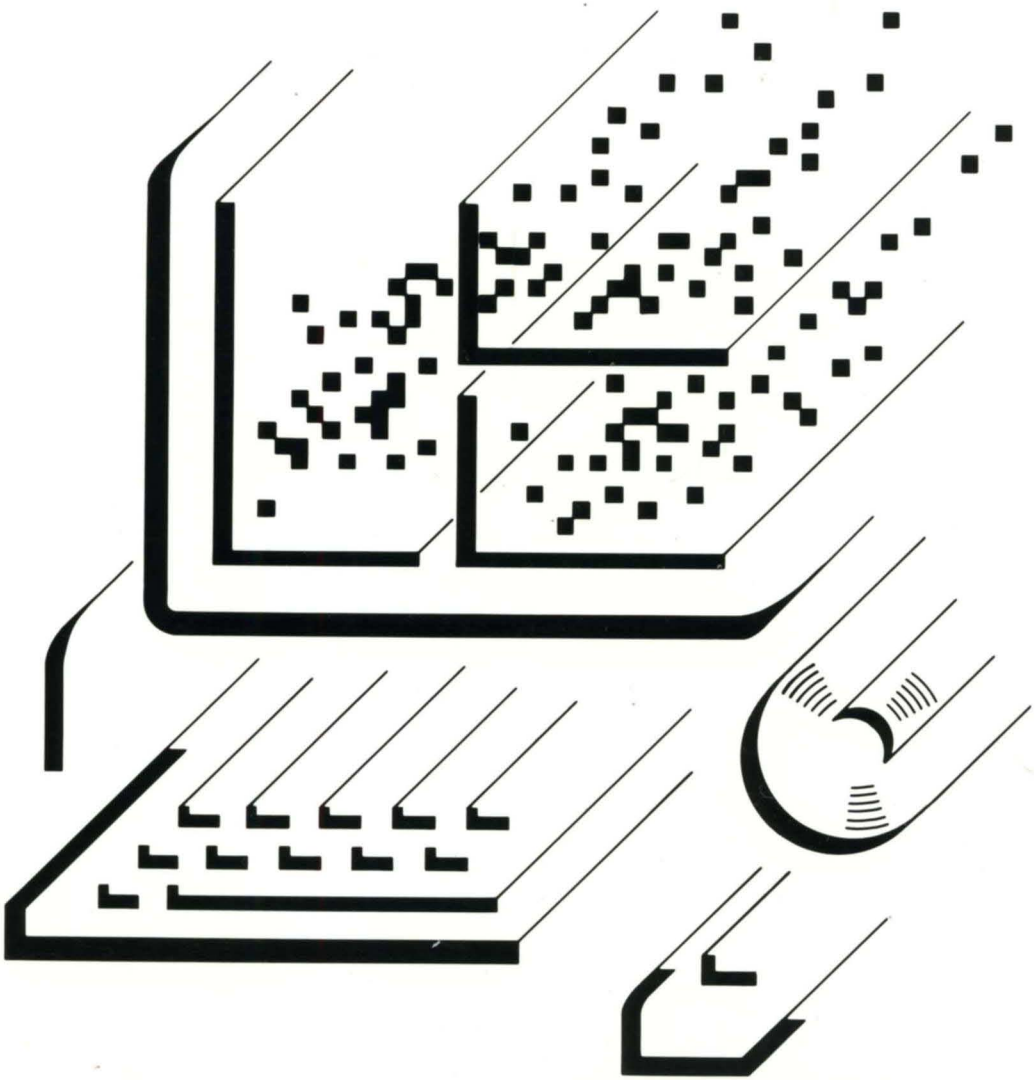
intel®

DVI® Technology
i750® Video Processor
Technical Specifications



The i750® Video Processor Technical Specifications

Contents:

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:

287, 376, 386, 387, 486, 4-SITE, Above, ACE51, ACE96, ACE186, ACE196, ACE960, ActionMedia, BITBUS, COMMputer, CREDIT, Data Pipeline, DVI, ETOX, FaxBACK, Genius, i, $\overset{\Delta}{1}$, i486, i750, i860, ICE, iCEL, ICEVIEW, iCS, iDBP, iDIS, i^2ICE, iLBX, iMDDX, iMMX, Inboard, Insite, Intel, int$_e$l, Intel386, int$_e$lBOS, Intel Certified, Intelevision, int$_e$ligent Identifier, int$_e$ligent Programming, Intellec, Intellink, iOSP, iPAT, iPDS, iPSC, iRMK, iRMX, iSBC, iSBX, iSDM, iSXM, Library Manager, MAPNET, MCS, Megachassis, MICROMAINFRAME, MULTICHANNEL, MULTIMODULE, MultiSERVER, ONCE, OpenNET, OTP, Pro750, PROMPT, Promware, QUEST, QueX, Quick-Erase, Quick-Pulse Programming, READY LAN, RMX/80, RUPI, Seamless, SLD, SugarCube, ToolTALK, UPI, VAPI, Visual Edge, VLSiCEL, and ZapCode, and the combination of ICE, iCS, iRMX, iSBC, iSBX, iSXM, MCS, or UPI and a numerical suffix.

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.

CHMOS and HMOS are patented processes of Intel Corp.

Intel Corporation and Intel's FASTPATH are not affiliated with Kinetics, a division of Excelan, Inc. or its FASTPATH trademark or products.

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

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

CG-090190

ADVANCE INFORMATION

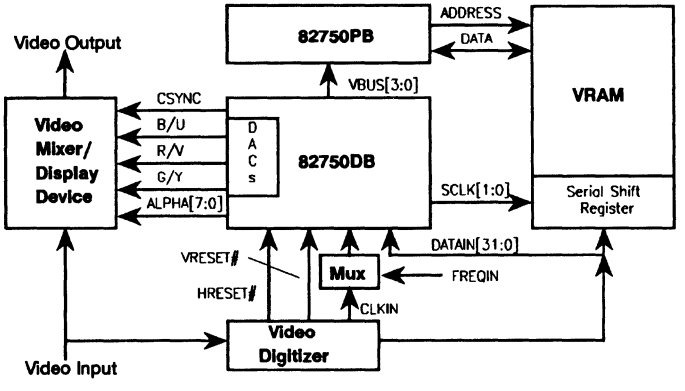
82750PB
PIXEL PROCESSOR

- **25 MHz Clock with Single Cycle Execution**
- **Zero Branch Delay**
- **Wide Instruction Word Processor**
- **512 x 48-Bit Instruction RAM**
- **512 x 16-Bit Data RAM**
- **Two Internal 16-Bit Buses**
- **ALU with Dual-Add-With-Saturation Mode**
- **Variable Length Sequence Decoder**
- **Pixel Interpolator**
- **High Performance Memory Interface**
  - **— 32-Bit Memory Data Bus**
  - **— 50 MBytes per Second Maximum**
  - **— 25 MBytes per Second with Standard VRAMs or DRAMs**
- **16 General-Purpose Registers**
- **4 Gbyte Linear Address Space**
- **132-Pin PQFP**
- **Compatible with the 82750PA**

The 82750PB is a 25 MHz wide instruction processor that generates and manipulates pixels. When paired with its companion chip, the 82750DB, and used to implement a DVI Technology video subsystem, the 82750PB provides real time (30 images/sec) pixel processing, real time video compression, interactive motion video playback and real time video effects.

Real time pixel manipulations, including 30 images/sec video compression, are supported by the 25 MHz instruction rate. On-chip instruction RAM provides programmability for execution of a wide range of algorithms that support motion video decompression, text, and 2D and 3D graphics. Inner loops are optimized with the integration of sixteen 16-bit quad ported registers, on-chip DRAM, and two loop counters that provide zero delay two-way branching "free" in any instruction. Two, 16-bit internal buses enable two parallel register transfers on each 82750PB instruction, contributing to the real time performance of the video processing. Another feature that adds to the processing power of the 82750PB is the 16-bit ALU, which includes an 8-bit dual-add-with-saturate operation critical for pixel arithmetic. Other specialized features for pixel processing include a 2D pixel interpolator for image processing functions and a variable length sequence decoder for decoding compressed data.

The 82750PB is implemented using Intel's low-power CHMOS IV Technology and is packaged in a 132-lead space-saving, plastic quad flat pack (PQFP) package.



240854-1

82750PB Subsystem Diagram

82750PB Pixel Processor

CONTENTS

82750PB Pixel Processor

CONTENTS

FIGURES

CONTENTS

TABLES

82750PB Pixel Processor

CONTENTS

CONTENTS

1.0 82750PB PIN DESCRIPTION

Pinout

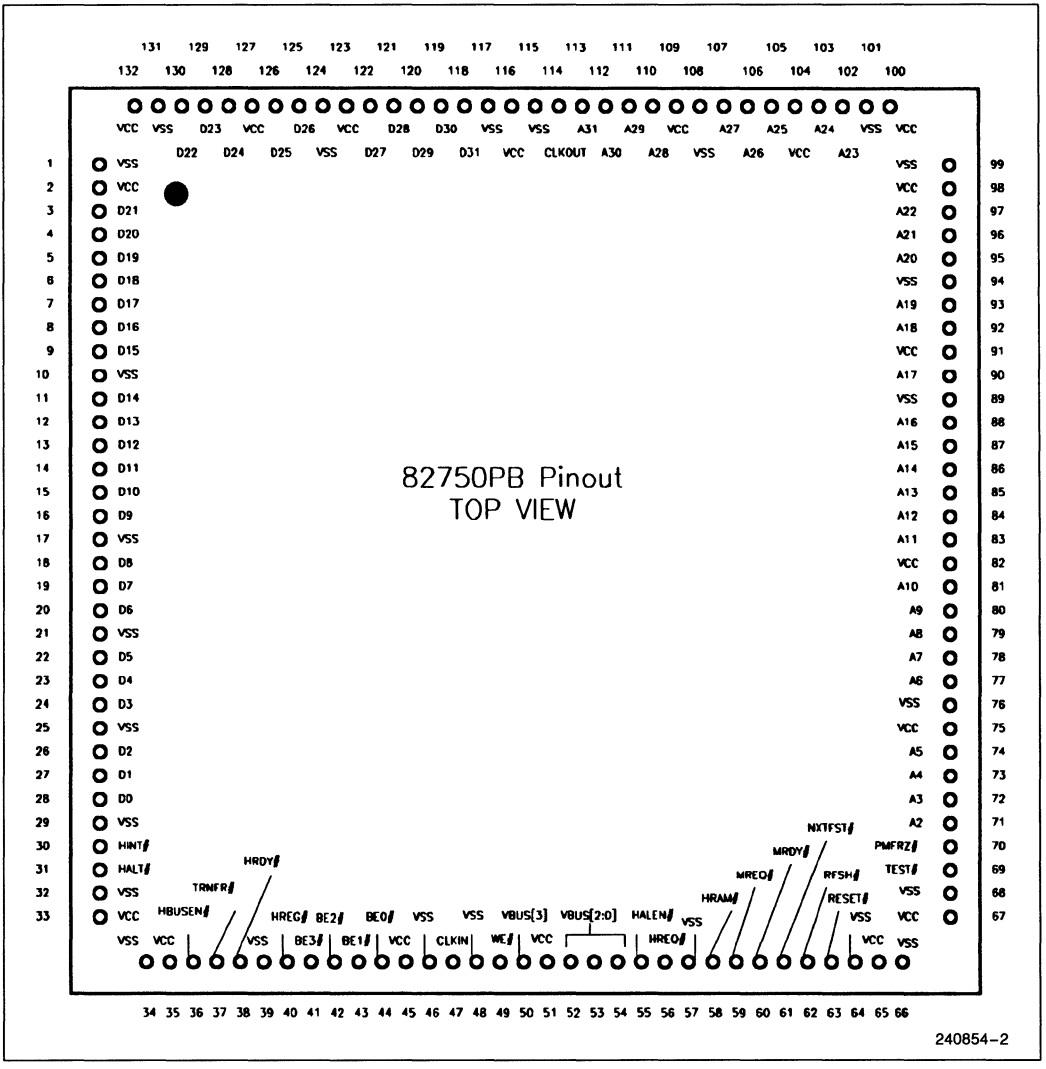


Figure 1-1. 82750PB Pinout

Table 1-1. Pin Cross Reference by Pin Name

Pin Name	Location	Pin Name	Location	Pin Name	Location	Pin Name	Location
A2	71	BE3\	41	D30	119	V_{CC}	100
A3	72	CLKIN	47	D31	118	V_{CC}	104
A4	73	CLKOUT	114	HALEN#	55	V_{SS}	94
A5	74	D0	28	HALT#	31	V_{CC}	109
A6	77	D1	27	HBUSEN#	36	V_{CC}	116
A7	78	D2	26	HINT#	30	V_{CC}	123
A8	79	D3	24	HRAM#	58	V_{CC}	127
A9	80	D4	23	HRDY#	38	V_{CC}	132
A10	81	D5	22	HREG#	40	V_{SS}	1
A11	83	D6	20	HREQ#	56	V_{SS}	32
A12	84	D7	19	MRDY#	60	V_{SS}	34
A13	85	D8	18	MREQ#	59	V_{SS}	39
A14	86	D9	16	NXTFST#	61	V_{SS}	48
A15	87	D10	15	PMFRZ#	70	V_{SS}	57
A16	88	D11	14	RESET#	63	V_{SS}	66
A17	90	D12	13	RFSH#	62	V_{SS}	68
A18	92	D13	12	TEST#	69	V_{SS}	76
A19	93	D14	11	TRNFR#	37	V_{SS}	89
A20	95	D15	9	VBUS[0]	54	V_{SS}	99
A21	96	D16	8	VBUS[1]	53	V_{SS}	101
A22	97	D17	7	VBUS[2]	52	V_{SS}	108
A23	102	D18	6	VBUS[3]	50	V_{SS}	115
A24	103	D19	5	V_{CC}	2	V_{SS}	117
A25	105	D20	4	V_{CC}	33	V_{SS}	124
A26	106	D21	3	V_{CC}	35	V_{SS}	131
A27	107	D22	130	V_{CC}	45	V_{SS}	10
A28	110	D23	129	V_{CC}	51	V_{SS}	17
A29	111	D24	128	V_{CC}	65	V_{SS}	21
A30	112	D25	126	V_{CC}	67	V_{SS}	25
A31	113	D26	125	V_{CC}	75	V_{SS}	29
BE0\	44	D27	122	V_{CC}	82	V_{SS}	46
BE1\	43	D28	121	V_{CC}	91	V_{SS}	64
BE2\	42	D29	120	V_{CC}	98	WE#	49

Table 1-2. Pin Cross Reference by Location

Location	Pin Name	Location	Pin Name	Location	Pin Name	Location	Pin Name
1	V_{SS}	34	V_{SS}	67	V_{CC}	100	V_{CC}
2	V_{CC}	35	V_{CC}	68	V_{SS}	101	V_{SS}
3	D21	36	HBUSEN#	69	TEST#	102	A23
4	D20	37	TRNFR#	70	PMFRZ#	103	A24
5	D19	38	HRDY#	71	A2	104	V_{CC}
6	D18	39	V_{SS}	72	A3	105	A25
7	D17	40	HREG#	73	A4	106	A26
8	D16	41	BE3#	74	A5	107	A27
9	D15	42	BE2#	75	V_{CC}	108	V_{SS}
10	V_{SS}	43	BE1#	76	V_{SS}	109	V_{CC}
11	D14	44	BE0#	77	A6	110	A28
12	D13	45	V_{CC}	78	A7	111	A29
13	D12	46	V_{SS}	79	A8	112	A30
14	D11	47	CLKIN	80	A9	113	A31
15	D10	48	V_{SS}	81	A10	114	CLKOUT
16	D9	49	WE#	82	V_{CC}	115	V_{SS}
17	V_{SS}	50	VBUS[3]	83	A11	116	V_{CC}
18	D8	51	V_{CC}	84	A12	117	V_{SS}
19	D7	52	VBUS[2]	85	A13	118	D31
20	D6	53	VBUS[1]	86	A14	119	D30
21	V_{SS}	54	VBUS[0]	87	A15	120	D29
22	D5	55	HALEN#	88	A16	121	D28
23	D4	56	HREQ#	89	V_{SS}	122	D27
24	D3	57	V_{SS}	90	A17	123	V_{CC}
25	V_{SS}	58	HRAM#	91	V_{CC}	124	V_{SS}
26	D2	59	MREQ#	92	A18	125	D26
27	D1	60	MRDY#	93	A19	126	D25
28	D0	61	NXTFST#	94	V_{SS}	127	V_{CC}
29	V_{SS}	62	RFSH#	95	A20	128	D24
30	HINT#	63	RESET#	96	A21	129	D23
31	HALT#	64	V_{SS}	97	A22	130	D22
32	V_{SS}	65	V_{CC}	98	V_{CC}	131	V_{SS}
33	V_{CC}	66	V_{SS}	99	V_{SS}	132	V_{CC}

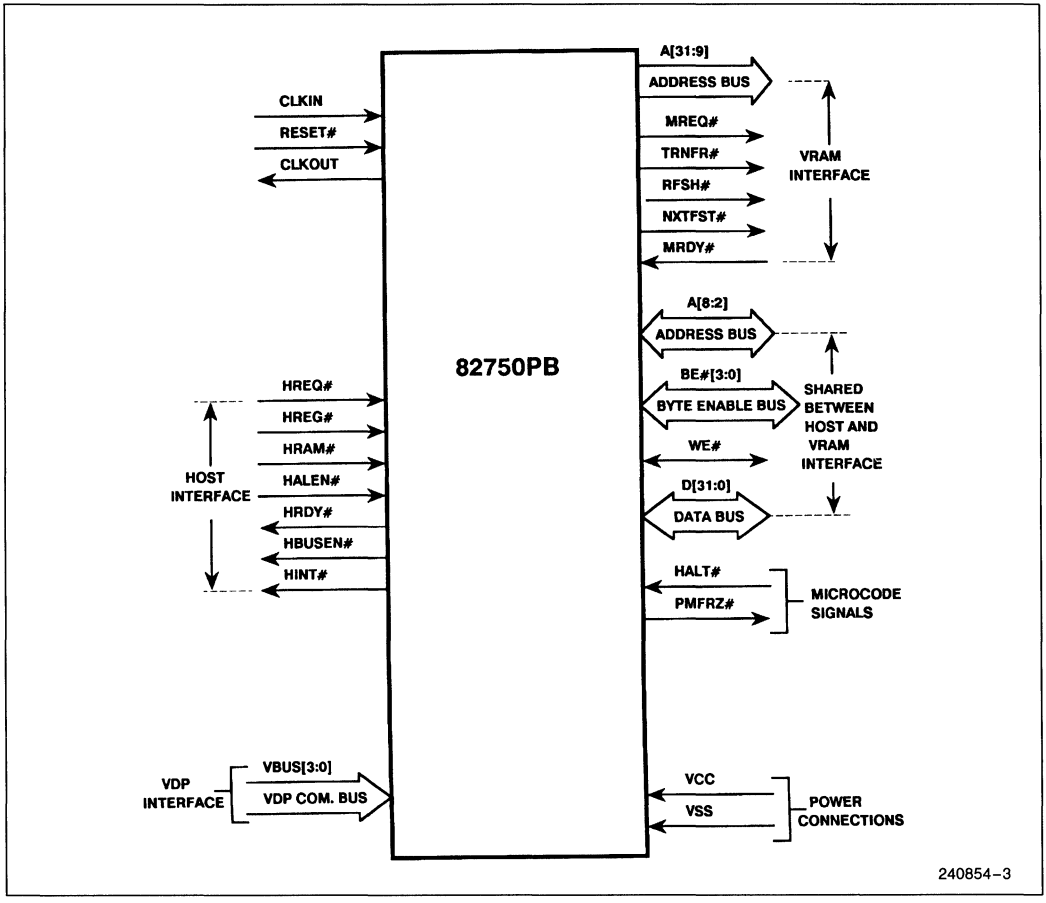


Figure 1-2. 82750PB Functional Signal Groupings

Quick Pin Reference

Table 1-3 provides descriptions of 82750PB pins.

Table 1-3. Pin Descriptions

Symbol	Type	Name and Function
CLKIN	I	CLKIN is a **1X CLOCK INPUT** that provides the fundamental timing for the 82750PB. One cycle of CLKIN is denoted as one T-cycle.
RESET#	I	The 82750PB is reset and initialized by holding this signal active for at least ten T-cycles. Refer to page 36.
HREQ#	I	The **HOST REQUEST** signal is a request from the host CPU to perform a read or write access to either registers on the 82750PB, an external device, or to VRAM shared by the 82750PB and the host. The type of access that is requested is determined by the host access definition signals: HREG#, HRAM#, and WE#.
HREG,# HRAM#	I	The **HOST REGISTER** and **HOST RAM** signals, when validated by HREQ#, are used to define three host access cycles. HRAM# active indicates the host is requesting a VRAM read or write cycle. HREG# active indicates that the host is requesting a 82750PB register read or register write cycle. When both signals are inactive, a host external cycle is requested.
HBUSEN#	O	**HOST BUS ENABLE** is asserted by the 82750PB at the start of a host access to indicate that the 82750PB Address and Data buses (A[31:2], BE#[3:0], and D[31:0]) have been tri-stated. This allows the host to drive the same buses either for accessing shared VRAM or the 82750PB internal registers.
HALEN#	I	The **HOST ADDRESS LATCH ENABLE** signal is used to indicate to the 82750PB that the host has asserted a valid address (A[31:2], BE#[3:0]) and write enable (WE#).
HRDY#	O	**HOST READY** is asserted by the 82750PB at the end of a host access to indicate that the access cycle is ready for data transfer. For a host write cycle, HRDY# indicates that the 82750PB is ready to accept data from the host. For a host VRAM write cycle, HRDY# indicates that the VRAM has latched the data from the host. For a host read cycle, HRDY# indicates that output data from the 82750PB or VRAM is ready to be latched by the host.
HINT#	O	**HOST INTERRUPT:** This output is asserted when an interrupt condition is detected by the 82750PB, and the enable bit in the PROCESSOR CONTROL register corresponding to that interrupt condition is set to a ONE. HINT# stays active until the host CPU reads the INTERRUPT STATUS register. If an interrupt condition that is enabled occurs during the same cycle that the INTERRUPT STATUS register is being read, HINT# remains active.
D[31:0]	I/O	The **DATA BUS** is used to transfer data between: 1. The 82750PB and VRAM, and 2. The Host CPU and internal 82750PB registers. During host VRAM accesses, this bus is tri-stated to allow the host to share the same VRAM data bus. During host accesses to internal 82750PB registers all 32 bits are used for data transfer.
A[31:9] A[8:2]	O I/O	The **ADDRESS BUS** is shared between the 82750PB and the host for addressing VRAM. This 30-pin bus addresses 32-bit double words in VRAM. Byte Enable signals are used to address individual bytes or words within a double word in VRAM. In addition, the address for host accesses to internal 82750PB registers are communicated to the 82750PB using the lower seven pins, A[8:2], and the BE# pins. During host access cycles to either VRAM or 82750PB internal registers, A[31:2] are tri-stated. For internal register accesses, as indicated by HREG# being low, the lower seven bits, A[8:2], are used as the host address input.
CLKOUT	O	The **CLOCK OUTPUT** signal is one of the two internal clocks and is synchronized with CLKIN. It is always driven and will have a 50% duty cycle.

Table 1-3. Pin Descriptions (Continued)

Symbol	Type	Name and Function
BE#[3:0]	I/O	The **BYTE ENABLE BUS** is shared by the 82750PB and the host for addressing VRAM down to the byte level. The correspondence between the four Byte Enable pins and the D[31:0] pins is: BE#[3]–D[31:24], BE#[2]–D[23:16], BE#[1]–D[15:8], and BE#[0]–D[7:0]. During VRAM read cycles, the 82750PB enables all four bytes. During write cycles the 82750PB only enables those bytes that are to be written. Bytes that are not enabled are not to be altered in VRAM. During host accesses to 82750PB on-chip registers, the BE#[0] pin is used as an input to select whether the even or odd word is being accessed; the double word address is provided by the host on the A[8:2] pins. BE#[0] = 0 indicates that data is transferred on D[15:0]. BE#[0] = 1 indicates that data is transferred on D[31:16].
MREQ#	O	**MEMORY REQUEST** is asserted for the first cycle, T1, of each VRAM cycle.
TRNFR#, RFSH#	O	The **MEMORY CYCLE DEFINITION SIGNALS:** Transfer, Refresh and Write Enable are asserted at the same time as MREQ#, but stay active for the entire VRAM cycle. TRNFR# active indicates a VRAM transfer cycle. RFSH# active indicates a VRAM refresh cycle. If neither TRNFR# nor RFSH# are active, a VRAM data read or write cycle is requested.
WE#	I/O	The **WRITE ENABLE** pin is used as an output during a 82750PB/VRAM cycle to drive the WE# signal, which defines the access as a VRAM read cycle (when inactive) or write cycle (when active). During Host/VRAM and Host External cycles, the 82750PB tri-states this pin to allow the host to drive the VRAM write enable signals directly. During Host/register cycles, this pin is used as an input for the Host Write Enable signal to determine whether the host is reading or writing the 82750PB register.
NXTFST#	O	The **NEXT FAST** signal indicates that the following vram cycle can be performed with a page-mode or bank-interleaved access. This signal is asserted during the first of a pair of VRAM cycles that is guaranteed to be within the same VRAM page and in opposite banks—a pair of accesses to two sequential double words in VRAM at addresses Even Address and Even Address + 1. In other words, A[2] is a zero for the first cycle and a one for the second cycle.
MRDY#	I	The **MEMORY READY** input indicates that the VRAM cycle has progressed to the point where it is ready to perform the data transfer. For a VRAM read cycle, the VRAM data can be latched by the transition of MRDY# to an active state. For a VRAM write cycle, MRDY# indicates that the data has been latched into the VRAMs.
VBUS[3:0]	I	The **VDP COMMUNICATION BUS** is used to communicate from the 82750DB to the 82750PB. Codes sent over this bus indicate interrupt requests, transfer requests, and status information. Since the 82750DB and 82750PB run asynchronously, the VBUS signals are sampled on the falling edge of CLKIN and compared with the previous sample. For a VBUS code to be detected by the 82750PB, it must be valid for two successive samples.
HALT#	I	The **HALT** signal causes the microcode processor on the 82750PB to halt prior to executing the next instruction. This signal does not halt the VRAM interface. The Halt signal will allow the design of a hardware emulator for the 82750PB based on an 82750PB chip.
TEST#	I	The **TEST** signal is used for test purposes only and must remain high for normal operation.

Table 1-3. Pin Descriptions (Continued)

Symbol	Type	Name and Function
PMFRZ#	O	The **PERFORMANCE MONITORING AND FREEZE** signal is toggled by specific microcode instructions and can be used to determine the time required to execute certain sections of microcode.
V_{CC}	I	**POWER** pins provide the +5V D.C. supply input.
V_{SS}	I	**GROUND** pins provide the 0V connection to which all inputs and outputs are referenced.

Table 1-4. Output Pins

Name	Active Level	When Floated
CLKOUT	High	Always Driven
A[31:9]	High	Reset*, Host Cycle
HBUSEN#	Low	Reset*
HRDY#	Low	Reset*
HINT#	Low	Reset*
MREQ#	Low	Reset*
TRNFR#, RFSH#	Low	Reset*
NXTFST#	Low	Reset*
PMFRZ#	Low	Reset*

*The reset state is caused by RESET# being active low.

Table 1-5. Input Pins

Name	Active Level	Synchronous/ Asynchronous
CLKIN	High	Synchronous
RESET#	Low	Asynchronous
HREQ#	Low	Asynchronous
HREG#	Low	Synchronous
HRAM#	Low	Synchronous
MRDY#	Low	Synchronous
VBUS[3:0]	High	Asynchronous
HALT#	Low	Synchronous
HALEN#	Low	Asynchronous

Table 1-6. Input/Output Pins

Name	Active Level	When Floated	Synch/Async
D[31:0]	High	Reset*, Host Cycle	Synchronous
A[8:2]	High	Reset*, Host Cycle	Synchronous
BE#[3:0]	Low	Reset*, Host Cycle	Synchronous
WE#	Low	Reset*, Host Cycle	Synchronous

*The reset state is caused by RESET# being active low.

All output pins are floated when RESET is active low.

2.0 ARCHITECTURE

Overview

The 82750PB includes a wide instruction word processor that comprises a number of processing, storage, and input/output elements. The wide instruction word architecture allows a number of these elements to operate in parallel. The 82750PB executesone instruction every internal clock cycle or T-cycle. The various elements are connected via two 16-bit buses, the A bus and B bus, as shown in Figure 2-1. During each instruction execution cycle, data can be transferred from a bus source to a bus destination element on both buses.

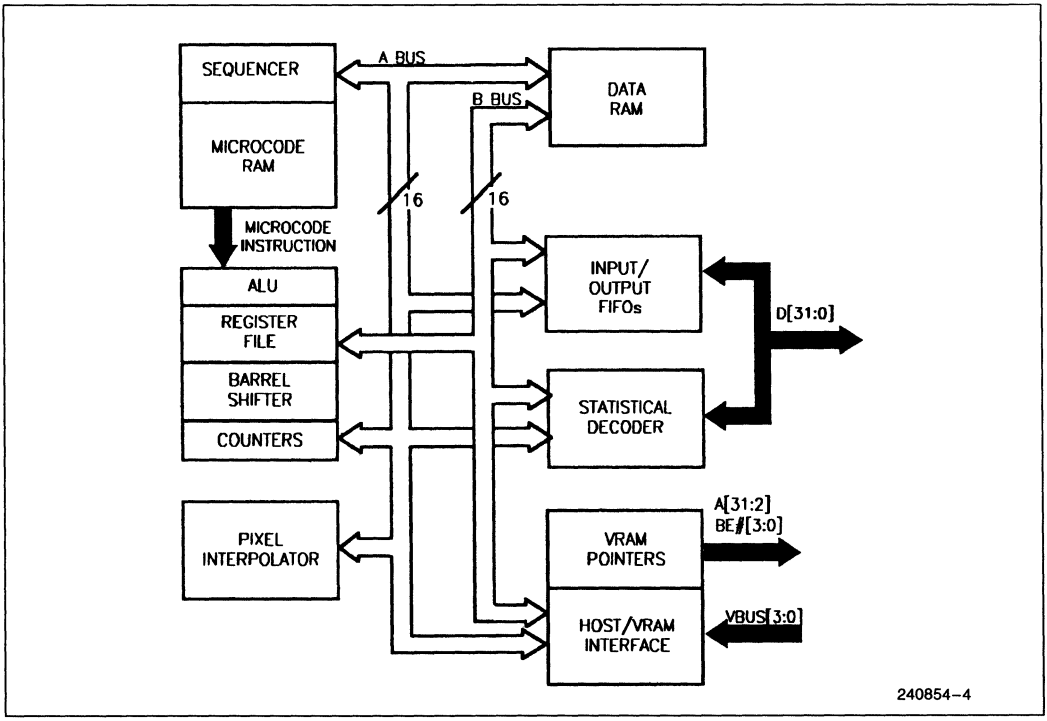


Figure 2-1. 82750PB Block Diagram

Registers

{*rN; N = 0–15*}

There are 16 general-purpose data registers, each 16 bits wide, that are connected to both the A bus and B bus as both sources and destinations. These registers are designated *r0–r15*. All the registers are functionally identical except *r0*, which also includes logic for bit shifting and byte swapping. A register can source both the A bus and the B bus in the same cycle. A register cannot be the destination of both the A bus and the B bus in a single instruction. Because the registers are doubly latched, the same register may be both a source and destination in the same cycle. The result is that the data in the register prior to the current cycle will be driven on the source bus, and the data on the destination bus will be latched into the register at the end of the cycle.

Register *r0* has additional logic to allow bit shifting and byte swapping. The value in *r0* can be shifted left or right one bit position per instruction cycle. For a right shift, the new MSB is equal to the old MSB; in other words, the value is sign-extended. For left shifting, the new LSB is equal to zero. *R0* can not be shifted and loaded in the same instruction. Byte swapping, on the other hand, only occurs when *r0* is being loaded with a value from the A bus or B bus. Byte swapping causes the most significant byte and the least significant byte of the 16-bit value being loaded into *r0* to be interchanged. Refer to Chapter 4 for a description of the SHFT microcode field that controls the shifting and swapping operations in *r0*.

ALU

{*alu, cc*}

The ALU performs 16-bit arithmetic and logic operations, and can also be operated as two independent 8-bit ALUs for the Dual-Add-with-Saturate operation. There are two fields in the microcode instruction that affect the operation of the ALU: the ALUOP field specifies the operation to be performed, and the ALUSS field specifies the source of the two ALU inputs. Refer to Chapter 4 for further information on these fields.

The two ALU operands either come from values held in the ALU input latches or from "eavesdropping" on the A or B buses. The result of any ALU operation is latched in the ALU output register, *alu*. In a following instruction this result can be transferred to any A or B destination.

The ALU has four condition flag outputs: CarryOut, Sign, Overflow, and Zero. CarryOut is the carry out of the most significant bit position. Sign is equal to the value of the most significant bit of the result. Overflow is the exclusive-OR of CarryOut and the CarryIn to the most significant bit position of the result. Zero is true (a value of 1) if all 16 bits of the ALU result are equal to zero. CarryOut and Overflow are defined as equal to Zero for all logical operations. For must ALU operations, the state of these four condition flags are latched when the operation is complete. There are six operations (NOP, A*, B*, ZERO, PMON, and INT) that are exceptions. These operations are performed without disturbing the condition state of the previous ALU operation.

Microcode routines can read and write the ALU condition flag register, *cc*. This can be used to save and restore the state of these flags. The bit ordering of the ALU condition flags within *cc* are given in Table 2-1.

Table 2-1. Bit Assignments for *cc* Register

Bit	Condition
Bit 0	False (This bit of the *cc* is always read as a zero.)*
Bit 1	ALU Carry Out
Bit 2	ALU Overflow
Bit 3	ALU Sign
Bit 4	ALU Zero
Bit 5	Loop Counter Zero*
Bit 6	R0 LSB*
Bit 7	R0 MSB*
Bit 15:8	RESERVED. The state of these bits is undefined when read; write as zeros.

*These are read-only values and are not affected by writes to the cc register.

The Dual-Add-with-Saturate operation performs independent 8-bit ADDs on the upper and lower bytes of the two ALU operands. The two bytes of the A operand are treated as unsigned binary numbers ($00{:}FF_{16}$ corresponds to $0{:}255_{10}$). The two bytes of the B operand are treated as offset binary numbers with an offset of +128 ($00{:}FF_{16}$ corresponds to $-128_{10}{:}127_{10}$). The upper and lower byte results are treated as 9-bit offset binary, including the carry output of each byte, with a +128 offset ($000{:}1FF_{16}$ corresponds to $-128_{10}{:}383_{10}$) and are saturated to a range of $0-255_{10}$. A result that is less than zero is set equal to zero or 00_{16} and a result that is greater than +255 is set equal to +255 or FF_{16}.

In fact, this operation is symmetric. Either the A operand or the B operand can be defined as the unsigned binary value, and the other operand will be treated as the offset signed binary value. The four ALU condition flags are latched after a Dual-Add-with-Saturate operation, but only the ALU ZERO flag has a defined value. It is set to ONE if the two 8-bit results are both zero, otherwise it is set to ZERO. The other three ALU flags are undefined after a Dual-Add-with-Saturate operation.

The ALU opcode INT generates the MCINT (microcode interrupt) condition. When this condition is detected by interrupt logic in the host CPU interface, and if the Enable MCINT bit in the PROCESSOR CONTROL register is set to a ONE, the host interrupt output, HINT#, will be asserted. Refer to Chapter 3 for further information.

Barrel Shifter

{*shift, shift-r, shift-rl, shift-l*}

The barrel shifter performs a single cycle, n-bit left or right shift. The barrel shifter operates independent of the ALU. The three barrel shifter operations are: *Shift-r* for a right shift with sign extend; *Shift-rl* for right shift with zero fill; and *Shift-l* for a left shift with zero fill. The shift operation is invoked by writing a 4-bit value (the shift amount) to one of three A bus registers, depending on which of the three operations is to be performed. The operand is taken from the B bus, and the result is stored in the barrel shifter output register, *Shift*. Like the ALU result register, the value in *Shift* can be read onto the A bus or B bus in the following instruction cycle.

A barrel shifter operation does not affect any of the condition flags.

Data RAM

{*dramN, *dramN, ++, --; N = 1-4*}

The Data RAM holds 512, 16-bit words that are accessed using four pointers. To access a value in a particular location, the microcode routine must first load a pointer with the address to be accessed, and then perform a read or write using the same pointer. In parallel with the data RAM access, the pointer can optionally be post-incremented or post-decremented. The four pointers, referred to as *dram1–dram4*, can be written and read via the A bus. When a dram pointer, which is only 9 bits wide, is read onto the A bus, its upper seven bits are set to zeros.

CAUTION:
The width of the dram pointers may change in later versions of the 82750PB. Software should not rely on the width of a pointer to, for example, mask the upper seven bits of a value to zero.

All four pointers can be used to read or write the Data RAM from either the A or B bus. Only one Data RAM access can be performed in any cycle. A Data RAM access is referred to, using C language syntax, as **dram1*. The * means "the value pointed to by". As another example, **dram3++* means access the Data RAM using the pointer *dram3* and increment *dram3*. The symbol -- in place of the ++ would indicate autodecrement.

Loop Counters

{*cnt,cnt2*}

Two 16-bit loop counters are available to microcode programs for automatically counting iterations of a microcode loop. In parallel with other operations performed in an instruction, either loop counter can be decremented, and a conditional branch can be made based on the loop counter value being equal or not equal to zero. Since the two loop counters can be written and read on the A bus, as *cnt* and *cnt2* respectively, they can also be used for variable storage when not being used as loop counters. The loop counters can be written to and decremented during the same instruction cycle. The value in the counter at the start of the next cycle will be the value written to the counter minus one.

The LC microcode bit determines the loop counter that is selected for decrementing and/or branching in an instruction. The LC microcode bit does not affect the loop counter that is written or read over the A bus, since each loop counter is separately addressable as a A bus source or destination. Refer to Chapter 4 for a description of the CNT -- microcode bit that causes the select loop counter to be decremented, and for a description of the CFSEL microcode field that is used to perform a conditional branch based on the selected loop counter's value.

Microcode RAM

{*mcode1-3, maddr, pc*}

The 82750PB executes instructions stored in an on-chip microcode RAM. This RAM holds 512 instructions and each instruction is 48 bits wide. Normally, to start the microcode processor, the host CPU will load a microcode program into the microcode RAM, point the program counter, *pc*, to the start of the program, and then release the HALT bit to start executing the microcode program. The microcode processor can also load its own microcode RAM to overlay new routines and therefore, does not require constant intervention by the host to perform multiple operations.

Writing an instruction into Microcode RAM is done by first loading the three registers *mcode3, mcode2*, and *mcode1* with the three 16-bit words of the instruction (the most significant word goes into *mcode1*), and then loading the address where the instruction should be written into *maddr*.

NOTE:
In the 82750PA mode, only the lower eight bits of *maddr* are used. The upper eight bits are forced to zeros in order to accurately emulate the 82750PA.

The host CPU can also read the Microcode RAM by first loading the *pc* with the address of the instruction to be read and then reading the three 16-bit words of the instruction from the *mcode1–mcode3* registers. Normally, this would be done by the Host CPU while the 82750PB is halted. Since *mcode1–mcode3* hold the instruction pointed to by the *pc* (i.e. the instruction that is about to be executed), normally reading these three registers from a microcode routine is not useful.

CAUTION:
The read registers named mcode1–mcode3 and the write registers also named mcode1–mcode3 are in fact different registers. Writing values into mcode1–mcode3 and then reading the values of mcode1–mcode3 will not read back the same values just written. The read registers hold the instruction stored in the instruction latch (the instruction to be executed). The write registers hold an instruction that is about to be written into microcode RAM.

CAUTION:
In the 82750PA mode, the bit positions of the microcode fields are scrambled. As a result, data written to microcode RAM via the mcode1–mcode3 registers in the 82750PA mode will not be read as the same values from the mcode1–mcode3 registers. In the 82750PB mode, data written to the microcode RAM will be read back without any bit scrambling. Therefore, testing of the microcode RAM is less difficult in the 82750PB mode.

After writing to *maddr* to load an instruction into microcode RAM, a one cycle freeze occurs and during the freeze a write to the microcode RAM takes place. The instruction following the write to *maddr* can either jump to the address just loaded or start loading the *mcode1–mcode3* registers with the next instruction to be written.

CAUTION:
The 82750PB requires at least one instruction between the write to maddr and the execution of the instruction that is loaded by the write to maddr.

Here are two examples:

Example 1:

```
maddr = ADDR1          /* load instruction */
jmp addr1              /* jump to it, this is the extra inst. required between */
                       /* writing to maddr and executing the loaded inst. */
. . .
ADDR1:
???????????            /* here's where new instruction gets loaded */
```

Example 2:

```
maddr = INST
nop                    /* extra instruction */
INST:
???????????            /* instruction gets loaded here */
```

When a microcode routine writes to *pc*, one more instruction is executed before the jump to the new address takes effect. For example:

```
pc = ADDR1
r0 = r1   jmp ADDR2    /* this instruction gets executed but */
                       /* its jump to ADDR2 is ignored. */
. . .
ADDR1:
r3 = r0                /* after this instruction executes r3 = r0 = r1 */
```

When the host CPU writes to the *pc*, the instruction at the address that was written is loaded into the *mcode1–mcode3* registers and, when the microcode processor is released from its Halt condition, this is the first instruction that will be executed.

When the host CPU reads the *pc*, the result returned is the address of the instruction that will be executed when Halt is released, that is, the address of the instruction held in the *mcode1–mcode3* registers.

Horizontal Line Counter

{*lcnt*}

The Horizontal Line Counter is updated by VBUS codes from the 82750DB to track the horizontal display line that is currently being scanned by the 82750DB. The counter is reset by a VODD code and incremented each time an HLINE code is received. A value can also be written into a Horizontal Line Counter but this is used primarily for testing the 82750PB.

Input FIFOs

{*inN-lo, inN-hi, inN-c, *inN; N = 1, 2*}

There are two input channels, referred to as input FIFOs, through which the processor can read pixels or data from VRAM. Each channel automatically fetches 64-bit quad words from VRAM and breaks them into 8-bit bytes or 16-bit words that are read by microcode. Each FIFO operates independently and can be programmed to automatically increment or decrement through bytes or words in VRAM. The FIFOs are double buffered so that while values are being extracted from one quad word (64 bits), the next quad word is being prefetched from VRAM.

The mode control register for each input FIFO, designated *in1-c* or *in2-c*, contains four mode bits as seen in Figure 2-2. The WORD/BYTE bit (bit 0) determines whether the input FIFO is in word mode (WORD/BYTE = 0) or byte mode (WORD/BYTE = 1). In byte mode, the FIFO can start reading on any byte boundary and in word mode on any word boundary.

The INC/DEC bit (bit 1) determines the order that bytes or words are read from VRAM. In INCREMENT mode, with INC/DEC = 0, the FIFO reads from the least significant byte or word to the most significant byte or word of each double word and increments through double words in VRAM. In DECREMENT mode, with INC/DEC = 1, the FIFO reads from most significant byte or word to least significant byte or word within a double word and decrements through double words in VRAM.

The PREFETCH OFF bit (bit 2) specifies whether the FIFO will automatically prefetch successive quad words from VRAM or will only fetch a new quad word when a value from that quad word is requested. In PREFETCH-ON mode, bit 2 = 0, the input FIFO prefetches successive quad words from VRAM as necessary to keep its buffer full (either from ascending or descending addresses, depending on the state of the INC/DEC bit). In PREFETCH-OFF mode, the FIFO will still prefetch the first two quad words to fill its buffer (when started at a new address location), but will only fetch a new quad word when a read request is made to the FIFO for a value in the next unfetched quad word.

