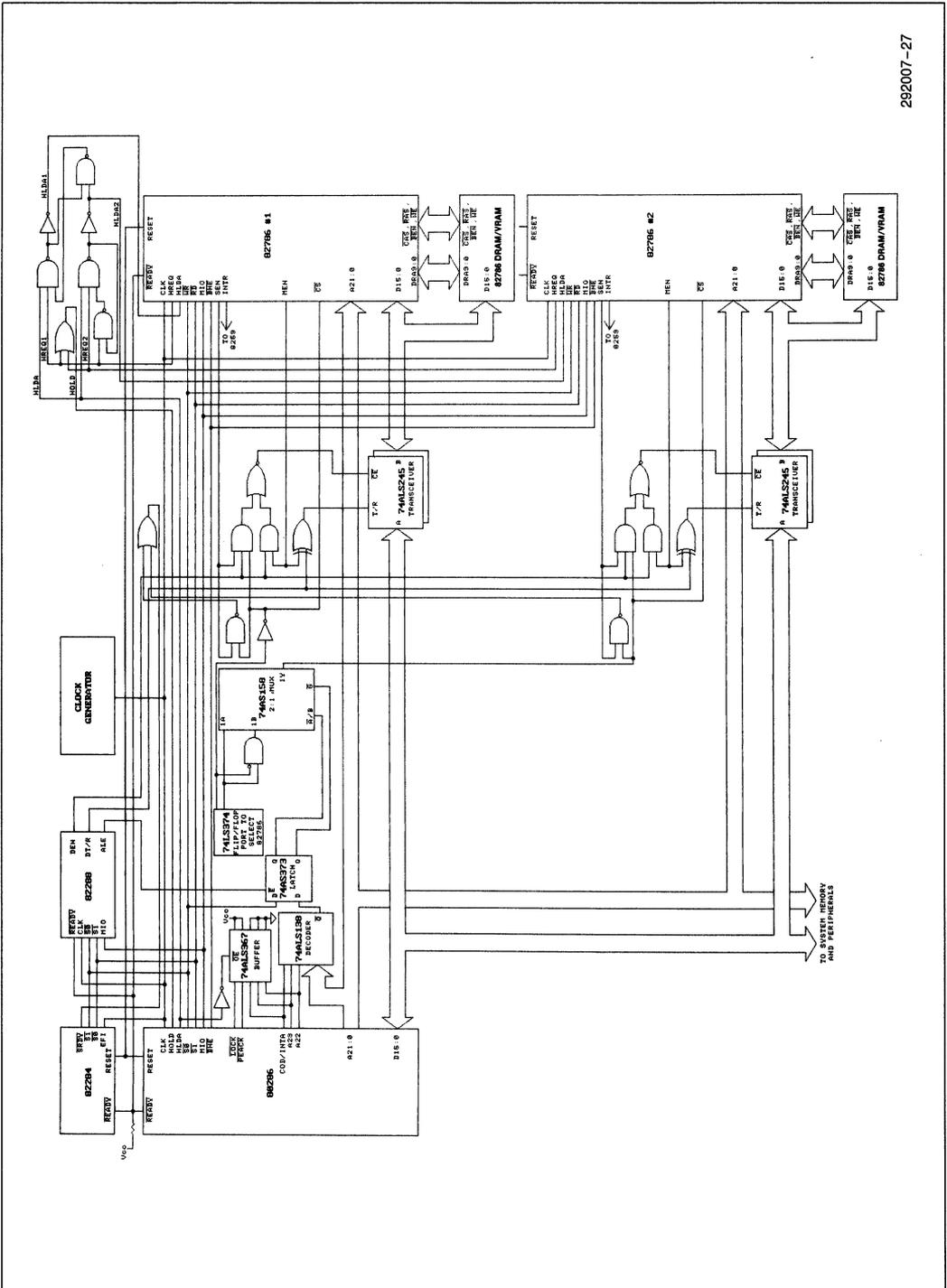
CRT monitors use a wide variety of interfaces. Some use TTL-levels on all inputs, others require analog inputs. Some use separate color inputs (red, green and blue) and separate horizontal and vertical sync while others require that some or all of these signals be combined into composite signals. This application note will concentrate on the generation of separate color and horizontal and vertical sync signals. Standard techniques can be used to convert these separate signals into composite signals to meet the requirements of other displays.

The video clock (VCLK) required by the 82786 may be generated by a simple oscillator with TTL-outputs. Alternatively, the VCLK can be tied to the bus clock (CLK) (or any other available clock) if they are to run at the same speed.



292007-26

Figure 23. This Configuration Allows Several 82786s to be Written by 80286 Simultaneously—Only Slave Accesses are Supported



292007-27

Figure 24. Two 82786s Connected to 80286, Permits Slave and Master Accesses

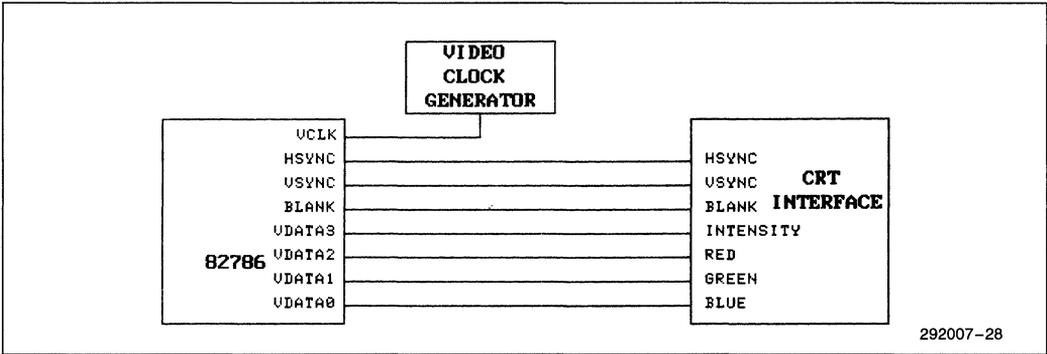


Figure 25. 82786 Can Directly Drive TTL-Input CRT Interface

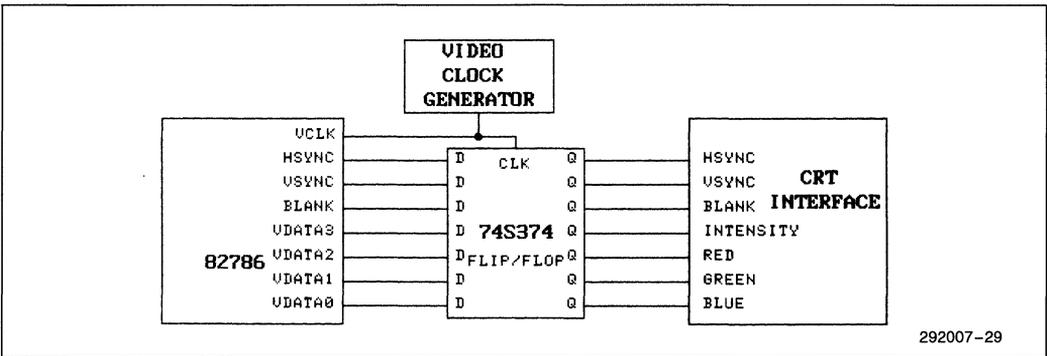


Figure 26. Buffer Used to Drive TTL-Input CRT Interface

## 5.2 CRTs with TTL-level Inputs

The simplest interface is to CRTs that use TTL-level inputs. The 82786 can generate these signals directly (Figure 25). However, the drive requirements of the CRT and cabling may make it necessary to buffer the signals (Figure 26). The example monitor in both of these cases happens to use a CRT that uses four-bits of color information per pixel. This means that 16 different colors are available and the CRT can use the 82786 1, 2, and 4 bits/pixel modes but can not take advantage of the 8 bit/pixel (256 color) mode. A monochrome monitor with only one TTL-level input could be connected directly to VDATA0 and use the 82786 1 bit/pixel mode but it then can not take advantage of any of the higher bit/pixel modes.

## 5.3 CRTs with Analog Inputs

Taking advantage of the 8 bit/pixel mode of the 82786 usually requires using a CRT with analog inputs. Signals for color CRTs with three separate analog video inputs, (red, green, and blue) can be generated using three digital-to-analog converters (Figure 27). Often these digital-to-analog converters can be constructed using simple resistor ladders (Figure 28). With 8 bits/pixel, usually three bits are used to select red, three for green and two for blue. This is because our eyes are much more sensitive to variations of red and green than of blue. These configurations can take advantage of all the 82786 modes; 1, 2, 4, and 8 bits/pixel.

The VDATA pins may be assigned to the three colors in any manner desired. In Figure 29 they are assigned so that a variety of colors are available for each mode (1, 2, 4, and 8 bits/pixel).

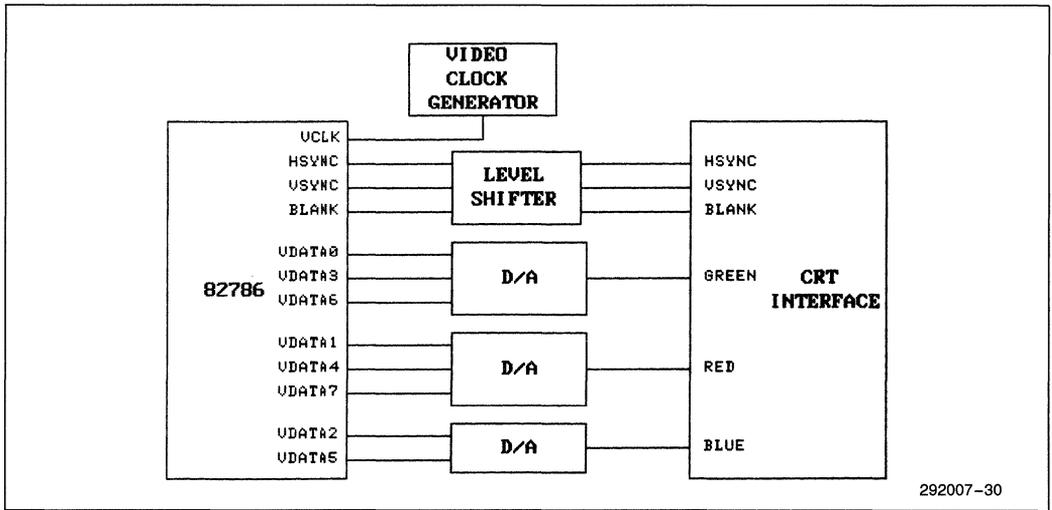


Figure 27. Analog CRT Interface Allows 256 Colors

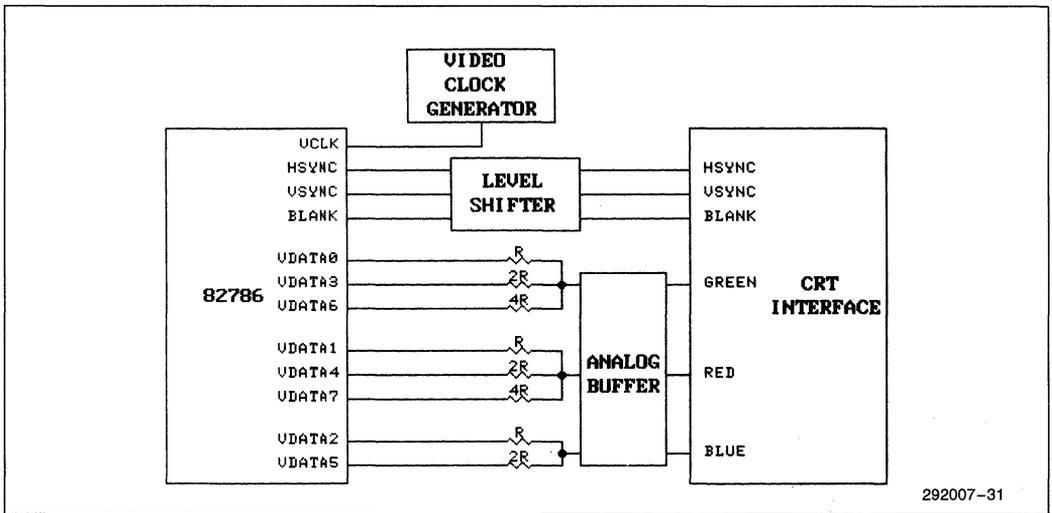


Figure 28. Resistor-Ladder Used for D/A

VDATA7	VDATA6	VDATA5	VDATA4	VDATA3	VDATA2	VDATA1	VDATA0
Red bit 0	Green bit 0	Blue bit 0	Red bit 1	Green bit 1	Blue bit 1	Red bit 2	Green bit 2

Figure 29. VDATA Pin Assignments

- The most-significant Green bit is connected to VDATA0 so that in the one bit/pixel mode this bit is controlled while the other bits are set to a constant level by the padding register internal to the Display Processor. If, for example, the padding bits are all set to zero, then a green and black image is shown in one bit/pixel windows.
- With two bits/pixel the most significant Green and Red bits are controlled while the rest are padded to a constant value. If, for example, the padding bits are set to zero then the colors black, green, red, and yellow are available in two bits/pixel windows.
- Four bit/pixel windows contain two Green bits and the most significant Red and Blue bits making 16 colors available.
- Eight bit/pixel windows allow control of all eight bits to make all 256 colors available.

