



---

# ***TMS320C80 (MVP) Video Controller***

*User's Guide*

***1995***

***Digital Signal Processing Products***

---





*User's  
Guide*

# ***TMS320C80 (MVP) Video Controller***

*1995*

# ***TMS320C80 (MVP) Video Controller User's Guide***

SPRU111A  
March 1995



## **IMPORTANT NOTICE**

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

# Read This First

---

---

---

### *About This Manual*

The TMS320C80 MVP (multimedia video processor) is Texas Instruments first single-chip multiprocessor DSP (digital signal processor) device. The MVP contains five powerful, fully programmable processors: a master processor (MP) and four parallel processors (PPs). The MP is a 32-bit RISC (reduced instruction set computer) with an integral, high-performance IEEE-754 floating-point unit. Each PP is an advanced 32-bit DSP; thus, in addition to having similar processing capabilities as conventional DSPs, each PP has advanced features to accelerate operation on a variety of data types.

The MVP supports a variety of parallel-processing configurations, which facilitates a wide range of multimedia and other applications that require high processing speeds. Applications include image processing, two- and three-dimensional and virtual reality graphics, audio/video digital compression, and telecommunications.

This manual describes the MVP video controller (VC). The VC provides the video interface between the MVP and the image capture and display systems to perform complex video and audio applications. The VC contains dual frame timers that provide simultaneous image capture and display. This manual provides information about the VC features, architecture, and operation; it also includes procedures and examples for programming the serial register transfer (SRT) controller and the frame timer registers.

## *Related Documentation From Texas Instruments*

The following books describe the TMS320C80 MVP and related support tools. To obtain a copy of any of these TI documents, call the Texas Instruments Literature Response Center at (800) 477–8924. When ordering, please identify the book by its title and literature number.

***TMS320C80 Multimedia Video Processor Data Sheet***

(literature number SPRS023) describes the features of the 'C80 device and provides pinouts, electrical specifications, and timings for the device.

***TMS320C80 Multimedia Video Processor (MVP) Technical***

***Brief*** (literature number SPRU106) provides an overview of the 'C80 features, development environment, architecture, and memory organization.

***TMS320C80 (MVP) C Source Debugger User's Guide***

(literature number SPRU107) describes the 'C80 master processor and parallel processor C source debuggers. This manual provides information about the features and operation of the debuggers and the parallel debug manager; it also includes basic information about C expressions and a description of progress and error messages.

***TMS320C80 (MVP) Code Generation Tools User's Guide***

(literature number SPRU108) describes the 'C80 code generation tools. This manual provides information about the features and operation of the linker and the master processor (MP) and parallel processor (PP) C compilers and assemblers. It also includes a description of the common object file format (COFF) and shows you how to link MP and PP code.

***TMS320C80 (MVP) Master Processor User's Guide***

(literature number SPRU109) describes the 'C80 master processor (MP). This manual provides information about the MP features, architecture, operation, and assembly language instruction set; it also includes sample applications that illustrate various MP operations.

***TMS320C80 (MVP) Multitasking Executive User's Guide***

(literature number SPRU112) describes the 'C80 multitasking executive software. This manual provides information about the multitasking executive software features, operation, and interprocessor communications; it also includes a list of task error codes.

***TMS320C80 (MVP) Parallel Processor User's Guide*** (literature number SPRU110) describes the 'C80 parallel processor (PP). This manual provides information about the PP features, architecture, operation, and assembly language instruction set; it also includes software applications and optimizations.

***TMS320C80 (MVP) Transfer Controller User's Guide*** (literature number SPRU105) describes the 'C80 transfer controller (TC). This manual provides information about the TC features, functional blocks, and operation; it also includes examples of block write operations for big- and little-endian modes.

***TMS320C80 (MVP) System-Level Synopsis*** (literature number SPRU113) contains the 'C80 system-level synopsis, which describes the 'C80 features, development environment, architecture, memory organization, and communication network (the crossbar).

*If You Need Assistance. . .*

<b>If you want to. . .</b>	<b>Do this. . .</b>
Request more information about Texas Instruments Digital Signal Processing (DSP) products	Write to: Texas Instruments Incorporated Market Communications Manager, MS 736 P.O. Box 1443 Houston, Texas 77251-1443
Order Texas Instruments documentation	Call the TI Literature Response Center: <b>(800) 477-8924</b>
Ask questions about product operation or report suspected problems	Call the DSP hotline: <b>(713) 274-2320</b> <b>FAX: (713) 274-2324</b>
Report mistakes in this document or any other TI documentation	Fill out and return the reader response card at the end of this book, or send your comments to: Texas Instruments Incorporated Technical Publications Manager, MS 702 P.O. Box 1443 Houston, Texas 77251-1443  Electronic mail: <b>comments@books.sc.ti.com</b>





# Contents

---

---

---

<b>1</b>	<b>Overview of the Video Controller .....</b>	<b>VC:1-1</b>
	Describes the video controller and its relative location within the TMS320C80 (MVP) architecture. Provides the memory maps for the frame timers and serial register transfer controller.	
1.1	Description of the VC .....	VC:1-2
1.1.1	VC Functional Blocks .....	VC:1-4
1.1.2	Typical Applications .....	VC:1-4
1.2	The VC Memory Map .....	VC:1-5
1.2.1	Frame Timer Register Maps .....	VC:1-6
1.2.2	SRT Controller Register Maps .....	VC:1-8
<b>2</b>	<b>Signal Descriptions .....</b>	<b>VC:2-1</b>
	Describes the configuration, direction, and timing intervals of the video controller signals.	
2.1	Timing Control Signals .....	VC:2-2
2.2	Frame Clock Signals .....	VC:2-3
2.3	Horizontal Timing Signals .....	VC:2-4
2.4	Vertical Timing Signals .....	VC:2-5
2.5	Composite Timing Signals .....	VC:2-6
2.6	SRT Controller Signals .....	VC:2-8
<b>3</b>	<b>Frame Timer Registers .....</b>	<b>VC:3-1</b>
	Describes the registers associated with each of the frame timers.	
3.1	Summary of the Frame Timer Registers .....	VC:3-2
3.2	Description of the Frame Timer Registers .....	VC:3-4

<b>4</b>	<b>Video Timing</b>	<b>VC:4-1</b>
	Discusses several video timing features of the video controller and how to program them.	
4.1	Understanding Horizontal and Vertical Timing	VC:4-2
4.2	Horizontal Video Timing (Internal)	VC:4-4
4.3	Vertical Video Timing (Internal)	VC:4-6
4.3.1	Noninterlaced Vertical Timing	VC:4-6
4.3.2	Interlaced Vertical Timing	VC:4-8
4.4	Composite Video Timing	VC:4-9
4.4.1	The Theory Behind Serration and Equalization Pulses	VC:4-10
4.4.2	Equalization Pulses on $\overline{\text{CSYNC}}$	VC:4-12
4.4.3	Serration Pulses on $\overline{\text{CSYNC}}$	VC:4-13
4.5	Noninterlaced Video Timing	VC:4-15
4.5.1	Understanding Noninterlaced Mode	VC:4-15
4.5.2	Noninterlaced Horizontal Timing Operation	VC:4-17
4.5.3	Noninterlaced Vertical Timing Operation	VC:4-19
4.5.4	Summary for Programming Noninterlaced Video Timing	VC:4-21
4.6	Interlaced Video Timing	VC:4-22
4.6.1	Understanding Interlaced Mode	VC:4-22
4.6.2	Interlaced Horizontal Timing Operation	VC:4-25
4.6.3	Interlaced Vertical Timing Operation	VC:4-28
4.6.4	Summary for Programming Interlaced Video Timing	VC:4-32
4.7	Programming VSAREA, VEAREA, and VFTINT in Interlaced Mode	VC:4-33
4.7.1	Vertical Start Area Programming	VC:4-33
4.7.2	Vertical End Area Programming	VC:4-36
4.7.3	Vertical Frame Timer Interrupt Programming	VC:4-39
4.7.4	Alternative Programming for VFTINT, VSAREA, and VEAREA	VC:4-42
4.8	External Synchronization	VC:4-44
4.8.1	Video Counter Operation	VC:4-44
4.8.2	External Synchronization Modes	VC:4-46
4.8.3	Sync Output Operation	VC:4-47
4.8.4	Synchronizing to FCLK	VC:4-49
4.9	Synchronization Delay Compensation	VC:4-50
4.10	High-Impedance Capability	VC:4-52
4.11	Frame Lock	VC:4-53

## **5 Serial Register Transfer Controller ..... VC:5-1**

Describes the serial register transfer controller. Includes an alphabetical listing of the control registers, along with information pertaining to several characteristics of the serial register transfer controller.

5.1	Understanding the SRT Controller .....	VC:5-2
5.2	SRT Controller Registers .....	VC:5-3
5.3	SRT Control Masks .....	VC:5-16
5.4	SRT Events .....	VC:5-17
5.4.1	Start of Field Event .....	VC:5-18
5.4.2	End of Field Event .....	VC:5-19
5.4.3	Line Event .....	VC:5-19
5.4.4	SAM Overflow Event .....	VC:5-20
5.4.5	VC-Initiated Packet Transfer Requests .....	VC:5-22
5.5	Event Prioritization .....	VC:5-23
5.6	Address Generation .....	VC:5-25
5.7	Understanding the SRT Modes .....	VC:5-27
5.7.1	Display Mode .....	VC:5-27
5.7.2	Capture Mode .....	VC:5-29
5.7.3	Merge Capture Mode .....	VC:5-31
5.7.4	Interlaced Line Repeat .....	VC:5-34

## **A Programming Procedures and Examples ..... VC:A-1**

Describes the procedures for programming the serial register transfer controller and the frame timer registers for both interlaced and noninterlaced modes.

A.1	Relationship Between Horizontal and Vertical Timing Signals ..	VC:A-2
A.2	Screen Resolution .....	VC:A-4
A.3	Noninterlaced Monitor Specifications .....	VC:A-5
A.4	Pixel Clocks .....	VC:A-6
A.5	Programming the Frame Timer Registers for Noninterlaced Video .....	VC:A-7
A.6	Programming the SRT Controller Registers .....	VC:A-10
A.7	Programming the Horizontal Line Event Registers .....	VC:A-12
A.8	NTSC Composite Interlaced Video Display .....	VC:A-13
A.9	Programming the Frame Timer Registers for Interlaced Video .	VC:A-16
A.10	Frame Timer Interrupts .....	VC:A-18

## **B Glossary ..... VC:B-1**

Defines acronyms and key terms used in this book.

# Figures

---



---



---

1–1	MVP Block Diagram .....	VC:1-2
1–2	VC Block Diagram .....	VC:1-3
1–3	VC Memory Map .....	VC:1-5
1–4	Frame Timer 0 Register Map .....	VC:1-6
1–5	Frame Timer 1 Register Map .....	VC:1-6
1–6	32-Bit Frame Timer Register Access .....	VC:1-7
1–7	SRT Controller Register Map .....	VC:1-8
4–1	Horizontal and Vertical Timing Relationship .....	VC:4-3
4–2	Horizontal Timing Signals .....	VC:4-4
4–3	Noninterlaced Vertical Timing Signals .....	VC:4-6
4–4	Interlaced Vertical Timing Signals .....	VC:4-8
4–5	Vertical Blanking Regions of Equalization and Serration .....	VC:4-11
4–6	<u>CSYNC</u> During Equalization Regions (Interlaced) .....	VC:4-12
4–7	<u>CSYNC</u> During Serration Region (Interlaced) .....	VC:4-13
4–8	<u>CSYNC</u> During Serration Region (Noninterlaced) .....	VC:4-14
4–9	Scan Pattern for Noninterlaced Monitors .....	VC:4-15
4–10	Noninterlaced Horizontal Timing Summary .....	VC:4-18
4–11	Noninterlaced Vertical Timing Summary .....	VC:4-20
4–12	Scan Pattern for Interlaced Monitors—Odd Field .....	VC:4-22
4–13	Scan Pattern for Interlaced Monitors—Even Field .....	VC:4-23
4–14	Scan Pattern for Interlaced Monitors—Both Fields .....	VC:4-24
4–15	Interlaced Horizontal Timing Summary .....	VC:4-26
4–16	NTSC-Type Interlaced Frame Timing Example .....	VC:4-30
4–17	PAL-Type Interlaced Frame Timing Example .....	VC:4-31
4–18	Programming VSAREA for NTSC .....	VC:4-34
4–19	Programming VSAREA for PAL .....	VC:4-35
4–20	Programming VEAREA for NTSC .....	VC:4-37
4–21	Programming VEAREA for PAL .....	VC:4-38
4–22	Programming VFTINT for NTSC (IIM = 1) .....	VC:4-40
4–23	Programming VFTINT for PAL (IIM = 1) .....	VC:4-41
4–24	Synchronization Delay Compensation .....	VC:4-50
5–1	SAM Overflow Event at Beginning of Frame .....	VC:5-20
5–2	SAM Overflow Event After the First SCLK Pulse .....	VC:5-21
5–3	SAM Overflow Event After 128th SCLK Pulse .....	VC:5-21

---

5-4	SRT Event Priority .....	VC:5-23
5-5	Display Mode SRT Sequence .....	VC:5-28
5-6	Capture Mode SRT Sequence .....	VC:5-30
5-7	Merge Capture Mode SRT Sequence .....	VC:5-33
A-1	Horizontal and Vertical Timing Relationship .....	VC:A-2
A-2	The Porches .....	VC:A-3
A-3	Noninterlaced Monitor Timing (Separate Sync) .....	VC:A-5
A-4	Horizontal and Vertical Timing Relationship for Noninterlaced Video .....	VC:A-9
A-5	Interlaced Video Timing—Even Field .....	VC:A-14

# Tables

2-1	Timing Control Signals .....	VC:2-2
3-1	Frame Timer Control Register Addresses .....	VC:3-2
3-2	Horizontal Timing Register Addresses .....	VC:3-2
3-3	Vertical Timing Register Addresses .....	VC:3-3
3-4	$\overline{\text{HSYNC}}$ Pin Modes (HPM) .....	VC:3-5
3-5	$\overline{\text{VSYNC}}$ Pin Modes (VPM) .....	VC:3-5
3-6	$\overline{\text{CSYNC}}/\overline{\text{HBLNK}}$ and $\overline{\text{CBLNK}}/\overline{\text{VBLNK}}$ Pin Modes (CPM) .....	VC:3-6
4-1	Programming the Video Timing Registers for Noninterlaced Video ..	VC:4-21
4-2	Programming the Video Timing Registers for Interlaced Video .....	VC:4-32
5-1	SRT Controller Registers .....	VC:5-3
5-2	Frame Timer Sequencing Modes (FTS) .....	VC:5-9
5-3	SRT Event Modes (EMS) .....	VC:5-10
5-4	VCPT Event Modes (EMS) .....	VC:5-11
5-5	SRT Event Modes (TMS) .....	VC:5-11
5-6	Linked-List Start Addresses .....	VC:5-22
A-1	Typical Noninterlaced Monitor Timing .....	VC:A-5
A-2	Configuring the Values of the Frame Timer Registers for a Noninterlaced 1024 × 768 Display .....	VC:A-8
A-3	NTSC Timings .....	VC:A-15
A-4	Configuring the Values of the Frame Timer Registers for Interlaced Video .....	VC:A-17

# Overview of the Video Controller

---

---

---

The video controller (VC) is an integral part of the TMS320C80 (MVP). The VC provides the video interface so that you can perform complex video and audio applications using the MVP.

This chapter describes the VC and its relative location within the MVP architecture. It also provides the memory maps for the frame timers and serial register transfer (SRT) controller.

## Topics

<b>1.1</b>	<b>Description of the VC .....</b>	<b>VC:1-2</b>
<b>1.2</b>	<b>The VC Memory Map .....</b>	<b>VC:1-5</b>

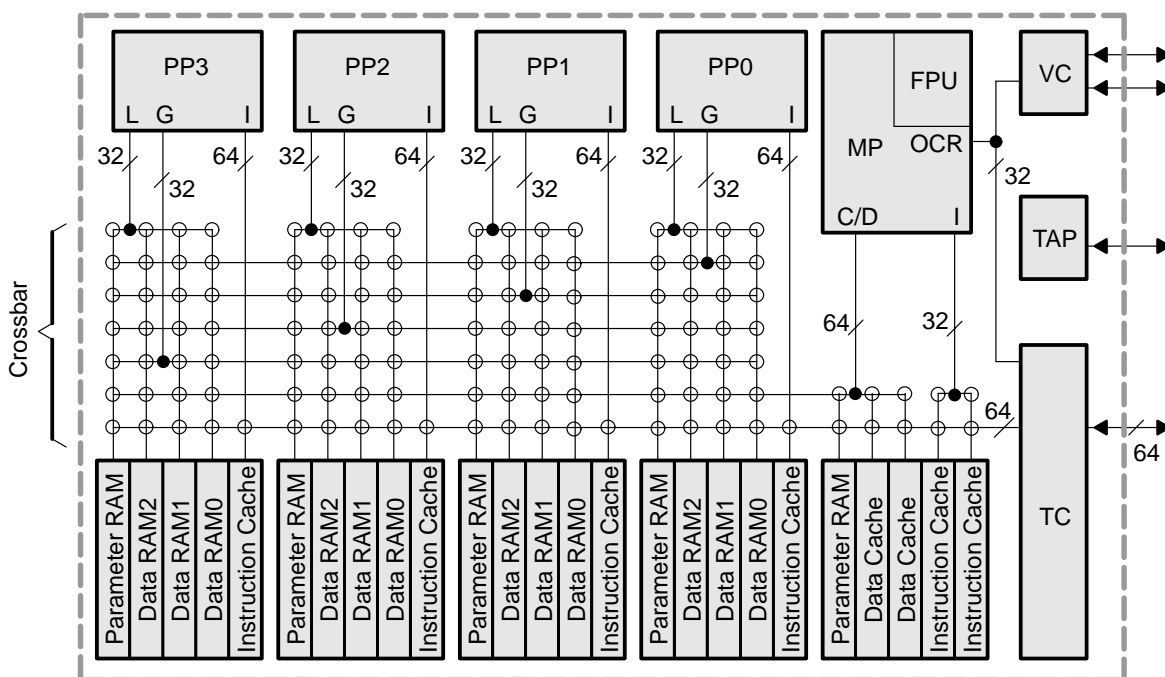


## 1.1 Description of the VC

The VC portion of the MVP is responsible for the video interface. The VC provides simultaneous control over two independent frame systems. A frame system consists of an external frame buffer memory and control logic. The frame systems can function as either capture or display systems.

Figure 1–1 illustrates the individual processors of the MVP and the crossbar network that connects them to memory. The VC is located in the top right corner of this diagram.

Figure 1–1. MVP Block Diagram

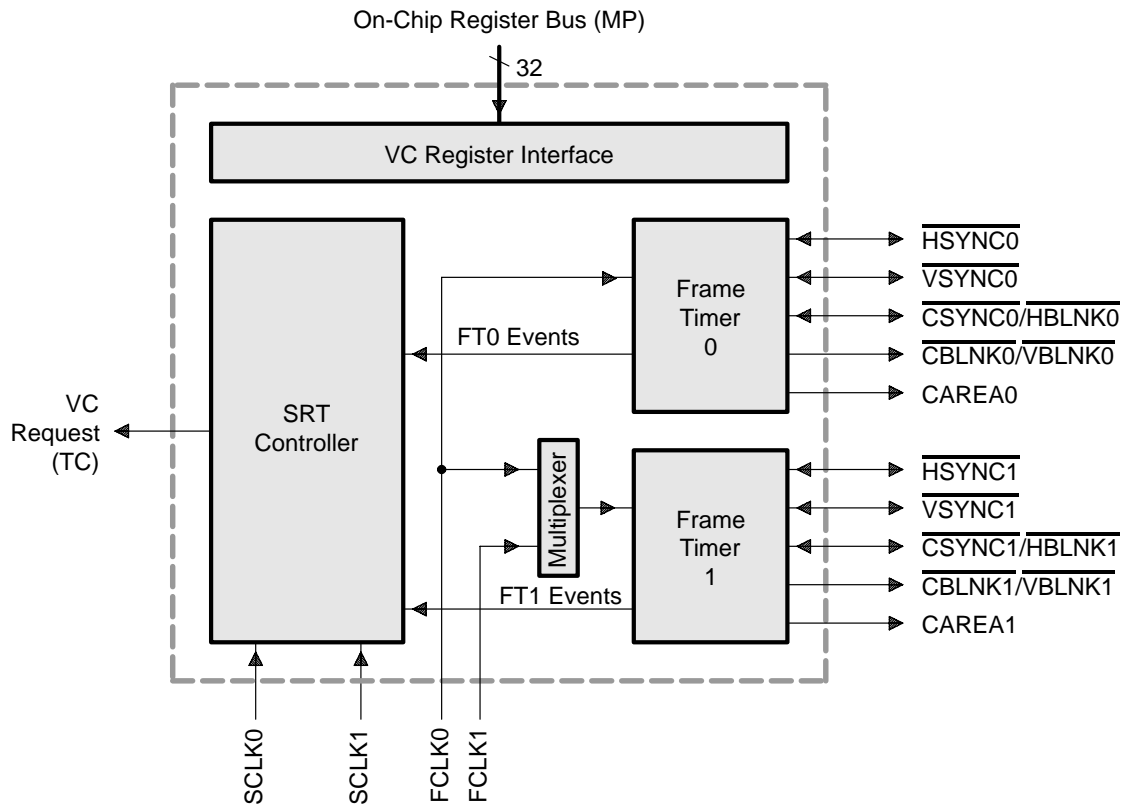


**Note:** L = Local port  
 I = Instruction port  
 OCR = On-chip register port  
 TAP = Test access port  
 VC = Video controller  
 PP0–3 = Parallel processors 0–3

G = Global port  
 FPU = Floating-point unit  
 C/D = Cache/data port  
 TC = Transfer controller  
 MP = Master processor

The VC can be split into functional blocks, as shown in Figure 1–2. The following subsection describes the blocks.

Figure 1–2. VC Block Diagram



### 1.1.1 VC Functional Blocks

Following are descriptions of the VC functional blocks.

#### ☐ Frame timers

Two identical frame timers provide video timing control. Each timer has its own input frame clock (FCLK) and operates asynchronously with respect to the rest of MVP logic. Each frame timer can be programmed to generate timing pulses on five video signals that can control an image capture or display device. These signals provide:

- Separate or composite synchronization and blanking
- Synchronization to internal or external signals
- Interlaced or noninterlaced frame control
- Limitless screen resolutions

#### ☐ Serial register transfer (SRT) controller

An SRT controller generates SRT cycle requests to the transfer controller (TC) to transfer data into and out of VRAM frame memories. The frame timers indicate to the SRT controller when an SRT is necessary, and the SRT controller generates the required addresses and synchronizes and prioritizes the requests before passing them on to the TC. The SRT controller may also generate packet transfer requests in place of SRT requests.

#### ☐ VC register interface

The VC register interface provides access to the VC registers via the master processor's (MP) 32-bit on-chip register bus.

#### ☐ Multiplexer

The multiplexer allows synchronization of the two frame timers to a single frame clock.

### 1.1.2 Typical Applications

The VC is ideal for applications such as:

- ☐ Video capture and display
- ☐ Image capture and display
- ☐ Printer, copier, or scanner control

## 1.2 The VC Memory Map

The video controller (VC) contains fourteen 32-bit and forty-two 16-bit registers. These registers are accessible to the master processor (MP) as memory-mapped registers. The on-chip memory regions for the VC registers are shown in Figure 1–3. Note that not all memory locations contain valid VC registers. The individual registers and their addresses are discussed in Chapter 3, *Frame Timer Registers*.

Figure 1–3. VC Memory Map

	<u>MP Address</u>
frame timer 0	0x0182 0200 – 0x0182 023F
frame timer 1	0x0182 0240 – 0x0182 027F
reserved	0x0182 0280 – 0x0182 02BF
reserved	0x0182 02C0 – 0x0182 02FF
SRT controller (for frame memory 0)	0x0182 0300 – 0x0182 033F
SRT controller (for frame memory 1)	0x0182 0340 – 0x0182 037F
reserved	0x0182 0380 – 0x0182 03BF
reserved	0x0182 03C0 – 0x0182 03FF

### Note:

The on-chip register data bus used for accessing the VC registers is only 32 bits wide and allows for only word and halfword access. Byte (8-bit) and doubleword (64-bit) load and store instructions must not be used to access these registers.

## 1.2.1 Frame Timer Register Maps

Each frame timer contains twenty-one 16-bit registers. The register addresses for frame timer 0 and frame timer 1 are shown in Figure 1–4 and Figure 1–5, respectively.

Figure 1–4. Frame Timer 0 Register Map

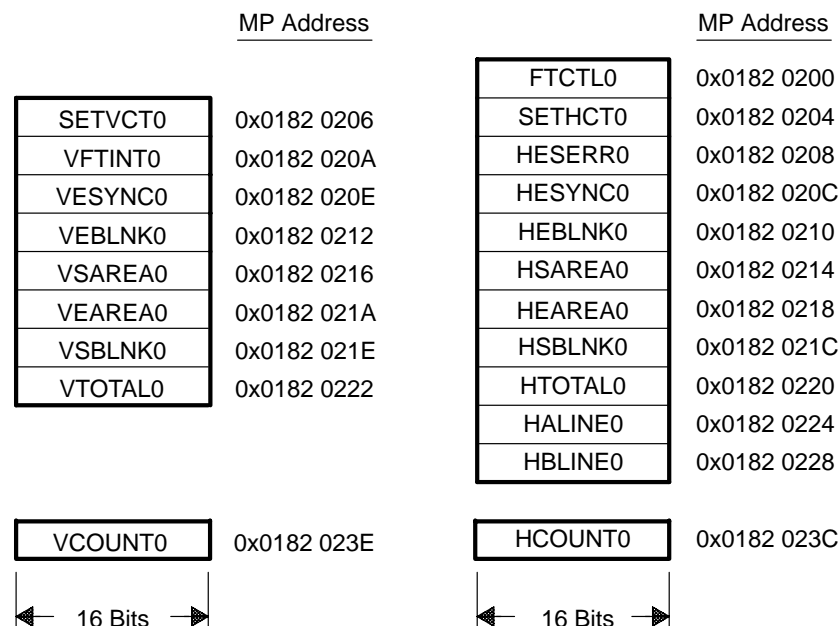
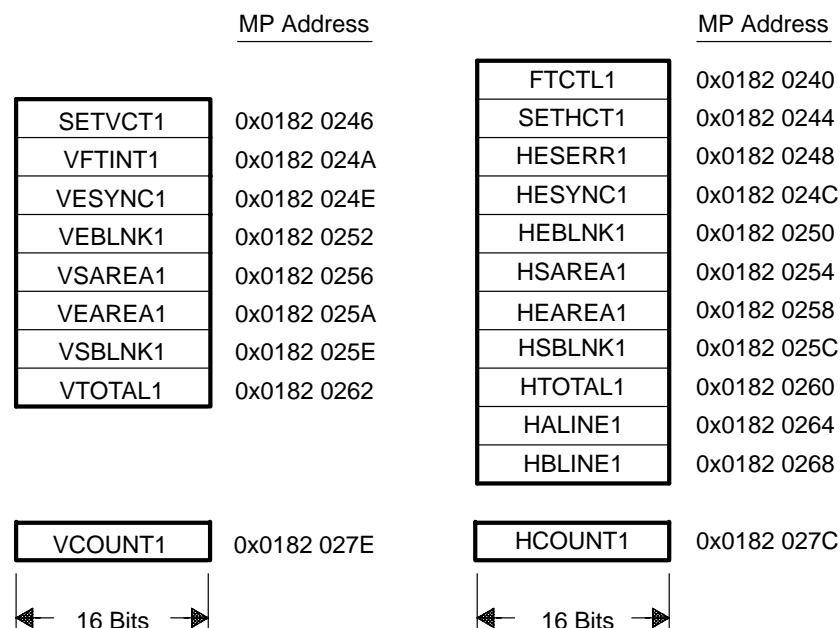


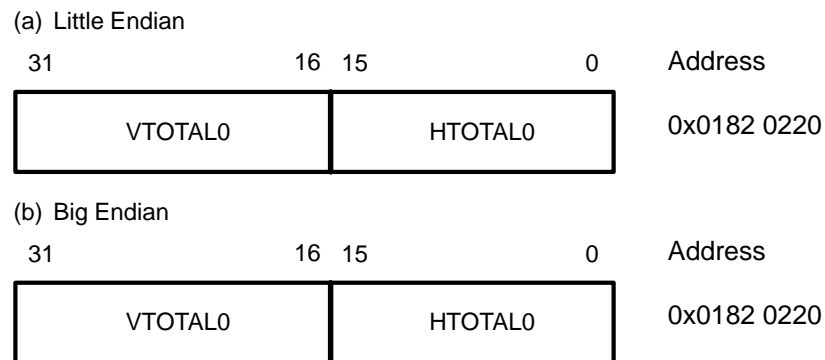
Figure 1–5. Frame Timer 1 Register Map



As Figure 1–4 and Figure 1–5 illustrate, horizontal timing registers are located at even halfword addresses. Their vertical counterparts are located at odd halfword addresses.

Normal procedure is to access a single horizontal or vertical timing register via a 16-bit load or store. The master processor can access the horizontal/vertical timing register pair with a single 32-bit access. The horizontal register is always represented by the 16 least significant bits (LSBs) of the word. The vertical register is always represented by the 16 most significant bits (MSBs) of the word. Since the timing registers are accessed via the MVP's on-chip register bus, the addresses are the same, regardless of the current endian mode, as shown in Figure 1–6.

Figure 1–6. 32-Bit Frame Timer Register Access



The following 16-bit locations return a zero value when read. Writing to these locations has no effect:

- ☐ 0x0182 0202
- ☐ 0x0182 0226
- ☐ 0x0182 022A
- ☐ 0x0182 0242
- ☐ 0x0182 0266
- ☐ 0x0182 026A

## 1.2.2 SRT Controller Register Maps

Two sets of 32-bit registers are associated with the SRT controller—one set for each of the two supported frame memories. Figure 1–7 shows the SRT controller register maps for both frame memories. The registers and register functions for each set are identical and are described in Section 5.2, *SRT Controller Registers*.

Figure 1–7. SRT Controller Register Map

	MP Address		MP Address
FMEMCTL0	0x0182 0300	FMEMCTL1	0x0182 0340
F1STADR0	0x0182 0304	F1STADR1	0x0182 0344
F0STADR0	0x0182 0308	F0STADR1	0x0182 0348
LINEINC0	0x0182 030C	LINEINC1	0x0182 034C
SAMMASK0	0x0182 0310	SAMMASK1	0x0182 0350
NEXTADR0	0x0182 0314	NEXTADR1	0x0182 0354
CRNTADR0	0x0182 033C	CRNTADR1	0x0182 037C
<div> <div></div> <div>32 Bits</div> <div></div> </div>		<div> <div></div> <div>32 Bits</div> <div></div> </div>	

# Signal Descriptions

---

---

---

The MVP video controller (VC) timing signals provide programmable sync (separate or composite) and blanking. This chapter describes the configuration, direction, and timing intervals of each signal.

## Topics

<b>2.1</b>	<b>Timing Control Signals .....</b>	<b>VC:2-2</b>
<b>2.2</b>	<b>Frame Clock Signals .....</b>	<b>VC:2-3</b>
<b>2.3</b>	<b>Horizontal Timing Signals .....</b>	<b>VC:2-4</b>
<b>2.4</b>	<b>Vertical Timing Signals .....</b>	<b>VC:2-5</b>
<b>2.5</b>	<b>Composite Timing Signals .....</b>	<b>VC:2-6</b>
<b>2.6</b>	<b>SRT Controller Signals .....</b>	<b>VC:2-8</b>



## 2.1 Timing Control Signals

The VC timing control signals are summarized in Table 2–1. The following sections describe each signal.

Table 2–1. Timing Control Signals

(a) Frame Timer 0 Signals

Signal Name	Direction	Description	See Page
FCLK0	I	Frame clock for frame timer 0	VC:2-3
$\overline{\text{HSYNC0}}$	I/O/Hi-Z	Horizontal sync on frame timer 0	VC:2-4
$\overline{\text{VSYNC0}}$	I/O/Hi-Z	Vertical sync on frame timer 0	VC:2-5
$\overline{\text{CSYNC0}}/\overline{\text{HBLNK0}}$	I/O/Hi-Z	Composite sync on frame timer 0 / Horizontal blank on frame timer 0	VC:2-6/ VC:2-4
$\overline{\text{CBLNK0}}/\overline{\text{VBLNK0}}$	O	Composite blank on frame timer 0 / Vertical blank on frame timer 0	VC:2-7/ VC:2-5
CAREA0	O	Composite area on frame timer 0	VC:2-6

(b) Frame Timer 1 Signals

Signal Name	Direction	Description	See Page
FCLK1	I	Frame clock for frame timer 1	VC:2-3
$\overline{\text{HSYNC1}}$	I/O/Hi-Z	Horizontal sync on frame timer 1	VC:2-4
$\overline{\text{VSYNC1}}$	I/O/Hi-Z	Vertical sync on frame timer 1	VC:2-5
$\overline{\text{CSYNC1}}/\overline{\text{HBLNK1}}$	I/O/Hi-Z	Composite sync on frame timer 1 / Horizontal blank on frame timer 1	VC:2-6/ VC:2-4
$\overline{\text{CBLNK1}}/\overline{\text{VBLNK1}}$	O	Composite blank on frame timer 1 / Vertical blank on frame timer 1	VC:2-7/ VC:2-5
CAREA1	O	Composite area on frame timer 1	VC:2-6

(c) Serial Register Transfer Signals

Signal Name	Direction	Description	See Page
SCLK0	I	VRAM serial clock 0	VC:2-8
SCLK1	I	VRAM serial clock 1	VC:2-8

**Note:** I = input  
O = output  
Hi-Z = high impedance

## 2.2 Frame Clock Signals

The clock signals, FCLK0 and FCLK1, control the internal video logic for the VC's frame timers. FCLK0 and FCLK1 are derived from the dot clocks of the external video system.

The FCLK0 input clocks frame timer 0. It can also be used to clock frame timer 1 when the frame timers are synchronized. The FCLK1 input clocks frame timer 1.

## 2.3 Horizontal Timing Signals

The bidirectional horizontal timing signals control horizontal timing.  $\overline{\text{CSYNC0/HBLNK0}}$  and  $\overline{\text{CSYNC1/HBLNK1}}$  are multipurpose signals that can be used as composite sync **or** horizontal blanking signals. Following is a description of their use as horizontal blanking signals. For a description of their use as composite sync signals, refer to page VC:2-6.

### ☐ **Horizontal synchronization: $\overline{\text{HSYNC0}}$ and $\overline{\text{HSYNC1}}$**

These signals occur once per line and have a pulse duration defined as an integral number of FCLK periods.  $\overline{\text{HSYNC}}$  signals can be used to generate horizontal retrace or its equivalent for CRTs, LCDs, scanners, printers, and cameras.

You can configure  $\overline{\text{HSYNC}}$  signals for input, output, or high impedance with the  $\overline{\text{HSYNC}}$  pin mode (HPM) bits in the FTCTL register.

### ☐ **Horizontal blanking: $\overline{\text{CSYNC0/HBLNK0}}$ and $\overline{\text{CSYNC1/HBLNK1}}$**

You configure the horizontal blanking signals via the  $\overline{\text{CSYNC0/HBLNK0}}$  pin mode (CPM) bits in the frame timer control (FTCTL) register.

When configured as  $\overline{\text{HBLNK}}$  signals, these signals occur once per line and have a pulse duration defined as an integral number of FCLK periods. Horizontal blanking signals can be used to disable pixel capture or display during horizontal retrace.

In general, the  $\overline{\text{HBLNK}}$  signal is used in conjunction with the  $\overline{\text{VBLNK}}$  signal to determine when pixels are displayed or captured. The  $\overline{\text{HBLNK}}$  signal can also be used for any timing signal that requires repetition at line intervals. When configured as  $\overline{\text{HBLNK}}$ , these signals function as output only.

## 2.4 Vertical Timing Signals

The bidirectional vertical timing signals control vertical timing.  $\overline{\text{CBLNK0}}/\overline{\text{VBLNK0}}$  and  $\overline{\text{CBLNK1}}/\overline{\text{VBLNK1}}$  are multipurpose signals; they can be configured as composite blanking **or** vertical blanking signals. Following is a description of their use as vertical blanking signals. For information on their use as composite blanking signals, refer to page VC:2-7.

### ☐ Vertical synchronization: $\overline{\text{VSYNC0}}$ and $\overline{\text{VSYNC1}}$

These signals occur once per frame (once per field in an interlaced system) and have a pulse duration defined as an integral number of lines (half-lines for interlaced). The vertical sync signals can be used to generate the vertical retrace or its equivalent for CRTs, LCDs, scanners, CCDs, and cameras.

These signals can also be used to indicate the “end of page” to a printer. In general, the vertical sync signals can be used for any timing signal that requires repetition at frame or field intervals. You can configure  $\overline{\text{VSYNC}}$  for input, output, or high impedance with the  $\overline{\text{VSYNC}}$  pin mode (VPM) bits in the FTCTL register.

### ☐ Vertical blanking: $\overline{\text{CBLNK0}}/\overline{\text{VBLNK0}}$ and $\overline{\text{CBLNK1}}/\overline{\text{VBLNK1}}$

These signals can be configured as vertical blanking signals via the CPM bits in the FTCTL register.  $\overline{\text{VBLNK}}$  is always an output.