In "BY-32" MODE (bit 3), the pointer increments or decrements by 32 bits, independent of whether the FIFO is in 8-bit pixel mode or 16-bit pixel mode. This mode was added to facilitate microcode that operates on one component of a 32-bit per pixel image.

The standard sequence for initializing an input FIFO is to write to the control register *(in-c)*, the high address *(in-hi)*, and then the low address *(in-lo)* of the appropriate FIFO. Refer to the access state diagram in Chapter 3. The write to *in-lo* causes the FIFO to start reading from VRAM. A byte or word is then read from **in*. Successive reads from **in* will read sequential bytes or words from VRAM. Writing to the control register each time the FIFO is started at a new address is not necessary, except to change the FIFO's mode. Also, if the new address is within the same 64 kByte page of VRAM, only the lo-address needs to be written in order to start the FIFO reading from the new address.

If microcode attempts to read a value from an empty input FIFO, the processor is frozen prior to the execution of the instruction, until the FIFO's control logic has fetched another double word from VRAM and extracted the next value. At this point, the processor is released from the frozen state, and the instruction that reads the value is executed. When the processor is frozen waiting for a particular FIFO that isn't yet ready, that FIFO's VRAM access priority is raised above all other FIFOs.

bits:	15 . . . 4	3	2	1	0
	Set to Zeros	BY-32 MODE	PF OFF	INC/DEC	WORD/BYTE

Figure 2-2. Input FIFO Control Register

Output FIFOs

{*outN-lo, outN-hi, outN-c, *outN, outN++; N = 1, 2*}

There are two output channels, referred to as output FIFOs, through which the graphics processor writes pixels or data to VRAM. Each channel automatically collects bytes or words into 64-bit quad words and writes the quad words to VRAM. Each output FIFO operates independently and can be programmed to write bytes or words into sequential addresses in VRAM (either incrementing or decrementing). The FIFOs are double buffered so that while one quad word is waiting to be written to VRAM, the next quad word can be assembled from individual bytes or words.

The mode control register for each output FIFO, designated *out1-c* or *out2-c*, contains six mode bits as shown in the Figure 2-3. The WORD/BYTE bit (bit 0) determines whether the output FIFO is in word mode (WORD/BYTE = 0) or byte mode (WORD/BYTE = 1). In byte mode the FIFO can start writing on any byte boundary in VRAM and in word mode on any word boundary.

The INC/DEC bit (bit 1) determines the order that bytes or words are written to VRAM. In INCREMENT mode, with INC/DEC = 0, the FIFO writes from the least significant byte or word to the most significant byte or word in a double word and increments through double words in VRAM. In DECREMENT mode, with INC/DEC = 1, the FIFO writes from most significant byte or word to least significant byte or word within a double word and decrements through double words in VRAM.

When the DISABLE INC/DEC bit (bit 2) is set, the output FIFO quad word address is not incremented or decremented. In this mode, the FIFO continues to output to a single quad word in VRAM.

The FORCE-LSB bits (bits 3 and 4) are used to force the least significant bit of each byte written to VRAM to either a zero or a one. This can be used, for example, to force the LSB to the correct polarity when writing to the U bitmap during motion video decompression. In certain display modes for the 82750DB, the LSB of the 8-bit samples in the U or Y bitmap are used to select VIDEO or GRAPHICS display mode for the n x n group of display pixels corresponding to the particular U or Y sample. A one in the FORCE-LSB ENABLE bit (bit 4) enables the forcing; a zero results in normal operation. The FORCE-LSB VALUE bit (bit 3) is used as the value to which the LSB is forced. Whether in byte mode or word mode, the LSB of *each byte* is forced to the FORCE-LSB value.

In "BY-32" MODE (bit 5), the pointer increments or decrements by 32 bits, independent of whether the FIFO is in 8-bit pixel mode or 16-bit pixel mode. This mode is used to facilitate microcode that operates on one component of a 32-bit per pixel image. The bytes or words that are skipped over will be unchanged in VRAM.

The standard sequence for initializing an output FIFO is to write to the control register *(out-c)*, the low address *(out-lo)*, and then the high address *(out-hi)* of the appropriate FIFO. A series of bytes or words is then written to **out*. Refer to the access state diagram in Chapter 3.

In order to flush any remaining data in an output FIFO before changing its VRAM pointer, it is necessary to write to the control register. When pointing to a new location in VRAM, if the new address is within the same 64 kByte page of VRAM, only the lo-address needs to be written.

bits:	15–6	5	4	3	2	1	0
	Set to Zeros	BY-32 MODE	FORCE-LSB ENABLE	FORCE-LSB VALUE	DISABLE INC/DEC	INC/DEC	WORD/BYTE

Figure 2-3. Output FIFO Control Register

CAUTION:
There must be one instruction between the write to the output FIFOs low address and the first write to *outN. Therefore, it is recommended that outN-lo be written before outN-hi. The write to outN-hi insures that this requirement is met. If only the outN-lo value is being changed, it is still necessary to have one additional instruction before the first write to *outN.

When writing bytes or words to VRAM through an output FIFO, a byte or word can be skipped over by writing to *outN++* instead of **outN*. When the values are written to VRAM, any byte or word that was skipped will retain its original value in VRAM, and its value is not altered by the VRAM write. This can be used when writing a series of pixels, some of which are "transparent", allowing whatever was behind them to show through.

If the microcode routine attempts to write a value to a full output FIFO, the processor is frozen prior to the execution of the instruction. The processor remains frozen until the FIFO has a chance to write one of the buffered quad words to VRAM. At that point, the processor is released from the frozen state, and the instruction that writes the value is executed. When the processor is frozen, waiting for a particular FIFO that isn't yet ready, that FIFO's VRAM access priority is raised above all other FIFOs.

Statistical Decoder

{*stat-lo, stat-hi, stat-c, stat-ram, *stat, *stat#*}

The Statistical Decoder (also referred to as the Huffman Decoder) is a specialized input channel that can read a variable-length bit sequence from VRAM and convert it into a fixed-length bit sequence that is read by the microcode processor. In image compression, as well as in other applications such as text compression, certain values occur more frequently than others. A means of compressing this data is to use fewer bits to encode more frequently occurring values and more bits to encode less frequently occurring values. This type of encoding results in a variable-length sequence in which the length of a symbol (the group of bits used to encode a single value) can range for example, from one bit to sixteen bits.

The statistical code that the statistical decoder can decode is of either of the two forms:

0x		1x
10x		01x
110xxx		001xxx
1110xxxxx		0001xxxxx
...	or	...
11111110xxxxxx		00000001xxxxxx
111111110xxxxxx		000000001xxxxxx
...		...

Each symbol of a given length (one per line as shown here) consists of a run-in sequence followed by some number of x-bits. The run-in sequence is defined as a series of zero or more ONEs followed by a ZERO or, as in the code on the right above, zero or more ZEROs followed by a ONE. The remainder of this description will use examples of the code on the left. A bit in the decoder's control register determines the polarity of the run-in sequence bits.

In the example on the left, there would be two symbols of length two: 00 and 01. Each x-bit can take on a ZERO or ONE value. The number of x-bits following a run-in sequence can range from zero to six. Since the goal, in general, is to have a few short codes and a larger number of long codes, typically, codes with fewer run-in bits will have fewer x's following. However, this is not a hardware constraint. A code of this form is completely described by a code description table indicating: for each length of run-in sequence, R = the number of ONEs in the run-in, and how many x-bits follow the ZERO. The value of R is used as an index into the code description table. Due to the hardware implementation, the number actually stored in the table is 2^x, where x is the number of x-bits.

For the example above, the corresponding code description values are given in Table 2-2.

Table 2-2. Sample Code Description Table

R	X	2^x(dec.)	2^x(bin.)
0	1	2	000 0010
1	1	2	000 0010
2	3	8	000 1000
4	5	32	010 0000
...			
7	6	64	100 0000

Note that the table only goes up to symbols with seven ONEs in the run-in. For symbols with more than seven ONEs, the value of X and 2^x for seven ONEs is used for all symbols having seven or more ONEs in the run-in sequence. For example, in the code above a symbol with eight or more ONEs in the run-in sequence has six x-bits following the ZERO, which is the same as symbols having seven ONEs.

For each different symbol, including all symbols of the same run-in length with different x-bit values, the decoder generates a unique fixed-length, 16-bit value. Some of the decoded values for the sample code given above are provided in Table 2-3.

Table 2-3. Decoded Values

Symbol*	Decoded Value
0**0**	0
0**1**	1
10**0**	2
10**1**	3
110**000**	4
110**001**	5
110**010**	6
. . .	. . .
110**111**	11
111**00000**	12
. . .	. . .
111**11111**	43
. . .	. . .

*The x-bits of the symbol are in **boldface** for clarity

The algorithm for generating a decoded value from a symbol is as follows: all symbols of a given run-in length are assigned a base value, B; the value corresponding to a particular symbol is equal to B plus the binary value of the x-bits in the symbol. The base valule B for a symbol with a run-in length of R is calculated by:

$$B(R) = SUM[2^{X(r)}] \text{ with } r = 0 \text{ to } R - 1,$$

where X(r) corresponds to the X value in the table entry corresponding to R = r.

For example, in the above code:
B(0) = 0, B (0) is always zero
B(1) = 0 + 2 = 2
B(2) = 0 + 2 + 2 = 4
B(3) = 0 + 2 + 2 + 8 = 12
B(4) = 0 + 2 + 2 + 8 + 32 = 44

This is one of the reasons that the table holds 2^X instead of X. The calculation of B(R) are easier to implement in logic.

There are two enhancements that are made to this coding scheme in the implementation on the 82750PB. These two modes are referred to as END mode and SHORT mode. If neither END nor SHORT mode are enabled, the decoding is performed as described above. SHORT mode allows the decoder to be switched easily to a simpler code format without having to reload the code description table. In the SHORT form, all symbols have the same number of x-bits, as though all entries in the table had been filled with the same value of 2^X. When SHORT mode is invoked, this value of 2^X is obtained from a field in the statistical decoder's CONTROL word, instead of from the individual table entries.

END mode is added in recognition of the fact that, for codes with few symbols, some increase in efficiency is possible by not having to place a zero at the end of the longest run-in sequence. For example, consider the code:

0
10x
110x

The END mode allows us to shorten the last symbol to 11x instead of 110x. The trailing ZERO is not required because the decoder has been told that the maximum length of a run-in is two ONEs. The resulting symbol set and corresponding decoded values are given in Table 2-4.

Table 2-4. END Mode Decoded Values

Symbol	Decoded Value
0	0
100	1
101	2
110	3
111	4

The number of x-bits must be constant for all symbols of the same run-in length. Therefore, a code such as:

0
10xx
11xxx ← NOT CORRECT! . . . Must be 11xx.

is not allowed. The last symbol (11xxx, in this case) uses the same table entry for 2^X as the next to last symbol (10xx) and, therefore, the last symbol will be 11xx.

The maximum length of the run-in sequence in END mode is specified by placing an END flag in the code description table. For example, a code and the corresponding table is shown in Table 2-5.

Table 2-5. END Flag Decoded Values

Code	Table Entries		
	Index	END Bit	2^X
0	0	0	0
10xx	1	0	4
110xxx	2	1	8
111xxx	3	-	-
	4	-	-
	5	-	-
	6	-	-
	7	-	-

The hyphens indicate that those table entries aren't used to decode this code. Note that the symbol 111xxx has three x-bits because of the value of 2^X in Index 2; it is not based on the 2^X value in Index 3.

The SHORTED and END modes can be invoked simultaneously, resulting in a code such as:

0x
10x
110x
111x

with a SHORT -2^X value = 2 (for 1 x-bit in each symbol) and the END bit set in Index 2.

Packed binary fields with one to seven bits per field can be read using the statistical decoder by setting the END bit in Index 0 and by programming the X value to be N − 1, where N is the number of bits per field. For example, packed three-bit fields could be decoded as shown in Table 2-6.

Table 2-6. Packed 3-Bit Field Decoded Values

Code	Table Entries		
	Index	END Bit	2^X
0xx	0	1	4{N = 3, so X = 2}
1xx	1	-	-
	2	-	-
	3	-	-
	4	-	-
	5	-	-
	6	-	-
	7	-	-

NOTE:
The unpacked bits are in reverse order relative to how they are stored in VRAM. For example, if three-bit values are packed in VRAM, the pattern 110 in VRAM is read from right to left and gives an unpacked or decoded value of 3.

The CONTROL register for the statistical decoder *(stat-c)* is used to specify the mode to use for decoding, as well as to invoke certain modes for writing and reading the code description table. Refer to the bit assignments for this register below. To write to the code description table, the WRITE bit (bit 4) is set to a ONE; the starting table index is reset to zero. Each write to the table causes the index to increment by one. This index will wrap around from seven back to zero. For example, to write all eight table entries the user would write a value of 0x10 to *stat-c* register and then write eight 8-bit values to the register *stat-ram*. The most significant bit of each 8-bit value is the END bit, and the lower seven bits are the values of 2^X. To read the code description table, the TEST bit (bit 4) of the CONTROL register is set to a one. The table entries are then read from the decoder's data register *(*stat)*. Reads and writes always start at table entry zero.

CAUTION:
When reading the table, it is necessary to wait one instruction time before each read, that is, between the write to stat-c and the first read from *stat and between each read from *stat. (An access diagram showing all legal sequences for read and write FIFO registers is shown in Chapter 3.)

The code for reading the eight table entries into the first eight locations of data RAM would be:

```
dram3 = 0          stat-c = 0x20          /test mode to read the stat-ram (the table)
cnt = 8                                   /wait one inst. before first read
LOOP:
                   *dram3++ = *hd cnt--
                   jcp loop               /two inst. loop necessary to wait one inst.
                                          /between each read from *hd.
```

	POL	Definition
	0	ONEs ending in ZERO (e.g., 1110xxx)
	1	ZEROs ending in ONE (e.g., 0001xxx)

Bits	15	14:13	12:8	7	6	5	4	3:0
	POL	ND*	SVAL	SHORT	END	TEST	WRITE	RAM ADDRESS

* Not Defined; write zeros to these bits.

Figure 2-4. Statistical Decode CONTROL Register

END mode is enabled by setting the END bit (bit 6) in the CONTROL register to a ONE. The SHORT mode is enabled by setting the SHORT bit (bit 7) in the CONTROL register to a ONE. When in SHORT mode, the five SVAL bits (bits 12:8) in the CONTROL register are used as the SHORT -2^X value.

The POL bit, bit 15, determines the polarity of the run-in sequence bits, as shown in Figure 2-4.

The decoding parameters may be changed between symbols by writing to the CONTROL register and, if necessary, writing new values into the code description table. The correct procedure for changing the code type or decode mode is to read the last value from the decoder prior to the change, using **stat#* instead of **stat*. This keeps the decoder from automatically starting to decode the next symbol. At this point, the code description table and the SHORT and END mode bits can be changed as desired. The next time the CONTROL register is written with both TEST = 0 and WRITE = 0, the decoder will begin to decode the next symbol using the new parameters.

The statistical decoder buffers one quad word read from VRAM so that the decoding of bits in one 32-bit word and the fetch of the next 32-bit word may overlap. As with the input and output FIFOs, the decoder has a VRAM pointer associated with it that points to the location in VRAM from which it is reading data. This pointer increments twice each time a new quad word is read; there is no decrement mode. When the least significant word of the decoder's pointer *(statlo)* is written, any data that had previously been pre-fetched from VRAM is ignored, and the decoder fetches one quad word starting from this new location.

NOTE:
The 82750PB assumes that the statistically encoded bitstream in VRAM starts with the least significant bit of a *double* word. That is, the two LSBs of the address written to start-lo are ignored.

The statistical decoder decodes data at a rate of one bit per T-cycle. To a first approximation, the decode time for an N-bit symbol is:

decode time (in T-cycles) = N + 1

Since it takes at least 64 T-cycles to decode data from one quad word, which is the time required for eight quad word reads from VRAM, the decoder should rarely run out of data. Therefore, the above estimate should very accurately model the actual decoding rate of the statistical decoder.

The statistical decoder always begins to read the bitstream from the least significant bit of the double word found at the starting location in VRAM. That is, the decoder does not start on a byte or word boundary as an input FIFO or output FIFO does, but only on double word boundaries. The bitstream moves from the least significant bit to the most significant bit of a double word and then to the least significant bit of the next double word (at the next higher ad-

dress location). For the x-bits, the first x-bit read from the bitstream becomes the most significant bit of the x-bit field when it is interpreted as a binary number. The example below shows a code definition, a bitstream stored in VRAM, and the resulting decoded values.

The code definition and range of values for each symbol length are indicated in Table 2-7.

Table 2-7. VRAM Bitstream Decode Values

Symbol	Values	Comments
0	0	
10x	1, 2	100 = 1, 101 = 2
110xx	3–6	11000 = 3, . . . , 11011 = 6
1110xxx	7–14	1110000 = 7, . . . , 1110111 = 14

Decoding starts at address 0 in this example. The two double words at addresses 0 and 1 are:

0: 0xAC98E14D

1: 0x372E74CB

The bitstream in VRAM, with colons dividing the symbols (read from right to left starting at LSB of address 0) is shown in Figure 2-5.

Table 2-8 lists the symbols, in the order they are encountered in the bitstream, and the corresponding decoded values.

Table 2-8. Decoding Symbols

Symbol	Value	Comments
101	2	Starts at LSB, Address 0, Scanning Left
100	1	
101	2	
0	0	
0	0	
0	0	
0	0	
1110001	8	
100	1	
100	1	
11010	5	
1110100	11	Spans First and Second Double Word
11001	4	
0	0	
1110011	10	
101	2	
0	0	
0	0	
1110110	13	
. . .	. . .	

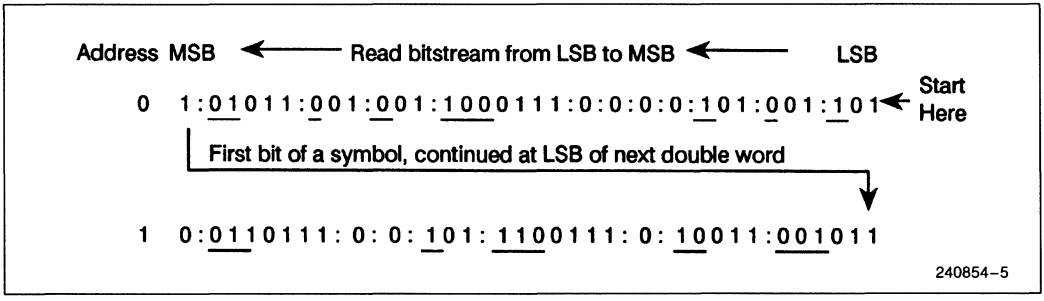


Figure 2-5. VRAM Bitstream Decoding Addresses

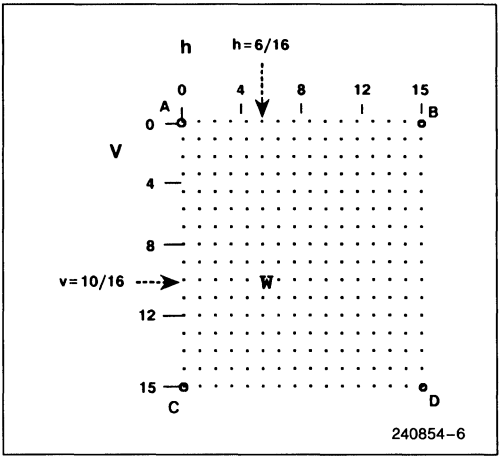


Figure 2-6. Pixel Interpolation

Pixel Interpolator

{*Pixint-c, Pixint*}

The pixel interpolator performs bilinear interpolation on four 8-bit pixels to generate, in effect, a pixel shifted by a fraction of a pixel position. See the diagram below. If the four pixels have values of A, B, C, and D; and the horizontal weight and vertical weight are h and v, respectively, the interpolated value W, ignoring any quantization effects, is given by:

$$W = A*(1-h)(1-v)+B*h(1-v)+C*(1-h)v+D*hv$$

The values of h and v are even multiples of 1/16. Figure 2-6 illustrates pixel interpolation with an h weight of 6/16 or 3/8 and a v weight of 10/16 or 5/8.

The pixel interpolar can operate in two modes: sequential-2D and random-2D. Sequential-2D mode is used for motion video decoding and when an array of pixels are interpolated with a common weighting. Random-2D mode is used either when the pixel arrays to be interpolated are not adjacent pixels in two rows or when the weight is changed for each interpolation. (The word random is used here to mean non-sequential.)

The example in Figure 2-7 shows a single row of pixels being interpolated in Sequential-2D mode using two rows from the original (source) bitmap. The h and v weighting are constant for all three interpolated pixels. In this case, the weights appear to be approximately h = 10/16 and v = 6/16.

A B E F I ...	—First Input Row
W X Y Z ...	—Interpolated Row
C D G H K ...	—Second Input Row

Figure 2-7. Sequential-2D Pixel Interpolation

The pixel interpolator is pipelined and requires some startup sequence to fill the pipeline. The pipelining described for the Sequential-2D and Random-2D modes are for compatibility with the 82750PA. A bit in the Control register may be used to select a mode that has less pipelining but that is incompatible with the 82750PA. The pipeline delays associated with this fast mode are listed below.

Once filled, the pixel interpolator generates a new interpolated pixel every two T-cycles when in Sequential-2D mode. Source pixels are written into the interpolator as pixel pairs. In the case above, the pixel pair BA would be written first, followed by the pixel pair DC. It would seem more natural to refer to the pixel pair as AB, but because of the way 8-bit pixels are arranged in 16-bit words in VRAM, the leftmost pixel on the screen is the least significant byte position. For example, if pixel A had a hex value of 0xAA and B had a value of 0xBB, the 16-bit word containing pixels A and B would have a value of 0xBBAA.

Then, two pixels are read from the interpolator. Because the pipeline isn't full yet, these pixels are read and discarded. This loop of writing two pixel pairs and reading two output pixels continues four times. The two pixels that are read this fourth time are the first two valid output pixels: W and X. The interpolator may also collect output (interpolated) pixels into pixel pairs. For exmple, pixels W and X, instead of being output separately, would be combined into a 16-bit pixel pair XW. Since there are two possible phase relationships between the input pixel pairs and output pixel pairs, the desired phasing (either X and W paired or Y and X paired) can be specified.

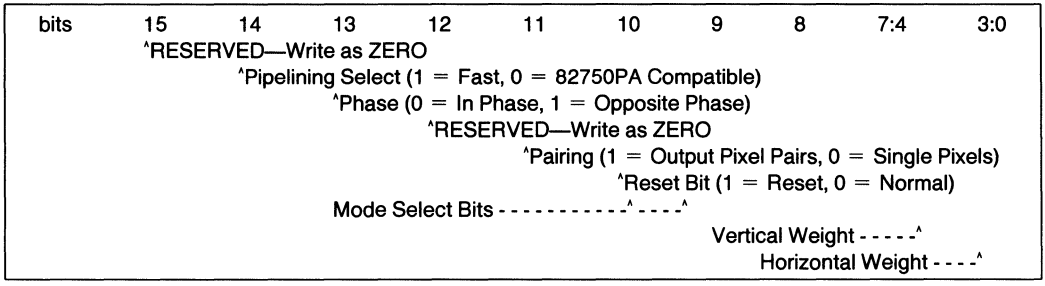


Figure 2-8. Pixel Interpolator Control Register

Random-2D interpolation is used either when the pixels to be interpolated are not in horizontal rows or when the weight is changed for each interpolated pixel. Examples for this are smooth warping or smooth scaling operations. In the case of Random-2D, the processing for successive interpolated pixels can not take advantage of pipelining; each pixel is considered to be the first pixel of a Sequential mode interpolation. The weight and the two input pixel-pairs are written into the interpolator. After waiting at least 10 T-cycles, the one interpolated pixel can be read. (The delay is 10 cycles when in 82750PA emulation mode (bit 14 = 0) and 6 T-cycles when in the 82750PB mode (bit 14 = 1).) Then, the next two input pixel-pairs and if necessary, the new weight value, are written, and 10 cycles later the next interpolated pixel can be read.

The h and v weight values, the mode selection, and other control bits are written to the pixel interpolator control register *(avg-c)*. The bit assignment for this register is in Figure 2-8. The least significant byte holds the 4-bit v value (bits 7:4) and the 4-bit h value (bits 3:0).

NOTE:
The values used for h and v here are numerators of the fraction where the implied denominator is 16. The effects of the other control bits are listed in Table 2-9.

MODE SELECT

Bits 8 and 9 are used to select on of four operating modes, of which only two are presently defined. These modes are given in Table 2-9.

Table 2-9. Mode Select Operating Modes

Bits 9:8	Mode
00	RANDOM-2D
01	Sequential-2D
10	RESERVED
11	RESERVED

RESET

Writing a ONE to bit 10 resets the pixel interpolator. The pixel interpolator must be reset prior to changing modes.

PAIRING

A ZERO in bit 11 causes the pixel interpolator to output individual pixels. A ONE causes the interpolator to collect adjacent pixels (in Sequential-2D mode) into 16-bit pixel pairs. This feature assists in motion video decoding, when combined with the ALU's dual-add-with-saturate operation, by allowing two pixels to be processed each cycle. The phasing used in collecting the pixel pairs is determined by the Phase bit described below.

PHASE

When output pixels are collected into pixel pairs, there are two possible alignments of the input pixel pairs to the output pixel pairs. The Phase bit (bit 13) selects the alignment to be used, based on the relative word alignment of the source and destination bitmaps in VRAM. When the Phase bit is set to a ZERO, this indicates that the bitmaps are in-phase. In this case, the first two output pixels are grouped into one 16-bit pixel pair (with the first pixel in the least significant byte). When the Phase bit is set to a ONE, the bitmaps are out-of-phase. In this case, the first pixel is placed in the most significant byte of the first pixel pair, with invalid data in the least significant byte, and the second and third output pixels are collected into the second pixel pair. This is illustrated in Figure 2-9.

PIPELINING

A ZERO in bit 14 causes the pixel interpolator to use an amount of pipeline delay that is compatible with the 82750PA. A ONE in this field will select a mode that has less pipeline delay, but that is not compatible with the 82750PA. Table 2-10 shows the pipelining delay for the 82750PA compatible mode and the

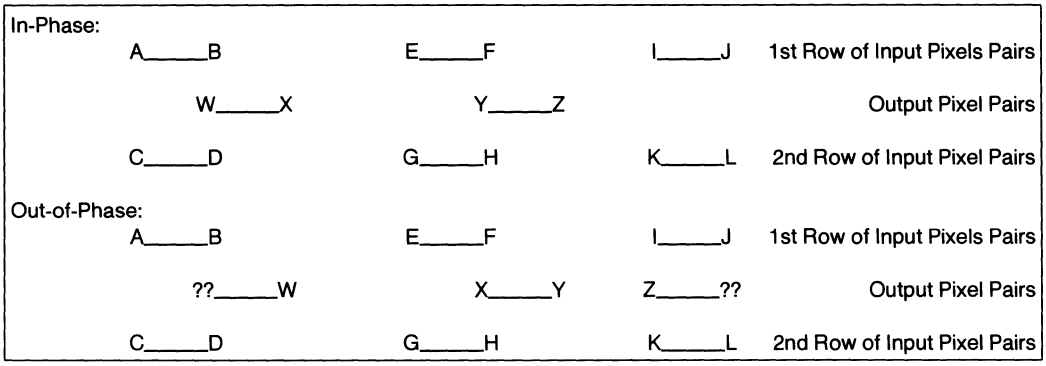


Figure 2-9. Pixel Pair Phases

faster 82750PB mode. Note that the effect of the pairing bit is to add an additional one pixel delay.

Table 2-10. Pipelining Delay for Sequential-2D NON-PAIR Mode

Pipelining Bit (Bit 14)	Phase Bit (Bit 13)	Pipeline Delay in Output Pixels
0	0	6
0	1	7
1	0	2
1	1	3

Also note that when in PAIR mode (with bit 11 = one), the amount of pixel delay does not change, but half as many reads and writes are required to fill the pipeline because each read or write of the averager transfers two pixels. For example, when in the 82750PA emulation mode (bit 14 = 0), with zero phase (bit 13 = 0) and pair mode (bit 11 = 1), three indeterminate pixel pairs must be read before the first good pixel pair is read. In the same case but with the phase bit = 1, the fourth pixel pair read contains one good pixel and one indeterminate pixel, and the fifth pixel pair read contains two good pixels.

RESERVED

Bits 15 and 12 are reserved for future use. Write ZEROs into these bit positions.

Signature Register

{*hwid*}

The signature register can be read either by the host CPU or by microcode to determine the version of the 82750PB. The value of the signature register can be used to distinguish between the 82750PA, the 82750PB in the 82750PA emulation mode, and the 82750PB in native mode. The currently defined signature values given in Table 2-11.

Table 2-11. Signature Values

Value	Definition
0xFFFF	The 82750PA
0xFFFE	The 82750PB Emulating the 82750PA
0XFFFC	The 82750PB in Native Mode

NOTE:
All other signature values are presently undefined but may be used in the future to denote other versions of the 82750 architecture.

Display Format Registers

{*yeven, yodd, vu, vptr*}

The 82750PB's processor can write to the display registers in the VRAM interface. These registers are pointers and pitch values that address display bitmaps and 82750DB register loads in VRAM. Pointers are 32-bit values that specify the specify the starting byte address of a bitmap or register load within a 4 GByte address space. The bottom two address bits are ignored since display bitmaps and register loads must start on a double word boundary. Therefore, the internal representation of a pointer is a 30-bit value. The pitch value associated with each pointer indicates the number of bytes between the start of two lines of a display bitmap or between the start of two register loads. The pitch is a single 16-bit value with its two least significant bits ignored, since the pitch must be an integer number of double words. Currently, there is also a restriction in the 82750DB limiting all display bitmap pitches to powers of two; so, the maximum display bitmap pitch is $\pm 2^{14}$ Bytes = $\pm$16 kBytes. The display registers are described in Table 2-12.

Table 2-12. Display Registers

Register	Description
yeven-lo, hi	This register pair points to the start of the Y bitmap or main bitmap that is to be displayed during an even field scan.
yodd-lo, hi	This register pair points to the start of the Y bitmap or main bitmap that is to be displayed during the odd field scan.
ypitch	The value in this register is added to the current Y bitmap pointer value each time a Y transfer is performed.
vu-lo, hi	This register pair points to the start of the VU bitmap. This bitmap is read to generate the VU values for both odd and even field scans.
vupitch	This value is added to the current VU bitmap pointer value each time a VU transfer is performed.
vptr-lo, hi	This register pair points to the start of a series of 82750DB register loads stored in VRAM.
vpitch	This value is added to the current 82750DB register load pointer each time a 82750DB register load is performed. The pitch is equal to the number of bytes from the start of one register load to the start of the next register load.

3.0 HARDWARE INTERFACE

VRAM Interface

The VRAM interface performs the following operations:

- Maintains VRAM pointers for the two input FIFOs, the two output FIFOs, the statistical decoder, the Y (main) bitmap, the VU bitmap, and the 82750DB register load.
- Decodes VBUS codes and takes appropriate actions such as generating a transfer cycle, scheduling refresh cycles, or generating interrupt conditions.
- Arbitrates VRAM accesses between the two input FIFOs, the two output FIFOs, the statistical decoder, the transfer request logic, the VRAM refresh logic, and the external VRAM access logic.
- During a memory cycle, performs appropriate address arithmetic on the VRAM pointer used for that memory cycle.
- As a result of certain VBUS codes, performs a shadow copy that consists of copying display-related VRAM pointer values from shadow registers (that are loaded by the host CPU or the microcode processor) to working registers where the various pointers are used for transfer cycles when the 82750DB is refreshing the display screen.

Table 3-1. VRAM Interface Signals

Signal	Description
MREQ#	**MEMORY REQUEST** is asserted during the first cycle of a VRAM memory access.
TRNFR#	The **TRANSFER** output indicates the current memory cycle is a result of a 82750DB transfer request.
RFSM#	The **REFRESH** output indicates the current memory cycle is a result of a 82750DB refresh request.
NXTFST#	The **NEXT FAST** output indicates the next memory access will use the same row address as the current memory access. This facilitates the use of page mode memory accesses.
MRDY#	The **MEMORY READY** input indicates the availability of valid data on the D[31:0] pins.

VRAM ACCESSES

The 82750PB can initiate five different types of memory accesses: FIFO read, FIFO write, transfer read, transfer write, and refresh. In addition, the 82750PB supports VRAM accesses by external logic. During an external access VRAM cycle, the 82750PB tri-states its VRAM address and data buses and performs a host VRAM read or host VRAM write cycle. There is another operation performed by the 82750PB, a shadow copy, that is not a VRAM cycle but is arbitrated as though it were, since no VRAM cycles can take place during a shadow copy.

The seven types of VRAM cycles initiated by the 82750PB, including host VRAM read and host VRAM write, begin with the 82750PB asserting a combination of its three VRAM cycle definition outputs: TRNFR#, RFSH#, and WE#. External logic detects the state of these signals, validated by MREQ#, and produces the appropriate sequence of VRAM control signals (RAS, CAS, etc.) to perform the type of memory cycle the 82750PB has requested. The 82750PB requires that each of these VRAM cycles take a minimum of two T-cycles, or T-states, denoted T1 and T2. External logic can insert additional T2 states in order to stretch the VRAM cycle to more than two T-cycles. The start of a new VRAM access cycle is signaled by the assertion of MREQ# for the first T-cycle, T1. The VRAM access cycle definition signals, TRNFR#, RFSH#, and WE#, are asserted at the start of T1 and remain asserted until the end of the last T2. Other VRAM operations can be described similarly by sequences of T-states. Refer to Figures 3-4 and 3-5 on page 38 for timing diagrams.

Table 3-2 defines the states used for all VRAM access operations. A state diagram for the VRAM/Host Interface is provided in Figure 3-1. This diagram includes the FIFO access states

Table 3-2. 82750PB VRAM Access States

State	Description
Ti	Idle State, No VRAM Activity
T1, TF1	First State of a VRAM FIFO Cycle
T2, TF2	Last State of a VRAM FIFO Cycle
TSC	The T-State required to perform a shadow copy
TTX1	First State of a VRAM Transfer Cycle
TTX2	Last State of a VRAM Transfer Cycle
TRF1	First State of a VRAM Refresh Cycle
TRF2	Last State of a VRAM Refresh Cycle

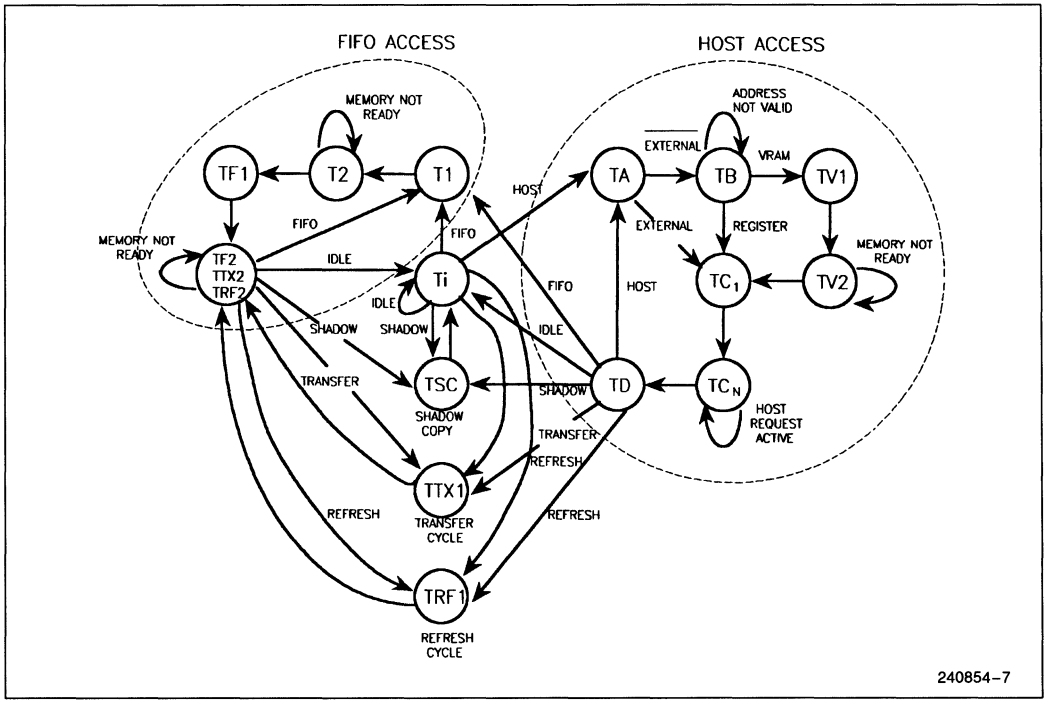


Figure 3-1. Access State Diagram

Note that during successive VRAM cycles it is not necessary to go back to the idle state, Ti, between each cycle; the T_{F2} state can be followed directly by a T1 state, starting at the next VRAM cycle. This results in efficient utilization of the 82750PB/VRAM bandwidth by allowing a VRAM cycle time of 2 T-states.

FAST VRAM CYCLES

When the 82750PB performs Data Read or Data Write VRAM cycles for the input or output FIFOs, it performs two 32-bit accesses to read or write one 64-bit value. These accesses are always performed in a sequence of EvenAddress followed by EvenAddress + 1, which guarantees both that the two sequential accesses will be in opposite banks and that the two accesses will be within the same VRAM page. This allows external logic to use either bank-interleaving or a page-mode access to complete the second access of the sequence and improve the VRAM bandwidth. However, the second access does not need to be handled differently from the first. Except for the assertion of the NXTFST# signal, both accesses are treated as standard VRAM accesses. External logic can ignore the NXTFST# signal, though, and treat the two accesses as two normal data read or data write cycles. Note that NXTFST# is not asserted for transfer, refresh, or host memory accesses.

The NXTFST# output signal is provided for cases when external logic can generate a faster access for the second access of the two sequential accesses. During such a pair of accesses, NXTFST# is asserted during the first of the two accesses in order to provide sufficient time for the external logic to generate the appropriate fast memory cycle for the second access. Refer to the timing diagrams in Figures 3-4 and 3-5 (page 38) for examples illustrating the use of the NXTFST# signal.

VBUS CODES

Transfer request, interrupt, and synchronization codes are sent over the BUS from the 82750DB to the 82750PB. The codes recognized by the 82750PB are listed in Table 3-3, along with the actions taken by the 82750PB as a result of receiving each code. Codes that cause TRANSFER cycles must be asserted for TC_{VBTR} cycles of the 82750PB to insure that, in the worst case, the 82750PB completes the transfer cycle before the code is released and the 82750DB starts shifting data from the VRAM shift registers. Other codes must be asserted for a minimum of two 82750PB clock cycles. Only the codes given in the Table 3-3 are valid codes for the VBUS. Other codes are reserved for future use and should not be used.

Table 3-3. VBUS Codes

Binary	Name	Cycles	Action
0000	YBMX	TC_{VBTR}	TXRD Cycle Using Yc; Yc = Yc + Yp*
0001	VUBMX	TC_{VBTR}	TXRD Cycle Using VUc; VUc = VUc + VUp
0010	REGX	TC_{VBTR}	TXRD Cycle Using Vc; Vc = Vc + Vp
0011	WRDIGX	TC_{VBTR}	TXWR Cycle Using Yc; Yc = Yc + Yp
0100	YNPBMX	TC_{VBTR}	TXRD Cycle Using Yc; Yc = Yc
0101	Reserved		Reserved
0110	Reserved		Reserved
0111	WRDIGNPX	TC_{VBTR}	TXWR Cycle Using Yc; Yc = Yc
1000	DFL	2	DFL Int; Shadow Copy**
1001	82750DBSD	2	82750DB Shutdown Interrupt
1010	REFRESH	2	Schedule N Refresh Cycles
1011	Reserved		Reserved
1100	VODD	2	VBI Int; OF Int; Shadow Copy Odd; Hline = 0***
1101	VEVEN	2	VBI Int; EF Int; Shadow Copy Even
1110	HLINE	2	Icnt+ + (Increment Line Counter)
1111	NULL	2	No Action

NOTES:
*Yc—Y bitmap pointer, current; Yp—Y bitmap pitch; VU—VU bitmap; V—82750DB register load.
**Shadow Copy with Yc = Y-start-odd in odd field; Yc = Y-start-even in even field.
***Hline—Horizontal Line Counter.

METHOD FOR CALCULATING TC_{VBTR} CYCLES

In a system containing the 82750PB and the 82750DB, the 82750DB is programmed with a value that indicates the maximum latency from when the chip asserts a TRANSFER REQUEST VBUS code to the 82750PB until the 82750PB has completed the TRANSFER cycle. This value is programmed so that the 82750DB can begin to generate SCLKs that will shift data from the VRAMs. This latency, designated TC_{VBTR}, is dependent on system parameters according to the following equation:

$$TC_{VBTR} = \left[T_{TRNFR} + \text{Max}\left[\frac{T_{HOST}}{T_{FIFO}}\right]\right] + 2$$

where

T_{TRNFR} = Number of T-states to perform a TRANSFER cycle

T_{HOST} = Maximum number of T-states for any HOST cycle (measured from states T_A to T_D)

T_{FIFO} = Maximum number of T-states for any FIFO access (measured from T_1 to the last T_{F2} state)

Example:

For a Transfer cycle timing of

$$T_{TX1}\ T_{TX2}\ T_{TX2}\ T_{TX2}\ T_{TX2}\ T_{TX2}$$

$$T_{TRNFR} = 6,$$

a FIFO access consisting of two memory cycles

$$T_1\ T_2\ T_2\ T_2\ T_2\ T_{F1}\ T_{F2}\ T_{F2}$$

$$T_{FIFO} = 8,$$

and the longest HOST cycle being

$$T_A\ T_B\ T_B\ T_B\ T_{V1}\ T_{V2}\ T_{V2}\ T_{V2}\ T_{V2}\ T_{C1}\ T_{CN}\ T_{CN}\ T_D$$

$$T_{HOST} = 13$$

then,

$$TC_{VBTR} = \left[6 + \text{Max}\left[\frac{8}{13}\right] + 2\right]$$

$$= 21 \text{ T-states (82750PB clocks).}$$

The value loaded into the 82750DB SCLK Delay register is:

$$TDB_{SCLK} = \text{82750DB SCLK Delay} = \text{INT}\left[\frac{\text{82750DB Input Clk Freq.}}{\text{82750PB Input Clk Freq.}} \times TC_{VBTR}\right] + 1$$

If the 82750DB is runnign at 28 MHz and the 82750PB is running at 25 MHz, the value would be:

$$TDB_{SCLK} = \text{INT}\left[\frac{28}{25} \times 21 + 1\right] = 24$$

This value would be loaded into the 82750DB SCLK Delay register.

PRIORITY

Each time the VRAM state machine completes a VRAM operation and returns to the Ti state, it examines all pending VRAM access requests and selects the highest priority request for the next VRAM operation. The priority ordering of these requests are listed in Table 3-4.

Table 3-4. Priority of VRAM Operations

Request Type	Priority
Transfer Cycle	Highest
Shadow Copy	•
Host Access	•
VRAM Refresh	•
FIFO Read/Write	Lowest

NOTE:
The shadow copy is treated as a VRAM operation even though it does not result in an access to VRAM.

The VRAM refresh operation is placed low on the priority list to reduce the latency in servicing transfer requests and external VRAM requests. Since a single REFRESH code from the 82750DB schedules a number of refresh cycles, a higher priority for refresh would cause all the refresh cycles to occur in a burst that would lock out all lower priority requests until all refresh cycles completed. Instead, the following restriction applies to all request types with higher priority than refresh: high priority requests, such as transfer cycles, shadow copies, and external VRAM access must occur infrequently enough to allow proper refresh of the VRAM chips. Transfer cycles and shadow copies, by their nature, occur infrequently so they are not generally a problem.

There is a separate priority scheme for the five FIFO channels. The scheme used is rotating priority with automatic override and single cycle arbitration. Rotating priority means that the priority is assigned in a fixed cyclic order with the lowest priority given to the FIFO channel that "won" the last FIFO access.

