VESA®

Video Interface Port Standard

Video Electronics Standards Association

2150 North First Street, Suite 440
San Jose, CA 95131-2029

Phone: (408) 435-0333
FAX: (408) 435-8225

VESA Video Interface Port (VIP)

Version: 2

Date: October 21, 1998

Purpose

The VESA Video Interface Port (VIP) Specification defines a standard method of connecting digital video devices to graphics display adapters.

Summary

Many digital video devices require access to the graphics display adapter's frame buffer memory to allow advanced feature support such as Video conferencing, DVD playback, and enhanced TV support. VESA VIP is a dedicated physical connection between a graphics adapter and one or more 3rd party hardware devices, such as MPEG-2 or HDTV decoders, video digitizers, video encoders, etc. With a dedicated connection, devices supplying video data do not have to compete with other data movement on the current industry standard PCI bus.

VESA - The Video Electronics Standards Association
Video Interface Port (VIP) Specification

Intellectual Property

Trademarks

Patents

VESA proposals and standards are adopted by the Video Electronics Standards Association without regard to whether their adoption may involve any patents or articles, materials, or processes. Such adoption does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the proposals or standards documents.

Support

Clarifications and application notes to support this standard may be written. To obtain the latest standard and any support documentation, contact VESA.

If you have a product which incorporates *VESA VIP*, you should ask the company that manufactured your product for assistance. If you are a manufacturer, VESA can assist you with any clarification you may require. All comments or reported errors should be submitted in writing to VESA using one of the following methods.

- Fax 408-435-8225, *direct this note to Technical Support at VESA*
- e-mail support@vesa.org
- mail to Technical Support
  VESA - Video Electronics Standards Association
  2150 North First Street, Suite 440
  San Jose, CA 95131-2029

VESA VPORT Committee Members

Any industry standard requires input from many sources. The people listed below were members of the VESA VPORT Committee which were responsible for combining all of the industry input into this standard:

MEMBERS

Chairman, Chris Lam, Innovative Semiconductors, Inc.

Bruce Busby, 3Dfx Interactive

Max Ma, 3Dlabs, Inc.

Ed Callway, ATI Technologies, Inc.

Hiten Patel, Fujitsu Microelectronics, Inc.

Peter Sherlock, Integrated Device Technology

Nilesh Shah, Intel Corporation

Bob Matthews, LSI Logic

John Wong, Macrovision

Nick Glantzis, Matrox Technology, Inc.

Syed Zaidi, NeoMagic Corporation

Philip Wang and Morgan Tang, Oak Technology, Inc.

Leo Warmuth, Philips Semiconductors, Inc.

Ben Felts, Rockwell Semiconductors, Inc.

Steve Gibson and Kiumars Sabeti, S3, Inc.

John Gerard, Samsung Semiconductor, Inc.

Khanh Dang, SGS-THOMSON Microelectronics

Jon Kiachian and James Kim, Silicon Image, Incorporated

Ken Mano, Sony Electronics, Inc.

Wen Li, Texas Instruments Semiconductors

John Matsumoto, Toshiba America Information System

Simon Hong, Trident Microsystems, Inc.

Table of Content

1 Overview

1.1 Scope

This specification describes the VESA VIP2 Standard. VIP2 is backward compatible with VIP1.1. The first eight chapters of this document cover VIP1.1, while chapter 9 and beyond cover the VIP2 extensions.

1.2 What is VIP?

Video Interface Port, VIP, is a standard interface between a "video-enabled" graphics chip and one or more video devices (e.g. MPEG2 decoder, video decoder). The key objectives for VIP are:

- Common open standard for all new graphics chips and PC-oriented video devices
- Low cost
- Low Pin Count
- Support multiple devices concurrently
- Adequate performance for today and future applications