$\overline{\text{CBLNK}}/\overline{\text{VBLNK}}$  signals are configured as  $\overline{\text{VBLNK}}$  in conjunction with  $\overline{\text{HBLNK}}$ . The  $\overline{\text{VBLNK}}$  signal pulses occur once per frame (once per field in an interlaced system) and have a pulse duration defined as an integral number of lines (half-lines for interlaced).  $\overline{\text{CBLNK}}/\overline{\text{VBLNK}}$  signals can be used to disable pixel capture/display during vertical retrace.

In general,  $\overline{\text{VBLNK}}$  is used in conjunction with  $\overline{\text{HBLNK}}$  to determine when pixels are displayed or captured. They are used to control both the capture/display device and the frame memory. The  $\overline{\text{VBLNK}}$  signal can be used for any timing signal that requires repetition at frame intervals.

## 2.5 Composite Timing Signals

Following are descriptions of the composite timing signals.  $\overline{\text{CSYNC0/HBLNK0}}$  and  $\overline{\text{CSYNC1/HBLNK1}}$  can be configured independently as composite sync **or** horizontal blanking signals. For information on the use of  $\overline{\text{CSYNC0/HBLNK0}}$  and  $\overline{\text{CSYNC1/HBLNK1}}$  as horizontal blanking signals, refer to page VC:2-4.  $\overline{\text{CBLNK0/VBLNK0}}$  and  $\overline{\text{CBLNK1/VBLNK1}}$  can be configured as composite blanking **or** vertical blanking signals. For information on the use of  $\overline{\text{CBLNK0/VBLNK0}}$  and  $\overline{\text{CBLNK1/VBLNK1}}$  as vertical blanking signals, refer to page VC:2-5.

### ☐ **Composite area: CAREA0 and CAREA1**

These signals represent the logic AND of the internal horizontal and vertical area signals. They are active (high) if both horizontal or vertical area signals are active (high).

CAREA signals provide a general-purpose region that you can use to define a special region, such as an overscan boundary. These signals behave identically in both noninterlaced and interlaced mode to define a purely rectangular region. CAREA is always an output.

### ☐ **Composite sync: $\overline{\text{CSYNC0/HBLNK0}}$ and $\overline{\text{CSYNC1/HBLNK1}}$**

You can configure the composite sync signals via the CPM bits in the FTCTL register.

The  $\overline{\text{CSYNC}}$  signal combines horizontal and vertical timing information into a single signal that is more complex than either horizontal or vertical sync. The  $\overline{\text{CSYNC}}$  signal is compatible with the existing standards used for image capture and display (RS-170, RS-330, RS-343, NTSC, PAL, and SECAM).

During an active frame,  $\overline{\text{CSYNC}}$  looks exactly like  $\overline{\text{HSYNC}}$ . However, special serration pulses are generated during the vertical sync interval, and equalization pulses are generated before and after vertical sync. Serration and equalization pulses have durations defined as an integral number of FCLK periods and occur every halfline in interlaced mode. You can configure  $\overline{\text{CSYNC}}$  for input, output, or high impedance via the CPM bits in the FTCTL register.

□ **Composite blanking:  $\overline{\text{CBLNK0}}$ / $\overline{\text{VBLNK0}}$  and  $\overline{\text{CBLNK1}}$ / $\overline{\text{VBLNK1}}$**

You can configure  $\overline{\text{CBLNK0}}$ / $\overline{\text{VBLNK0}}$  and  $\overline{\text{CBLNK1}}$ / $\overline{\text{VBLNK1}}$  using the CPM bits in the FTCTL register. When configured as  $\overline{\text{CBLNK}}$ , these signals represent the composite blanking signals.  $\overline{\text{CBLNK}}$  is always an output.

These signals are configured as  $\overline{\text{CBLNK}}$  in conjunction with  $\overline{\text{CSYNC}}$ . They represent the (negative logic) logical OR of the horizontal and vertical blanking signals. They are active (low) if either horizontal or vertical blanking signals are active (low).  $\overline{\text{CBLNK}}$  can be used to disable pixel capture/display during both horizontal and vertical retrace periods.

## 2.6 SRT Controller Signals

The serial register transfer (SRT) controller signals, SCLK0 and SCLK1, are input clocks used by the SRT controller to track the VRAM tap point when using midline reload. SCLK0 should be the shift clock used by the VRAMs that make up frame memory 0. SCLK1 should be the shift clock used by the VRAMs that make up frame memory 1.

# Frame Timer Registers

---

---

---

The video controller (VC) contains two identical frame timers: frame timer 0 and frame timer 1. Each frame timer is controlled by a set of on-chip registers. This chapter describes these registers, which are listed alphabetically.

## Topics

- |            |   |               |
|------------|---|---------------|
| <b>3.1</b> | <b>Summary of the Frame Timer Registers . . . . .</b>     | <b>VC:3-2</b> |
| <b>3.2</b> | <b>Description of the Frame Timer Registers . . . . .</b> | <b>VC:3-4</b> |



### 3.1 Summary of the Frame Timer Registers

The tables below list the address of each register. Table 3–1 lists the frame timer control registers. Table 3–2 lists the horizontal timing registers, which control the timing of horizontal signals and horizontal components of composite signals. Table 3–3 lists the vertical timing registers, which control the timing of vertical signals and vertical components of composite signals.

Table 3–1. Frame Timer Control Register Addresses

Description	Register	Address	Register	Address	See Page
Frame timer control register	FTCTL0	0x0182 0200	FTCTL1	0x0182 0240	VC:3-4

Table 3–2. Horizontal Timing Register Addresses

Description	Register	Address	Register	Address	See Page
Horizontal area line event register	HALINE0	0x0182 0224	HALINE1	0x0182 0264	VC:3-8
Horizontal blanking line event register	HBLINE0	0x0182 0228	HBLINE1	0x0182 0268	VC:3-9
Horizontal counter	HCOUNT0	0x0182 023C	HCOUNT1	0x0182 027C	VC:3-10
Horizontal end area register	HEAREA0	0x0182 0218	HEAREA1	0x0182 0258	VC:3-11
Horizontal end blanking register	HEBLNK0	0x0182 0210	HEBLNK1	0x0182 0250	VC:3-12
Horizontal end serration register	HESERR0	0x0182 0208	HESERR1	0x0182 0248	VC:3-13
Horizontal end sync register	HESYNC0	0x0182 020C	HESYNC1	0x0182 024C	VC:3-14
Horizontal start area register	HSAREA0	0x0182 0214	HSAREA1	0x0182 0254	VC:3-15
Horizontal start blanking register	HSBLNK0	0x0182 021C	HSBLNK1	0x0182 025C	VC:3-16
Horizontal total register	HTOTAL0	0x0182 0220	HTOTAL1	0x0182 0260	VC:3-17
Set HCOUNT register	SETHCT0	0x0182 0204	SETHCT1	0x0182 0244	VC:3-18

Table 3–3. Vertical Timing Register Addresses

<b>Description</b>	<b>Register</b>	<b>Address</b>	<b>Register</b>	<b>Address</b>	<b>See Page</b>
Set VCOUNT register	SETVCT0	0x0182 0206	SETVCT1	0x0182 0246	VC:3-19
Vertical counter	VCOUNT0	0x0182 023E	VCOUNT1	0x0182 027E	VC:3-20
Vertical end area register	VEAREA0	0x0182 021A	VEAREA1	0x0182 025A	VC:3-21
Vertical end blanking register	VEBLNK0	0x0182 0212	VEBLNK1	0x0182 0252	VC:3-22
Vertical end sync register	VESYNC0	0x0182 020E	VESYNC1	0x0182 024E	VC:3-23
Vertical frame timer interrupt register	VFTINT0	0x0182 020A	VFTINT1	0x0182 024A	VC:3-24
Vertical start area register	VSAREA0	0x0182 0216	VSAREA1	0x0182 0256	VC:3-25
Vertical start blanking register	VSBLNK0	0x0182 021E	VSBLNK1	0x0182 025E	VC:3-26
Vertical total register	VTOTAL0	0x0182 0222	VTOTAL1	0x0182 0262	VC:3-27

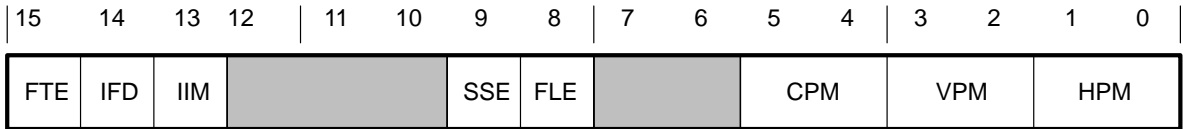
### 3.2 Description of the Frame Timer Registers

This section contains an alphabetical list and description of the registers for frame timers 0 and 1.

**Name**                      Frame Timer Control Register: FTCTL0/FTCTL1

**Addresses**              FTCTL0: 0x0182 0200  
FTCTL1: 0x0182 0240

**Format**



**Fields**

Bits	Name	Function
1–0	HPM	$\overline{\text{HSYNC}}$ pin mode
3–2	VPM	$\overline{\text{VSYNC}}$ pin mode
5–4	CPM	$\overline{\text{CSYNC}}/\overline{\text{HBLNK}}$ pin mode
8	FLE	Frame lock enable
9	SSE	Set sync enable
13	IIM	Interlaced interrupt mode
14	IFD	Interlaced frame disable
15	FTE	Frame timer enable

**Description**

The frame timer control (FTCTL) register contains the mode bits that determine the frame timer's behavior. Upon reset, the FTCTL register defaults to all 0s.

☐ **HPM** ( $\overline{\text{HSYNC}}$  pin mode)

Bits 1 and 0 of the FTCTL register determine whether the  $\overline{\text{HSYNC}}$  pin is configured as:

- High impedance
- Input that can lock the frame timer to an external timing source
- Output that can control an external device

Table 3–4 shows the values for these bits.

Table 3–4.  $\overline{\text{HSYNC}}$  Pin Modes (HPM)

Bits		
1	0	$\overline{\text{HSYNC}}$ configured as:
0	0	High impedance
0	1	Input
1	0	Output
1	1	Reserved

☐ **VPM** ( $\overline{\text{VSYNC}}$  pin mode)

Bits 3 and 2 of the FTCTL register determine whether the  $\overline{\text{VSYNC}}$  pin is configured as:

- High impedance
- Input that can lock the frame timer to an external timing source
- Output that can control an external device

Table 3–5 shows the values for these bits.

Table 3–5.  $\overline{\text{VSYNC}}$  Pin Modes (VPM)

Bits		
3	2	$\overline{\text{VSYNC}}$ configured as:
0	0	High impedance
0	1	Input
1	0	Output
1	1	Reserved

☐ **CPM** ( $\overline{\text{CSYNC}}$ / $\overline{\text{HBLNK}}$  pin mode)

Bits 5 and 4 of the FTCTL register determine whether the  $\overline{\text{CSYNC}}$ / $\overline{\text{HBLNK}}$  pin is configured as:

- $\overline{\text{CSYNC}}$  high impedance
- $\overline{\text{CSYNC}}$  input
- $\overline{\text{CSYNC}}$  output
- $\overline{\text{HBLNK}}$  output

These bits also determine whether  $\overline{\text{CBLNK}}$ / $\overline{\text{VBLNK}}$  is configured as:

- $\overline{\text{CBLNK}}$  output
- $\overline{\text{VBLNK}}$  output

Table 3–6 shows the values for these bits.

Table 3–6.  $\overline{\text{CSYNC}}$ / $\overline{\text{HBLNK}}$  and  $\overline{\text{CBLNK}}$ / $\overline{\text{VBLNK}}$  Pin Modes (CPM)

Bits		$\overline{\text{CSYNC}}$ / $\overline{\text{HBLNK}}$	$\overline{\text{CBLNK}}$ / $\overline{\text{VBLNK}}$
5	4	configured as:	configured as:
0	0	$\overline{\text{CSYNC}}$ — High impedance	$\overline{\text{CBLNK}}$ — Output
0	1	$\overline{\text{CSYNC}}$ — Input	$\overline{\text{CBLNK}}$ — Output
1	0	$\overline{\text{CSYNC}}$ — Output	$\overline{\text{CBLNK}}$ — Output
1	1	$\overline{\text{HBLNK}}$ — Output	$\overline{\text{VBLNK}}$ — Output

☐ **FLE** (frame lock enable)

Bit 8 (FLE) is valid for the FTCTL1 register only. Setting FLE in the FTCTL0 register has no effect.

To lock the frame timers together (synchronized), set FLE to 1. When frame lock is enabled, frame timer 1 is clocked by FCLK0. See Section 4.11, *Frame Lock*.

When frame lock is disabled (FLE = 0), frame timer 1 is clocked by FCLK1 independently of frame timer 0.

☐ **SSE** (set synchronization enable)

Bit 9 (SSE) of the FTCTL register determines how the frame timers operate when they are synchronizing to an external source.

When SSE is set to 0, external synchronization pulses do **not** cause the corresponding horizontal or vertical sync output to be activated. This allows precise synchronization to the external source via the SETHCT and/or SETVCT registers.

Setting SSE to 1 enables external synchronization pulses to activate the corresponding sync output. This mode can be used when only a crude synchronization is required. See subsection 4.8.3, *Sync Output Operation*.

SSE has no effect on internal synchronization operation.

☐ **IIM** (interlaced interrupt mode)

Bit 13 (IIM) of the FTCTL register selects between field and frame interrupt mode **when the frame timer is in interlaced mode**. When IIM is 0, frame interrupt mode is selected, and a frame timer interrupt is generated only once per frame.

Setting bit IIM to 1 selects field interrupt mode, and a frame timer interrupt will be generated **twice** per frame (once in each field). See subsection 4.7.3, *Vertical Frame Timer Interrupt Programming* and Figure 4–22, *Programming VFTINT for NTSC (IIM = 1)*, on page VC:4-40.

☐ **IFD** (interlaced frame disable)

Bit 14 (IFD) of the FTCTL register selects interlaced or noninterlaced frame timer operation. Setting IFD to 1 selects noninterlaced operation. Setting IFD to 0 selects interlaced operation.

☐ **FTE** (frame timer enable)

Setting bit 15 (FTE) of the FTCTL register to 1 enables image capture/display. When FTE is set to 0, blanking and area outputs are forced low. SRT events still occur, and the frame timer interrupt is still active.

## HALINE0/HALINE1 Horizontal Area Line Event Register

**Name** Horizontal Area Line Event Register: HALINE0/HALINE1

**Addresses** HALINE0: 0x0182 0224  
HALINE1: 0x0182 0264

### Format



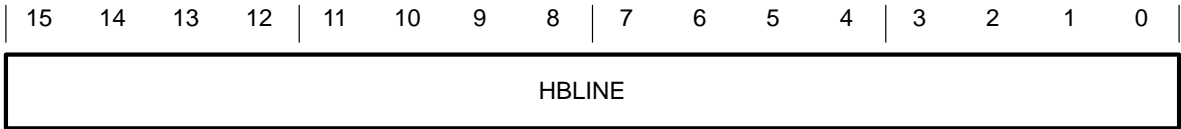
**Description** The horizontal area line event (HALINE) register does not affect the frame timer's video timing signals. It determines when line events are generated when the serial register transfer (SRT) request timing is controlled by **area**. You should normally program the HALINE register to be equal to or slightly larger than the horizontal end area (HEAREA) register (see page VC:5-19).

**Equation** 
$$\text{HALINE} = (\text{the number of FCLKs from the start of horizontal sync to the active area line event}) - 1$$

**Name** Horizontal Blanking Line Event Register: HBLINE0/HBLINE1

**Addresses** HBLINE0: 0x0182 0228  
HBLINE1: 0x0182 0268

**Format**



**Description** The horizontal blanking line event (HBLINE) register does not affect the frame timer’s video timing signals. HBLINE determines when line events are generated when the serial register transfer (SRT) request timing is controlled by **blanking**. You should normally program the HBLINE register to be equal to or slightly larger than the horizontal start blanking (HSBLNK) register (see page VC:5-19).

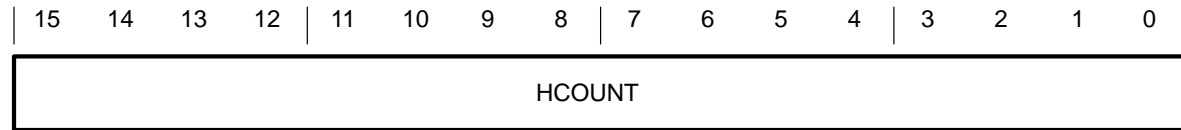
**Equation** 
$$\text{HBLINE} = (\text{the number of FCLKs from the start of horizontal sync to the end blanking line event}) - 1$$



**Name** Horizontal Counter: HCOUNT0/HCOUNT1

**Addresses** HCOUNT0: 0x0182 023C  
HCOUNT1: 0x0182 027C

**Format**



**Description**

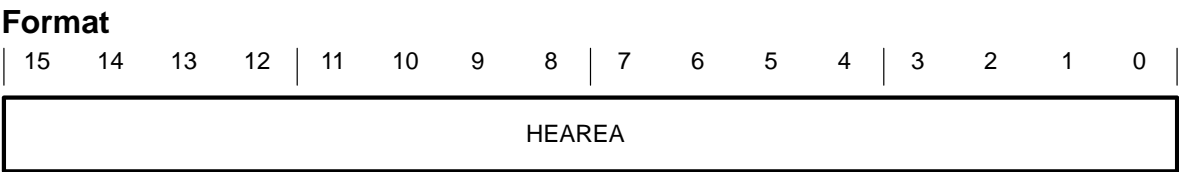
The horizontal counter (HCOUNT) is incremented by the frame clock (FCLK). HCOUNT generates horizontal timing signal transitions. When the condition  $\text{HCOUNT} = \text{HTOTAL}$  (horizontal total register) occurs, HCOUNT is reset to 0 on the next FCLK. Since HCOUNT is dynamically updated, HCOUNT cannot be reliably read while FCLK is running.

When you configure the frame timer for **interlaced composite-sync** operation, equalization and serration pulses occur on the  $\overline{\text{CSYNC}}$  output every halfline during the equalization and serration regions within vertical blanking (see Appendix A, *Programming Procedures and Examples*). Within these regions, the condition  $\text{HCOUNT} = 1/2 \text{ HTOTAL}$  causes HCOUNT to be reset to 0 on the next FCLK.

If external sync is enabled, HCOUNT is reloaded with the value in the set horizontal count (SETHCT) register when horizontal sync occurs ( $\overline{\text{HSYNC}}$  or  $\overline{\text{CSYNC}}$  input is driven low).

**Name**                      Horizontal End Area Register: HEAREA0/HEAREA1

**Addresses**                HEAREA0: 0x0182 0218  
                                 HEAREA1: 0x0182 0258



**Description**                The horizontal end area (HEAREA) register determines the end-point of the horizontal component of the area signal output on CAREA. When the condition HCOUNT = HEAREA occurs, the horizontal component of CAREA ends on the next FCLK. For more information on CAREA, see page VC:2-6.

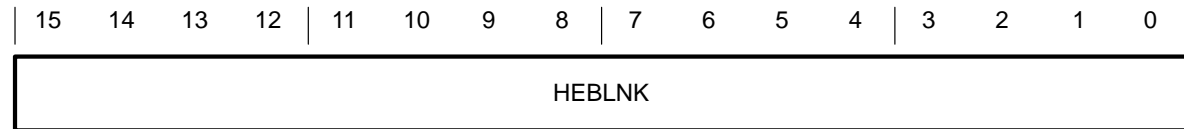
**Equation**                      HEAREA = (the number of FCLKs from the start of horizontal sync to the end of horizontal area) – 1

## HEBLNK0/HEBLNK1 Horizontal End Blanking Register

**Name** Horizontal End Blanking Register: HEBLNK0/HEBLNK1

**Addresses** HEBLNK0: 0x0182 0210  
HEBLNK1: 0x0182 0250

### Format



**Description** The horizontal end blanking (HEBLNK) register determines the endpoint of the horizontal blanking pulses output on the  $\overline{\text{HBLNK}}$  and  $\overline{\text{CBLNK}}$  pins. When the condition  $\text{HCOUNT} = \text{HEBLNK}$  occurs, the horizontal blanking pulse ends on the next FCLK.

**Equation** 
$$\text{HEBLNK} = (\text{the number of FCLKs from the start of horizontal sync to the end of horizontal blanking}) - 1$$

<b>Name</b>	Horizontal End Serration Register: HESERR0/HESERR1
<b>Addresses</b>	HESERR0: 0x0182 0208 HESERR1: 0x0182 0248

<b>Format</b>	
	15   14   13   12       11   10   9   8       7   6   5   4       3   2   1   0
	<div>HESERR</div>

<b>Description</b>	The horizontal end serration (HESERR) register determines the endpoint of the serration pulses output on the $\overline{\text{CSYNC}}$ signal during the serration period (active vertical sync). When the condition $\text{HCOUNT} = \text{HESERR}$ occurs, the serration pulse ends on the next FCLK.
--------------------	---

**Note:**

Serration timing is applicable only when you configure the frame timer for composite mode. HESERR can be left unprogrammed otherwise (see Appendix A, *Programming Procedures and Examples*).

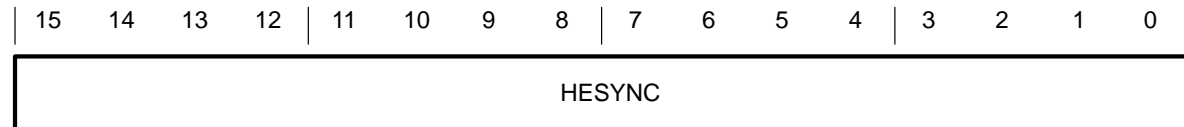
<b>Equation</b>	$\text{HESERR} = (\text{the number of FCLKs from the start of horizontal sync to the end of serration pulse}) - 1$
-----------------	--

## HESYNC0/HESYNC1 Horizontal End Synchronization Register

**Name** Horizontal End Synchronization Register: HESYNC0/HESYNC1

**Addresses** HESYNC0: 0x0182 020C  
HESYNC1: 0x0182 024C

### Format



### Description

The horizontal end synchronization (HESYNC) register determines the endpoint of the horizontal sync pulses output on HSYNC and CSYNC. When the condition  $HCOUNT = HESYNC$  occurs, the horizontal sync pulse ends on the next FCLK.

When you configure the frame timer for **interlaced composite** mode, the condition  $HCOUNT = 1/2 HESYNC$  causes the end of the equalization pulses output on CSYNC. This requires programming HESYNC to an **odd** number so that length of the horizontal sync pulses is an even number of FCLKs and thus divisible into halves.

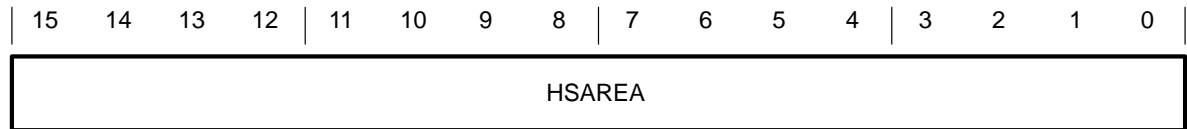
### Equation

$$HESYNC = (\text{the number of FCLKs from the start of horizontal sync to the end of horizontal sync pulse}) - 1$$

**Name** Horizontal Start Area Register: HSAREA0/HSAREA1

**Addresses** HSAREA0: 0x0182 0214  
HSAREA1: 0x0182 0254

**Format**



**Description** The horizontal start area (HSAREA) register determines the starting point of the horizontal component of the area signal output on CAREA. When the condition HCOUNT = HSAREA occurs, the horizontal component of CAREA begins on the next FCLK. For more information on CAREA, see page VC:2-6.

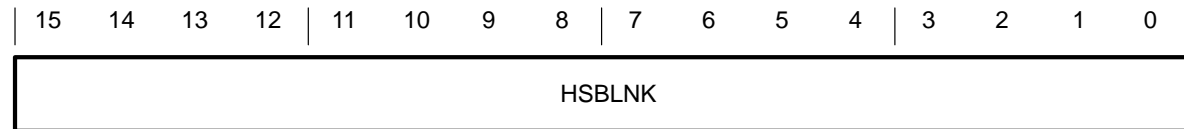
**Equation** 
$$\text{HSAREA} = (\text{the number of FCLKs from the start of horizontal sync to the start of horizontal area}) - 1$$

## HSBLNK0/HSBLNK1 Horizontal Start Blanking Register

**Name** Horizontal Start Blanking Register: HSBLNK0/HSBLNK1

**Addresses** HSBLNK0: 0x0182 021C  
HSBLNK1: 0x0182 025C

### Format



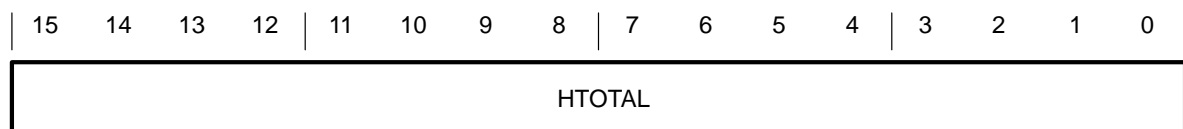
**Description** The horizontal start blanking (HSBLNK) register determines the starting point of horizontal blanking pulses output on HBLNK or CBLNK. When the condition  $HCOUNT = HSBLNK$  occurs, the horizontal blanking pulse begins on the next FCLK.

**Equation**  $HSBLNK = (\text{the number of FCLKs from the start of horizontal sync to the start of horizontal blanking}) - 1$

**Name** Horizontal Total Register: HTOTAL0/HTOTAL1

**Addresses** HTOTAL0: 0x0182 0220  
HTOTAL1: 0x0182 0260

**Format**



**Description** The horizontal total (HTOTAL) register defines the upper limit for HCOUNT. HCOUNT counts from 0 to HTOTAL. When the condition HCOUNT = HTOTAL occurs, HCOUNT is reset on the next FCLK. Thus, HCOUNT is reset every HTOTAL + 1 FCLK cycles. The condition HCOUNT = HTOTAL also determines when the horizontal and composite sync pulses are activated.

When you configure the frame timer for **interlaced composite** mode, equalization and serration pulses occur every halfline at certain times within vertical blanking. At these times, the condition HCOUNT = 1/2 HTOTAL causes the equalization or serration pulses to start, and HCOUNT is reset. This requires programming HTOTAL to an odd number so that the length of a horizontal line is an even number of FCLKs and thus divisible into halves (see Appendix A, *Programming Procedures and Examples*).

**Equation** 
$$\text{HTOTAL} = (\text{the number of FCLKs per horizontal line}) - 1$$

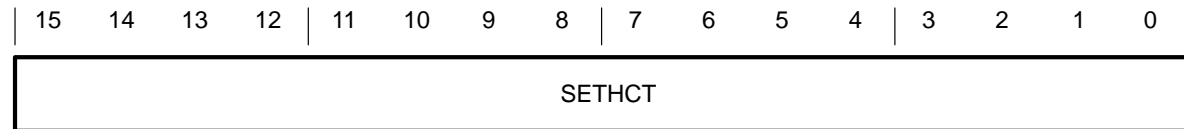


## SETHCT0/SETHCT1 Set HCOUNT Register

**Name** Set HCOUNT Register: SETHCT0/SETHCT1

**Addresses** SETHCT0: 0x0182 0204  
SETHCT1: 0x0182 0244

### Format



### Description

If the frame timer is in **external sync** mode, the value in the set horizontal counter (SETHCT) register is loaded into HCOUNT when a horizontal sync signal is input on  $\overline{\text{HSYNC}}$  or  $\overline{\text{CSYNC}}$ . To advance or retard the frame timer by a specified number of FCLKs, the SETHCT register is used to set HCOUNT to a non-zero value when the frame timer is synchronizing to an external timing source.

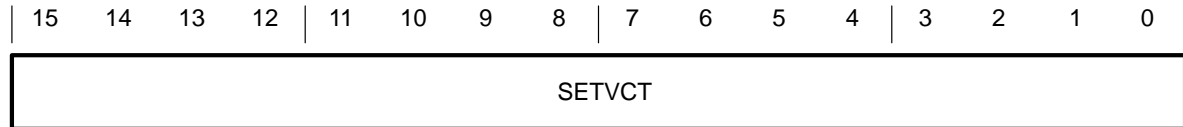
#### Note:

When the frame timer is in **internal sync** mode, SETHCT can be left unprogrammed.

**Name** Set VCOUNT Register: SETVCT0/SETVCT1

**Addresses** SETVCT0: 0x0182 0206  
SETVCT1: 0x0182 0246

**Format**



**Description**

If the frame timer is in **external sync** mode, the value in the set vertical counter (SETVCT) register is loaded into VCOUNT when a vertical sync signal is input on  $\overline{\text{VSYNC}}$  or  $\overline{\text{CSYNC}}$ . To advance or retard the frame timer by a specified number of lines (noninterlaced) or halflines (interlaced), the SETVCT register is used to set VCOUNT to a nonzero value when the frame timer is synchronizing to an external timing source.

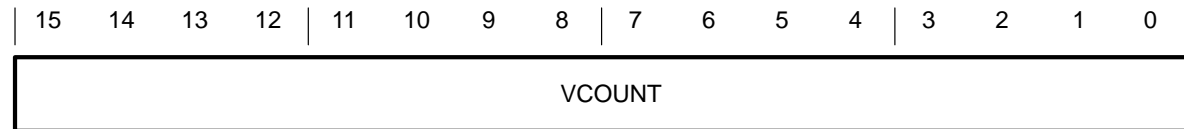
**Note:**

When the frame timer is in **internal sync** mode, SETVCT can be left unprogrammed.

**Name** Vertical Counter: VCOUNT0/VCOUNT1

**Addresses** VCOUNT0: 0x0182 023E  
VCOUNT1: 0x0182 027E

**Format**



**Description**

The value in the vertical counter (VCOUNT) is incremented at the end of each horizontal line when the condition  $HCOUNT = HTOTAL$  is true. If the condition  $VCOUNT = VTOTAL$  occurs, then VCOUNT is reset to 0 at the end of the next horizontal line.

If you configure the frame timer for **interlaced** mode, VCOUNT is also incremented at the midpoint of each line when the condition  $HCOUNT = 1/2 HTOTAL$  is met (see Appendix A, *Programming Procedures and Examples*).

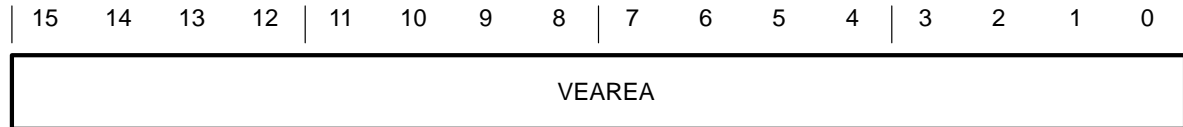
If you enable external sync, VCOUNT is reloaded with the value in SETVCT when vertical sync occurs ( $\overline{VSYNC}$  input is driven low) or when the first serration pulse occurs on  $\overline{CSYNC}$ .

There is no synchronization between VCOUNT and the MP; therefore, care should be taken when reading from and writing to VCOUNT.

**Name** Vertical End Area Register: VEAREA0/VEAREA1

**Addresses** VEAREA0: 0x0182 021A  
VEAREA1: 0x0182 025A

**Format**



**Description** The vertical end area (VEAREA) register determines the end-point of the vertical component of the area signal output on CAREA. The vertical component of the area ends at the end of the line in which VCOUNT = VEAREA.

In **interlaced mode**, the vertical component of CAREA **does not** change in the middle of a line. The condition VCOUNT = VEAREA occurs at the midpoint of a line in one field but has no effect until the end of the line. This means that the number of halflines from the start of vertical sync to the start of vertical area is different for the even and odd fields. You should program VEAREA with the value of VCOUNT that is common to the last even and odd lines before the vertical component of CAREA is to be inactive. (See Section 4.7, *Programming VSAREA, VEAREA, and VFTINT in Interlaced Mode*, for detailed programming examples.)

**Equations**

**Noninterlaced:**  $VEAREA = (\text{the number of lines from the start of vertical sync to the end of vertical area}) - 1$

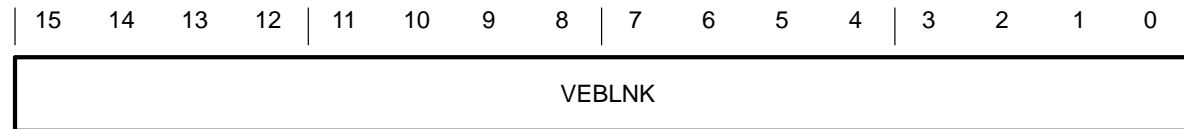
**Interlaced:**  $VEAREA = (\text{the number of halflines from the start of vertical sync to the end of vertical area}) - 1$  (in whichever field vertical area first goes inactive)

## VEBLNK0/VEBLNK1 Vertical End Blanking Register

**Name** Vertical End Blanking Register: VEBLNK0/VEBLNK1

**Addresses** VEBLNK0: 0x0182 0212  
VEBLNK1: 0x0182 0252

### Format



**Description** The vertical end blanking (VEBLNK) register determines the end-point of the vertical blanking pulses output on the  $\overline{\text{VBLNK}}$  or  $\overline{\text{CBLNK}}$  pin. The vertical blanking pulse ends at the end of the line (noninterlaced) or halfline (interlaced) during which  $\text{VCOUNT} = \text{VEBLNK}$ .

**Equations**

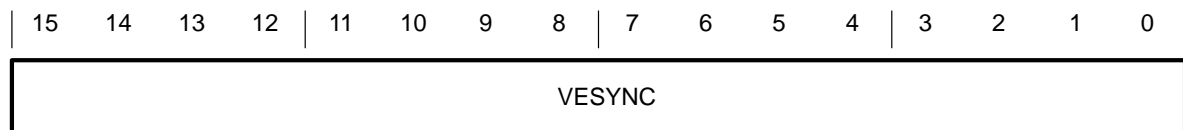
**Noninterlaced:**  $\text{VEBLNK} = (\text{the number of lines from start of vertical sync to the end of vertical blanking}) - 1$

**Interlaced:**  $\text{VEBLNK} = (\text{the number of halflines from start of vertical sync to the end of vertical blanking}) - 1$

**Name** Vertical End Synchronization Register: VESYNC0/VESYNC1

**Addresses** VESYNC0: 0x0182 020E  
VESYNC1: 0x0182 024E

**Format**



**Description** The vertical end synchronization (VESYNC) register determines the endpoint of the vertical sync pulses output on  $\overline{\text{VSYNC}}$  and  $\overline{\text{CSYNC}}$ . The vertical sync pulse ends at the end of the line **(noninterlaced)** or halfline **(interlaced)** during which  $\text{VCOUNT} = 1/2 \text{ VESYNC}$ .

When the frame timer is configured for **composite** mode, the condition  $\text{VCOUNT} = 1/2 \text{ VESYNC}$  also marks the end of the serration region.

In **interlaced composite** mode,  $\text{VCOUNT} = 1/2 \text{ VESYNC}$  marks the beginning of the postequalization region, causing equalization pulses to be output on  $\overline{\text{CSYNC}}$ . The condition  $\text{VCOUNT} = \text{VESYNC}$  indicates the end of the postequalization region, at which point  $\overline{\text{CSYNC}}$  reverts back to outputting horizontal sync pulses. This constrains the postequalization region to be the same length (same number of pulses) as the serration region.

**Equations**

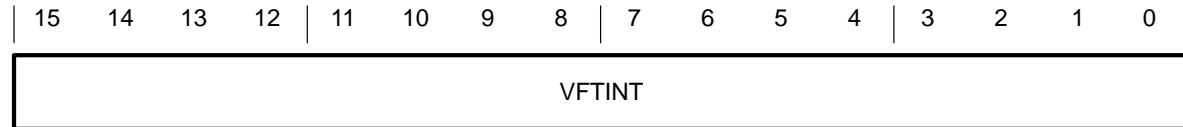
**Noninterlaced:**  $\text{VESYNC} = (\text{Twice the number of lines in vertical sync}) - 1$

**Interlaced:**  $\text{VESYNC} = (\text{Twice the number of halflines in vertical sync}) - 1$

**Name** Vertical Frame Timer Interrupt Register: VFTINT0/VFTINT1

**Addresses** VFTINT0: 0x0182 020A  
VFTINT1: 0x0182 024A

**Format**



**Description**

The vertical frame timer interrupt (VFTINT) register provides a mechanism for interrupting the master processor (MP) at a particular point in a frame. This facilitates the synchronization of software to the VC. The interrupt signal to the MP (bit f0 or f1 of the MP's INTPEN register) is always generated at the **end** of the specified horizontal line during which the condition  $VCOUNT = VFTINT$  occurs.

In interlaced mode, the interrupt may operate in either field or frame mode, as specified by the IIM bit in the FTCTL register:

- ☐ When  $IIM = 0$ , frame mode is enabled, and an interrupt is generated **once** per frame. The interrupt occurs in either the odd or even field at the end of the line where  $VCOUNT = VFTINT$ .
- ☐ When  $IIM = 1$ , field mode is selected, and an interrupt is generated **twice** per frame. Since the interrupt always occurs at the end of the horizontal line (not the halfline) where  $VCOUNT = VFTINT$ , the number of halflines from the start of vertical sync to the interrupt point is different for the two fields. You should program VFTINT to equal the number of the line in the field in which the interrupt first occurs. See page VC:4-39 for a detailed programming example.