There is only one level of memory , so the order that requests arrive is not a factor in the arbitration. The cyclic order is given in Figure 3-2.

As an example, if input FIFO 0 (abbreviated if0) was the last channel to perform a cycle, the priority order for the next FIFO access (from highest to lowest) would be: if1, sd, of0, of1, and if0.

Automatic override that the rotating cyclic priority can be bypassed if there is an URGENT condition for one of the channels. A channel is urgent if the microcode processor is frozen because the processor is waiting for that channel to be ready. The channel can be either an input channel that is empty or an output channel that is full. In this case, the urgent channel gets the next available cycle. However, the priority will still be lower than non-FIFO requests, such as refresh cycles.

Single clock cycle arbitration means that the selection of the next channel that will get an access occurs in a single T-cycle or T-state, either in a Ti state or during the last T2 state of the previous VRAM cycle.

VRAM POINTERS

The VRAM interface maintains VRAM pointers for the FIFOs, as well as display-related pointers for the 82750DB. Internally each pointer or address is stored as a 30-bit value addressing a double word in VRAM. The pointer values are read and written as two 16-bit words representing a 32-bit byte address (refer to the Figure 3-3). With a 30-bit double word address, the 82750PB can decode a VRAM address space of 1G double words or 4 GBytes.

Input and output FIFOs can address down to a single word or byte in VRAM. A FIFO's pointer is post-incremented or post-decremented in parallel with its VRAM read or write cycle.

The statistical decoder can only start decoding bitstreams on double word boundaries in VRAM and can only increment through VRAM. The decoder's pointer is post-incremented in parallel with each of its VRAM read cycles.

Display-related pointers are updated by adding a pitch value to the current value during the corresponding transfer cycle.

SHADOW COPY

When a VODD, VEVEN, or DFL code is received from the 82750DB over the VBUS, a shadow copy is scheduled. The actual shadow copy will occur as soon as the priority logic allows. Any VRAM access in progress must complete and a pending transfer cycle, if any, must be performed before the shadow copy can start. During the operation, shadow registers for the Y-START, Y-PITCH, VU-START, VU-PITCH, 82750DB-START, and 82750DB-PITCH are copied into the corresponding working registers. During display refresh, the address arithmetic is performed on the working registers. The shadow registers can be loaded by the host CPU or by a microcode routine with less critical timing constraints, and then copied instantly by a shadow copy with it is time to update the registers, either prior to the next field or during the active display for split screen effects.

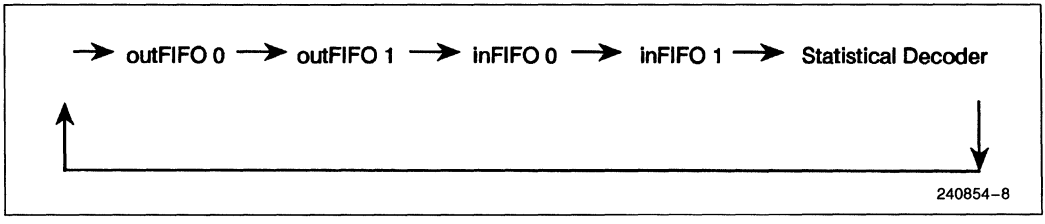


Figure 3-2. Cyclic Ordering of FIFOs

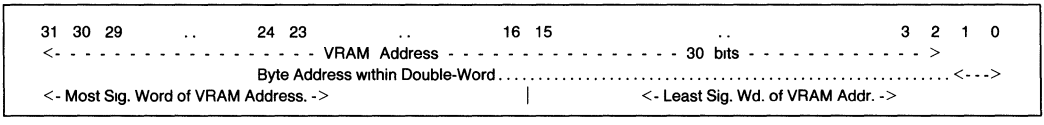


Figure 3-3. VRAM Addressing

There are actually two shadow registers for Y-START. One for start of odd fields and one for start of even fields. A VODD code causes Y-START-ODD to be copied into the working register Y-CURRENT. Similarly, a VEVEN code causes the Y-START-EVEN to be copied into Y-CURRENT. A DFL code causes the Y-START-ODD value to be copied if the most recent start of field code received is a VODD, or a Y-START-EVEN value if the most recent start of field code was a VEVEN. This allows a simple interlaced or non-interleaced display to be refreshed with no host CPU intervention. For more complex displays, such as split screens, the host CPU must update the shadow registers prior to each shadow copy. A shadow copy operation requires 2 T-cycles.

Host Interface

The Host Interface provides the following functions:

- Arbitrates host CPU and 82750PB access to VRAM.
- Provides the host access to external devices.
- Provides the host access to 82750PB internal registers and memories.

Signals specific to the Host Interface are listed in Table 3-5.

Table 3-5. Host Interface Signals

Signal	Description
HREQ#	**HOST REQUEST:** Asynchronous request from the host for all types of host access. Used both to request and release system buses.
HREG#	**HOST REGISTER:** Single-ranked control to request host access to 82750PB internal registers in concert with HRAM#.
HRAM#	**HOST VRAM:** Single-ranked control to request host access to VRAM in concert with HREG#.
HALEN#	**HOST ADDRESS LATCH ENABLE:** Asynchronous status from the host indicating the presence of valid address, write enable (transaction direction control), and the byte enables at the interface of the 82750PB.
HBUSEN#	**HOST BUS ENABLE:** 82750PB synchronous status granting the host access to the address, write enable, data bus, and byte enables at the interface of the 82750PB.
HRDY#	**HOST READY:** 82750PB synchronous status to the host indicating the presence of valid data appearing at the 82750PB's databus for VRAM and register accesses and optionally for external accesses.
HINT#	**HOST INTERRUPT:** 82750PB synchronous interrupt to the host, set under direct or indirect microprogram control.

Signals common to the host, VRAM, and external device interfaces are listed in Table 3-6.

Table 3-6. Host, VRAM, and External Device Interfaces

Signal	Description
A[31:2]	**ADDRESS BUS:** System address bus used to select unique VRAM, the 82750PB register, and external device locations that will be accessed under host control. The lower seven bits A[8:2] are bidirectional and are used during register accesses
D[31:0]	**DATA BUS:** Bidirectional system data bus used to transfer data to and from all sources and destinations. When transferring 16-bit host register values, the data bus MSH and LSH will both carry identical values.
WE#	**WRITE ENABLE:** Bidirectional, single-ranked signal used to determine the data transfer direction. When active during host register cycles, data flows from the host to an 82750PB destination. During host VRAM cycles, WE# active will define the data direction to be from the host to VRAM.
BE[3:0]#	**BYTE ENABLE:** Bidirectional signals used to select the bytes that will be modified during data transactions. All host register transactions are performed 16 bits at a time, while VRAM may be modified 8 bits at a time.

As with VRAM operations, host operations are described through a sequence of T-states. Table 3-7 defines the T-states used to implement all host transactions with VRAM, external devices, and the 82750PB.

The master execution state diagram that defines the VRAM/Host transactions is provided in Figure 3-1.

Table 3-7. 82750PB Host Transaction States

State	Description
TA	First state of any host transaction. Entry into TA will be granted after HREQ# has been held active for a minimum of two clock cycles. During this state, the 82750PB will tri-state its address, data bus, write enable, and byte enable signals to provide a full cycle of "dead-band" before the assertion of HBUSEN#. In the state immediately following TA HBUSEN# will assert, allowing the host to drive the host buses.
TB	First cycle in which the host is granted bus access for register or VRAM transactions. The sequencer will remain in TB until HALEN# is received, indicating that the address write enable and byte enable signals are stable at the 82750PB pins.
TC1	First cycle that output data is valid.
TCn	This state is entered to wait for the completion of the current host cycle. The cycle is defined as complete when HREQ# deasserts. HRDY# is asserted along with valid data until the transition to state TD occurs.
TD	The last cycle of a host transaction. HBUSEN# is deasserted allowing one dead-band cycle to allow control of the address, data, write enable, and byte enable signals to be returned to the 82750PB.
TV1	First cycle of a Host VRAM transaction. Memory is requested and is followed by a transition to TV2.
TV2	Last cycle of a Host VRAM transaction. The sequencer will remain in TV2 until MRDY# is received.

A single stage of input synchronization is employed for HREG#, HRAM#, WE#, and BE[0]#, while HREQ# and HALEN# each require two stages. T-state transitions are caused by the synchronized versions of these signals.

The synchronized versions of HREG# and HRAM# must be stable before entry into T-state TA. The synchronized versions of WE#, BE[0]#, and HALEN# should be stable before exiting T-State TB. Once asserted, all of the above signals should remain stable until the deassertion of HBUSEN#.

The type of host cycle to perform is determined by the states of HREG# and HRAM# as indicated in Table 3-8.

Table 3-8. Host Cycle Types

HREG#	HRAM#	Host Cycle Type
1	1	External
0	1	Register
1	0	VRAM
0	0	Reserved

HOST REGISTER ACCESS

The host has access to the 82750PB's internal registers and memories to monitor and control the operation of the microcode processor, provide a means of debugging microprogram routines, and to function as the primary test port for production testing.

Register access is initiated by the host asserting HREQ#, HREG#, and HRAM# as shown in Table 4-8 and in the timing diagrams on pages 38 through 41. After the host has been granted bus access by an active HBUSEN# in state TB, the address, write enable, and byte enables may be driven. After these signals have stabilized HALEN# is asserted, enabling a read or a write operation to occur.

In the case of a register read, state TC1 is entered and the data bus is driven with the internal value. One cycle later, a transition to state TC occurs, and HRDY# activates, signaling the presence of stabilized data at the 82750PB data pins. This state (TC) will be maintained until the host deasserts HREQ#, signaling the completion of the cycle that caused a transition to state TD.

In the case of a register write, TC1 is again entered (from TB), but the data bus may now be driven by the host. (During host cycles, data bus drive activity is indirectly controlled by WE# and an additional dead-band is provided by entry into state TC1 to allow for internal WE# stabilization.) Stable data at the 82750PB interface, as well as the completion of the write cycle, is signaled by the deassertion of HREQ#. As with reads, the deactivation of HRDY# signals the transition to state TD.

As state TD is entered, HRDY# and HBUSEN# deassert, the address data, write enable, and byte enables tri-state, and bus control is returned to the 82750PB in the following cycle.

HOST VRAM ACCESS

Because the 82750PB is so closely coupled with VRAM, host accesses to VRAM are arbitrated and controlled by the 82750PB. VRAM access is initiated by the host asserting HREQ#, HREG#, and HRAM# as shown in the Host Cycle Table above and in the timing diagrams on pages 38 through 41. After the host has been granted bus access by an active HBUSEN#, the address, write enable, and byte enables may then be driven. After these signals have stabilized at the memory devices (or longest relevant propagation path), HALEN# is asserted, enabling a read or a write operation to occur.

Because VRAM will not drive the data bus until after a memory request, a transition into state TC1 to allow for data bus direction stabilization is not required. Instead, a transition to state TV1 occurs, which asserts MREQ# for a single cycle and is followed by a transition to TV2. TV2 will remain the current state until the reception of an active MRDY#.

In the case of a VRAM read, the memory data bus will be driven during TV1, and valid data will appear in state TV2. Data will be guaranteed valid coincident with the deassertion of MRDY# from memory.

In the case of a VRAM write, the memory data bus is driven with valid data during TV1. Again the reception of MRDY# will serve to indicate the completion of the memory operation.

NOTE:
The host device must be able to transmit or receive memory data in order to be valid at the trailing edge of MRDY# at the data's destination (memory or host).

After MRDY# becomes active, a transition from TV2 into TC1 is accomplished to allow time to propagate data to the host. TC is then entered to await the deassertion of HREQ# (if it has not already occurred). TD is then entered, duplicating the dead-banding previously described.

HOST EXTERNAL ACCESS

In addition to VRAM and register host access, and external device access mechanism is provided. During this access, upon the receipt of HREQ# with HREQ# and HRAM# inactive, the 82750PB releases the address, data, write enable, and byte enables in state TA.

The difference here is that state TC1 is directly entered from TA, thereby ignoring any transitions of HALEN#. Since the 82750PB also ignores the data bus direction control (write enable) the host and an external device may communicate unencumbered by the 82750PB.

Entry into state TC directly follows TC1 in the expected sequence and remains there until HREQ# is released. This is followed by entry into TD. HBUSEN# is asserted during the timing that TC1 and TCN are active.

NOTE:
During an external access, HRDY# is not asserted unless the external logic asserts MRDY#. Refer to the host external access timing diagram on page 40.

HOST REGISTER ADDRESS MAPPING

Table 3-9 shows the host address mapping of the on-chip registers and memories, in terms of the offset in bytes, from the base address for 82750PB accesses. Note that the 82750PB only supports word accesses to these registers. Therefore, the least significant bit of the byte offset should be set to zero. The 82750PB forms the register address from inputs on the A[31:2] pins and BE#[3:0] pins. The A[31:2] specify the double word address of the register, and combinations of the BE# pins determine which of the two words with the double word is being addressed. BE#[3:0] = 1100_2 selects the least significant word within a double word, and BE#[3:0] = 0011_2 selects the most significant word within a double word. These are the only two valid patterns for BE# inputs during a host register access cycle.

CAUTION:
The host should only perform 16-bit word reads or writes to 82750PB registers. The 82750PB does not support byte reads or writes or double word reads or writes to on-chip registers.

Table 3-9. Host Address Mapping

Byte Offset	Description
0x000–0x07E	(a) A source and destination registers
0x080–0x0FE	(b) B source and destination registers
0x100–0x17E	(c) Microcode processor control and status registers
0x180–0x1FE	(d) VRAM pointer RAM

NOTE:
When the host CPU reads or writes to areas (a, b, or d) and the 82750PB is not already in a HALT state, the microcode processor is automatically HALTED for the one T-cycle actually required to complete the data transfer, and then the processor is restarted after the transfer is complete. If the 82750PB is in a HALT state when the host access is initiated, it will remain in the HALT state following the completion of the access. This is transparent to both the host CPU and the microcode processor.

NOTE:
During an access to areas (a) or (b), bits 6:1 of the byte offset should be set to the source or destination code for the register that will be read or written. The coding is the same as used in the microcode instruction word. Bit 0 is always set to a zero. Refer to the 82750PB Source and Destination Coding Table found in Chapter 6.

Area (c) contains one write-only register, the CONTROL register, and two read-only registers, the INTERRUPT FLAG register and the microcode PROCESSOR STATUS register. The CONTROL register is used to halt or single-step the microcode processor, which enables or masks interrupts to the host CPU, selects the signal that is output via the PMON/FRZ pin, and enables or disables the 82750PA emulation mode. The bit assignments for the CONTROL register are given in Table 3-10.

During reset of the 82750PB, the HALT bit is set to a one, the six Interrupt Enable bits are reset to zero, the PMON/FRZ bit is set to zero (so that the FRZ signal is output), and the Enable 82750PB bit is reset to zero (so that on reset, the 82750PB starts in a 82750PA emulation mode).

Table 3-10. Bit Assignments for Microcode Processor CONTROL Register {Write-Only, Byte Offset = 0x100}

Bit	Name	Description
Bit 0	HALT	1 = Microcode Processor Halt 0 = Microcode Processor Run
Bit 1	SINGLE-STEP	1 = Execute One Instruction and then Halt (Only when Already Halted, Bit 0 = 1) 0 = No Action
Bit 2	Enable MCINT	1 = Enable Microcode Interrupts to Host CPU 0 = Mask Microcode Interrupts
Bit 3	Enable VBI	1 = Enable Vertical Blanking Interrupt to Host CPU 0 = Mask Vertical Blanking Interrupt
Bit 4	Enable DFL	1 = Enable DFL Interrupt to Host CPU 0 = Mask DFL Interrupt
Bit 5	Enable SD	1 = Enable 82750DB Shutdown Interrupt to Host 0 = Mask SD Interrupt
Bit 6	Enable OFI	1 = Enable Odd Field Interrupt 0 = Mask OF Interrupt
Bit 7	Enable EFI	1 = Enable Even Field Interrupt 0 = Mask EF Interrupt
Bits 8–12		1 = RESERVED; Write as Zeros
Bit 13	PMON/FRZ	1 = Output PMON# Signal on PMFRZ# Pin 0 = Output FRZ# Signal on PMFRZ# Pin
Bit 14		1 = RESERVED; Write as Zero
Bit 15	Enable 82750PB	1 = Enable 82750PB Mode 0 = Enable 82750PA Emulation Mode

NOTE:
All other bits are reserved for future use, and should be written as zeros.

The INTERRUPT FLAG register holds a flag for each of the six interrupt sources. A flag bit is set to a one when the interrupt condition is detected (independent of the state of the corresponding Interrupt Enable/Mask bit in the CONTROL register), and all flags are cleared to zero each time the INTERRUPT FLAG register is read. If this register is read during the same cycle that an interrupt condition is detected, the flag bit corresponding to that interrupt condition will remain at a one. This new interrupt condition will then be seen by the host processor when it next reads the INTERRUPT FLAG register. The flag insures that an interrupt is not lost if it occurs at the same cycle that the INTERRUPT FLAG register is read (and reset). In addition, the Microcode Interrupt source has an overflow flag that indicates if more than one Microcode Interrupt has occurred since the Interrupt Flag register was last read. The bit assignments for the INTERRUPT FLAG register are listed in Table 3-11.

The PROCESSOR STATUS register holds three status bits: HALT, FREEZE, and PMON. HALT indicates that the processor is HALTED due to a HALT bit in the CONTROL register being set to a ONE or due to the HALT# pin being asserted. FREEZE indicates that the processor is waiting for one of the VRAM channels to become ready or is waiting for an access to the VRAM pointer RAM. PMON is a signal that can be toggled by a special ALU opcode or a special B source code. This signal can be used for performance monitoring of microcode. In addition, the Interrupt Mask bits that are written into the PROCESSOR CONTROL register can be read from this register. These mask bits are read in the same polarity that they are written, but note that the bit positions and bit ordering are not consistent with the PROCESSOR CONTROL register. The bit assignments for this register are given in Table 3-12.

Area (d), the VRAM point RAM, address mapping is given in Table 3-13.

Table 3-11. Bit Assignments for INTERRUPT FLAG Register (Read-Only, Byte Offset = 0x100)

Bit	Description
Bit 8:0	Not Used, the State of These Bits Are Not Specified
Bit 9	EF Interrupt Flag
Bit 10	OF Interrupt Flag
Bit 11	MCINT Overflow Flag
Bit 12	82750DB Shutdown Interrupt
Bit 13	MCINT Microcode Interrupt
Bit 14	VBI Vertical Blanking Interrupt
Bit 15	DFL Display Format Load Interrupt

Table 3-12. Bit Assignments for PROCESSOR STATUS Register (Read-Only, Byte Offset = 0x102)

Bit	Description
Bit 0	HALT (1 = Halted, 0 = Running)
Bit 1	FREEZE (1 = Frozen, 0 = Running)
Bit 2	PMON (1 = Active, 0 = Inactive)
Bit 9:3	Not Used, the State of These Bits is Not Specified
Bit 10	MCINT Microcode Interrupt Mask
Bit 11	VBI Vertical Blanking Interrupt Mask
Bit 12	DFL Display Format Load Interrupt Mask
Bit 13	82750DB Shutdown Interrupt Mask
Bit 14	OF Interrupt Mask
Bit 15	EF Interrupt Mask

Table 3-13. VRAM Pointer RAM Mapping

Byte Offset	Name	Description
0x180 0x182	Yc-lo Yc-hi	Working Copy of Y Pointer
0x184 0x186	out1-lo out1-hi	Output FIFO 1 Pointer
0x188	Yc-pitch	Working Copy of Y Pitch
0x18A		RESERVED
0x18C 0x18E	out2-lo out2-hi	Output FIFO 2 Pointer
0x190 0x192	VUc-lo VUc-hi	Working Copy of VU Pointer
0x194 0x196	in1-lo in1-hi	Input FIFO 1 Pointer
0x198	VUc-pitch	Working Copy of VU Pitch
0x19A	82750DBc-pitch	Working Copy of 82750DB Pitch
0x19C 0x19E	in2-lo in2-hi	Input FIFO 2 Pointer
0x1A0 0x1A2	82750DBc-lo 82750DBc-hi	Working Copy of 82750DB Pointer
0x1A4 0x1A6	huff-lo huff-hi	Working Copy of Statistical Decoder Pointer
0x1A8 0x1AA	Yeven-lo Yeven-hi	Shadow Copy of Y Start Even Pointer
0x1AC 0x1AE	Yodd-lo Yodd-hi	Shadow Copy of Y Start Odd Pointer
0x1B0	Y-pitch	Shadow Copy of Y Pitch
0x1B2	rfcnt	RFSH Cycles per RFSH Code from 82750DB
0x1B4 0x1B6	VU-lo VU-hi	Shadow Copy of VU Start Pointer
0x1B8	VU-pitch	Shadow Copy of VU Pitch
0x1BA	82750DB-pitch	Shadow Copy of 82750DB Pitch
0x1BC 0x1BE	82750DB-lo 82750DB-hi	Shadow Copy of 82750DB Pointer

Initializing the 82750PB

The 82750PB is placed in a RESET state by asserting RESET# for at least ten T-cycles. In the RESET state, which continues until RESET# is released, all of the 82750PB's outputs are tri-stated for compatibility with board test requirements.

Proper initialization of the 82750PB requires that the 82750PB is held in a RESET state by keeping RESET# active for at least 10 T-cycles, and then releasing RESET#. This is referred to as the INITIAL state. In the INITIAL state:

- The microcode processor is halted.
- All six interrupts are masked, and the interrupt latches are cleared.
- The 82750PA/82750PB instruction format select bit is set to the 82750PA.
- The VRAM interface is ready to service VRAM requests; however, none of the VRAM pointers are valid.

- The number of refresh cycles that will be generated each time a RFSH code is received from the 82750DB is set to 14 cycles.
- All bidirectional I/O pins are tristated.

After the 82750PB has been initialized, i.e., placed in the INITIAL state, but prior to releasing the 82750DB's reset signal, the following operations must be performed:

- Load the REFRESH-CYCLES-PER-LINE register with the appropriate value (the equation for the value is: VALUE = $(2^N - 1)$, where N is the number of cycles; for example, 5 refresh cycles would result in VALUE = $2^5 - 1 = 31_{10} = 001F_{16}$.
- Load the shadow copies of Y, VU, and 82750DB pointers and pitches.
- Load the appropriate 82750DB Register Load list into VRAM starting at the address pointed to by the 82750DB pointer.

Prior to releasing the microcode processor from its HALTed state to run a microcode program, the following operations must be performed:

- If 82750PB code is to be executed, bit 15 of the 82750PB CONTROL register must be set to a one.
- Load a microcode program into microcode RAM on the 82750PB by writing to the three instruction word registers (*mcode1* – the most significant word of the instruction, *mcode2,* and *mcode3* – the least significant word of the instruction, the one containing the next address field) and then writing to *maddr,* the address in microcode RAM where the instruction will be loaded.
- Load the PC with the address in microcode RAM of the first instruction to be executed.
- Write to the 82750PB CONTROL register with the HALT bit (bit 0) set to zero, causing the processor to start executing an instruction sequence, or with the SINGLE-STEP bit (bit 1) set to a one (keeping HALT also set to one), causing the processor to execute a single instruction.

Performance Monitoring

Two signals, FRZ# and PMON#, which are useful for microcode performance monitoring, are available both as external signals, multiplexed on a single output pin, and as bits in the Processor Status register. FRZ# is active for each T-cycle when the microcode processor is frozen, waiting for access to VRAM or to the VRAM Pointer RAM. PMON# can be toggled by a special ALU opcode or a special B bus source code. This allows PMON# to be used to indicate what particular segment of microcode is being execute. The PMON/FRZ bit in the Processor Control register selects the signal that is being output.

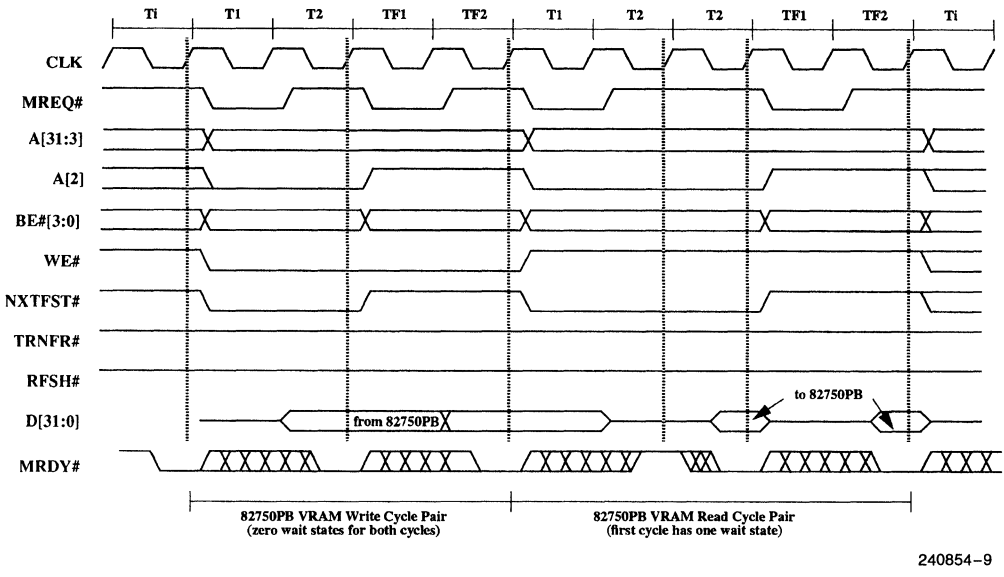
Freezes may indicate that the microcode routine is not making the most efficient use of the input and output FIFO buffering. This is particularly important for the inner loops of graphics and video routines that are memory-bandwidth limited. Ideally, inner loops should be balanced so that the rate pixels are processed is equal to the rate that they can be read from and written to VRAM with no freezes. The buffering in the input and output FIFOs serve to make sequential reads and writes to VRAM more efficient by performing full 64-bit reads and writes, instead of individual 8-bit or 16-bit accesses. This has the effect of averaging the VRAM read/write rate over a number of instruction times. For example, if the 82750PB is performing a 64-bit read or write every 8 T-cycles, for an average of 8 bits per T-cycle, a two instruction inner loop could read one 8-bit pixel and write one 8-bit pixel without any freezes occurring (assuming the source pixels and the destination pixels are each sequential).

The PMON# provides a more standard performance monitoring capability by indicating when a particular segment of microcode, bracketed by special instructions that toggle the PMON# signal, is being executed. This allows either absolute execution-time measurement or measurement of the fraction of the total execution time that is required by the segment. Either the ALU opcode PROF or the B bus source code PROF will toggle the PMON signal.

An external HALT pin is provided on the 82750PB to allow external debugging hardware to immediately halt the microcode processor. Activating this input causes the microcode processor to halt prior to executing the next instruction. When the processor is halted, the VRAM interface portion of the 82750PB continues to operate normally, performing transfer cycles, refresh cycles, and shadow copies as requested by the 82750DB.

Host/VRAM Timing Diagrams

Figures 3-4 through 3-8 are Host/VRAM Timing Diagrams.



NOTES:
1. Address pin A[2] is always ZERO for the first cycle of a cycle pair and ONE for the second cycle.
2. The two cycles of a cycle pair are both writes or both reads.

Figure 3-4 VRAM Read and Write Cycles

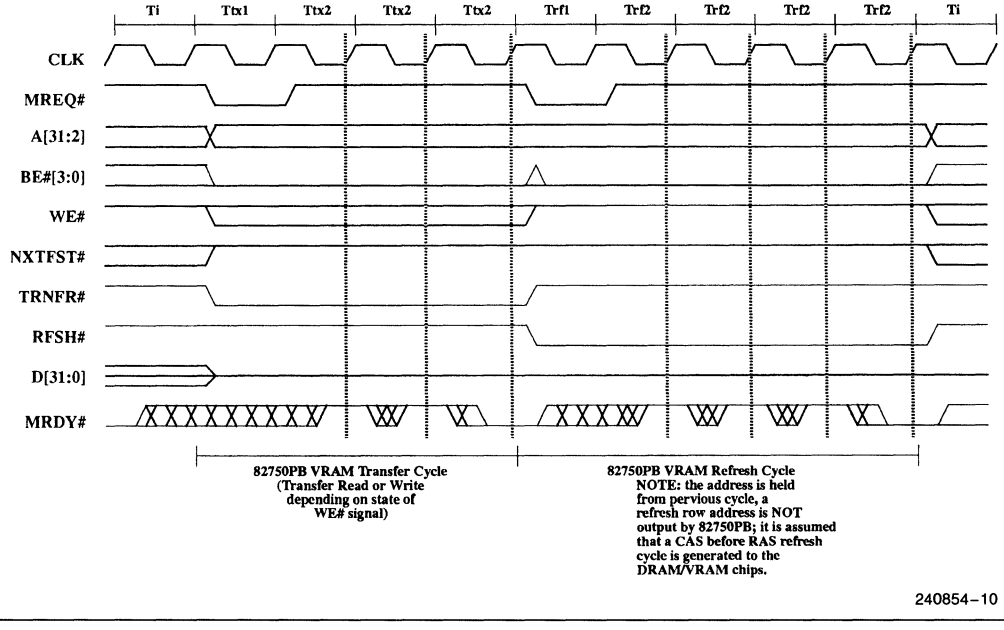


Figure 3-5. VRAM Transfer and Refresh Cycles

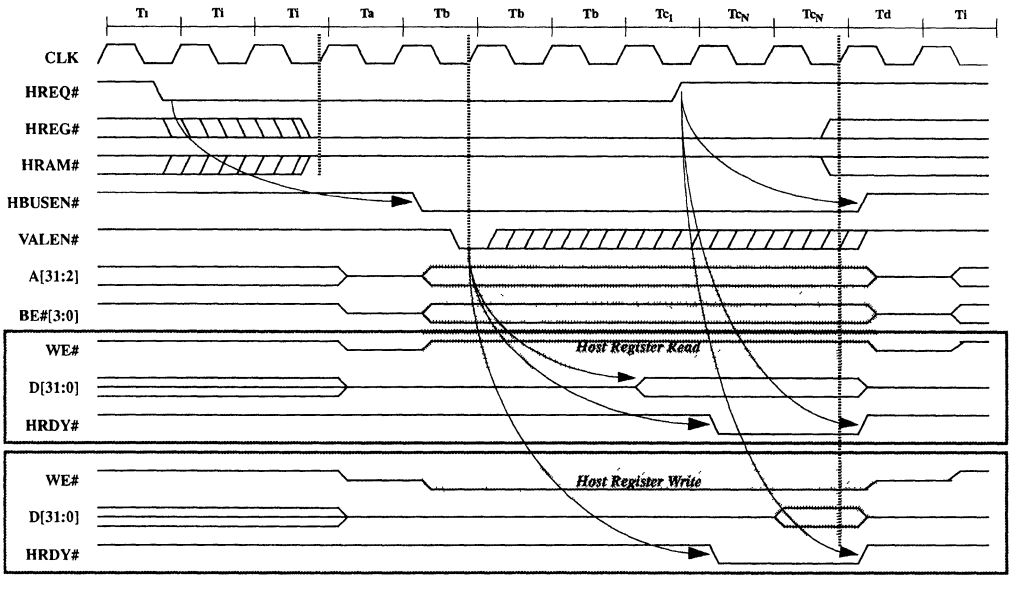
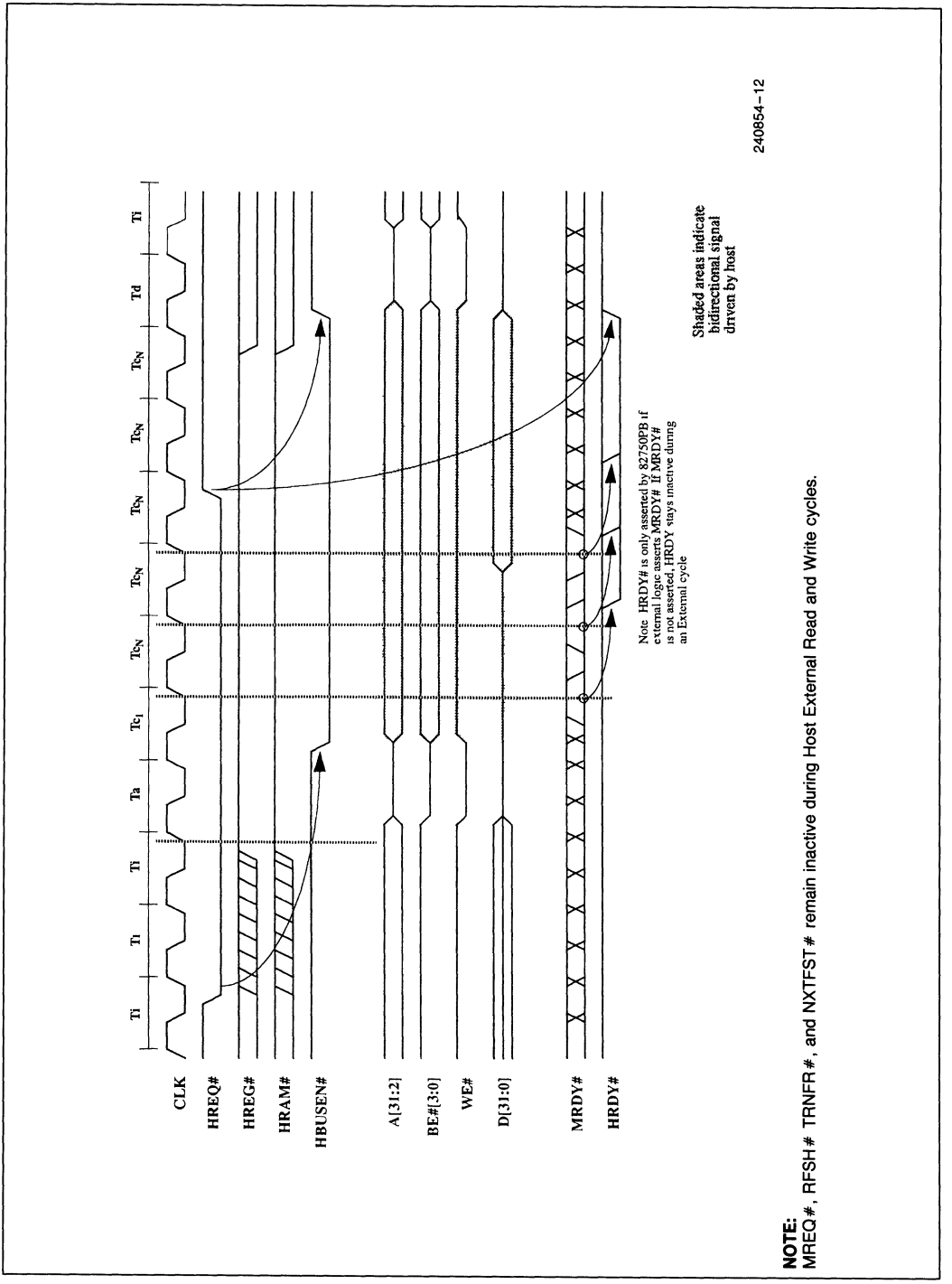
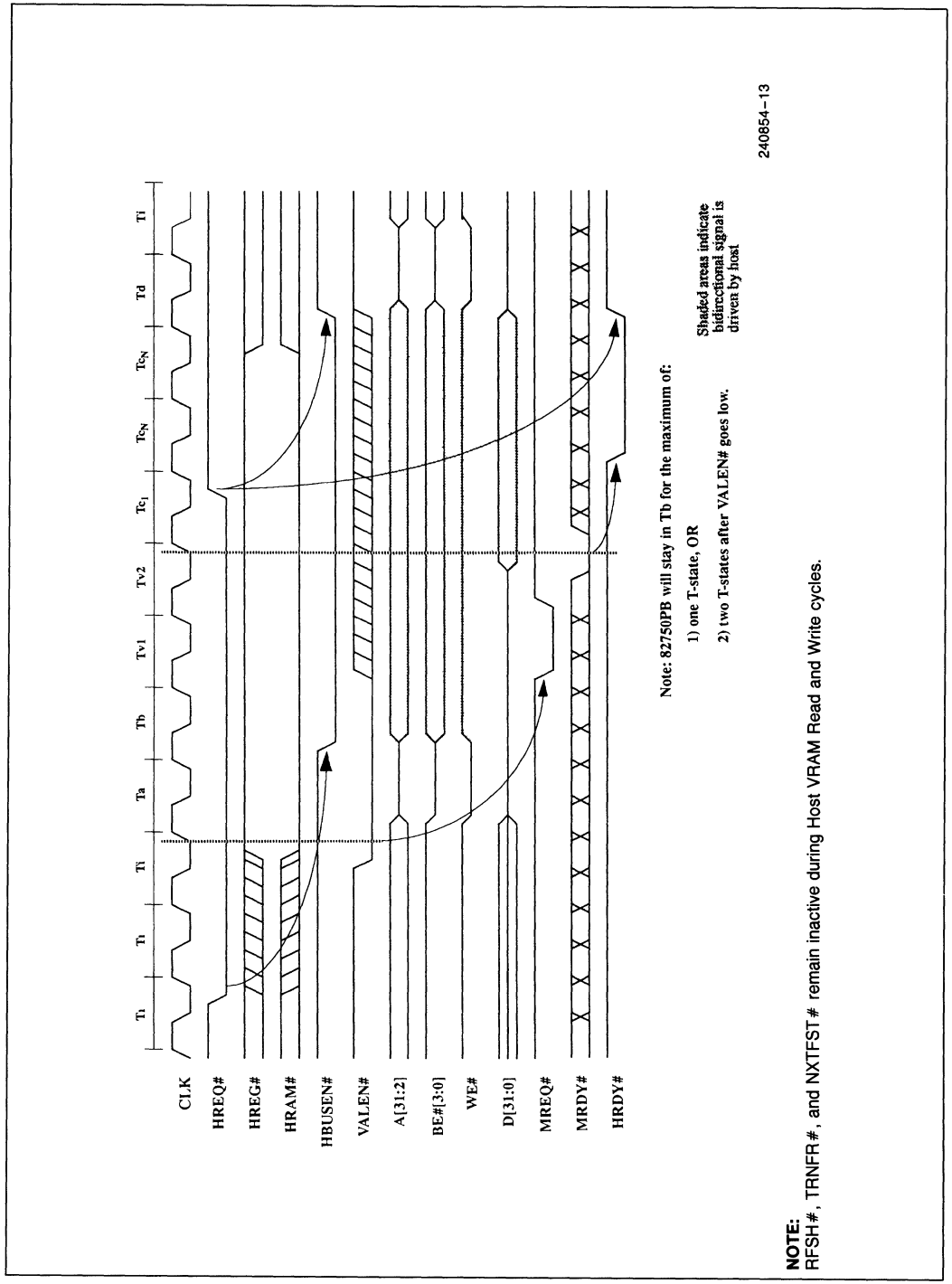


Figure 3-6. Host Register Read and Write Cycles



NOTE:
MREQ#, RFSH#, TRNFR#, and NXTFST# remain inactive during Host External Read and Write cycles.

Figure 3-7. Host External Read and Write Cycles



NOTE:
RFSH#, TRNFR#, and NXTFST# remain inactive during Host VRAM Read and Write cycles.

Figure 3-8. Host VRAM Read and Write Cycles

4.0 MICROCODE INSTRUCTION FORMAT

Overview

The 82750PB executes two slightly different instruction formats: one that is backward compatible with the 82750PA and another that allows full access to the microcode resources of the 82750PB. The 82750PA/82750PB bit in the 82750PB processor control register determines which instruction format is in effect (see Chapter 3). On reset, the 82750PB is placed in 82750PA instruction format mode. In this mode the 82750PB will execute binary microcode originally assembled for the 82750PA in a manner that is functionally equivalent to the 82750PA.

The following description applies to the 82750PB instruction format. Tables showing exact definitions of both the 82750PA and 82750B instruction formats and field codings can be found in Chapter 6.

Instruction Sequencing

The instruction word for 82750PB's microcode processor is 48 bits wide. The Microcode RAM holds 512 instructions. Nine bits of each instruction specify the address of the next instruction to be executed. Each instruction fetch reads two instructions (of odd address and even address pair) using the upper eight bits of the 9-bit instruction address. Both the LSB of the instruction address and a Condition Flag bit, selected from eight possible branching conditions, are used to determine whether the next instruction to be executed is the even address instruction or odd address instruction, according to the logic table shown as Table 4-1.

Table 4-1. Microcode Next Instruction Selection

LSB of Address	Condition Flag State	Next Instruction
0	0 (FALSE)	EVEN
0	1 (TRUE)	EVEN
1	0 (FALSE)	ODD
1	1 (TRUE)	EVEN

For an unconditional branch, the condition flag FALSE (which is always zero) is selected; this causes the LSB of the address to be passed through to select the next instruction: LSB = 0 selects EVEN and LSB = 1 selects ODD. This allows unconditional branching to any of the 512 instructions in the RAM. For a conditional branch, the LSB of the address is set to a one; this causes the state of the condition flag to select the next instruction: FALSE selects the ODD instruction and TRUE selects the EVEN instruction. Therefore, a conditional branch jumps to either the odd or even instruction of an odd/even pair depending on the state of the condition.

Instruction Word Field Descriptions

Each field of the microcode instruction format is described in the following sections.

NADDR—NEXT INSTRUCTION ADDRESS FIELD

This field holds the address of the next instruction to be executed. Taking advantage of the fact that the microcode RAM is physically organized as 256 deep by 96 wide (two instructions are fetched per read cycle), a zero delay two-way branch can be achieved. The only case in which this field is not used to determine the address of the next instruction to be executed is when an instruction writes to the PC. (The term PC refers to the register that holds the address of the next instruction to be executed.) When an instruction loads the PC a one instruction delay occurs before the load takes effect. Therefore, the instruction pointed to by the next instruction field of the instruction that loads the PC is executed before the jump to the new address occurs. This is shown in Table 4-2.

There are no restrictions on the instruction following a PC load; it will always be executed, even while single stepping the processor or if the processor is frozen on that instruction.

CFSEL—CONDITION FLAG SELECT FIELD

This field selects which condition flag will be used with the LSB of NADDR to select the next instruction from the odd/even pair. The condition flag assignment is given in Table 4-3.

Table 4-2. PC Load Example

Addr	Instruction	NADDR	Comments
10	pc = 0	55	Load PC with zero.
55	r0 = 1	X	This instruction is executed but its next address field is ignored.
0	r1 = r0	25	PC load takes effect after a one instructon delay, the result is that r1 = r0 = 1.

Table 4-3. Condition Flag Select Field Assignments

Value	Flag	Description
000	FALSE	Select for Unconditional Branch
001	CARRY	Carry Out from ALU Condition Flag Latch
010	OVF	Overflow from ALU Condition Flag Latch
011	SIGN	Sign from ALU Condition Flag Latch
100	ZERO	Zero from ALU Condition Flag Latch
101	LCNTZ	TRUE if Selected Loop Counter = 0
110	LSB	LSB of Data Register r0
111	MSB	MSB of Data Register r0

NOTE:
The ALU condition flags (CARRY, OVF, SIGN, and ZERO) are latched in the ALU Condition Flag register. This register is updated for most—but not all—ALU operations. The remaining flags (LCNTZ, LSB, and MSB) are updated and latched each cycle.

ASRC—A BUS SOURCE SELECT FIELD

This field selects the element that should drive its data onto the A bus during the execution of this instruction. The mapping for this and the following three fields is provided in Chapter 6.

ADST—A BUS DESTINATION SELECT FIELD

This field selects which element should latch data from the A bus during the execution of this instruction. See ASRC above.

BSRC—B BUS SOURCE SELECT FIELD

Same as ASRC, but for B bus. See ASRC above.

BDST—B BUS DESTINATION SELECT FIELD

Same as ADST, but for B bus. See ADST above.

CNT—DECREMENT LOOP COUNTER BIT

A one in this bit position causes the selected Loop Counter (selected by LC, the loop counter select bit) to be decremented. The new value of the loop counter and the updated LCNTZ condition flag are not ready until the next instruction cycle. Therefore, in a loop where the loop counter is decremented and tested for zero in the same instruction (typically in a one instruction loop), the start value for the loop counter should be one less than the number of times the loop should be executed.