### 5.4 Using a Color Look-Up Table

Color Look-up Tables, also known as Video Palette RAMs, allow more colors to be available with a minimal of actual bits/pixel and thus a minimal amount of display memory is required for the bit-map. For example, in a system using 16 bits of color information, 65536 different colors are possible. In such a system it is rarely necessary to display all 65536 colors on the screen simultaneously. It may be feasible to support a maximum of 256 colors simultaneously, providing that these 256 selections can be any combination of the 65536 available colors. Color look-up tables permit such a cost-effective system.

A block diagram of such a system is shown in Figure 30 and Figures 30a and 30b show actual circuits. The color look-up table can be loaded with up to 256 16-bit

colors. In this way an 8-bit/pixel bit-map can be used to control the 16-bit colors.

The host CPU is responsible for loading the 16-bit colors into the look-up table. To load a color into a specific location in the look-up table, the 82786 Display Processor can be programmed to output the 8-bit address on the 8 VDATA pins during the horizontal and vertical blank times or on RESET by setting the DefaultVDATA register. Then the CPU can load the color value into the 16-bit latch.

The circuitry in Figure 30 will then automatically write the 16-bit value into the look-up table during the next horizontal sync time. The CPU should generate the 74AS373 latch enable input so that the latch can be mapped into memory or I/O space and loaded by a CPU write. The register between the 82786 and the palette RAM is used to allow the use of a RAM with a slower access time. This register is not necessary if a faster RAM is used.

The CPU program should wait until the color is loaded into the look-up table before loading the next color. One way to ensure this is to route the LookupLoading signal through a port which the CPU may poll. Sample assembly language code for this configuration follows this section. Another way is for the CPU program to delay a sufficient amount of time to ensure that HSYNC has occurred before writing the next value.

Hybrid circuits can be used which combine the functions of the look-up table, analog-to-digital conversions, and voltage shift for composite sync signals into one package. Figure 30b shows such a configuration. This particular hybrid circuit internally contains a 16 × 12-bit look-up table, 4 bits for each red, green, and blue.

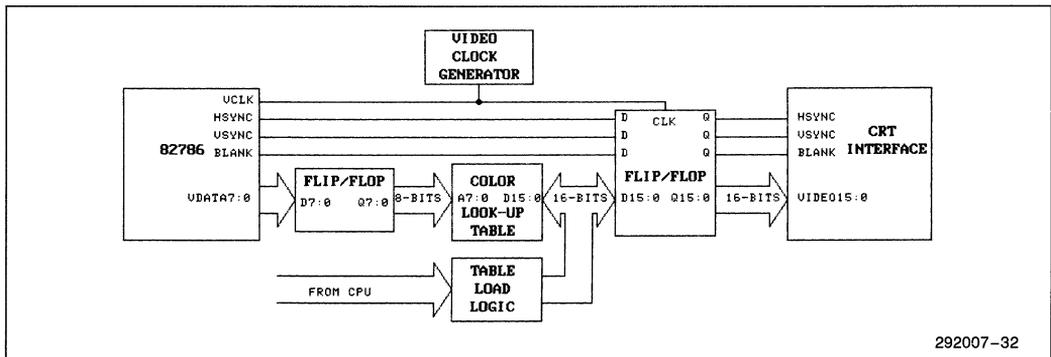


Figure 30. Block Diagram of Color Look-Up Table Used to Generate 16 Video Bits From 8

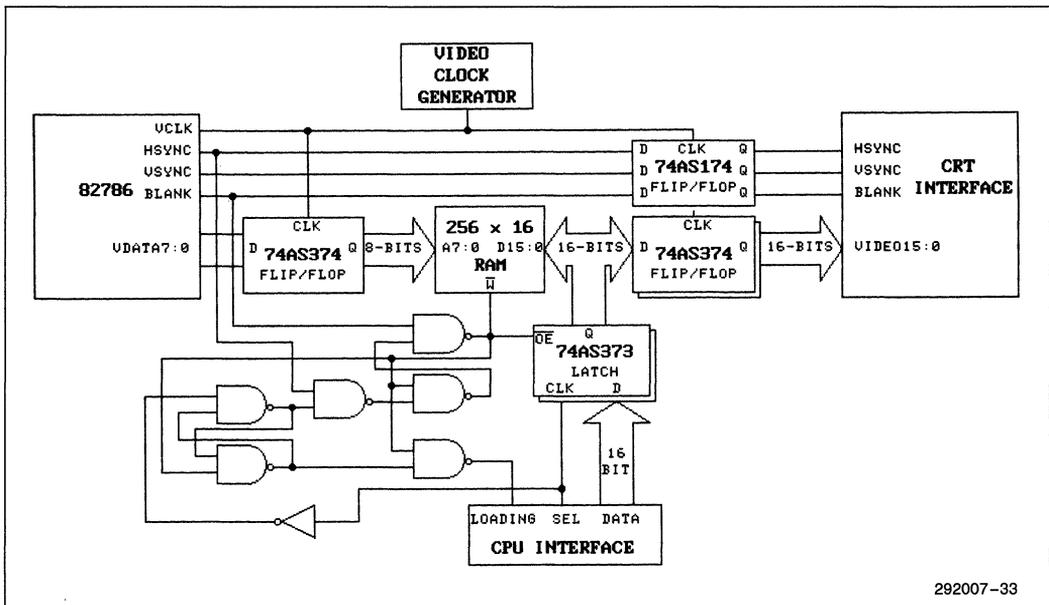


Figure 30a. Circuit for Color Look-Up Table Used to Generate 16 Video Bits From 8

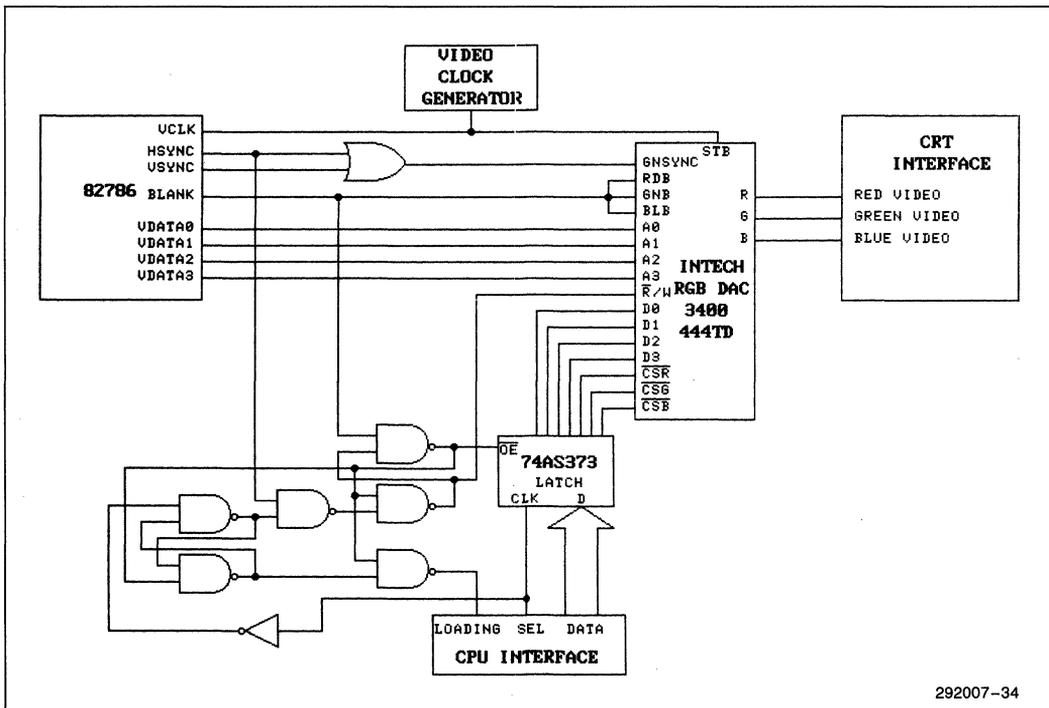


Figure 30b. Hybrid Color Look-Up Table and DAC Simplifies Interface

```

Wait:  in    al,StatusPort      ;read port
       test  al,LookupLoadingBit ;test LookupLoading bit
       jnz   Wait              ;wait til last load completed
       mov   ax,EightBitAddr    ;get 8-bit value to load
       mov   DefaultVDATA,ax    ;make 82786 output during BLANK
       mov   ax,SixteenBitColor ;get 16-bit color
       out   LookupLatch,ax     ;write color into latch

```

The look-up table is loaded by first writing the location into the 82786 DefaultVDATA register. Then a 4-bit color value is loaded into the latch along with color-select information. Therefore, in one load it is possible to place this 4-bit color value into any combination of the red, green, and/or blue tables.

## 5.5 Using the Window Status Signals

A graphics system design may require that the video data bits for different windows be interpreted in different ways. For example, the attributes controlled by various video data bits may need to be changed between windows for different tasks or number of bits/pixel. For these reasons, two Window Status bits are available externally which reflect a value which may be individually programmed for each window. These two pins always reflect the window which the display is currently scanning. The software is responsible for placing the two bit values for each window in the Tile Descriptor list.

In addition, the cursor can be programmed with a value for the window status bits which can be programmed to override the status bits from the windows for the portion of the display where the cursor resides.

The Window Status bits are multiplexed onto the HSYNC and VSYNC pins. Since they are only applicable during the visible display time, and since HSYNC and VSYNC are only applicable during the non-visible display time, BLANK can be used to de-multiplex these pins (Figure 31).

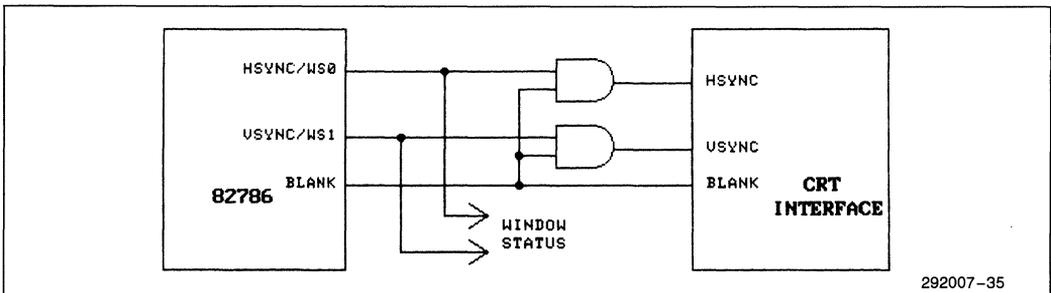
A mode bit (bit 4 of CRTMode) in the Display Processor enables the Window Status bits so they become multiplexed onto the HSYNC and VSYNC signals. This bit must be set when the Window Status signals

are used. In systems where the Window Status bits are not needed, this bit can be reset so that the HSYNC and VSYNC pins remain low during the visible display. This allows simpler systems to use HSYNC and VSYNC directly eliminating the need to AND these pins with BLANK.