**Equations**

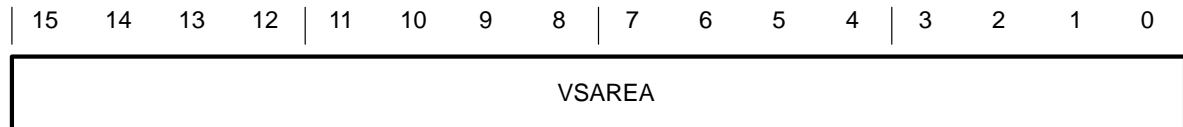
**Noninterlaced:**  $VFTINT = (\text{the number of lines from the start of vertical sync to the interrupt line}) - 1$

**Interlaced:**  $VFTINT = (\text{the number of halflines from the start of vertical sync to the interrupt line}) - 1$  (in whichever field the interrupt first occurs)

**Name** Vertical Start Area Register: VSAREA0/VSAREA1

**Addresses** VSAREA0: 0x0182 0216  
VSAREA1: 0x0182 0256

**Format**



**Description** The vertical start area (VSAREA) register determines the starting point of the vertical component of the area signal output on CAREA. The vertical component of the area begins at the end of the line during which VCOUNT = VSAREA.

In **interlaced** mode, the vertical component of CAREA does **not** change in the middle of a line. The condition VCOUNT = VSAREA occurs during the beginning halfline in one field but has no effect until the end of the line. This means that the number of halflines from the start of vertical synchronization to the start of vertical area is different for the even and odd fields. You should program VSAREA with the value of VCOUNT that is common to the last even and odd lines before the vertical component of CAREA is to become active.

See Section 4.7, *Programming VSAREA, VEAREA, and VFTINT in Interlaced Mode*, for detailed programming examples, and see subsection 4.7.4, *Alternative Programming for VFTINT, VSAREA, and VEAREA* for programming alternatives.

**Equations**

**Noninterlaced:** VSAREA = (the number of lines from the start of vertical synchronization to the start of vertical area) – 1

**Interlaced:** VSAREA = (the number of halflines from the start of vertical synchronization to the start of vertical area) – 1  
(in whichever field vertical area first becomes active)

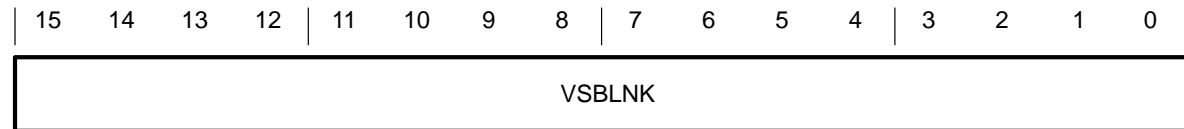


## VSBLNK0/VSBLNK1 Vertical Start Blanking Register

**Name** Vertical Start Blanking Register: VSBLNK0/VSBLNK1

**Addresses** VSBLNK0: 0x0182 021E  
VSBLNK1: 0x0182 025E

### Format



**Description** The vertical start blanking (VSBLNK) register determines the starting point of vertical blanking pulses output on  $\overline{\text{VBLNK}}$  or  $\overline{\text{CBLNK}}$ . The vertical blanking pulse begins at the **end** of the line (noninterlaced) or halfline (interlaced) during which  $\text{VCOUNT} = \text{VSBLNK}$ .

In interlaced composite mode,  $\text{VCOUNT} = \text{VSBLNK}$  also marks the beginning of the pre-equalization region.

**Equations**

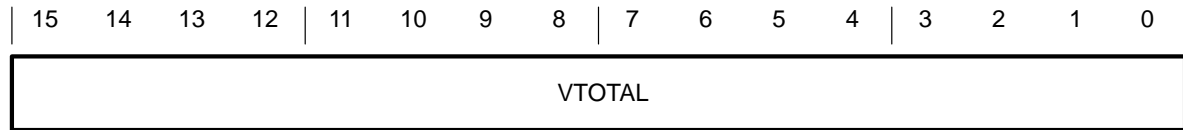
**Noninterlaced:**  $\text{VSBLNK} = (\text{the number of lines from the start of vertical synchronization to the start of vertical blanking}) - 1$

**Interlaced:**  $\text{VSBLNK} = (\text{the number of halflines from the start of vertical synchronization to the start of vertical blanking}) - 1$

**Name** Vertical Total Register: VTOTAL0/VTOTAL1

**Addresses** VTOTAL0: 0x0182 0222  
VTOTAL1: 0x0182 0262

**Format**



**Description**

The vertical total (VTOTAL) register defines the upper limit for VCOUNT. VCOUNT counts from 0 to VTOTAL. When the condition VCOUNT = VTOTAL occurs, VCOUNT is reset on the next horizontal line (noninterlaced) or halfline (interlaced).

The condition VCOUNT = VTOTAL also determines when the vertical sync pulse is activated. In **noninterlaced** mode, this always occurs at the end of a line, coincident with the beginning of the horizontal synchronization pulse. In **interlaced** mode, it occurs at either the end or the middle of a line, depending on whether the current field is odd or even.

When you configure the frame timer for **composite** mode, the condition VCOUNT = VTOTAL marks the start of the serration region; in **interlaced composite** mode, it also marks the end of the pre-equalization region (see Appendix A, *Programming Procedures and Examples*).

**Equations**

**Noninterlaced:**  $VTOTAL = (\text{the number of lines per frame}) - 1$

**Interlaced:**  $VTOTAL = (\text{the number of halflines per field}) - 1$



# Video Timing

This chapter describes several video timing features of the VC and how to program them. These features include:

- ☐ Separate or composite sync and blanking
- ☐ Synchronization to internal or external signals
- ☐ Interlaced or noninterlaced video

## Topics

4.1	Understanding Horizontal and Vertical Timing .....	VC: 4-2
4.2	Horizontal Video Timing (Internal) .....	VC: 4-4
4.3	Vertical Video Timing (Internal) .....	VC: 4-6
4.4	Composite Video Timing .....	VC: 4-9
4.5	Noninterlaced Video Timing .....	VC: 4-15
4.6	Interlaced Video Timing .....	VC: 4-22
4.7	Programming VSAREA, VEAREA, and VFTINT in Interlaced Mode .....	VC: 4-33
4.8	External Synchronization .....	VC: 4-44
4.9	Synchronization Delay Compensation .....	VC: 4-50
4.10	High-Impedance Capability .....	VC: 4-52
4.11	Frame Lock .....	VC: 4-53

## 4.1 Understanding Horizontal and Vertical Timing

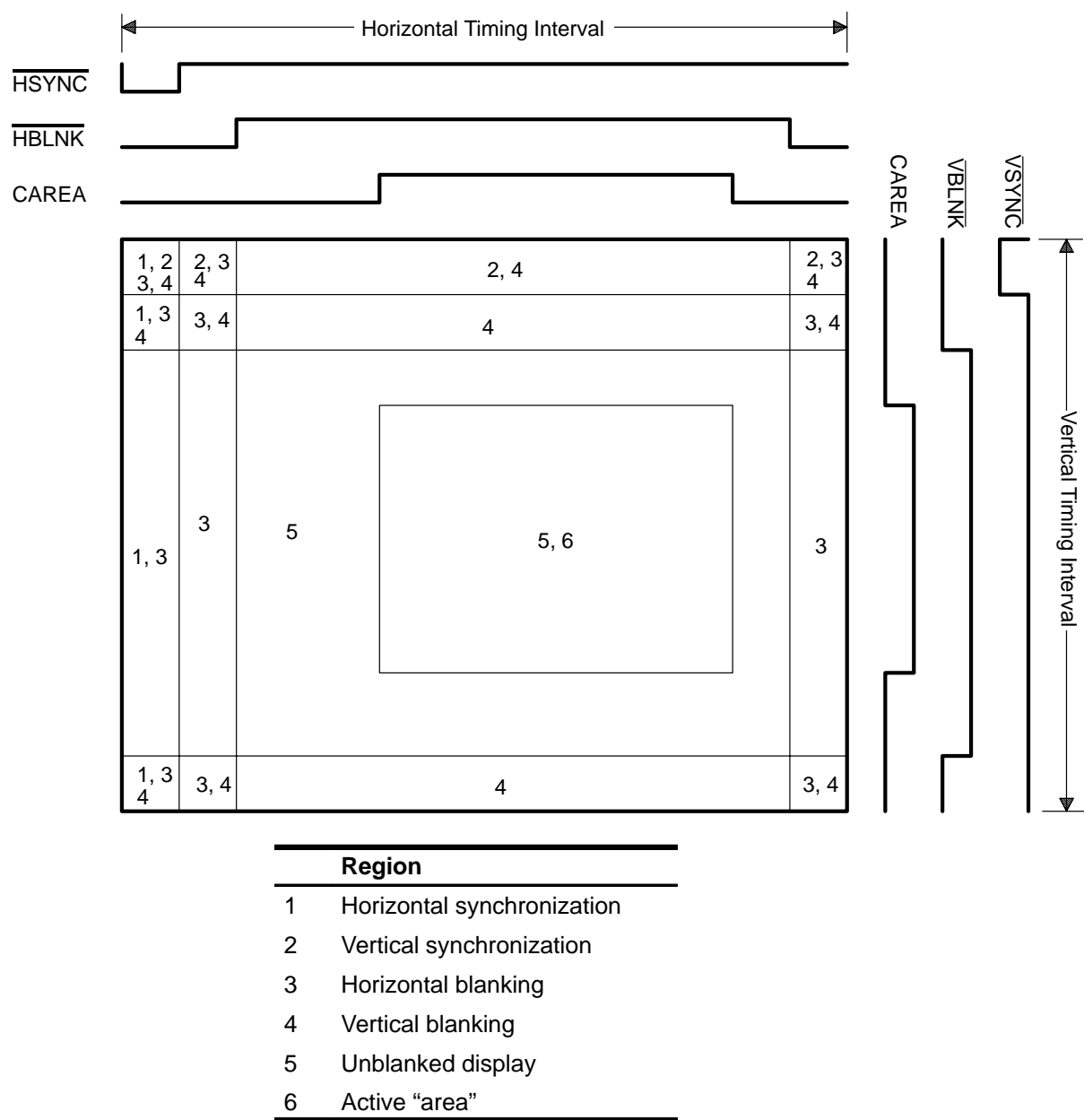
Both horizontal and vertical timing signals are usually required to control a capture or display device. Two signals for each dimension are typical. Some devices may require fewer signals, and others are designed to use composite signals that contain information about both dimensions.

Since CRT-type devices have established standards and signal terminology, CRT terminology is used throughout this chapter. However, the frame timers are not restricted to CRT control and can be adapted for the control of LCDs, CCDs, scanners, printers, or other applications.

The basic CRT control signals are  $\overline{\text{HSYNC}}$ ,  $\overline{\text{VSYNC}}$ ,  $\overline{\text{HBLNK}}$ , and  $\overline{\text{VBLNK}}$ . Figure 4–1 shows the relationship between these signals and the horizontal and vertical dimensions of a frame.

In addition to controlling the sync and blanking signals, the frame timers also provide control over a general-purpose region (area). This region can be used to define an overscan border or secondary window within a display. In other (non-CRT) applications, it can provide a required third timing control signal.

Figure 4–1. Horizontal and Vertical Timing Relationship

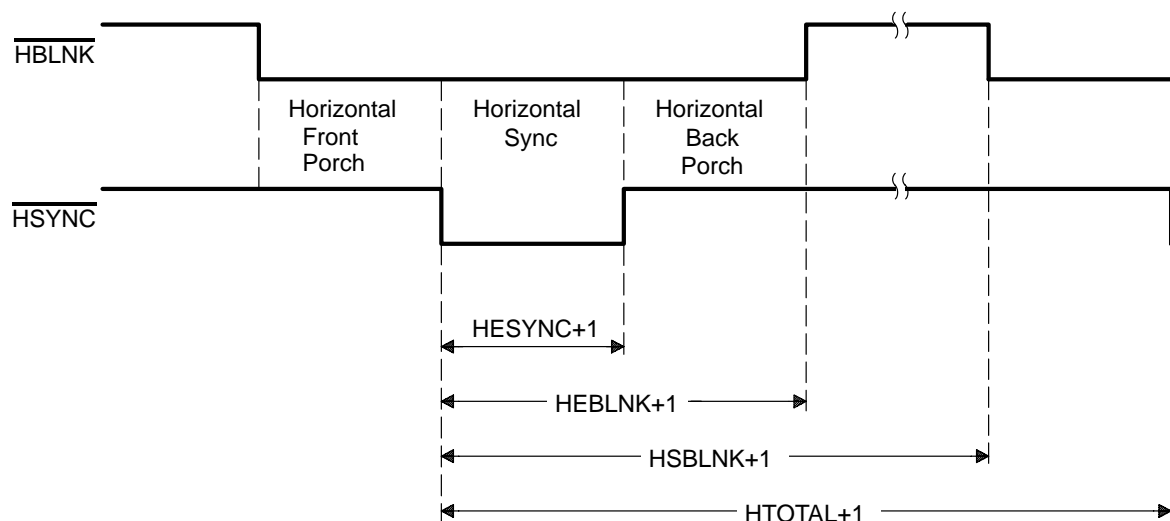


## 4.2 Horizontal Video Timing (Internal)

This section discusses horizontal video timing generated internally by the frame timers. Refer to Section 4.8, *External Synchronization*, for information about external synchronization signals.

The operation of the horizontal timing signals is identical for both interlaced and noninterlaced displays. The length of the horizontal timing periods as defined by the frame timer registers is shown in Figure 4–2.

Figure 4–2. Horizontal Timing Signals



Following are three characteristics of horizontal timing:

- ☐ Horizontal timing periods are made up of an integral number of FCLK periods. FCLK pulses are counted by the HCOUNT register.
- ☐ HCOUNT is incremented once every FCLK period (on the FCLK low-to-high transition) until it equals the value in HTOTAL. During the FCLK period following the occurrence of HCOUNT = HTOTAL, HCOUNT is reset to 0.
- ☐ Because HCOUNT begins counting at 0, the values in the horizontal timing registers represent one less than the number of FCLK periods that occur before the appropriate signal change.

The horizontal sync pulse is defined by the values in HESYNC and HTOTAL. HSYNC is driven low after HCOUNT = HTOTAL and is driven high after HCOUNT = HESYNC.

A 2-1/2-FCLK delay occurs between the time that HCOUNT = HTOTAL or HESYNC and the appropriate transition on the HSYNC pin.

The horizontal blanking pulse is defined by the values in HSBLNK and HEBLNK.  $\overline{\text{HBLNK}}$  is driven low after HCOUNT = HSBLNK and is driven high after HCOUNT = HEBLNK.

A 2-1/2-FCLK delay occurs between the time that HCOUNT = HSBLNK or HEBLNK and the appropriate transition on the  $\overline{\text{HBLNK}}$  pin.



## 4.3 Vertical Video Timing (Internal)

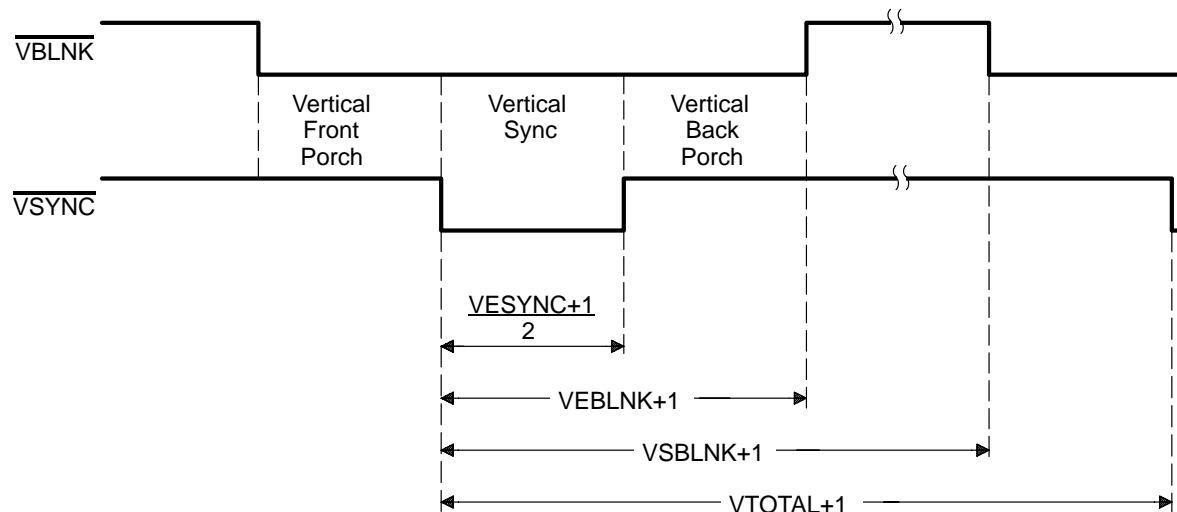
This section describes video timing generated internally by the frame timers. For information about external sync signals, refer to Section 4.8, *External Synchronization*.

### 4.3.1 Noninterlaced Vertical Timing

The operation of the vertical timing signals and the vertical counter is dependent upon whether the frame timer is operating in interlaced or noninterlaced mode.

The length of the noninterlaced vertical timing periods as defined by the frame timer registers is shown in Figure 4–3.

Figure 4–3. Noninterlaced Vertical Timing Signals



Following are three characteristics of the noninterlaced vertical timing signals:

- ☐ Noninterlaced vertical timing periods are made up of an integral number of horizontal lines. Horizontal lines are counted by the  $\text{VCOUNT}$  register.
- ☐  $\text{VCOUNT}$  is incremented once per horizontal line when the condition  $\text{HCOUNT} = \text{HTOTAL}$  occurs. At the end of the horizontal line on which the condition  $\text{VCOUNT} = \text{VTOTAL}$  occurs,  $\text{VCOUNT}$  is reset to 0.
- ☐ Because  $\text{VCOUNT}$  begins counting at 0, the values in the vertical timing registers represent one less than the number of horizontal lines that occur before the appropriate signal change.

The vertical sync pulse is defined by the values in VESYNC and VTOTAL.  $\overline{\text{VSYNC}}$  is driven low at the end of the line on which  $\text{VCOUNT} = \text{VTOTAL}$  and driven high at the end of the line on which  $\text{VCOUNT} = \text{VESYNC}/2$ .

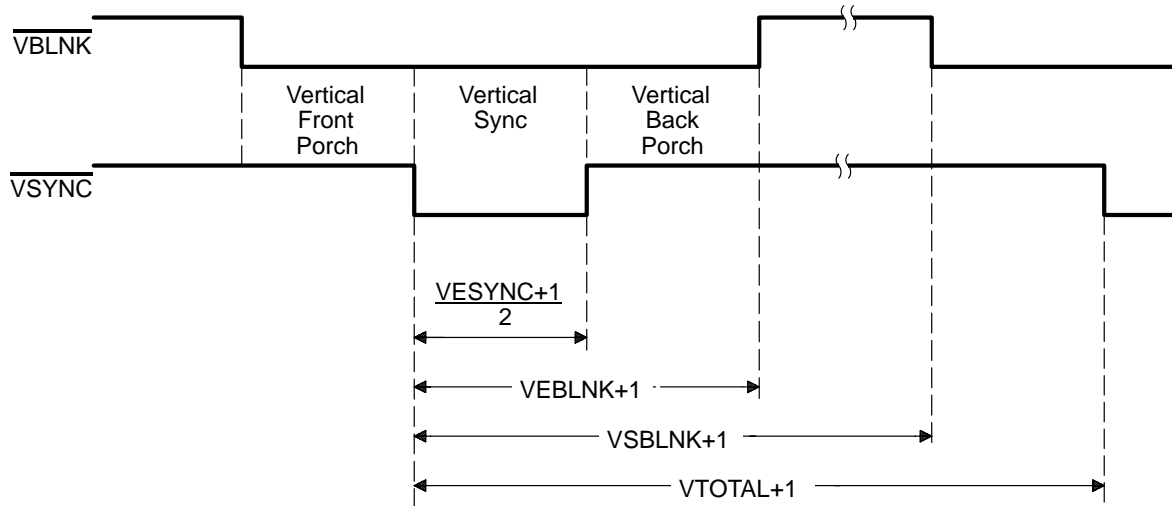
The vertical blanking pulse is defined by the values in VEBLNK and VSBLNK.  $\overline{\text{VBLNK}}$  is driven low at the end of the line on which  $\text{VCOUNT} = \text{VSBLNK}$  and high at the end of the line on which  $\text{VCOUNT} = \text{VEBLNK}$ .

For more information on noninterlaced vertical timing, refer to subsection 4.5.3, *Noninterlaced Vertical Timing Operation*.

### 4.3.2 Interlaced Vertical Timing

The length of the interlaced vertical timing regions as defined by the frame timer registers is shown in Figure 4–4.

Figure 4–4. Interlaced Vertical Timing Signals



Following are three features of interlaced vertical timing signals:

- ☐ Interlaced vertical timing periods are made up of an integral number of horizontal **halflines**. Horizontal halflines are counted by the VCOUNT register.
- ☐ VCOUNT is incremented twice per horizontal line: once when the condition  $\text{HCOUNT} = 1/2 \text{ HTOTAL}$  occurs and again when the condition  $\text{HCOUNT} = \text{HTOTAL}$  occurs. At the end of the horizontal halfline on which the condition  $\text{VCOUNT} = \text{VTOTAL}$  occurs, VCOUNT is reset to 0.
- ☐ Because VCOUNT begins counting at 0, the values in the vertical timing registers represent one less than the number of horizontal halflines that occur before the appropriate signal change.

The vertical sync pulse is defined by the values in VESYNC and VTOTAL.  $\text{VSYNC}$  is driven low at the end of the halfline on which  $\text{VCOUNT} = \text{VTOTAL}$  and driven high at the end of the halfline on which  $\text{VCOUNT} = \text{VESYNC}/2$ .

The vertical blanking pulse is defined by the values in VEBLNK and VSBLNK.  $\overline{\text{VBLNK}}$  is driven low at the end of the halfline on which  $\text{VCOUNT} = \text{VSBLNK}$  and high at the end of the halfline on which  $\text{VCOUNT} = \text{VEBLNK}$ .

For additional descriptions of interlaced vertical timing, refer to Section 4.6.3, *Interlaced Vertical Timing Operation*.

## 4.4 Composite Video Timing

This section describes video timing generated internally by the frame timers. For information about external synchronization signals, refer to Section 4.8, *External Synchronization*.

Composite video signals combine the horizontal and vertical video timing into a single sync signal,  $\overline{\text{CSYNC}}$ , and a single blanking signal,  $\overline{\text{CBLNK}}$ .

- ☐  $\overline{\text{CBLNK}}$  is the logical OR (negative logic) of the  $\overline{\text{HBLNK}}$  and  $\overline{\text{VBLNK}}$  signals. Thus, when either  $\overline{\text{HBLNK}}$  or  $\overline{\text{VBLNK}}$  is active (low),  $\overline{\text{CBLNK}}$  is active (low).
- ☐  $\overline{\text{CSYNC}}$  behaves identically to the  $\overline{\text{HSYNC}}$  signal, except within the vertical blanking region.

During portions of the vertical blanking interval, special sync pulses (**equalization** and **serration** pulses) are generated. These pulses both allow a monitor to detect the vertical-sync interval and ensure that the monitor remains in horizontal sync.

Some video display and input devices are designed to use composite sync and blanking signals rather than separate sync and blanking signals. Video systems used in television broadcasting use composite video signals.

### 4.4.1 The Theory Behind Serration and Equalization Pulses

Video displays that use a composite-sync signal pass the signal through integrating (low-pass) filters to extract the vertical-sync information. Serration pulses that occur during the vertical-sync interval are used by the monitor to determine when the vertical retrace will occur.

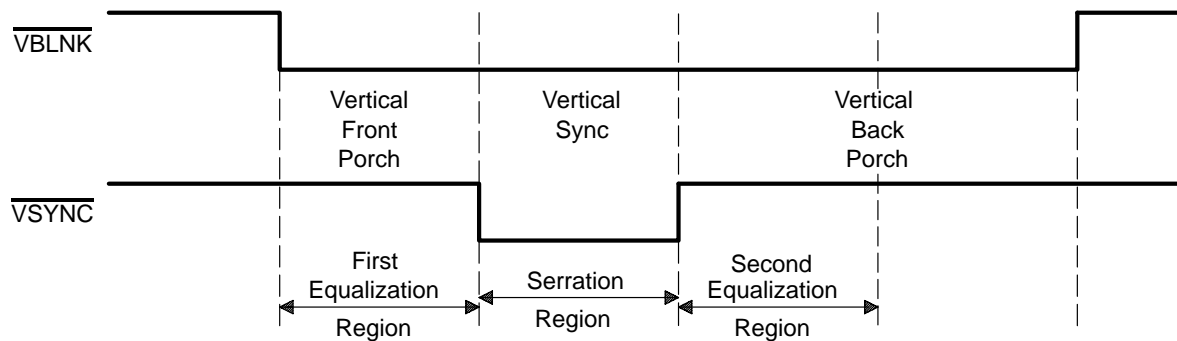
In noninterlaced video, serration pulses begin at the end of each scan line at the same time as horizontal-sync pulses. In interlaced video mode, they also begin at the midpoint of each horizontal line. This is required so that the vertical-sync pulse extracted by the monitor will begin in the even field when the vertical-sync interval begins and will end midway through a horizontal line.

In interlaced video, the equalization pulses occur in the regions immediately before and after the vertical-sync interval. They occur at the midpoint and endpoint of each horizontal line within the equalization region. The equalization pulses have a pulse width equal to half the width of a normal horizontal-sync pulse. They ensure that the monitor extracts a vertical-sync interval of exactly the same width in both even and odd fields.

In noninterlaced video, the vertical sync pulse always begins and ends at the end of a horizontal line, so equalization pulses are not required.

Figure 4–5 shows the regions of vertical blanking in which the equalization and serration pulses occur. Each region is made up of an integral number of horizontal halflines specified by the vertical timing registers. Outside of these regions,  $\overline{CSYNC}$  behavior is identical to that of  $\overline{HSYNC}$ .

Figure 4–5. Vertical Blanking Regions of Equalization and Serration



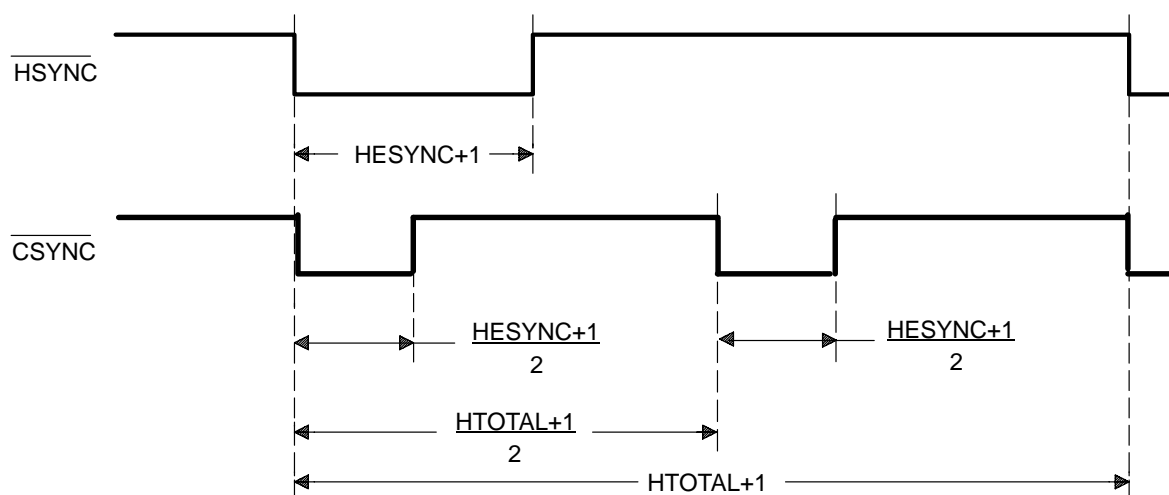
Following are descriptions of the equalization and serration regions in vertical blanking:

- ☐ The **first equalization region** corresponds to the vertical front porch region. In broadcast-quality video standards, such as NTSC and PAL, this region has the same duration as the vertical-sync pulse. The frame timers, however, allow the duration of this region to be changed by adjusting the duration of the vertical front porch (using VSBLNK). See page VC:4-28 for a description of these video standards.
- ☐ The **serration region** coincides with the vertical-sync region and immediately follows the first equalization region.
- ☐ The **second equalization region** immediately follows the serration region. The frame timers define the duration of this region as that of vertical-sync; this definition is in accordance with current video standards.

## 4.4.2 Equalization Pulses on $\overline{\text{CSYNC}}$

If the frame timer is in interlaced mode, equalization pulses are generated on  $\overline{\text{CSYNC}}$  during the equalization regions. These pulses begin at the midpoint ( $\text{HCOUNT} = \text{HTOTAL}/2$ ) and the endpoint ( $\text{HCOUNT} = \text{HTOTAL}$ ) of each horizontal line. The pulses end at  $\text{HCOUNT} = (\text{HTOTAL} + \text{HESYNC})/2$  and at  $\text{HCOUNT} = \text{HESYNC}/2$ , respectively, as shown in Figure 4–6.

Figure 4–6.  $\overline{\text{CSYNC}}$  During Equalization Regions (Interlaced)



Broadcast-quality composite-video standards, such as NTSC and PAL, require that equalization pulses are exactly half the duration of horizontal-sync pulses. Since HCOUNT counts from 0, you should set HESYNC to an odd value so that horizontal sync is an even number of FCLK cycles in length and, thus, evenly divisible by 2. HTOTAL should also be programmed to an odd value so that  $\text{HTOTAL}/2$  occurs exactly midway through each horizontal line.

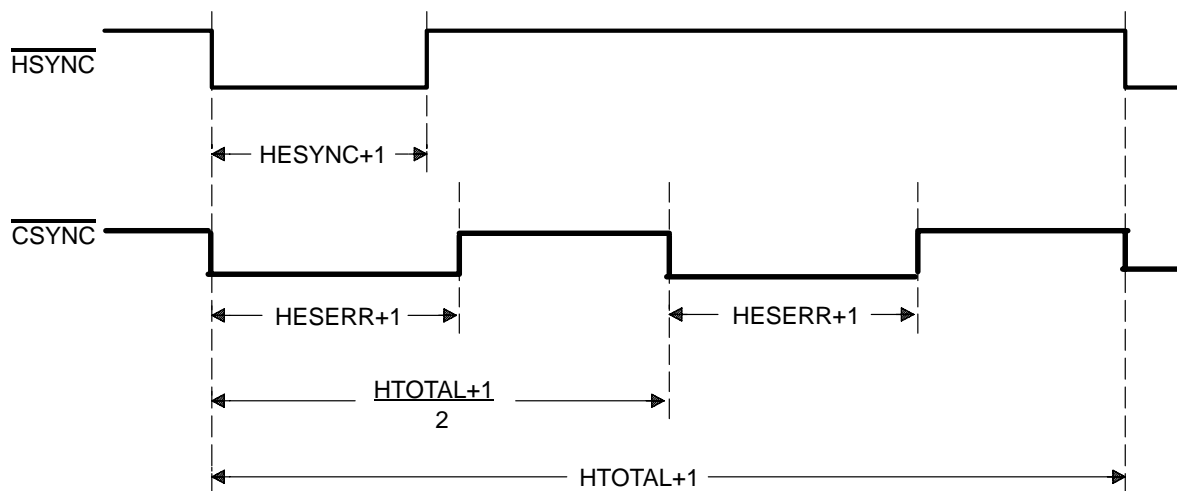
In noninterlaced mode, equalization pulses do not occur.  $\overline{\text{CSYNC}}$  behaves identically to  $\overline{\text{HSYNC}}$  within the regions preceding and following vertical sync.

### 4.4.3 Serration Pulses on $\overline{\text{CSYNC}}$

Serration pulses occur during the serration region coincident with vertical sync. The frequency of the serration pulses in interlaced and noninterlaced modes is different.

When the frame timer is configured for interlaced operation, serration pulses begin at the midpoint ( $\text{HCOUNT} = \text{HTOTAL}/2$ ) and the endpoint ( $\text{HCOUNT} = \text{HTOTAL}$ ) of each horizontal line. These pulses end at  $\text{HCOUNT} = (\text{HTOTAL}/2) + \text{HESERR}$  and at  $\text{HCOUNT} = \text{HESERR}$ , respectively.

Figure 4–7.  $\overline{\text{CSYNC}}$  During Serration Region (Interlaced)

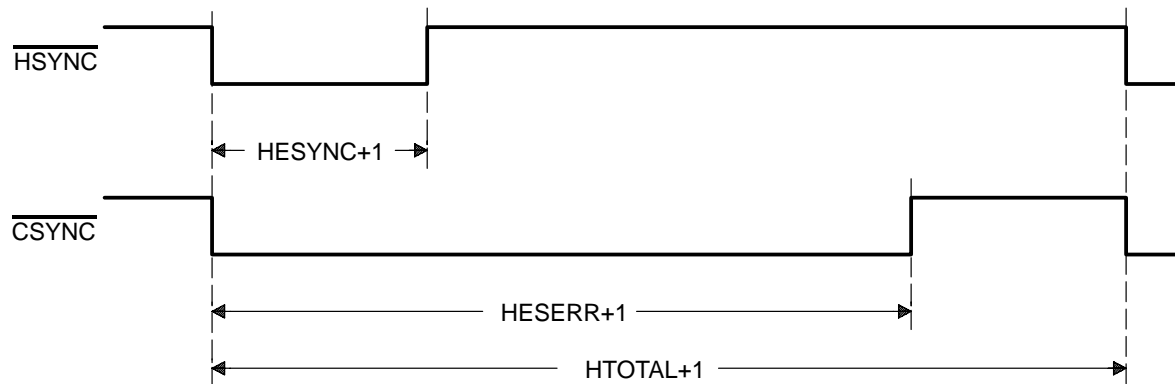


As Figure 4–7 shows, serration pulses are usually longer in duration than horizontal-sync pulses. Broadcast-quality standards, such as NTSC and PAL, require that serration pulses be of a duration such that the inactive (high) period of  $\overline{\text{CSYNC}}$  is equal to the active (low) period of  $\overline{\text{HSYNC}}$ . You can achieve this by programming  $\text{HESERR} = ((\text{HTOTAL} - 1)/2) - (\text{HESYNC} + 1)$ .



During noninterlaced operation, serration pulses are generated only once per horizontal line. Serration pulses begin when  $HCOUNT = HTOTAL$  and end when  $HCOUNT = HESERR$ . This is shown in Figure 4–8.

Figure 4–8.  $\overline{CSYNC}$  During Serration Region (Noninterlaced)



## 4.5 Noninterlaced Video Timing

You can select noninterlaced video mode by setting the interlaced frame disable (IFD) bit in the frame timer control (FTCTL) register to 1. See page VC:3-4 for more information about the FTCTL register.

### 4.5.1 Understanding Noninterlaced Mode

In noninterlaced mode, each frame consists of a single vertical field. Figure 4–9 illustrates the path for the electron beam on the screen of a noninterlaced monitor.

Figure 4–9. Scan Pattern for Noninterlaced Monitors

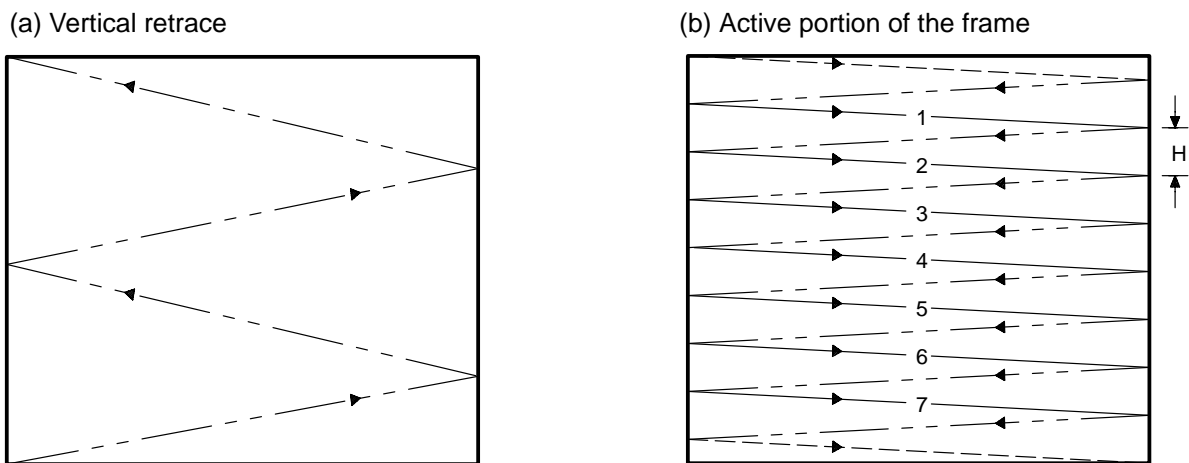


Figure 4–9 (a) shows the vertical retrace that consists of an integral number of horizontal lines.

Figure 4–9 (b) shows the scan portion of the frame.

- ☐ Solid lines represent displayed lines
- ☐ Dashed lines represent blanked lines
- ☐ Partially dashed lines show horizontal retrace
- ☐ H represents the horizontal sweep time where:

$$H = (HTOTAL + 1) \times (FCLK \text{ period})$$

Figure 4–10 and Figure 4–11 illustrate the relationship of the video timing signals for noninterlaced operation. For completeness, all outputs are shown. However, only the following combinations (based on the CPM bits in the FTCTL register) are allowed:

Combination 1 (CPM = 10)	Combination 2 (CPM = 11)
$\overline{\text{HSYNC}}$	$\overline{\text{HSYNC}}$
$\overline{\text{VSYNC}}$	$\overline{\text{VSYNC}}$
$\overline{\text{CSYNC}}$	$\overline{\text{HBLNK}}$
$\overline{\text{CBLNK}}$	$\overline{\text{VBLNK}}$
CAREA	CAREA

## 4.5.2 Noninterlaced Horizontal Timing Operation

Figure 4–10 illustrates horizontal timing operation for noninterlaced mode. The following register values are used in the example:

HESYNC = 3	HTOTAL = n
HEBLNK = eb	HSBLNK = sb
HEAREA = ea	HSAREA = sa
HESERR = er	

As shown in Figure 4–10, there is a delay of 2-1/2 FCLK periods between the point where HCOUNT is equal to a horizontal timing register and the transition of the corresponding signal on the external pin. Since this skew is consistent, it has no net effect on the video timing externally.