LIT—LITERAL SELECT BIT

When this bit is a one, the ASRC and CFSEL fields are replaced with a 9-bit literal value that is driven as a source in the least significant 9 bits of the A bus. In this case, the upper 7 bits of the A bus are forced to zeros. The mapping of bits from the literal field to the A bus is shown in Figure 4-1.

NOTE:
A conditional branch and a literal on the A bus are not allowed in the same instruction. A 3-bit literal can be placed on the B bus in any instruction. Refer to the A/B bus Source/Destination table in the Chapter 6.

A bus bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Inst. Word Bits	←	- - - - -	Forced to Zero	- - - -	→			17	16	15	14	13	12	11	10	9
ASRC Field								←	- - - - - - - - - - - - - - - -				→			
CFSEL Field														←	- -	→

Figure 4-1. Literal Field Mapping onto a Bus

SHFT—SHIFT CONTROL FIELD

This field controls the bit shifting and byte swapping logic associated with register *r0*. The encoding of this field is given in Table 4-4.

Table 4-4. SHIFT Control Field Coding

SHFT	Operation
00	No Shift or Swap Operation
01	Shift r0 Right One Bit Position, Sign Extend
10	Shift r0 Left One Bit Position, Zero Fill
11	Byte Swap the Value Being Loaded into r0*

NOTE:
*Byte swapping only works when r0 is the destination on the A bus or the B bus. It does not swap data held in r0, only data being loaded. In order to byte swap data in register r0, r0 must be both a source and destination for either the A or B bus.

ALUSS—ALU SOURCE SELECT BITS

These two bits are used as enables for the two ALU input latches. Bit 39 enables the latch that connects to the A bus; bit 38 enables the latch connected to the B bus. A one in either bit position causes the corresponding input latch to latch the value on the bus to which it is connected (the A or B bus). A zero on either bit causes the corresponding latch to hold its current content. This allows the ALU operands either to come from "eavesdropping" on the A or B bus transfers occurring in the current instruction cycle or to be held for multiple instruction cycles in either the A or B input latch.

ALUOP—ALU OPERATION CODE FIELD

This field specifies the ALU instruction to be performed during the current instruction cycle. The encoding of this field is given in the ALU OpCodes Coding table in Chapter 6. Normally, at the end of the instruction execution, the result of the ALU operation is latched in the ALU output latch that can be a source on either the A or B buses. However, if a NOP is selected for the ALU operation, the ALU output latch is not latched. The data is held from the previous instruction. In addition to NOP, certain other ALU opcodes do not actually perform ALU operations and therefore, do not latch the ALU results. They are INT (microcode interrupt) and the PROF instruction.

LC—LOOP COUNTER SELECT BIT

This bit selects which of the two loop counters is to be used for decrementing or Loop-Counter-Zero conditional branching in the current instruction. A zero selects loop counter zero and a one selects loop counter one.

5.0 HOW MICROCODE EXECUTES

DVI™ Technology Microcode Execution

Figure 5-1 shows the processes involved when DVI Technology system software executes an 82750PB microcode function.

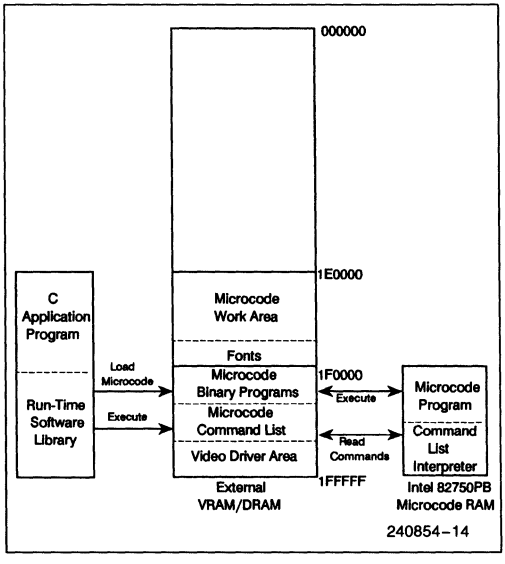


Figure 5-1. Execution of an Intel 82750PB Microcode Function

Microcode binary (.BIN) files initially reside on disk. When a microcode function on disk is needed, it is loaded from disk into a section of the system-reserved VRAM (by default, the high 128K of VRAM, from addresses 1E0000 to 1FFFFF on a two-megabyte system). To execute a microcode function that has been loaded, system software writes a message into the *microcode command list* area of VRAM. That message contains the address in VRAM of the microcode function, as well as parameters to be passed to it.

The Command List Interpreter

The microcode command list permits asynchronous operation of the 82750PB and the host processor. The 82750PB contains, at all times, a small microcode program called the *command list interpreter*, which controls the overall execution of microcode functions from the application. When the 82750PB finishes the execution of a function, the command list interpreter looks at the command list in VRAM to see if there are any pending requests. If so, it takes the next one from the list and executes it. Meanwhile, the host processor can be adding additional commands to the command list. A handshaking protocol is employed to prevent race conditions in which the 82750PB and the host processor both attempt to access the command list at the same time.

Refer to the *Intel 82750PB Microcode Programming Guide* for more information on programming the 82750PB.

6.0 DIFFERENCES WITH THE 82750PA

82750PA/82750PB Compatibility Issues

The functionality differences between the 82750PA and the 82750PB that are relevant to the microcode programmer are listed in Table 6-1.

There are several functions that can be performed with the 82750PA that are not legal or do not have exactly the same effect on the 82750PB. The microcode assembler, VASM, enforces compatibility where possible. However, there are several items that the assembler cannot catch. They are as follows:

- Do not rely on the width of the **dram** pointers; i.e., do not write a value to a **dram** pointer and then read it back to get a "free" masking of the upper eight bits. In the 82750PB, the same operation will only mask the upper seven bits. Similarly, do not rely on the **dram** pointers wrapping around from 0xFF to 0x00 or vice versa.
- Do not rely on the width of **pc** or **maddr**, i.e., do not write a 16-bit value to **maddr** expecting the upper 8 bits to be truncated to zeros. In the 82750PB, only the upper seven bits would be truncated.

Table 6-1. 82750PB/82750PA Functionality Differences

Function	82750PB	82750PA
General-Purpose Registers	16	8
DRAM Words	512	256
Microcode RAM Instructions	512	256
A Bus DRAM Pointers	1, 2, 3, 4	1, 2
B Bus DRAM Pointers	1, 2, 3, 4	3, 4
FIFO I/O	A and B Buses	B Bus Only
Literal Size	9 Bits	8 Bits

82750PA and 82750PB Source/Destination Coding

Tables 6-2 and 6-3 give the source and destination coding for the 82750PA and the 82750PB.

Table 6-2. 82750PA Source/Destination Coding

Address (Hex)	BDST	BSRC	ADST	ASRC
0x0	Null	Null	Null	Null
0x1		alu		hwid
0x2	*dram3	*dram3		cc
0x3	*dram4	*dram4	maddr	
0x4	*dram3++	*dram3++		alu
0x5	*dram4++	*dram4++	cnt	cnt
0x6	*dram3--	*dram3--	cnt2	cnt2
0x7	*dram4--	*dram4--	lcnt	lcnt
0x8	r0	r0	r0	r0
0x9	r1	r1	r1	r1
0xA	r2	r2	r2	r2
0xB	r3	r3	r3	r3
0xC	r4	r4	r4	r4
0xD	r5	r5	r5	r5
0xE	r6	r6	r6	r6
0xF	r7	r7	r7	r7
0x10		*in1	mcode3	mcode3
0x11		*in2	mcode2	mcode2
0x12		*stat	mcode1	mcode1
0x13		*stat#	pc	pc
0x14			pixint-c	
0x15			pixint	pixint
0x16			*dram1	*dram1
0x17			*dram2	*dram2
0x18		literal 0	*dram1++	*dram1++
0x19		literal 1	*dram2++	*dram2++
0x1A		literal 2	*dram1--	*dram1--
0x1B		literal 3	*dram2--	*dram2--
0x1C		literal 4	dram1	dram1

Table 6-2. 82750PA Source/Destination Coding (Continued)

Address (Hex)	BDST	BSRC	ADST	ASRC
0x1D		literal 5	dram2	dram2
0x1E		literal 6	dram3	dram3
0x1F		literal 7	dram4	dram4
0x20	*out1			
0x21	out1 + +			
0x22	out1-lo			
0x23	out1-hi			
0x24	*out2			
0x25	out2 + +			
0x26	out2-lo			
0x27	out2-hi			
0x28	out1-c			
0x29	in1-c			
0x2A	in1-lo			
0x2B	in1-hi			
0x2C	out2-c			
0x2D	in2-c			
0x2E	in2-lo			
0x2F	in2-hi			
0x30	stat-ram			
0x31	stat-c			
0x32	stat-lo			
0x33	stat-hi			
0x34	yeven-lo			
0x35	yeven-hi			
0x36	yodd-lo			
0x37	yodd-hi			
0x38	ypitch			
0x39	reserved			
0x3A	vu-lo			
0x3B	vu-hi			
0x3C	vupitch			
0x3D	vpitch			
0x3E	vptr-lo			
0x3F	vptr-hi			

Table 6-3. 82750PB Source/Destination Coding

Address (Hex)	BDST	BSRC	ADST	ASRC
0x0	Null	Null	Null	Null
0x1		alu		hwid
0x2	*dram3	*dram3		cc
0x3	*dram4	*dram4	maddr	
0x4	*dram3++	*dram3++		alu
0x5	*dram4++	*dram4++	cnt	cnt
0x6	*dram3--	*dram3--	cnt2	cnt2
0x7	*dram4--	*dram4--	lcnt	lcnt
0x8	r0	r0	r0	r0
0x9	r1	r1	r1	r1
0xA	r2	r2	r2	r2
0xB	r3	r3	r3	r3
0xC	r4	r4	r4	r4
0xD	r5	r5	r5	r5
0xE	r6	r6	r6	r6
0xF	r7	r7	r7	r7
0x10	r8	*in1	mcode3	mcode3
0x11	r9	*in1	mcode2	mcode2
0x12	r10	*stat	mcode1	mcode1
0x13	r11	*stat#	pc	pc
0x14	r12		pixint-c	
0x15	r13		pixint	pixint
0x16	r14		*dram1	*dram1
0x17	r15		*dram2	*dram2
0x18		literal 0	*dram1++	*dram1++
0x19		literal 1	*dram2++	*dram2++
0x1A	*dram1	literal 2	*dram1--	*dram1--
0x1B	*dram2	literal 3	*dram2--	*dram2--
0x1C	*dram1++	literal 4	dram1	dram1
0x1D	*dram2++	literal 5	dram2	dram2
0x1E	*dram1--	literal 6	dram3	dram3
0x1F	*dram2--	literal 7	dram4	dram4
0x20	*out1	prof	*out1	*in1

Table 6-3. 82750PB Source/Destination Coding (Continued)

Address (Hex)	BDST	BSRC	ADST	ASRC
0x21	out1 + +		out1 + +	*in2
0x22	out1-lo	out1-lo	shift-rl	*stat
0x23	out1-hi	out1-hi	out1-hi	*stat#
0x24	*out2	stat-low	*out2	
0x25	out2 + +	stat-hi	out2 + +	
0x26	out2-lo	out2-lo	shift-r	
0x27	out2-hi	out2-hi	out2-hi	
0x28	out1-c	out1-c	out1-c	
0x29	in1-c	in1-c	in1-c	
0x2A	in1-lo	in1-lo	shift-l	
0x2B	in1-hi	in1-hi	in1-hi	
0x2C	out2-c	out2-c	out2-c	
0x2D	in2-c	in2-c	in2-c	
0x2E	in2-lo	in2-lo		
0x2F	in2-hi	in2-hi	in2-hi	
0x30	stat-ram	r8	r8	r8
0x31	stat-c	r9	r9	r9
0x32	stat-lo	r10	r10	r10
0x33	stat-hi	r11	r11	r11
0x34	yeven-lo	r12	r12	r12
0x35	yeven-hi	r13	r13	r13
0x36	yodd-lo	r14	r14	r14
0x37	yodd-hi	r15	r15	r15
0x38	ypitch	shift	cc	shift
0x39		stat-c		
0x3A	vu-lo	*dram1	*dram3	*dram3
0x3B	vu-hi	*dram2	*dram4	*dram4
0x3C	vupitch	*dram1 + +	*dram3 + +	*dram3 + +
0x3D	vpitch	*dram2 + +	*dram4 + +	*dram4 + +
0x3E	vptr-lo	*dram1 − −	*dram3 − −	*dram3 − −
0x3F	vptr-hi	*dram2 − −	*dram4 − −	*dram4 − −

82750PA and 82750PB Instruction Word Formats

Figures 6-4 and 6-5 show the instruction word formats for the 82750PA and the 82750PB.

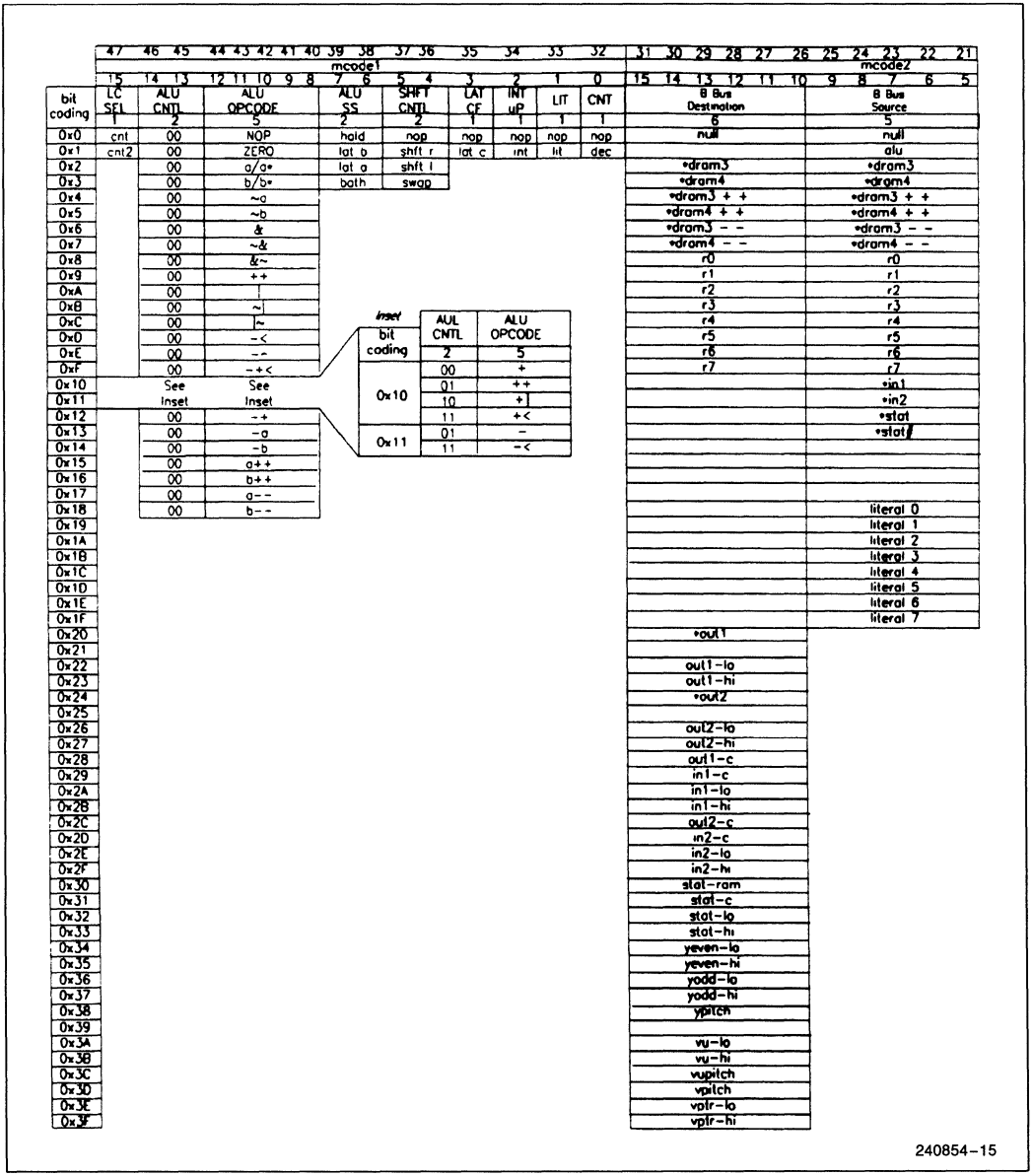


Figure 6-1. 82750PA Instruction Word Format

	20 19 18 17 16	15 14 13 12 11	10 9 8	7 6 5 4 3 2 1 0
	mcode 2	mcode 3		
	4 3 2 1 0	15 14 13 12 11	10 9 8	7 6 5 4 3 2 1 0
bit coding	A Bus Destination	A Bus Source	Cond Flag Select	Next Address
	5	5	3	8
0x0	null	null	FALSE	
0x1		hwid	CARRY	
0x2		cc	OVERFLOW	
0x3	moddr		SIGN	
0x4		alu	ZERO	
0x5	cnt	cnt	CNTO	
0x6	cnt2	cnt2	LSB r0	
0x7	lcnt	lcnt	MSB r0	
0x8	r0	r0		
0x9	r1	r1		
0xA	r2	r2		
0xB	r3	r3		
0xC	r4	r4		
0xD	r5	r5		
0xE	r6	r6		
0xF	r7	r7		
0x10	mcode 3	mcode 3		
0x11	mcode 2	mcode 2		
0x12	mcode 1	mcode 1		
0x13	pc	pc		
0x14	pixint-c			
0x15	pixint	pixint		
0x16	*dram1	*dram1		
0x17	*dram2	*dram2		
0x18	*dram1 + +	*dram1 + +		
0x19	*dram2 + +	*dram2 + +		
0x1A	*dram1 − −	*dram1 − −		
0x1B	*dram2 − −	*dram2 − −		
0x1C	dram1	dram1		
0x1D	dram2	dram2		
0x1E	dram3	dram3		
0x1F	dram4	dram4		
0x20				
0x21				
0x22				
0x23				
0x24				
0x25				
0x26				
0x27				
0x28				
0x29				
0x2A				
0x2B				
0x2C				
0x2D				
0x2E				
0x2F				
0x30				
0x31				
0x32				
0x33				
0x34				
0x35				
0x36				
0x37				
0x38				
0x39				
0x3A				
0x3B				
0x3C				
0x3D				
0x3E				
0x3F				

Figure 6-1. 82750PA Instruction Word Format (Continued)

	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24
	mcode 1																mcode2							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
bit coding	LC SEL	SHFT CNTL		ALU OPCODE					ALU SS		LIT	CNT	B Bus Destination				B Bus Source							
	1	2		5					2		1	1	6				6							
0x0	cnt	nop		NOP					hold		nop	nop	null				null							
0x1	cnt2	shft r		ZERO					lat b		lit	dec					alu							
0x2		shft l		a					lat a															
0x3		swap		b					both															
0x4				~a																				
0x5				~b																				
0x6				&																				
0x7				~&																				
0x8				&~																				
0x9				+ +																				
0xA				|																			
0xB				~ |																			
0xC				| ~																			
0xD				- <																				
0xE				-																				
0xF				- + <																				
0x10				+																				
0x11				-																				
0x12				- +																				
0x13				- a																				
0x14				- b																				
0x15				a + +																				
0x16				b + +																				
0x17				a - -																				
0x18				b - -																				
0x19				int																				
0x1A				prof																				
0x1B				a*																				
0x1C				b*																				
0x1D				+ <																				
0x1E				+]																				
0x1F				-]																				
0x20																								
0x21																								
0x22																								
0x23																								
0x24																								
0x25																								
0x26																								
0x27																								
0x28																								
0x29																								
0x2A																								
0x2B																								
0x2C																								
0x2D																								
0x2E																								
0x2F																								
0x30																								
0x31																								
0x32																								
0x33																								
0x34																								
0x35																								
0x36																								
0x37																								
0x38																								
0x39																								
0x3A																								
0x3B																								
0x3C																								
0x3D																								
0x3E																								
0x3F																								

	B Bus Destination	B Bus Source
0x0	null	null
0x1	*dram3	*dram3
0x2	*dram4	*dram4
0x3	*dram3 + +	*dram3 + +
0x4	*dram4 + +	*dram4 + +
0x5	*dram3 - -	*dram3 - -
0x6	*dram4 - -	*dram4 - -
0x7	r0	r0
0x8	r1	r1
0x9	r2	r2
0xA	r3	r3
0xB	r4	r4
0xC	r5	r5
0xD	r6	r6
0xE	r7	r7
0xF	r8	*in1
0x10	r9	*in2
0x11	r10	*stat
0x12	r11	*stat#
0x13	r12	
0x14	r13	
0x15	r14	
0x16	r15	
0x17		literal 0
0x18		literal 1
0x19	*dram1	literal 2
0x1A	*dram2	literal 3
0x1B	*dram1 + +	literal 4
0x1C	*dram2 + +	literal 5
0x1D	*dram1 - -	literal 6
0x1E	*dram2 - -	literal 7
0x1F	*out1	prof
0x20	out1 + +	
0x21	out1 - lo	out1 - lo
0x22	out1 - hi	out1 - hi
0x23	*out2	stat-lo
0x24	out2 + +	stat-hi
0x25	out2 - lo	out2 - lo
0x26	out2 - hi	out2 - hi
0x27	out1 - c	out1 - c
0x28	in1 - c	in1 - c
0x29	in1 - lo	in1 - lo
0x2A	in1 - hi	out2 - hi
0x2B	out2 - c	out2 - c
0x2C	in2 - c	in2 - c
0x2D	in2 - lo	in2 - lo
0x2E	in2 - hi	in2 - hi
0x2F	stat - ram	r8
0x30	stat - c	r9
0x31	stat - lo	r10
0x32	stat - hi	r11
0x33	yeven - lo	r12
0x34	yeven - hi	r13
0x35	yodd - lo	r14
0x36	yodd - hi	r15
0x37	ypitch	shift
0x38		stat - c
0x39	vu - lo	*dram1
0x3A	vu - hi	*dram2
0x3B	vupitch	*dram1 + +
0x3C	vpitch	*dram2 + +
0x3D	vptr - lo	*dram1 - -
0x3E	vptr - hi	*dram2 - -
0x3F		

Figure 6-2. 82750PB Instruction Word Format (Continued)

	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	mcode 2								mcode 3															
	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

bit coding	A Bus Designation	A Bus Source	Cond Flag Select	Next Address
	6	6	3	9
0x0	null	null	FALSE	
0x1		hwid	CARRY	
0x2		cc	OVERFLOW	
0x3	moddr		SIGN	
0x4		alu	ZERO	
0x5	cnt	cnt	CNT0	
0x6	cnt2	cnt2	LSB r0	
0x7	lcnt	lcnt	MSB r0	
0x8	r0	r0		
0x9	r1	r1		
0xA	r2	r2		
0xB	r3	r3		
0xC	r4	r4		
0xD	r5	r5		
0xE	r6	r6		
0xF	r7	r7		
0x10	mcode3	mcode3		
0x11	mcode2	mcode2		
0x12	mcode1	mcode1		
0x13	pc	pc		
0x14	pixint − c			
0x15	pixint	pixint		
0x16	*dram1	*dram1		
0x17	*dram2	*dram2		
0x18	*dram1 + +	+ dram1 + +		
0x19	*dram2 + +	+ dram2 + +		
0x1A	*dram1 − −	+ dram1 − −		
0x1B	*dram2 − −	+ dram2 − −		
0x1C	dram1	dram1		
0x1D	dram2	dram2		
0x1E	dram3	dram3		
0x1F	dram4	dram4		
0x20	*out1	*in1		
0x21	out1 + +	*in2		
0x22	shift − rl	*stat		
0x23	out1 − hi	*stat#		
0x24	*out2			
0x25	out2 + +			
0x26	shift − r			
0x27	out2 − hi			
0x28	out1 − c			
0x29	in1 − c			
0x2A	shift − 1			
0x2B	in1 − hi			
0x2C	out2 − c			
0x2D	in2 − c			
0x2E				
0x2F	in2 − hi			
0x30	r8	r8		
0x31	r9	r9		
0x32	r10	r10		
0x33	r11	r11		
0x34	r12	r12		
0x35	r13	r13		
0x36	r14	r14		
0x37	r15	r15		
0x38	cc	shift		
0x39				
0x3A	*dram3	*dram3		
0x3B	*dram4	*dram4		
0x3C	*dram3 + +	*dram3 + +		
0x3D	*dram4 + +	*dram4 + +		
0x3E	*dram3 − −	*dram3 − −		
0x3F	*dram4 − −	*dram4 − −		

Figure 6-2. 82750PB Instruction Word Format (Continued)

7.0 ELECTRICAL DATA

D.C. Characteristics

Table 7-1. Absolute Maximum Requirements

Condition	Maximum Requirement
Case Temperature under Bias	−65°C to +110°C
Storage Temperature	−65°C to +150°C
Voltage on Any Pin with Respect to Ground	−0.5V to V_{CC} + 0.5V
Supply Voltage with Respect to V_{SS}	−0.5V to +6.5V

Table 7-2. D.C. Characteristics (V_{CC} = 5V ±10%, T_{CASE} = 0°C to +95°C)

Symbol	Parameter	Min	Typ	Max	Unit	Notes
V_{IL}	Input LOW Voltage	−0.3		0.8	V	(Note 1)
V_{IH}	Input HIGH Voltage	2.0		V_{CC} + 0.3	V	(Note 1)
V_{OL}	Output LOW Voltage		0.2	0.4	V	I_{OL} = 4.0 mA(1)
V_{OH}	Output HIGH Voltage	2.4	3.0		V	I_{OH} = −1.0 mA(1)
I_{IL}	Input Leakage Current	−10		+10	μA	$V_{SS} < V_{IN} < V_{CC}$
I_{OZ}	Output Leakage Current	−10		+10	μA	$V_{SS} < V_{IN} < V_{CC}$
I_{CC}	Power Supply Current		185	250	mA	25 MHz(2)
C_{IN}	Input Capacitance			10.0	pF	F_C = 1 MHz(3)
C_{OUT}	Output Capacitance			12.0	pF	F_C = 1 MHz(3)
C_{CLKIN}	CLKIN Input Capacitance			20.0	pF	F_C = 1 MHz(3)

NOTES:
1. Measured with CLKIN = 8 MHz.
2. Typical current value measured under typical conditions. Maximum current value guaranteed with 50 pF maximum output loading.
3. Not 100% tested.

A.C. Characteristics

Table 7-3. A.C. Characteristics at 25 MHz V_{CC} = 5V ±10%; T_{CASE} = 0°C to +95°C; C_L = 50 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
	Frequency	8	25	MHz		1xClock
t_1	CLKIN Period	40	125	ns	7-1	
t_2	CLKIN High Time	14		ns	7-1	
t_3	CLKIN Low Time	14		ns	7-1	
t_4	CLKIN Fall Time		4	ns	7-1	
t_5	CLKIN Rise Time		4	ns	7-1	
t_{6a}	A[31:2], BE#[3:0] D[31:0], HBUSEN#, HINT#, HRDY#, PMFRZ# Valid Delay	3	25	ns	7-2	
t_{6b}	MREQ#, TRNFR#, RFSH#, NXTFST#, WE# Valid Delay	3			7-2	
t_7	A[31:2], BE#[3:0], WE#, D[31:0] Float Delay		30	ns	7-2	(Note 1)
t_8	MRDY# Setup	10		ns	7-3	
t_9	MRDY# Hold	6		ns	7-3	
t_{10}	MREQ#, VBUS[3:0], RESET#, HALEN# Setup	8		ns	7-3	
t_{11}	HREQ#, VBUS[3:0], RESET#, HALEN# Hold	6		ns	7-3	
t_{12}	A[8:2], BE#[3:0], WE#, D[31:0] Setup	8		ns	7-3	(Note 2)
t_{13}	A[8:2], BE#[3:0], WE#, D[31:0] Hold	6		ns	7-3	(Note 2)
t_{14}	HREG#, HRAM# Setup	10		ns	7-3	
t_{15}	HREG#, HRAM# Hold	6		ns	7-3	
t_{16}	HALT# Setup	8		ns	7-3	
t_{17}	HALT# Hold	6		ns	7-3	

NOTES:
1. Not 100% tested. Guaranteed by design characterization.
2. Inputs must remain valid throughout all cycles of host accesses. See Figures 3-6 through 3-8.
3. All A.C. specifications are measured at the 1.5V crossing point with a 50 pF load.

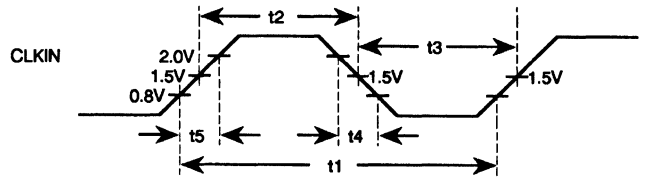


Figure 7-1. Clock Waveforms

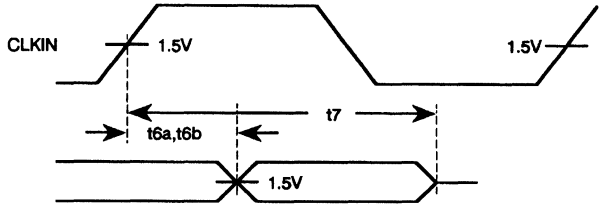
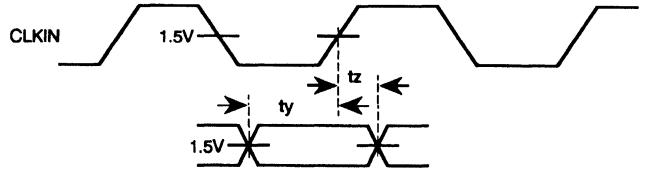


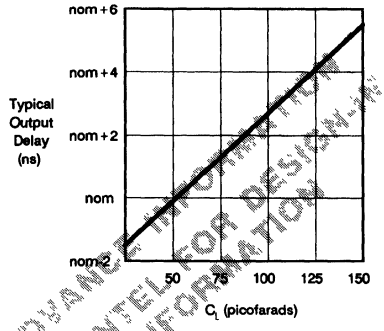
Figure 7-2. Output Waveforms



NOTES:
ty = t8, t10, t12, t14, t16 (setup times)
tz = t9, t11, t13, t15, t17 (hold times)

Figure 7-3. Input Waveforms

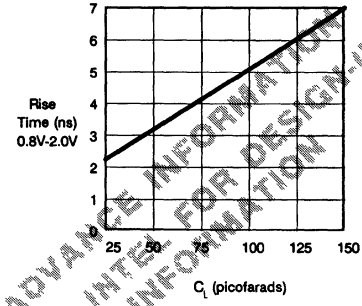
Output Delay and Rise Time Versus Load Capacitance



240854-22

NOTE:
This graph will not be linear outside of the C_L range shown.
nom = nominal value given in A.C. Characteristics table.

Figure 7-4. Typical Output Valid Delay Versus Load Capacitance under Worst Case Conditions



240854-23

NOTE:
This graph will not be linear outside of the C_L range shown.

Figure 7-5. Typical Output Rise Time Versus Load Capacitance under Worst Case Conditions

8.0 MECHANICAL DATA

Packaging Outlines and Dimensions

Intel packages the 82750PB in a Plastic Quad Flat Pack (PQFP). Table 8-1 gives the symbol list for the PQFP.

Table 8-1. PQFP Symbol List

Letter or Symbol	Description of Dimensions
A	Package Height: Distance from Seating Plane to Highest Point of Body
A_1	Standoff: Distance from Seating Plane to Base Plane
D/E	Overall Package Dimension: Lead Tip to Lead Tip
D_1/E_1	Plastic Body Dimension
D_2/E_2	Bumper Distance
D_3/E_3	Footprint
L_1	Foot Length
N	Total Number of Leads

The PQFP has the following specifications:

1. All dimensions and tolerances conform to ANSI Y14.5M-1982.
2. Datum plane —H— is located at the mold parting line and coincident with the bottom of the lead where lead exits plastic body.
3. Datums A-B and —D— are to be determined where center leads exit plastic body at datum plane —H—.
4. Controlling dimension is the inch.
5. Dimensions D_1, D_2, E_1, and E_2 are measured at the mold parting line and do not include mold protrusion. Allowable mold protrusion is 0.18 mm (0.007 in.) per side.
6. Pin 1 identifier is located within one of the two zones indicated.
7. Measured at datum plane —H—.
8. Measured at seating plane datum —C—.

Table 8-2 provides outline characteristics for 0.025 in. pitch.

Table 8-2. Intel Case Outline Drawings for PQFP at 0.025 inch Pitch

Symbol	Description	Min	Max
N	Leadcount	132	132
A	Package Height	0.160	0.170
A_1	Standoff	0.020	0.030
D, E	Terminal Dimiension	1.075	1.085
D_1, E_1	Package Body	0.0947	0.953
D_2, E_2	Bumper Distance	1.097	1.103
D_3, E_3	Lead Dimension	0.800 REF	0.800 REF
L_1	Foot Length	0.020	0.030

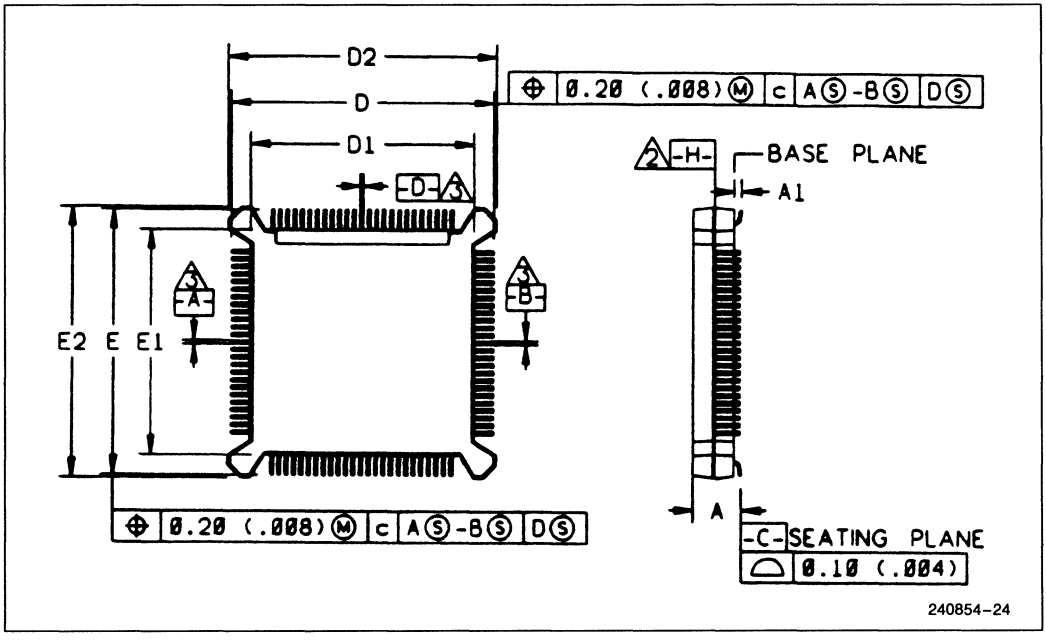


Figure 8-1. Principal Dimensions and Datums

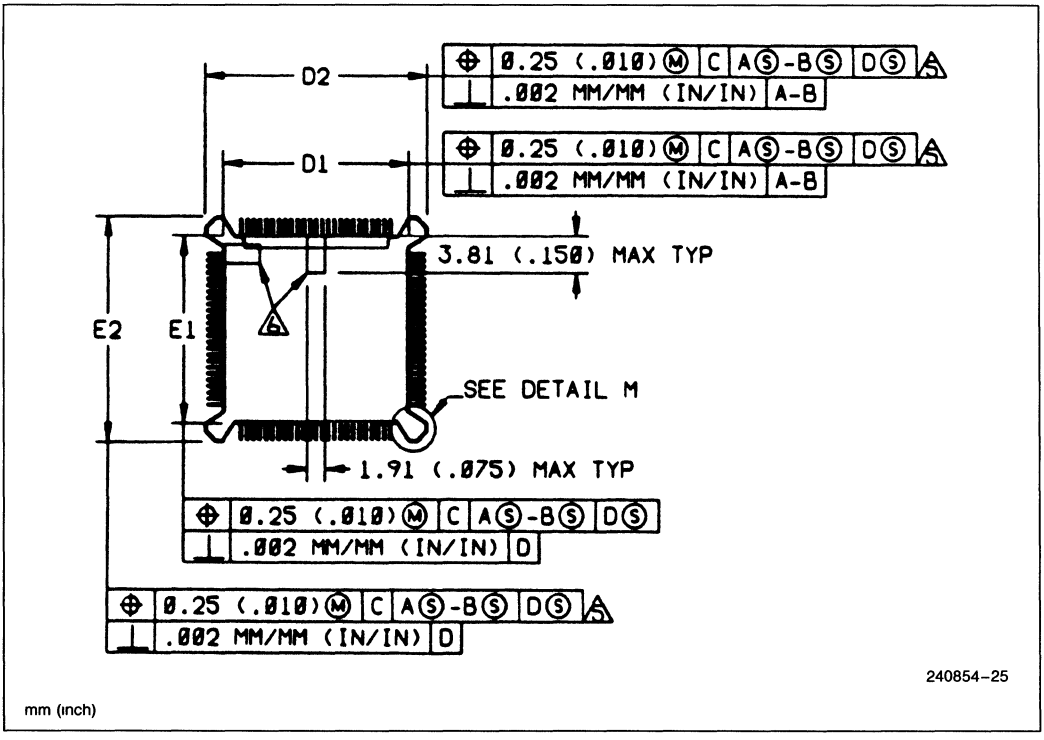


Figure 8-2. Molding Details

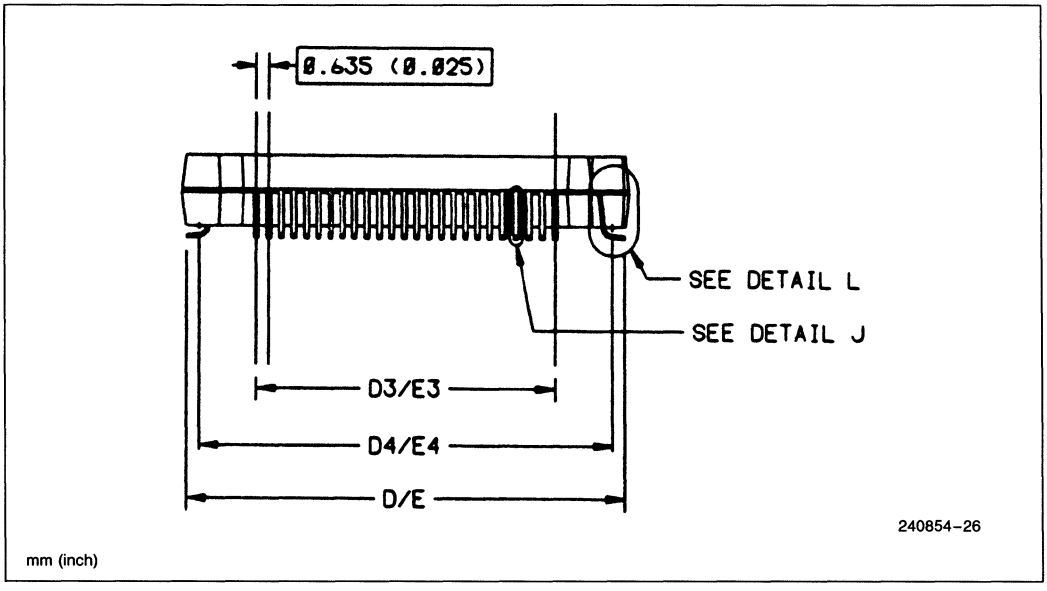
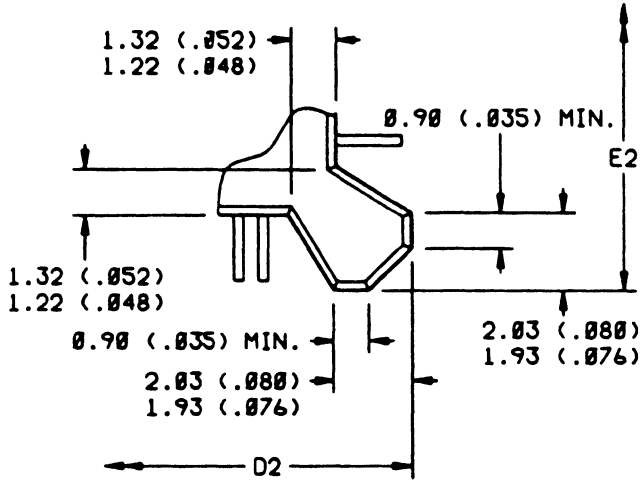
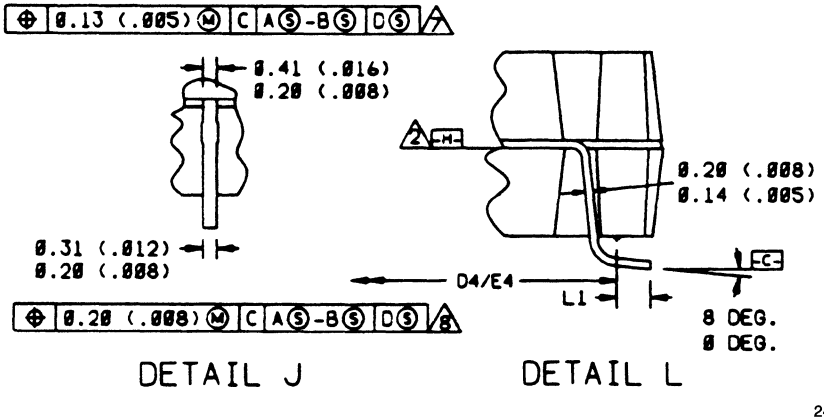


Figure 8-3. Terminal Details



mm (inch)

Figure 8-4. Typical Lead



mm (inch)

Figure 8-5. Bumper (Detail M)

NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M-1982
2. DATUM PLANE -H- LOCATED AT THE MOLD PARTING LINE AND COINCIDENT WITH THE BOTTOM OF THE LEAD WHERE LEAD EXITS PLASTIC BODY
3. DATUMS A-B AND -D- TO BE DETERMINED WHERE CENTER LEADS EXIT PLASTIC BODY AT DATUM PLANE -H-
4. CONTROLLING DIMENSION, INCH
5. DIMENSIONS D1, D2, E1 AND E2 ARE MEASURED AT THE MOLD PARTING LINE. D1 AND E1 DO NOT INCLUDE AN ALLOWABLE MOLD PROTRUSION OF 0.18 MM (.007 IN) PER SIDE. D2 AND E2 DO NOT INCLUDE A TOTAL ALLOWABLE MOLD PROTRUSION OF 0.18 MM (.007 IN) AT MAXIMUM PACKAGE SIZE.
6. PIN 1 IDENTIFIER IS LOCATED WITHIN ONE OF THE TWO ZONES INDICATED
7. MEASURED AT DATUM PLANE -H-
8. MEASURED AT SEATING PLANE DATUM -C-

240854-29

Package Thermal Specifications

The 82750PB is specified for operation when T_C (the case temperature) is within the range of 0°C to 95°C. T_C may be measured in any environment to determine whether the 82750PB is within specified operation range. The case temperature should be measured at the center of the top surface.

T_A (the ambient temperature) can be calculated from θ_{CA} (thermal resistance from case to ambient) with the following equation:

$$T_A = T_C - P^* \theta_{CA}$$

Typical values for θ_{CA} at various airflows are given in Table 8-3 for the 132-lead PQFP package. Table 8-4 shows the maximum T_A allowable (wihout exceeding T_C) at various airflows. The power dissipation (P) is calculaed by using the typical supply current at 5V as shown in Table 7-2.