As an example, suppose the interpretation of the video data bits by a color look-up table was to be different for different windows. Possibly four different look-up tables are required for four different types of 8 bit/pixel windows. A large look-up table (1024 words) could be divided into four areas, one for each of the window interpretations. Then the Window Status bits could be used to select the area of the look-up table to be used for each specific window. Essentially four look-up tables would be available, one for each of four different types of windows. Figure 32 illustrates such a system.

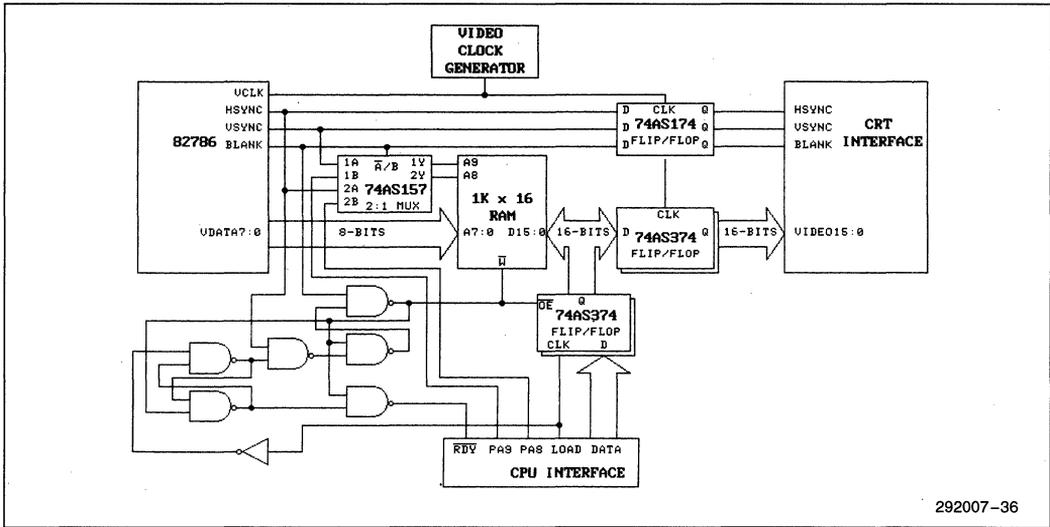
The system also requires circuitry to load the look-up table such as that in the previous section. Note that the look-up table's Window Status inputs must be generated directly from the CPU when the RAM is to be loaded since they can not be programmed in specific states during the blank time as the VDATA pins can.

Another use of the Window Status bits is to allow 1, 2, 4, and 8 bit/pixel windows to each use a different look-up table along with a fifth look-up table for the cursor. A 1024 word look-up table above could be split up into four areas as above. Two of the areas can be used for two separate 8 bit/pixel look-up table and the other two shared by the 1, 2, and 4 bit/pixel windows for two separate look-up table for each of 1, 2, and 4 bits/pixel (Figure 33). The padding bits can be used to sub-divide this second area into separate tables for 1, 2 and 4 bit/pixel windows. Finally, this same table could also be used for twelve look-up tables, four each for 1, 2, and 4 bit/pixel windows.



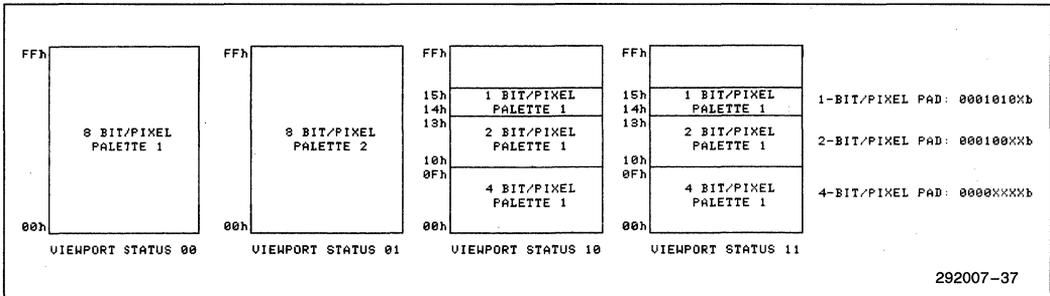
292007-35

Figure 31. Using Blank to De-Multiplex Window Status



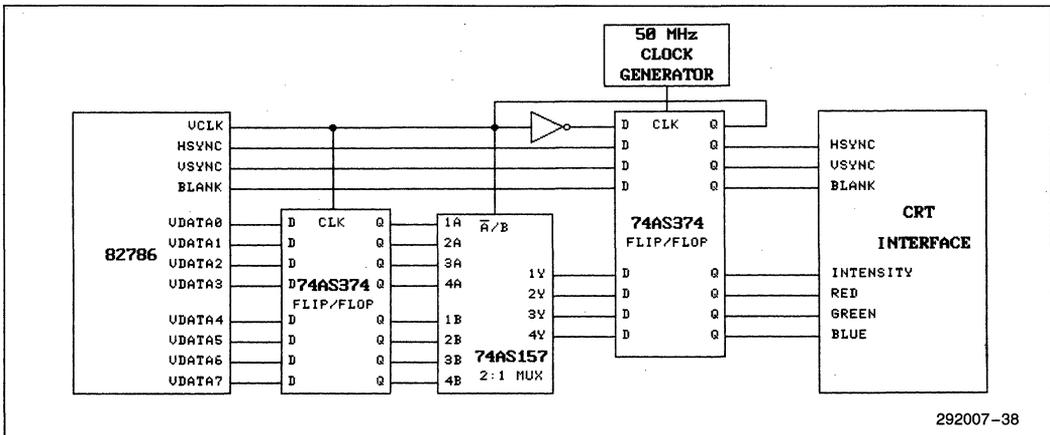
292007-36

Figure 32. Four Color Look-Up Tables—Selectable by Window Status Outputs



292007-37

Figure 33. Window Status and Padding Bits Allow Two Separate Look-Up Tables for Each of 1, 2, 4, and 8 Bit/Pixels



292007-38

Figure 34. External Multiplexer Allows Up to 50 MHz Video with 4 Bits/Pixel

## 5.6 Higher Resolutions with Standard DRAMs

The Video Clock rate on the 82786 can be a maximum of 25 MHz. For a non-interlaced display refreshed 60 times per second this limits the resolution to  $512 \times 512$  or  $640 \times 400$  or equivalent displays.  $640 \times 480$  can also be achieved using a CRT with fast horizontal retrace. Still, some graphics system designs may require more detailed displays and therefore more resolution. It may very well be cost-effective to trade-off the number of bits/pixel for higher resolution. This is especially true in the case of monochrome displays where 256 grey-shades are not required but high resolution is.

The 82786 allows this trade-off to be made very effectively. Figure 34 shows how a video data rate of up to 50 MHz may be obtained with 4 bits/pixel (16 colors). The 82786 is used to output 8 bits of video data at a 25 MHz rate. The external multiplexer switches between the low 4 bits and the high 4 bits at a rate of 50 MHz. The register before the multiplexer is used to ensure that enough set-up time is provided for the multiplexer. The register after the multiplexer ensures that the video data out has smooth transitions. The circuit uses an inverter and one register stage to divide the 50 MHz clock by 2 to create the 25 MHz video clock for the 82786. Instead of the 74S157 multiplexer, a 74AS298 chip could be used which contains the multiplexer and the register in the same package.

The software has a minimum number of changes. The Graphics Processor is programmed identically and manipulates the bit-maps in the conventional manner (although it does not make sense to use 8 bits/pixel bit-maps since they cannot be displayed). The display processor programming is slightly different. The Accelerated Video control bits (CRTMode bits 1,0) are set for High Speed video (01). The HSynStp, HFldStrt, HfldStp, and LineLen registers are programmed for half the number of dot clocks (because the 82786 VCLK is half the speed of the pixel dot clock).

The Strip and Tile Descriptors list also change only slightly. Windows are programmed for same number of bits/pixel and FetchCount as they would be for non-accelerated modes. However, windows may only be positioned horizontally to start on even pixel boundaries. That is, they may only start at every-other pixel, not at any pixel as permitted with non-accelerated modes. This is because both an even and odd pixels are output on the VData pins simultaneously and it is not possible to mix windows during a single VCLK. The only valid values for the start/stop bits are listed in the following table. Notice that the accelerated modes do not permit all possible bit-map depths because fewer than 8 bits/pixel are available to the display.

Vertically, the windows may still be positioned at any pixel. The programming of the one pixel horizontal and vertical borders also does not change.

High-Speed video mode also requires that the Field windows are programmed with half the number of actual pixels for the pixel count (BPP/Start/Stopbit) register which again restricts horizontal positioning to a two pixel resolution.

The horizontal cursor position is programmed as half the actual value so the positioning is also restricted to a two pixel resolution. Vertically, the cursor is programmed as normal. Since the cursor is only a 1 bit/pixel region, every other horizontal pixel reflects only the cursor padding value so although simple cursor patterns are possible, arbitrary shapes are not possible with the box cursor. For this reason, the programmer may wish to create the cursor in software when using these high-resolution modes rather than use the 82786 hardware cursor. The cross-hair style cursor works well in accelerated mode, although the horizontal and vertical lines become two pixels wide and horizontal positioning is also limited to two pixels.

Bit-Map Depth	Acceleration			
	None (25 MHz)	High-Speed (50 MHz)	Very-High-Speed (100 MHz)	Super-High-Speed (200 MHz)
1 bit/pixel:	any (0–15)	even numbers	0,4,8,12	0,8
2 bit/pixel:	even numbers	0,4,8,12	0,8	—
4 bit/pixel:	0,4,8,12	0,8	—	—
8 bit/pixel:	0,8	—	—	—

It is also possible to use external hardware to create the cursor. One method is to program the cursor as invisible (transparent and all background) and use the cursor's window status signals to activate the external hardware.

The horizontal zoom capability is also affected. Rather than replicating each individual pixel, pairs of pixels are replicated. Vertical zoom works as normal.

Figure 35 shows a configuration for video data rates of up to 100 MHz with 2 bits/pixel. A shift-register is used to multiplex the 8 video bits from the 82786 into 2-bit streams. A 74AS74 flip/flop is used to divide the 100 MHz clock by four. Every fourth clock the 82786 VCLK is raised and the shift registers are loaded with the previous 82786 VDATA. The video data is delayed two cycles by this circuit while the synch and blank are delayed only one. This should not be a problem if the 82786 is programmed to generate the correct sync. The 82786 is limited to positioning the sync transitions at multiples of four pixels. If more accurate positioning is required, extra flip/flops can be used to delay sync for more cycles.

The timing in Figure 35 is very tight and the circuit may not operate at 100 MHz over all operating temperatures. The limiting speed path is the 74F195 shift-reg-

ister parallel-load time (delay from clock to outputs valid) which must meet the set-up time of the 74AS374.

Figure 36 shows a configuration for video data rates of up to 200 MHz with 1 bit/pixel. Unfortunately, there is no TTL-logic available today which will run at the speeds required for 200 MHz. Therefore ECL or some other high-speed logic must be used to generate video at these high rates. Figure 36 converts the video data signals from the 82786 from TTL to ECL levels and then uses ECL shift-registers to generate the 200 MHz signal.

The software for the configurations in Figures 35 and 36 requires changes similar to the Figure 34 case. The window start/stop bits are programmed restricted as shown in the table above. The pixel count for Field regions is also one-fourth or one-eighth the actual size. Horizontal positioning is also restricted to four and eight pixels for the 100 MHz and 200 MHz rates respectively. The Accelerated Video control bits must also be programmed for these configurations.

After the video signals are accelerated to these higher speeds, color look-up tables and analog-to-digital converters may be used. The circuits in the previous sections must be adapted for these higher speeds.