HCOUNT begins counting at value 0 and increments on each rising FCLK edge. When HCOUNT reaches HESYNC (3 in this example), HSYNC and CSYNC are driven high. During vertical sync (serration region), CSYNC is **not** driven high until HCOUNT reaches HESERR.

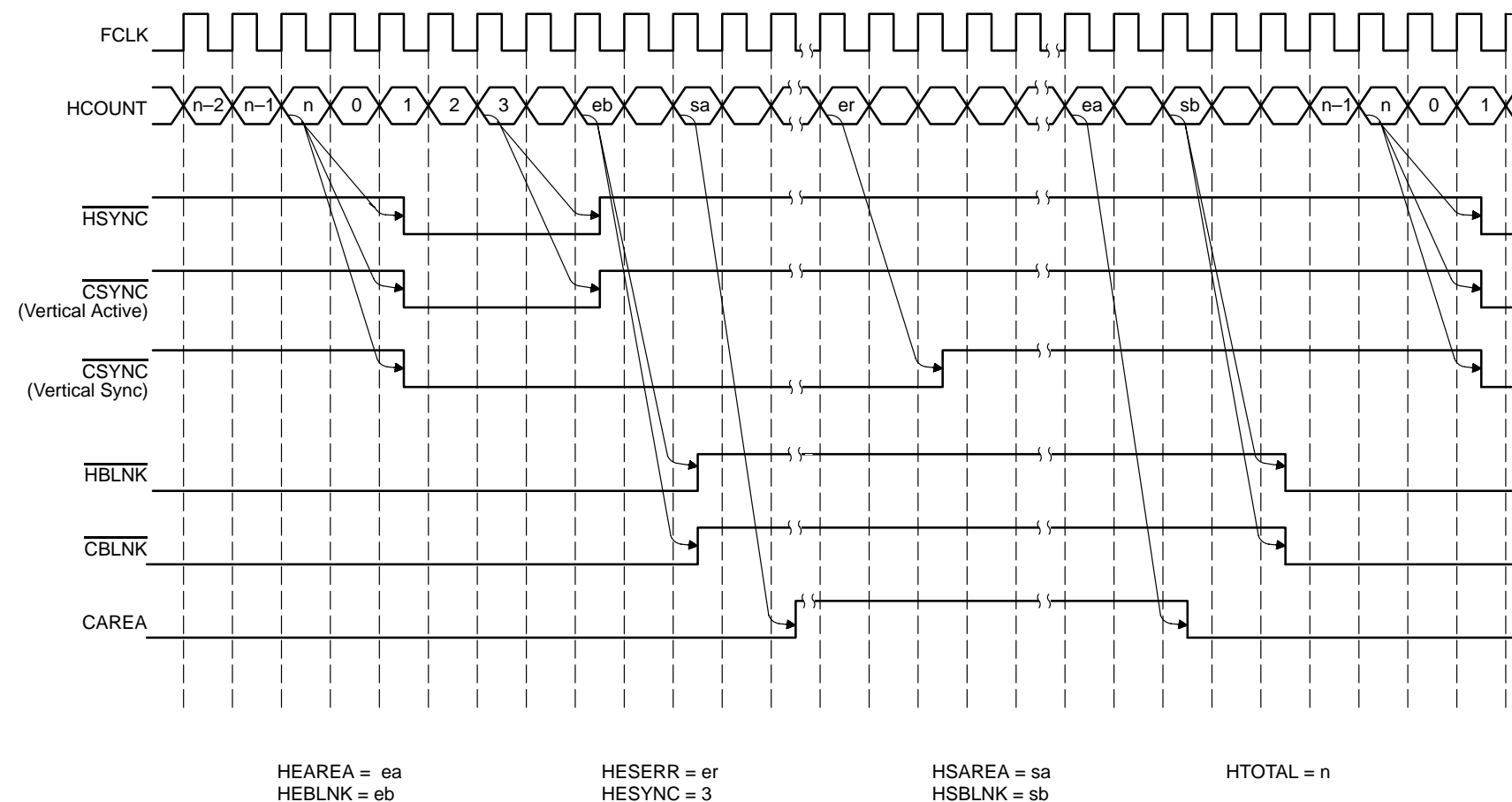
When HCOUNT reaches HEBLNK, the HBLNK and CBLNK outputs are driven high. These signals are driven low at the start of the horizontal blanking interval (HCOUNT = HSBLNK). CAREA is driven high when HCOUNT reaches HSAREA and low when HCOUNT reaches HEAREA.

When HCOUNT reaches HTOTAL, the horizontal-sync period begins. HSYNC and CSYNC are driven low, and HCOUNT is reset to 0 to begin counting for the next horizontal line.

### Note:

Figure 4–10 shows how you would typically program signal transitions. There are no restrictions concerning the relationships between the horizontal signals in noninterlaced mode. Horizontal sync is not required to be nested within horizontal blanking or within the horizontal area.

Figure 4–10. Noninterlaced Horizontal Timing Summary



### 4.5.3 Noninterlaced Vertical Timing Operation

Figure 4–11 illustrates an example of a vertical timing operation for noninterlaced operation. The following register values are used in the example:

VESYNC = 5	VTOTAL = n
VEBLNK = 3	VSBLNK = n – 1
VEAREA = n – 1	VSAREA = 3

VCOUNT begins counting at value 0 and increments at the end of each horizontal line (when HCOUNT is reset to 0). When VCOUNT reaches  $(\text{VESYNC} - 1) / 2$  (2 in this example),  $\overline{\text{VSYNC}}$  is driven high at the end of the next horizontal line. This also marks the end of the serration region, causing  $\overline{\text{CSYNC}}$  to stop outputting serration pulses and begin outputting normal horizontal sync pulses.

When VCOUNT reaches VEBLNK, the  $\overline{\text{VBLNK}}$  output is driven high and  $\overline{\text{CBLNK}}$  begins behaving identically to  $\overline{\text{HBLNK}}$ .  $\overline{\text{VBLNK}}$  and  $\overline{\text{CBLNK}}$  are driven low at the start of the vertical blanking interval (VCOUNT = VSBLNK). CAREA behaves as a horizontal area signal when VCOUNT = VSAREA and is driven low when VCOUNT reaches VEAREA.

When VCOUNT reaches VTOTAL, the vertical-sync period begins.  $\overline{\text{VSYNC}}$  is driven low, and VCOUNT is reset to 0 to begin counting for the next frame. Also, the serration region begins, causing serration pulses to be output on  $\overline{\text{CSYNC}}$ .

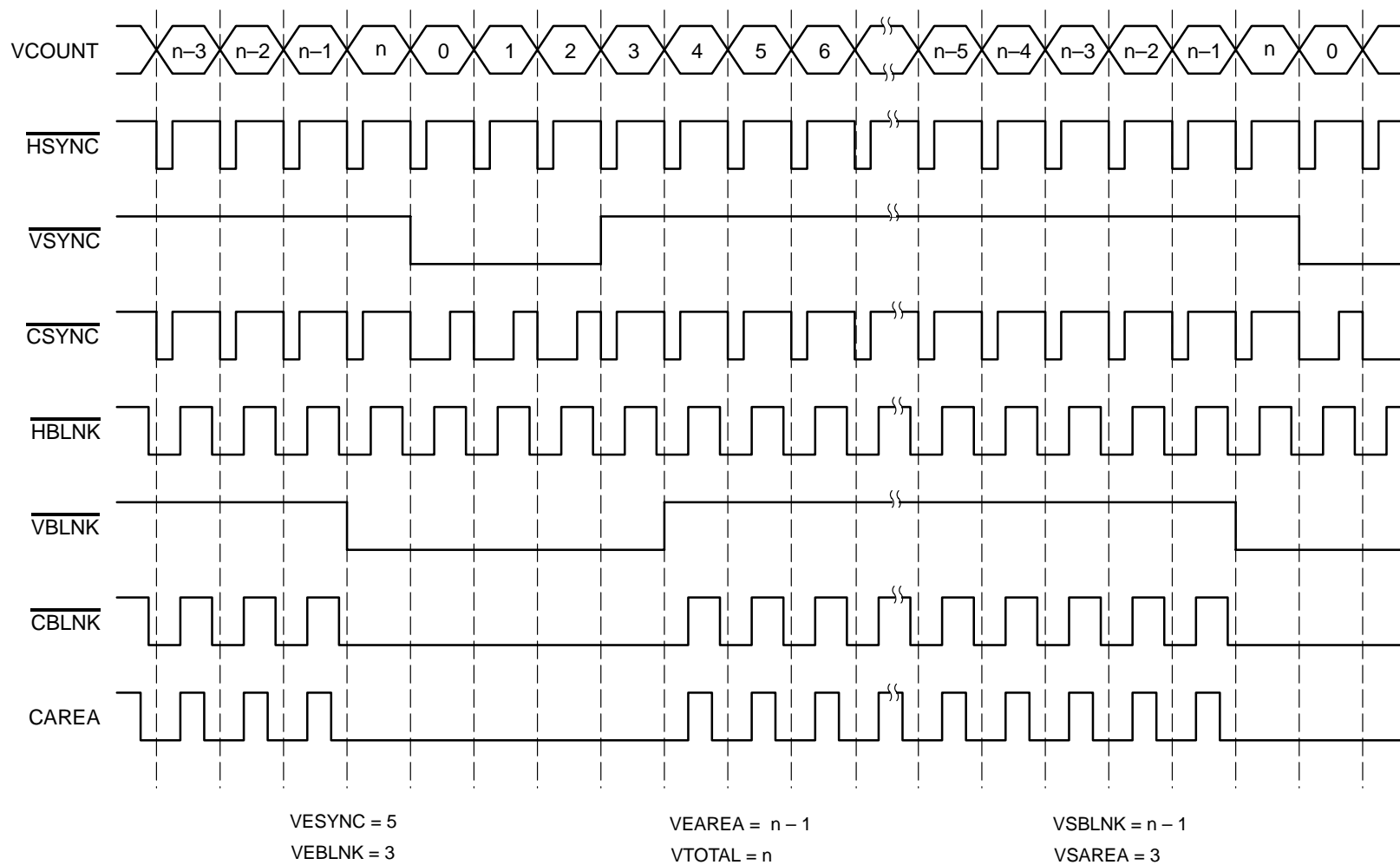
---

**Note:**

Figure 4–11 shows how you would typically program signal transitions. There are no restrictions concerning the relationships between the vertical signals in noninterlaced mode. Vertical sync is not required to be nested within vertical blanking or within the vertical area.

---

Figure 4–11. Noninterlaced Vertical Timing Summary



#### 4.5.4 Summary for Programming Noninterlaced Video Timing

Table 4–1 summarizes all the video timing registers and how to program them for noninterlaced video.

Table 4–1. Programming the Video Timing Registers for Noninterlaced Video

Register	Is programmed to...
HTOTAL	(The number of FCLKs per line) – 1
HESYNC	(The number of FCLKs in horizontal sync) – 1
HESERR <sup>†</sup>	(The number of FCLKs in horizontal serration) – 1
HEBLNK	(The number of FCLKs from the start of horizontal sync to the end of horizontal blank) – 1
HSAREA	(The number of FCLKs from the start of horizontal sync to the start of horizontal area) – 1
HEAREA	(The number of FCLKs from the start of horizontal sync to the end of horizontal area) – 1
HSBLNK	(The number of FCLKs from the start of horizontal sync to the start of horizontal blank) – 1
VTOTAL	(The number of lines per frame) – 1
VESYNC	(Twice the number of lines in vertical sync) – 1
VEBLNK	(The number of lines from the start of vertical sync to the end of vertical blank) – 1
VSAREA	(The number of lines from the start of vertical sync to the start of vertical area) – 1
VEAREA	(The number of lines from the start of vertical sync to the end of vertical area) – 1
VSBLNK	(The number of lines from the start of vertical sync to the start of vertical blank) – 1
VFTINT	(The number of lines from the start of vertical sync to the interrupt point) – 1

<sup>†</sup>HESERR needs to be programmed only for composite mode.



## 4.6 Interlaced Video Timing

You can select interlaced video mode by setting the interlaced frame disable (IFD) bit in the frame timer control (FTCTL) register to 0. See page VC:3-4 for more information about the FTCTL register.

### 4.6.1 Understanding Interlaced Mode

In interlaced mode, each frame consists of two vertical fields. One vertical field displays odd horizontal lines, and the other field displays even horizontal lines. Figure 4–12, Figure 4–13, and Figure 4–14 show the scan path for the electron beam on the screen of an interlaced monitor.

Figure 4–12. Scan Pattern for Interlaced Monitors—Odd Field

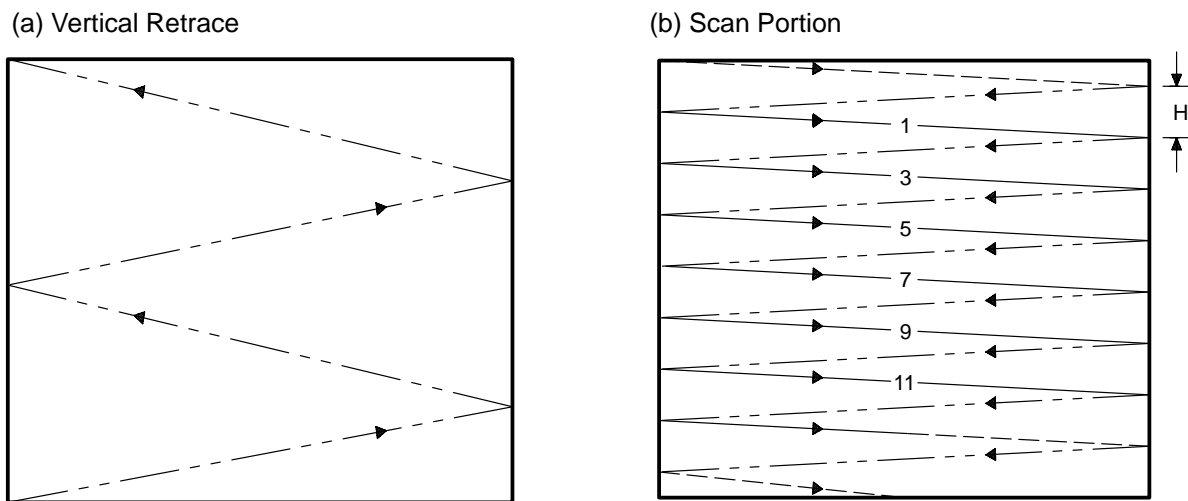


Figure 4–12 (a) shows the vertical retrace that precedes the odd field. It consists of an integral number of horizontal halflines.

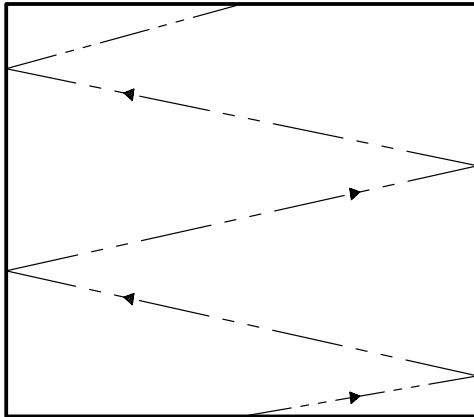
Figure 4–12 (b) shows the scan portion of the odd field.

- ☐ Solid lines represent displayed lines
- ☐ Dashed lines represent blanked lines
- ☐ Partially dashed lines show horizontal retrace
- ☐ H represents the horizontal sweep time where:

$$H = (HTOTAL + 1) \times (FCLK \text{ period})$$

Figure 4–13. Scan Pattern for Interlaced Monitors—Even Field

(a) Vertical Retrace



(b) Scan Portion

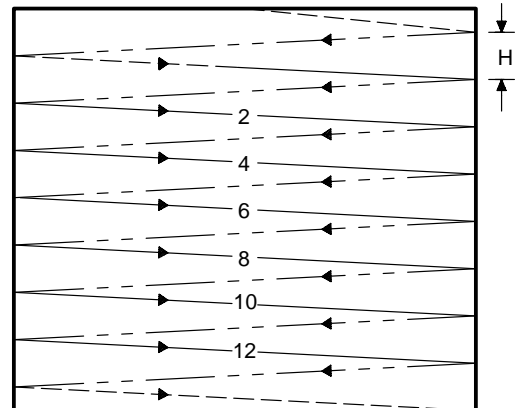


Figure 4–13 (a) shows the vertical retrace that precedes the even field. It consists of an integral number of horizontal halflines.

Figure 4–13 (b) shows scan portion of the even field.

- ☐ Solid lines represent displayed lines
- ☐ Dashed lines represent blanked lines
- ☐ Partially dashed lines show horizontal retrace
- ☐ H represents the horizontal sweep time where:

$$H = ( HTOTAL + 1 ) \times ( FCLK \text{ period} )$$

Figure 4–14. Scan Pattern for Interlaced Monitors—Both Fields

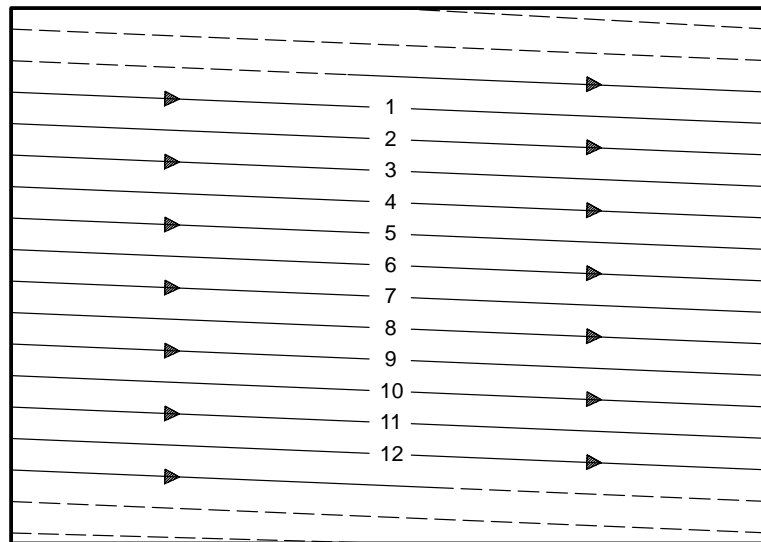


Figure 4–14 shows the juxtaposition of odd and even fields on the display.

- ☐ Solid lines represent displayed lines
- ☐ Dashed lines represent blanked lines

Because the starting position of the even field begins halfway across a horizontal line, the lines of the even field fall between the lines of the odd field.

Figure 4–15, Figure 4–16, and Figure 4–17 illustrate the relationship of the video timing signals for interlaced operation. For completeness, all outputs are shown. However, only the following combinations (based on the CPM bits in FTCTL) are allowed:

Combination 1 (CPM = 10)	Combination 2 (CPM = 11)
HSYNC	HSYNC
VSYNC	VSYNC
CSYNC	HBLNK
CBLNK	VLNK
CAREA	CAREA

## 4.6.2 Interlaced Horizontal Timing Operation

Figure 4–15 illustrates the horizontal timing operation for interlaced mode. The following register values are used in the example:

HESYNC = 3	HTOTAL = n
HEBLNK = eb	HSBLNK = sb
HEAREA = ea	HSAREA = sa
HESERR = er	

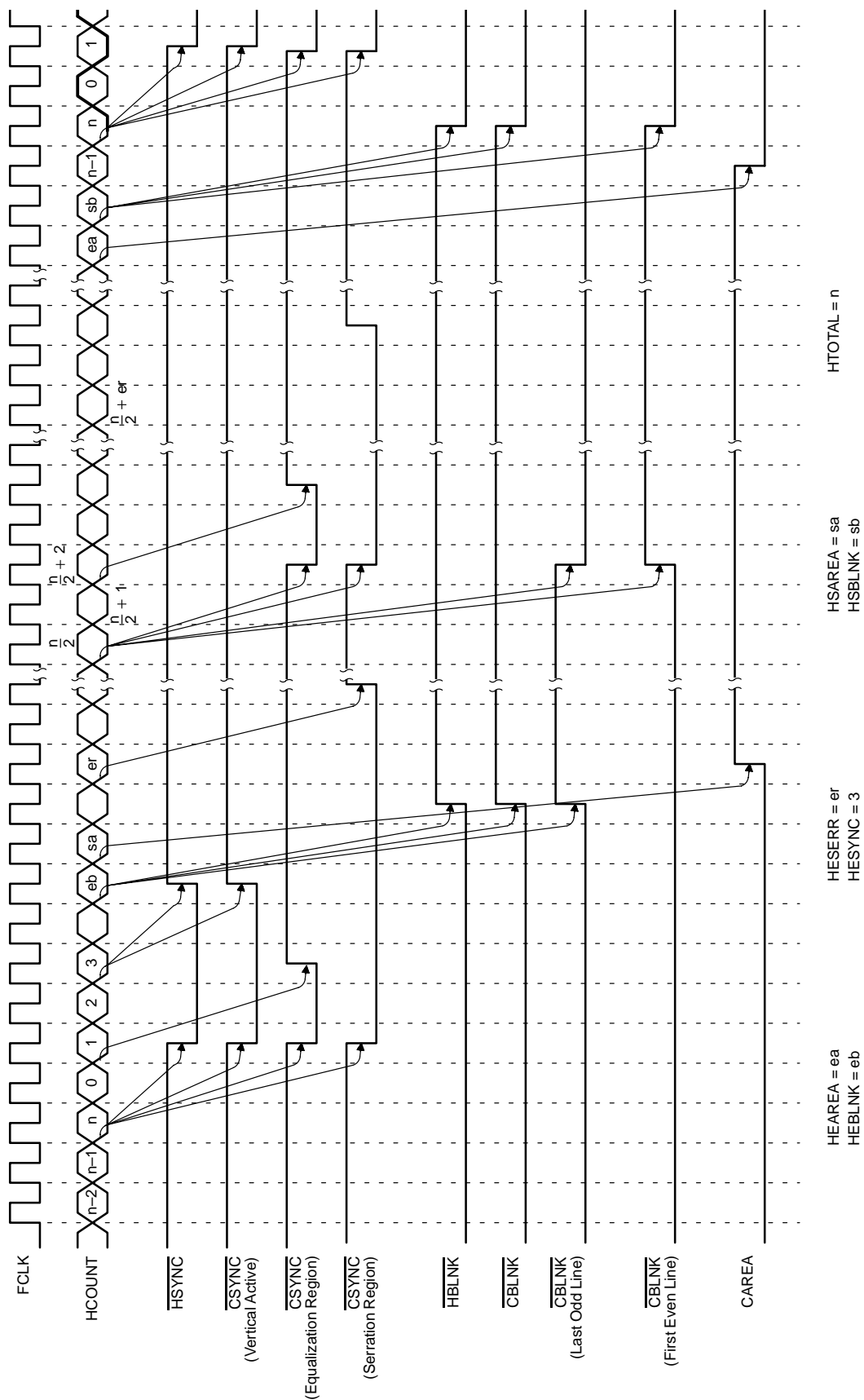
As the figure shows, there is a 2-1/2-FCLK delay between the point where HCOUNT is equal to a horizontal timing register and the transition of the corresponding signal on the external pin. Since this skew is consistent, it has no net effect on the video timing externally.

HCOUNT begins counting at 0 and increments on each rising FCLK edge. When HCOUNT reaches HESYNC (3 in this example),  $\overline{\text{HSYNC}}$  and  $\overline{\text{CSYNC}}$  are driven high. During equalization regions,  $\overline{\text{CSYNC}}$  is driven high when  $\text{HCOUNT} = \text{HESYNC} / 2$ . During vertical sync (the serration region),  $\overline{\text{CSYNC}}$  is not driven high until HCOUNT reaches HESERR.

When HCOUNT reaches HEBLNK, the  $\overline{\text{HBLNK}}$  and  $\overline{\text{CBLNK}}$  outputs are driven high. The signals are driven low at the start of the horizontal blanking interval ( $\text{HCOUNT} = \text{HSBLNK}$ ). CAREA is driven high when HCOUNT reaches HSAREA and low when HCOUNT reaches HEAREA.

Note that on the last line of the odd field,  $\overline{\text{CBLNK}}$  is driven low when  $\text{HCOUNT} = \text{HTOTAL} / 2$ . This is because the vertical component of  $\overline{\text{CBLNK}}$  begins in the middle of the active horizontal line. In the same manner,  $\overline{\text{CBLNK}}$  is driven high when  $\text{HCOUNT} = \text{HTOTAL} / 2$ , because the vertical component of  $\overline{\text{CBLNK}}$  ends in the middle of the active horizontal line.

Figure 4-15. Interlaced Horizontal Timing Summary



During equalization and serration regions,  $\overline{\text{CSYNC}}$  is driven low when  $\text{HCOUNT} = \text{HTOTAL} / 2$ , because equalization and serration pulses occur every halfline.  $\overline{\text{CSYNC}}$  then remains low for  $(\text{HESYNC} + 1) / 2$  FCLK pulses for equalization and for  $\text{HESERR} + 1$  FCLK pulses for serration.

When  $\text{HCOUNT}$  reaches  $\text{HTOTAL}$ , the horizontal-sync period begins.  $\overline{\text{HSYNC}}$  and  $\overline{\text{CSYNC}}$  are driven low, and  $\text{HCOUNT}$  is reset to 0 to begin counting for the next horizontal line.

---

**Note:**

In **interlaced mode**, the relationship between the horizontal timing signals must be similar to that illustrated in Figure 4–10. The horizontal sync pulse must lie entirely within horizontal blanking. CAREA must be inactive throughout horizontal sync. The video timing registers must be programmed so that the video outputs obey these constraints.

---

### 4.6.3 Interlaced Vertical Timing Operation

Using interlaced composite mode, the frame timers can be programmed to support either the NTSC RS-170 American television standard or the PAL European television standard. The fundamental difference between these two standards is that RS-170 has a vertical sync pulse (vertical retrace) composed of an integral number of horizontal scan lines while the PAL vertical sync pulse does not. (The PAL vertical sync pulse has an extra half-line).

Figure 4–16 and Figure 4–17 both illustrate the vertical timing operation for interlaced mode. The following description applies to both diagrams.

Figure 4–16 illustrates the video timing signals for an NTSC-type display. The following register values are used in the example:

VESYNC = 11	VTOTAL = n
VEBLNK = e	VSBLNK = s
VEAREA = s	VSAREA = e

Figure 4–17 illustrates the video timing signals for a PAL-type display. The following registers values are used in the example:

VESYNC = 9	VTOTAL = n
VEBLNK = e	VSBLNK = s
VEAREA = s	VSAREA = e

Notice that in both examples,  $VSAREA = VEBLNK$  and  $VEAREA = VSBLNK$ . This was done to highlight the difference in operation between the vertical component of  $\overline{CBLNK}$ , which can change in the middle of a horizontal line, and the vertical component of  $CAREA$ , which can change only at the end of a horizontal line.

VCOUNT begins counting at value 0 and increments at both the center and the end of each horizontal line. When VCOUNT reaches  $(VESYNC - 1)/2$ ,  $\overline{VSYNC}$  is driven high at the end of the next horizontal line.

This also marks the end of the serration region, where  $\overline{CSYNC}$  stops outputting serration pulses and begins outputting equalization pulses. These pulses continue until  $VCOUNT = VESYNC$ , where the postequalization region ends and  $\overline{CSYNC}$  resumes normal horizontal sync behavior. Note that by definition, the serration and postequalization regions must be an equal number of pulses in duration.

When VCOUNT reaches VEBLNK, the  $\overline{\text{VBLNK}}$  output is driven high and  $\overline{\text{CBLNK}}$  begins behaving identically to  $\overline{\text{HBLNK}}$ . This occurs in the center of a horizontal line in the even field and at the end of a horizontal line in the odd field. The signals are driven low at the start of the vertical blanking interval (VCOUNT = VSBLNK). (This occurs in the center of a horizontal line in the even field and at the end of a horizontal line in the odd field.) The start of vertical blanking also marks the beginning of the pre-equalization region and the point at which equalization pulses begin on  $\overline{\text{CSYNC}}$ .

In order to create a true rectangular region, the vertical component of CAREA (unlike blanking) must not change in the middle of a horizontal line. Thus, CAREA behaves as a horizontal area signal at the end of the line on which VCOUNT = VSAREA, and it is driven low at the end of the line on which VCOUNT = VEAREA. This is explained in detail in Section 4.7, *Programming VSAREA, VEAREA, and VFTINT in Interlaced Mode*.

When VCOUNT reaches VTOTAL, the vertical-sync period begins.  $\overline{\text{VSYNC}}$  is driven low (active), and VCOUNT is reset to 0 to begin counting for the next frame. Also, the serration region begins causing serration pulses to be output on  $\overline{\text{CSYNC}}$ .

---

**Note:**

In interlaced mode, the frame timers are designed to produce signals that conform to interlaced timing standards. Therefore, vertical sync must lie within vertical blanking, and CAREA must be inactive through the equalization and serration region ( $\text{VSAREA} \geq \text{VESYNC}$  and  $\text{VEAREA} \leq \text{VSBLNK}$ ).

---



Figure 4-16. NTSC-Type Interlaced Frame Timing Example

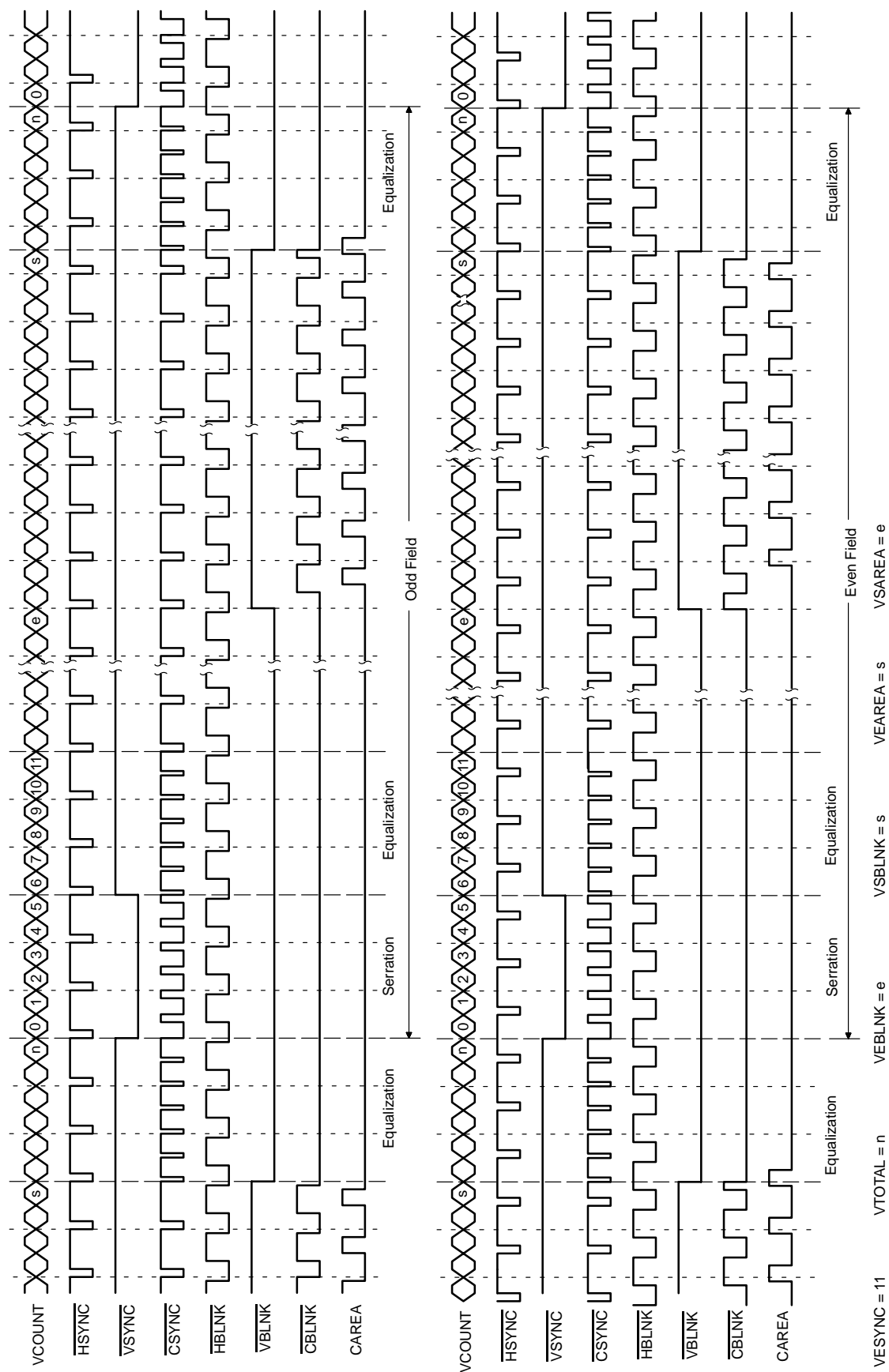
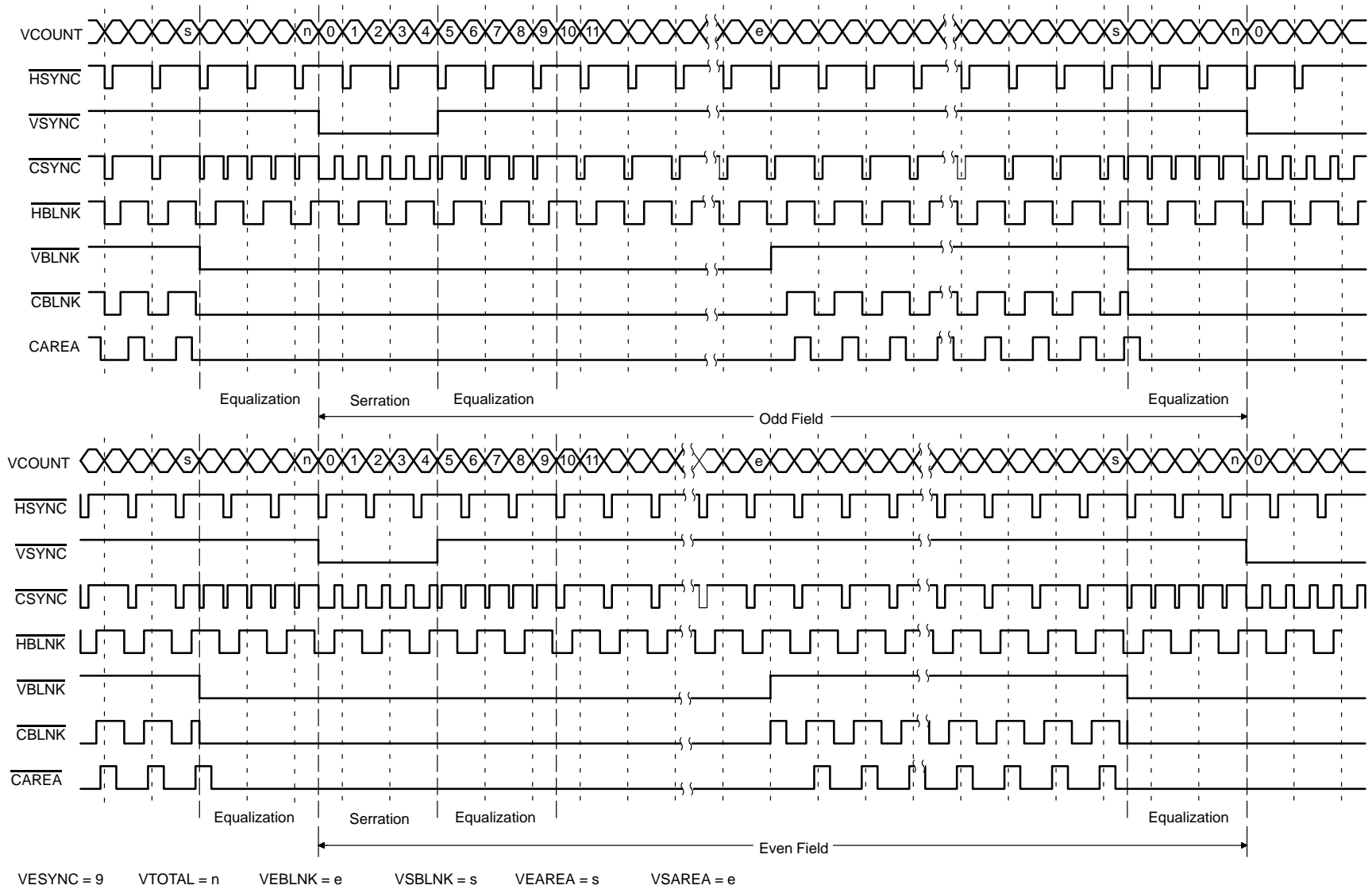


Figure 4–17. PAL-Type Interlaced Frame Timing Example



#### 4.6.4 Summary for Programming Interlaced Video Timing

Table 4–2 summarizes the programming of the video timing registers for interlaced displays.