Table 8-3. Thermal Resistance (°C/W)

Package	θ_{CA} Versus Airflow—ft/min (m/sec)					
	0 (0)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)	1000 (5.07)
132-Lead PQFP	26.0	17.5	14.0	11.5	9.5	8.5

Table 8-4. Maximum T_A at Various Airflows (°C)

Package		T_A Versus Airflow—ft/min (m/sec)					
	Frequency (MHz)	0 (0)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)	1000 (5.07)
132-Lead PQFP	25	71	79	82	84	86	87

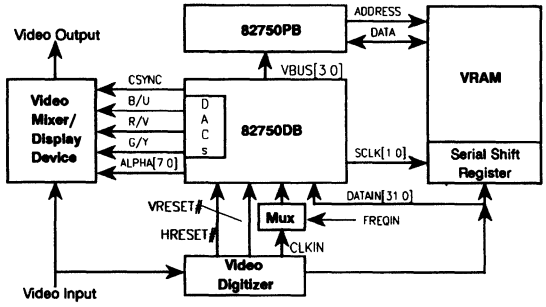
82750DB
DISPLAY PROCESSOR

- **Programmable Video Timing**
  - **— 28 MHz Operating Frequency**
  - **— Pixel/Line Address Range to 4096**
  - **— Fully Programmable Sync, Equalization, and Serration Components**
  - **— Fully Programmable Blanking and Active Display Start and Stop Times**
  - **— Genlocking Capability**
- **Flexible Display Characteristics**
  - **— 8-, Pseudo 16-, 16-, and 32-Bit/Pixel Modes**
  - **— Selectable Pixel Widths of 1.0, 1.5, 2.0, 2.5, through 14 Periods of the Input Frequency**
  - **— Support Popular Display Resolutions: VGA, NTSC, PAL, and SECAM**
  - **— On-Chip Triple DAC for Analog RGB/ YUV Output**
  - **— Mix Graphics and Video Images on a Pixel by Pixel Basis**
  - **— Real Time Expansion of the Reduced Sample Density Video Color Components (U, V) to Full Resolution**
  - **— Three Independently Addressable Color Palettes**
  - **— Programmable 2X Horizontal Interpolation of Y Channel**
  - **— 16 x 16 x 2-Bit Cursor Map with Independently Programmable 2X Expansion Factors in X and Y Dimensions**
  - **— YUV to RGB Color Space Conversion**
  - **— 2X Vertical Replication of Y, U, and V Data for Displaying Full Motion Video on VGA Monitor**
  - **— Register and Function Compatible with the 82750DA**

The 82750DB is a custom designed VLSI chip used for processing and displaying video graphic information. It is register and function compatible with the 82750DA.

Reset inputs allow the 82750DB to be genlocked to an external sync source. By programming internal control registers, this sync can be modified to accommodate a wide variety of scanning frequencies. A large selection of bits/pixel, pixels/line, and pixel widths are programmable, allowing a wide latitude in trading-off image quality vs update rate and VRAM requirements.

The 82750DB can operate in a digitizing mode, wherein it generates timing and control signals to the 82750PB and VRAM, but does not output display information. Besides digitizer support signals and video synchronization, the 82750DB outputs digital and analog RGB or YUV information and an 8-bit digital word of alpha data. This alpha channel data may be used to obtain a fractional mix of 82750DB outputs with another video source.



240855-1

82750DB Subsystem Diagram

Order Number: 240855-001

82750DB Display Processor

CONTENTS

CONTENTS

CONTENTS

PAGE

CONTENTS

PAGE

TABLES

1.0 82750DB PIN DESCRIPTION

Pinout

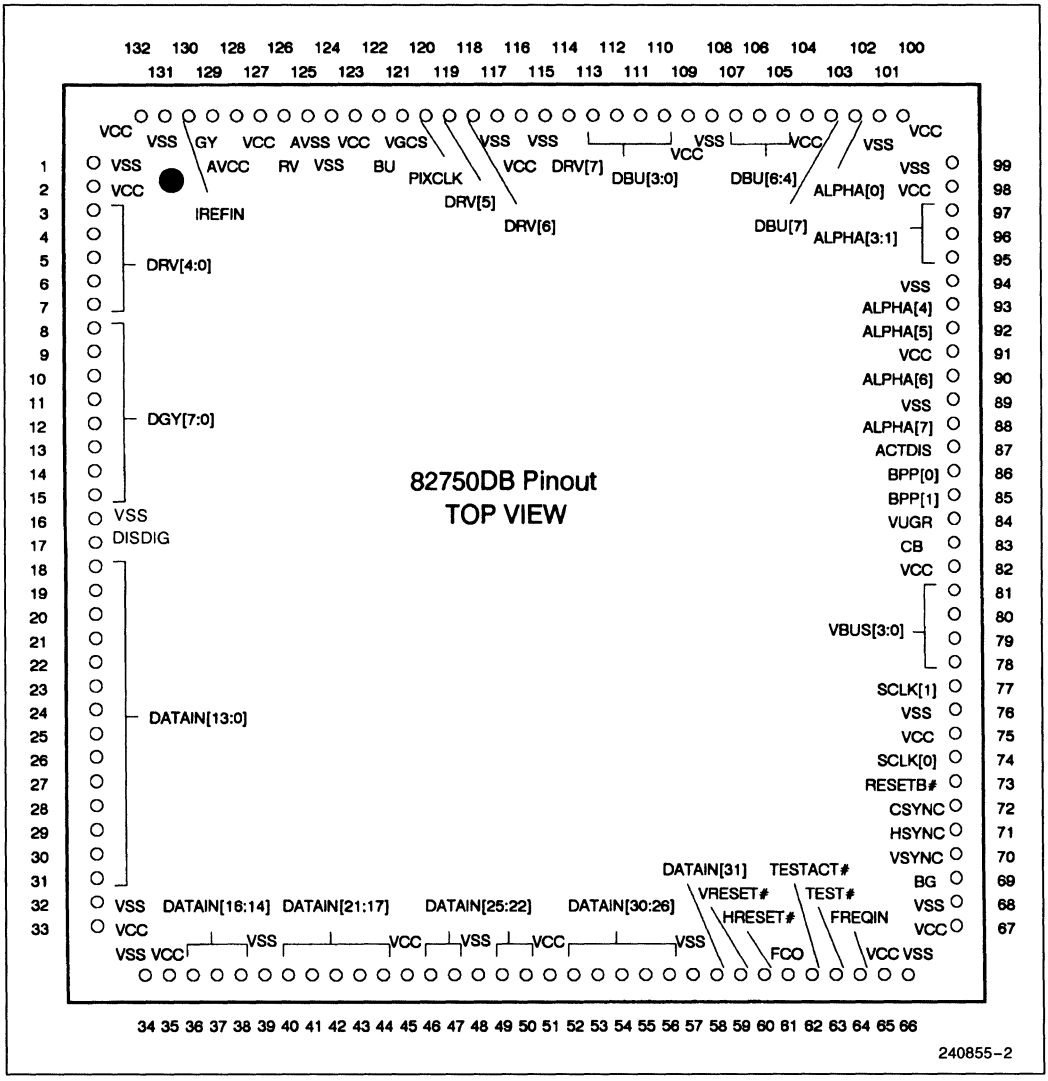


Figure 1-1. 82750DB Pinout

Table 1-1. Pin Cross Reference by Pin Name

Pin Name	Location	Pin Name	Location	Pin Name	Location	Pin Name	Location
ACTDIS	87	DATAIN[15]	37	DRV[7]	114	V_{CC}	82
ALPHA[7]	88	DATAIN[14]	36	DRV[6]	118	V_{CC}	91
ALPHA[6]	90	DATAIN[13]	31	DRV[5]	119	V_{CC}	98
ALPHA[5]	92	DATAIN[12]	30	DRV[4]	3	V_{CC}	100
ALPHA[4]	93	DATAIN[11]	29	DRV[3]	4	V_{CC}	104
ALPHA[3]	95	DATAIN[10]	28	DRV[2]	5	V_{CC}	109
ALPHA[2]	96	DATAIN[9]	27	DRV[1]	6	V_{CC}	116
ALPHA[1]	97	DATAIN[8]	26	DRV[0]	7	V_{CC}	123
ALPHA[0]	102	DATAIN[7]	25	FCO	61	V_{CC}	127
AVCC	128	DATAIN[6]	24	FREQIN	64	V_{CC}	132
AVSS	125	DATAIN[5]	23	GY	129	V_{CC}	121
BG	69	DATAIN[4]	22	HRESET#	60	VRESET#	59
BPP[1]	85	DATAIN[3]	21	HYSNC	71	V_{SS}	1
BPP[0]	86	DATAIN[2]	20	IREFIN	130	V_{SS}	16
BU	122	DATAIN[1]	19	PIXCLK	120	V_{SS}	32
CB	83	DATAIN[0]	18	RESETB#	73	V_{SS}	34
CSYNC	72	DBU[7]	103	RV	126	V_{SS}	39
DATAIN[31]	58	DBU[6]	105	SCLK[1]	77	V_{SS}	48
DATAIN[30]	56	DBU[5]	106	SCLK[0]	74	V_{SS}	57
DATAIN[29]	55	DBU[4]	107	TEST#	63	V_{SS}	66
DATAIN[28]	54	DBU[3]	110	TESTACT#	62	V_{SS}	68
DATAIN[27]	53	DBU[2]	111	VBUS[3]	81	V_{SS}	76
DATAIN[26]	52	DBU[1]	112	VBUS[2]	80	V_{SS}	89
DATAIN[25]	50	DBU[0]	113	VBUS[1]	79	V_{SS}	94
DATAIN[24]	49	DGY[7]	8	VBUS[0]	78	V_{SS}	99
DATAIN[23]	47	DGY[6]	9	V_{CC}	2	V_{SS}	101
DATAIN[22]	46	DGY[5]	10	V_{CC}	33	V_{SS}	108
DATAIN[21]	44	DGY[4]	11	V_{CC}	35	V_{SS}	115
DATAIN[20]	43	DGY[3]	12	V_{CC}	45	V_{SS}	117
DATAIN[19]	42	DGY[2]	13	V_{CC}	51	V_{SS}	124
DATAIN[18]	41	DGY[1]	14	V_{CC}	65	V_{SS}	131
DATAIN[17]	40	DGY[0]	15	V_{CC}	67	VSYNC	70
DATAIN[16]	38	DISDIG	17	V_{CC}	75	VUGR	84

Table 1-2. Pin Cross Reference by Location

Location	Pin Name	Location	Pin Name	Location	Pin Name	Location	Pin Name
1	V_{SS}	34	V_{SS}	67	V_{CC}	100	V_{CC}
2	V_{CC}	35	V_{SS}	68	V_{SS}	101	V_{SS}
3	DRV[4]	36	DATAIN[14]	69	BG	102	ALPHA[0]
4	DRV[3]	37	DATAIN[15]	70	VSYNC	103	DBU[7]
5	DRV[2]	38	DATAIN[16]	71	HSYNC	104	V_{CC}
6	DRV[1]	39	V_{SS}	72	CSYNC	105	DBU[6]
7	DRV[0]	40	DATAIN[17]	73	RESETB#	106	DBU[5]
8	DGY[7]	41	DATAIN[18]	74	SCLK[0]	107	DBU[4]
9	DGY[6]	42	DATAIN[19]	75	V_{CC}	108	V_{SS}
10	DGY[5]	43	DATAIN[20]	76	V_{SS}	109	V_{CC}
11	DGY[4]	44	DATAIN[21]	77	SCLK[1]	110	DBU[3]
12	DGY[3]	45	V_{CC}	78	VBUS[0]	111	DBU[2]
13	DGY[2]	46	DATAIN[22]	79	VBUS[1]	112	DBU[1]
14	DGY[1]	47	DATAIN[23]	80	VBUS[2]	113	DBU[0]
15	DGY[0]	48	V_{SS}	81	VBUS[3]	114	DRV[7]
16	V_{SS}	49	DATAIN[24]	82	V_{CC}	115	V_{SS}
17	DISDIG	50	DATAIN[25]	83	CB	116	V_{CC}
18	DATAIN[0]	51	V_{CC}	84	VUGR	117	V_{SS}
19	DATAIN[1]	52	DATAIN[26]	85	BPP[1]	118	DRV[6]
20	DATAIN[2]	53	DATAIN[27]	86	BPP[0]	119	DRV[5]
21	DATAIN[3]	54	DATAIN[28]	87	ACTDIS	120	PIXCLK
22	DATAIN[4]	55	DATAIN[29]	88	ALPHA[7]	121	VGCS
23	DATAIN[5]	56	DATAIN[30]	89	V_{SS}	122	BU
24	DATAIN[6]	57	V_{SS}	90	ALPHA[6]	123	V_{CC}
25	DATAIN[7]	58	DATAIN[31]	91	V_{CC}	124	V_{SS}
26	DATAIN[8]	59	VRESET#	92	ALPHA[5]	125	AV_{SS}
27	DATAIN[9]	60	HRESET#	93	ALPHA[4]	126	RV
28	DATAIN[10]	61	FCO	94	V_{SS}	127	V_{CC}
29	DATAIN[11]	62	TESTACT#	95	ALPHA[3]	128	AV_{CC}
30	DATAIN[12]	63	TEST#	96	ALPHA[2]	129	GY
31	DATAIN[13]	64	FREQIN	97	ALPHA[1]	130	IREFIN
32	V_{SS}	65	V_{CC}	98	V_{CC}	131	V_{SS}
33	V_{CC}	66	V_{SS}	99	V_{SS}	132	V_{CC}

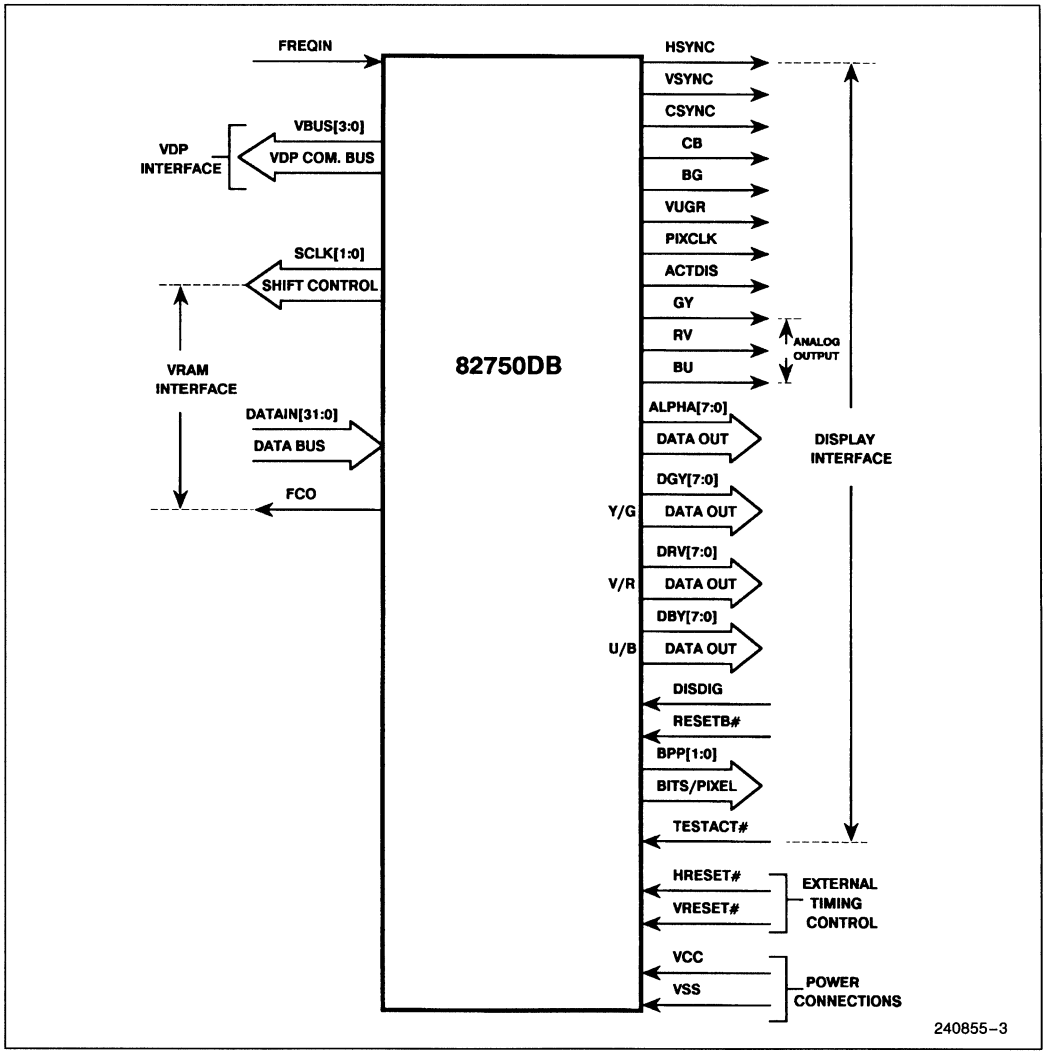


Figure 1-2. 82750DB Functional Signal Groupings

Quick Pin Reference

Table 1-3. Pin Descriptions

Symbol	Type	Name and Function
FREQIN	I	**FREQUENCY INPUT CLOCK:** In normal use, the 82750DB supplies refresh timing for an associated VRAM through the 82750PB. This places a lower limit on the line frequency, which is a programmed multiple of FREQIN. It must generate enough refresh cycles, so a minimum line rate of 4 kHz is required. Furthermore, the 82750PB may run no less than 1/8 the speed of the 82750DB, since the 82750PB samples the timing and control signals generated by the 82750DB. The period of FREQIN is known as a "T" cycle.
RESETB#	I	**EXTERNAL RESET:** Input signal which places all units in the 82750DB into an initialized state, and sets the transfer rate to a default value of 1/3X the operating frequency. It is an edge sensitive iniput which must be held low for a minimum of ten T-cycles. The slowest transfer rate is selected to ensure that the 82750DB will read the register information correctly during the first register transfer, independent of the speed of the VRAMs. During the reset state, the analog video outputs and digital outputs are set to the black level. This will occur a maximum of four cycles after RESETB# is set to a zero. This signal is also used in conjunction with the TESTACT# input to disable outputs.
VRESET#	I	**VERTICAL RESET:** By programming a bit in an internal register, the 82750DB may be placed in the Genlock mode. If this mode is selected, assertion of VRESET# resets all vertical timing to the first line of the next field. It does not affect the horizontal timing, but does generate the on-chip end of field signals. It is an edge sensitive input that is sampled in the 82750DB at the internal time corresponding to the rising edge of FREQIN. If the Genlock mode has not been enabled, this signal will have no effect on the sync timing. The 82750DB will then operate in a free-running mode. Refer to Chapter 3 for a detailed description of genlocking the 82750DB.
HRESET#	1	**HORIZONTAL RESET:** When in the Genlock mode, this input will reset all of the horizontal timing to the start of the line (beginning of horizontal sync). HRESET# does not affect vertical timing (except for an up-to one-line delay) or any other 82750DB registers. This signal is an edge sensitive input that is sampled in the 82750DB at that internal time corresponding to the rising edge of FREQIN. As was the case with the VRESET# signal, this input will be ignored when not in the Genlock mode.
VBUS[3:0]	O	**VDP COMMUNICATION BUS:** The 82750DB outputs status and VRAM transfer requests over these lines to the 82750PB, for 2 to 16 T-cycles (as programmed by the user). Transfer requests can tie up the 82750DB/VRAM, 82750PB/VRAM, or 82750PB/82750DB (VBUS) interfaces for a longer period due to VRAM arbitration. When signals are not being sent out, the VBUS has value 1111, the "null command."
SCLK[1:0]	O	**VRAM SHIFT CLOCKS:** Transfer requests to the 82750PB cause a VRAM address to be set up, and the VRAM serial registers loaded (in the case of displaying) or unloaded (in the case of digitizing). These signals are used to shift data out of and into the VRAMs. Both signals are identical, and run at a maximum rate of 1X of the pixel frequency, except during transfer requests, at which time they run at 1X, 1/2X, or 1/3X of the operating frequency of the 82750DB, as programmed by the user.
DATAIN[31:0]	I	**DATA INPUT BUS:** This is the input data clocked in from VRAM by the SCLK[1:0] signals. The format of the input data is a function of the programmed number of bits/pixel and of the type of transfer cycle being executed. Data will be sampled internally on the rising edge of FREQIN.

Table 1-3. Pin Descriptions (Continued)

Symbol	Type	Name and Function
FCO	O	**FRAME CAPTURE ON:** This is the output signal which indicates to the digitizer that the VRAM serial port has been turned from read mode to write mode. The digitizer may then drive the (common) VRAM serial register data I/O pins. FCO will be asserted after the programmer specifies digitization, five lines after the start of the active vertical display, at the time of HSYNC. This gives the external logic time to switch directions of the VRAM serial data bus. This signal will end four lines after vertical active stops, at the next HSYNC, to make sure the digitizer is off before the next beginning-of-field register transfer.
HSYNC	O	**HORIZONTAL SYNCHRONIZATION:** Video synchronization signal which is asserted at the beginning of every line and ends a programmed time later. (The duration of this signal is specified in T-cycles.)
VSYNC	O	**VERTICAL SYNCHRONIZATION:** Video synchronization signal which can be programmed to start (once) and end (once) in every field. (The start and stop position may be specified in half-line units.)
CSYNC	O	**COMPOSITE SYNCHRONIZATION PULSE:** This contains the programmed vertical serration and equalization information, as well as horizonal synchronization pulses.
CB	O	**COMPOSITE BLANKING:** This signal can be programmed to end once and start once in each line, and end once and start once every field.
BG	O	**BURST GATE:** This signal starts and stops at user-programmable horizontal positions in each line, in a programmable vertical group of lines. The primary use of this signal is to provide a "window" during which the BURST output should be inserted to generate a baseband NTSC signal. The output frequency is set by an integer divisor (0–31) and the rate of the FREQIN clock input. To use this effectively, the 82750DB must operate at an integer multiple of the NTSC 3.58 MHz color subcarrier. The number is programmed in two's complement form in the General Control register.
PIXCLK	O	**PIXEL CLOCK:** This output signals valid data on the DGY, DRV, DBU, GY, RV, and BU lines. PIXCLK becomes active one-half of a T-cycle after valid data appears on DGY, DRV, or DBU, and coincident with GY, RV, and BU. During active display time it is issued at a steady rate of 1/(T-cycles/pixel) times per T-cycle, and otherwise at a steady rate of once per T-cycle. Its duration is one-half of a T-cycle, and its rising edge may synchronize with either rising or falling edges of FREQIN depending on the pixel frequency. This signal may be used to synchronize off-chip processing of the pixel data outputs.
GY, RV, BU	O	**ANALOG PIXEL OUTPUTS:** These signals are the processed pixel data from the 82750DB in analog form. During the display, these signals may be programmed to output pixel data in either YUV or RGB format.

Output Format	DGY	DRV	DBU
YUV	Y	V	U
RGB	G	R	B

DGY[7:0], DRV[7:0], DBU[7:0]	O	**DIGITAL VIDEO OUTPUTS:** These are the digital outputs of the GY, RV, and BU channels, respectively. They are valid with respect to the rising edge of PIXCLK.
ALPHA[7:0]	O	**ALPHA CHANNEL:** These 8 bits are used to output a digital value for mixing the 82750DB output with another video signal off-chip. The alpha channel information may be included in the pixel data, or may be output based on a comparison of the pixel data with user-programmed values.

Table 1-3. Pin Descriptions (Continued)

<table>
<tr><th>Symbol</th><th>Type</th><th>Name and Function</th></tr>
<tr><td>VUGR</td><td>O</td><td>VIDEO OR GRAPHICS INDICATOR: When asserted, the G, R, and B outputs are derived from the subsampled VU data. When negated, the pixel information is completely derived from the bitmap.</td></tr>
<tr><td>ACTDIS</td><td>O</td><td>ACTIVE DISPLAY: This is the active portion of the display as programmed by the user. It is delayed by the pipeline through the 82750DB, which is 5 lines vertically and a variable number horizontally, depending on the display mode.</td></tr>
<tr><td>BPP[1:0]</td><td>O</td><td>BITS PER PIXEL: During the nonactive display, the user programmed bits/pixel is encoded on these lines. During active display, the BPP[0] signal is multiplexed with a signal which indicates that the cursor is active (non-transparent). The BPP[1] signal remains as it was during nonactive. This allows the users to latch the BPP[1:0] signals during nonactive displayed time (as indicated by ACTDIS being zero) for post-processing of the 82750DB output. The active cursor window on BPP[0] can be used during active display, to multiplex in other video streams into the output display. The following table illustrates the encoding on the BPP signal.
<table>
<tr><th>Bits/Pixel</th><th>ACTDIS</th><th>BPP[0]</th><th>BPP[1]</th></tr>
<tr><td>8</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>16</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>32</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>pseudo 16</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>8</td><td>1</td><td>Cursor Active</td><td>0</td></tr>
<tr><td>16</td><td>1</td><td>Cursor Active</td><td>1</td></tr>
<tr><td>32</td><td>1</td><td>Cursor Active</td><td>0</td></tr>
<tr><td>pseudo 16</td><td>1</td><td>Cursor Active</td><td>1</td></tr>
</table></td></tr>
<tr><td>DISDIG</td><td>I</td><td>DISABLE DIGITAL OUTPUTS: When this input is active, the digital outputs of the 82750DB will be set to zero. In applications that use only the analog outputs of the 82750DB, the digital outputs must be disabled.</td></tr>
<tr><td>TESTACT#</td><td>I</td><td>TEST ACTIVE: Active low signal that is used in conjunction with the RESETB# signal to allow the chip to perform one of the following functions:
<table>
<tr><th>RESETB#</th><th>TESTACT#</th><th>82750DB State</th></tr>
<tr><td>0</td><td>1</td><td>Enter Reset State</td></tr>
<tr><td>0</td><td>0</td><td>Enter Reset State
Tristate All Outputs
Analog Outputs are Zero</td></tr>
<tr><td>1</td><td>1</td><td>Normal Operation</td></tr>
<tr><td>1</td><td>0</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>TEST#</td><td>I</td><td>TEST INPUT: This signal must be set to VCC to guarantee correct chip operation.</td></tr>
<tr><td>VGCS</td><td>O</td><td>INTERNAL VOLTAGE REFERENCE: This signal must be decoupled to AVCC.</td></tr>
<tr><td>IREFIN</td><td>I</td><td>ANALOG CURRENT REFERENCE: Under normal operation, this signal should be tied to a temperature compensated current reference to AVSS. This signal must be decoupled to AVCC.</td></tr>
<tr><td>AV_{CC}</td><td>I</td><td>ANALOG POWER pin provides +5 V_{DC} supply to the Digital to Analog Converter.</td></tr>
<tr><td>AV_{SS}</td><td>I</td><td>ANALOG GROUND pin provides the 0V connection to which the analog outputs are referenced. This must be connected to VSS.</td></tr>
<tr><td>V_{CC}</td><td>I</td><td>POWER pins provide +5 V_{DC} supply input.</td></tr>
<tr><td>V_{SS}</td><td>I</td><td>GROUND pins provide the 0V connection to which all inputs and outputs are referenced.</td></tr>
</table>

Table 1-4. Input Pins

Name	Active Level	Synchronous/ Asynchronous
FREQIN	HIGH	Synchronous
RESETB#	LOW	Asynchronous
VRESET#	LOW	Asynchronous
HRESET#	LOW	Asynchronous
DISDIG	HIGH	Asynchronous
TESTACT#	LOW	Asynchronous
TEST#	LOW	ASynchronous

All output pins have an active level of HIGH, and are floated when RESETB# and TESTACT# are set to a zero. The exceptions are GY, RV, and BU which will be forced to a zero level.

2.0 ARCHITECTURE

Overview

There are 10 units in the 82750DB. Each of the units operates independently at the maximum clock rate input to the chip. The control information for each block is distributed in programmable registers throughout the chip. These registers are loaded on user-specified lines during the horizontal and vertical blanking intervals of the field. The register data that was read in from VRAM is passed from block to block during the blanking intervals of the display, on the same lines that the pixel information is passed during the active display. The Functional Block Diagram is shown in Figure 2-1.

In order to maximize speed and compensate for processing delays, the chip is heavily pipelined. All inter-block information is delay-equalized to accommodate the different pipeline lengths in each module. As a result, the total pipeline delay is dependent on the number of processing units that are used to generate the display. Chapter 4 describes how the user programming is affected by these pipeline delays.

Each of the units are described in more detail in the following sections of this chapter.

Sync Generation and Timing

The sync generation and timing block generates all of the internal timing and control signals, as well as the video synchronization signals. Sync and timing information may be derived from two sources: from the master clock, in which case the control registers on the 82750DB are programmed to provide the desired display frequency in terms of periods of the master clock (T-cycles), or from the horizontal and vertical external reset signals. (The latter is known as the genlock mode.) Characteristics such as line rate, blanking and border intervals, and composite synchronization parameters can be independently set. Since the 82750DB can be reprogrammed once each line, horizontal strips of different resolutions can be supported on the same display. However, the horizontal strips that can be supported are limited by the host processor's response to redefining the bitmap pointers resident on the 82750PB.

The horizontal and vertical display parameters are fully programmable. Figure 2-2 illustrates the horizontal programming parameters. The line starts at dot zero, with the length of half of a line programmed in T-cycles. The length of the total line is twice the half-line length. Parameters such as horizontal sync width, horizontal blanking start and stop, and horizontal active start and stop are all specified by the user. Note that the border time is not explicitly programmed, but is defined as the region of the display line where neither active display nor blanking is programmed to occur. In order for the 82750DB to function correctly, the width of the horizontal active display should be programmed such that the end of the horizontal active display coincides with the end of the last displayed pixel.

Figure 2-3 shows the vertical programming parameters. The basic unit for vertical programming is in units of half lines, with the half-line count for each field starting at zero. Where appropriate for a parameter, the count is programmed in units of full lines. The length of the complete field is programmed in half lines, which makes it convenient for distinguishing between interlaced and non-interlaced displays. (For interlaced displays, the number of half lines is odd, for non-interlaced displays, it is even.) The vertical active and blanking regions may be independently programmed, with the border time defined as the region where blanking and active display is not on.

NOTE:

Sync parameters are completely independent of the display parameters. This allows the sync signals to be positioned anywhere in the field (even during active display).

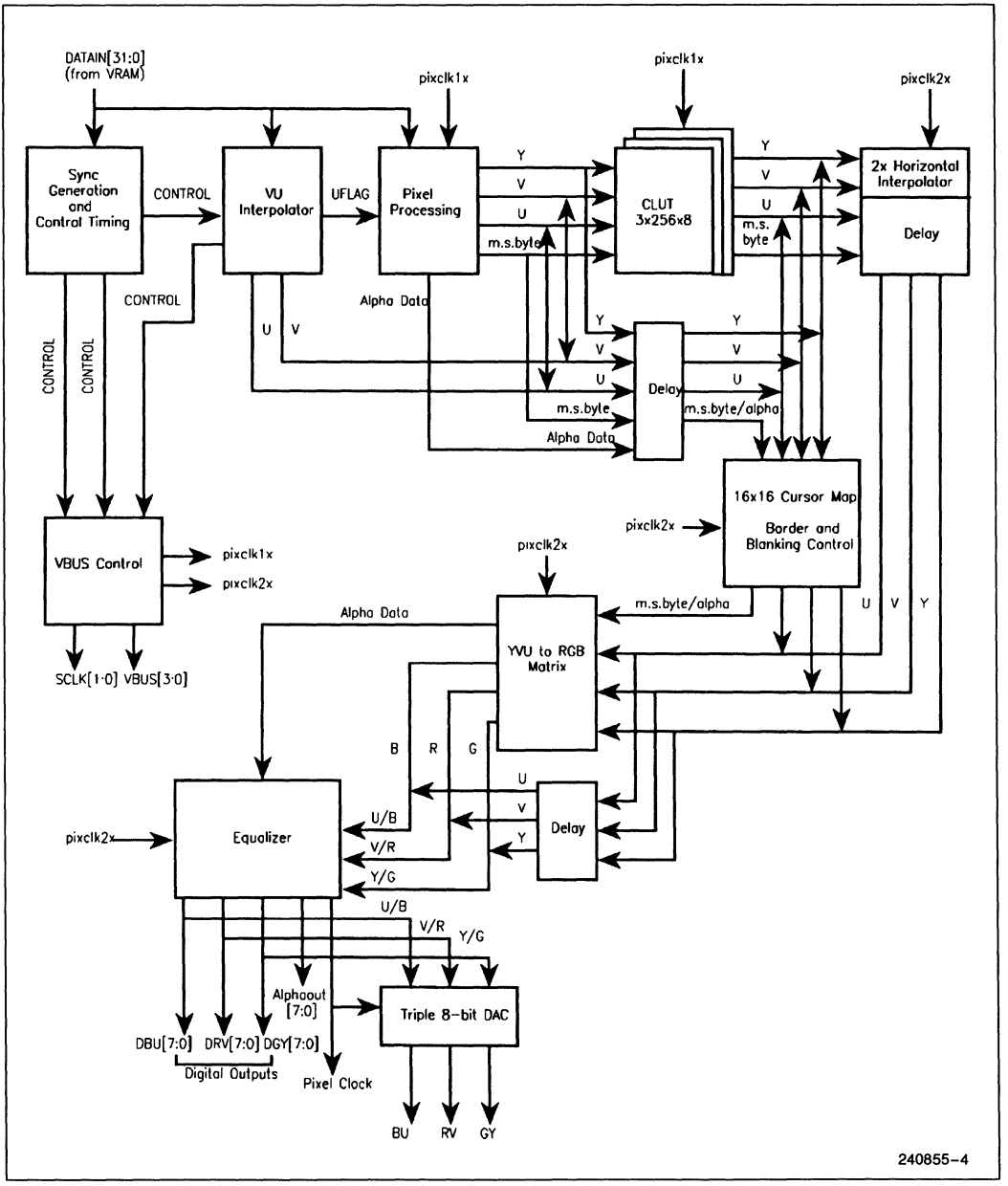
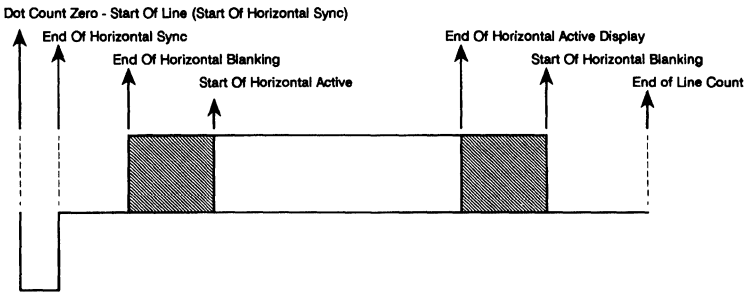


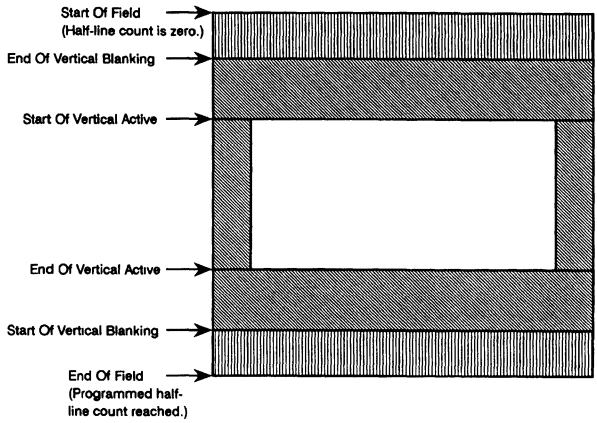
Figure 2-1. 82750DB Unit Level Diagram



- All horizontal programming parameters are in periods of the master clock.
- Border may be eliminated by programming the blanking time to abut the active display
- Pixel widths must be an integer divisor of horizontal active width

240855-5

Figure 2-2. Horizontal Programming Parameters



- All vertical programming parameters shown are in half lines
- Vertical border may be eliminated by abutting active display and blanking times.
- Positioning of vertical blanking and sync are fully independent.

240855-6

Figure 2-3. Vertical Programming Parameters

VBUS Control

The VBUS controller sends all 82750DB requests for display bitmaps, VRAM refresh, and synchronization information to the 82750PB, at programmable times during a field. Transfer requests are scheduled to occur on a line basis, so only their vertical position (or line) is specified by the user. Other commands, like refresh requests, occur every line, and their horizontal position (or dot position) in the line must be specified by the user. Transfer requests are given the highest priority by the VBUS control circuit and are performed first during a blanking interval. The programmer has the responsibility of scheduling the line oriented codes, like refresh, so that they do not collide with the transfer requests.

Besides arbitrating the scheduled transfer requests, the VBUS controller also reads the data from the VRAM shift registers using the two shift clock outputs (SCLK[1:0]). The code corresponding to the type of data to be read is asserted for a programmable number of cycles on the 4-bit VBUS. The 82750DB then waits a programmable delay before reading the data from the VRAM. This delay should be long enough to guarantee that the 82750PB has completed loading the information into the serial shift register of the VRAM. Both signals are off while the code causing the transfer cycle is active on the VBUS, as well as during the read delay time. Figure 2-4 illustrates this communication between the 82750PB and the 82750DB.

When the delay wait is over, the shift clock outputs are activated. the SCLK[1:0] signals' behavior is dependent on the transfer rate that the user has selected—either 1X, 1/2X, or 1/3X the operating frequency. Note that if the RESETB# signal is applied, the transfer rate is automatically set to 1/3X during the first automatic register transfer, regardless of the state of the transfer rate selection. The transfer rate may be changed in the first register transfer after RESETB# is set to a logic one value.

Figure 2-5 illustrates how the SCLKs operate in the 1X mode in a system. SCLK[1:0] signals will toggle between zero and one on the rising edge of FREQIN, after an internal logic delay. The data is read into the 82750DB on the rising edge of the internal clock, one 82750DB clock cycle after the SCLK outputs are asserted. Since there are 32 data input pins, each SCLK can read in the serial data from eight 256 x 4 VRAM memory devices. Adding external buffering to the SCLKs (to drive more memory) will also add delay to the memory access. The delay increase may require more than one T-cycle before the VRAM data is valid. In this case, the time between the rising edge of the internal 82750DB clock that generates the SCLKs and the edge that latches the data must be increased.

There are two solutions, the operating frequency of 82750DB can be lowered to accommodate a longer T-cycle, or the 1/2X SCLK mode may be selected (as shown in Figure 2-6). When using the 1/2X transfer rate, the data is read into the 82750DB on the rising edge of the internal clock, two 82750DB clock cycles after the SCLK outputs are asserted.

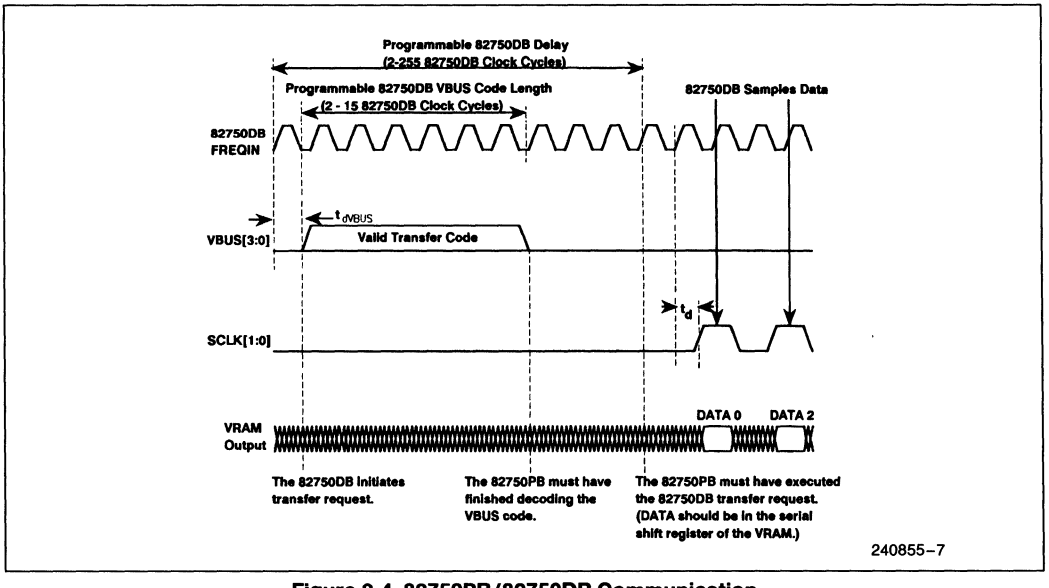


Figure 2-4. 82750PB/82750DB Communication

Figure 2-7 illustrates 1/3X (default) shift clock operation that is used during the RESET mode or may be programmed by the user. The first word of data is latched by the 82750DB on the rising ede of the FREQIN that is three T-cycles after the SCLK outputs were asserted. This allows three full 82750DB cycles for the VRAMs to output valid data, which gives extra margin for applications that need longer shift read cycles (due to slower memories or external logic delays) and do not wish to operate the 82750DB at a slower speed.

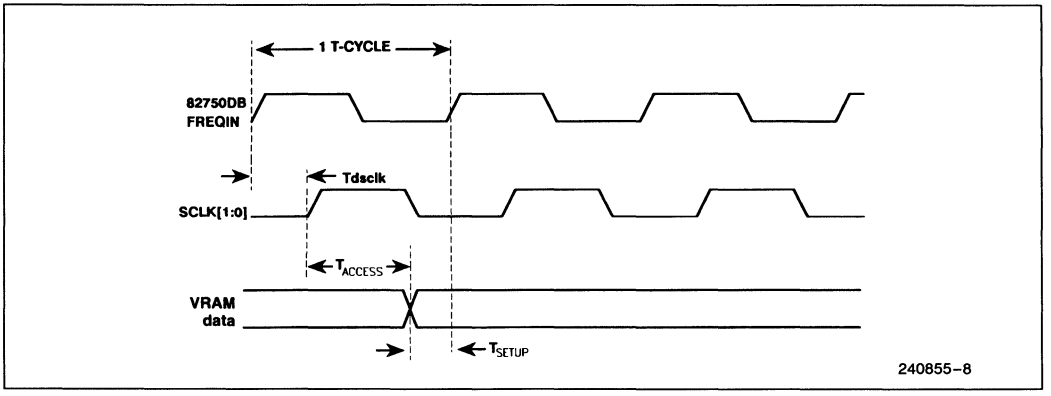


Figure 2-5. 82750DB 1X Shift Clock Operation

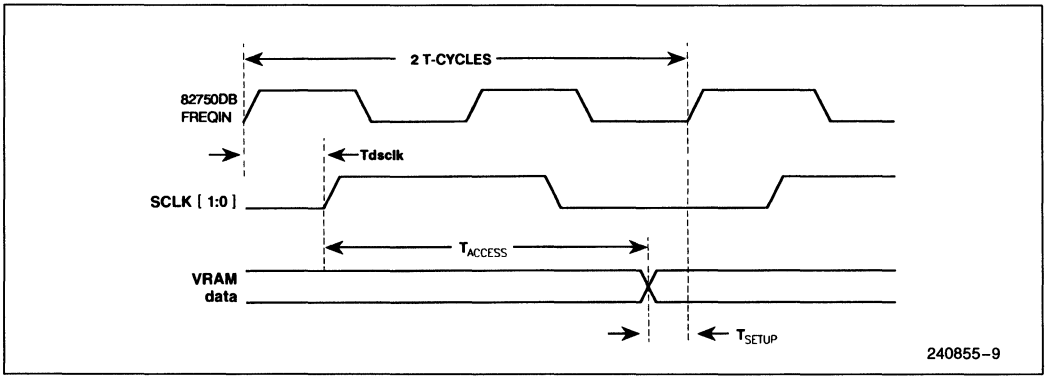


Figure 2-6. 82750DB 1/2X Shift Clock Operation

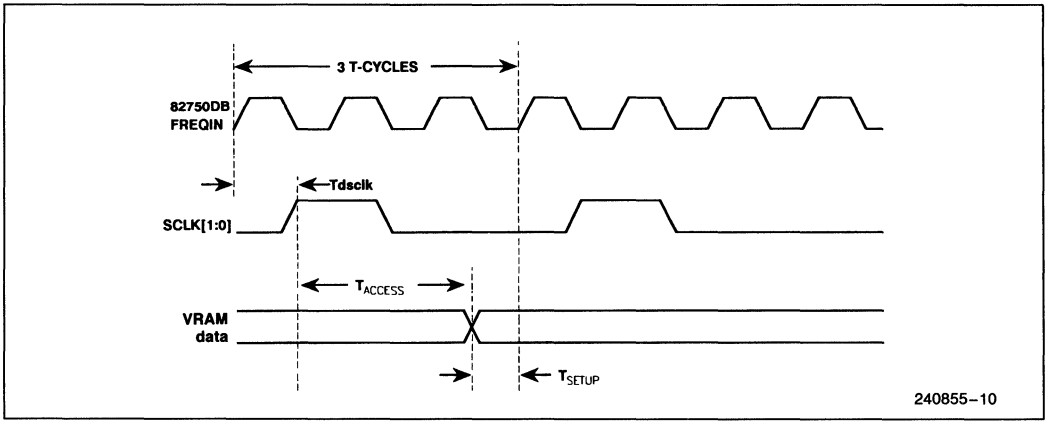


Figure 2-7. 82750DB 1/3X Shift Clock Operation

When reading data from memory during active display, the SCLK[1:0] outputs operate at a rate required to support the programmed display rate. This rate is determined from the following equation:

$$RATE = \frac{(\#\ \text{of bits/pixel})}{(\text{32-bit/word}) * (\#\ \text{word/fetch}) * (\#\text{T-cycle/pixel})}$$

where: # bits/pixel and # T-cycles/pixel are user-programmed

word/fetch is: 1

The SCLK[1:0] outputs will be the same frequency as the input clock in the 1X shift clock mode, and one half the input clock frequency when using the 1/2X mode. The frequency will be one third in the input clock when using the 1/3X mode. In the 1/3X mode the SCLK[1:0] outputs will be high for one T-cycle, and low for 2 T-cycles.