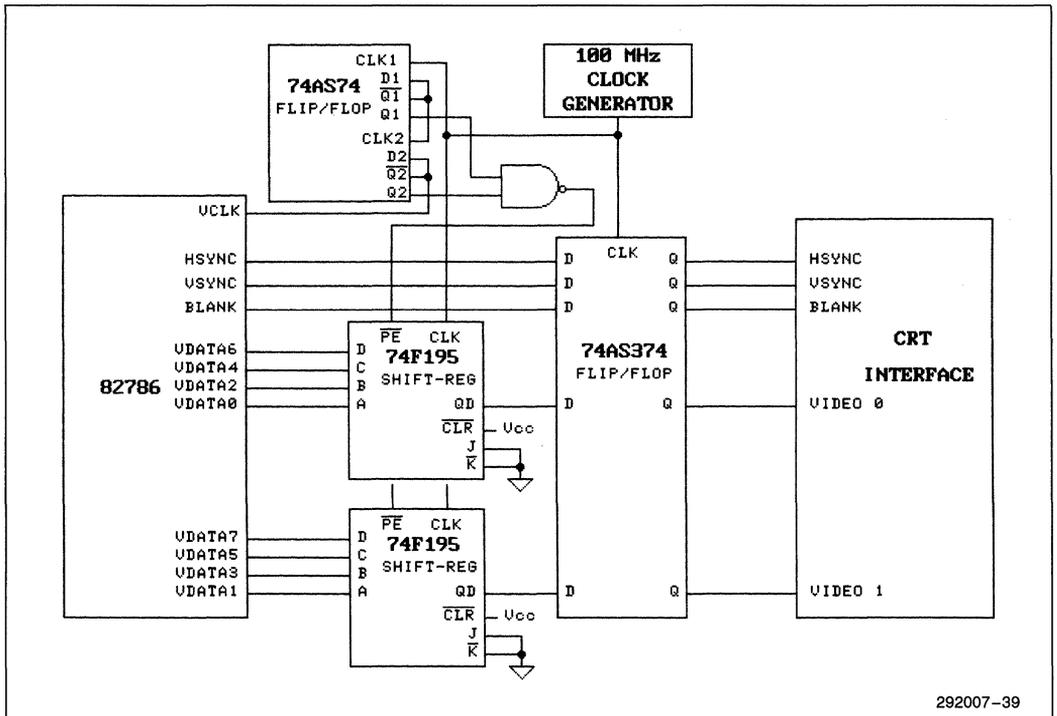


Figure 35. External Shift-Register Allows Up to 100 MHz Video with 2 Bits/Pixel



## 5.7 Multiple 82786s

If more colors or resolution are required than possible with one 82786 at a given resolution, several 82786s can be used together to generate the necessary bits/pixel. Figure 37 shows two 82786s used together to generate 16 bits/pixel at a 25 MHz video rate. This configuration would allow a  $640 \times 480$  display with 65536 colors.

Both 82786s' video must be kept in sync. To allow this, one 82786 is programmed as normal to generate the master video horizontal and vertical sync. The second 82786 is programmed for slave video sync with the Slave CRT control bit in the CRTMode Register (Display Processor register 5-bit 3 set). The HSYNC and VSYNC lines of the slave 82786 then become inputs and are driven by the HSYNC and VSYNC output lines of the master 82786. If the window status signals are used, the master's HSYNC and VSYNC signals should be qualified with the BLANK signal (similar to Figure 31) to correctly drive the slave 82786 HSYNC and VSYNC inputs. Window status signals are only available from the master 82786 since the slave uses these pins as inputs.

Both 82786s should have six of their eight video timing registers (HSynStp, HFldStrt, HFldStp, LineLen, VSynStp, VFldStp, VFldStrt, FramLen) programmed identically; HFldStrt and HFLdStp should be programmed to be 2 greater than the master. (These parameters are calculated in Section 5.11.)

The slave 82786 will then automatically sync itself up to the master 82786 by waiting for its HSYNC input to fall before each scan line and waiting for its VSYNC input to fall before beginning a new display field. If a non-interlaced display is used, the two 82786s will always be in sync.

If an interlaced display is used, care must be taken to ensure both 82786s start on the same field. The easiest way to ensure they lock in sync correctly is to ensure they start scanning the display simultaneously. First set up the slave 82786 CRTMode and video timing registers with a LD\_ALL command. The slave 82786 will then be ready to begin scanning the display but will wait until HSYNC and VSYNC fall. HSYNC and VSYNC will be floating high because they are tristated by all the 82786s. Then the master 82786 can be set up with a LD\_ALL command to program its CRTMode and video timing registers. Once the master starts scanning, the HSYNC and VSYNC signals will be driven by the master and all 82786s will begin on the even interlaced field.

To create a 16 bit/pixel bit-map, both 82786 Graphics Processors should be programmed for 8-bit/pixel bit-maps of identical size. To draw in both bit-maps, a graphics command block (GCMB) can be created for both 82786s. These GCMBs are generally identical for both 82786s except for the color values for the Def\_Color and the mask value for the Def\_Logical\_Op commands. To display 16 bit/pixel bit-maps, both 82786s should be given an identical strip descriptor list for each to display 8 bits of each 16 bit pixel.

Similarly, 8-bit/pixel bit-maps could be created by splitting the bit-maps between both 82786s having each 82786 responsible for 4 of the 8 bits/pixel. This would split the work between the two 82786s so that the BitBlt and Scan\_Line fill graphic commands will execute twice as fast. Also, because the Display Processor bus overhead is split between the 82786s, there will be less bus contention so all other drawing commands will be faster.

Alternatively, 8-bit/pixel bit-maps could be generated by only one of the 82786s. This would minimize the overhead between the host CPU and the 82786 since the CPU must communicate with only one 82786.

The method in which the 16 video data bits are mapped into colors for the display interface will determine which of the two above methods will be used for bit-maps of 8 bits/pixel or less. If the mapping is flexible enough, it may be feasible to create any bit-map depth. For example, 9 bits/pixel bit-maps could potentially be created using one 82786 for 8 bits and the other for only 1 bit of each pixel.

The displays discussed in the previous section obtained high resolutions at the expense of bits/pixel. Several 82786s can be combined to provide more bits/pixel at these high resolutions.

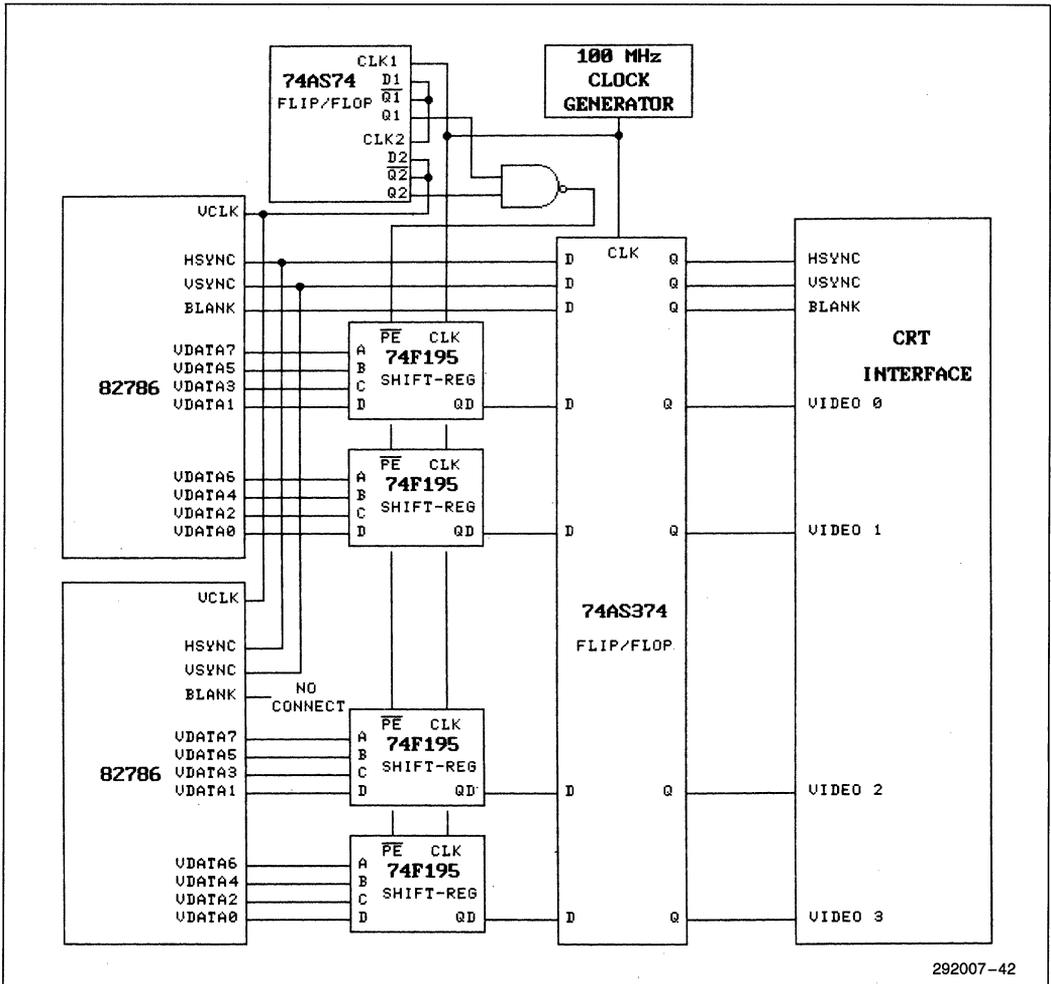
Figure 38 shows a configuration that uses two 82786s to create a 4-bit/pixel display at a video rate of 100 MHz. This configuration would support a  $1144 \times 860$  sixteen-color non-interlaced 60 Hz display. Each 82786 is required to generate 2 bits of each 4-bit/pixel. Therefore, both 82786s draw and maintain half of the bit-map in their own graphics memory, 4-bit/pixel windows are divided into two 2-bit/pixel bit-maps, one generated by each 82786. The Graphics Processors are programmed as normal for 2-bit/pixel bit-maps. The Display Processors are programmed the way mentioned in the previous section. Each window is programmed with one-fourth the horizontal positioning resolution.

### 5.8 Video RAM Interface

The 82786 can use dual-port video DRAMs (VRAMs) to generate the video data stream. The VR bit in the BIU Control Register must be set to 1 to enable the mode. In this mode the first tile in each strip generates VRAM cycles; subsequent tiles in the strip generate VRAM cycles. There is no limit on the number of strips. The pixel data for every scan line in the entire display must be contained in a single row in memory (256 words for non-interleaved memory and 512 words for interleaved memories). The strip descriptors for each tile are set up to indicate only 1 pixel. The address specified for this pixel corresponds to the first display pixel.

During the horizontal retrace period, the 82786 transfers the contents of the memory row containing the first pixel into the VRAM shift register. The VRAM shift clock is gated with a BLANK signal. During the active display time, the shift clock is active and periodically clocks out the video data. External multiplexers must be used to convert the 16 bit (32 interleaved) data stream into a serial stream depending upon the bits per pixel needed (Figure 9).

In this mode, pixel depth is fixed by external hardware and all Display Processor registers referring to video data fetch should be programmed to zero.



292007-42

Figure 38. Two 82786s Can Generate 100 MHz Video with 4 Bits/Pixel

### 5.9 External Character ROM

Few 82786 applications will require, or even benefit from, the use of an external character ROM.

The 82786 Graphics Processor can very rapidly draw characters. It can fill an 80x25 character screen with highly detailed 16x16 characters in less than one tenth of a second.