Table 4–2. Programming the Video Timing Registers for Interlaced Video

Register	Is programmed to...
HTOTAL	(The number of FCLKs per line) – 1
HESYNC	(The number of FCLKs in horizontal sync) – 1
HESERR	(The number of FCLKs in horizontal serration) – 1
HEBLNK	(The number of FCLKs from the start of horizontal sync to the end of horizontal blank) – 1
HSAREA	(The number of FCLKs from the start of horizontal sync to the start of horizontal area) – 1
HEAREA	(The number of FCLKs from the start of horizontal sync to the end of horizontal area) – 1
HSBLNK	(The number of FCLKs from the start of horizontal sync to the start of horizontal blank) – 1
VTOTAL	(The number of halflines per field) – 1
VESYNC	(Twice the number of halflines in vertical sync) – 1
VEBLNK	(The number of halflines from the start of vertical sync to the end of vertical blank) – 1
VEAREA	(The number of halflines from the start of vertical sync to the end of vertical area) – 1, in whichever field vertical area goes active first
VSAREA	(The number of halflines from the start of vertical sync to the start of vertical area) – 1, in whichever field vertical area goes active first
VSBLNK	(The number of halflines from the start of vertical sync to the start of vertical blank) – 1
VFTINT	(The number of halflines from the start of vertical sync to the interrupt point) – 1

## 4.7 Programming VSAREA, VEAREA, and VFTINT in Interlaced Mode

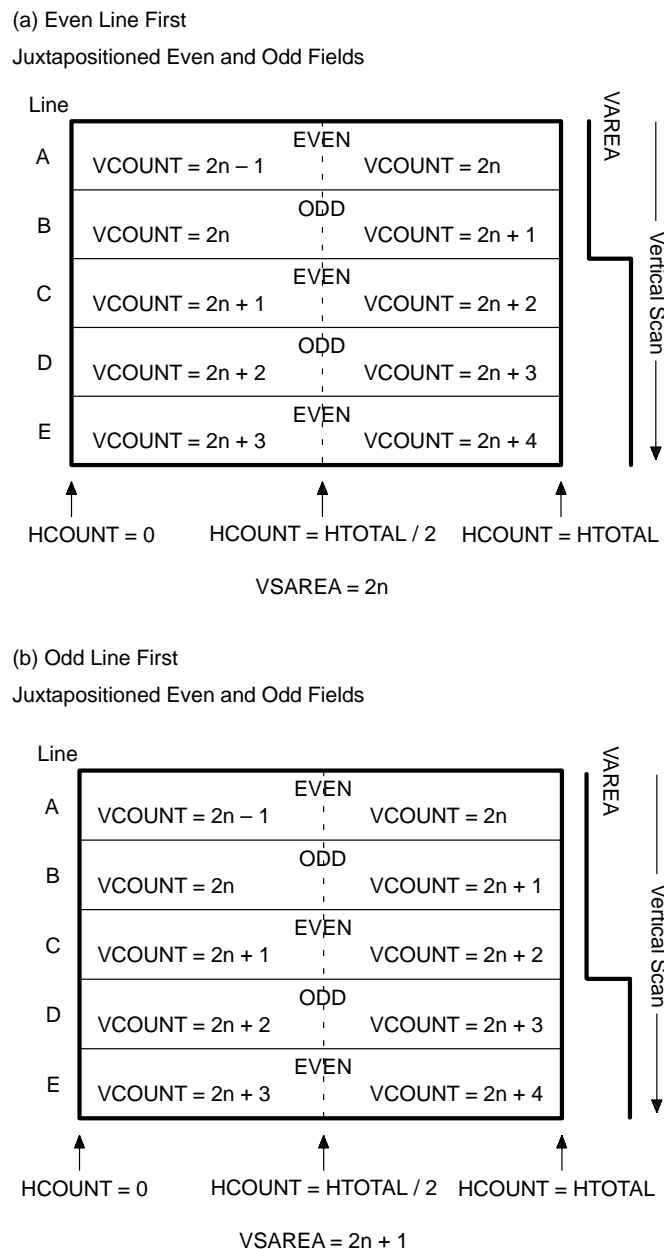
The VCOUNT register increments twice per line in interlaced mode to provide the necessary halfline resolution for the vertical timing registers. The beginning and end of the (vertical) area region and the frame timer interrupt, however, can occur only at the end of a horizontal line. Because vertical sync starts in the middle of a line in one field, the number of halflines that occur before the event are different for the two fields. Which number to use depends on whether the event occurs on an even line (a line in the even field) or an odd line (a line in the odd field).

### 4.7.1 Vertical Start Area Programming

Figure 4–18 shows examples of how to program the VSAREA register for an NTSC interlaced frame.

- ☐ **Even Line First:** Figure 4–18 (a) shows how to program VSAREA so that VAREA (the vertical component of CAREA) begins on an **even** line (line C). For this to take place, the condition  $\text{VCOUNT} = \text{VSAREA}$  must occur on the preceding lines in both fields (lines A and B). Thus, VSAREA should be set to  $2n$ , which is the value of VCOUNT at the end of the last **even** line before VAREA is active.
- ☐ **Odd Line First:** Figure 4–18 (b) shows how to program VSAREA so that VAREA begins on an **odd** line (line D). For this to take place, the condition  $\text{VCOUNT} = \text{VSAREA}$  must occur on the preceding lines in both fields (lines B and C). Thus, VSAREA should be set to  $2n + 1$ , which is the value of VCOUNT at the end of the last **odd** line before VAREA is active.

Figure 4–18. Programming VSAREA for NTSC

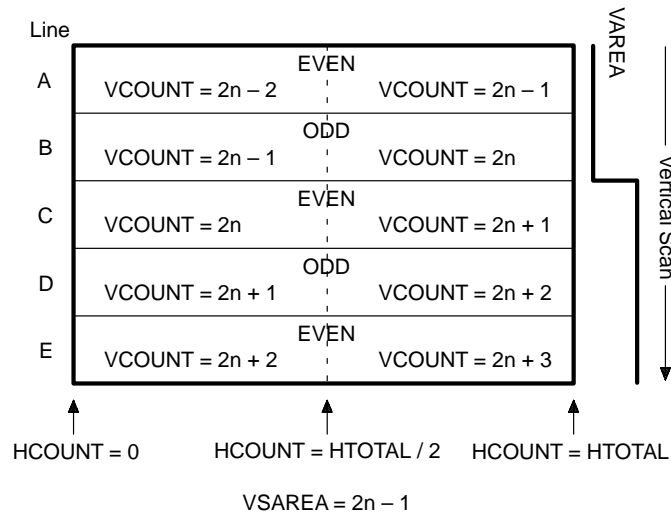


In general, VSAREA must be set to whichever value of VCOUNT is common to the last even and odd lines before VAREA is required to be active. Note that for NTSC, VCOUNT is always even at the end of even lines and odd at the end of odd lines, as shown in the example in Figure 4–18. For PAL, the opposite is true, as shown in the example in Figure 4–19. Because the PAL standard contains an odd number of halflines in its serration region, VCOUNT is always **odd** at the end of even lines and **even** at the end of odd lines.

Figure 4–19. Programming VSAREA for PAL

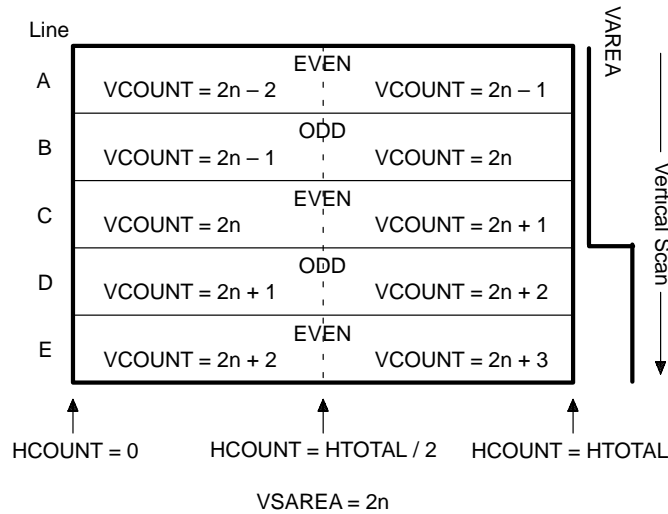
(a) Even Line First

Juxtapositioned Even and Odd Fields



(b) Odd Line First

Juxtapositioned Even and Odd Fields



An alternative frame-independent mechanism for programming VSAREA is described in subsection 4.7.4, *Alternative Programming for VFTINT, VSAREA, and VEAREA*.

## 4.7.2 Vertical End Area Programming

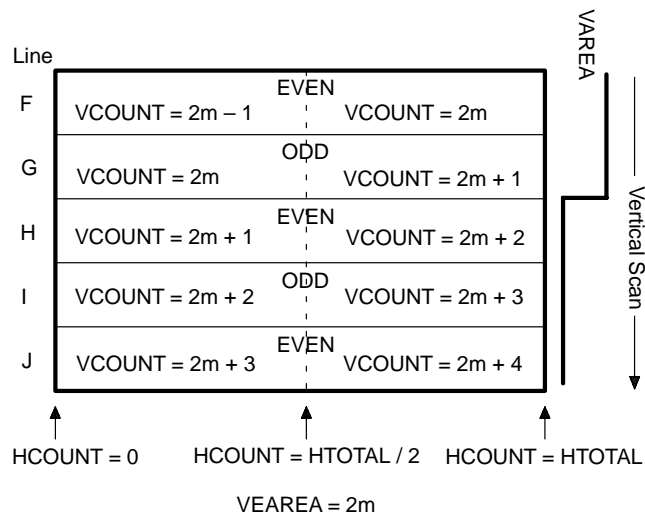
Figure 4–20 shows examples of how to program the VEAREA register for an NTSC interlaced frame.

- ❑ **Odd Line Last:** Figure 4–20 (a) shows how to program VEAREA so that VAREA (the vertical component of CAREA) ends on an **odd** line (line G). For this to take place, the condition  $VCOUNT = VEAREA$  must occur on the preceding lines (lines F and G) in both fields. Thus, VEAREA should be set to  $2m$ , which is the value of VCOUNT at the end of the last **even** line before VAREA is inactive.
- ❑ **Even Line Last:** Figure 4–20 (b) shows how to program VEAREA so that VAREA ends on an **even** line (line H). For this to take place, the condition  $VCOUNT = VEAREA$  must occur on the preceding lines (lines G and H) in both fields. Thus, VEAREA should be set to  $2m + 1$ , which is the value of VCOUNT at the end of the last **odd** line before VAREA is inactive.

Figure 4–20. Programming VEAREA for NTSC

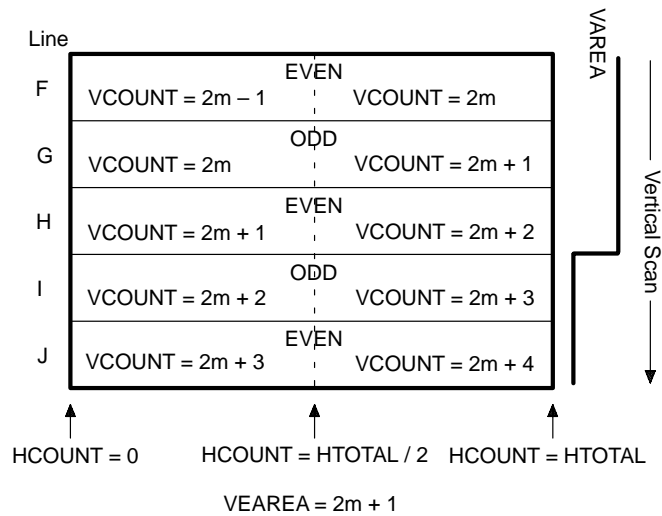
(a) Odd Line Last

Juxtapositioned Even and Odd Fields



(b) Even Line Last

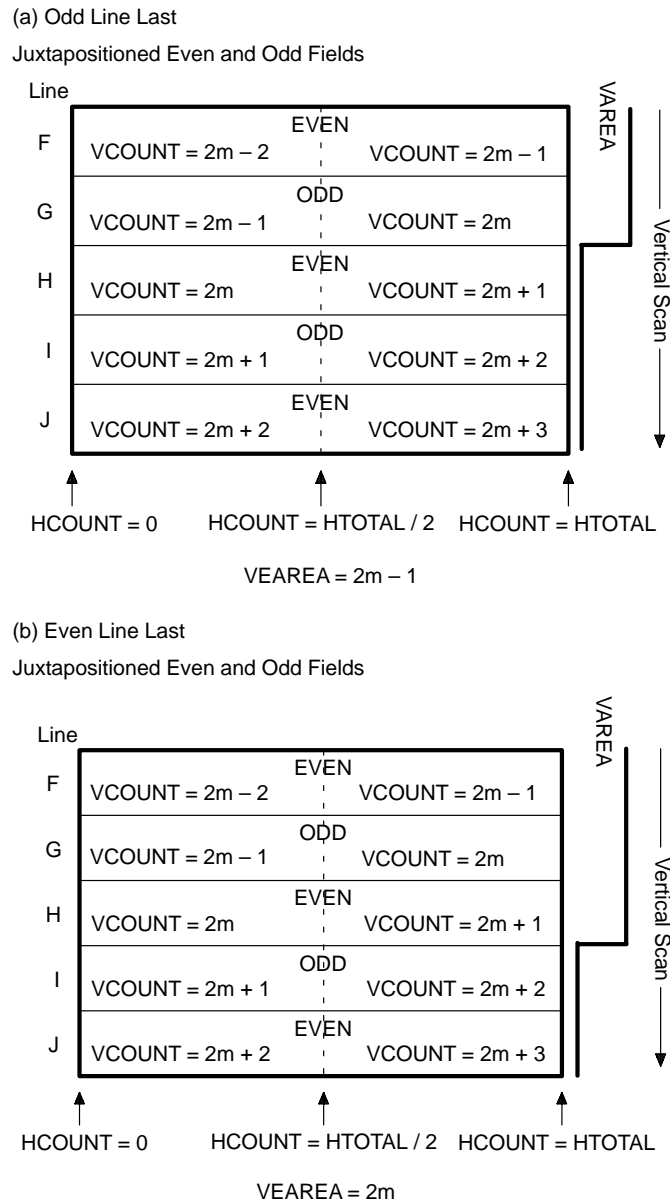
Juxtapositioned Even and Odd Fields





In general, VEAREA must be set to whichever value of VCOUNT is common to the last even and odd lines before VAREA is required to be inactive. For the NTSC standard, VCOUNT is always **even** at the end of even lines and **odd** at the end of odd lines. For the PAL standard, the opposite is true; VCOUNT is always **odd** at the end of even lines and **even** at the end of odd lines. VEAREA programming for the PAL standard is shown in Figure 4–21.

Figure 4–21. Programming VEAREA for PAL



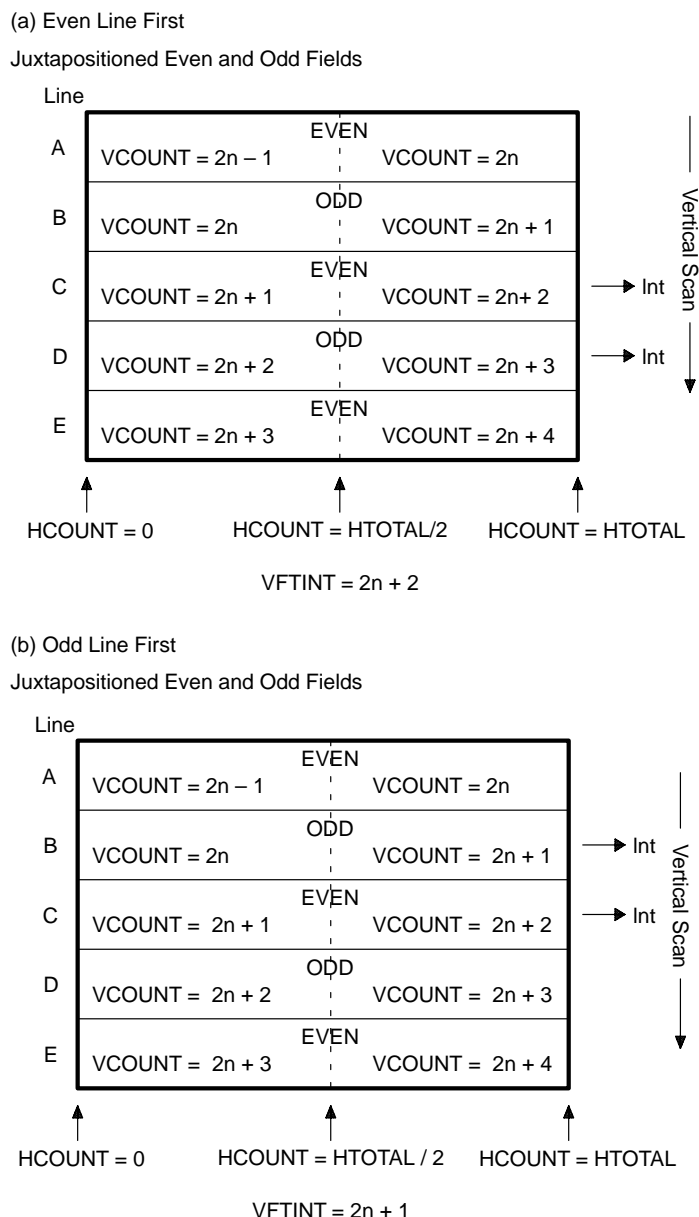
An alternative frame-independent mechanism for programming VEAREA is described in subsection 4.7.4, *Alternative Programming for VFTINT, VSAREA, and VEAREA*.

### 4.7.3 Vertical Frame Timer Interrupt Programming

In interlaced mode, the operation of the vertical frame timer interrupt is dependent on the value of the IIM bit in the frame timer control (FTCTL) register.

- ☐ When IIM = 0, an interrupt will be generated once per frame. The value of VCOUNT is compared to VFTINT at the end of each line. Because the parity of VCOUNT at the end of the even lines is the opposite of that at the end of the odd lines, the condition VCOUNT = VFTINT is true at the end of a line only once per frame in either the even or odd field (depending on the value of VFTINT).
- ☐ When IIM = 1, an interrupt will be generated twice per frame, once in each field. In this case, the mechanism for determining when the interrupt occurs is analogous to the mechanism for beginning or ending VAREA. (See subsection 4.7.1, *Vertical Start Area Programming* and subsection 4.7.2 *Vertical End Area Programming*.) The number of halflines from the start of vertical sync to the interrupt point is different in the two fields. The value in VFTINT must be set to the value of VCOUNT at the end of the interrupt line in the field in which the interrupt first occurs. Figure 4–22 shows how to program VFTINT for NTSC.

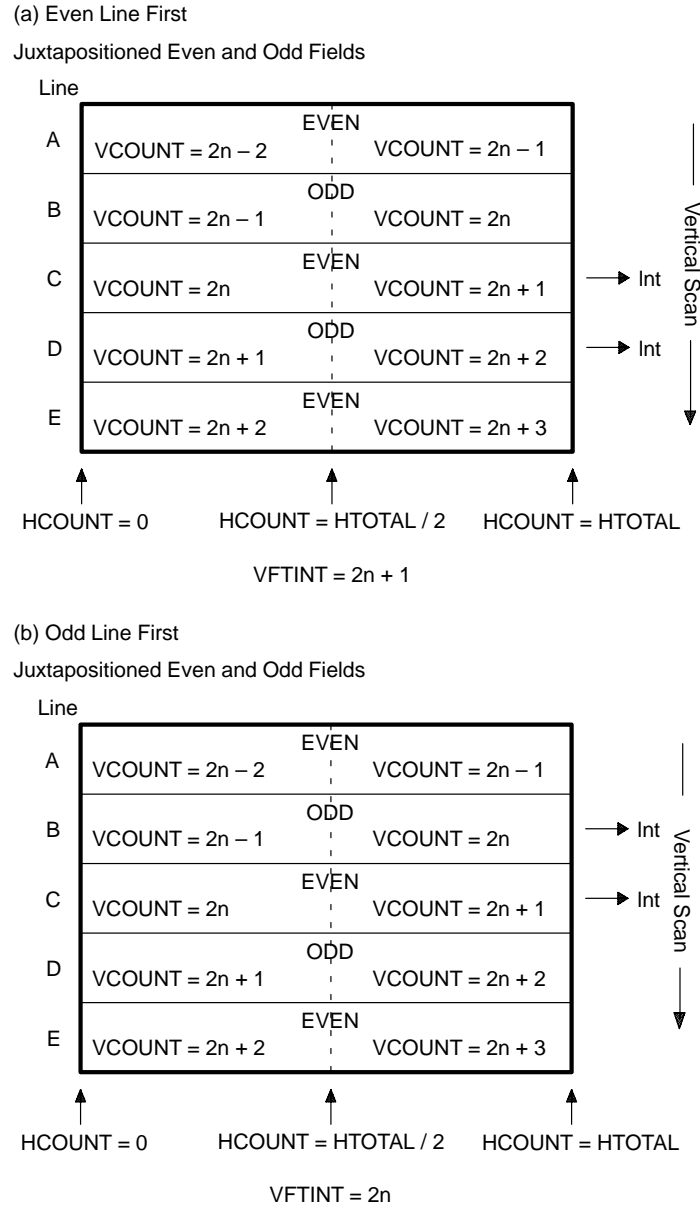
Figure 4–22. Programming VFTINT for NTSC (IIM = 1)



- ❑ **Even Field First:** Figure 4–22 (a) shows how to program VFTINT so that the interrupt occurs on an even line first (line C). Programming VFTINT =  $2n + 2$  causes an interrupt to be generated at the end of line C in the even field and at the end of line D in the odd field.
- ❑ **Odd Field First:** Figure 4–22 (b) shows how to program VFTINT so that the interrupt occurs on an odd line first (line B). Programming VFTINT =  $2n + 1$  causes an interrupt to be generated at the end of line B in the odd field and at the end of line C in the even field.

For the PAL standard, VCOUNT will have the opposite parity at the end of the lines for each field. Programming VFTINT for this case is shown in Figure 4–23.

Figure 4–23. Programming VFTINT for PAL (IIM = 1)



An alternative frame-independent mechanism for programming VFTINT is described in the next subsection.

#### 4.7.4 Alternative Programming for VFTINT, VSAREA, and VEAREA

In absolute terms, VFTINT is equal to the number of lines (noninterlaced) or half-lines (interlaced) from the start of vertical sync to the interrupt point, minus 1. However, programming VFTINT in this manner is sometimes undesirable because:

- ☐ It is not independent of the frame timer mode or the frame dimensions.
- ☐ In interlaced mode with IIM = 0, it is not obvious whether an even value in VFTINT produces an interrupt in the even or the odd field; nor is it obvious whether an odd value in VFTINT produces an interrupt in the even or the odd field.
- ☐ In interlaced mode with IIM = 1, the field in which the interrupt should occur first must be established before VFTINT is programmed.

Frame timer interrupts are most commonly required to occur at a given line within the active (unblanked) portion of the frame. It is, therefore, desirable to be able to program VFTINT independently of the frame timer mode or frame dimensions.

The first active line in a frame is usually called line 0. In a noninterlaced frame, the active line number and VCOUNT increment by 1 on each line. In an interlaced frame, the active line follows the sequence  $n$ ,  $n + 2$ ,  $n + 4$ , etc., in each field (because the even lines are generated in the even field, and the odd lines are generated in the odd field). Since VCOUNT also increments by 2 on each line (once per half-line), the active line number and VCOUNT increase in parallel. Thus, in both cases, the difference between VCOUNT and the active line number is simply the value of VCOUNT on the first active line, which is VEBLNK + 1. Therefore, VFTINT may be programmed by the equation:

$$\text{VFTINT} = \text{VEBLNK} + (\text{frame line number of interrupt} + 1)$$

Referencing interrupts in this way makes the value in VFTINT independent of all factors except the overall frame dimensions.

An interrupt at the start of vertical blanking is achieved by setting  $VFTINT = VSBLNK$ . If an interrupt is required during vertical blanking, this can be achieved by indexing backwards from  $VEBLNK$  as in:

$$VFTINT = VEBLNK - (\text{number of lines before end of blank} + 1)$$

You must ensure that the result is a positive number.

If the vertical components of CAREA are within the unblanked region ( $VEBLNK < VSAREA$  and  $VEAREA < VSBLNK$ ), as is typically the case when CAREA controls an overscan region, the same method can be used to program VEAREA and VSAREA:

$$VEAREA = VEBLNK + (\text{frame line number of end of area} + 1)$$

$$VSAREA = VEBLNK + (\text{frame line number of start of area} + 1)$$

## 4.8 External Synchronization

External synchronization modes allow the frame timers to use horizontal, vertical, and composite sync signals from an external source. Any of the pins configurable as SYNC signals are also configurable as either input, output, or high impedance. You can select external synchronization by using the HPM, VPM, and CPM bits in the frame timer control (FTCTL) register to configure the appropriate sync pins as inputs. At power-up, all three SYNC signals are configured as high impedance to avoid any possible drive conflicts within the system.

### 4.8.1 Video Counter Operation

When you have configured all the sync pins as outputs, the horizontal and vertical counters clear to 0 after  $HCOUNT = HTOTAL$  and after  $VCOUNT = VTOTAL$ , respectively.

Clearing the counters to 0 also initiates the corresponding sync pulse. When one or more of the sync pins is configured as an input, this behavior is changed. A high-to-low transition on a sync input (the beginning of an external sync pulse) causes one or both of the video counters to be loaded from  $SETHCT$  or  $SETVCT$ . Loading the video counters with a programmable value rather than 0 allows the counters to compensate for:

- ☐ Delays inherent in the process of synchronizing an external signal to FCLK within the frame timer
- ☐ The time required for the frame timer to respond to the external sync signal
- ☐ Any external signal skews

HCOUNT and VCOUNT are reloaded as follows:

- ☐ HCOUNT is reloaded with the value in `SETHCT` by a high-to-low transition on either `HSYNC` or `CSYNC`.
- ☐ VCOUNT is reloaded with the value in `SETVCT` either by a high-to-low transition on `VSYNC` or by the first serration pulse on `CSYNC`, which occurs at the beginning of vertical sync. The serration pulse is detected by sampling the `CSYNC` input when the condition `HCOUNT = HESERR` occurs on each horizontal line. If `CSYNC` is sampled low, it is interpreted as a serration pulse. To ensure proper operation, you must program `HESERR` so that the condition `HCOUNT = HESERR` occurs after the end of the equalization pulse but before the end of the external serration pulse.

By causing VCOUNT and HCOUNT to follow the external synchronization signals in this manner, the blanking intervals and screen update (SRT) cycles are also forced to follow the external signals, and the frame timer is, therefore, synchronized to the external video source.



## 4.8.2 External Synchronization Modes

The frame timer can be synchronized to an external timing source by configuring one or more of the  $\overline{\text{SYNC}}$  pins as inputs. Generally, the four types of synchronization considered most useful are:

- ☐ **No external synchronization:** no inputs.
- ☐ **Complete synchronization achieved from a single composite signal:**  $\overline{\text{CSYNC}}$  input. The beginning of each composite-sync signal reloads HCOUNT, and the first serration pulse reloads VCOUNT.
- ☐ **Complete synchronization achieved from separate signals:**  $\overline{\text{HSYNC}}$  and  $\overline{\text{VSYNC}}$  inputs. The beginning of external horizontal sync reloads HCOUNT and the beginning of external vertical sync reloads VCOUNT.
- ☐ **Vertical synchronization only:**  $\overline{\text{VSYNC}}$  input. The beginning of external vertical sync reloads VCOUNT. Horizontal synchronization can be achieved via a phase-locked loop (PLL) that controls FCLK.

The  $\overline{\text{SYNC}}$  inputs are edge triggered and require a minimum pulse width of one FCLK cycle. The frame timer will recognize a falling edge and synchronize it to the FCLK. In systems where the external sync inputs are synchronous to the FCLK, the internal synchronization can be compensated for.

### 4.8.3 Sync Output Operation

The  $\overline{\text{HSYNC}}$  and  $\overline{\text{CSYNC}}$  signals are activated at the beginning of a line, and the  $\overline{\text{VSYNC}}$  signal is activated at the beginning of a vertical field. For external sync modes, the beginning of a line is indicated by a falling edge on the  $\overline{\text{HSYNC}}$  or  $\overline{\text{CSYNC}}$  input, and the beginning of a field is represented by a falling edge on the  $\overline{\text{VSYNC}}$  input or on the first serration pulse on  $\overline{\text{CSYNC}}$ .

However, depending on the way in which external synchronization is used, you may not always want the external transitions to activate the sync signals. The SSE bit value in the frame timer control (FTCTL) register determines how the frame timer activates the sync signals.

- ☐ **SSE = 1:** The sync signals **will be** activated by the external inputs. In other words, if  $\overline{\text{CSYNC}}$  is an input, then  $\overline{\text{HSYNC}}$  is activated by the falling edge of  $\overline{\text{CSYNC}}$  (at the start of a line), and  $\overline{\text{VSYNC}}$  is activated by the first serration pulse on  $\overline{\text{CSYNC}}$ .

If  $\overline{\text{HSYNC}}$  and  $\overline{\text{VSYNC}}$  are inputs, then  $\overline{\text{CSYNC}}$  horizontal pulses are activated by  $\overline{\text{HSYNC}}$ , and  $\overline{\text{CSYNC}}$  serration is activated by  $\overline{\text{VSYNC}}$ .

You can use this mode to attempt a crude synchronization to the external source without skew compensation. It is useful, for example, when the objective is simply to schedule SRT cycles in the appropriate time frame. In this case, you must program HTOTAL to a value much larger than the line length and VTOTAL to a value much larger than the field length. This assures that the condition  $\text{HCOUNT} = \text{HTOTAL}$  does **not** occur (HCOUNT is not cleared, and no horizontal sync pulse is activated), and that the condition  $\text{VCOUNT} = \text{VTOTAL}$  does **not** occur (VCOUNT is not cleared, and no vertical sync pulse is activated).

- **SSE = 0:** The sync signals are **not** activated by the external inputs. This mode must be used when a precise synchronization to the external signals is required. This is accomplished by using SETHCT and/or SETVCT to compensate for skews or bias the sync signals.

In this case, HTOTAL must be programmed to precisely match the number of FCLKs per line in the external system so that the condition  $HCOUNT = HTOTAL$  **will occur** (clearing HCOUNT) and will activate  $\overline{HSYNC}$  (or  $\overline{CSYNC}$ ) at the correct time.

Likewise, VTOTAL must be programmed to precisely match the number of lines per field in the external system so that the condition  $VCOUNT = VTOTAL$  **will occur** (clearing VCOUNT) and will activate  $\overline{VSYNC}$  (or serration pulses on  $\overline{CSYNC}$ ) at the correct time.

---

**Note:**

The SSE bit has no effect on the clearing or reloading of the video counters. HCOUNT will **always** be cleared when the condition  $HCOUNT = HTOTAL$  occurs. If you enable external sync by configuring one of the sync pins as an input, HCOUNT is **always** loaded with the value in SETHCT when a horizontal sync input occurs. Likewise, VCOUNT is always reset when  $VCOUNT = VTOTAL$  and, in external sync mode, is always loaded with the SETVCT value when a vertical sync input occurs.

---

#### 4.8.4 Synchronizing to FCLK

The frame timers synchronize video input signals to their respective FCLKs before passing them on to internal logic. This allows the inputs to be asynchronous to FCLK. The delay from the high-to-low transition of an external sync input to the change at the video output pin is from four to five FCLK periods, depending on the phase relationship of the transition to FCLK.

If the frame timer does not need to follow external video to a one-FCLK accuracy (that is, the frame timer uses the external sync only to schedule SRT cycles or line interrupts), the asynchronous operation may be acceptable.

If the frame timer's video output signals (blank or output sync) are used to control other parts of the system, then the frame timer must synchronize precisely to the external source. In this case, the external sync inputs must be presented synchronously to the FCLK.

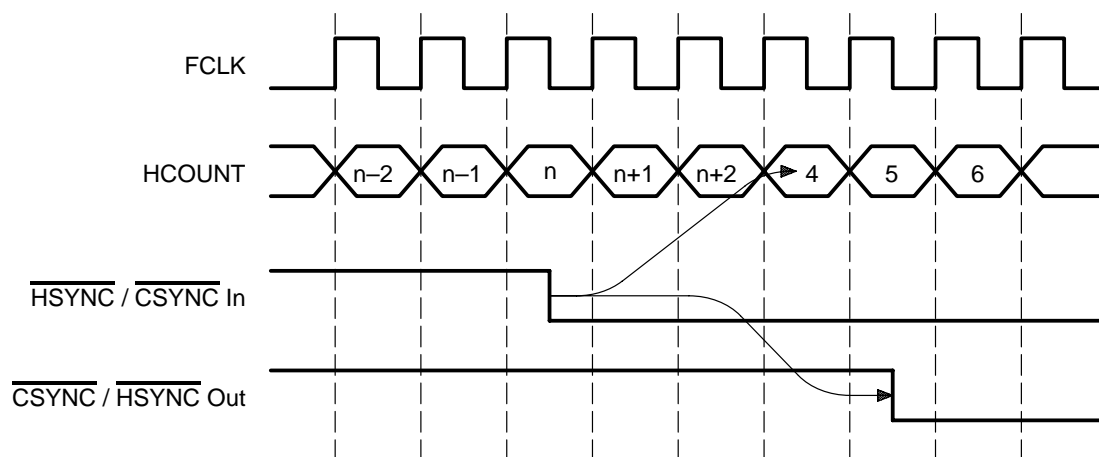
If the setup and hold times of the inputs with respect to the FCLK (see the *TMS320C80 Data Sheet*) are met, then the number of FCLK periods between the input and output signals is a constant number of FCLK cycles.

## 4.9 Synchronization Delay Compensation

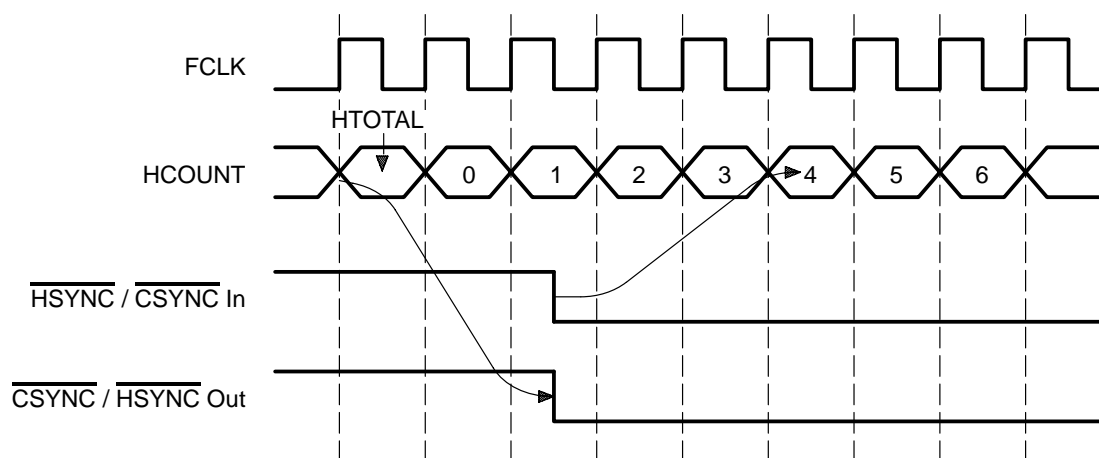
When you configure the frame timer for external horizontal sync, there is a 4-FCLK synchronization delay between the time that the  $\overline{\text{HSYNC}}$  input goes low and the time that it takes effect on an output pin. The SETHCT register is provided to compensate for this delay.

Figure 4–24 (a) illustrates how HCOUNT is loaded on the third FCLK after the transition on the  $\overline{\text{SYNC}}$  input occurs. Programming SETHCT to 4 causes HCOUNT to be loaded with 4 three FCLKs after the transition on the external sync pin. This is equivalent to having HCOUNT clear to 0 on the FCLK cycle just before sync becomes active. Loading HCOUNT with 4 causes the video timing logic to be aligned exactly to the external source.

Figure 4–24. Synchronization Delay Compensation



(a) Initial Synchronization to External Sync (SSE = 1)



(b) Subsequent Synchronization to External Sync (SSE = 0)

SETHCT = 4

Once the video timing logic is synchronized, you can program the timing registers to match the external system. This allows you to align the video timing outputs. Figure 4–24 (b) illustrates how to use SETHCT to align the  $\overline{\text{CSYNC}}/\overline{\text{HSYNC}}$  output with the  $\overline{\text{HSYNC}}/\overline{\text{CSYNC}}$  input. Note that you must clear the synchronization enable (SSE) bit to 0.

The transition on the  $\overline{\text{HSYNC}}/\overline{\text{CSYNC}}$  input causes HCOUNT to be loaded with 4 three FCLKs later. If you program HTOTAL as shown, HCOUNT will be reset to 0 (by the condition  $\text{HCOUNT} = \text{HTOTAL}$ ). The  $\overline{\text{CSYNC}}/\overline{\text{HSYNC}}$  output is driven low during the same FCLK cycle as the input sync.

You can program SETHCT to values other than 4 to account for other external synchronization or propagation skews. Programming SETVCT to a nonzero value causes internal video to be displaced from the external video by a number of scanlines.

Note that if you set SETVCT to an odd number in interlaced mode, it causes the frame timer to have field parity opposite to that of the external frame.

## 4.10 High-Impedance Capability

You can set a  $\overline{\text{SYNC}}$  pin to a high-impedance state. This means that the frame timer does not recognize edges driven in on that pin by an external source, nor does it drive the signal out. This can be particularly useful if you want the ability to drive in an external signal but want the frame timer to recognize it only part of the time.

For example, setting a  $\overline{\text{SYNC}}$  pin to high impedance would be useful if you were attempting to synchronize to an interlaced external source by using a phase-locked loop (PLL). In this case,  $\overline{\text{VSYNC}}$  must be an input so that the gross alignment is correct. Either  $\overline{\text{HSYNC}}$  or  $\overline{\text{CSYNC}}$  would be used as an output to control the PLL, which, in turn, regulates FCLK.

In order for the frame timer to synchronize to the correct (odd or even) field, at least one horizontal sync pulse must be input. However, leaving either  $\overline{\text{CSYNC}}$  or  $\overline{\text{HSYNC}}$  (depending on which one is not an output) as an input can make synchronization difficult because the aligning effects of an edge on this signal conflict with the activities of the PLL.