VBUS CODE DESCRIPTION

When the 82750DB is actively fetching and displaying pixels, VUXFER, BMX/YBMNPX, and REGX are typically sent over the VBUS. Of the three codes, REGX has top priority, followed by VUXFER, and last by BMX/YBMNPX. These commands may be programmed to occur each active line during the blanking interval for the line just completed. If a register transfer has been programmed for an active line, it takes priority and is executed first. Otherwise, or immediately after the register transfer, any scheduled VUXFER and BMX/YBMNPX commands are executed. The programmer has the responsibility for verifying that the sum of times required by these commands does not exceed horizontal non-active display time. The 82750DB will commence fetching pixels at the subsequent start of active display. A detailed explanation of the different types of VBUS commands and their corresponding codes follows.

Transfer Requests

The following commands request the 82750PB to transfer information from the VRAM array into the VRAM shift register. They are listed in the priority they are sent, when multiple requests are programmed for a given line. When asserting a transfer request, the programmer must be aware of two other programmed parameters, VBLEN and SCLK delay.

The VBLEN parameter is a user programmed value whose bits lie in the General Control Register. It is the length of time, in 82750DB T-cycles, that a particular VBUS code will be held at the outputs. It is used to ensure that the asynchronously operating 82750PB chip will have enough time to recognize and begin operating on an 82750DB transfer request.

The other parameter the programmer needs to set is the SCLK delay. This can be found in the Pixel Control Register. It is the number of 82750DB clock cycles that the DB will wait before clocking in data, out of the VRAM, after the initiation of a transfer request on the VBUS outputs.

REGX (0010) This command requests that the 82750PB transfer 82750DB register information into the VRAM shift registers. Besides the automatic 82750DB register transfer that occurs on the second line (line 2) of each field, the programmer can specify the next horizontal line on which another register transfer is to take place. The transfers may be scheduled many times during the field. On the first transfer, the 82750PB uses the contents of its 82750DBc register as the starting address of the 82750DB register data. On each subsequent access, the programmed pitch value in 82750PB's 82750DBc-PITCH register is added to the accumulated start address. The programmer must ensure that the data is stored in VRAM at the correct address. Since the pitch remains constant, the longest register load will determine the pitch value.

The VBUS unit forms a vertical checksum on all the register information. Each bit in the register word undergoes an exclusive-OR with the corresponding bit in the previous data word. The 82750DB compares this information with the user generated checksum, which is the last 32-bit data word read into the 82750DB during a register transfer. If the values do not match, the 82750DB will disable all of its digital sync and data outputs, enter the reset state, and send a SHUTDOWN code (82750DBSD) to the 82750PB over the VBUS[3:0] outputs. If the new checksum is correct, the new register values will take effect immediately.

VUXFER (0001) This code is used to request VU data, providing new VU data is required by the 82750DB. This command is issued only on vertically active lines (as programmed in the register, not as seen on the screen) and possibly the four lines after. On each line, a row of V and/or U samples are loaded into the VU interpolator line stores. The pattern of requests depends upon the mode in which the VU interpolator is operating. In the interlaced VU mode, one line of samples for both the V and U components are fetched during each transfer; in the non-interlaced VU mode, only one line of samples for either the V or U components is fetched. Table 2-1 illustrates the pattern of requests. M is the programmed first vertical active line, and N the last active line. The modes listed have VU transfer requests following the end of horizontal active of the lines specified, stopping with the last line, N + 4.

Table 2-1. VU Transfer Request Patterns

Mode	Active Line	Request VU Data
2x Non-Interlaced	M	Fetch 1st Line of V
	M + 1	Fetch 1st Line of U
	M + 4	Fetch 2nd Line of V
	M + 5	Fetch 2nd Line of U
	N + 4	Fetch Last Line of V
2x Interlaced (Odd and Even Fields)	M	Fetch 1st Line of V and U
	M + 4	Fetch 2nd Line of V and U
	M + 5	Fetch 3rd Line of V and U
	N + 4	Fetch Last Line of V and U
4x Non-Interlaced	M	Fetch 1st Line of V
	M + 1	Fetch 1st Line of U
	M + 4	Fetch 2nd Line of V
	M + 5	Fetch 2nd Line of U
	M + 8	Fetch 3rd Line of V
	N + 4	Fetch Last Line of V
4x Interlaced (Odd and Even Fields)	M	Fetch 1st Line of V and U
	M + 4	Fetch 2nd Line of V and U
	M + 6	Fetch 3rd Line of V and U
	N + 4	Fetch Last Line of V and U

The 82750PB uses another internal pointer to cause the VRAM to load the desired VU data into its shift registers (incrementing the pointer by a pitch value). This command is asserted for a programmable number of T-cycles (m), as specified in the Miscellaneous Control register. Then, the 82750DB fetches them, tying up the 82750DB/VRAM interface for (n + 2) cycles, where n is ¼ the programmable total number of 8-bit samples of V and U fetched. Note that one extra word, which may overlap the next VBUS command, is fetched.

By setting a bit in the Miscellaneous Control register, it is possible to replicate lines of V and U generated by the interpolator for the entire field. Since each line of VU data is displayed twice, the rate that the VU sample map has to be fetched from VRAM is reduced by ½. Table 2-2 lists the sequence of VU loads.

In some cases, the VU interpolator may cover only a portion of the display. In those instances, M in the above examples would be the first line that VU interpolation is enabled. N would be the last line that VU interpolation is enabled. Regardless of the state of the Line Replicate bit, there would be no vertical pipeline delay between the loading of the first line of samples and the second line of samples. The first line of samples would be loaded at M-1, and the second line at M. This reduces the delay between switching interpolation modes during a single display.

Table 2-2. VU Transfer Request Patterns with Line Replicate

Mode	Active Line	Request
2x Non-Interlaced	M	Fetch 1st Line of V
	M + 1	Fetch 1st Line of U
	M + 4	Fetch 2nd Line of V
	M + 5	Fetch 2nd Line of U
	M + 8	Fetch 3rd Line of V
	M + 9	Fetch 3rd Line of U
	N + 4	Fetch Last Line of V
2x Interlaced (Odd and Even Fields)	M	Fetch 1st Line of V and U
	M + 4	Fetch 2nd Line of V and U
	M + 6	Fetch 3rd Line of V and U
	N + 4	Fetch Last Line of V and U
4x Non-Interlaced	M	Fetch 1st Line of V
	M + 1	Fetch 1st Line of U
	M + 4	Fetch 2nd Line of V
	M + 5	Fetch 2nd Line of U
	M + 12	Fetch 2nd Line of V
	M + 13	Fetch 2nd Line of U
	N + 4	Fetch Last Line of V
4x Interlaced (Odd and Even Fields)	M	Fetch 1st Line of V and U
	M + 4	Fetch 2nd Line of V and U
	M + 8	Fetch 3rd Line of V and U
	N + 4	Fetch Last Line of V and U

BMX (0000) This command requests a bitmap. BMX (0000) is sent after horizontal active stops, beginning on the fifth line after vertical active starts, and continuing until the fifth line after vertical active stops. (There is a vertical pipeline delay of five lines through the 82750DB, due to internal timing requirements.) A line programmed to start at line M, wil have its first active line displayed at line M + 5. The 82750PB uses an internal pointer to cause the VRAM shift registers to be loaded with pixel values. The 82750DB subsequently fetches them as required for display. This command is asserted on the VBUS for the user-programmed number of T-cycles and must be completed before active display begins.

YBMNPX (0100) This command performs a Y bitmap transfer without performing a pitch calculation. When the line replicate mode is selected by Bit 22 in the Miscellaneous Control register, this code is asserted every other display line so that the same line of information can be used twice.

Digitizer Commands

When in the line replicate mode, and digitizing an NTSC source (for example, when genlocking an NTSC source to a system that uses only a VGA monitor), each line of captured data is effectively output at twice the rate. Since each line need only be stored once in memory (it is duplicted automatically in the display mode) only one WRDIGI code, followed by a WRDIGINP, is sent every other line. On alternate lines, two WRDIGINP are sent and will select the last address that was written, without incrementing the 82750PB bitmap address pointer. This is described in detail in Chapter 3.

WRDIGI (0011) This command requests a write of digitized data. The operation of this command is dependent upon the external hardware and is discussed in the section on genlocking (page 29). If digitizing is enabled, this command is asserted on the VBUS for a programmable number of T-cycles. The pointer is then incremented by a pitch value. Since each horizontal line is stored in a single row of memory, this pitch value is equal to the horizontal resolution, in bytes, for non-interlaced bitmaps. For interlaced bitmaps, the pitch value is equal to twice the horizontal resolution, in bytes. This allows alternate lines of data to be skipped over in successive fields.

WRDIGINP (0111) This command allows access to digitized data without performing a pitch calculation. WRDIGINP (0111) requests that the 82750PB perform a transfer request at the last calculated address. Note that only a memory transfer cycle is performed—the pitch value is not added to this address. This will always ensure that the digitized data is written into the last selected memory address, in case a physical memory boundary has been crossed. This command is asserted after the WRDIGI transfer has completed.

Refresh and Control Commands

The following signals are used to pass refresh requests and control information to the 82750PB.

DFL (1000) The Display Format Load command is a maskable host processor interrupt that can be programmed to occur at any time during the display. This is used by the 82750PB to transfer the shadow register contents into the working register set in the VRAM interface. This is useful in supporting split-screen-type applications, where it is desirable to change the bitmap pointers at some point before the end of the display.

82750DBSD (1001) This command is the 82750DB Shut Down code. During every register transfer, the 82750DB keeps an internal vertical exclusive-or checksum of the register data as it is read onto the chip. The last word of data that is read during the register transfer is the user-generated checksum. If the two checksums match, operation proceeds as normal. If they do not match, the 82750DB enters the reset state and sends this code to the 82750PB. The 82750DB will remain reset until the reset pin is asserted and negated by the host processor.

REFRESH (1010) This command asks the 82750PB to generate up to 15 refresh cycles every horizontal line. The 82750DB transfer cycles have a higher priority than refresh requests in the 82750PB. REFRESH will not be asserted if programmed to occur at the same time as a transfer request code.

Video Synchronization Information

The following codes are used to pass the video line and field information from 82750DB to the pixel processor.

VEVEN (1101) This code indicates the start of an even (i.e. second) field of a frame. This command is sent coincident with line one of each even field.

VODD (1100) This code indicates the start of an odd (i.e. first or only) field of a frame. This command is sent immediately after RESETB# is negated, and coincident with line one of the odd field.

HLIN (1110) This code marks every horizontal line at a programmable point in the line. HLIN is used by the 82750PB to increment its horizontal line counter.

Pixel Processing Path

This logic accepts the 32-bit word from the input latch and divides the word into the programmed pixel format. This will result in either four 8-bit pixels, two 16-bit pixels, one 32-bit pixel, or an 8-bit pixel with an 8-bit alpha value (pseudo 16-bit mode). The pixels act as addresses to the color table, or may bypass the table completely as described below.

Pixel information may be mixed with the output of the VU interpolator, which outputs interpolated samples derived from a reduced sample bitmap. The least significant bit of Y or LSB of U can be programmed to act as a switch between using the explicit pixel value of YUV or using the luminance portion of the pixel with the VU portion obtained from the interpolator. If the value of the LSB of Y (or U, whichever is selected) is zero, the pixel data is used. If the LSB of Y (or U) is one, the output of the VU interpolator is used. Note that if the LSB of Y is used as the switch flag, the luminance portion of the word will be only 7 bits wide.

The alpha information is also processed in this block. The alpha data may come from one of two sources: it may be explicitly coded in the pixel word, as is the case in the 32-bit/pixel and pseudo 16-bit/pixel mode, or it may be obtained by comparing the Y portion of the pixel with a preprogrammed value and outputting one preprogrammed value if they match and a different value if they do not match. This latter capability is known as Alpha Trap.

VU Interpolation

When VU interpolation is enabled by the programmer, and when the display is in the active region, "VU data" will be fetched, as required by the interpolator (by the mechanisms discussed previously in the section titled "VBUS Code Description"). This data has the format V, V, . . . , V, U, U, . . . , U where each V or U is 8 bits, and the bytes are grouped into 32-bit double-words with earliest in lowest order. The number, "N", of V bytes and U bytes is the same; N is programmed to be either 256 samples, or one of 32 to 192 samples in 32-byte increments.

The first V data and the first U data fetched on the first line of VU interpolation supplies the VU value for the first active pixel on that line. All the other VU pairs that are fetched define values for the grid of pixels defined below and to the right of this one by the VU expansion factor every other or every fourth horizontally and vertically. Most other VU values are filled in recursively by interpolation. Wherever there is a pixel which lies between two pixels with known values, it is given the value of the weighted average of the known values. Values are understood to be non-negative integers. When the final value is outputted, any fractions are truncated or rounded to the closest odd integer according to the programmed value of the interpolation round flag. This process is iterated until all pixels have assigned color values. If the number of VU data samples loaded into the 82750DB is not enough to cover the active display area, then the last data sample will be replicated horizontally across the active display window.

As mentioned previously in the VBUS Control discussion, each line of VU data can be used twice by setting the Line Replicate bit in the Miscellaneous Control register. Also, each horizontal VU sample can be replicated by setting the VU Replicate bit in the Pixel Control register. This will cause the V and U pixels generated by the VU interpolator every pixel time to be used twice. This can result in an effective 8X horizontal expansion, which is useful when horizontal blanking time is at a premium. This bit affects the horizontal interpolation algorithm only, and will not affect the line loading sequence for VU during the active display.

When interpolation is turned on by the programmer (by specifying a non-zero number of samples to be fetched), VU interpolation may nevertheless be disabled for each pixel if the following conditions are met:

1. Conditional interpolation has been selected by the programmer,

   AND

   Either of the two user-programmed conditions:

   a. Switching on the LSB of the U bit has been selected, and the lowest-order bit of the U value fetched for the upper left pixel in the block has value zero. This allows switching to occur on a 2 x 2-pixel or 4 x 4-pixel grid, depending on the expansion mode the user has selected. The full 8 bits of Y and V are used, but the usable space of U has been decreased to 7 bits.

   b. Switching on the LSB of the Y bit has been selected, and the low order bit of the Y value for the current pixel has a value of zero.

2. Display of fetched and interpolated VU values may also be suppressed by setting the Interpolation Output Enable bit (in the miscellaneous control register) to zero. This will allow VU data to be loaded into the VU line stores without displaying VU data. This is useful when a mid-screen transition is made between two interpolation modes, to compensate for the vertical latency of the interpolation process.

Colormap Lookup Table (CLUT) Operation

The 82750DB contains three 256 x 8-bit color lookup tables. The color maps can be accessed separately, or may act as one large 256 x 24-bit table. The manner in which the tables are addressed is determined by the programmed bits/pixel and depends on whether the pixel is a graphics or video pixel. Also each Y, U, and V color table address can be masked. The masks can be used in all the bit/pixel modes, but are most useful with the 16-bit/pixel mode. In this mode, the mask allows the YUV values to be mapped to 8-bit values instead of 6-5-5.

Each channel (Y, U, V) has a MASK SET register and a MASK DATA register that selects the color lookup address bit to be changed and the new value of the bit, respectively. A simple mask operation on one channel is illustated in Figure 2-8.

The CLUT address mask operation is determined by a logical equation given by:

$$\text{Result} = (\text{mask set and mask data}) \mid (\overline{\text{mask set}} \text{ and data byte})$$

Each bit of the Result byte is determined individually by this equation. The Result byte is then further processed in order to produce the CLUT RAM address.

For modes that require both video and graphics to pass through the color table, the table can be split into two halves: one half for graphics and the other for video pixels. By using the SPLITCLUT bit in the Miscellaneous Control register in conjunction with the LSB of Y or U, the color table address is forced to either the video table or graphics table automatically. In this case, the masking operation is still used, but the address is forced to either an even or odd entry, regardless of the results of the masking operation. The flag bit that decides between the two types of pixels automatically selects the correct portion of the CLUT table for a single channel. Note the LSB of Y on U selects the proper half of the CLUT for that single component. The SPLIT CLUT mode assures the proper half of the CLUT is used for all three components.

The color table can be bypassed completely when displaying either graphics or video, independent of the programmed bits/pixel. This is programmed by the user via the VIDEO PASS and GRAPHICS PASS bits in the Miscellaneous Control register. Table 2-3 summarizes the various modes when using the CLUT.

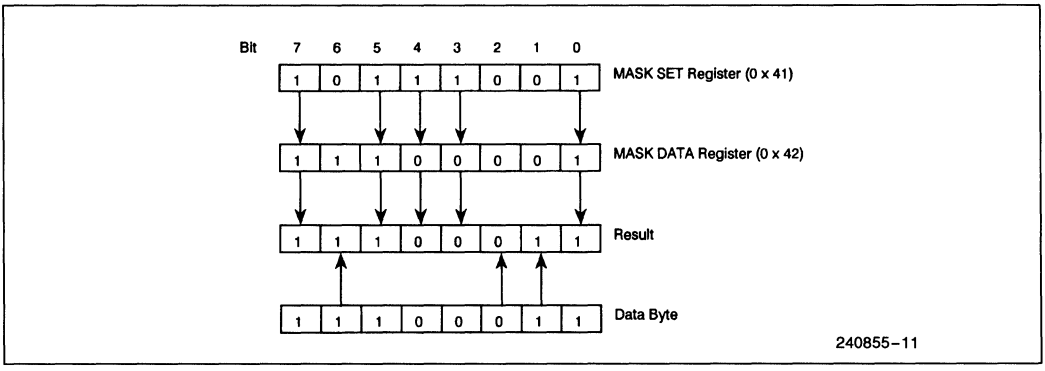


Figure 2-8. Mask Operation on CLUT Address

Table 2-3. CLUT Modes

Graphics Pass	Video Pass	LSB Y or U	SPLITCLUT	Colormap Address
0	X	0	0	Masked Graphics Data
1	X	0	X	Graphics Pixels Bypass CLUT
X	0	1	0	Masked Video Data
X	1	1	X	Video Pixels Bypass CLUT
0	X	0	1	Even Address Only (Graphics)
X	0	1	1	Odd Address Only (Video)
1	1	X	X	CLUT Not Used at All

When writing to the CLUT, the most significant byte of the data word corresponds to the address, and the least significant 24 bits are the YUV data (least significant to most significant, respectively). An index register is used to allow the 6-bit address to be mapped to an 8-bit number. (Refer to Chapter 4 for more information.) By resetting the 82750DA Disable bit, it is possible to make the CLUT look like the reduced entry color table on the 82750DA.

The following paragraphs summarizes the possible bit/pixel modes, using the LSB of Y or U switching ability and the various graphics and video bypass modes. Note that there are modes where the LSB of Y or U are not used to switch between graphics and video.

8-BIT/PIXEL GRAPHICS MODE

This is the graphics-only mode, in which the 8 bits are used as inputs to all three color tables. This makes the color maps look like a single, 256 x 24-bit CLUT and allows 256 unique colors from a palette of 16 million to be available at any given time. If the Graphics Pass bit is asserted, the CLUT will be bypassed and the 8-bit values of the Y, U, and V channels will be input to each channel of the converter matrix.

8-BIT/PIXEL VIDEO MODE

When used with subsampled VU information from the interpolator, the 8 bits are actually a luminance value. The Y portion addresses the Y color table, V the V color table, and U the U color table. By using the color table, a one-to-one mapping exists, allowing non-linear transformations to be applied to the pixel data to enhance the quality of the reconstructed image. By asserting the VIDEOPASS bit in the Miscellaneous Control register, the color table can be bypassed.

8-BIT/PIXEL MIXED MODE

In the 8-bit/pixel mixed mode the LSB of Y or U is used as a switch flag to change the index to the color tables. When the switch flag is set to a one, the Y value corresponds to a luminance value, and the VU values are the chrominance information obtained from the VU interpolator. In this case each video component is used as an address to its corresponding CLUT as described above. When the switch flag is set to a zero, the VU values are not used and the Y value is used as the address to all color tables. These pixels are treated the same as in the 8-bit/pixel graphics mode.

In this mode the applications programmer must ensure that the proper information has been loaded into specific areas of the color maps. For example, all the video pixels will use the odd address values. By restricting the address used in the graphics and video mode, two unique maps may coexist in the tables. One map is used for non-linear transformations on video data, and the other for graphics color lookup table applications.

As illustrated above, the CLUT can be bypassed by asserting either or both of the bypass controls.

PSEUDO 16-BIT/PIXEL GRAPHICS MODE

In the pseudo 16-bit/pixel graphics mode each 32-bit data word is made up of two, 16-bit pixel words. The 82750DB processes each 16-bit pixel word, so that the least significant 8 bits correspond to pixel information, and the most significant 8 bits are used as alpha information. The 82750DB uses the lower 8 bits as inputs to all three color tables. This makes the color maps look like a single, 256 x 24-bit color table. If the Graphics Pass bit is asserted, the CLUT will be bypassed and the 8-bit values of the Y, U, and V channels will be input to each channel of the converter matrix.

PSEUDO 16-BIT/PIXEL VIDEO MODE

When used with subsampled VU information, the least significant 8 bits of the pixel word are actually a luminance value. The most significant 8 bits are used as alpha information. The VU information is generated by the 82750DB interpolator. Each of the color maps uses the corresponding 8-bit video component as an addess. By asserting the Video Pass bit in the Miscellaneous Control register, the color table can be bypassed.

PSEUDO 16-BIT/PIXEL MIXED MODE

In this mode the LSB of Y or U is used as switch flag to change the index to the color tables. When the LSB of Y or U is set to a one, the lower 8-bit value corresponds to a luminance value, and the V and U values are the chrominance information. In this case, each video component of the 82750DB is used as a colormap address as described above. When the LSB of Y or U is set to zero, the V and U values from the interpolator are not used, and the Y value is used as the address to all color tables.

16-BIT/PIXEL GRAPHICS MODE

The 16-bit pixel word is broken up on the 82750DB to yield 6 bits of Y, and 5 bits each of V and U. The Y bits are the least significant, and the U bits are the most significant. These values are then padded with zeros in the lower order bits, to obtain an 8-bit word for each pixel component. Each component addresses its respective CLUT. However, the Y channel may access only 64 unique locations, and 5-bit resolution for VU restricts them to 32 unique locations each. The address range may be extended by using the colormap mask registers to add 2 bits of precision in the least significant bits for Y and 3 least significant bits each for VU channels. This allows the programmer to access all the entries in the color table by reprogramming the MASK DATA and MASK SET registers during the blanking interval.

16-BIT/PIXEL VIDEO MODE

This mode works like the 8-bit/pixel video mode described above, except that the 82750DB has processed the information so that the Y channel contains the least significant 8 bits of the 16-bit data word. The V and U information is generated by the VU interpolator. If the SPLITCLUT mode is selected, the LSB of the address is forced to an odd entry in the three color tables.

16-BIT/PIXEL MIXED MODE

When the switch flag is zero, the graphics mode is selected and the inputs to the CLUT are the respective YUV data in the 6-5-5 format. These pixel values are extended by using the colormap masking registers. When the switch flag indicates the video mode, the lower 8 bits of the 16-bit pixel word and the VU values obtained from the interpolar are input to their respective CLUTs. If the SPLITCLUT mode is selected, the LSB of the address is forced to either an odd or even entry in the three color tables, depending on whether the data is video or graphics information.

32-BIT/PIXEL GRAPHICS MODE

Eight bits each of Y, U, and V are used as addresses to each segment of the color table. Since the size of the addressable color space is not increased, the advantage of using the color map is for special effects or gamma correction. The most significant 8 bits of the 32-bit data word are used for the alpha channel data. If the Graphics Pass bit is asserted, the CLUT will be bypassed and the 8-bit values of the Y, V, and U will be input to each channel of the converter matrix.

32-BIT/PIXEL VIDEO MODE

The Y channel contains the least significant 8 bits of the 32-bit data word. The U and V information is generated by the VU interpolator. The YUV channels are input to their respective color tables. The size of the addressable color space is not increased, but this can be used to take advantage of a non-linear transformation, which may aid in the decompression process. The most significant 8 bits of the data word are used for the alpha channel data.

32-BIT/PIXEL MIXED MODE

When the switch flag is zero, the graphics mode is selected, and the inputs to the CLUT are the respective 8 bits each of YUV data. These pixel values may be masked by using the colormap mask data and mask set registers. When the switch flag indicates the video mode, the lower 8 bits of the pixel word and the VU values obtained from the interpolator are input to their respective CLUTs. If the SPLITCLUT mode is selected, the LSB of the address is set to either an odd or even entry in the three color tables, depending on whether the data is video or graphics information. The most significant 8 bits of the data word are used for the alpha channel data.

Y Interpolator

The Y Interpolator performs a 2X horizontal linear interpolation on each line of Y values. When Y interpolation is enabled, the internal pixel clock is twice the frequency of PIXCLK output.

NOTE:

If Y interpolation is enabled, then only the integer values of pixel times greater than 1X may be used.

The interpolation may be separately controlled for both video and graphics pixels, via the Viden and Gren bits (bits 12 and 11) of the General Control register. A video pixel is defined as one generated using VU interpolated values. A graphics pixel does not use the VU interpolator. The effects of setting the control bits, the 82750DB enable flag, and video/graphics pixel switch (V/G Switch) on the output of the interpolator are summarized in Table 2-4.

Because of the asymmetric nature of the internal pixel clock used on 82750DB, the number of T-cycles between successive Y pixels varies depending on the programmed pixel width. When enabled, there is a pipeline delay through the Y Interpolator equal to the number of T-cycles between each internal pixel clock.

When the interpolator is bypassed as described above, there is a fixed delay through this block. The V and U data are delayed by one pixel clock to allow the chroma data to line up with the luminance data. Other control signals, such as the register address byte (most significant byte of the 32-bit data word read from VRAM), the pixel clock, horizontal and vertical active displays, composite blanking, and register load enable signals are also delayed by one pixel clock in order to line up with the YUV data. The programmer must ensure that the active display timing is programmed to take the appropriate delay through the Y Interpolator into account.

Table 2-4. Control Bit Settings and Resulting Interpolator Output

82750DB Enable	Viden	Gren	V/G Switch	Result
0	X	X	X	Interpolator Bypassed
1	0	0	X	Interpolator Bypassed
1	0	1	0	Interpolate Graphics Pixel
1	0	1	1	Do Not Interpolate Video Pixel
1	1	0	1	Interpolate Video Pixel
1	1	0	0	Do Not Interpolate Graphics Pixel
1	1	1	X	Interpolate Both Video and Graphics Pixels

Cursor

Hardware support for a 16 x 16-pixel cursor has been included on the 82750DB. The cursor is capable of providing sharp color transitions, when using subsampled VU bitmaps. Software intervention is minimized, leaving the host with more processing cycles to perform other operations.

Under normal operation, the XY starting display position of the cursor is loaded into the Cursor Control register during a 82750DB register load. On the display line corresponding to the Y start position, the

cursor is displayed when the X starting position (specified in periods of the input frequency to the 82750DB, herein referred to as T-cycles) is reached. On the following 15 lines, the cursor will be displayed at this X position every line, for both interlaced and non-interlaced displays.

A normal 82750DB register transfer is used to load the entire 16 x 16 x 2 bits (16 words of 32 bits each) of cursor data. During this register transfer, the cursor data is distinguished from normal register data by placing the Cursor Control register immediately before the 16 words of cursor data. When the 82750DB loads the Cursor Control register, it will interpret the next sixteen 32-bit words of register data as the cursor bitmap, and will disable the other registers on the 82750DB from decoding the address field of the 32-bit data word. (The checksum of the 82750DB register data is still performed during the loading of the cursor bitmap data.) The cursor bitmap will be loaded a line at a time, starting at line zero and continuing in sequential order to line 15. Each line in the cursor map actually contains sixteen 2-bit cursor pixels, with the two least significant bits corresponding to the first cursor pixel in that line, and the two most significant bits corresponding to the 16th cursor pixel on that line. Each 2-bit pixel may select one of the three Cursor Color registers or transparency, according to the format indicated in Table 2-5.

Table 2-5. Cursor Color Registers

Cursor Pixel	Output
00	Transparency (Cursor Pixel Not Displayed)
01	Cursor Color Register 1
10	Cursor Color Register 2
11	Cursor Color Register 3

Three 24-bit color registers that hold the color information for the cursor may be written to at any time during the register load. The cursor may be loaded any time during the blanking intervals of the display. For displays that do not program the cursor during the display, the cursor bitmap may be loaded during the vertical blanking interval.

When the T-cycle count equals the value programmed into the X start position of the Cursor Control register, the first cursor pixel can be displayed. Each 2-bit cursor pixel will select one of the three Cursor Color registers or transparency. The 24-bit output of one of the three color registers (or the actual display pixel data if transparency is used) is input to the YUV converter.

The cursor bitmap length is 16 lines, and the width is 16 pixels. Although the length of the cursor may be changed dynamically by chaining register loads to update the cursor map, the size of the cursor is dependent on the type of display. For interlaced displays, each line of cursor data will appear on the same line of each field. This results in a cursor of 16 x 32 pixels. For non-interlaced displays, the same line of cursor information will appear on the same line every field. The cursor in this case will be 16 x 16 pixels. The size of the cursor may be doubled independently in the horizontal and/or vertical direction by setting the 2X Horizontal Cursor or 2X Vertical Cursor bit in the General Control register. In this case, no new data is loaded into the cursor map; the data is just replicated in the corresponding dimension. Table 2-6 summarizes some of the possible cursor sizes. Note that by loading the cursor bitmap with different data at the start of every field, cursor sizes not listed below may be achieved.

Table 2-6. Cursor Sizes

2X Horz. Cursor	2X Vert. Cursor	Display	Cursor Size (in Pixels)
Off	Off	Interlaced	16 x 32
On	Off	Interlaced	32 x 32
Off	On	Interlaced	16 x 64
On	On	Interlaced	32 x 64
Off	Off	Non-Interlaced	16 x 16
On	Off	Non-Interlaced	32 x 16
Off	On	Non-Interlaced	16 x 32
On	On	Non-Interlaced	32 x 32

There is a complex relationship between the cursor and the pixel data especially when using non-integral divisors of the pixel clocks. Since the pixel data output from the 82750DB pixel path always changes coincident with the rising edge of the clock, the cursor start position must be positioned on the rising edge of any period of the pixel clock. The programmer must enforce the corresponding restrictions on the start and stop position of the cursor.

YUV to RGB Converter

The following equations give the theoretical relationship between analog RGB components, R, G, B, and analog YUV components, Y, U, V.

$$Y = 0.298822\,R + 0.586816\,G + 0.114363\,B \quad (1a)$$

$$V = R - Y = 0.701178\,R - 0.586816\,G - 0.114363\,B \quad (1b)$$

$$U = B - Y = -0.298822\,R - 0.586816\,G + 0.885637\,B \quad (1c)$$

where: $0.0 < G, R, B < 1.0$

$0.0 < Y < 1.0$

$-0.701 < V < +0.701$

$-0.886 < U < -0.886$

Solving for G, R, B, we can obtain the inverse relationship:

$$G = Y - 0.509228\,V - 0.194888\,U \quad (2a)$$

$$R = Y + V \quad (2b)$$

$$B = Y + U \quad (2c)$$

where: $0.0 < G, R, B < 1.0$

$0.0 < Y < 1.0$

$-0.701 < V < +0.701$

$-0.886 < U < +0.886$

The luminance channel for the YUV inputs is presumed to swing between 0.0V and 1.0V. However, the chroma components do not and need to be normalized to a 0V to 1V range. The offset binary encoding used to obtain unsigned numbers must also be accounted for. This encoding should center the V and U inputs at the midpoint of the voltage range. The equations for the normalized version of Y, V, and U (Y′, V′, and U′ respectively) are:

$$Y' = Y \quad (3a)$$

$$V' = \frac{0.5V}{0.701} + 0.5 \quad (3b)$$

$$U' = \frac{0.5U}{0.886} + 0.5 \quad (3c)$$

where: $0.0 < Y', V'\ U' < 1.0$

$0.0 < Y < 1.0$

$-0.701 < V < +0.701$

$-0.886 < U < +0.886$

When converting the normalized analog values Y′, V′, U′ to digital y, v, u values, the D.C. offset and conversion ranges are compatible with the CCIR 601 standard for digital video. The ranges for the components and the corresponding Digital to Analog equivalent equations are given below:

$$y = (235 - 16)\,Y' + 16 \quad (4a)$$

where: $16 < y < 235$

$$v = (240 - 16)\,V' + 16 \quad (4b)$$

where: $16 < v < 240$

$$u = (240 - 16)\,U' + 16 \quad (4c)$$

where: $16 < u < 240$

Substituting the normalized analog voltages of Equation 3 into Equation 4, we obtain the digital version of the input data, used in the DVI™ Technology system:

$$y = (219)\,Y + 16 \quad (5a)$$

$$v = \frac{112V}{0.701} + 128 \quad (5b)$$

$$u = \frac{112U}{0.886} + 128 \quad (5c)$$

where: $0.0 < Y < 1.0$

$-0.886 < U < 0.886$

$-0.701 < V < 0.701$

$16 < y < 235$

$16 < v, u < 240$

By solving equations 5 for Y, U, V, and substituting into Equation 2, we get the relationship between analog R, G, B and the digital DVI y, u, v data:

$$G = 0.004566\,y - 0.003187\,v - 0.001541\,u + 0.532242 \quad (6a)$$

$$R = 0.004566\,y + 0.006259\,v - 0.874202 \quad (6b)$$

$$B = 0.004566\,y + 0.007911\,u - 1.085631 \quad (6c)$$

where: $0.0 < R, G, B < 1.0$

$16 < y < 235$

$16 < v, u < 240$

If the inputs of the Digital to Analog Converter are scaled to accommodate the nominal input range of 0 to 219, we obtain the following relationship between the inputs to the DVI Technology system, (y, v, u) and inputs to the Digital to Analog Converters (r, g, b). Note that all out of range RGB values (> 255 or < 0 due to excursions in the inputs) are clipped to 255 or 0.

$$g = y - 0.698001\,v - 0.337633\,u + 116.56116 \quad (7a)$$

$$r = y + 1.370705\,v - 191.45029 \quad (7b)$$

$$b = y + 1.732446\,u - 237.75314 \quad (7c)$$

where: $16 < y < 235$

$16 < v, u < 240$

$0 < g, r, b < 255$

By substitution of Equation 5 into Equation 1, and by converting G, R, and B to digital values, we can obtain the inverse relationship of Equation 7:

$$y = +0.298822\,r + 0.586816\,g + 0.114363\,b + 16 \quad (8a)$$

$$u = -0.172486\,r - 0\,338721\,g + 0.511206\,b + 128 \quad (8b)$$

$$jpv = +0.511545\,r - 0.428112\,g - 0.083434\,b + 128 \quad (8c)$$

where: $16 < y < 235$

$16 < v, u < 240$

$0 < g, r, b < 255$

Output Equalization

The units on the 82750DB process the pixel information at the operating frequency of the chip. If the output pixel rate is not equal to the maximum frequency, the units have null states during which processing is suspended. This type of operation is necessary on the 82750DB because of the large amount of pipelining. Table 2-7 gives the pattern of T-cycles on the 82750DB during which processing is active, according to the programming shown in Table 4-2.

The pixel information must be output at a rate that is some sub-multiple of the operating frequency. The divisor is programmed by the user, and may be from 1 to 12 times slower than the period of FREQIN, in increments of ½. Divisors of 13 and 14 are also programmable. Because non-integral divisors are used, it is necessary for the 82750DB to output different information on both phases of FREQIN. This is illustrated in Figure 2-9, which uses a 2.5 divisor for the clock. Notice that the pixel clock output (PIXCLK) transitions fall alternately on the active and inactive phase of the input frequency, while the internal pixel clock transitions always occur on the active phase. Also note that PIXCLK does not have a 50% duty cycle.

The equalizing logic derives a clock that has a period equal to the programmed pixel rate, providing an edge to sample the output information. This allows the Digital to Analog Converter to directly sample the output of the pixel data path before performing the analog conversion.

Table 2-7. 82750DB Active T-Cycle Patterns

Pixel Time (T-Cycles)	Pattern Of Internal Pixel Clock
1	Always On
1.5	1 On/1 On/1 Off
2	1 On/1 Off
2.5	1 On/1 Off/1 On/2 Off
3	1 On/2 Off
3.5	1 On/2 Off/1 On/3 Off
4	1 On/3 Off
4.5	1 On/3 Off/1 On/4 Off
5	1 On/4 Off
5.5	1 On/4 Off/1 On/5 Off
6	1 On/5 Off
6.5	1 On/5 Off/1 On/6 Off
7	1 On/6 Off
7.5	1 On/6 Off/1 On/7 Off
8	1 On/7 Off
8.5	1 On/7 Off/1 On/8 Off
9	1 On/8 Off
9.5	1 On/8 Off/1 On/9 Off
10	1 On/9 Off
10.5	1 On/9 Off/1 On/10 Off
11	1 On/10 Off
11.5	1 On/10 Off/1 On/11 Off
12	1 On/11 Off
13	1 On/12 Off
14	1 On/13 Off

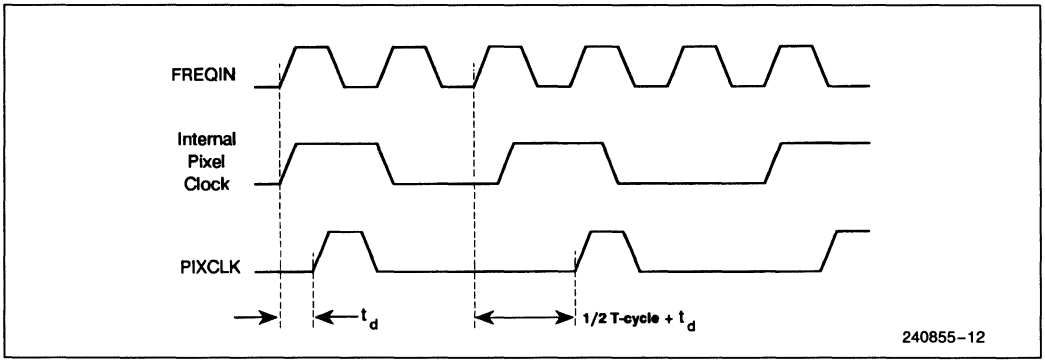


Figure 2-9 Divide by 2.5 Pixel Clock

Digital to Analog Converters

The Digital to Analog Converters (DACs) take three channels of video information output from the pixel data path, converting it from 8-bit digital values to analog voltage levels typically between 0V and 1V. The conversion is monotonic, and a pixel clock is used to derive a two-phase clock internal to the DAC. The data is sampled from the output of either the pixel path, or the YUV to RGB matrix on the rising edge of the internal active phase of this clock.

The analog outputs of the triple DAC are referenced to an external current source, which must be connected to the IREFIN pin. The current source actually sinks a current, which is typically 706 μA. All the analog outputs are scaled by this current reference. The value of the analog output full scale current is as follows:

$$Ifs = Iref * \frac{255}{18}$$

where: Iref is the magnitude of the reference current.

The output voltage generated at full scale is:

$$Voutfs = Ifs * Rext$$

where: Iref is the magnitude of the reference current.

A typical output load for the analog outputs (RV, BU, GY) is 100Ω. The speed of the DAC analog output rise and fall times is determined by the time constant:

$$Rext * (Cext + Cout)$$

where: Cext is the external capacitance applied and Cout is the intrinsic capacitance of an analog output.

For high performance the objective would be to minimize Rext and Cext. The voltage Voutfs can be determined by any combination of Ifs and Rext, but must not exceed 1.5V. In addition Ifs must not exceed 15 mA. The analog outputs must go through an external buffer to drive doubly-terminated 75Ω coax line.

Table 2-8 lists pins which are used to configure the triple DAC.

Table 2-8. Digital To Analog Converter Pins

Signal	Description
IREFIN	Analog Current Reference. Must Be Decoupled to AVCC.
VGCS	Internal Voltage Reference. Must Be Decoupled to AVCC.
AV_{CC}	Analog Power
AV_{SS}	Analog Ground
GY, RV, BU	Analog Pixel Outputs
DISDIG	Disable Digital Outputs

NOTE:

The digital video outputs must be disabled by setting DISDIG high whenever the analog outputs are used. Otherwise the A.C. and D.C. characteristics of the DAC are not guaranteed.

3.0 HARDWARE INTERFACE

82750DB Reset Operation

Upon power-up, the 82750DB is in an indeterminate state and must be reset. The RESETB# signal asserted by the host processor is sampled on the rising edge of FREQIN. The 82750DB will enter the reset state a maximum of four cycles after RESETB# is sampled. The 82750DB will request the 82750PB to generate VRAM refresh cycles by asserting a REFRESH code on the VBUS for 16 T-cycles. This code is repeated every 256 T-cycles, until RESETB# is negated.

NOTE:

The RESETB# input is an edge-triggered input. After power-up, the host processor must set the RESETB# input low for a minimum of ten T-cycles in order to reset the 82750DB. The host must then set the RESETB# input high to start normal operation.

When the RESETB# input is released, a Start of Vertical Field command (VODD) is sent for 16 T-cycles to the 82750PB via the VBUS. This code is immediately followed by a Register Transfer Request command (REGX) that is held for 256 T-cycles. This 256 T-cycle wait assures that the 82750PB has ample time to honor the 82750DB register transfer request. The register data is then read into the 82750DB from the serial port of the VRAMs at a rate that is equal to 1/3 of the operating frequency. If the register transfer does not terminate after 256 T-cycles, the 82750DB will automatically stop the transfer, send an 82750DBSD code to the 82750PB, and re-enter the reset state.

During this register transfer, and on all subsequent register transfers (programmed or automatic), the 82750DB performs a vertical checksum on the register data. The last 32-bit word read in during a register transfer is the user-generated checksum of that register data. If the 82750DB-generated checksum error does not match the user-generated checksum, the 82750DB sends a SHUTDOWN code to the 82750PB via the VBUS, and will automatically re-enter the reset state. The 82750DB will remain in the reset state until the RESETB# input is toggled by the host processor. Any VRAM requests or control signals programmed to occur during this time will be ignored.