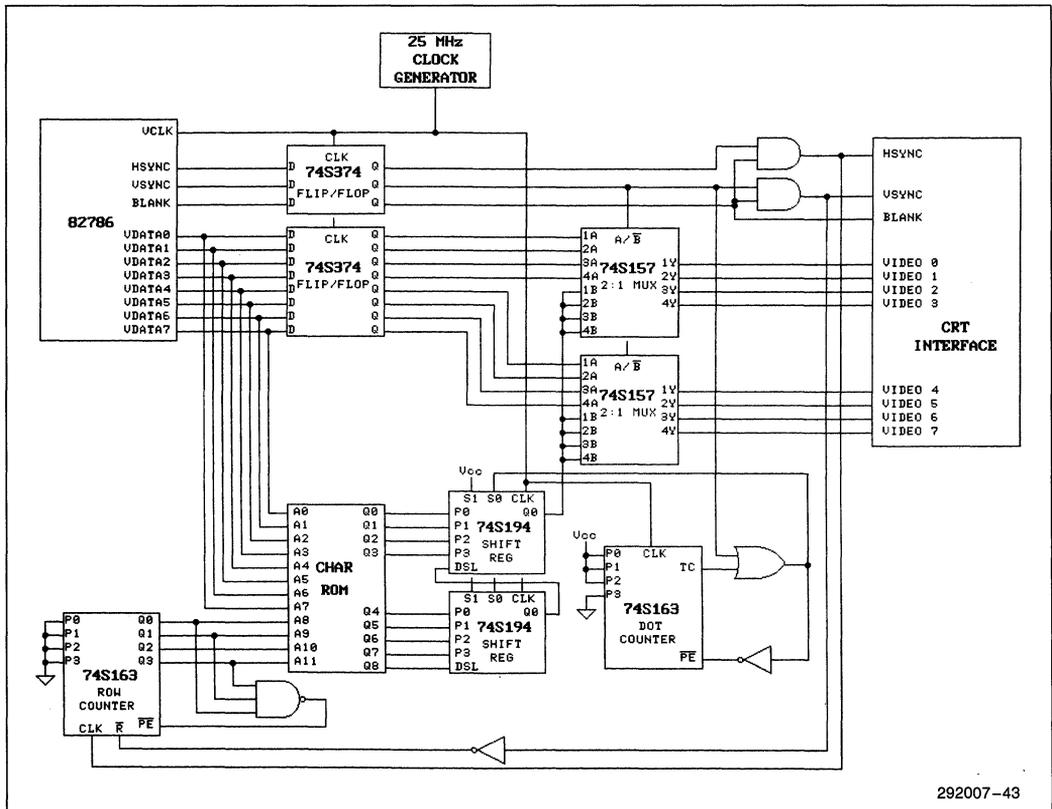
The Graphics Processor is also very flexible in the way it draws characters. Characters may be:

- formed from an unlimited number of character fonts
- placed at any pixel on the screen
- rotated in 4 directions with 4 paths
- combined with graphics
- drawn in any color
- have transparent or opaque background

A character ROM display forces characters from a pre-defined font to be restricted to character-cell positions on the screen with few, if any, of the above flexibilities.

For downward compatibility reasons, however, it may be necessary to provide the character ROM function in a 82786 system. Figure 39 illustrates a system capable of displaying both character ROM text and 82786 graphics. A multiplexer is used to switch between the character ROM output and the direct 82786 output. One of the window status bits is used to switch the multiplexer so both the character ROM and the 82786 graphics windows can be shown simultaneously on the same screen. It is important to delay the direct 82786 VDATA and window status signal the same number of clocks as the character-cell video so that all signals get to the multiplexer on the same clock. The extra D-flip/flops before the multiplexer are used to perform the needed delay.

The character ROM in Figure 39 is capable of displaying 256 characters using a 9x14 pixel character-cell. The characters are stored as an 8-bit pixels within a 82786 bit-map. To display the character, the window is programmed as an 8-bit/pixel bit-map with a horizontal zoom of 9 and a vertical zoom of 14. The 82786 will then place the 8-bit character code on its VDATA pins



292007-43

Figure 39. Support of Character-ROM and Bit-Mapped Graphics on Same Screen

during the scan lines when the character is to be displayed. The pixel counter is used to load the shift register every 9 pixels. This counter is synchronized to the beginning pixel of the window by starting when the window status pin falls. The row counter is used to supply row information to the character ROM. This counter is synchronized to the frame by starting from the end of the VSYNC pulse. Therefore, any character ROM window must begin at a multiple of 14 scan lines after VSYNC.

Another situation in which a character ROM display may be practical is if a very large character set is required. The Japanese Kanji characters are an example. The size of this character set is so large that it may be more practical to store the characters in a character ROM rather than load them from disk into the 82786 graphics memory. Figure 40 illustrates a configuration that can display up to 65536 characters from a very detailed (32x32) font. This circuit allows both text and graphics windows to be displayed on the screen simultaneously. One of the window status signals is used to select between text and graphics.

Such a character set requires a high resolution, generally monochrome display. The circuit in Figure 40 allows up to 200 MHz video (one bit/pixel) for very high resolution screens. The 82786 is programmed in super high-speed acceleration mode as described in Section 5.6.

The character-codes to be displayed should be placed in one bit/pixel bit-maps with 16 consecutive bits for each character. The hardware combines the 8-bit VDATA values from two consecutive pixels to generate the 16-bit character-code for the Character-ROM. If less than 65536 characters are required, not all of the 16-bit character code addresses need be used for the character-ROM. Some of these bits may be used for attributes such as blinking and reverse video. The ROM contains a 32x32 character font, each character is split up into 32-lines of four 8-bit bytes. The "pane" counter selects one of the four 8-bit bytes at a time. The "row" counter determines the current row of the character.

Character cell windows should be zoomed by 2 horizontally and by 32 vertically. The window must be placed at a multiple of 4-pixels from HSYNC and a multiple of 32-lines from VSYNC. It is possible to place windows at non-multiples from HSYNC and VSYNC if the "pane" and "row" counter parallel inputs are tied to other than ground.

## 5.10 Combining the 82786 With Other Video Sources

It is possible to combine graphics output from the 82786 with output from other video sources such as

broadcast TV, video recorders, and video laser disc players. The main requirement to perform such a feat is that both 82786 and the video source are locked in sync.

The 82786 has two independent Video Slave modes and HSYNC/VSYNC and BLANK can be independently configured as outputs or inputs. When HSYNC/VSYNC are programmed as inputs, then they are still outputs during the active display period if the window status is enabled. External HSYNC/VSYNC reset the 82786 horizontal and vertical counters respectively.

When BLANK is configured as output, the active display period is determined by the programmed values of VFLDSTRT, VFLDSTP, HFLDSTRT, and HFLDSTP. When BLANK is configured as an input, the external system determines the active display period. The internal video shift register generates video data only during the active display period.

HSYNC/VSYNC and BLANK would normally be programmed as input/output as follows:

HSYNC/ VSYNC	Blank	Application
Output	Output	Normal display generated by 82786
Input	Output	82786 generated display superimposed on externally-generated video
Input	Input	Multiple 82786 systems

The 82786 sync timing registers should be programmed to be as close to the frequency of the video source as possible. The 82786 should also be programmed for slave video-sync. The sync signals from the video source must be converted into separate TTL-level horizontal and vertical sync and fed to the 82786 HSYNC and VSYNC pins. The 82786 will then automatically sync itself up to the video source by waiting for its HSYNC input to fall before each scan line and waiting for its VSYNC input to fall before beginning a new display field.

For many applications, the 82786 video clock can be derived directly from a crystal oscillator. Since the 82786 syncs up to the nearest pixel on every scan line, even video sources with imperfect timings, such as video recorders where speed variations are common, will produce an acceptable picture. The frame-to-frame deviation of the 82786 graphics information on the screen relative to the video source will never be more than one pixel.

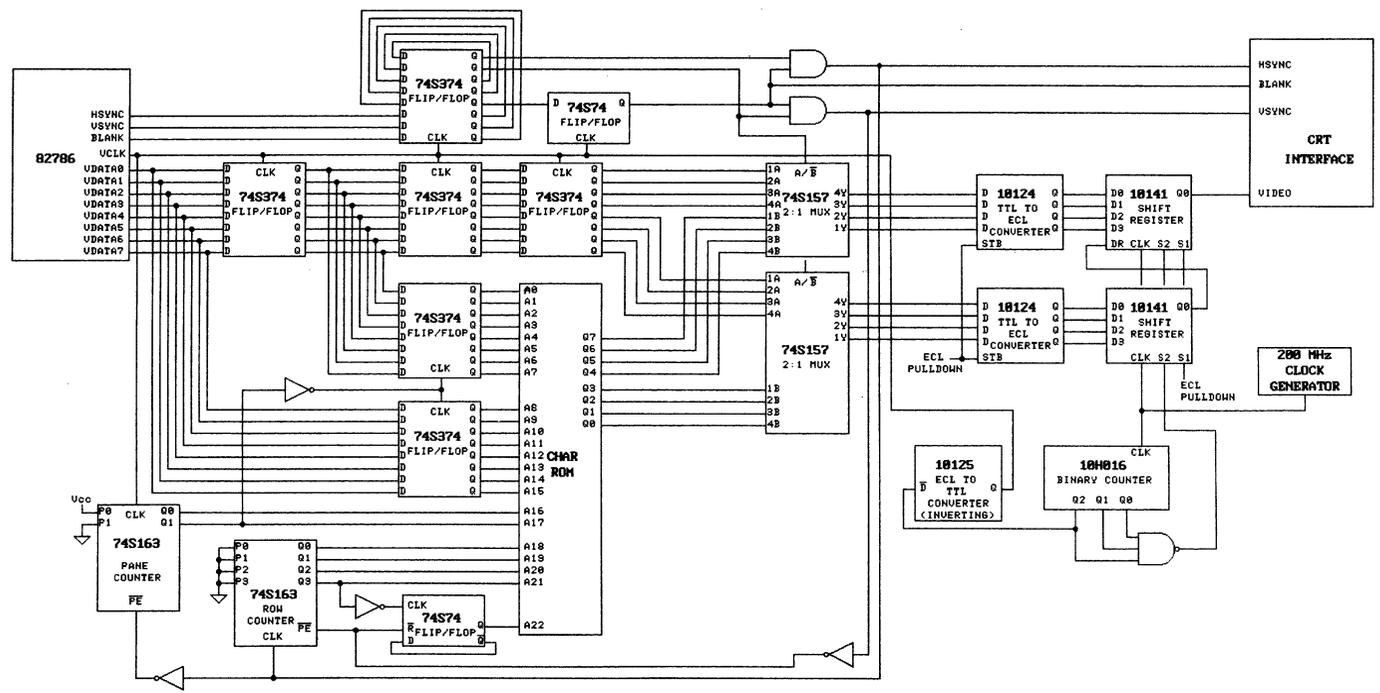


Figure 40. Support of Very Large Character-ROM and Bit-Map

292007-44

For more demanding applications, the 82786 video clock can be synthesized directly from the video source timings using a phase-locked-loop circuit. The 82786 will still sync itself up every scan line, but now the relationship between HSYNC and the 82786 VCLK will remain constant. This implementation will create virtually no deviation between the 82786 graphics and the video source.

In the case of interlaced video, care must be taken to initially start the 82786 display just prior to the VSYNC before an even-field. The 82786 initialization software is responsible to guarantee that the first LD\_ALL to start the 82786 display occurs sufficiently before the VSYNC during an odd-field so the first 82786 display field will match the video source even-field.

Once the 82786 is locked in sync with the video source, then the VDATA information from the 82786 can easily be combined with the video from the video source. Although the two video signals could be mixed on top of each other, probably the most common implementation is to switch between one or the other source. For example, the 82786 could create letters that are to be placed over the video picture. During the display scan, whenever a portion of a letter is to be displayed, the video from the 82786 can be switched in, otherwise the video source is switched in.

If the output of the video source is analog, the 82786 VDATA can be converted into an analog signal and an analog switch can be used. The state of the switch can be derived in a number of ways. If the switching is to be done on window boundaries, one window status pin can be used to control the switch. If the switching must be done within a window, a special graphics color code can be used to indicate that the 82786 video should be replaced with that from the video source. Possibly the color 11111111 could be placed on the VDATA pins and an 8-input NAND gate used to control the analog switch.

## 5.11 Other Types of Displays and Printers

The 82786 not only can be used with CRTs, but can also be used with other types of displays such as LCD, plasma, and intelligent printers. These devices have such a wide range of interface requirements that space does not permit each individual situation to be addressed in detail. Rather, some example requirements are discussed to illustrate how the 82786 can meet those needs.

### PIXEL CLOCK RATE

The rate at which pixels are clocked into displays can vary immensely. The 82786 allows a very wide range of video clock frequencies from DC levels to 25 MHz to accommodate such devices. In addition, faster clock rates can be generated using the method described in Section 5.6

### NO REFRESH

Printers and some displays are not required to be continuously refreshed. Needlessly running the 82786 Display Processor through refresh cycles steals bus bandwidth from the Graphics and other Processors. To eliminate this waste, the display can be turned off by resetting the DspOn bit (bit 0) in the Display Processor VStat Register (register number 0). When DspOn is reset, the Display Processor will continue to generate HSYNC, VSYNC, and BLANK and place Default-VDATA on the VDATA lines, but no bus bandwidth will be required.