To resolve this conflict, initially configure both horizontal and vertical syncs as inputs to allow field synchronization. Then, set the pin accepting the horizontal timing signal (and possibly  $\overline{\text{VSYNC}}$  as well) to high impedance. Subsequent input pulses are ignored, allowing the PLL to be the sole source of synchronization.

## 4.11 Frame Lock

It is possible to lock the two frame timers together so that they are synchronous. This may be particularly useful when different timing signal parameters (such as the amount of blanking) are required in the capture and display systems, but where it is otherwise desirable to coordinate the two.

In frame lock mode (the FLE bit of the FTCTL1 register is set to 1), both frame timers operate from FCLK0; frame timer 0 is effectively the master, and frame timer 1 is the slave.

HCOUNT and VCOUNT in frame timer 1 will be reloaded one FCLK0 cycle after HCOUNT and VCOUNT in frame timer 0. To synchronize the two frame timers precisely, frame timer 1's SETHCT register must be set to 1. The alignment between the frame timers may then be adjusted via changes to frame timer 1's SETHCT and SETVCT registers.

---

**Note:**

In frame lock mode, frame timer 1's SSE bit must be programmed appropriately for approximate or precise synchronization (as described on page VC:3-7) to frame timer 0, which effectively acts as an external source to frame timer 1.

---





# Serial Register Transfer Controller

---

---

---

This chapter describes the serial register transfer (SRT) controller. It includes an alphabetical listing of the SRT control registers, along with information pertaining to several characteristics of the SRT controller.

## Topics

<b>5.1</b>	<b>Understanding the SRT Controller .....</b>	<b>VC:5-2</b>
<b>5.2</b>	<b>SRT Controller Registers .....</b>	<b>VC:5-3</b>
<b>5.3</b>	<b>SRT Control Masks .....</b>	<b>VC:5-16</b>
<b>5.4</b>	<b>SRT Events .....</b>	<b>VC:5-17</b>
<b>5.5</b>	<b>Event Prioritization .....</b>	<b>VC:5-23</b>
<b>5.6</b>	<b>Address Generation .....</b>	<b>VC:5-25</b>
<b>5.7</b>	<b>Understanding the SRT Modes .....</b>	<b>VC:5-27</b>

## 5.1 Understanding the SRT Controller

The SRT controller operates synchronously with the rest of the MVP and is responsible for generating all the SRT cycles required to maintain the serial access memories (SAMs) of the VRAMs within the two supported frame memories.

The SRT controller can also generate video controller-initiated packet transfer (VCPT) requests in place of normal SRT requests. You can use VCPTs to control movement of data into or out of a frame buffer.

- ☐ **Frame timer SRTs** are scheduled by timing events controlled by the frame timers' blanking or area timing registers and occur while the frame is inactive; the VRAM serial clock (SCLK) is off, and no serial data transfer is occurring. These SRTs predominantly involve transfers of the entire SAM serial register.
- ☐ **SAM overflow SRTs** are scheduled by the current address (CRNTADR) register associated with each frame memory. This register is incremented by SCLK and keeps a real-time record of the VRAM tap point. When the counter overflows, the active SAM in the VRAM has changed, and an event is generated. These SRTs involve transfers of only half the SAM serial register at a time (split SAM transfers) and occur while the frame is active.
- ☐ **VCPTs** are scheduled by either timing events or SAM overflow events. The data transfer depends on the way that the packet transfer parameters are set up for that particular VCPT.

## 5.2 SRT Controller Registers

Two sets of 32-bit registers are associated with the SRT controller—one set for each of the two supported frame memories. The SRT memory map is shown in Figure 1–7, *SRT Controller Register Map*, on page VC:1-8.

Table 5–1 lists the SRT controller registers and their addresses. Detailed descriptions of these registers follow this table.

Table 5–1. SRT Controller Registers

Description	Register	Address	Register	Address	See Page
Current address register	CRNTADR0	0x0182 033C	CRNTADR1	0x0180 037C	VC:5-4
Field 0 start address register	F0STADR0	0x0182 0308	F0STADR1	0x0182 0348	VC:5-5
Field 1 start address register	F1STADR0	0x0182 0304	F1STADR1	0x0182 0344	VC:5-7
Frame memory control register	FMEMCTL0	0x0182 0300	FMEMCTL1	0x0182 0340	VC:5-8
Line increment register	LINEINC0	0x0182 030C	LINEINC1	0x0182 034C	VC:5-13
Next address register	NEXTADR0	0x0182 0314	NEXTADR1	0x0182 0354	VC:5-14
VRAM serial access memory mask register	SAMMASK0	0x0182 0310	SAMMASK1	0x0182 0350	VC:5-15

**Name** Current Address Register: CRNTADR0/CRNTADR1

**Addresses** CRNTADR0: 0x0182 033C  
CRNTADR1: 0x0182 037C

**Format**

31	16	15	0
CRNTADR			

**Description**

The current address (CRNTADR) register acts as a counter. It is loaded automatically with the address of the first pixel in a frame (noninterlaced) or field (interlaced). In both modes, this will usually correspond to the first pixel in a line as well. CRNTADR is incremented in real time by the SCLK input, which is driven by the VRAMs' serial clock. The least significant 1 in the SAMMASK register determines at which bit CRNTADR is to be incremented. In this way, it keeps track of the address of the current pixel. The CRNTADR register cannot be reliably read when SCLK is running.

The CRNTADR register has a number of uses:

- ☐ When the bits of the CRNTADR register corresponding to the 1s in the SAMMASK register overflow, the active SAM in the VRAMs has changed. This condition generates a SAM overflow event that causes the SRT controller to schedule a split SAM SRT cycle.
- ☐ The value of the CRNTADR register generates the address for split SAM SRTs (by adding or subtracting SAMINC and masking the SAM tap point address bits) so that the resulting address is aligned to the beginning of the other (inactive) SAM.
- ☐ When a register-to-memory transfer is generated at the end of a line, the CRNTADR register indicates to which row in memory data is transferred, as CRNTADR indicates the address of the current row.

**Note:**

Although you do not normally program CRNTADR, you should initialize it to the same value as F0STADR when F0STADR is first written. Failure to do so results in CRNTADR containing an arbitrary address until the first start of field event occurs. Any SRTs generated by a SAM overflow before the first start of field event would be sent to the arbitrary address and could corrupt a system's memory.

**Name** Field 0 Start Address Register: F0STADR0/F0STADR1

**Addresses** F0STADR0: 0x0182 0308  
F0STADR1: 0x0182 0348

**Format**

31	16	15	0
F0STADR			

**Description**

The field 0 start address (F0STADR) register's function is determined by whether interlaced or noninterlaced mode is being used.

☐ Noninterlaced mode

The register should be loaded with the memory address of the first displayed pixel in the frame. Normally this is identical to the value in the F1STADR register. However, the loaded value depends on the configuration of the UED bit in the frame memory control (FMEMCTL) register:

■ **UED = 0**

If UED = 0 **and** horizontal blanking ends before vertical blanking ends ( $\overline{\text{HBLNK}}$  is high when  $\overline{\text{VBLNK}}$  goes high), then the first displayed pixel is not at the beginning of the first line but some distance from it.

■ **UED = 1**

If UED = 1 **and** the horizontal area component ends before the vertical area component ends, then the first displayed pixel is not at the beginning of the first line but some distance from it.

In either case, the value in the F0STADR register is larger than the value in the F1STADR address, and the first complete line to be displayed starts at F1STADR + LINEINC.

☐ Interlaced mode

The F0STADR register should be loaded with the memory address of the first pixel in field 0.

- If the interlaced line repeat (ILR) bit is 0, then field 0 always precedes field 1, and the value loaded in the register should be equal to:

$$F0STADR = F1STADR - LINEINC + Offset$$

- If the interlaced line repeat (ILR) bit is 1, then field 0 and field 1 start on the same line, and the value loaded in the register should be of the form:

$$F0STADR = F1STADR + Offset$$

The value of *Offset* in either case is determined by whether timing events are controlled by frame timer blanking or area registers as selected by the UED bit in the FMEMCTL register:

■ **UED = 0**

Timing events are controlled by frame timer blanking registers. The first pixel in field 0 is in the middle of a horizontal line because vertical blanking ends in the middle of a line during the even field. Therefore, *Offset* is equal to the difference between the memory addresses of the pixel in the center and the pixel at the beginning of the line.

■ **UED = 1**

Timing events are controlled by frame timer area registers. The first pixel in field 0 is at the beginning of a horizontal line because vertical area always starts and ends at the end of a line. Therefore, *Offset* should be equal to 0.

**Name** Field 1 Start Address Register: F1STADR0/F1STADR1

**Addresses** F1STADR0: 0x0182 0304  
F1STADR1: 0x0182 0344

**Format**

31	16	15	0
F1STADR			

**Description**

The function of the field 1 start address (F1STADR) register is determined by whether the frame memory is being used in interlaced or noninterlaced mode.

☐ Noninterlaced mode

The register contains the memory address of the pixel at the beginning of the first line in the frame. Note that this is not necessarily the first pixel that is displayed.

☐ Interlaced mode

The register contains the memory address of the first pixel in field 1. This address should **always** correspond to the pixel at the beginning of a line.



## FMEMCTL0/FMEMCTL1 Frame Memory Control Register

**Name** Frame Memory Control Register: FMEMCTL0/FMEMCTL1

**Addresses** FMEMCTL0: 0x0182 0300  
FMEMCTL1: 0x0182 0340

### Format

31	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			HSS	ILR	PTS			TMS			EMS		UED		FTS		

### Fields

Bits	Name	Function
1–0	FTS	Frame Timer Sequencer
3	UED	Unblanked Event Disable
6–5	EMS	Event Mode Select
9–8	TMS	Transfer Mode Select
12	PTS	Packet Transfer Select
13	ILR	Interlaced Line Repeat
14	HSS	Half SAM Select

**Description** The frame memory control (FMEMCTL) register contains the mode bits that determine the behavior of the frame memory.

**□ FTS** (frame timer sequencer)

Bits 1–0 (FTS) indicate which frame timer generates the frame memory's SRTs. The frame memory can be controlled by either frame timer or disabled completely, as shown in Table 5–2. Disabling the frame memory prevents SRT cycles (except those generated by packet transfers) from occurring.

Table 5–2. Frame Timer Sequencing Modes (FTS)

Bits		Frame Timer Mode
1	0	
0	0	Frame memory disabled (frame timer 0 selected)
0	1	Frame memory sequenced by frame timer 0
1	0	Frame memory disabled (frame timer 1 selected)
1	1	Frame memory sequenced by frame timer 1

The coding of the FTS bits facilitates glitch-free switching between the frame timers. Bit 1 indicates which frame timer sequences the frame memory and should be changed only when bit 0 = 0. Because the frame timer signals are asynchronous with respect to the SRT controller, failure to do so could result in generating spurious SRT events.

**□ UED** (unblanked event disable)

When bit 3 (UED) is set to 0, line events are controlled by blanking. In this case, the line event will be generated when the condition  $HCOUNT = HBLINE$  occurs. Setting UED to 1 causes line events to be controlled by active area. When  $UED = 1$ , the line event will be generated when  $HCOUNT = HALINE$ .

### ☐ **EMS** (event mode select)

Bits 6–5 (EMS) determine which events generate requests to the SRT controller. Three types of events can generate requests to the SRT state machine:

- Start of field
- Line/end of field
- Serial access memory (SAM) overflow

The 2-bit EMS field determines which of the three types of events is recognized. The possible combinations are shown in Table 5–3. The start of field events are always enabled if one of the frame timers is enabled (bit 1 of the FMEMCTL register = 1). When an end of field event occurs, it is treated the same as a line event, but an end of field event is used only when line events are disabled (EMS = 01).

Table 5–3. SRT Event Modes (EMS)

Bits		Events Enabled
6	5	
0	0	Start of field, line, SAM overflow
0	1	Start of field, end of field, SAM overflow
1	0	Start of field, line
1	1	Start of field, line, SAM overflow (address tracking only; no SRTs are generated)

When EMS is set to 11, no SRTs are generated. However, the SRT address generation logic continues to operate as if EMS were 00, updating the NEXTADR register for every line. This can be useful if, for instance, you want to keep track of the current pixel address but do not want to perform any SRTs.

For example, you could overlay a number of lines in a display with data from another source: during this time, SRTs for the background would be unnecessary. When you want to revert to the frame memory as the source, EMS can be set to 00, and the SRTs will begin from the point in the frame memory where they would normally be if EMS had been set to 00 originally.

The SRT controller may generate video controller-initiated packet transfer (VCPT) requests in place of normal SRTs. This functionality is controlled by the packet transfer select (PTS) bit. When the VCPT requests are selected (PTS = 1), the EMS field controls the scheduling of requests, as shown in Table 5–4.

Table 5–4. VCPT Event Modes (EMS)

Bits		Events Enabled
6	5	
0	0	Start of field, SAM overflow
0	1	SAM overflow
1	0	Start of field
1	1	Start of field, SAM overflow (address tracking only; no VCPTs are generated)

#### ☐ **TMS** (serial register transfer mode)

Bits 9–8 determine how data is transferred between the VRAM memory array and its SAM. Table 5–5 shows the available transfer modes and the types of SRT cycles they generate.

Table 5–5. SRT Event Modes (TMS)

Bits		Transfer Mode
9	8	
0	0	Display (reads SRTs)
0	1	Reserved
1	0	Capture (writes SRTs)
1	1	Merge capture (reads and writes SRTs)

☐ **PTS** (packet transfer select)

When bit 12 (PTS) is set to 1, frame timer and SAM overflow events generate VCPT requests in place of normal SRT requests. This allows the video controller to control the movement of data into or out of a frame buffer. VCPT requests are scheduled according to the status of the EMS bits, as shown in Table 5–4.

☐ **ILR** (interlaced line repeat)

When bit 13 is set to 1, interlaced line repeat is enabled. This causes each horizontal line to occur once per field rather than once per frame.

☐ **HSS** (half length SAM select)

Bit 14 should be set to 1 when the frame memory consists of VRAMs that have a SAM length that is one-half the length of a row in the memory array (for example, a VRAM with 512 columns per row and a SAM of 256 bits). Setting this bit modifies the way in which shift register transfer addresses are generated to provide the proper tap point for half-length SAM type devices. This bit should be cleared to 0 when VRAMs with a SAM length identical to the row length are used.

**Name** Line Increment Register: LINEINC0/LINEINC1

**Addresses** LINEINC0: 0x0182 030C  
LINEINC1: 0x0182 034C

**Format**

31	16	15	0
LINEINC			

**Description**

The line increment (LINEINC) register contains the difference between the memory addresses of two vertically adjacent pixels in a frame. Thus, adding LINEINC to the address of the first pixel on a given line yields the address of the first pixel on the next line.

This is independent of whether the frame is interlaced or not. LINEINC is automatically doubled in interlaced mode to obtain the address of the next vertically adjacent pixel in the same field.

## NEXTADR0/NEXTADR1 Next Address Register

**Name** Next Address Register: NEXTADR0/NEXTADR1

**Addresses** NEXTADR0: 0x0182 0314  
NEXTADR1: 0x0182 0354

### Format

31	16	15	0
NEXTADR			

### Description

The next address (NEXTADR) register is automatically updated by start of field and line events with the address of the first pixel on the next line. All or part of the address (depending on the mode) output during the SRT for the next line is taken from this register **before** it is updated.

When full SAM SRTs are performed on every line (line events are enabled), you can use a frame timer interrupt to modify the value in NEXTADR with software. This allows you to change the area of memory in which image data is stored or out of which video data is displayed, part way through the frame. For example, you can split the screen by displaying one frame buffer in the upper half of the screen and another in the lower half. There is no synchronization between NEXTADR and the MP. You should keep this in mind when reading from or writing to this register.

#### Note:

Although you do not normally program NEXTADR, you should initialize it to the same value as F0STADR when F0STADR is first written. Failure to do so results in NEXTADR containing an arbitrary value until the first start of field event occurs. If an SRT is generated by a SAM overflow before the first start of field event, the SRT is sent to an arbitrary address and could corrupt a system's memory.

**Name** VRAM Serial Access Memory Mask Register: SAMMASK0/  
SAMMASK1

**Addresses** SAMMASK0: 0x0182 0310  
SAMMASK1: 0x0182 0350

**Format**

31	16	15	0
SAMMASK			

**Description**

The VRAM serial access memory mask (SAMMASK) register contains a mask of 1s to indicate which bits of the addresses derived from F1STADR, F0STADR, and LINEINC are used to indicate the tap point within each of the VRAM's **split** serial registers (or SAMs). Thus, if **each** split SAM contains  $2^n$  bits, there are  $n$  contiguous 1s in SAMMASK. The alignment of the field of 1s within SAMMASK depends on which address pins are physically wired to the VRAMs. The least significant 1 in SAMMASK should correspond to the logical address bit connected to the least significant bit (LSB) of the VRAM's address lines at the falling edge of  $\overline{\text{CAS}}$ .

The SAMMASK register determines:

- ☐ Which bit of the CRNTADR register should be incremented on each SCLK cycle
- ☐ Which bits of the CRNTADR register are to be monitored for overflow
- ☐ The increment or decrement applied to the CRNTADR register in order to point to the data to be transferred into or out of the inactive SAM during a split SAM SRT
- ☐ Which bits of the CRNTADR register are to be masked to 0 during a split SAM SRT
- ☐ How the CRNTADR and the NEXTADR registers should be combined when performing write SRTs

For an example of how the SAMMASK register is used, see subsection 5.4.4, *SAM Overflow Event*.



## 5.3 SRT Control Masks

In addition to the registers, the SRT controller contains two 32-bit control masks (SAMINC and COLMASK) that it uses for SRT address calculation. Although they are not accessible to you, a description of the masks can help you understand how SRT addresses are generated:

☐ Serial access memory increment (SAMINC)

This mask is all 0s, except for a 1 at the bit position immediately above the most significant 1 in the SAMMASK register.

☐ Column mask (COLMASK)

This mask looks like SAMMASK but has one or two extra 1s in the bit position(s) above the SAMMASK's most significant 1. The number of extra 1s depends on the length of the VRAM's SAM, as indicated by the half-length SAM select (HSS) bit in the frame memory control (FMEMCTL) register.

- If HSS = 0, the VRAM SAM is the same length as the memory array rows, and COLMASK looks like SAMMASK with an extra 1 in the bit position above the most significant 1 in SAMMASK.
- If HSS = 1, the VRAM SAM is half the length of the memory array rows, and COLMASK looks like SAMMASK with **two** additional 1s in the bit positions above the most significant 1 in SAMMASK.

## 5.4 SRT Events

In most applications, some or all of the event types occur during blanking or during the inactive area (depending on whether blanking or area is being used to control events). In some cases, the occurrence of these events must be carefully controlled to prevent timing problems.

Typically, the clock (SCLK) that is used to control the data stream between the VRAMs and the capture or display device is high frequency, and pipelining of the serial data often occurs. When blanking begins (or the active area ends), this pipeline must be drained of its remaining data. Thus, the video controller's blanking and/or area outputs are often delayed before being applied to the display or capture device. The length of this delay depends on the SCLK frequency and the depth of the serial data pipeline, but it must generally be long enough to allow enough SCLK pulses to completely drain the pipeline.

In order to avoid corrupting the serial data stream, it is essential that the SRTs that are scheduled during blanking (or during the inactive area) not occur until after SCLK has stopped and the display or capture device itself is in blanking. Since the video controller has no way to determine when this has happened, the HBLINE (for blanking control) and the HALINE (for area control) registers allow you to specify precisely where on the horizontal line the SRT events occur by using the condition  $HCOUNT = HBLINE$  or  $HCOUNT = HALINE$  to generate the SRT.

The HBLINE or HALINE registers are normally programmed so that the SRTs are generated at or shortly (a few SCLK pulses) after the start of horizontal blanking (or the end of the horizontal active area). If  $HBLINE = HSBLNK$  (or  $HALINE = HEAREA$ ), SRTs are guaranteed **not** to occur until after the  $\overline{HBLNK}$  output has gone low or HAREA has gone low.

### 5.4.1 Start of Field Event

A start of field event causes the SRT controller to reset the current address (CRNTADR) register to the beginning of the frame (non-interlaced) or field (interlaced) and then generate the required SRT cycle(s) for the first active line. The next address (NEXTADR) register is then updated to the address for the second line.

This event can be derived from one of two sources, depending on the state of the unblanked event disable (UED) bit in the frame memory control (FMEMCTL) register:

☐ **UED = 0**

The start of field event is caused by the end of vertical blanking. It is generated by the condition  $VCOUNT = VEBLNK$ . In noninterlaced mode (where vertical blanking may last only one line), the event is further qualified by the condition  $HCOUNT = HBLINE$ .

☐ **UED = 1**

The start of field event occurs at the beginning of the vertical active area when  $VCOUNT = VSAREA$ . In noninterlaced mode (where vertical area may last only one line), the event is further qualified by the condition  $HCOUNT = HALINE$ .

### 5.4.2 End of Field Event

End of field events are used only when image capture SRT cycles are being performed. They cause the data captured on the last active line of the frame (noninterlaced) or field (interlaced) to transfer into the memory array, but only when line events are disabled (the EMS bits in the FMEMCTL register are set to 01). The state of the UED bit in the FMEMCTL register determines which of the two sources generates the event:

☐ **UED = 0**

The end of field event is generated after the start of vertical blanking (VCOUNT = VSBLNK) when HCOUNT = HBLINE.

☐ **UED = 1**

The end of field event is generated at the end of the vertical active area when VCOUNT = VEAREA and HCOUNT = HALINE.

### 5.4.3 Line Event

The line event causes the SRT controller to generate the required SRT cycle(s) for the next horizontal line, and then update the next address (NEXTADR) register for the subsequent line. This event occurs only during the active region of the frame (field) and is derived from the state of the UED bit in the FMEMCTL register:

☐ **UED = 0**

The line event is generated when HCOUNT = HBLINE at the end of an active horizontal line. The event does not occur for any lines which begin after vertical blanking.

☐ **UED = 1**

The line event is generated when HCOUNT = HALINE at the end of an active horizontal area. The event does not occur for any lines which begin after vertical area goes inactive.

## 5.4.4 SAM Overflow Event

Unlike the other events, the SAM overflow event is generated within the SRT controller itself. This event determines when a split SRT cycle must be performed to service the inactive half of the VRAMs' split SAM. It occurs when the value in the current address (CRNTADR) register is incremented past the value contained in the SAMMASK register.

For example, if the frame memory is physically connected to address lines A[20:12], then the logical address bits used by the VRAM are A(20:12) at  $\overline{\text{RAS}}$  and A(11:3) at  $\overline{\text{CAS}}$ . (For more information on how these address bits are determined, see subsection 7.4.1, *Address Multiplexing*, in the *MVP Transfer Controller User's Guide*.) If the length of the split SAM is 128 bits, then the SAMMASK register would be programmed to 0x0000 03F8, as shown in Figure 5–1.

### Note:

In notation used in this subsection, [ ] (square brackets) indicate pin locations, ( ) (parentheses) indicate bit locations, and a : (colon) indicates a range. For example, A[12] refers to pin 12 of address A, A(12) refers to bit 12 of address A, and A(20:12) refers to bits 20 to 12 of address A. Because of address shifting, A[20:12] and A(20:12) are not necessarily identical.

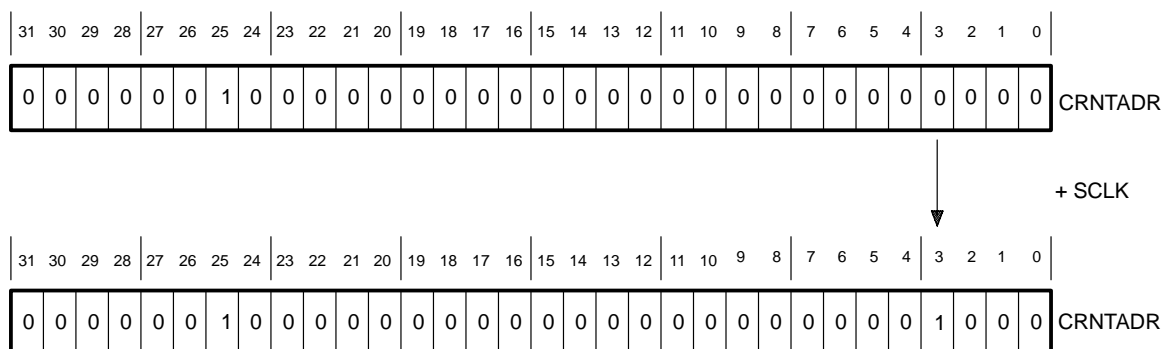
Figure 5–1. SAM Overflow Event at Beginning of Frame



The least significant 1 corresponds to the least significant address bit of the VRAM's column address. Since there are 128 ( $2^7$ ) bits in the split SAM, the SAMMASK register contains seven contiguous 1s.

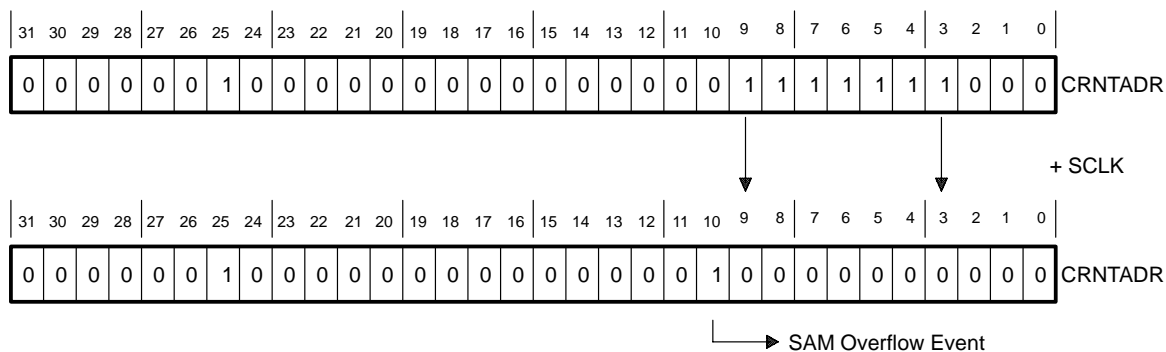
If the frame memory begins at address 0x0200 0000 (the address of the first pixel in the frame), then the CRNTADR register is loaded with 0x0200 0000 at the beginning of the frame. With each SCLK pulse, the value in the CRNTADR register increments, beginning with bit 3 because this bit corresponds to the least significant 1 in the SAMMASK register (see Figure 5–2).

Figure 5–2. SAM Overflow Event After the First SCLK Pulse



Once the value in the CRNTADR register has reached 0x0200 03F8, the next SCLK causes an overflow in the bits corresponding to the SAMMASK register, and an overflow event is generated, as Figure 5–3 shows.

Figure 5–3. SAM Overflow Event After 128th SCLK Pulse



### 5.4.5 VC-Initiated Packet Transfer Requests

The SRT controller can generate video controller-initiated packet transfer (VCPT) requests in place of normal SRT requests. Like SRTs, VCPTs can be scheduled by either timing events or SAM overflow events.

VCPTs control the movement of data into or out of a frame buffer. The type of data transfer depends on the way that the packet transfer parameters are set up for that particular VCPT.

VCPT requests are selected by setting the packet transfer select (PTS) bit in the appropriate frame memory control (FMEMCTL) register. Packet transfer requests are scheduled according to the status of the event mode select (EMS) field in the FMEMCTL register as shown in Table 5–4.

VCPTs share their linked-list starting addresses with externally initiated packet transfers XPT4–XPT7, as shown in Table 5–6.

Table 5–6. Linked-List Start Addresses

Linked-List Start Address	XPT	VCPT
0x0101 00E0	XPT7	Start of Field 0 (SOF0)
0x0101 00E4	XPT6	SAM Overflow 0 (SAM0)
0x0101 00E8	XPT5	Start of Field 1 (SOF1)
0x0101 00EC	XPT4	SAM Overflow 1 (SAM1)

Faulting is supported during both XPTs and VCPTs; however, there is no suspend area for faulting VCPTs. A faulting VCPT immediately terminates, and the appropriate bits in the TC's fault status (FLTSTS) register are set. The linked-list starting address still points to the faulting packet transfer's parameter table in the event of a fault.

For more information on XPTs, see Section 5.1, *Externally Initiated Packet Transfers (XPTs)*, in the *MVP Transfer Controller User's Guide*.

## 5.5 Event Prioritization

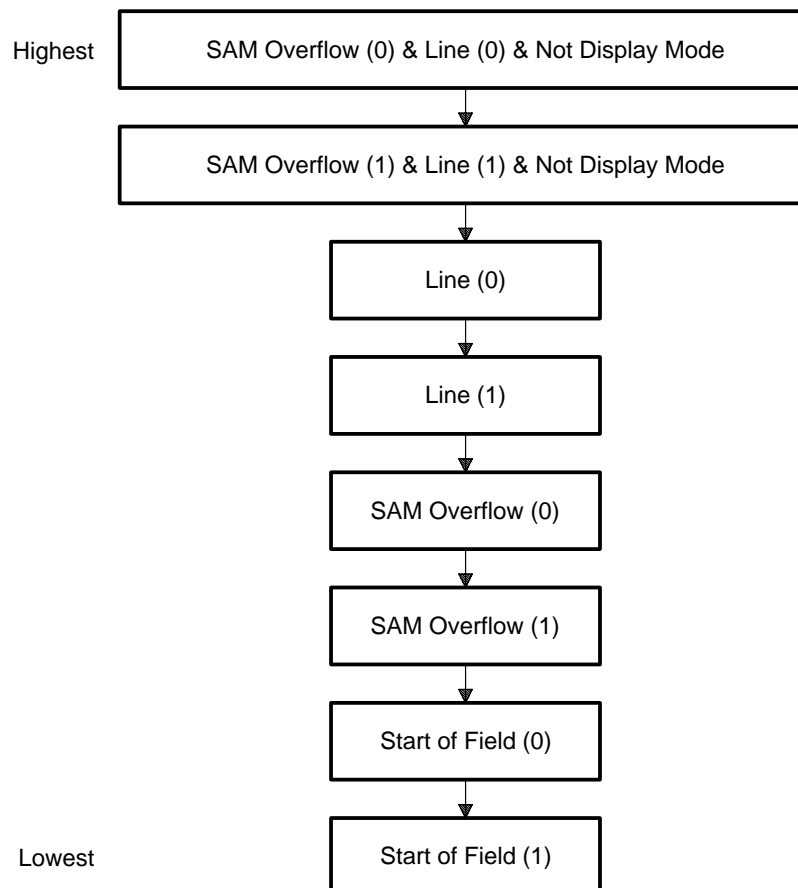
Fourteen possible event signals can cause an SRT to be scheduled (six from each frame timer and one from each CRNTADR register). These event signals can be divided into three categories:

- ☐ Start of field (unblanked or active area)
- ☐ Line/end of field (unblanked or active area)
- ☐ SAM overflow

The FTS, UED, EMS, and TMS bits of the FMEMCTL registers determine which of these events are recognized by the SRT controller and which frame memory they apply to.

With four independent sources of timing events, it is possible for more than one of the eight SRT requests to be activated simultaneously. Therefore, a hierarchy governs requests, as shown in Figure 5–4.

Figure 5–4. SRT Event Priority





Three factors determine where in the hierarchy an event is placed:

- ☐ **Event type**

The shortest SRT events are given highest priority.

- ☐ **Frame memory**

If two similar events for different frame memories occur simultaneously, frame memory 0 is serviced first.

- ☐ **Event sequence**

A line event has a higher priority than a serial access memory (SAM) overflow event. However, in capture modes, a SAM overflow request scheduled at the end of the line **must** be serviced before the line request is serviced if both events pertain to the same frame memory.

## 5.6 Address Generation

The exact sequence of SRTs generated by each type of request varies according to the transfer mode specified by the TMS bits in the FMEMCTL register. However, the way in which the address is manipulated is determined only by the event type. This depends on whether the frame is interlaced and, if so, whether the interlaced line repeat is enabled (the ILR bit in the FMEMCTL register is set to 1). For more information on interlaced line repeat, refer to page VC:5-34.

Ignoring line repeat, the algorithm by which the next address (NEXTADR) register and the current address (CRNTADR) register are maintained can be summed up as follows:

☐ Start of field events

- One or more SRTs are performed using the address in the field 1 start address (F1STADR) or field 0 start address (F0STADR) register.
- This address is copied into the CRNTADR register.
- The values in the F1STADR and line increment (LINEINC) registers are used to calculate the address for the beginning of the next line. This address is stored in the NEXTADR register.

☐ Line events

- One or more SRTs are performed using the value in the CRNTADR or NEXTADR register (or a combination of the two) to either capture data from the previous line or set up data for the next line.
- The address in the NEXTADR register is transferred to the CRNTADR register.
- The value in the NEXTADR register is then incremented using the value in the LINEINC register.

### ☐ SAM overflow events

- The value in the CRNTADR register is incremented by the VRAM serial clock (SCLK), thereby tracking the address of the current pixel in real time.
- When a SAM boundary is crossed, one or more split SAM SRTs are performed to either capture data from or set up data for the inactive SAM. The addresses used are derived by adding or subtracting the value in the SAMINC control mask to or from the value in CRNTADR and masking the tap point to 0 using the value in the SAMMASK register.

The value in the NEXTADR register is incremented by the value in the LINEINC register for noninterlaced frames or by twice the value in the LINEINC register for interlaced frames.

## 5.7 Understanding the SRT Modes

The two-bit SRT mode (TMS) field in the FMEMCTL register determines the manner in which data is transferred between the SAM registers and the VRAM memory array. Table 5–5 shows how each mode is selected.

The following subsections describe the transfer modes and the SRT sequences generated for each type of scheduling request.

### 5.7.1 Display Mode

Display mode is used when an image in the frame memory must be shifted out of the VRAMs and displayed.

The following SRT sequences are generated for each type of scheduling request:

#### ☐ Start of field requests

- A full SAM read SRT is performed using the value in the NEXTADR register (which, in this case, is the value in the F1STADR or F0STADR register).
- If SAM overflow requests are enabled **and** the pixel pointed to by the address in the NEXTADR register is aligned to the upper SAM, a split SAM read is also performed to ensure that the inactive (lower) SAM contains valid data.

The address is derived from the value in NEXTADR by masking all the tap point address bits to 0 using the value in SAMMASK and then adding the value in SAMINC. The address thereby transfers the next subrow into the inactive SAM and sets the tap point to the first pixel in the inactive SAM.

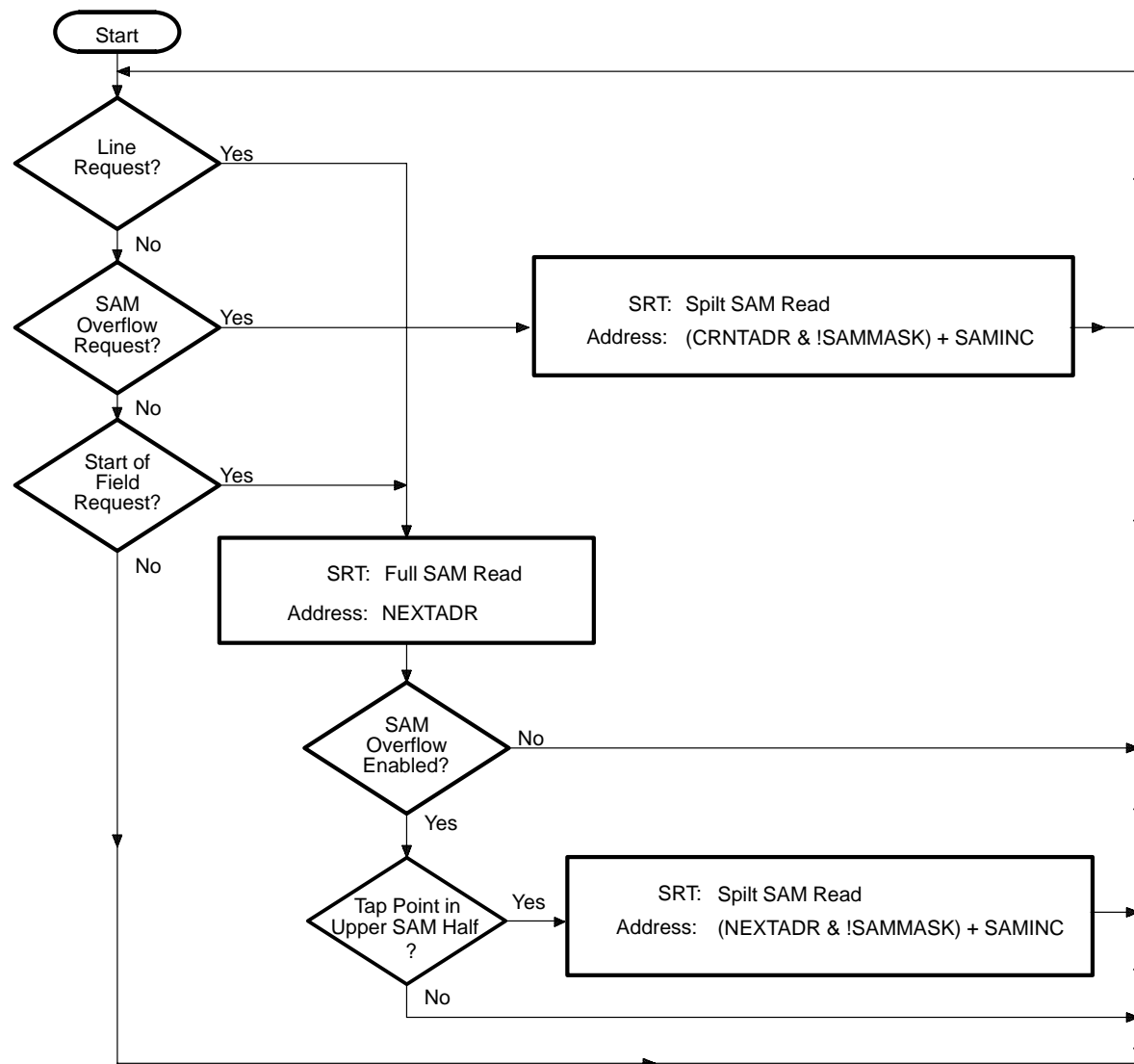
#### ☐ Line requests

Line requests follow the same procedures as that of start of field requests.