Normal programmed operations start after the first successful register load. Frame timing will start at the beginning of a horizontal line and at the beginning of the first field sometimes referred to as line 1 of field 1. There will not be a horizontal sync pulse on the first line after reset, but HSYNC will be generated on every line thereafter. All horizontal and vertical programming parameters as well as scheduling of any transfer requests and control information to be sent on the VBUS must be set up by the user during the first register load. Included in the control information are parameters for the 82750PB to refresh the VRAM. Refresh must occur on every line. This requires that the line rate of the 82750DB must be at least 4 kHz to guarantee that enough refresh cycles are generated. Additional register transfers (up to one per line) may be programmed to occur on any line during the field. As a result of this transfer display characteristics and programming parameters may be changed.

After the first field automatic register transfers will occur on the second line of each subsequent field. Note that all register transfers will occur at 1/3 of the operating frequency of the 82750DB, unless the 1X or 1/2X SCLK mode has been programmed by the user.

Input/Output Transformation

In general, the control outputs, including the sync signals, are delayed by pipelining effects from their corresponding inputs. If the output sync signals are taken as the time base, the first pixel in a line is actually fetched by an SCLK that is up to 19 T-cycles before its corresponding PIXCLK. Some later pixels may be delayed by an additional number of T-cycles, depending upon bits/pixels, pixel timing, and whether Y interpolation is enabled.

Outside of the active display region and before the blanking output is asserted, border pixels are output. Where the blanking region has been entered and the display is not active, the output is the value contained in the Blanking Color register.

Pixel handling in the active region is defined by three parameters:

1. The bits/pixel parameter.
2. Whether VU interpolation is in effect or not.
3. If the 82750DB Enable bit has been selected.

VU interpolation is in effect for a given pixel if:

1. The VU interpolator is turned on (VU sample load set to non-zero load value),

   AND

2. VU interpolation display is permitted (VU interpolation display operations bit equals 1),

   AND

3. One of the two following conditions is met:

   a. Either the interpolation is unconditional,

      OR

   b. The controlling Y or the controlling U sample for this pixel has a least significant bit of 1.

The value of the alpha output may come from one of the following three sources:

1. It may be explicitly coded into the pixel data (32-bit/pixel and pseudo 16-bit/pixel with Alpha modes only).
2. It may be output from one of two programmable registers, Alpha0 and Alpha1.
3. During the portion of the display when the border is active, the 8 most significant bits of the Border Alpha register may be output.

Table 3-1 illustrates how the Alpha outputs are selected.

Table 3-1. Selecting Alpha Outputs

Alpha Enable	Alpha Trap Select	Alpha Output
0	X	Alpha0 Register
1	0	Alpha0 Register (8, 16 bpp)
1	0	MS Byte of Pixel (32, Pseudo 16 bpp)
1	1	Trap Match = 0, Alpha0 Register
1	1	Trap Match = 1, Alpha1 Register

Genlocking on the 82750DB

The genlocking algorithm on the 82750DB uses horizontal and vertical resets from an external device. When the Genlock bit in the Miscellaneous Control register is off, the 82750DB will ignore all horizontal and vertical sync pulses from the digitizer board. The 82750DB will resync itself when the programmed end of line count is received. This allows the user to turn off genlock without having to worry about the state of the input video.

When the Genlock bit is set to one, the 82750DB will use the external resets to reset its internal horizontal and vertical sync counters. In this case, the width of the active line is determined by the HRESET# signal, and the length of the field is governed by VRESET#. The programmed values for these registers will be ignored. As shown in Figure 3-1, when asserted VRESET# and HRESET# are effected just after the third falling edge of FREQIN. VRESET# has no effect on the 82750DB if the first half of the first line of an odd field or the second (and only) half of the first line of an even field is already in progress. HRESET# has no effect on the 82750DB if it occurs during the programmed first half of the line. The user may decrease the effect of jitter by reducing the "window" during which the vertical reset signal is supposed to occur. This can be done by scheduling a register load to occur after the active display time has ended, thereby decreasing the programmable active window to a size acceptable for the video source. When VRESET# is received during this reduced, programmed active window, the 82750DB is reset to an even vertical field. When VRESET# occurs at any other time in the horizontal scan line, the 82750DB is set to an odd field.

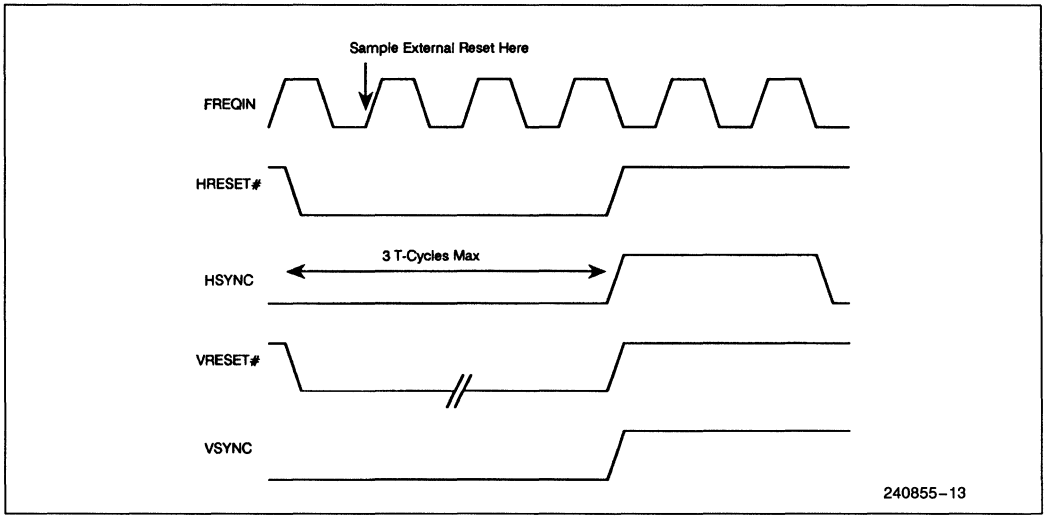


Figure 3-1. Horizontal and Vertical Reset Timing

Digitizing Images with the 82750DB

Digitizing is enabled by setting the Digitize Enable bit in the Miscellaneous Control register. Note that enabling the digitize mode does not automatically enable genlocking. The Genlock bit must be set separately, if it is required. When digitizing, the 82750DB is used to shift digitized data into the VRAM shift registers, and then transfer this data into the VRAM array.

The 82750DB also provides an external "digitizer window" signal, FCO. This signal defines the vertical active region that the digitizer enabled. Typically, the user sets up the display parameters to reflect the "window" of the display to be digitized. The horizontal and vertical active window size can be selected by programming the Active Start and Stop registers. FCO is derived from the Vertical Start and Stop registers, and is used to enable the digitizer to drive the VRAM bus. During the programmed vertical blanking interval the FCO signal will be negated, and therefore, the digitizer is prohibited from driving the VRAM bus. This will allow data to be read from the VRAM serial data bus during the automatic register transfer that is performed at the start of the field. Note that it will still be possible to program the 82750DB to digitize during the vertical blanking interval, in order, for example, to capture time codes from a VCR.

When capturing and displaying NTSC data during the horizontal blanking interval of the first display line, a WRDIGINP command is sent on the VBUS to the 82750PB. (Refer to Figure 3-2.) Recall that there is a 5-line vertical pipeline delay through the 82750DB. If the first display line is programmed to be n, the first display line will occur at n + 5. Similarly, if the last line is programmed to be m, then the last display will be line m + 5. The WRDIGINP VBUS code causes a dummy write transfer cycle that places the VRAMs in the write mode. The 82750PB then sets the bitmap pointers to the first line's address (L0). This code is immediately followed by another WRDIGINP command that causes the 82750PB to perform a write transfer cycle at the L0 address. Since no digitized data has been read in, invalid data is loaded into row L0 of the VRAM array.

During the active display of the first display line, the 82750DB provides shift clocks at the programmed pixel rate. The digitized data is shifted into the VRAMs while the user-programmed horizontal active window is active. During the horizontal blanking interval of the next line, the 82750DB sends a WRDIGI code to the 82750PB, thereby transferring the L0 data from the shift register to the VRAM array at the L0 address. The 82750PB performs a pitch calculation, pointing it to the L1 row. After the WRDIGI

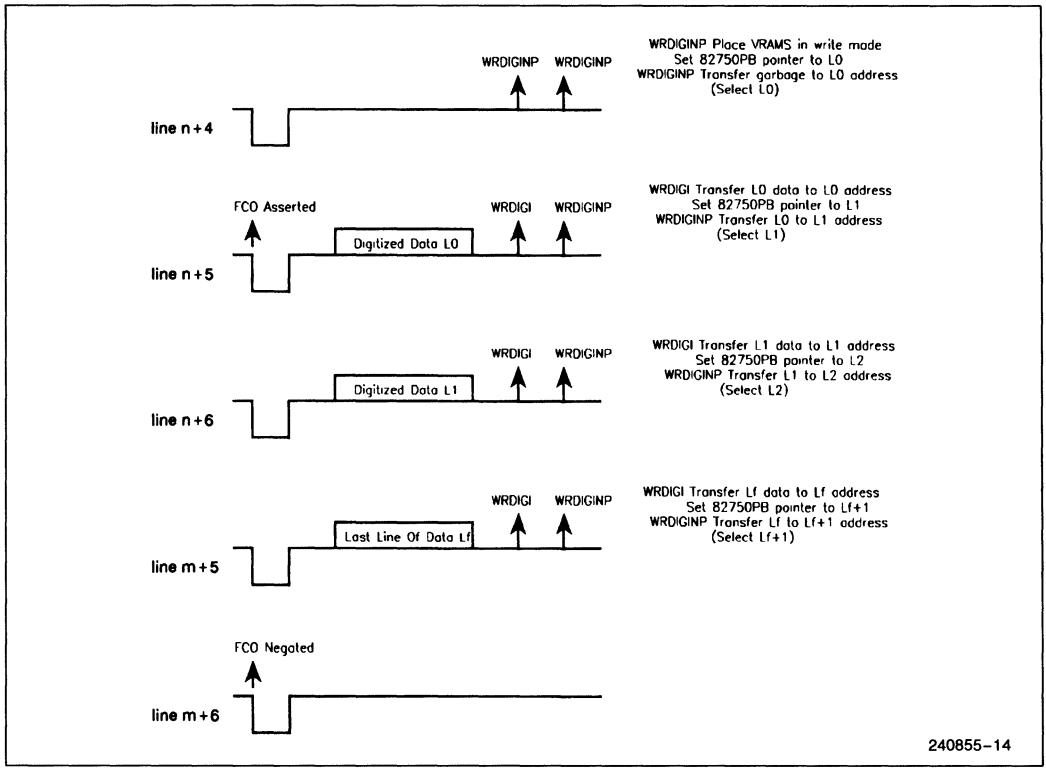


Figure 3-2. Digitizing Example

transfer has finished, the 82750DB issues a WDIGINP command to the 82750PB that performs a write transfer cycle at L1 address. This will write the L0 data into the L1 address. The next line the L1 row will be written over with L1 data. This same procedure continues for the entire active display, until the last active line is reached (m + 5). A final pair of WRDIGI and WRDIGINP codes are sent to the 82750PB to load in the last line of data. At the start of horizontal sync of the next line, the FCO signal will be negated.

The purpose of the WDIGINP may not be apparent at first glance. This signal ensures that the correct data is written into the last selected VRAM address. This is necessary when crossing the physical boundaries of VRAM memory.

When the 82750DB is genlocked, the digitizing device must also provide the HRESET# and VRESET# signals. The device must ensure that VRESET# is never asserted during the start of the line. This allows a register transfer (which shortens the active display and is required for digitizing) to complete before the start of a field register transfer. The vertical sync pulses are buffered, so the start of the field transfer request can be honored immediately after the previous transfer request is finished.

Also, captured NTSC data may be displayed on a VGA-type monitor. This requires the 82750DB to operate at a VGA frequency (approximately 31.5 kHz), which is twice that of NTSC. Each line of captured NTSC data is read into the 82750DB twice. Setting the line replicate bit makes doubling of memory is unnecessary. Figure 3-3 illustrates how the 82750DB operates in such a mode. The Line Replicate, Digitizer, and Genlock bits in the Miscellaneous Control register are assumed to be set to one. During the HBI of the first display line, a dummy write transfer cycle (WRDIGINP) places the VRAMs in the write mode. The 82750PB then sets the bitmap pointers to the first line's address (L0). This code is immediately followed by a WDIGINP command, causing the 82750PB to perform a write transfer cycle at the L0 address. Since no digitized data has been read in, unknown values are loaded into row L0 of the VRAM array.

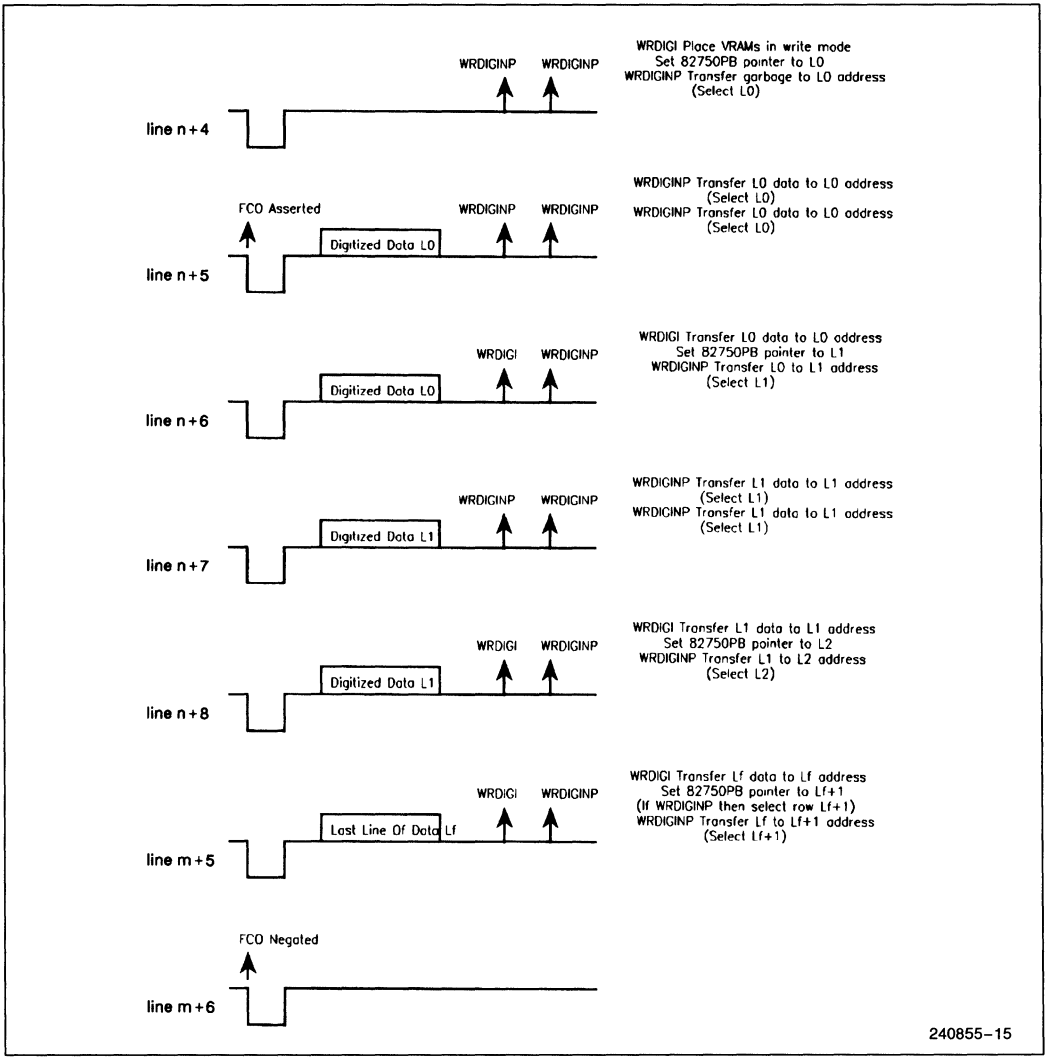


Figure 3-3. Digitizing Example with Line Replicate

At the end of the first line the 82750DB sends two WRDIGINP codes to the 82750PB, thereby transferring the L0 data from the shift register to the VRAM array at the L0 address. The 82750PB does not perform a pitch calculation, so the pointer remains at the address for L0. After the second display line (which has the same data as the first line), a WRDIGI code is sent to the 82750PB that writes the L0 data to the L0 address and updates the bitmap pointer to L1. The WRDIGINP signal immediately following this selects the L1 address. After the third line of data, two WRDIGINP codes that select the L1 address are sent. After the fourth line, (which has the same data as the third line) a write operation is performed to load L1 data into the L1 address, and the 82750PB pointer is updated to address L2. A WRDIGINP code is sent to select the L2 address. This same procedure continues for the entire active display, until the last active line is reached (m + 5). A final pair of WRDIGI and WRDIGINP or two WRDIGINP codes are set to the 82750PB to load in the last line of data. At the start of horizontal sync of the next line, the FCO signal will be negated.

4.0 PROGRAMMING THE 82750DB

Overview

All registers are loaded by the issuance of a REGX command from the 82750DB to the 82750PB over the VBUS. This causes the 82750PB to load a sequence of register values into the VRAM serial output registers from an address designated by a 82750DB register pointer. After the request is granted, a new 82750DB register word is read in with each SCLK. Each 32-bit word consists of a register address in the high byte and register values in the rest of the word. The sequence is terminated by a stop code that corresponds to the address byte being equal to 0xff. A variable number of 32-bit words can be loaded. During reset, if a stop bit is not found within 256 T-cycles, the register transfer is terminated, a SHUTDOWN code is asserted on the VBUS, and the 82750DB returns to the reset state. All transfer requests are terminated at the start of a new field. This ensures that non-terminating register transfers caused by bad register data will be halted.

82750DA/82750DB Register Load Compatibility

Many of the registers that were in the 82750DA have been assigned new addresses in the 82750DB. This was done to allow continuity and grouping of compatible functions in the register space. The 82750DB uses a dual decoding scheme that enables decoding of all of the addresses that were in the 82750DA address space, as well as the new 82750DB register mapping. When the 82750DB Mode Enable bit in the Miscellaneous Control register is zero, the 3 out of 7 register decoding used on the 82750DA will be performed. When this bit is set to a one, the full 8-bit decoding using the 82750DB register addresses will be used. Note that during the first register load after RESETB# is released, the 3 out of 7 register decoding will be used to ensure 82750DA compatibility. Since the control bytes have the same address in both decoding schemes, users who wish to use the new encoding should set the 82750DB Mode Enable bit immediately, so that subsequent register addresses will be decoded correctly. In order to ensure that the 82750DB address decoding is done correctly, the Miscellaneous Control register should be the first register loaded during reset. When the 82750DB Mode Enable bit is zero, the most significant bit of the address will act as a stop bit. If the bit is set, the code 0xff will stop the register transfer. The address field is shown below.

← bit location →	Action
31 . . . 23 . . . 15 . . . 7 . . . 0	
0abcdefgx x	Write register abcdefg (82750DA)
abcdefghx x	Write register abcdefgh (82750DB)
1xabcdefx x	Stop load (82750DA)
11111111x x	Stop load (82750DB)

Pipeline Delay through the 82750DB

The actual horizontal pipeline delay through the 82750DB is dependent on processing elements used to generate the output. If Y interpolation is not used, the pipeline delay is:

Horiz. Pipeline Delay = 16 cycles + SCLK Transfer Timing Delay

Here the SCLK Transfer Timing Delay is 1 for 1X, 2 for 1/2X, and 3 for 1/3X. Horizontal active display, blanking, and the graphics/video switching are delayed by this fixed amount.

If Y interpolation is used, the pipeline delay is:

Horiz. Pipeline Delay = 16 cycles + SCLK Transfer Timing Delay + Integer (Pixel Time)

The integer (Pixel Time) is simply the integer value of the programmed pixel time. For example, if Pixel Time is 2.5, the Integer (Pixel Time) equals 2.

Thus all horizontal parameters, (e.g. horizontal blanking start, active stop) must be programmed to account for the total horizontal pipeline delay. The vertical pipeline delay is fixed at five lines. All vertical parameters must be programmed so that this delay is taken into account.

PROGRAMMING CONSIDERATIONS

The user must ensure that the 82750DB is programmed correctly. Illegal or illogical combinations of display parameters are not corrected in hardware, and may cause the 82750DB to output erroneous display or timing information. The following list highlights some basic guidelines to follow when programming the 82750DB.

1. The maximum rate that data may be read into the 82750DB is determined by the type of memory used. This in turn effects the maximum rate and depth of data that can be displayed. If 32 bits of data can only be read into the 82750DB every two clock cycles, only 16 bits of data may be displayed every clock cycle. The programmer should match the transfer rate (1X, 1/2X, or 1/3X) with the memory speed, and the display pixel rate with the pixel depth and memory bandwidth.
2. When operating in the 82750DA compatibility mode, 82750DB registers must not be programmed. Valid 82750DA register addresses are listed in Table 4-6.
3. Blanking intervals of the display are defined by the non-active programmed time. During this portion of the display, programmed transfers take place. If a transfer does not complete before the start of the active display, it is terminated, and active display data is shifted into the 82750DB at the programmed rate. During horizontal blanking intervals, the user should allow enough time for all programmed register, colormap, and VU data transfers to complete.
4. When digitizing (capturing) images, no other bitmap transfers (e.g., REGX,VU) should be scheduled to occur during the active portion of the field.
5. Active start and stop times should not be programmed to overlap the blanking stop and start times, taking the pipeline delay through the 82750DB into account.
6. Programming the Y interpolation to occur in a non-integral pixel width will cause the Y channel to output incorrect data.

CURSOR REGISTERS

The following registers are used to program the characteristics of the on-chip cursor.

When in the 82750DA mode, these registers are not loaded.

Cursor Control Register 0x5a

31 24	23 12	11 0
01011010	Vertical Position	Horizontal Position

— Horizontal Position in units of T-cycles
— Vertical Position in units of full lines

This register gives the horizontal and vertical position of the cursor. The cursor will extend 16-pixel periods, starting at the prescribed horizontal position, for the next 16 lines. (Or 32-pixel periods for 32 lines if the 2X Cursor Mode bits in the General Control register are set to one.) Receipt of this address also causes the 82750DB to interpret the next sixteen 32-bit words of register data as the 16 x 16 x 2-bit cursor map. This will cause the register address decoding logic internal to the 82750DB to be disabled, and the next 16 words of information will be loaded into the Cursor table. Each 32-bit word will be interpreted as a line (16 pixels) of cursor data, with the two least significant bits corresponding to the first cursor pixel to be displayed.

Cursor Color 3 0x59

31 24	23 16	15 8	7 0
01011001	Blue/U Color	Red/V Color	Green/Y Color

If the cursor is enabled and the 24 bits of data in this register are selected, the data will be sent directly to the YUV conversion matrix during active display. The bits should be programmed as RGB values when the YUV to RGB matrix is not being used.

Cursor Color 2 0x58

31 24	23 16	15 8	7 0
01011000	Blue/U Color	Red/V Color	Green/Y Color

If the cursor is enabled and the 24 bits of data in this register are selected, the data will be sent directly to the YUV conversion matrix during active display. The bits should be programmed as RGB values when the YUV to RGB matrix is not being used.

Cursor Color 1 0x57

31 24	23 16	15 8	7 0
01010111	Blue/U Color	Red/V Color	Green/Y Color

If the cursor is enabled and the 24 bits of data in this register are selected, the data will be sent directly to the YUV conversion matrix during active display. The bits should be programmed as RGB values when the YUV to RGB matrix is not being used.

DISPLAY TIMING REGISTERS

Each register has two, 12-bit components, listed with least significant bits first, followed by the 12 most significant bits. Horizontal timing is measured in units of T-cycles (periods of the master clock) from the start of horizontal sync. The register content defines the number of T-cycles that elapse before the event controlled by this register takes place. The exception to this rule is the base counter, which specifies the number of T-cycles/half line. Zero is not an allowable value; use the total number of T-cycles per half line or full line instead. Unused bits should be zero. Sync signals are RESET to initial values as specified for each; "start" means to set to 1, and "stop" means to be reset to zero.

Base Counter 0x56

31 24	23 12	11 0
01010110	# of Lines/Field	# of T-Cycles/Half Lines

— T-cycles/Hal Line in units of T-cycles (Periods of the master Clock)
— Half Lines/Field in units of half lines

As defined by NTSC standards, vertical timing can be measured from the start of a field in one of two ways: either in units of half lines, or in units of full lines. When programmed for an interlaced display, (i.e. an odd number of half lines per field) the start of a field coincides with the start of a line on odd fields and with the midpoint of a line on even fields. In the latter case, for an event that is programmed in full lines, the first half line is ignored, and counting begins with the first full line. With this interpretation, the register content defines the number of half or full lines that elapse before the event controlled by this register takes place. The same may be said for the horizontal component, which is defined by the number of T-cycles/half line. The hardware does not look for nor correct illogical combinations of register settings. The monitor should be protected from damage with external circuitry when debugging is in progress.

All of the internal timing is derived from comparing the programmed values with the values of this register. The horizontal base counter is programmed using the least significant 12 bits. In this case the values loaded into this register should be one less than the desired value. Bits 23 through 12 are used to specify the number of half lines per field.

Sync Stops 0x55

31 24	23 12	11 0
01010101	VSYNC Stop	HSYNC Stop

— HSYNC Stop in units of T-cycles
— VSYNC Stop in units of half lines

This register is used in conjunction with the Sync Start register to specify the widths of the HSYNC and VSYNC outputs. The HSYNC output is asserted at the beginning of every line, when the Base Counter register is one. This start position is not programmable. The number of T-cycles that the HSYNC signal should be active is specified in Bits 11 through 0 of the Sync Stop register. The horizontal component of this register also affects the composite sync, or CSYNC output. In this case, the CSYNC output will be the same as the HSYNC output, except during the vertical sync and equalization interval. In the latter case, the CSYNC output is determined by the Serration and Equalization registers discussed below.

The vertical sync output, VSYNC, may be programmed to start and stop at any time during the field on a half-line interval. Bits 23 through 12 of this register are used to specify the half line on which the VSYNC pulse is to become inactive.

VSYNC Start 0x54

31 24	23 12	11 0
01010100	VSYNC Start	Not Used

— Not used
— VSYNC Start in units of half lines

Blanking Stops 0x53

31 24	23 12	11 0
01010011	Vertical Blank Stop	Horizontal Blank Stop

— HB Stop in units of T-cycles
— VB Stop in units of half lines

The Blanking Start and Stop registers control the composite blanking output (CB). The horizontal blanking start and stop position, in units of T-cycles, can be specified to occur at any time during the line. By the same token, the vertical blanking start and stop positions can be programmed to occur at any half-line interval.

The CB output combines both the horizontal and vertical blanking pulses programmed using these two registers. This information is independent from the HSYNC, VSYNC, and CSYNC outputs, so the user must specify the proper blanking intervals for the monitor that is being used. If the programmer specifies the blanking period to end before the active line starts, or start after the active line has ended, the border color is output. Due to internal pipeline delays on the 82750DB, the values should be one less than desired for VB Start and Stop. For HB Start and Stop subtract the total horizontal pipeline delay.

Blanking Starts **0x52**

31 24	23 12	11 0
01010010	Vertical Blank Start	Horizontal Blank Start

— HB Start in units of T-cycles Resets to 1
— VB Start in units of half lines Resets to 1

Program values one less than desired for VB Start and Stop. For horizontal blanking start, load numbers less than the total horizontal pipeline delay.

Serration Start **0x51**

31 24	23 12	11 0
01010001	Not Used	Serration Start

— SER Start in units of T-cycles Resets to 0
— (not used)

The vertical component of the CSYNC (composite sync) signal is made up of two types of pulses: equalization and serration pulses. The window during which the serration pulses are active, is determined by the VSYNC start and stop positions, as shown in Figure 4-1. When vertical sync (VSYNC) is active, in this case on line 3, the first serration pulse is output on the CSYNC signal. This pulse will start at the T-cycle count specified in Bits 11 to 0 of the Serration Start register. The pulse will end when the half-line count specified in the Base Counter register has been reached. This pulse will be repeated for every half line that the VSYNC output is programmed to be active, regardless of the position in the field. In Figure 4-1, this continues until half line 12, or line 6.

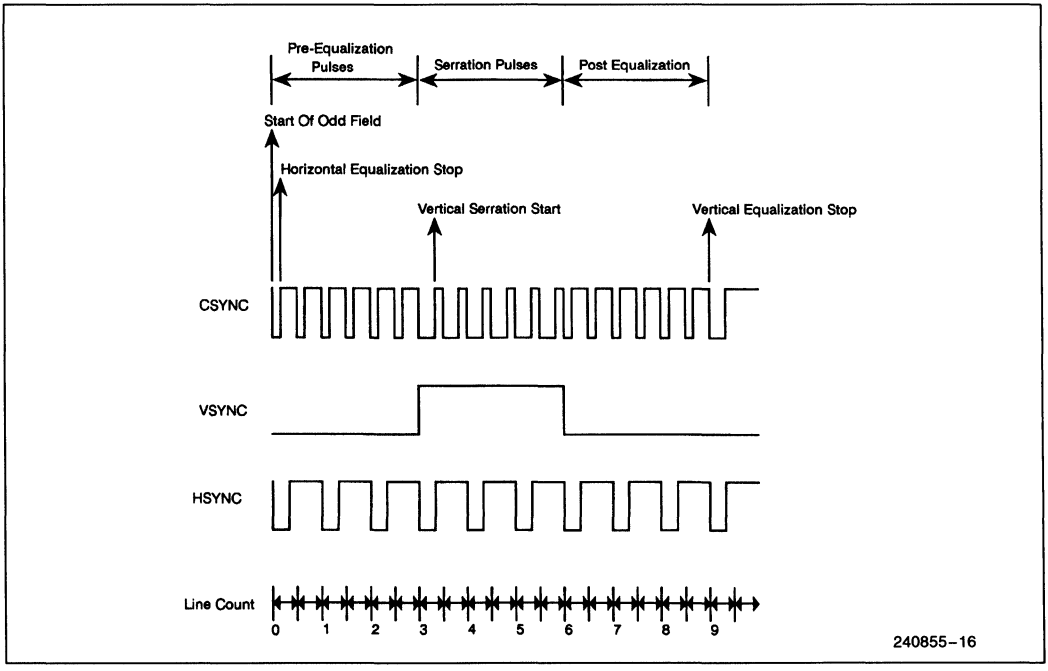
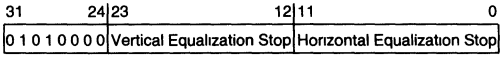


Figure 4-1. Programming the Video Sync Outputs

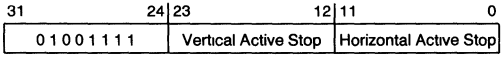
Equalization Parameters 0x50



— EQH Stop in units of T-cycles Resets to 1
— EQV Stop in units of half lines Resets to 1

During the vertical equalizing period, which starts at field-beginning, an equalization pulse is output on the CSYNC signal at the beginning of each half line, as shown in Figure 4-1. The width of this equalization pulse is determined by the value in bits 11 to 0 of this register. The half line on which these pulses are to stop is programmed in bits 23 through 12 of this register. If VSYNC is programmed to occur during the equalization interval (as it is for NTSC type displays), the serration pulses are output on the CSYNC signal.

Active Region Stops 0x4f

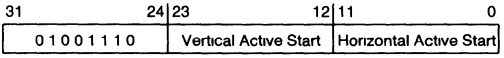


— Actdis Stop in units of T-cycles
— Vertical Stop in units of full lines

The active region window, during which pixels to be displayed are fetched from VRAM, is defined by the Active Region Start and Stop registers. The first display line is actually five lines after the line indicated in the vertical region of the Active Region Start register. The position of the active region on a horizontal line is determined by the horizontal component of the Active Region Start register. Pixels will be fetched from VRAM at a rate determined by the number of bits/pixel and pixel widths. In order for the 82750DB to operate properly, the horizontal width of the active region window must be an integral number of display pixel widths, taking into account the horizontal pipeline delay. Also, the Active Region Start and Stop must fall within a single line boundary, as dictated by the Base Counter register. When the first pixel actually appears at the output of the 82750DB, the output is a function of the processing elements used as discussed above.

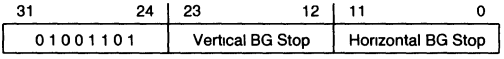
When the active region is over, the border color is output until the programmed blanking time is reached. Both the border and blanking information is output at the transfer rate programmed by the user.

Active Region Starts 0x4e



— Actdis Start in units of T-cycles
— Vertical Start in units of full lines

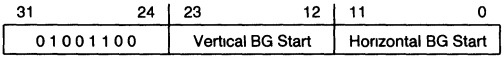
Burst Gate Stop 0x4d



— Horizontal Stop Position in units of T-cycles
— Vertical Stop Position in units of full lines

The Burst Gate Horizontal and Vertical Start and Stop registers allow the user to program a window into which burst can be added. This is useful when modulating the outputs of the 82750DB.

Burst Gate Start 0x4c

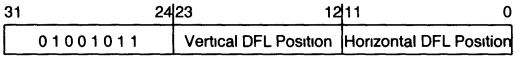


— Horizontal Start Position in units of T-cycles
— Vertical Start Position in units of full lines

VBUS CODE REGISTERS

The following group of registers are used by the programmer to schedule when VBUS transfer or control codes are to be sent to the 82750PB by the 82750DB.

Display Format Load Interrupt 0x4b



— Horizontal Position in units of T-cycles
— Vertical Position in units of full lines

This is the programmable XY interrupt, used by the 82750PB to perform a load of the Shadow Copy registers. This interrupt is sent on the VBUS when the bits 23 to 12 match the current display line position, and bits 11 to 0 match the T-cycle count.

Line Notification Timing 0x4a

31 24	23 12	11 0
01001010	Not Used	Horizontal HLIN Position

— HLIN timing in units of T-cycles
— Not Used

This indicates the position on each line to send a HLINE code on the VBUS. The 82750PB requires this information to keep track of the current display line when drawing graphics.

Refresh and Register Transfer 0x49

31 24	23 12	11 0
01001001	REGX Line Number	Refresh Horizontal Position

— REFRESH horizontal timing in units of T-cycles
— Register Transfer Line number in units of full lines

When the T-cycle count matches the value programmed into bit 11 to 0 of this register, a refresh code is sent to the 82750PB. Since these codes tie up the 82750PB for at least eight 82750PB cycles, the programmer must ensure that no transfer requests are scheduled to occur during this time.

The line number for the next register transfer is specified in bits 23 to 12 of this register. If programmed to occur, REGX will always be the first transfer request sent to the 82750PB, immediately after the end of active display.

COLOR REGISTERS

The following registers specify the state of DBU, DRV, DGY, and ALPHA signals during the field.

Border Color 0x48

31 24	23 16	15 8	7 0
01001000	Blue/U Color	Red/V Color	Green/Y Color

The 24 bits of data in this register are sent directly to the YUV conversion matrix during border time. Border time is defined as the region in which neither active display nor blanking is programmed to occur. The bits should be programmed as RGB values when the YUV to RGB matrix is not being used.

Alpha Register 0x47

31 24	23 16	15 8	7 0
01000111	Border Alpha	Alpha1 Register	Alpha0 Register

The least significant 8 bits are for the ALPHA0 register and are used during blanking and if the alpha trap value is not matched. The next 8 bits are for the ALPHA1 register when the alpha trap value is matched. The most significant 8 bits provide the alpha channel value during the border time.

Blanking Color 0x46

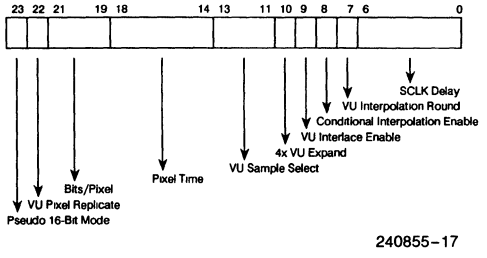
31 24	23 16	15 8	7 0
01000110	Blue/U Color	Red/V Color	Green/Y Color

The 24 bits of data in this register are sent directly through the YUV conversion matrix during the programmed blanking time.

CONTROL REGISTERS

The following registers are used to define the operating modes of the 82750DB.

Pixel Control 0x45



240855-17

Bits 6:0—SCLK Delay

The number "m" of T-cycles from initiation of a transfer request on the VBUS until the first SCLK is asserted by the 82750DB.

Bit 7—VU Interpolation Round

When equal to 0, this bit means truncate during interpolation. When set to one, this bit means round to odd during interpolation.

Bit 8—Conditional Interpolation Enable

When reset to zero, this bit means all values of Y and U are a full 8 bits of precision. When set to one, this bit means the least bit of the Y sample or the U sample controls the switching between VU interpolation and graphics mode.

Bit 9—VU Interlace Enable

Setting this bit to a one causes the interpolator to output different data on the odd and even fields. During the odd field, the odd lines of the interpolation sequence will be output. During the even field, the even lines of the interpolation sequence will be output. Full lines of the programmed number of samples of both the V and U data will be read in during each VU transfer. Setting this bit to a zero will cause horizontally and vertically interpolated data to be output on both fields. Only a full line of either V or U samples will be read in during each transfer request in this mode.

Bit 10—4X VU Expand

When this bit is set to a zero, a 2X expansion in both directions is performed. By setting this bit to a one, a 4X expansion is performed.

Bits 13:11—VU Sample Select

Table 4-1 provides the code and number of V and U samples for bits 13:11.

Table 4-1. VU Sampling

Code	Number of V And U Samples
000	0 Samples for Each V and U
111	32 Samples for Each V and U
110	64 Samples for Each of V and U
101	96 Samples for Each of V and U
100	128 Samples for Each of V and U
011	160 Samples for Each of V and U
010	192 Samples for Each of V and U
001	256 Samples for Each of V and U

Bits 18:14—Pixel Time

Table 4-2 lists the codes and pixel duration for bits 18:14.

Table 4-2. Pixel Times

Code	Duration of Pixel
00001	1.0 T-cycle
00010	1.5 T-cycles
00100	2.0 T-cycles
01000	2.5 T-cycles
10000	3.0 T-cycles
10001	3.5 T-cycles
10010	4.0 T-cycles
10100	4.5 T-cycles
11000	5.0 T-cycles
11001	5.5 T-cycles
11010	6.0 T-cycles
11100	6.5 T-cycles
11101	7.0 T-cycles
11110	7.5 T-cycles
00011	8.0 T-cycles
00101	8.5 T-cycles
00110	9.0 T-cycles
00111	9.5 T-cycles
01001	10.0 T-cycles
01010	10.5 T-cycles
01011	11.0 T-cycles
01100	11.5 T-cycles
01101	12.0 T-cycles
01110	13.0 T-cycles
01111	14.0 T-cycles

Bits 21:19—Bits/Pixel

Table 4-3 provides the code and number of bits/pixel for bits 21:19.

Table 4-3. Number of Bits/Pixel

Code	Number of Bits/Pixel
001	8
010	16
100	32

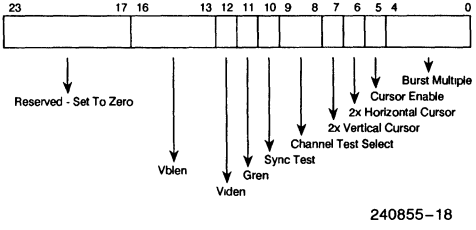
Bit 22—VU Pixel Replicate

When set to one, each pixel generated by the VU Interpolator is held for 2-pixel times. This allows an effective 8X expansion of VU data. This is useful for high resolution applications where the blanking time is not sufficient to support higher VU sample loads.

Bit 23—Pseudo 16-Bit Mode

When set to one and 16 bits per pixel is chosen, the 82750DB is in 16-bit with Alpha mode. Setting this signal to zero while in the 16-bit/pixel mode puts the 82750DB in 16-bit (655) mode.

General Control 0x44



Bits 4:0—Burst Multiple

These bits are used to program a divisor of the FREQIN clock input in order to recover the 3.58 MHz NTSC color subcarrier. The programmed value is the two's complement of the desired divisor. The allowed range of values is 00001 through 11111 which corresponds to divisions of 31 through 1. Note that the 82750DB must be operating at an integer multiple of 3.58 MHz for this to work effectively.

Bit 5—Cursor Enable

When set to one, the hardware cursor will output the cursor data at prescribed intervals if programmed to do so.

Bit 6—2X Horizontal Cursor

When this bit is set to one, and the Cursor Enable bit is set to one, every pixel on each line of the cursor will be replicated once. Thus a cursor that was 16 x 16 pixels will become 32 x 16 pixels.

Bit 7—2X Vertical Cursor

When this bit is set to one, and the Cursor Enable bit is set to one, each line of the cursor will be replicated once. Thus a cursor that was 16 x 16 pixels will become a 16 x 32-pixel cursor.

Bit 9:8—Channel Select

These two bits control which output channel is muxed onto the alpha digital outputs. It allows Y, U, or V data to be available at the alpha channel. The coding is provided in Table 4-4.

Table 4-4. Test Mode Select Coding

Code	Alpha Channel Output
00	Alpha Channel
01	Y Channel
10	V Channel
11	U Channel

Bit 10—Sync Test

This bit must be set to zero for proper operation.

Bit 11—Gren

This is the Graphics Enable bit for the Y Interpolator. When this bit is set to one and the pixel is a graphics pixel, switch is zero, a 2X interpolation will be performed on the pixel.

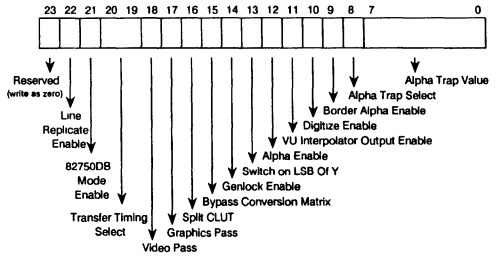
Bit 12—Viden

This is the Video Enable bit of the Y Interpolator. When this bit is set to one and the pixel is a video pixel, switch is one, a 2X interpolation will be performed on the pixel.

Bit 16:13—Vblen

These bits program the T-cycle length of each VBUS code. The VBUS code length will be one T-cycle longer than the programmed value. These bits must have a minimum value of 2, and a maximum value of 15.

Miscellaneous Control **0x43**



240855-19

Bits 7:0—Alpha Trap

Bits 7:0 are 8-bit values used for comparison with the current pixel's Y value, to select one of two programmable alpha values.

Bit 8—Alpha Trap Select

A value of one enables the Y value of the current pixel to be compared with the value in the Alpha Trap register. If the two values match and Alpha has been enabled via the Alpha Enable bit, the contents of the ALPHA1 register are output on ALPHA[7:0]. If the two values don't match and Alpha Enable has been set to one, the content of the ALPHA0 register is output. When Alpha Trap Select is set to a zero in the pseudo 16- or 32-bit mode, the most significant byte of the pixel word is output. When Alpha Trap Select is set to zero in all other modes, the value of the ALPHA0 register is output.

Bit 9—Border Alpha Enable

A value of one enables the eight most significant bits in the ALPHA register to be output. When set to a zero, the ALPHA0 register is output during border time.

Bit 10—Digitize Enable

When this bit is set to a one, the FCO signal will be set to a one, and the transfer codes for bitmaps will indicate that write operations should occur.

Bit 11—VU Interpolator Output Enable

This bit enables VU interpolation data to be displayed. When set to a zero, all pixels are treated as graphic pixels.

Bit 12—Alpha Enable

When set to one, the alpha output is governed by the alpha trap value, as described above. When reset to zero, the contents of the ALPHA0 register is the alpha output in the 8- and 16-bit modes, and the explicit ALPHA data encoded in the pseudo 16- and 32-bit modes.

Bit 13—Switch on LS Bit of Y

When set to one, the least significant bit of Y is used as a Video/Graphics switch in all modes. When reset to zero, the least significant bit of U from the interpolator acts as a switch.