When a change to the display is required, the DspOn bit can be set using the LD\_REG or LD\_ALL command. Once the refresh starts, another LD\_REG command to turn the display back off can be placed in the Display Processor Opcode Register. The Display Processor will then automatically execute it when the refresh is completed.

### PARTIAL DISPLAY UPDATES

Some displays that do not require continuous refresh, do have a long update time. It may take several seconds to update every pixel on the display. For small changes to the display, such as adding each character as it is typed by the user, it may be much more feasible to update only the portion of the display which is affected.

Using the very flexible windowing capability of the 82786, it is possible to only scan through a specific portion of the display.

### PIXEL ADDRESS GENERATION

Some displays, especially those which allow only partial display updates, require that pixel location addresses be generated along with the pixel data. Although external circuitry could be used to generate these addresses, the 82786 can be used to generate them directly. If a single 8-bit address is all that is required, the DefaultVDATA register can be programmed to a value that the VDATA pins will reflect during blank time. With proper programming of the sync timing registers, this value can be clocked into the display before each scan line using HSYNC.

More complex pixel addresses can be generated by using the 82786 windowing capability. By creating a thin window at the beginning of each scan line, one or more bytes of address information can be sequentially clocked out over the VDATA pins before each line.

### ULTRA HIGH RESOLUTION

Because some displays use either slow refresh times or don't require refresh at all, it is possible to have very high resolutions. All of the display counters in the 82786 are 12 bits allowing up to a  $4096 \times 4096$  display size (some of this resolution may not be available depending on the number of sync clock cycles required). Trading off bits/pixel for resolution, the accelerated modes can provide 2, 4, or 8 times this resolution horizontally, up to 32768 pixels.

Still, some applications, such as printers, may require even greater resolutions. This is possible with the 82786 using external counters to generate the HSYNC, VSYNC, and BLANK inputs for the 82786. The 82786 should be programmed for slave video mode by setting the CRTMode register (Display Processor register 5 bits 2 and 3 should be set). Using all 16 horizontal windows, the horizontal resolution may be up to  $4096 \times 16 = 65536$  pixels. Again, trading off bits/pixel for resolution, the accelerated modes can provide 2, 4, or 8 times this resolution, up to 524288 pixels. Vertically there is no limit to the resolution.

Such high resolutions require a lot of memory. For example, suppose a printer can generate 300 x 300 dots per square inch and is used on 8.5 x 11 inch paper. Assuming only one bit/pixel (no gray scale) the entire page would require:

$$\frac{300 \times 300 \times 8.5 \times 11}{8 \text{ bits}} = 1.05 \text{ Megabytes}$$

It may not be feasible to place this much memory into a printer. But it may be feasible to generate the display one strip at a time. Suppose that the first 300 lines are generated and printed. Once printed the next 300 lines can be generated and printed using the same memory. Now the memory required is only:

$$\frac{300 \times 300 \times 8.5}{8 \text{ bits}} = 96 \text{ Kbytes}$$

If the image to be printed can be described by a set of commands for the Graphics Processor, each strip can be very easily generated. The drawing bit-map and clipping rectangle are set for the first strip and the Graphics processor then runs through the command list. Once completed the strip may be printed. Then the bit-map and clipping rectangle are set for the second strip and the Graphics Processor again traverses the same command list to generate the second strip.

If there is enough memory for two strips, double buffering can be used to pipeline the operation. While the Display Processor is busy printing one strip, the Graphics Processor can be generating the next strip. The same approach can be extended to multiple pages, even using more than one 82786.

## 5.12 Calculating the Video Parameters

The 82786 video Display Processor is programmable to afford a wide variety of display formats. To determine the display format(s) that one would like to generate, several parameters must be considered.

**Application parameters:** these are dependent on the needs of the specific application and must be chosen by the designer.

**Hres**—horizontal resolution — number of pixels per horizontal line. When using accelerated video, Hres must be a multiple of Accel (following pages).

**Vres**—vertical resolution — number of vertical pixels (scan lines) per display

**Vfreq**—vertical frequency — rate at which CRT beam makes one pass from the top of the screen to the bottom. It is common to use 60 Hz but almost any other frequency can be generated by the 82786. US broadcast television standards use a 59.95 Hz rate. European video systems are based on a 50 Hz field rate. High resolution displays often use 40 Hz or lower. Slower rates reduce the speed requirements of the monitor and the 82786 video circuitry and also permit higher resolutions. However, slower rates also flicker more and may be intolerable to the operator. Generally, rates significantly under 60 Hz will tend to cause some perceptible flicker unless CRTs with long persistence phosphor are used.

**IIC**—interlacing — A non-interlaced display generates the entire display frame in one field scan. One method to double the resolution of a display is to use interlacing. Rather than use just one field to display all the information, two consecutive fields are used to create the entire display frame (Figure 41). Alternate scan lines are written during each alternate field. For TV-like pictures, where the image generally doesn't change drastically from one line to the next, interlacing allows a 30 Hz frame rate with a 60 Hz field rate without perceptible flicker. For detailed computer graphics, however, one line may change drastically from the next in color and/or intensity, in which case interlacing at such rates do cause perceptible flicker.

The 82786 supports both non-interlaced and interlaced displays. In addition, an interlaced-sync mode is available which generates sync signals in the manner used by interlaced displays, but generates the video signals in

the manner used by non-interlaced displays (both fields identical). This permits interlaced screens with consecutive pairs of lines identical.

**Monitor parameters:** these are dependent on the specific requirements of the display monitor used.

**Hblank**—horizontal blanking time — the time required for the CRT beam to jump from the right side of the display back to the left and stabilize. This is also called horizontal retrace time and is the sum of the horizontal sync and front and back porch times (Figure 42). Monitors typically range from 5–12  $\mu$ s.

**Vblank**—vertical blanking time — the time required for the CRT beam to jump from the bottom of the display back to the top and stabilize. This is also called vertical retrace time and is the sum of the vertical sync and front and back porch times (Figure 43). Monitors typically range from 600–1400  $\mu$ s.

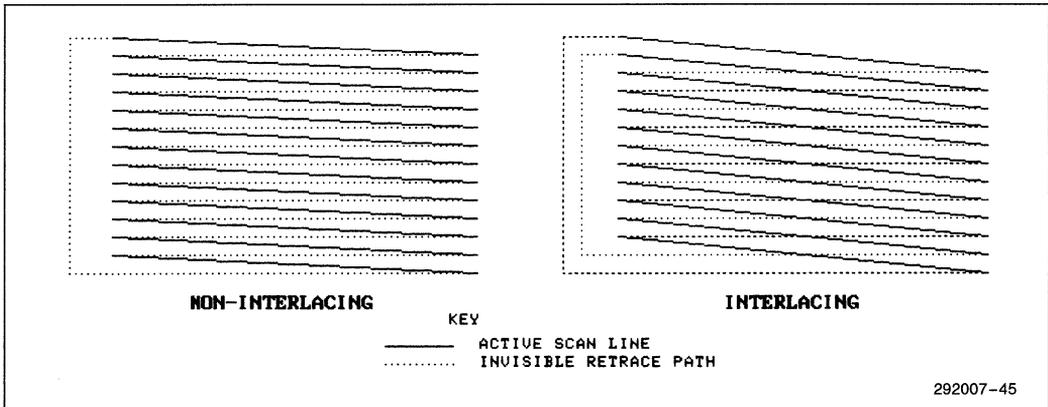


Figure 41. Non-Interlacing and Interlacing Displays

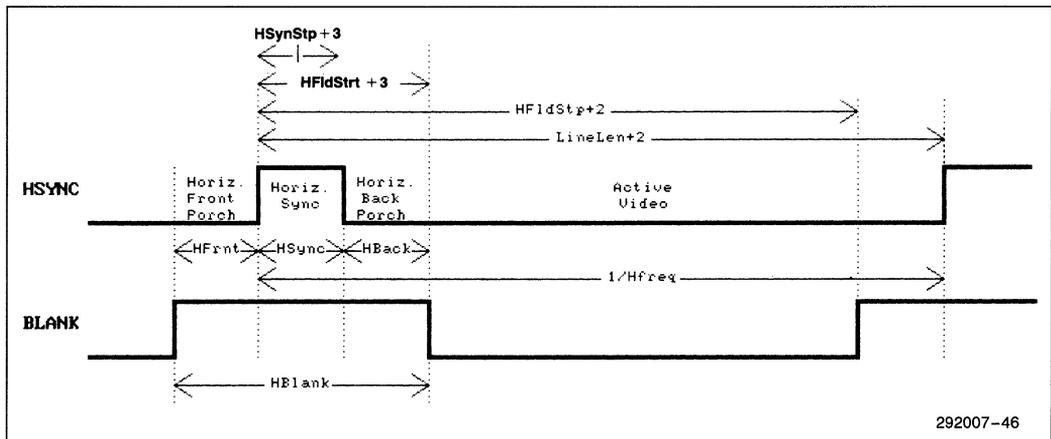
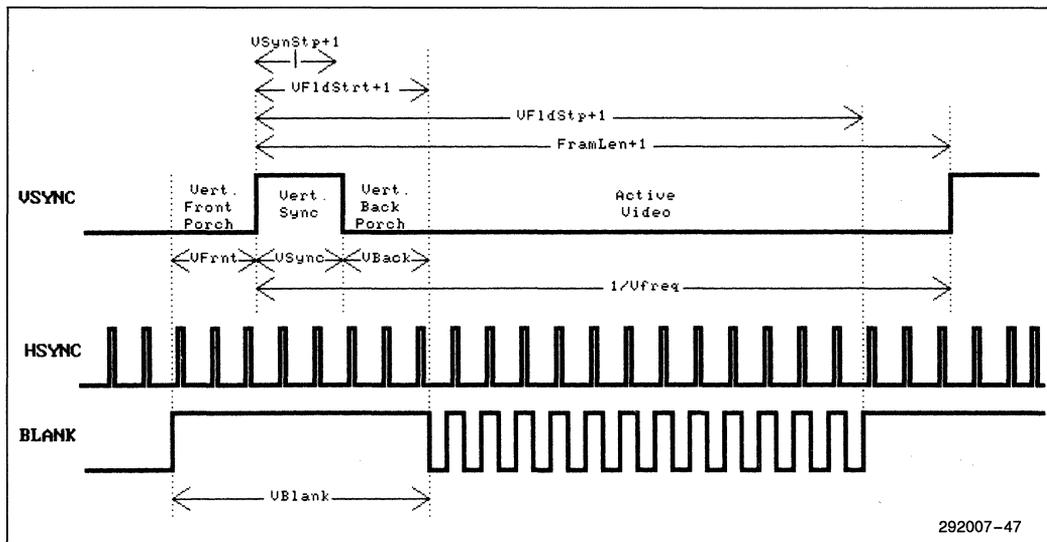


Figure 42. Horizontal Sync and Blank Timing Parameters


**Figure 43. Vertical Sync and Blank Timing Parameters**

**Hfreq**—horizontal frequency — the frequency at which horizontal lines are scanned. Monitors typically range from 15–36 kHz.

**Vfreq**—vertical frequency — the frequency at which the display field is scanned. Monitors typically range from 40–70 Hz.

**BPP**—bits per pixel — monitors with digital inputs restrict the number of usable bits/pixel. Monitors with analog inputs allow a virtually unlimited range of intensities with the use of Digital-to-Analog converters. This parameter is mainly dependent on the video interface hardware described in the previous sections.

Color monitors generally limit the perceivable horizontal and vertical resolution due to their shadow mask. See the specific monitor specifications for more details.

Video interface parameters: these are dependent on the 82786 component and the video interface logic.

**VCLK**—video clock frequency — the video input clock into the 82786. It has a maximum rate of 25 MHz and may be chosen so that the frequency evenly divides by both Hfreq and Vfreq.