#### ☐ SAM overflow requests

A split SAM read is performed to load the next subrow into the inactive SAM. The address is derived from the value in CRNTADR by masking the tap point to 0 and adding the value in SAMINC.

Figure 5–5. Display Mode SRT Sequence



## 5.7.2 Capture Mode

Capture mode is used when an image must be captured from an input device without regard for the data that may already be in the VRAM. In this mode, memory locations other than those corresponding to the captured image can be overwritten.

The following SRT sequences are generated for each type of scheduling request:

### ☐ Start of field requests

- A full SAM write SRT is performed using the **row address** from the CRNTADR register and the **column address** from the NEXTADR register. (In this case, NEXTADR and CRNTADR are equal, so the address is the value in NEXTADR.) This configures the SAM for input and sets the tap point for the active SAM. For half-length SAM VRAMs, the MSB of the tap point specifies which half of the memory row the SAM will (eventually) be transferred to.

- If SAM overflow requests are **enabled**, a split SAM write SRT is performed to enable VRAM split mode operation and set the tap point for the inactive SAM.

The address is derived from the value in NEXTADR by masking the tap point to 0 and then adding it to the value in SAMINC. For half-length SAM VRAMs, the MSB of the tap point specifies the half of the memory row to which the inactive SAM will (eventually) be written.

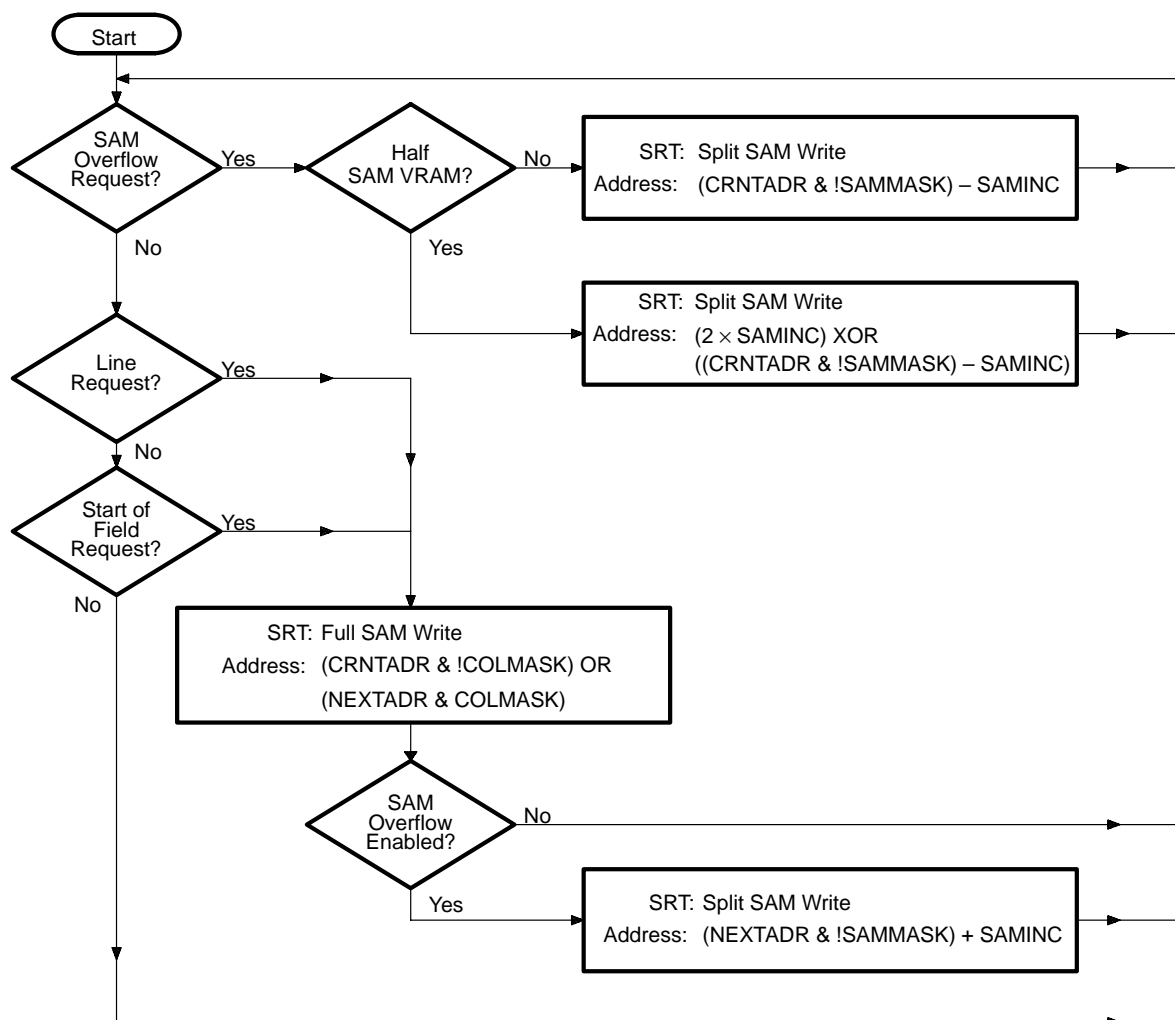
### ☐ Line requests

Line requests follow the same procedures as that of start of field requests except the value in NEXTADR is not necessarily equal to the value in CRNTADR.

## SAM overflow requests

- A split SAM write is performed to transfer the inactive SAM to the previous subrow in the VRAM matrix. The address is derived from the value in the CRNTADR by masking the tap point to 0 and then subtracting the value in the SAMINC from it.
- If HSS = 1, the MSB of the column address specifies the half of the VRAM row to which the SAM will be written the next time it is transferred. Since this is always opposite the half of the row to which the SAM is currently being transferred, this address is derived by inverting the MSB of the current column address. This is accomplished by XORing the address with twice the value in the SAMINC control mask.

Figure 5–6. Capture Mode SRT Sequence



### 5.7.3 Merge Capture Mode

Merge capture mode can be used to capture an image while preserving the data in the rest of the frame memory. Thus, only those memory locations corresponding to bits that are actually shifted into the VRAMs from the input device are overwritten. Merge capture mode requires extra transfer cycles to achieve this, so if the nonimage frame memory locations do not need to be preserved, then capture mode can be used to maximize bus bandwidth.

The following SRT sequences are generated for each type of scheduling request:

#### ☐ Start of field requests

- A full SAM read SRT is performed using the value in the NEXTADR register.
- Then a full SAM write is performed using the value in NEXTADR. This configures the SAM for input and sets the tap point to the active SAM half. For VRAMs with half-length SAMs, the MSB of the tap point also specifies the half of the VRAM row to which the SAM will (eventually) be written.
- If SAM overflow events are enabled and the pixel pointed to by the value in NEXTADR is aligned to the **upper** SAM, a split SAM write is performed to set the tap point for the inactive SAM. The address is generated by using the value in SAMMASK to mask the tap point address bits of the value in CRNTADR and then subtracting the value in SAMINC from the result.

A split SAM read is also performed to ensure that the inactive SAM contains data from the **next** subrow. The address output is derived from the value in CRNTADR by masking all the tap point address bits to 0 using the value in SAMMASK and then adding it to the value in SAMINC.

- If SAM overflow events are enabled and the pixel pointed to by the value in NEXTADR is aligned to the **lower** SAM, a split SAM write is performed to set the tap point for the inactive SAM. The address is generated by using the value in SAMMASK to mask the tap point address bits of the value CRNTADR and then adding it to the value in SAMINC.

This initializes the SAMs with the data that will be overwritten by subsequent SAM write SRTs so that only the bits actually captured are changed.



☐ Line requests

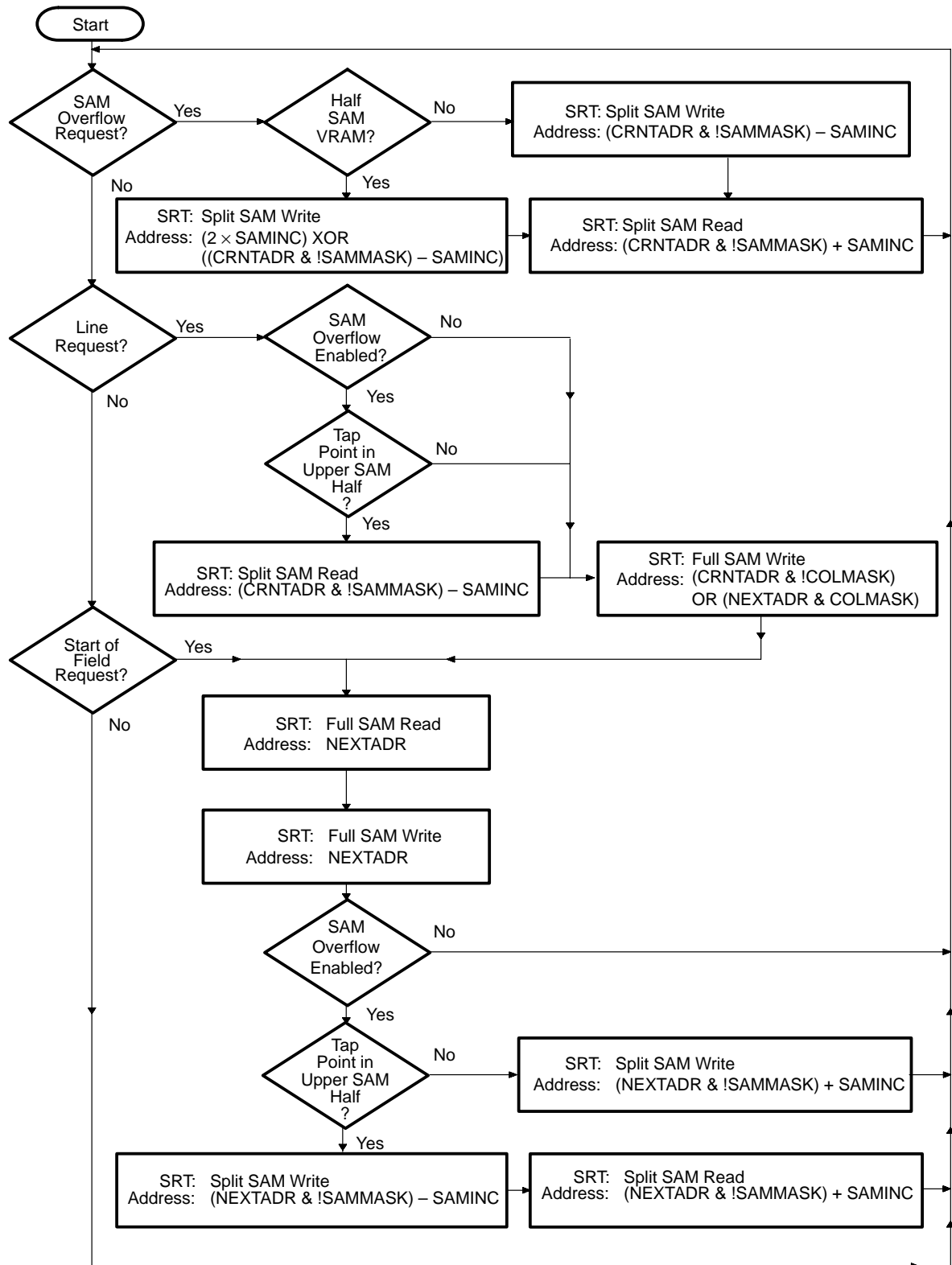
- If SAM overflow events are enabled and the pixel pointed to by the address in CRNTADR is aligned to the upper SAM, a split SAM read SRT is performed to ensure that the inactive SAM contains data from the current row. The address is derived from the value in CRNTADR by masking the tap point address bits to 0 using SAMMASK and then subtracting the value in SAMINC from it.
- The SRT that is performed is a full SAM write. This is performed using the row address from CRNTADR and the column address from NEXTADR to transfer the current subrow back into the array.
- The start of field sequence is then performed: for example, a full SAM read SRT, followed by a split SAM write SRT, possibly followed by a split SAM read SRT (see page VC:5-31 for more information about the start of field sequence).

☐ SAM overflow requests

- A split SAM write is performed to transfer the inactive SAM to the previous subrow in the VRAM memory array. The address is derived from the value in CRNTADR by masking the tap point to 0 and then subtracting the value in SAMINC from it.
- If HSS = 1, the address is XORed with twice the value in SAMINC to specify the half of the VRAM row to which SAM will be written the next time it is transferred.

Then a split SAM read SRT is performed to ensure that the inactive SAM contains data from the next subrow. The address output is derived from the value in CRNTADR by masking the tap point address bits to 0 using the value in SAMMASK and then adding the value in SAMINC.

Figure 5–7. Merge Capture Mode SRT Sequence



## 5.7.4 Interlaced Line Repeat

When the interlaced line repeat (ILR) bit in the FMEMCTL register is set to 1 in interlaced mode, each horizontal line is addressed in both even and odd fields (rather than odd lines only in odd fields and even lines only in even fields). Setting the ILR bit in noninterlaced mode has no effect.

Line repeat is achieved by incrementing the value in the next address (NEXTADR) register with the value in the line increment (LINEINC) register. This generates the address of the next line in the **frame**. (Incrementing the value in NEXTADR by twice the value in the LINEINC register would generate the address of the next line in the **field**.)

---

**Note:**

Because the same line is addressed at the beginning of both fields, the address in the F1STADR register must point to the beginning and the address in the F0STADR register must point to the middle of the **same** line, rather than pointing to the beginning of line 1 and the middle of line 0.

---

## Programming Procedures and Examples

This appendix describes the procedures for programming the serial register transfer (SRT) controller and the frame timer registers for both interlaced and noninterlaced modes.

### Topics

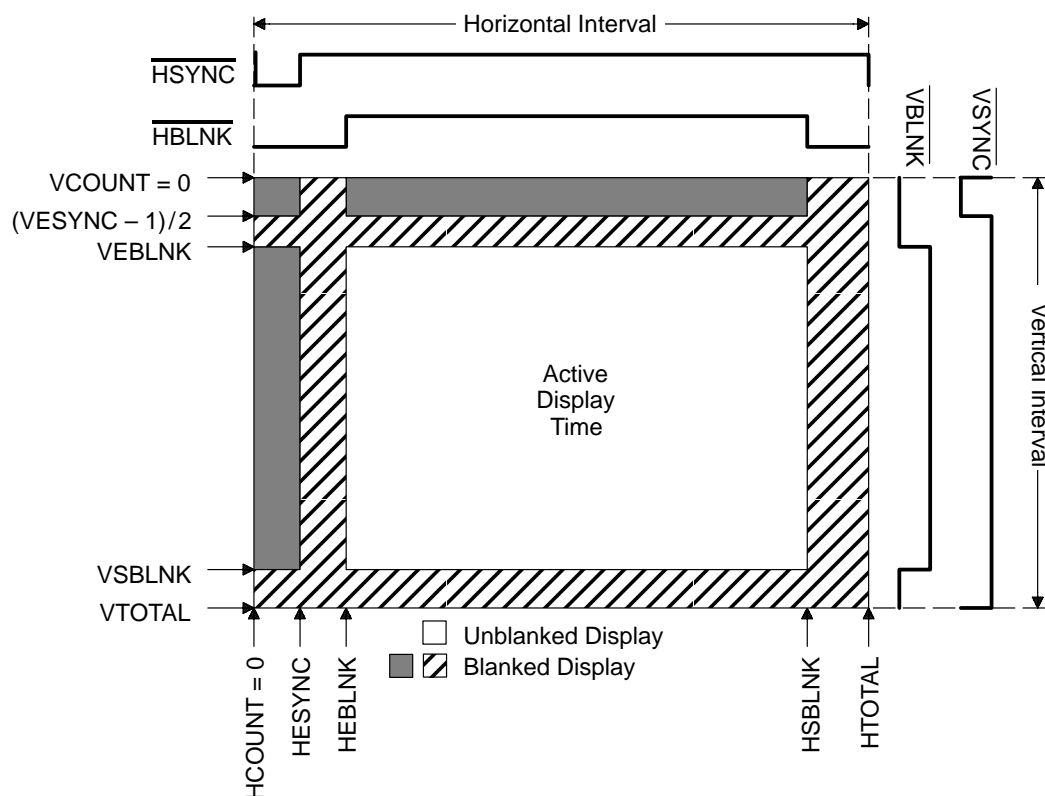
<b>A.1</b>	<b>Relationship Between Horizontal and Vertical Timing Signals</b>	<b>VC: A-2</b>
<b>A.2</b>	<b>Screen Resolution</b>	<b>VC: A-4</b>
<b>A.3</b>	<b>Noninterlaced Monitor Specifications</b>	<b>VC: A-5</b>
<b>A.4</b>	<b>Pixel Clocks</b>	<b>VC: A-6</b>
<b>A.5</b>	<b>Programming the Frame Timer Registers for Noninterlaced Video</b>	<b>VC: A-7</b>
<b>A.6</b>	<b>Programming the SRT Controller Registers</b>	<b>VC: A-10</b>
<b>A.7</b>	<b>Programming the Horizontal Line Event Registers</b>	<b>VC: A-12</b>
<b>A.8</b>	<b>NTSC Composite Interlaced Video Display</b>	<b>VC: A-13</b>
<b>A.9</b>	<b>Programming the Frame Timer Registers for Interlaced Video</b>	<b>VC: A-16</b>
<b>A.10</b>	<b>Frame Timer Interrupts</b>	<b>VC: A-18</b>

## A.1 Relationship Between Horizontal and Vertical Timing Signals

Figure A–1 illustrates the relationship between the horizontal and vertical timing signals and the frame timer registers that control the signal transitions.

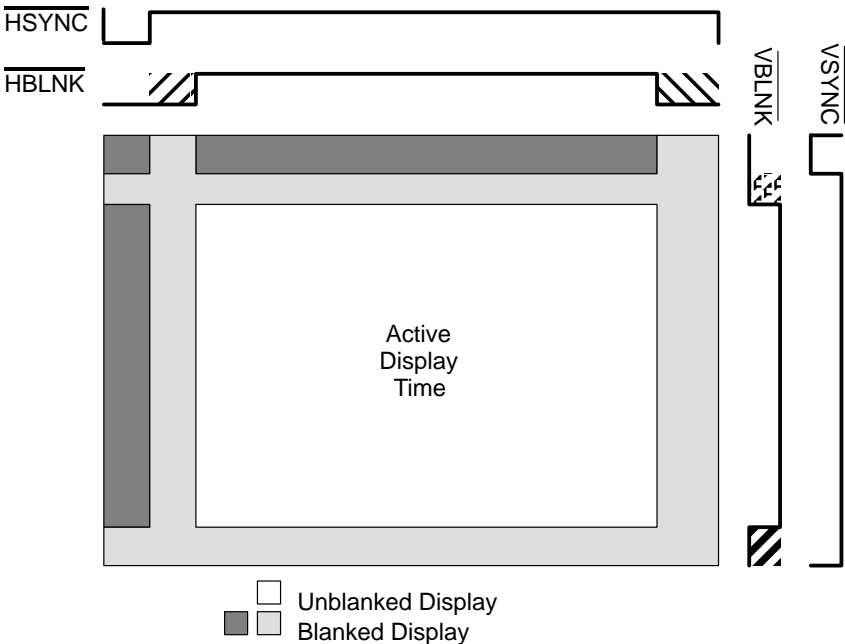
- The horizontal sync and blanking signals span a single horizontal scan within the frame and are repeated for each line.
- The vertical sync and blanking signals span one complete frame.





Figure A–1. Horizontal and Vertical Timing Relationship



The timing of the horizontal and vertical sync and blanking pulses depends on the display that you are using. The relationship between the sync and blanking pulses shown in Figure A–1 defines what are known as the **porches** in video timing. The porches are shown in Figure A–2.

Figure A-2. The Porches



- |   |                        |  |
|---|------------------------|--|
|    | Horizontal Back Porch  | The interval between the end of horizontal sync and the end of horizontal blanking             |
|   | Horizontal Front Porch | The interval between the beginning of horizontal blanking and the beginning of horizontal sync |
|  | Vertical Back Porch    | The interval between the end of vertical sync and the end of vertical blanking                 |
|  | Vertical Front Porch   | The interval between the beginning of vertical blanking and the beginning of vertical sync     |

## A.2 Screen Resolution

Screen resolution depends on several factors, including the monitor timing, the VRAM size and speed, and the pixel width and height. The maximum possible screen resolution is determined by:

- ☐ Pixel clock frequency
- ☐ Hardware constraints
- ☐ Size of the VRAM

To determine if the resolution that you want is supported by the VRAM, you must determine the pixel size. The pixel size is often a function of the palette and/or the RAMDAC that interfaces with the monitor.

To determine the required frame memory size, multiply the pixel size by the number of pixels per screen. For example, two megabytes of VRAM memory supports the following screen resolutions:

- ☐ 1024-by-512 pixels at 32 bits per pixel
- ☐ 1024-by-1024 pixels at 16 bits per pixel
- ☐ 1280-by-1024 pixels at 8 bits per pixel

In the following example, a 1024 × 768 display is used.

## A.3 Noninterlaced Monitor Specifications

Figure A–3 and Table A–1 show the specifications for a typical monitor. This information is usually provided in your monitor's user's manual.

Figure A–3. Noninterlaced Monitor Timing (Separate Sync)

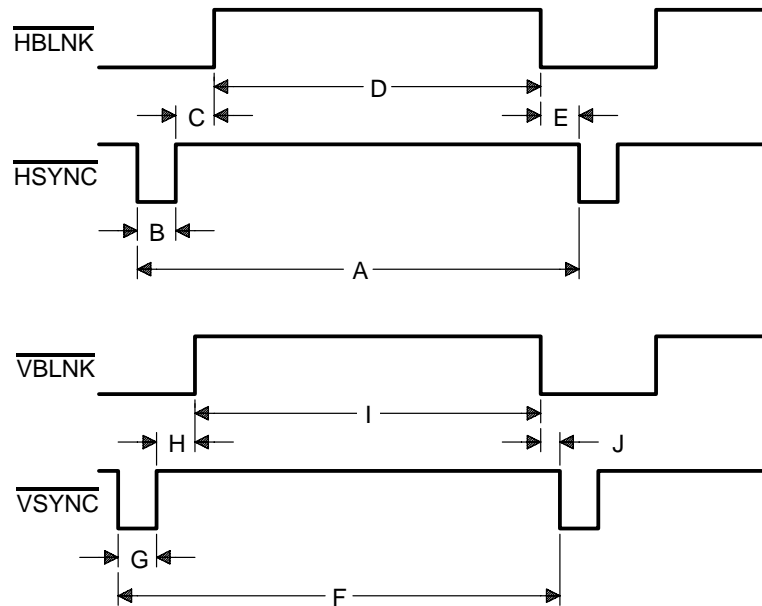


Table A–1. Typical Noninterlaced Monitor Timing

Line rate			at 48.5 kHz
A	LD	Line duration	20.625 $\mu$ s
B	HS	Horizontal sync	1.0 $\mu$ s
C	HBP	Horizontal back porch	2.875 $\mu$ s
D	HAT	Horizontal active time	16.0 $\mu$ s
E	HFP	Horizontal front porch	0.75 $\mu$ s
Frame rate			at 60 Hz
F	FD	Frame duration	16.665 ms
G	VS	Vertical sync	83 $\mu$ s
H	VBP	Vertical back porch	660 $\mu$ s
I	VAT	Vertical active time	15.84 ms
J	VFP	Vertical front porch	82 $\mu$ s



## A.4 Pixel Clocks

There are two clocks of interest in video timing: the serial clock (SCLK) of the VRAM and the pixel clock. SCLK is normally a fraction (external divide down) of the pixel clock. The amount of division depends on the palette size and speed and the bus interface. For example, if the bus between the VRAMs and the palette is 32 bits wide and you are using 16-bit pixels, the frequency of SCLK is the frequency of the pixel clock divided by 2, since 2 pixels are transferred per SCLK cycle.

The pixel clock rate is influenced by the amount of horizontal and vertical blanking needed to conform to the monitor specifications. Since blanking is controlled by the frame timer, this becomes a function of the frame clock (FCLK). FCLK is normally a fraction (external divide down) of the pixel clock, because pixel clock frequency is higher than the allowable FCLK frequency (50 MHz). The amount of divide down will affect the line length. For example, if the divide down circuitry is such that  $FCLK = \text{pixel clock} / 8$ , the displayed line can be only a multiple of 8 pixels in length. Use the following horizontal timing information to determine the appropriate FCLK frequency:

$$\text{horizontal active time} = LD - HS - HBP - HFP = 16.000 \mu\text{s}$$

At 1024 pixels per line, this corresponds to 15.625 ns per pixel.

In this example, assume that the external divide down circuitry is such that FCLK equals the pixel clock divided by 4. To support the required pixel rate, the pixel clock must be 64 MHz ( $1/15.625 \text{ ns}$ ); therefore, FCLK is 16 MHz.

## A.5 Programming the Frame Timer Registers for Noninterlaced Video

Table 4–1, *Programming the Video Timing Registers for Noninterlaced Video*, on page VC:4-21 summarizes all the VC frame timer register configurations for noninterlaced video. The values loaded into the horizontal timing registers represent an integral number of FCLK cycles.

VCOUNT increments only after HCOUNT = HTOTAL for noninterlaced video. Therefore, all the vertical timing registers, except VESYNC, are programmed in terms of an integral number of horizontal lines. For purposes associated with composite interlaced video, the VESYNC register detects the end of vertical sync at half the value it is programmed to and, therefore, represents twice the number of horizontal lines in vertical sync. You should program VESYNC as an odd number.

Since all of the timing registers begin at 0, their contents represent the number of appropriate events minus 1.

Follow the steps in Table A–2 to configure the frame timer registers for noninterlaced video. Refer to Table A–1 for the noninterlaced monitor timings. Recall that

$$\text{FCLK} = 16 \text{ MHz} \quad (T_{\text{FCLK}} = 62.5 \text{ ns})$$

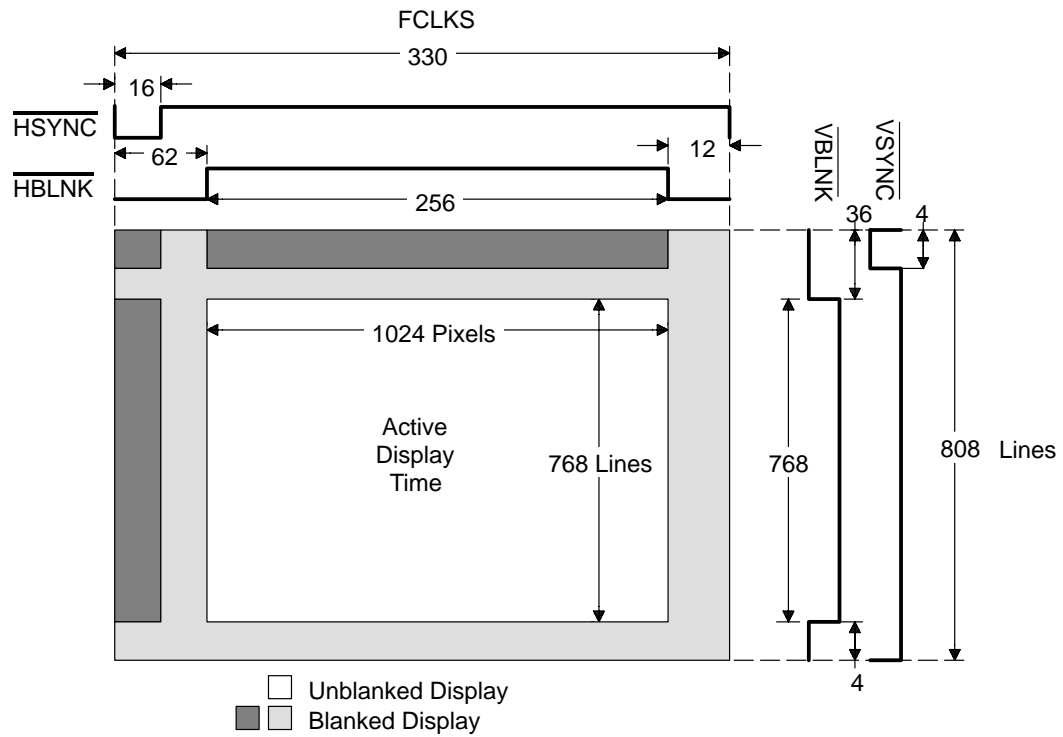
Table A–2. Configuring the Values of the Frame Timer Registers for a Noninterlaced 1024 × 768 Display

	To calculate	Use the formula	Result†	Register value
1	Number of FCLKs per line	$\frac{LD}{T_{FCLK}}$	330	HTOTAL = Result – 1 = 0x0149
2	Number of FCLKs in horizontal sync	$\frac{HS}{T_{FCLK}}$	16	HESYNC = Result – 1 = 0x000F
3	Number of FCLKs in horizontal back porch	$\frac{HBP}{T_{FCLK}}$	46	N/A
4	Number of FCLKs from the start of horizontal sync to the end of horizontal blank	Add the results from steps 2 and 3	62	HEBLNK = Result – 1 = 0x003D
5	Number of FCLKs in horizontal front porch	$\frac{HFP}{T_{FCLK}}$	12	N/A
6	Number of FCLKs from the start of horizontal sync to the start of horizontal blank	Subtract the result of step 5 from step 1	318	HSBLNK = Result – 1 = 0x013D
7	Number of lines per frame	$\frac{FD}{LD}$	808	VTOTAL = Result – 1 = 0x0327
8	Number of lines in vertical sync	$\frac{VS}{LD}$	4	VESYNC = 2 × Result – 1 = 0x0007
9	Number of lines in vertical back porch	$\frac{VBP}{LD}$	32	N/A
10	Number of lines from the start of vertical sync to the end of vertical blank	Add the results from steps 8 and 9	36	VEBLNK = Result – 1 = 0x0023
11	Number of lines in vertical front porch	$\frac{VFP}{LD}$	4	N/A
12	Number of lines from the start of vertical sync to the start of vertical blank	Subtract the result of step 11 from step 7	804	VSBLNK = Result – 1 = 0x0323

† The results have been rounded to the nearest whole number.

Figure A–4 illustrates the horizontal and vertical timing relationship defined by the results listed in Table A–2.

Figure A–4. Horizontal and Vertical Timing Relationship for Noninterlaced Video



- Notes:** 1) SRT events are controlled by blanking.  
2) No vertical interrupt is programmed.

## A.6 Programming the SRT Controller Registers

Programming the SRT controller registers requires some information about the associated hardware that you are using. To program these registers, you must determine the memory address of the first pixel displayed and the difference in the memory addresses between two vertically adjacent pixels. This result should be placed in the LINEINC0 register. The address of the first pixel displayed will most often be the base address of the VRAM. This address should be programmed into the F0STADR0, F1STADR0, NEXTADR0, and CRNTADR0 registers.

The NEXTADR0 and CRNTADR0 registers are automatically updated by the VC during display, but you should program them before display to ensure that the frame memory will not be corrupted. The SAMMASK0 and FMEMCTL0 registers should be programmed as dictated by the structure of the VRAM that you are using. For example, note how you should program these registers for the following conditions:

### ☐ SAMMASK0

- The VRAM has a row length of 512 bits with a SAM length of 256 bits.

The split SAM length is, therefore, 128 bits, or seven contiguous 1s. The placement of these seven contiguous 1s in the SAMMASK0 register depends on which address pins the VRAM is wired to when  $\overline{\text{CAS}}$  goes low.

- The VRAM is physically connected to A(11:3) at column time.

The least significant 1 in SAMMASK corresponds to the least significant bit of the address bus wired to the VRAM (bit 3 in this example). Therefore, SAMMASK should be programmed to 0x0000 03F8 (127 shifted left by 3). For more information on column time, refer to subsection 7.4.1, *Address Multiplexing*, in the *MVP Transfer Controller User's Guide*.

---

### **Note:**

In notation used in this section, ( ) (parentheses) indicate bit locations and a : (colon) indicates a range. For example, A(12) refers to bit 12 of address A and A(20:12) refers to bits 20 to 12 of address A.

---

## □ FMEMCTL0

The FMEMCTL0 register controls how and when data is transferred to and from the VRAM. In this example, SRT events are scheduled at the start of the field, during line events, and when the SAM overflows (EMS = 00). Line events are controlled by blanking (UED = 0) and are scheduled three FCLKs after blanking begins (HBLINE = HSBLNK + 3).

In this example, FMEMCTL is set to 0x0000 4001, which enables frame timer 0 and sets the half SAM select (HSS) bit to 1. See page VC:5-8 for a description of the FMEMCTL register.

## A.7 Programming the Horizontal Line Event Registers

The horizontal area line event (HALINE) register schedules line events that are controlled by the active area. The horizontal blanking line event (HBLINE) register schedules line events controlled by blanking. Selection between the two registers is dictated by the unblanked event disable (UED) bit in the FMEMCTL register. In display mode, line events cause a full SAM read, scheduled when HCOUNT = HALINE or HBLINE.

In general, you should set HALINE or HBLINE equal to, or slightly larger than, HEAREA or HSBLNK. This ensures that the SAM reload does not occur during the active portion of the line.

## A.8 NTSC Composite Interlaced Video Display

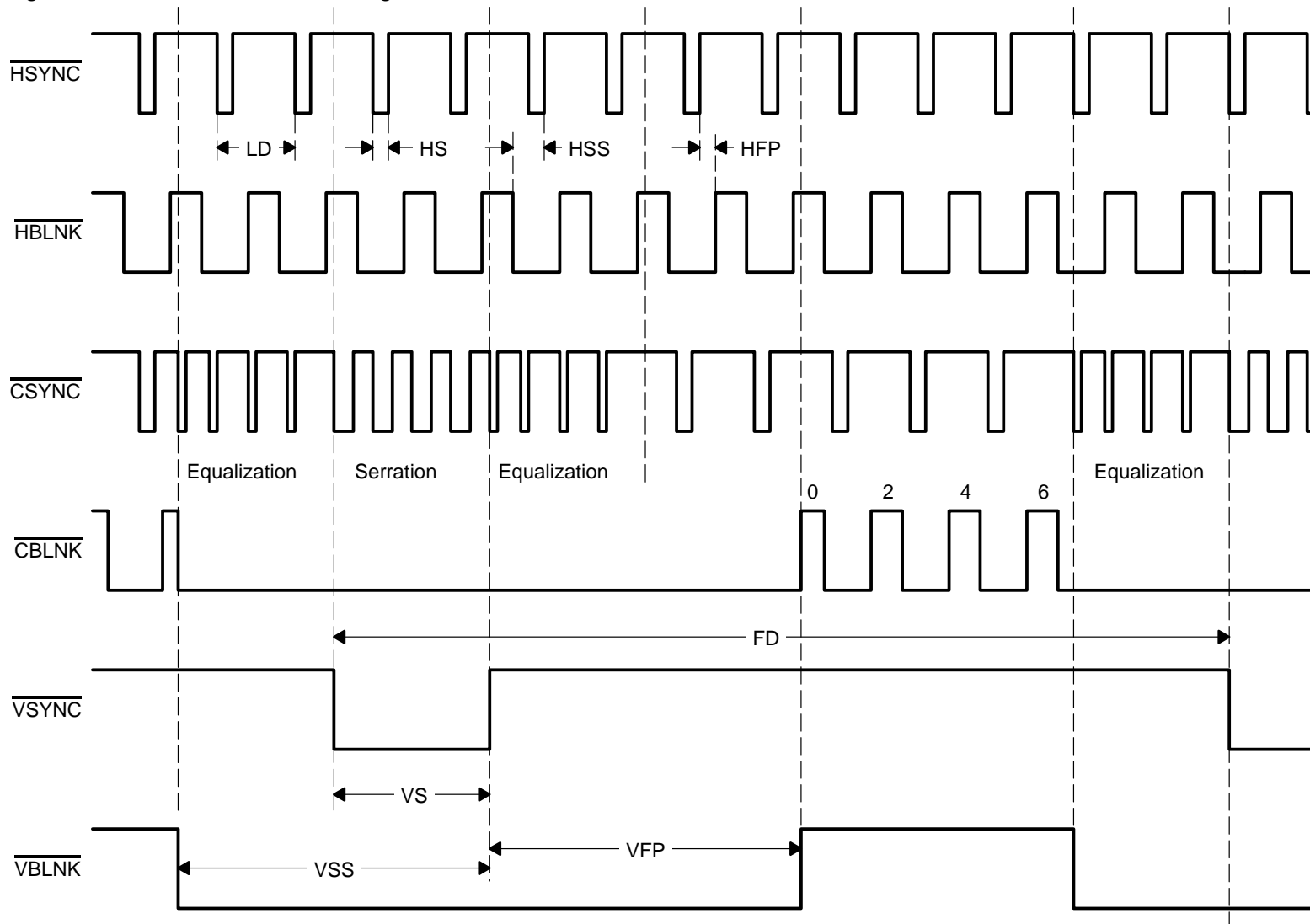
In interlaced video, program the frame timer registers as shown in Table 4–2, *Programming the Video Timing Registers for Interlaced Video*, on page VC:4-32.