Bit 14—Genlock Enable

This bit enables the genlock mode of the 82750DB. In this mode, receipt of the external HRESET# signal during the second half of a scan line will cause the termination of that scan line. Similarly, receipt of the externally produced VRESET# signal will terminate the field. In both cases, terminate denotes that the proper on-chip signals are produced to signify end of the line and end of the field.

Bit 15—Bypass Conversion Matrix

When this bit is set to a one the YUV to RGB matrix will be bypassed, and the Y, U, and V data will feed directly into the Digital to Analog Converters.

Bit 16—Split CLUT

This bit divides the CLUT into an odd and an even half, depending on the polarity of the Video/Graphics switch. This switch is selectable and may be either the LSB of U from the interpolator or Y from the pixel word. The LSB of the CLUT address is set to one (odd address) if the Video/Graphics switch is one; the LSB of the CLUT address is set to zero (even address) if the Video/Graphics switch is zero.

Bit 17—Graphics Pass

Setting this bit to a one bypasses the CLUT for graphics pixels, even in non-mixed modes.

Bit 18—Video Pass

When set to a one, all video pixels (luminance values associated with sub-sampled UV values) will bypass the color table. For mixed modes, this corresponds to the switch flag having a value of one.

Bit 20:19—Transfer Timing Select

These bits are two-bit codes that select one of three possible transfer shift clock rates. This allows the operating speed of the 82750DB to be tailored to the external memory access time. After RESET, the transfer rate is set to the slowest possible clock rate (1/3X). The programmed rate is used during all non-active display times for transferring data from VRAMs. It also defines the rate that the border and blanking data is output. During active display, the data is read as needed from VRAM using the programmed timing. The coding of these bits is listed in Table 4-5.

Table 4-5. Coding of Transfer Timing Select Bits

Bit 20	Bit 19	Result
0	0	1/3X Transfer (Default)
0	1	1/2X Transfer
1	0	1X Transfer

Bit 21—82750DB Enable

When set to zero, the 82750DB will be the register equivalent of a 82750DA. When set to a one all the features of the 82750DB will be enabled.

Bit 22—Line Replicate Enable

When this bit is set to one, every line in the active display is generated twice. Each new bitmap transfer occurs at half the line rate, with a new VBUS code being used to indicate that a transfer is to take place without the pitch calculation. The VU Interpolator will also duplicate the lines it generates, yielding more time between transfer cycles. This mode is useful for obtaining a 2X increase in vertical resolution without the need for increasing the VRAM transfer bandwidth.

COLOR MAP REGISTERS

The following registers are used to access and control the three 256 x 8-bit Color Lookup Tables.

Mask Data Registers 0x42

31 24	23 16	15 8	7 0
01000010	Blue/U Mask Data	Red/V Mask Data	Green/Y Mask Data

Each of the three 8-bit registers contains the bit pattern used when the corresponding bit in the Mask Set register is asserted.

Mask Set Registers 0X41

31 24	23 16	15 8	7 0
01011001	Blue/U Color	Red/V Color	Green/Y Color

This is a 24-bit register that contains the mask bit pattern for the RGB/YUV color map addresses. When a bit in this register is asserted, the corresponding bit in the address is set to the value defined in the Mask Data registers.

CLUT Index Register 0x40

31 24	23 16	15 8	7 0
01000000	Not Used	Not Used	YUV CLUT Index

The CLUT Index register is an 8-bit register used for loading the color tables. This register maps the user-specified 6-bit color map address into an 8-bit address. A logical OR operation is performed between the 6-bit address and the 8-bit index word to obtain the new CLUT address.

Color Lookup Table Addresses 0x00–0x3f

If the 82750DB Enable mode bit in the Miscellaneous Control register is set to zero, the CLUT addresses are decoded to appear as addresses to the reduced-size 82750DA color table. The least significant four bits of the address are used for the Y color table address, and the upper nibble is used to address the V and U color table simultaneously. This is a compatibility mode for the 82750DA, which has a reduced-size color table.

31 28	27 24	23 16	15 8	7 0
UV Address	Y Address	U Data	V Data	Y Data

If the 82750DB Enable mode bit is set to one, the full color table is used. In this case, the most significant byte of the 32-bit data word is used as an address to the color table. The address is ORed with the most recently loaded CLUT Index register.

31 30	29 24	23 16	15 8	7 0
0 0	YUV Address	U Data	V Data	Y Data

82750DB Register Summary

The following table illustrates the register space of the 82750DB. The old register addresses and the corresponding registers for the 82750DA are given for reference.

Table 4-6. 82750DB/DA Register Space

Address	82750DB Register	82750DA Register
0x00–0x0f	CLUT Locations 0–15	CLUT Locations 00–0f
0x10–0x30	CLUT Locations 16–48	Not Used
0x31	CLUT Location 49	Burst Gate Stop
0x32	CLUT Location 50	Burst Gate Start
0x33	CLUT Location 51	Not Used
0x34	CLUT Location 52	Alpha Register
0x35–0x37	CLUT Location 53–55	Not Used
0x38	CLUT Location 56	Refresh and Register Transfer Timings
0x39–0x3f	CLUT Location 57–63	Not Used
0x40	CLUT Index Register	Not Used
0x41	CLUT Mask Set Register	Not Used
0x42	CLUT Mask Data Register	Not Used
0x43	Miscellaneous Control	Miscellaneous Control
0x44	General Control	Not Used
0x45	Pixel Control	Pixel Control
0x46	Blanking Color	Border Color
0x47	Alpha Register	Not Used
0x48	Border Color	Not Used
0x49	Register Transfer	Line Notification and Timing
0x4a	Line Notification and Timing	Blanking Color
0x4b	DFL Load	Not Used
0x4c	Burst Gate Start	DFL Load
0x4d	Burst Gate Stop	Not Used
0x4e	Active Region Start	Not Used
0x4f	Active Region Stop	Not Used
0x50	Equalization Parameters	Not Used
0x51	Serration Start	Active Region Stop
0x52	Blanking Start	Active Region Start
0x53	Blanking Stop	Not Used
0x54	Sync Start	Equalization Parameters
0x55	Sync Stop	Not Used
0x56	Base Counters	Not Used
0x57	Cursor Color 1	Not Used
0x58	Cursor Color 2	Serration Start
0x59	Cursor Color 3	Not Used
0x5a	Cursor Control	Not Used
0x5b	Not Used	Not Used
0x5c	Not Used	Not Used
0x5d	Not Used	Not Used
0x5e	Not Used	Not Used
0x5f	Not Used	Not Used
0x60	Not Used	Not Used
0x61	Not Used	Blanking Stop
0x62	Not Used	Blanking Start
0x63	Not Used	Not Used
0x64	Not Used	Sync Stops
0x65	Not Used	Not Used
0x66	Not Used	Not Used
0x67	Not Used	Not Used
0x68	Not Used	Sync Start
0x69–0x6e	Not Used	Not Used
0x6f	Not Used	Not Used
0x70	Not Used	Base Counter
0x71–0x7f	Not Used	Not Used
0x80–0xfe	Not Used	Stop Code
0xff	Stop Code	Stop Code

5.0 ELECTRICAL DATA

D.C. Characteristics

Table 5-1. Absolute Maximum Requirements

Condition	Maximum Requirement
Case Temperature under Bias	−65°C to 110°C
Storage Temperature	−65°C to 110°C
Voltage on Any Pin with Respect to Ground	−0.5V to V_{CC} + 0.5V
Supply Voltage with Respect to V_{SS}	−0.5V to +6.5V

Table 5-2. D.C. Characteristics V_{CC} = 5V ± 10%; T_{CASE} = 0°C to +95°C

Symbol	Parameter	Min	Typ	Max	Unit	Notes
V_{IL}	Input LOW Voltage	−0.3		0.8	V	(Note 1)
V_{IH}	Input HIGH Voltage	2.0		V_{CC} + 0.3	V	(Note 1)
V_{OL}	Output LOW Voltage		0.2	0.4	V	I_{OL} = 4.0 mA(1)
V_{OH}	Output HIGH Voltage	2.4	3.0		V	I_{OH} = −1.0 mA(1)
I_{IL}	Input Leakage Current	−10		+10	μA	$V_{SS} < V_{IN} < V_{CC}$
I_{OZ}	Output Leakage Current	−10		+10	μA	$V_{SS} < V_{IN} < V_{CC}$
I_{CC}	Power Supply Current		185	250	mA	28 MHz(2)
C_{IN}	Input Capacitance			10.0	pF	Fc = 1 MHz(3)
C_{OUT}	Output Capacitance			12.0	pF	Fc = 1 MHz(3)
C_{FREQIN}	FREQIN Input Capacitance			20.0	pF	Fc = 1 MHz(3)

NOTES:
1. Measured with FREQIN = 7 MHz.
2. Typical current value measured under typical conditions. Maximum current value guaranteed with 50 pF maximum output loading.
3. Not 100% tested.

A.C. Characteristics

Table 5-3. A.C. Characteristics at 28 MHz V_{CC} = 5V ±10%; T_{CASE} = 0°C to 95°C; C_L = 50 pF

Symbol	Parameter	Min	Max	Unit	Figure	Notes
	Frequency	7	28	MHz		1X Clock
t_1	FREQIN Period	35	140	ns	5-1	
t_2	FREQIN High Time	12		ns	5-1	
t_3	FREQIN Low Time	12		ns	5-1	
t_4	FREQIN Fall Time		4	ns	5-1	
t_5	FREQIN Rise Time		4	ns	5-1	
t_{6a}	HSYNC, VSYNCH, CSYNC, BG, FCO Valid Delay		24	ns	5-2	
t_{6b}	VBUS[3:0] Valid Delay		26	ns	5-2	
t_7	RESETB#, VRESET#, HRESET#, DATAIN[31:0], DISDIG, TESTACT Setup	0		ns	5-3	
t_8	RESETB#, VRESET#, HRESET#, DATAIN[31:0], DISDIG, TESTACT Hold	12		ns	5-3	
t_9	SCLK[1:0] Valid Delay		16	ns	5-4	1X Mode
t_{10}	SCLK[1:0] Valid Delay		16	ns	5-5, 5-6	1/2X, 1/3X Mode
t_{11}	PIXCLK Valid Delay		$\frac{1}{2}t_1$ + 16	ns	5-7	(Note 1)
t_{12}	PIXCLK Valid Delay		16	ns	5-7	(Note 3)
t_{13}	DRV[7:0], DGY[7:0], DBU[7:0], ALPHA [7:0], BPP[1:0], ACTDIS, CB, VUGR Output Setup	6		ns	5-8	
t_{14}	DRV[7:0], DGY[7:0], DBU[7:0], ALPHA[7:0], BPP[1:0] ACTDIS, CB, VUGR Output Hold	6		ns	5-8	
t_{15}	VBUS[3:0], SCLK[1:0], FCO, HSYNC, VSYNC, CSYNC, CB, BG, PIXCLK, DRV[7:0], DGY[7:0], DBU[7:0], ALPHA[7:0], VUGR, ACTDIS, BPP[1:0] Float Delay		30	ns	5-9	(Note 4)

NOTES:
1. For integer pixel times t_{11} is the Valid Delay on all assertions of PIXCLK during active display time.
2. For non-integer pixel times t_{12} is the Valid Delay on alternating assertions of PIXCLK during active display time.
3. Not 100% tested. Guaranteed by design characterization.
4. All A.C. specifications are measured at the 1.5V crossing point with a 50 pF load.

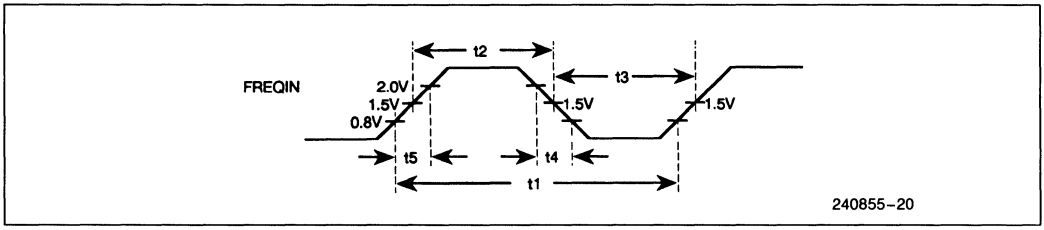


Figure 5-1. Clock Waveforms

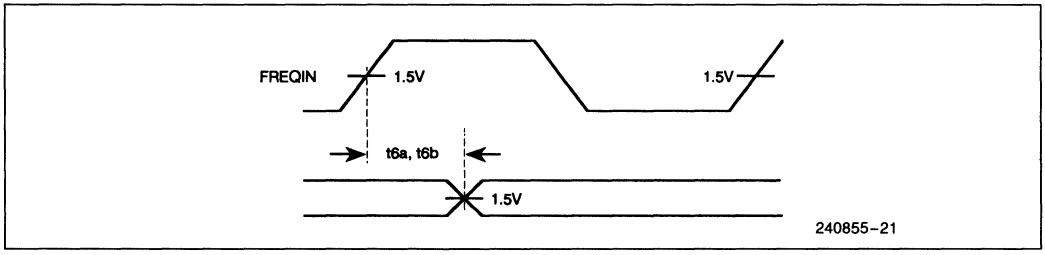


Figure 5-2. Output Waveforms

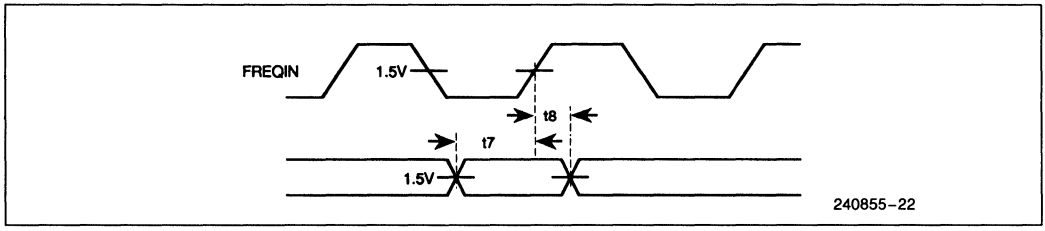


Figure 5-3. Input Waveforms

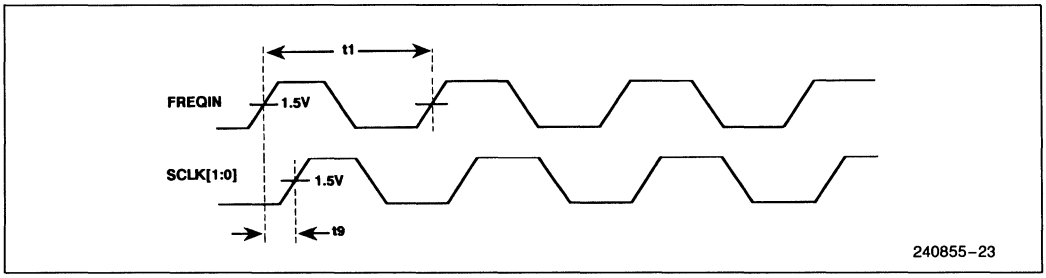


Figure 5-4. 1X SCLK Mode

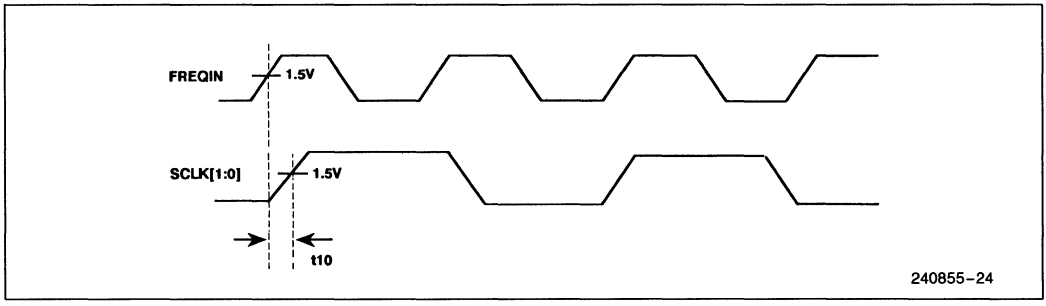


Figure 5-5. 1/2X SCLK Mode

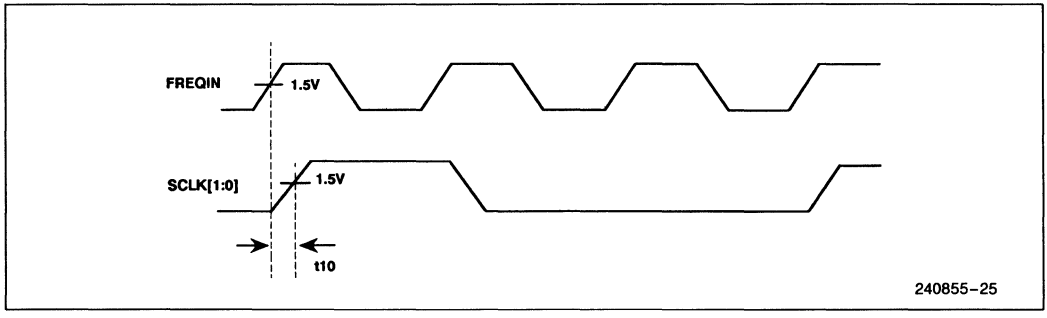


Figure 5-6. 1/3X SCLK Mode

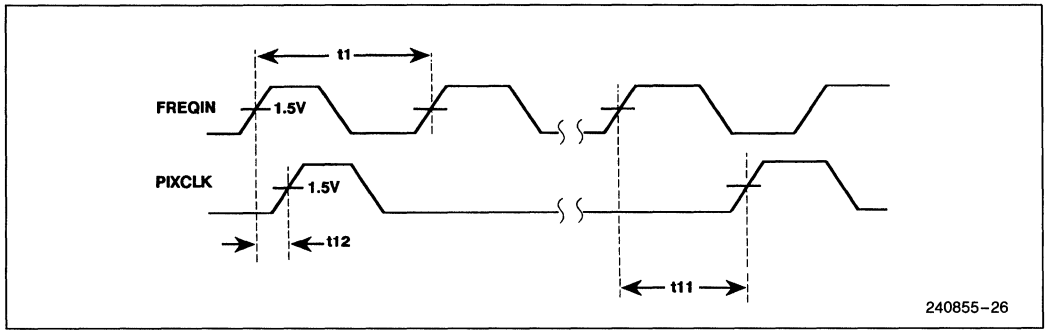


Figure 5-7. PIXCLK Waveforms

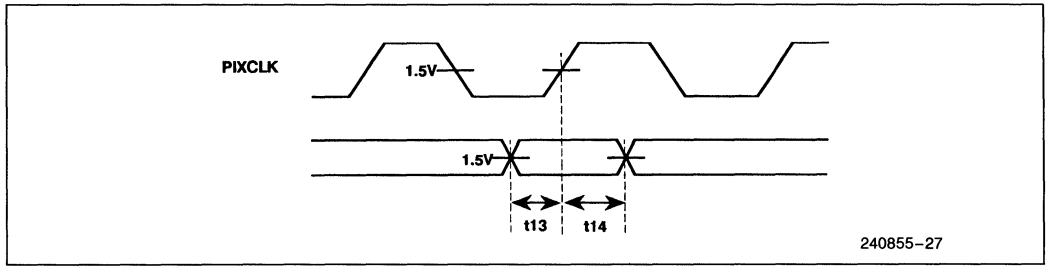


Figure 5-8. Output Setup and Hold

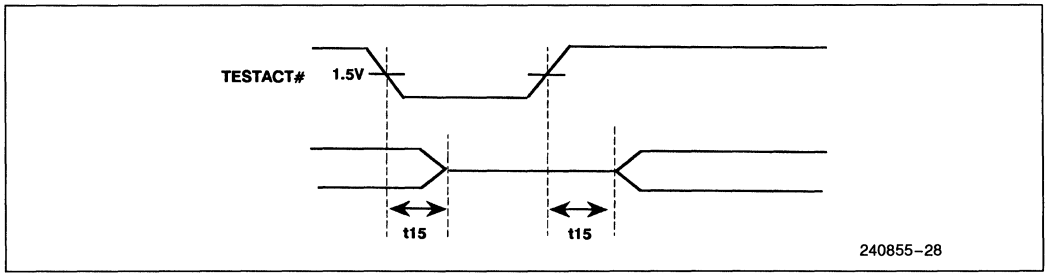


Figure 5-9. TESTACT# Float Delay

Digital to Analog Converter Electrical Characteristics

Table 5-4. DAC D.C. Characteristics AV_{CC} = 5V ±10%; T_{CASE} = 0°C to +95°C

Symbol	Parameter	Min	Typ	Max	Unit	Notes
Iref	Reference Current		706	1000	μA	
Ifs	Output Current* (Full Scale)	0.97 * (255/18) * Iref	10	1.03 * (255/18) * Iref	mA	(Note 1)
Vfs	Output Voltage (Full Scale)			1.5	V	
INL	Integral Nonlinearity			±1	LSB	
DNL	Differential Nonlinearity			±1	LSB	
IACC	Analog Supply Current		34	3 * Ifs + 4	mA	(Note 2)
DDTR	DAC to DAC Tracking at Full Scale		2.0		%	(Note 3)
Cout	Output Capacitance			12	pF	(Note 4)

NOTES:
1. Ifs = 10 mA for Iref = 0.706 mA.
2. Typical current for Ifs = 10 mA.
3. Maximum deviation between RV, GY and BU outputs at fullscale output voltage.
4. Not 100% tested.

Table 5-5. DAC A.C. Characteristics at 28 MHz

Symbol	Parameter	Min	Typ	Max	Unit	Notes
tr, tf	Rise/Fall Time			10	ns	(Note 1)
ClkF	Clock Feedthrough		−28		dB	(Note 2)
GlEn	Glitch Energy		100		pV-sec	(Note 2)
Skew	Output Skew			3	ns	
Xtlk	Crosstalk		200		pV-sec	(Notes 2, 3)

NOTES:
1. Maximum value is for R_L = 100Ω and C_L = 25 pF. Defined as 10% to 90% fullscale transition.
2. Assumes an 80 MHz filter on output.
3. Glitch energy generated by two active outputs on an idle output.
4. IREFIN and VGCS must be decoupled to AV_{CC}. DISDIG must be tied high.

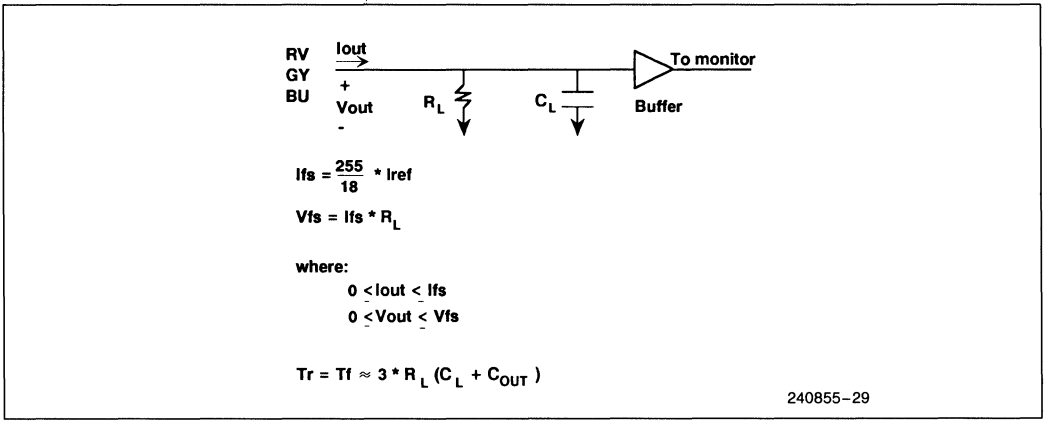
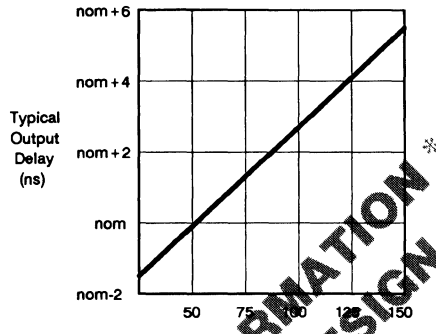


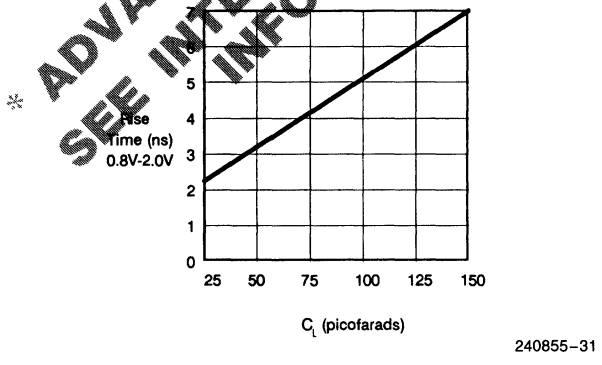
Figure 5-10. Typical Output Configuration

Output Delay and Rise Time versus Load Capacitance



NOTE:
This graph will not be linear outside of the C_L range shown. nom = nominal value given in A.C. Characteristics table.

Figure 5-11. Typical Output Valid Delay versus Load Capacitance under Worst Case Conditions



NOTE:
This graph will not be linear outside of the C_L range shown.

Figure 5-12. Typical Output Rise Time versus Load Capacitance under Worst Case Conditions

6.0 MECHANICAL DATA

Packaging Outlines and Dimensions

Intel packages the 82750DB in a Plastic Quad Flat Pack (PQFP). Table 6-1 gives the symbol list for the PQFP.

Table 6-1. PQFP Symbol List

Letter or Symbol	Description of Dimensions
A	Package Height: Distance from Seating Plane to Highest Point of Body
A_1	Standoff: Distance from Seating Plane to Base Plane
D/E	Overall Package Dimension: Lead Tip to Lead Tip
D_1/E_1	Plastic Body Dimension
D_2/E_2	Bumper Distance
D_3/E_3	Footprint
L_1	Foot Length
N	Total Number of Leads

The PQFP has the following specifications:

1. All dimensions and tolerances conform to ANSI Y14.5M-1982.
2. Datum plane-H-is located at the mold parting line and coincident with the bottom of the lead where lead exits plastic body.
3. Datums A-B and -D- are to be determined where center leads exit plastic body at datum plane -H-.
4. Controlling dimension is the inch.
5. Dimensions D_1, D_2, E_1, and E_2 are measured at the mold parting line and do not include mold protrusion. Allowable mold protrusion is 0.18 mm (0.007 in.) per side.
6. Pin 1 identifier is located within one of the two zones indicated.
7. Measured at datum plane -H-.
8. Measured at seating plane datum -C-.

Table 6-2 provides outline characteristics for 0.025-in. pitch.

Table 6-2. Intel Case Outline Drawings for PQFP at 0.025 Inch Pitch

Symbol	Description	Min	Max
N	Leadcount	132	132
A	Package Height	0.160	0.170
A_1	Standoff	0.020	0.030
D,E	Terminal Dimension	1.075	1.085
D_1, E_1	Package Body	0.947	0.953
D_2, E_2	Bumper Distance	1.097	1.103
D_3, E_3	Lead Dimension	0.800 REF	0.800 REF
L_1	Foot Length	0.020	0.030

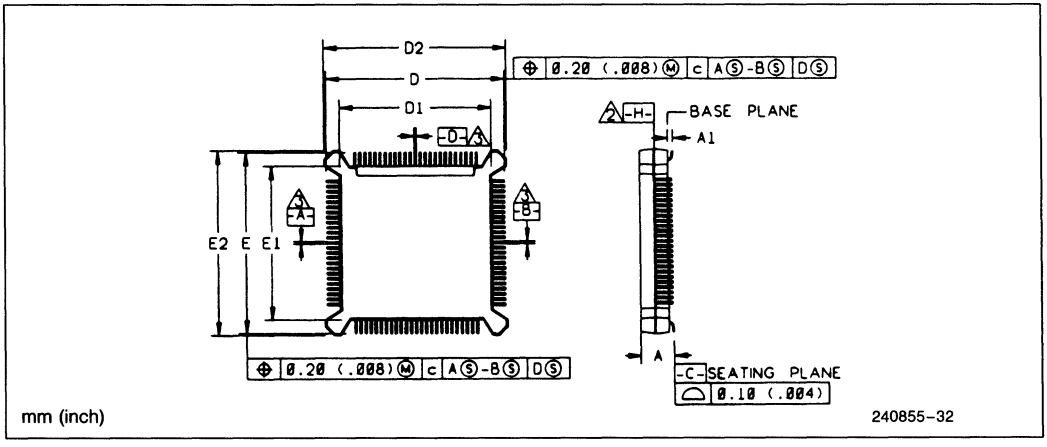


Figure 6-1. Principal Dimensions and Datums

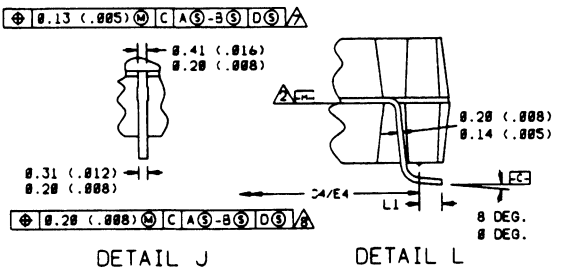


Figure 6-2. Molding Details

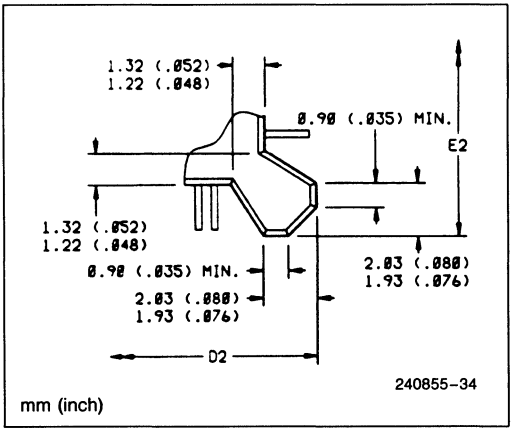


Figure 6-3. Terminal Details

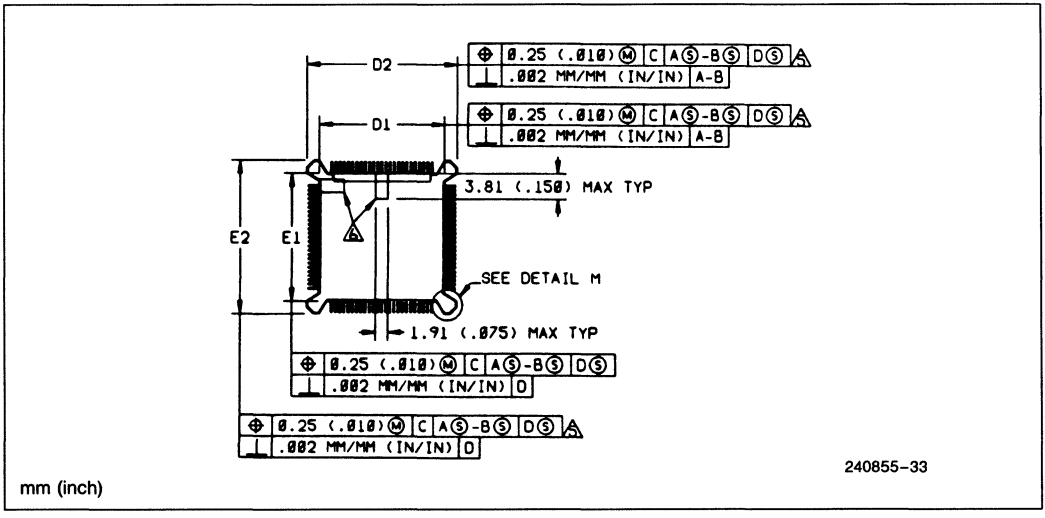
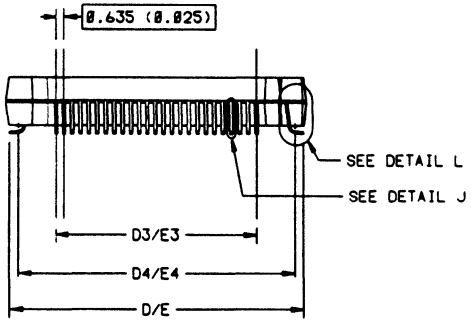


Figure 6-4. Typical Lead



240855-35

mm (inch)

Figure 6-5. Bumper (Detail M)

NOTES:

1. ALL DIMENSIONS AND TOLERANCES CONFORM TO ANSI Y14.5M-1982
2. DATUM PLANE -H- LOCATED AT THE MOLD PARTING LINE AND COINCIDENT WITH THE BOTTOM OF THE LEAD WHERE LEAD EXITS PLASTIC BODY
3. DATUMS A-B AND -D- TO BE DETERMINED WHERE CENTER LEADS EXIT PLASTIC BODY AT DATUM PLANE -H-
4. CONTROLLING DIMENSION, INCH
5. DIMENSIONS D1, D2, E1 AND E2 ARE MEASURED AT THE MOLD PARTING LINE. D1 AND E1 DO NOT INCLUDE AN ALLOWABLE MOLD PROTRUSION OF 0.18 MM (.007 IN) PER SIDE. D2 AND E2 DO NOT INCLUDE A TOTAL ALLOWABLE MOLD PROTRUSION OF 0.18 MM (.007 IN) AT MAXIMUM PACKAGE SIZE.
6. PIN 1 IDENTIFIER IS LOCATED WITHIN ONE OF THE TWO ZONES INDICATED
7. MEASURED AT DATUM PLANE -H-
8. MEASURED AT SEATING PLANE DATUM -C-

240855-37

Package Thermal Specifications

The 82750DB is specified for operation when T_C (the case temperature) is within the range of 0°C to 95°. T_C may be measured in any environment to determine whether the 82750DB is within specified operating range. The case temperature should be measured at the center of the top surface.

T_A (the ambient temperature) can be calculated from θ_{CA} (thermal resistance from case to ambient) with the following equation:

$$T_A = T_C - P * \theta_{CA}$$

Typical values for θ_{CA} at various airflows are given in Table 6-3 for the 132-lead PQFP package. Table 6-4 shows the maximum T_A allowable (without exceeding T_C) at various airflows. The power dissipation (P) is calculated by using the typical supply currents at 5V as shown in Table 5-2.

Table 6-3. Thermal Resistances (°C/W)

	θ_{CA} Versus Airflow—ft/min (m/sec)					
Package	**0 (0)**	**200 (1.01)**	**400 (2.03)**	**600 (3.04)**	**800 (4.06)**	**1000 (5.07)**
132-Lead PQFP	26.0	17.5	14.0	11.5	9.5	8.5

Table 6-4. Maximum T_A at Various Airflows (°C)

		θ_A Versus Airflow—ft/min (m/sec)					
Package	**Frequency (MHz)**	**0 (0)**	**200 (1.01)**	**400 (2.03)**	**600 (3.04)**	**800 (4.06)**	**1000 (5.07)**
132-Lead PQFP	28	71	79	82	84	86	87

DOMESTIC SALES OFFICES

ALABAMA

Intel Corp
5015 Bradford Dr , #2
Huntsville 35805
Tel (205) 830-4010
FAX (205) 837-2640

ARIZONA

†Intel Corp
410 North 44th Street
Suite 500
Phoenix 85008
Tel (602) 231-0386
FAX (602) 244-0446

Intel Corp
7225 N Mona Lisa Rd
Suite 215
Tucson 85741
Tel (602) 544-0227
FAX (602) 544-0232

CALIFORNIA

†Intel Corp
21515 Vanowen Street
Suite 116
Canoga Park 91303
Tel (818) 704-8500
FAX (818) 340-1144

†Intel Corp
300 N Continental Blvd
Suite 100
El Segundo 90245
Tel (213) 640-6040
FAX (213) 640-7133

Intel Corp
1 Sierra Gate Plaza
Suite 280C
Roseville 95678
Tel (916) 782-8086
FAX (916) 782-8153

†Intel Corp
9665 Chesapeake Dr
Suite 325
San Diego 92123
Tel (619) 292-8086
FAX (619) 292-0628

†Intel Corp *
400 N Tustin Avenue
Suite 450
Santa Ana 92705
Tel (714) 835-9642
TWX 910-595-1114
FAX (714) 541-9157

†Intel Corp *
San Tomas 4
2700 San Tomas Expressway
2nd Floor
Santa Clara 95051
Tel (408) 986-8086
TWX 910-338-0255
FAX (408) 727-2620

COLORADO

Intel Corp
4445 Northpark Drive
Suite 100
Colorado Springs 80907
Tel (719) 594-6622
FAX (303) 594-0720

†Intel Corp *
600 S Cherry St
Suite 700
Denver 80222
Tel (303) 321-8086
TWX 910-931-2289
FAX (303) 322-8670

CONNECTICUT

†Intel Corp
301 Lee Farm Corporate Park
83 Wooster Heights Rd
Danbury 06810
Tel (203) 748-3130
FAX (203) 794-0339

FLORIDA

†Intel Corp
800 Fairway Drive
Suite 160
Deerfield Beach 33441
Tel (305) 421-0506
FAX (305) 421-2444

†Intel Corp
5850 T G Lee Blvd
Suite 340
Orlando 32822
Tel (407) 240-8000
FAX (407) 240-8097

Intel Corp
11300 4th Street North
Suite 170
St Petersburg 33716
Tel (813) 577-2413
FAX (813) 578-1607

GEORGIA

†Intel Corp
20 Technology Parkway
Suite 150
Norcross 30092
Tel (404) 449-0541
FAX (404) 605-9762

ILLINOIS

†Intel Corp *
Woodfield Corp Center III
300 N Martingale Road
Suite 400
Schaumburg 60173
Tel (708) 605-8031
FAX (708) 706-9762

INDIANA

†Intel Corp
8910 Purdue Road
Suite 350
Indianapolis 46268
Tel (317) 875-0623
FAX (317) 875-8938

IOWA

Intel Corp
1930 St Andrews Drive N E
2nd Floor
Cedar Rapids 52402
Tel (319) 393-5510

KANSAS

†Intel Corp
10985 Cody St
Suite 140
Overland Park 66210
Tel (913) 345-2727
FAX (913) 345-2076

MARYLAND

†Intel Corp *
10010 Junction Dr
Suite 200
Annapolis Junction 20701
Tel (301) 206-2860
FAX (301) 206-3677
(301) 206-3678

MASSACHUSETTS

†Intel Corp *
Westford Corp Center
3 Carlisle Road
2nd Floor
Westford 01886
Tel (508) 692-0960
TWX 710-343-6333
FAX (508) 692-7867

MICHIGAN

†Intel Corp
7071 Orchard Lake Road
Suite 100
West Bloomfield 48322
Tel (313) 851-8096
FAX (313) 851-8770

MINNESOTA

†Intel Corp
3500 W 80th St
Suite 360
Bloomington 55431
Tel (612) 835-6722
TWX 910-576-2867
FAX (612) 831-6497

MISSOURI

†Intel Corp
4203 Earth City Expressway
Suite 131
Earth City 63045
Tel (314) 291-1990
FAX (314) 291-4341

NEW JERSEY

†Intel Corp *
Parkway 109 Office Center
328 Newman Springs Road
Red Bank 07701
Tel (201) 747-2233
FAX (201) 747-0983

Intel Corp
280 Corporate Center
75 Livingston Avenue
First Floor
Roseland 07068
Tel (201) 740-0111
FAX (201) 740-0626

NEW YORK

Intel Corp *
850 Crosskeys Office Park
Fairport 14450
Tel (716) 425-2750
TWX 510-253-7391
FAX (716) 223-2561

†Intel Corp *
2950 Express Dr , South
Suite 130
Islandia 11722
Tel (516) 231-3300
TWX 510-227-6236
FAX (516) 348-7939

†Intel Corp
Westage Business Center
Bldg 300, Route 9
Fishkill 12524
Tel (914) 897-3860
FAX (914) 897-3125

NORTH CAROLINA

†Intel Corp
5800 Executive Center Dr
Suite 105
Charlotte 28212
Tel (704) 568-8966
FAX (704) 535-2236

†Intel Corp
5540 Centerview Dr
Suite 215
Raleigh 27606
Tel (919) 851-9537
FAX (919) 851-8974

OHIO

†Intel Corp *
3401 Park Center Drive
Suite 220
Dayton 45414
Tel (513) 890-5350
TWX 810-450-2528
FAX (513) 890-8658

†Intel Corp *
25700 Science Park Dr
Suite 100
Beachwood 44122
Tel (216) 464-2736
TWX 810-427-9298
FAX (804) 282-0673

OKLAHOMA

Intel Corp
6801 N Broadway
Suite 115
Oklahoma City 73162
Tel (405) 848-8086
FAX (405) 840-9819

OREGON

†Intel Corp
15254 N W Greenbrier Parkway
Building B
Beaverton 97005
Tel (503) 645-8051
TWX 910-467-8741
FAX (503) 645-8181

PENNSYLVANIA

†Intel Corp *
925 Harvest Drive
Suite 200
Blue Bell 19422
Tel (215) 641-1000
FAX (215) 641-0785

†Intel Corp *
400 Penn Center Blvd
Suite 610
Pittsburgh 15235
Tel (412) 823-4970
FAX (412) 829-7578

PUERTO RICO

†Intel Corp
South Industrial Park
P O Box 910
Las Piedras 00671
Tel (809) 733-8616

TEXAS

Intel Corp
8911 Capital of Texas Hwy
Austin 78759
Tel (512) 794-8086
FAX (512) 338-9335

†Intel Corp *
12000 Ford Road
Suite 400
Dallas 75248
Tel (214) 241-8087
FAX (214) 484-1180

†Intel Corp *
7322 S W Freeway
Suite 1490
Houston 77074
Tel (713) 988-8086
TWX 910-881-2490
FAX (713) 988-3660

UTAH

†Intel Corp
428 East 6400 South
Suite 104
Murray 84107
Tel (801) 263-8051
FAX (801) 268-1457

VIRGINIA

†Intel Corp
1504 Santa Rosa Road
Suite 108
Richmond 23288
Tel (804) 282-5668
FAX (216) 464-2270

WASHINGTON

†Intel Corp
155 108th Avenue N E
Suite 386
Bellevue 98004
Tel (206) 453-8086
TWX 910-443-3002
FAX (206) 451-9556

Intel Corp
408 N Mullan Road
Suite 102
Spokane 99206
Tel (509) 928-8086
FAX (509) 928-9467

WISCONSIN

Intel Corp
330 S Executive Dr
Suite 102
Brookfield 53005
Tel (414) 784-8087
FAX (414) 796-2115

CANADA

BRITISH COLUMBIA

Intel Semiconductor of
Canada, Ltd
4585 Canada Way
Suite 202
Burnaby V5G 4L6
Tel (604) 298-0387
FAX (604) 298-8234

ONTARIO

†Intel Semiconductor of
Canada, Ltd
2650 Queensview Drive
Suite 250
Ottawa K2B 8H6
Tel (613) 829-9714
FAX (613) 820-5936

†Intel Semiconductor of
Canada, Ltd
190 Attwell Drive
Suite 500
Rexdale M9W 6H8
Tel (416) 675-2105
FAX (416) 675-2438

QUEBEC

†Intel Semiconductor of
Canada, Ltd
1 Rue Holiday
Suite 115
Tour East
Pt Claire H9R 5N3
Tel (514) 694-9130
FAX 514-694-0064

†Sales and Service Office
*Field Application Location

UNITED STATES
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

JAPAN
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26

FRANCE
Intel Corporation S.A.R.L.
1, Rue Edison, BP 303
78054 Saint-Quentin-en-Yvelines Cedex

UNITED KINGDOM
Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon
Wiltshire, England SN3 1RJ

WEST GERMANY
Intel GmbH
Dornacher Strasse 1
8016 Feldkirchen bei Muenchen

HONG KONG
Intel Semiconductor Ltd.
10/F East Tower
Bond Center
Queensway, Central

CANADA
Intel Semiconductor of Canada, Ltd.
190 Attwell Drive, Suite 500
Rexdale, Ontario M9W 6H8

Order Number 240873-001

Printed in U.S.A./3K/1090/BL JAG
DVI™ Technology