**Accel**—82786 video acceleration — this parameter is determined by what mode the 82786 is used in. Normally Accel = 1. If the trade-offs mentioned in Section 5.6 are used to attain higher video rates at the expense of fewer bits/pixel, then the value for Accel should be 2, 4, or 8.

	Video Mode	Max DotClk	Programmed Accel Bits
Accel = 1	Normal	25 MHz	0 0
Accel = 2	High Speed	50 MHz	0 1
Accel = 4	Very High Speed	100 MHz	1 0
Accel = 8	Ultra High Speed	200 MHz	1 1

**DotClk**—pixel dot clock frequency — this is normally the same as VCLK. However, when accelerated video modes are used, this is either 2, 4, or 8 times VCLK.

$$\text{DotClk} = \text{VCLK} \times \text{Accel}$$

**HSynStp, HFldStrt, HFldStp, LineLen**—these are values programmed into the 82786 Display Processor to determine the horizontal scan timing (Figure 42). They may be set to any value from 0 to 4095. Their values should also fit the formula:

$$\text{HSynStp} < \text{HFldStrt} < \text{HFldStp} < \text{LineLen}$$

**VSynStp, VFldStrt, VFldStp, FramLen**—these are values programmed into the 82786 Display processor to determine the vertical scan timing (Figure 43). They may be set to any value from 0 to 4095. Their values should also fit the formula:

$$\text{VSynStp} < \text{VFldStrt} < \text{VFldStp} < \text{FramLen}$$

Once the above parameters are evaluated, the video parameters can actually be calculated. The parameters interact quite heavily so that, for example, if a specific

horizontal and vertical resolution at a specific field rate is required, the monitor frequencies and blank times may need to be altered. It may take several iterations to optimize all the parameters. The calculations can be performed by hand. However, a much easier way to manipulate these values is by using a spreadsheet program. A spreadsheet allows the parameters to be easily manipulated with their affects immediately displayed. A spreadsheet template for this purpose is given in Section 5.13.

The following formulas are used to determine the video parameters. Along with the formulas is an example calculation. For the example, let's generate a  $640 \times 400 \times 8$  bit/pixel (256 color) screen at 60 Hz non-interlaced. We will assume:

Hres	=	640 pixels
Vres	=	400 pixels
Vfreq%	=	60 Hz
Hblank%	=	12 $\mu$ s
Vblank%	=	1300 $\mu$ s
Accel	=	1 (no external acceleration)

Variables with a percent (%) after them represent desired values, the actual value will be calculated below.

ROUND(X) will be used to denote rounding off X to the nearest integer.

First, calculate the vertical resolution per field. Since our display is non-interlaced, the value is the same as the vertical resolution.

If interlaced then:  $VresFld = Vres/2$   
 else:  $VresFld = Vres$   
 $VresFld = 400$  pixels

With interlaced screens,  $VresFld$  is half the vertical resolution. For example, with 525 lines, use 262.5 for  $VresFld$ .

Now, calculate the horizontal frequency required. Subtract the vertical blank time from the vertical period and divide by the active vertical lines to obtain the horizontal period. Inverting all that gives the horizontal frequency.

$$\begin{aligned} Hfreq\% &= \frac{1}{Hperiod\%} = \frac{VresFld}{(1/Vfreq\%) - Vblank\%} \\ &= \frac{400}{(1/60) - 1300 \mu s} = 26.03 \text{ kHz} \end{aligned}$$

In a similar manner, calculate the pixel dot clock required.

$$\begin{aligned} \text{DotClk}\% &= \frac{1}{\text{DotPeriod}\%} = \frac{\text{Hres}}{(1/Hfreq\%) - \text{Hblank}\%} \\ &= \frac{640}{(1/26.03 \text{ kHz}) - 12 \mu s} = 24.23 \text{ MHz} \end{aligned}$$

And then calculate the actual 82786 VCLK. Since external acceleration circuits are not used in our example, it turns out to be the same as the DotClk.

$$\begin{aligned} \text{VCLK}\% &= \text{DotClk}\% / \text{Accel} = 24.23 \text{ MHz} / 1 \\ &= 24.23 \text{ MHz} \end{aligned}$$

Great, now all we need is a 24.23 MHz crystal is needed to generate VCLK. But since such a crystal is tough to find, try a 25 MHz crystal instead and see how it affects the rest of the parameters. First of all, the pixel dot clock changes.

$$\text{DotClk} = \text{VClk} \times \text{Accel} = 25.00 \text{ MHz} \times 1 = 25.00 \text{ MHz}$$

Now, see how many VCLK's are required for the horizontal blank time.

$$\begin{aligned} \text{HblankClks} &= \text{ROUND}(\text{VCLK} \times \text{Hblank}\%) = \text{ROUND} \\ &(25 \text{ MHz} \times 12 \mu s) = 300 \end{aligned}$$

Now we calculate the actual horizontal blank time.

$$\text{Hblank} = \frac{\text{HblankClks}}{\text{VCLK}} = \frac{300}{25 \text{ MHz}} = 12 \mu s$$

The actual horizontal period is then the time required to display one line of pixels plus the blanking time.

$$\begin{aligned} \text{Hfreq} &= \frac{1}{(\text{Hres} / \text{DotClk}) + \text{Hblank}} \\ &= \frac{1}{(640 / 25 \text{ MHz}) + 12 \mu s} = 26.60 \text{ kHz} \end{aligned}$$

The number of horizontal lines per field can now be calculated:

$$\begin{aligned} \text{VFieldLines} &= \text{ROUND}(\text{Hfreq} / \text{Vfreq}\%) \\ &= \text{ROUND}(26.60 \text{ kHz} / 60 \text{ Hz}) = 443 \end{aligned}$$

If an interlaced display is used,  $VFieldLines$  should be rounded-off to the closest odd integer.

The number of scan lines determines the actual vertical frequency:

$$\begin{aligned} \text{Vfreq} &= \text{Hfreq} / \text{VFieldLines} = 26.60 \text{ kHz} / 443 \\ &= 60.05 \text{ Hz} \end{aligned}$$

Now that the major parameters are calculated and we are satisfied with them, we can break up the blanking times into sync, front and back porch times. Typical monitor values are:

$$\begin{aligned} \text{HSync} &= 2 \mu\text{s} & \text{VSync} &= 300 \mu\text{s} \\ \text{HBack} &= 6 \mu\text{s} & \text{VBack} &= 800 \mu\text{s} \end{aligned}$$

$$\begin{aligned} \text{HSyncClks} &= \text{ROUND}(\text{VCLK} \times \text{HSync}) \\ &= \text{ROUND}(25 \text{ MHz} \times 2 \mu\text{s}) = 50 \end{aligned}$$

$$\begin{aligned} \text{HBackClks} &= \text{ROUND}(\text{VCLK} \times \text{HBack}) \\ &= \text{ROUND}(25 \text{ MHz} \times 6 \mu\text{s}) = 150 \end{aligned}$$

$$\begin{aligned} \text{VSyncClks} &= \text{ROUND}(\text{Hfreq} \times \text{VSYNC}) \\ &= \text{ROUND}(26.6 \text{ kHz} \times 300 \mu\text{s}) = 8 \end{aligned}$$

$$\begin{aligned} \text{VBackClks} &= \text{ROUND}(\text{Hfreq} \times \text{VBack}) \\ &= \text{ROUND}(26.6 \text{ kHz} \times 800 \mu\text{s}) = 21 \end{aligned}$$

Now it's a simple matter to calculate the values for the eight 82786 Display Processor video timing registers.

$$\begin{aligned} \text{HSynStp} &= \text{HSyncClks} - 3 \\ &= 50 - 3 = 47 \end{aligned}$$

$$\begin{aligned} \text{HFldStrt} &= \text{HSynStp} + \text{HBackClks} \\ &= 47 + 150 = 197 \end{aligned}$$

$$\begin{aligned} \text{HFldStp} &= \text{HFldStrt} + (\text{Hres} / \text{Accel}) \\ &= 197 + (640 / 1) = 837 \end{aligned}$$

$$\begin{aligned} \text{LineLen} &= \text{HBlankClks} + (\text{Hres} / \text{Accel}) - 3 \\ &= 300 + (640 / 1) - 3 = 937 \end{aligned}$$

For non-interlaced displays:

$$\begin{aligned} \text{VSynStp} &= \text{VSyncLines} - 1 \\ &= 8 - 1 = 7 \end{aligned}$$

$$\begin{aligned} \text{VFldStrt} &= \text{VSyncStp} + \text{VBackLines} \\ &= 7 + 21 = 28 \end{aligned}$$

$$\begin{aligned} \text{VFldStp} &= \text{VFldStrt} + \text{Vres} \\ &= 28 + 400 = 428 \end{aligned}$$

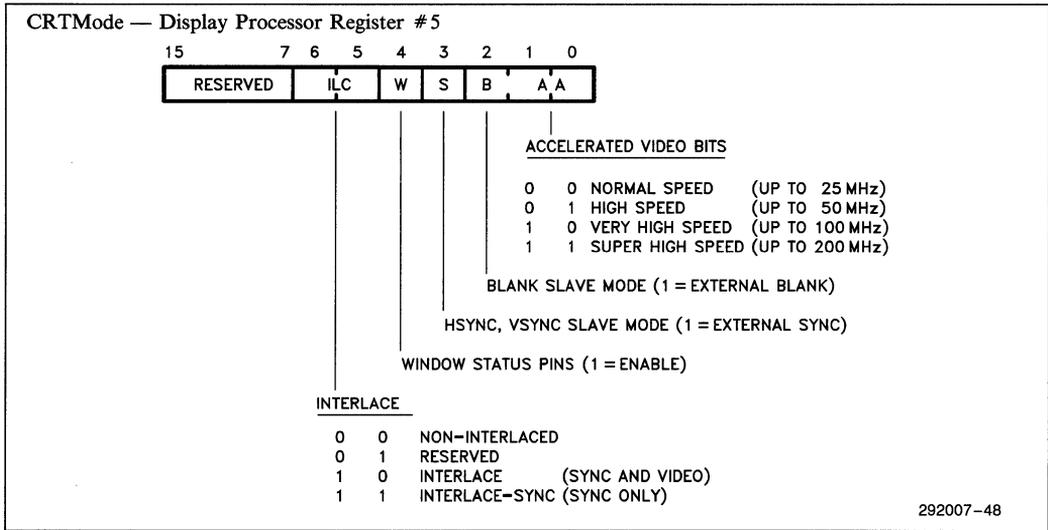
$$\begin{aligned} \text{FramLen} &= \text{VFieldLines} - 1 \\ &= 443 - 1 = 442 \end{aligned}$$

For interlaced displays:

$$\begin{aligned} \text{VSynStp} &= (\text{VSyncLines} - 1) \times 2 \\ \text{VFldStrt} &= \text{VSynStp} + (\text{VBackLines} \times 2) \\ \text{VFldStp} &= \text{VFieldsLines} - 2 \end{aligned}$$

Make sure  $\text{LineLen} > \text{HFldStp}$  and that  $\text{FramLen} > \text{VFieldLines}$ . If not, your parameters are inconsistent and you should modify your requirements and re-calculate.

Finally, the bits for the CRTMode Register should be determined. For our example, non-interlaced mode is used and no accelerated video is required. Assuming the 82786 is used to generate the HSYNC, VSYNC and BLANK signals and assuming the window Status pins are not used, the CRTMode registers should be loaded with all zeros.



The host CPU software is required to load the values of the eight video timing registers and the CRTMode register. Generally, this is done during system initialization. The registers should all be loaded simultaneously using the LD\_ALL command rather than using individual LD\_REG commands. This ensures that the video sync signals are never invalid while registers are being loaded.

Some CRTs can be permanently damaged by supplying the wrong sync frequencies to them. To prevent invalid video sync signals, the HSYNC, VSYNC, and BLANK

pins are tristated after RESET until the CRTMode Register has been written to.

### 5.13 A Spreadsheet for Calculating Video Parameters

As seen in the previous section, quite a number of calculations are required to determine the 82786 video parameter constants. Often several iterations through the calculations are required to optimize the display format. This process can be greatly simplified by using a spreadsheet.