In the example in Figure A–5, the frame timer registers are configured for interlaced video, with the  $\overline{\text{CSYNC}}/\overline{\text{HBLNK}}$  and  $\overline{\text{CBLNK}}/\overline{\text{VBLNK}}$  signals configured as  $\overline{\text{CSYNC}}$  and  $\overline{\text{CBLNK}}$ , respectively (composite video mode). The composite  $\overline{\text{CSYNC}}$  signal combines both horizontal and vertical timing information into one signal that is more complex than either  $\overline{\text{HSYNC}}$  or  $\overline{\text{VSYNC}}$ . During the active portion of a frame, the  $\overline{\text{CYSNC}}$  signal looks exactly like the  $\overline{\text{HSYNC}}$  signal. However, in interlaced mode, the vertical blanking and sync signals produce special pulses on  $\overline{\text{CSYNC}}$  during regions of equalization and serration.

$\overline{\text{CBLNK}}$  represents the negative logic (logical OR) of the  $\overline{\text{HBLNK}}$  and  $\overline{\text{VBLNK}}$  signals. The composite signals and equalization and serration regions are shown in Figure A–5.



Figure A-5. Interlaced Video Timing—Even Field



**Note:** SP is not shown.

Table A–3. NTSC Timings

## (a) Line Rates

Line rate		at 15.7 kHz
LD	Line duration	63.556 $\mu$ s
HS	Horizontal sync	4.7 ( $\pm 0.1$ ) $\mu$ s
HSS <sup>†</sup>	Horizontal sync to setup	9.4 ( $\pm 0.1$ ) $\mu$ s
SP	Subcarrier period	(2/455) $\times$ LD
HFP	Horizontal front porch	1.5 ( $\pm 0.1$ ) $\mu$ s

<sup>†</sup> SS (sync to setup) is an NTSC-defined parameter and is equal to sync plus the back porch.

## (b) Frame Rates

Frame rate		at 60 Hz
FD	Frame duration	525 lines
VS	Vertical sync	3 lines
VSS <sup>†</sup>	Vertical sync to setup	17 lines
VFP	Vertical front porch	3 lines

<sup>†</sup> SS (sync to setup) is an NTSC-defined parameter and is equal to sync plus the back porch.

As a result of the equalization and serration regions, there are several constraints to programming the frame timer registers:

- ☐ The horizontal sync duration must be an even number of FCLKs in length so that the equalization pulses are exactly half that duration. Therefore, HESYNC should be programmed to an odd value.
- ☐ NTSC requires that the serration pulses be of a duration such that the inactive (high) period of  $\overline{\text{CSYNC}}$  is equal to the active (low) period of horizontal sync.

## A.9 Programming the Frame Timer Registers for Interlaced Video

Before you can program the frame timer registers, you must determine the FCLK period. In NTSC video, it is important that the subcarrier frequency (which carries the color information) and the FCLK be harmonically related. The number of subcarrier cycles per scan line is:

$$\frac{LD}{SP} = 227.5$$

There must be an integral number of FCLK periods per line; therefore, FCLK must be at least twice the frequency of the subcarrier. This results in 455 FCLKs per line. However, equalization and serration pulses require that there also be an even number of FCLK cycles per line; therefore, FCLK must be at least four times the subcarrier frequency or 910 FCLKs per line.

$$\text{FCLK period (T}_{\text{FCLK}}) = \frac{LD}{910} = 69.84 \text{ ns and}$$

$$\text{FCLK frequency} = 14.318 \text{ MHz}$$

Follow the steps in Table A–4 to program the frame timer registers for interlaced video:

Table A-4. Configuring the Values of the Frame Timer Registers for Interlaced Video

	To calculate	Use the formula	Result†	Register value
1	Number of FCLKs per line	$\frac{LD}{T_{FCLK}}$	910	HTOTAL = Result – 1 = 0x038D‡
2	Number of FCLKs in horizontal sync	$\frac{HS}{T_{FCLK}}$	67	HESYNC = 0x0043‡
3	Number of FCLKs in horizontal serration	$\frac{HTOTAL - 1}{2} - (HESYNC + 1)$	386	HESERR = 0x0182
4	Number of FCLKs from the start of horizontal sync to the end of horizontal blank	$\frac{HSS}{T_{FCLK}}$	135	HEBLNK = Result – 1 = 0x0086
5	Number of FCLKs in horizontal front porch	$\frac{HFP}{T_{FCLK}}$	21	N/A
6	Number of FCLKs from the start of horizontal sync to the start of horizontal blank	Subtract the result of step 5 from step 1	889	HSBLNK = Result – 1 = 0x0378
7	Twice the number of halflines per field	$2 \times \frac{FD}{2}$	525	VTOTAL = Result – 1 = 0x020C
8	Number of halflines in vertical sync	$2 \times 2 \times VS$	12	VESYNC = Result – 1 = 0x000B‡
9	Number of halflines from the start of vertical sync to the start of vertical blank	$VTOTAL - VFP \times 2$	519	VSBLNK = Result – 1 = 0x0206
10	Number of halflines from the start of vertical sync to the end of vertical blank	$2 \times (VSS + VS)$	40	VEBLNK = Result – 1 = 0x0027

† The results have been rounded to the nearest whole number.

‡ You must program this register to an odd value.

## A.10 Frame Timer Interrupts

Each of the two frame timers has its own interrupt, which you may use to synchronize the master processor (MP) with the display. One of the most common uses of a frame timer interrupt is to control double buffering for animation.

In this application, there are two sections (buffers) of frame memory. Each buffer is large enough to hold one complete frame. While the frame in the first buffer is being displayed, the MVP can process data in the second buffer. The interrupt is set to occur on a line in the vertical front porch (that is, a line in blanked display).

When display from the first buffer is complete and the interrupt occurs, the MP interrupt routine reprograms the NEXTADR, CRNTADR, F0STADR, and F1STADR registers to point to the second buffer. Therefore, the next frame displayed will be from the updated buffer, and the MVP can start updating the first buffer. When the interrupt occurs again (after displaying the second buffer), the MP switches back to the first buffer. In this manner, the active display memory switches between the two buffers.

The frame timer interrupts are internal interrupts to the MP and are posted in the MP's interrupts pending (INTPEN) register. Bits 8 and 9 of the INTPEN register indicate that a frame timer interrupt is pending from frame timer 0 and frame timer 1, respectively. Bit 0 and bits 8 and/or 9 in the interrupt enable (IE) register, must be set in order for the service routine to be executed. For more information on frame timer interrupts, refer to Chapter 9, *Interrupts, Traps, and Reset*, in the *MVP Master Processor User's Guide*.

# Glossary

---

---

---

---

## A

**assembler:** A software utility that creates a machine-language program from a source file. There are two assemblers associated with the MVP: a mnemonic-based RISC-type assembler for the MP and an algebraic assembler for the PP.

## B

**back porch:** The interval between the end of synchronization and the corresponding blanking pulse. The horizontal back porch is specified as an integral number of FCLK periods; the vertical back porch is specified as an integral number of lines (half-lines for interlaced mode).

**big endian:** An addressing protocol in which bytes are numbered from left to right within a word. More significant bytes in a word have lower numbered addresses. Endian ordering is specific to hardware and is determined at reset. See also *little endian*

**blanking pulse:** A positive or negative pulse developed during retrace, appearing at the end of each field. Used to blank out scanning lines during the vertical or horizontal retrace interval.

**C**

**capture mode:** A serial-register-transfer (SRT) mode during which an image is captured and stored into memory. Memory locations not corresponding to the captured image may be overwritten. See also *display mode*, *merge mode*

**CAREA:** *composite area.* The signal generated by the frame timers that can be used to define a special area, such as an overscan boundary. CAREA acts identically in both interlaced and non-interlaced modes, defining a purely rectangular region.

**$\overline{\text{CAS}}$ :** *Column address strobe.* A memory interface signal that drives the column address strobe inputs of DRAMs/VRAMs.

**$\overline{\text{CBLNK}}$ :** *Composite blanking.* A signal that combines information about the horizontal and vertical timing intervals into one signal that is more complex than either  $\overline{\text{HBLNK}}$  or  $\overline{\text{VBLNK}}$ .  $\overline{\text{CBLNK}}$  can be used to disable pixel capture or display during horizontal and vertical retrace. See also *blanking pulse*

**COLMASK:** *Column mask.* A 32-bit control mask used by the SRT controller for address calculation.

**composite area:** See *CAREA*

**composite blanking:** See  $\overline{\text{CBLNK}}$

**composite synchronization:** See  $\overline{\text{CSYNC}}$

**crossbar:** A generally configurable, high-speed bus switching network for a multiprocessor system, permitting any of several processors to connect to any of several memory modules.

**$\overline{\text{CSYNC}}$ :** *Composite synchronization.* A signal that combines information about the horizontal and vertical timing intervals into one signal that is more complex than either  $\overline{\text{HSYNC}}$  or  $\overline{\text{VSYNC}}$ .  $\overline{\text{CSYNC}}$  can be used to enable retrace of the electron beam of a display screen that supports composite video.

**D**

**data cache:** The MP's two SRAM banks that hold cached data needed by the MP. Data RAMs for the PPs are not cached.

**data RAM:** On-chip RAM that is available for the general-purpose storage of data by the MP or PPs on the MVP.

**debugger:** A window-oriented software interface that helps you to debug MVP programs running on an MVP emulator or simulator.

**display mode:** A serial-register-transfer (SRT) mode during which information is transferred from the frame memory to the display device. See also *capture mode*, *merge mode*

**doubleword:** A 64-bit value.

## E

**executive:** The portion of a multitasking software system that is responsible for executing application tasks, providing communications among tasks, and managing shared resources.

**external address:** See *off-chip address*

**externally initiated packet transfer:** See *XPT*

## F

**FCLK:** *Frame clock.* The clock that controls the internal video logic of the VC's frame timers.

**frame:** The screen image output during a single vertical sweep.

**frame lock:** VC mode that allows the two frame timers to be locked together so that they are synchronous. This is useful when different timing signal parameters are required in the capture and display systems, but where it is otherwise desirable to coordinate the two.

**frame timers:** In the VC, timers that provide video timing control. The frame timers indicate to the serial-register-transfer (SRT) controller when an SRT is necessary.

**front porch:** The interval between the start of blanking and the corresponding sync pulse. The horizontal front porch is specified as an integral number of FCLK periods; the vertical front porch is specified as an integral number of lines (half-lines for interlaced mode).

## H

**halfword:** A 16-bit value.

**$\overline{\text{HBLNK}}$ :** *Horizontal blanking.* A bidirectional horizontal timing signal that enables or disables pixel capture and display.  $\overline{\text{HBLNK}}$  occurs once per line and has a pulse width defined as an integral number of FCLK periods. See also *blanking pulse*



**horizontal blanking:** See  $\overline{HBLNK}$

**horizontal synchronization:** See  $\overline{HSYNC}$

**$\overline{HSYNC}$ :** *Horizontal synchronization.* A bidirectional horizontal timing signal occurring once per line and with a pulse width defined as an integral number of FCLK periods. Synchronization signals can be used to enable retrace of the electron beam of a display screen.

## I

**instruction cache:** An on-chip SRAM that contains current instructions being executed by one of the MVP processors. Cache misses are handled by the transfer controller.

**interlaced mode:** A mode in which each frame consists of two vertical fields. One field displays odd horizontal lines, and the other field displays even horizontal lines. In effect, the number of transmitted pictures is doubled, thus reducing flicker.

**internal address:** See *on-chip address*

**interrupt:** An exceptional condition caused either by an event external to the processor or by a previously executed instruction that forces the current program to be interrupted. After the processor has serviced the interrupt, it typically resumes execution of the interrupted program at the instruction whose execution was interrupted.

## J

**JPEG standard:** *Joint Photographic Experts Group standard.* A standard used for compressed still-picture data.

## L

**linker:** A software tool that combines object files to form an object module that can be allocated into system memory and executed by the device.

**little endian:** An addressing protocol in which bytes are numbered from right to left within a word. More significant bytes in a word have higher numbered addresses. Endian ordering is specific to hardware and is determined at reset. See also *big endian*

**local bus:** A 32-bit bus allocated to each PP to access on-line SRAM data in a single cycle.

**LSB:** *Least significant bit.* The bit having the smallest effect on the value of a binary numeral, usually the rightmost bit. The MVP numbers the bits in a word from 0 to 31, where bit 0 is the LSB.

## M

**master processor:** See *MP*

**memory map:** A map of target system memory space that is partitioned into functional blocks.

**merge mode:** A serial register transfer (SRT) mode for the VC, during which an image is captured and stored in memory. Memory locations not corresponding to the captured image are preserved. See also *capture mode*, *display mode*

**MP:** *Master processor.* A general-purpose RISC processor that coordinates the activity of the other processors on the MVP. The MP includes an IEEE-754 floating-point hardware unit.

**MPEG standard:** *Moving Picture Experts Group standard.* A proposed standard for compressed video data.

**MSB:** *Most significant bit.* The bit having the greatest effect on the value of a binary numeral. It is the leftmost bit. The MVP numbers the bits in a word from 0 to 31, where bit 31 is the MSB.

**multimedia video processor:** See *MVP*

**MVP:** *Multimedia video processor.* A single-chip multiprocessor device that accelerates applications such as video compression and decompression, image processing, and graphics. The multimedia video processor contains a master processor and from one to eight parallel processors, depending on the device version. For example, the TMS320C80 device contains four PPs.

**MVP multitasking executive:** See *executive*

## N

**noninterlaced mode:** A mode for the VC in which each frame consists of a single vertical field.

## O

**off-chip address:** An address external to the MVP chip. Addresses from 0x0200 0000 to 0xFFFF FFFF are off-chip addresses. See also *on-chip address*

**on-chip address:** An address internal to the MVP chip. Addresses from 0x0000 0000 to 0x1FFF FFFF are on-chip addresses. See also *off-chip address*

## P

**packet:** A collection of patches of data.

**packet transfer:** See *PT*

**packet transfer request:** An I/O request submitted to the TC that is issued when a block of data is to be moved via packet transfer. Packet transfer requests can be submitted by the MP, the PPs, the VC, or an external device.

**parallel processor:** See *PP*

**parameter RAM:** A general-purpose 2K-byte RAM that is associated with a specific processor, part of which is dedicated to packet transfer information and the processor interrupt vectors.

**parameter table:** A group of parameters, eight doublewords long, that describe a data packet and how it is to be moved from source to destination.

**patch:** A group of lines of equal length whose starting addresses are an equal distance apart.

**poll:** A continuous test used by the program until a desired condition is met.

**porch:** The portion of a video display signal that corresponds to the blanking interval on either side of a horizontal or vertical synchronization pulse. The terms front porch and back porch refer to the blanking intervals that precede and follow, respectively, the sync pulse. See also *back porch*, *front porch*

**PP:** *Parallel processor.* The MVP's advanced digital signal processor that is used for video compression/decompression ( $P \times 64$  or MPEG), still-image compression/decompression (JPEG), 2-D and 3-D graphic functions such as line draw, trapezoid fill, antialiasing, and a variety of high-speed integer operations on image data. An MVP single-chip multiprocessor device may contain from one to eight PPs, depending on the device version.

**PT:** *Packet transfer.* A transfer of data blocks between two areas of memory. The MVP supports packet transfers of one, two, or three dimensions.

**PT options field:** Packet-transfer parameter field which selects the form of transfer for source and destination. It determines if the packet will end the linked list and enables the selection of additional features such as special transfer modes.

## R

**$\overline{\text{RAS}}$ :** *Row address strobe.* A memory interface signal that drives the row address strobe inputs of DRAMs/VRAMs.

**retrace:** A line traced by the scanning beam(s) of a display screen as it travels from the end of one horizontal line (vertical field) to the beginning of the next horizontal line (vertical field).

**RISC:** *Reduced instruction set computer.* A computer whose instruction set and related decode mechanism are much simpler than those of microprogrammed complex instruction set computers. The result is a higher instruction throughput and a faster real-time interrupt response from a smaller, cost-effective chip.

## S

**SAM:** *Serial access memory.* Memory array in VRAMs that can be accessed via serial register transfer cycles. See *SRT controller, VRAM*

**SAM overflow event:** In the VC, a serial access memory overflow event generated by the SRT controller that determines when cycles need to be performed to service the inactive half of the split SAM.

**SAMINC:** *Serial access memory increment.* A 32-bit control mask used by the SRT controller for address calculation. This mask contains a 1 in the bit position above the most significant 1 in the SAMMASK register.

**SAMMASK:** *Serial access memory mask.* An SRT controller register that contains a string of continuous 1s. The number and positions of these 1s depend on the VRAM SAM width and the address lines connected to the VRAM.

**SCLK:** *Serial clock.* An input clock signal used by the SRT controller to track the VRAM tap point.

**serial register transfer controller:** See *SRT controller*

**SRT controller:** *Serial register transfer controller.* Hardware that schedules requests to the transfer controller to move data into and out of VRAM frame memories.

## T

**tap point:** Address of the point at which data is shifted into or out of the VRAM's serial I/O port. The SRT controller tracks the VRAM tap point using the SCLK input.

**TC:** *Transfer controller.* The MVP's on-chip DMA controller for servicing the cache and for transferring one-, two-, and three-dimensional data blocks between each processor on the MVP and its external memory.

**transfer controller:** See *TC*

## V

**$\overline{\text{VBLNK}}$ :** *Vertical blanking.* Bidirectional vertical timing signals that occur once per frame (once per field for interlaced systems) and have a pulse width defined as an integral number of lines (halflines for an interlaced system).  $\overline{\text{VBLNK}}$  can be used to disable pixel capture and display during vertical retrace. See also *blanking pulse*

**VC:** *Video controller.* The portion of the MVP responsible for the video interface.

**vertical blanking:** See  $\overline{\text{VBLNK}}$

**vertical synchronization:** See  $\overline{\text{VSYNC}}$

**video controller:** See *VC*

**VRAM:** *Video random access memory.* A type of memory that is easily interfaced to a video display.

**$\overline{\text{VSYNC}}$ :** *Vertical synchronization.* A bidirectional vertical timing signal occurring once per frame with a pulse width defined as an integral number of lines (half-lines for interlaced mode).

**W**

**word:** A sequence of 32 adjacent bits that constitutes a register or memory value. The PP supports 32-bit words. The MP also supports doublewords of 64 bits for loads and stores.

**X**

**XPT:** *Externally initiated packet transfer.* A packet transfer initiated by an external device through the MVP's  $\overline{\text{XPT}}$  [2:0] inputs.



# Index

---

---

---

## A

addresses  
  frame timer 0 VC:1-6  
  frame timer 1 VC:1-6  
  FTCTL register VC:3-2  
  horizontal timing registers VC:3-2  
  SRT controller VC:1-8  
  vertical timing registers VC:3-3  
American television standard (NTSC)  
  VC:4-28  
  example VC:4-30  
  vertical end area programming  
    VC:4-36 to VC:4-38  
  vertical frame timer interrupt  
    programming VC:4-39 to  
    VC:4-41  
  vertical start area programming  
    VC:4-33 to VC:4-35

## B

blanking VC:1-4  
  composite signals VC:2-7  
  horizontal timing signals VC:2-4  
  vertical timing signals VC:2-5  
block diagram  
  MVP VC:1-2  
  VC VC:1-3  
bus  
  data VC:1-5

## C

CAREA signals VC:2-6  
CAS VC:5-15  
CBLNK/VBLNK signals VC:2-5,  
  VC:2-7  
  CBLNK signal  
    composite video timing VC:4-9  
    selecting VC:2-5  
  vertical retrace VC:2-5  
  with HBLNK VC:2-5  
clock  
  *See also* FCLK; SCLK  
  pixel VC:A-6  
  serial clock input  
    signal descriptions VC:2-8  
COLMASK. *See* SRT controller, control  
  masks  
composite signals  
  area VC:2-6  
  blanking VC:2-7  
  description VC:2-6  
  synchronization VC:2-6  
    equalization pulses VC:4-10 to  
    VC:4-11, VC:4-12  
    serration pulses VC:4-10 to  
    VC:4-11, VC:4-13 to VC:4-14  
composite video timing VC:4-9  
control mask. *See* SRT controller,  
  control masks  
CPM bits. *See* FTCTL register, CPM  
  bits



CRNTADR register VC:5-4  
bits determined by SAMMASK  
VC:5-15

$\overline{\text{CSYNC}}$  signal  
compatibility with standards  
VC:2-6  
composite video timing VC:4-9  
configuring VC:2-6  
equalization pulses VC:4-12  
serration pulses VC:2-6,  
VC:4-13 to VC:4-14

$\overline{\text{CSYNC}}/\overline{\text{HBLNK}}$  pin mode. See FTCTL  
register, CPM bits

$\overline{\text{CSYNC}}/\overline{\text{HBLNK}}$  signals VC:2-4,  
VC:2-6  
configuring VC:2-4

current address register. See  
CRNTADR register

## D

data  
bus VC:1-5  
disabling  
capture/display VC:2-4, VC:2-5,  
VC:2-7

## E

EMS bits. See FMEMCTL register,  
EMS bits  
equalization pulses VC:4-10 to  
VC:4-11  
on  $\overline{\text{CSYNC}}$  VC:4-12  
European television standard (PAL)  
VC:4-28  
example VC:4-31  
vertical end area programming  
VC:4-38  
vertical interrupt programming  
VC:4-41  
vertical start area programming  
VC:4-34 to VC:4-35

event mode select bits. See  
FMEMCTL register, EMS bits  
events. See SRT controller, events  
external synchronization. See synchro-  
nization, external

## F

F0STADR register VC:5-5 to VC:5-6  
interlaced mode VC:5-6  
noninterlaced mode VC:5-5

F1STADR register VC:5-7  
calculating F0STADR VC:5-6  
interlaced mode VC:5-7  
noninterlaced mode VC:5-7

FCLK VC:1-4  
determining frequency VC:A-6  
input VC:2-3

field 0 start address register. See  
F0STADR register

field 1 start address register. See  
F1STADR register

FLE bit. See FTCTL register, FLE bit

FMEMCTL register VC:5-8 to  
VC:5-12  
EMS bits VC:5-10 to VC:5-11  
FTS bits VC:5-9  
HSS bit VC:5-12  
ILR bit VC:5-6, VC:5-12  
PTS bit VC:5-12  
TMS bits VC:5-11  
UED bit VC:5-5, VC:5-6, VC:5-9

frame clock. See FCLK

frame control VC:1-4

frame lock VC:4-53  
enable bit. See FTCTL register, FLE  
bit

frame memory  
control register. See FMEMCTL  
register  
data transfer VC:1-4  
determining size VC:A-4  
registers VC:1-8

- frame timer
  - See also* interlaced, video timing;
  - noninterlaced, video timing
  - composite synchronization VC:3-6
  - composite video VC:4-9
  - control register. *See* FTCTL register
  - enable VC:3-7
  - external synchronization
    - VC:4-44 to VC:4-49
  - frame lock. *See* frame lock
  - frame timer 0
    - description VC:1-4
    - signals VC:2-2
  - frame timer 1
    - description VC:1-4
    - signals VC:2-2
  - horizontal blanking VC:3-6
  - horizontal synchronization VC:3-5
  - interrupts. *See* interrupts; VFTINT register
  - registers
    - addresses VC:3-2 to VC:3-3
    - descriptions VC:3-4 to VC:3-28
    - overview VC:3-1
    - programming for interlaced video VC:A-16
    - programming for noninterlaced video VC:A-7
    - summary VC:3-2
  - sequencer bits. *See* FMEMCTL register, FTS bits
  - setting synchronization enable VC:3-7
  - signals VC:2-2
    - horizontal. *See* horizontal timing, signals
    - vertical. *See* vertical timing, signals
  - SRT requests VC:5-2
  - synchronization delay compensation VC:4-50 to VC:4-51
  - vertical synchronization VC:3-5
- FTCTL register VC:3-4 to VC:3-9
  - addresses VC:3-2
  - composite blanking signals VC:2-7
  - configuring  $\overline{\text{CSYNC}}$  VC:2-6
  - configuring  $\overline{\text{HSYNC}}$  VC:2-4
  - configuring  $\overline{\text{VSYNC}}$  VC:2-5
  - CPM bits VC:3-6
    - configuring  $\overline{\text{CBLNK}}$  VC:2-7
    - configuring  $\overline{\text{CSYNC}}$  VC:2-6
  - horizontal blanking VC:2-4
  - vertical blanking VC:2-5
  - description VC:3-5
  - fields VC:3-4
  - FLE bit VC:3-6
  - FTE bit VC:3-7
  - HPM bits VC:3-5
    - horizontal synchronization VC:2-4
  - IFD bit VC:3-7
  - IIM bit VC:3-7
  - SSE bit VC:3-7
  - VPM bits VC:3-5
  - writing to upper half VC:1-7
- FTE bit. *See* FTCTL register, FTE bit
- FTS bits. *See* FMEMCTL register, FTS bits
- functional blocks VC:1-4

## H

- half SAM select bit. *See* FMEMCTL register, HSS bit
- HALINE register VC:3-8
  - programming VC:A-12
- HBLINE register VC:3-9
  - programming VC:A-12
- $\overline{\text{HBLNK}}$  signals VC:4-2
  - repetition at line intervals VC:2-4

- HCOUNT register VC:3-10
    - FCLK periods VC:4-4
    - horizontal video timing VC:4-4 to VC:4-5
  - HEAREA register VC:3-11
  - HEBLNK register VC:3-12
  - HESERR register VC:3-13
  - HESYNC register VC:3-14
  - high-impedance capability VC:4-52
  - horizontal blanking VC:2-4
  - horizontal timing
    - registers VC:1-7, VC:3-2
      - See also* registers, horizontal timing
    - addresses VC:3-2
    - signals
      - blanking VC:2-4, VC:4-5
      - description VC:2-4
      - interlaced VC:4-4
      - noninterlaced VC:4-4
      - synchronization VC:2-4
      - understanding VC:4-2
      - video VC:4-4
  - HPM bits. *See* FTCTL register, HPM bits
  - HSAREA register VC:3-15
  - HSBLNK register VC:3-16
  - HSS bit. *See* FMEMCTL register, HSS bit
  - HSYNC pin mode bits. *See* FTCTL register, HPM bits
  - HSYNC signals VC:4-2
    - configuration VC:2-4
    - description VC:2-4
  - HTOTAL register VC:3-17
- I**
- IFD bit. *See* FTCTL register, IFD bit
  - IIM bit. *See* FTCTL register, IIM bit
  - ILR bit. *See* FMEMCTL register, ILR bit
  - interlaced
    - frame control VC:1-4
    - frame disable bit. *See* FTCTL register, IFD bit
    - interrupt mode bit. *See* FTCTL register, IIM bit
    - line repeat bit. *See* FMEMCTL register, ILR bit
    - vertical timing VC:4-8 to VC:4-9
    - video timing VC:4-22 to VC:4-25
      - horizontal operation VC:4-4, VC:4-25
      - vertical operation VC:4-28 to VC:4-29
  - interlaced composite mode
    - frame timer configuration VC:3-10, VC:3-14, VC:3-17
  - interlaced mode
    - equalization pulses VC:4-12
    - frame timer configuration VC:3-20
    - registers for configuring
      - VEAREA VC:3-21
      - VEBLNK VC:3-22
      - VESYNC VC:3-23
      - VFTINT VC:3-24
      - VSAREA VC:3-25
      - VSBLNK VC:3-26
      - VTOTAL VC:3-27
    - serration pulses VC:4-13
    - signals in
      - composite area VC:2-6
      - composite sync VC:2-6
      - vertical blanking VC:2-5
      - vertical synchronization VC:2-5
    - understanding VC:4-22 to VC:4-24
  - interlaced video
    - equalization pulses VC:4-10
    - serration pulses VC:4-10
  - internal video logic VC:2-3
  - interrupts
    - See also* VFTINT register
    - frame timer VC:A-18

**L**

line increment register. *See* LINEINC register

LINEINC register VC:5-13  
calculating F0STADR VC:5-6

**M**

masks. *See* SRT controller, control masks

memory map  
SRT controller VC:1-8  
video controller VC:1-5

monitor  
specifications for noninterlaced VC:A-5  
timing for noninterlaced VC:A-5

multiplexers  
for VC VC:1-4

MVP  
block diagram VC:1-2

**N**

next address register. *See* NEXTADR register

NEXTADR register VC:5-14

noninterlaced  
frame control VC:1-4  
vertical timing VC:4-6 to VC:4-7  
video timing VC:4-15 to VC:4-17  
horizontal operation VC:4-4,  
VC:4-17  
vertical operation VC:4-19

noninterlaced mode  
registers for configuring  
VEAREA VC:3-21  
VEBLNK VC:3-22  
VESYNC VC:3-23  
VFTINT VC:3-24  
VSAREA VC:3-25  
VSBLNK VC:3-26  
VTOTAL VC:3-27

noninterlaced mode (continued)  
serration pulses VC:4-13  
signals in  
composite area VC:2-6  
vertical blanking VC:2-5  
vertical synchronization VC:2-5  
understanding VC:4-15 to VC:4-16

noninterlaced video  
equalization pulses VC:4-10  
serration pulses VC:4-10  
NTSC video standards VC:4-28  
example VC:4-30  
vertical end area programming  
VC:4-36 to VC:4-38  
vertical frame timer interrupt  
programming VC:4-39 to  
VC:4-41  
vertical start area programming  
VC:4-33 to VC:4-35

**O**

on-chip memory  
register bus VC:1-4  
VC memory map VC:1-5

**P**

packet transfers  
*See also* SRT controller, transfer requests  
request  
video controller-initiated VC:5-22  
select bit. *See* FMEMCTL register,  
PTS bit

PAL video standards VC:4-28  
example VC:4-31  
vertical end area programming  
VC:4-38  
vertical interrupt programming  
VC:4-41  
vertical start area programming  
VC:4-34 to VC:4-35

pixel size  
determining VC:A-4

porches VC:A-2

priority  
   of SRT events VC:5-23 to VC:5-24

programming  
   alternatives VC:4-42 to VC:4-44  
   composite interlaced NTSC  
     example VC:A-13  
   frame timer registers  
     interlaced video VC:A-16  
     noninterlaced video VC:A-7  
   HALINE register VC:A-12  
   HBLINE register VC:A-12  
   in interlaced mode VC:4-33  
     VEAREA register VC:4-36 to  
       VC:4-38  
     VFTINT register VC:4-39  
     VSAREA register VC:4-33 to  
       VC:4-35  
   SRT registers VC:A-10 to VC:A-11  
   video timing registers  
     interlaced video VC:4-32  
     noninterlaced video VC:4-21

PTS bit. See FMEMCTL register, PTS  
 bit

## R

register map  
   frame timer 0 VC:1-6  
   frame timer 1 VC:1-6  
   SRT controller VC:1-8

registers VC:1-6  
   accessing VC:1-7  
   frame timer control  
     FTCTL VC:3-4  
   horizontal timing  
     HALINE VC:3-8  
     HBLINE VC:3-9  
     HCOUNT VC:3-10  
     HEAREA VC:3-11  
     HEBLNK VC:3-12  
     HESERR VC:3-13  
     HESYNC VC:3-14  
     HSAREA VC:3-15

registers, horizontal timing (continued)  
   HSBLNK VC:3-16  
   HTOTAL VC:3-17  
   SETHCT VC:3-18  
   SRT controller  
     CRNTADR VC:5-4  
     F0STADR VC:5-5 to VC:5-6  
     F1STADR VC:5-7  
     FMEMCTL VC:5-8 to VC:5-12  
     LINEINC VC:5-13  
     NEXTADR VC:5-14  
     SAMMASK VC:5-15  
   unused locations VC:1-7  
   vertical timing  
     SETVCT VC:3-19  
     VCOUNT VC:3-20  
     VEAREA VC:3-21  
     VEBLNK VC:3-22  
     VESYNC VC:3-23  
     VFTINT VC:3-24  
     VSAREA VC:3-25  
     VSBLNK VC:3-26  
     VTOTAL VC:3-27

requests. See SRT controller, transfer  
 requests

## S

SAM overflow  
   events VC:5-2  
   SRT requests VC:5-2

SAMINC control mask. See SRT  
 controller, control masks, SAMINC

SAMMASK register VC:5-15

SCLK  
   description VC:A-6  
   determining frequency VC:A-6  
   signals VC:2-8

screen resolution VC:A-4  
   provided by frame timers VC:1-4

serial access memory  
   increment control mask. See SRT  
   controller, control masks, SAMINC  
   mask register. See SAMMASK  
   register  
   overflow. See SAM overflow

serial clock input  
   signal descriptions VC:2-8

serial register transfer  
   controller. *See* SRT controller  
   signals. *See* SRT signals

serration pulses VC:4-10 to VC:4-11  
   on  $\overline{\text{CSYNC}}$  VC:4-13

SETHCT register VC:3-18

SETVCT register VC:3-19

signals  
   *See also* composite signals;  
   horizontal timing, signals; SRT  
   signals; vertical timing, signals

FCLK  
   description VC:2-3

SRT controller  
   address generation VC:1-4,  
   VC:5-25 to VC:5-26  
   control masks  
     COLMASK VC:5-16  
     description VC:5-16  
     SAMINC VC:5-16  
   description VC:5-2  
   events VC:5-17 to VC:5-22  
     end of field VC:5-19  
     line VC:5-19  
     priority of VC:5-23  
     SAM overflow VC:5-20 to  
       VC:5-21  
     start of field VC:5-18  
   frame timer requests VC:5-2  
   masks. *See* SRT controller, control  
   masks  
   register transfer modes VC:5-27 to  
   VC:5-34  
     capture VC:5-29 to VC:5-30  
     display VC:5-27 to VC:5-28  
     interlaced line repeat VC:5-34  
     merge capture VC:5-31 to  
       VC:5-33

SRT controller (continued)  
   registers  
     addresses VC:5-3  
     CRNTADR VC:5-4  
     description VC:5-3  
     F0STADR VC:5-5 to VC:5-6  
     F1STADR VC:5-7  
     FMEMCTL VC:5-8 to VC:5-12  
     LINEINC VC:5-13  
     map VC:1-8  
     NEXTADR VC:5-14  
     SAMMASK VC:5-15  
   SAM overflow  
     transfer requests VC:5-2  
   transfer requests VC:1-4, VC:5-2  
   video controller-initiated packet  
     transfer requests. *See* VCPT  
     requests

SRT signals VC:2-2  
   descriptions VC:2-8

SSE bit. *See* FTCTL register, SSE bit

synchronization VC:1-4  
   composite signals VC:2-6  
   delay VC:4-50 to VC:4-51  
   external VC:1-4  
     FCLK VC:4-49  
     modes VC:4-46  
     sync output VC:4-47 to VC:4-48  
     video counter operation  
       VC:4-44 to VC:4-45  
   high-impedance VC:4-52  
   horizontal VC:2-4  
   internal VC:1-4  
   vertical VC:2-5

## T

timing  
   composite signals VC:4-2  
     description VC:2-6  
   composite video VC:4-9  
   control signals VC:2-2

timing (continued)  
     horizontal timing signals  
         relationship with vertical timing signals VC:A-2 to VC:A-3  
     interlaced video VC:4-22  
         horizontal operation VC:4-25  
         vertical operation VC:4-28 to VC:4-29  
     noninterlaced video VC:4-15 to VC:4-17  
         horizontal operation VC:4-17  
         vertical operation VC:4-19  
     porches VC:A-2  
     vertical timing signals VC:4-6  
         relationship with horizontal timing signals VC:A-2 to VC:A-3  
 TMS bits. *See* FMEMCTL register, TMS bits  
 transfer mode select bits. *See* FMEMCTL register, TMS bits

## U


UED bit. *See* FMEMCTL register, UED bit  
 unblanked event disable bit. *See* FMEMCTL register, UED bit

## V

$\overline{\text{VBLNK}}$  signals VC:4-2  
     capture VC:2-5  
     repetition VC:2-5  
 VC  
     *See also* SRT signals  
     applications VC:1-4  
     block diagram VC:1-3  
     crossbar VC:1-2  
     description VC:1-2  
     frame buffer VC:1-2  
     frame system VC:1-2  
     functional blocks VC:1-4  
     interface  
         with the MP VC:1-4  
     location VC:1-2

VC (continued)  
     MP addresses VC:1-5  
     overview VC:1-2  
     register interface VC:1-4  
 VCOUNT register VC:3-20  
     interlaced vertical video timing VC:4-8  
         horizontal halflines VC:4-8  
     noninterlaced vertical video timing VC:4-6 to VC:4-8  
         horizontal lines VC:4-6  
 VCPT requests VC:5-2, VC:5-22  
 VEAREA register VC:3-21  
 VEBLNK register VC:3-22  
 vertical  
     frame timer interrupts. *See* interrupts; VFTINT register  
 vertical timing  
     registers VC:1-7, VC:3-2  
         *See also* registers, vertical timing addresses VC:3-3  
     signals  
         blanking VC:2-5  
         description VC:2-5  
         interlaced VC:4-8  
         noninterlaced VC:4-7  
         synchronization VC:2-5  
         understanding VC:4-2  
         video VC:4-6  
 VESYNC register VC:3-23  
 VFTINT register VC:3-24  
     *See also* interrupts  
     programming VC:4-39 to VC:4-41  
 video controller. *See* VC  
 video controller-initiated packet transfer requests. *See* VCPT requests  
 video counter operation VC:4-44 to VC:4-45  
 video timing logic VC:4-50 to VC:4-51  
 VPM bits. *See* FTCTL register, VPM bits



- VRAM
  - serial access memory mask register.  
See SAMMASK register
- VRAM control logic
  - tap point VC:2-8
- VSAREA register VC:3-25
- VSBLNK register VC:3-26
- $\overline{\text{VSYNC}}$  pin mode bits. See FTCTL register, VPM bits
- $\overline{\text{VSYNC}}$  signals VC:4-2
  - configuration VC:2-5
  - description VC:2-5
  - indicating “end of page” VC:2-5
- VTOTAL register VC:3-27
- 
- XPT transfers
  - requests VC:5-22





## **IMPORTANT NOTICE**

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

**TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.**

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.