An example of the output from such a spreadsheet is shown below. This example illustrates a 1290 x 968 x 4-bit/pixel (16 color) interlaced 60 Hz display. The user has supplied all of the values under the "DESIRED" column and the spreadsheet program has calculated the rest. The "ACTUAL" column shows the closest timings and parameters that the 82786 can actually supply. The "82786 DP REGISTER VALUES" shows the values that should be programmed into the Display Processor registers to generate such a display.

The User can easily modify the "DESIRED" values until the "ACTUAL" values meet the application's needs. Care should be taken to ensure that all "ACTUAL" values are logically correct. If for example, any of the calculated parameters are negative, then the set of "DESIRED" parameters can not produce such a display, so some parameters must be adjusted.

8 2 7 8 6      V I D E O      P A R A M E T E R S

Type under DESIRED column only: ACTUAL & REGISTER columns are calculated

PARAMETER	DESIRED	ACTUAL	82786 DP REGISTER VALUES
Video Clock    VCLK (MHz):	25	25	
Acceleration (1,2,4 or 8):	2	2	
Interlacing (1 = no, 2 = yes):	2	2	
Horiz Resolution (Pixels):	1290	1290	
Vert. Resolution (Pixels):	968	968	
Horiz Line Rate    (kHz):	---	30.487	LineLen:        818
Horiz Sync Width    (μs):	2	2	HSynStp:        48
Horiz Back Porch    (μs):	4	4	HFldStrt:       148
Horiz Front Porch    (μs):	1	1	HFldStp:        793
Vert. Frame Rate    (Hz):	60	59.956	FramLen:        1015
Vert. Sync Width    (μs):	200	196.8	VSynStp:        10
Vert. Back Porch    (μs):	400	393.6	VFldStrt:       34
Vert. Front Porch    (μs):	---	213.2	VFldStp:        1002

The template follows. This template should be easily adaptable to nearly any spreadsheet program. This particular spreadsheet program uses @ROUND(X,0) to denote rounding to the nearest integer. If no rounding function is available in your spreadsheet program, you can substitute the integer function (which truncates the fractional portion to return the next lowest integer) for the round function:

substitute @INT(X+0.5) for @ROUND(X,0)

After entering the template into your favorite spreadsheet, you may wish to verify that it is working correctly by entering the "DESIRED" values of the above example and checking that the "ACTUAL" and "REGISTER" results match.

-----A-----		---B---		-----C-----		---D---		-----E-----	
1:	8 2 7 8 6	V I D E O P A R A M E T E R S							
2:	Type under DESIRED column only: ACTUAL & REGISTER columns are calculated								
3:	-----								
4:	PARAMETER	DESIRED	ACTUAL	82786 DP REGISTER VALUES					
5:	-----	-----	-----	-----					
6:	Video Clock VCLK (MHz):		+B6						
7:	Acceleration (1,2,4 or 8):		+B7						
8:	Interlacing (1=no, 2=yes):		+B8						
9:	Horiz Resolution (Pixels):		@ROUND(B9/C7,0)*C7						
10:	Vert. Resolution (Pixels):		@ROUND(B10,0)						
11:									
12:	Horiz Line Rate (kHz):	---	(C6*1000)/(E12+2)	LineLen:	@ROUND(C6*B15,0)+E15				
13:	Horiz Sync Width (μs):		(E13+2)/C6	HSynStp:	@ROUND(C6*B13,0)-3				
14:	Horiz Back Porch (μs):		(E14-E13)/C6	HFldStrt:	@ROUND(C6*B14,0)+E13				
15:	Horiz Front Porch (μs):		(E12-E15)/C6	HFldStp:	+E14+(C9/C7)				
16:									
17:	Vert. Frame Rate (Hz):		(C8*C12*1000)/(E17+C8)	FramLen:	@ROUND((C12*1000)/B17-(C8-1)/2,0)*C8-1				
18:	Vert. Sync Width (μs):		((E18+C8)*1000)/(C12*C8)	VSynStp:	(@ROUND((C12*B18)/1000,0)-1)*C8				
19:	Vert. Back Porch (μs):		((E19-E18)*1000)/(C12*C8)	VFldStrt:	@ROUND((C12*B19)/1000,0)*C8+E18				
20:	Vert. Front Porch (μs):	---	(E17-E20)*1000/(C12*C8)	VFldStp:	+E19+C10				



## APPENDIX A

### SAMPLE INITIALIZATION CODE

Many registers within the 82786 must be initialized to configure the 82786 for the particular hardware environment it resides in. This appendix contains assembly language code to initialize the 82786 for one particular configuration:

- synchronous 10 MHz 80286 interface (Sections 4.2 and 4.3, Figure 18)
- one row of two interleaved banks of 51C256 Fast-page-mode DRAM (Section 3.3, Figure 9)
- 640 x 300 x 8-bit/pixel non-interlaced 60 Hz display, 25 MHz VCLK (Section 5.11, Figure 27)

All of the parameters to be initialized for this configuration are calculated under their corresponding sections in the body of this application note. To calculate the parameters for other configurations, refer to these sections.

This example of initialization code can be used to initially test many of the hardware functions. The code should create a stable display on the CRT. The display will consist of a black field which covers the entire screen (a 640 x 400 black rectangle). In the center of the rectangle is a 16 x 16 pixel arrow-shaped red and yellow cursor.

```
name Initialization82786
```

```
Memory82786 segment at 0C000h
```

```
    ;segment located at start of CPU-mapped 82786 memory
```

```
    ;define locations of 82786 internal registers
```

```
        org 0
```

```
Internalrelocation    dw    ?    ;BIU registers
Reserved              dw    ?
BIUControl            dw    ?
RefreshControl        dw    ?
DRAMControl           dw    ?
DisplayPriority        dw    ?
GraphicsPriority       dw    ?
ExternalPriority       dw    ?
```

```
        org 20h
```

```
GP0opcode             dw    ?    ;Graphics Processor registers
GPLinkAddressLower    dw    ?
GPLinkAddressUpper    dw    ?
GPStatus              dw    ?
GPInstructionPtrLower dw    ?
GPInstructionPtrUpper dw    ?
```

```
        org 40h
```

```
DPOopcode             dw    ?    ;Display Processor registers
DPPParameter1         dw    ?
DPPParameter2         dw    ?
DPPParameter3         dw    ?
DPStatus              dw    ?
DefaultVDATA          dw    ?
```

;location of values for Display Processor LD\_ALL instruction

org 80h

DPLdAllRegs label word

```

dw 3 ;VStat: turn on display and cursor
dw OFFh ;IntMask: mask all interrupts
dw 24 ;TripPt: trip point = 24 FIFO dwords
dw 0 ;Frint: cause interrupt every frame (interrupt is masked)
dw 0 ; reserved
dw 0 ;CRTMode: non-interlaced, non-accelerated, master sync&blank
dw 47 ;HSynStp: horizontal sync stop :
dw 197 ;HFldStrt: horizontal field start :
dw 837 ;HFldStp: horizontal field stop : 8 video timing registers
dw 937 ;LineLen: horizontal line length : are programmed for
dw 7 ;VSynStp: vertical sync stop : 640 x 400 at 60 Hz
dw 28 ;VFldStrt: vertical field start : with 25 MHz VCLK
dw 428 ;VFldStp: vertical field stop :
dw 442 ;FramLen: vertical frame length :
dw offset WinDescL ;DescAddrL:descriptor address pointer lower
dw 0 ;DescAddrU:descriptor address pointer upper
dw 0 ;(Reserved)
db 0 ;ZoomY: no vertical zoom
db 0 ;ZoomX: no horizontal zoom
dw 0 ;FldColor: black field color
dw OFFh ;BdrColor: white border color
dw 0 ;Pad1BPP: pad with zeros for 1 bit/pixel
dw 0 ;Pad2BPP: pad with zeros for 2 bits/pixel
dw 0 ;Pad4BPP: pad with zeros for 4 bits/pixel
db 2 ;CursorPad:pad with red for cursor (yellow cursor in red box)
db 80h ;CsrStyle: opaque 16x16 block cursor, no window status
dw 510 ;CsrPosX: put cursor in middle of screen (horizontally)
dw 220 ;CsrPosY: put cursor in middle of screen (vertically)

dw 0000000110000000b ;CsrPat0: create arrow-shaped cursor pattern
dw 0000001111000000b ;CsrPat1:
dw 0000011111100000b ;CsrPat2:
dw 0000111111110000b ;CsrPat3:
dw 0001111111111000b ;CsrPat4:
dw 0011111111111100b ;CsrPat5:
dw 0111111111111110b ;CsrPat6:
dw 1111111111111111b ;CsrPat7:
dw 0000011111100000b ;CsrPat8:
dw 0000011111100000b ;CsrPat9:
dw 0000011111100000b ;CsrPatA:
dw 0000011111100000b ;CsrPatB:
dw 0000011111100000b ;CsrPatC:
dw 0000011111100000b ;CsrPatD:
dw 0000011111100000b ;CsrPatE:
dw 0000011111100000b ;CsrPatF:

```

;location of strip descriptor list

WinDescL label word ;strip descriptor list

```

;header of strip descriptor
dw 399 ;lines in strip (400 covers entire screen)
dw 0 ;lower link to next strip descr (there is none)
dw 0 ;upper link to next strip descr (there is none)
dw 0 ;number of tiles in strip (only one)

```

```

                                ;first (and only) tile descriptor
dw      0                        ;bit-map width (not applicable, this is field)
dw      0                        ;memory start lower addr (not applicable)
dw      0                        ;memory start upper addr (not applicable)
dw      639                      ;field width (640 covers entire screen)
dw      0                        ;fetch count (not applicable, this is field)
dw      00001h                   ;set field bit,use top,bottom,left,right borders

```

Memory82786 ends

Initialize82786 segment ;code to initialize 82786

```

mov     ax,seg BIUControl
mov     ds,ax                    ;put 82786 register segment in ds

assume cs:Initialize82786, ds:Memory82786

mov     byte ptr BIUControl, 30h   ;convert 82786 to 16-bit bus...
mov     byte ptr BIUControl+1, 0   ;...must use two 8-bit transfers

mov     InternalRelocation, 01h    ;locate reg's at 82786 mem addr 0h

mov     DRAMControl, 1Dh           ;1 row, interleaved 51C256 DRAM
mov     RefreshControl, 18         ;request refresh every 15.2 uS

mov     DisplayPriority, 110110b   ;set Display FPL, SPL = 6

mov     GraphicsPriority, 010010b  ;set Graphics FPL, SPL = 2
mov     ExternalPriority, 100000b  ;set External FPL      = 4

mov     DPPParameter1, offset DPLdAllRegs ;address for LD_All command
mov     DPPParameter2, 0CH
mov     DPOpcode, 5                ;let DP perform LD_All command

ret                                     ;end of initialization subrtn

```

Initialize82786 ends

If the constants in the CPU-mapped 82786 memory for the LD—ALL command and the strip descriptor list (in Memory82786 segment) cannot be loaded into 82786 memory by the system's program loader, they will have to be loaded by the initialization code. One method is to have the loader load them into CPU system memory and use a repeat-move-string command in the initialization code to move these constants into the 82786 graphics memory. Alternatively, it is possible to place these constants in the 82786-mapped CPU memory and allow the 82786 to fetch them using master-mode. This method, however, is not as efficient because the 82786 must re-fetch the strip descriptor list for every display frame.

The Graphics Processor is not used in this initialization code. To fully initialize the Graphics Processor, the following commands are required:

Def_Bit_Map	for all drawing and BitBlt commands
Def_Logical_Op	for all drawing and BitBlt commands
Def_Colors	if line/character drawing used
Def_Texture	if line drawing used
Def_Char_Set	if character drawing used
Def_Char_Orient	if character drawing used
Def_Char_Space	if character drawing used
Load_Reg	initialize stack pointer if macros used
Load_Reg	set poll-on-exception mask if used
Load_Reg	set interrupt mask if interrupts used



INTEL CORPORATION, 3065 Bowers Ave., Santa Clara, CA 95051; Tel. (408) 987-8080

INTEL CORPORATION (U.K.) Ltd., Swindon, United Kingdom; Tel. (0793) 696 000

INTEL JAPAN k.k., Ibaraki-ken; Tel. 029747-8511

102685319