1.3 VIP Key Features

- Requires only 14-signal pins from the graphics chip for a typical implementation
- Separate video and host control ports
- 26-pin VIP Connector port; can coexist with VESA Feature Connector
- Separate 14-pin connector for power, reset, and I^2S digital audio
- Supports one video module and one ribbon cable board simultaneously
- Supports up to four VIP slave devices
- Plug-and-play support through the graphics chip PCI interface
- Video Port
  - Simplified ITU-R-656 Video Format -- supports HSYNC, VSYNC, ODD FIELD, EVEN FIELD, and ANCILLARY DATA functions
  - VBI data output from video decoder is through ITU-R-656 Ancillary Data or through the host port
  - Supports variable video resolutions and scan rates
  - Supports both interlaced and non-interlaced video
- Host Port
  - Synchronous bus with clock frequency ranging from 25MHz to 33MHz
  - Two data bits; it takes 4 cycles to transfer one byte
  - Supports burst mode, master or slave-terminated transfers, wait-states, and timeout transfers

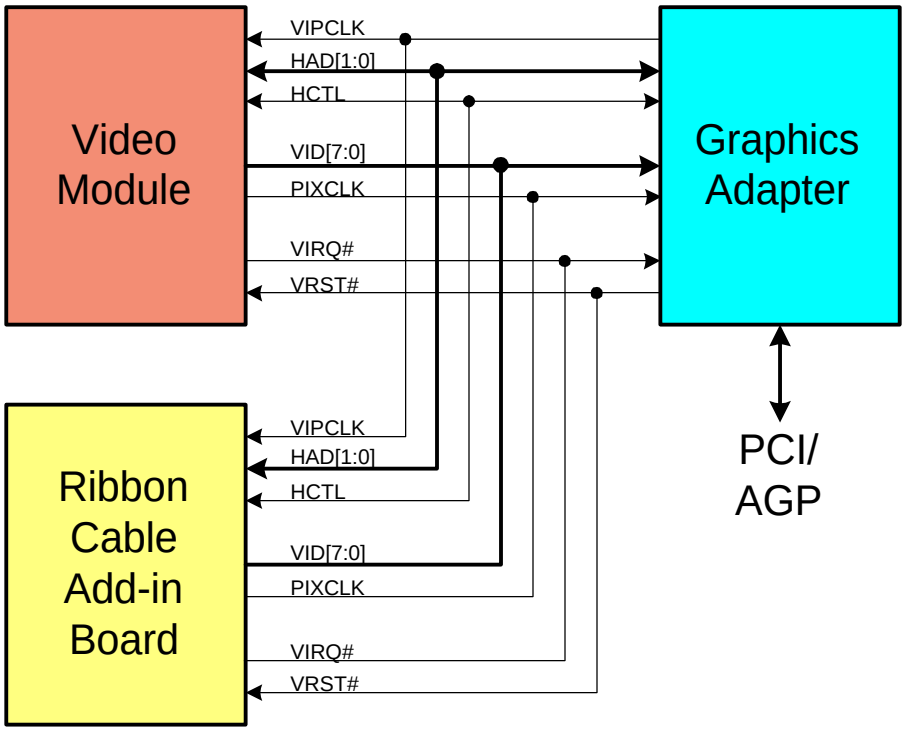
1.4 Device and Module Limitations

VIP allows up to four VIP devices in a system. However, only two physical modules are allowed. One is a plug-in module to the motherboard VIP Connectors A and B. Typically this is the DVD module. This module should also have a VIP 26-pin Connector A right angle male connector for the ribbon cable connection.

The other module is a ribbon cable add-in board. This is usually an ISA or PCI board with a VIP ribbon

cable. Typically this is the TV/Video Capture/Video Phone board. In addition to the two physical modules, VIP devices can also be implemented directly onto the motherboard or the graphics adapter.

1.5 Block Diagram



1.6 Signal Description

Signal	I/0 Type (From VIP Master Side)	Description
Host Port		
VIPCLK	O	VIP Host Clock (25-33MHz)
HAD[1:0]	I/O	Host Address/Data Bus
HCTL	I/O	Host Control; this includes the symbolic signals of VFRAME, DTACK#, and VSTOP#
Video Port		
VID[7:0]	I	VIP Video Data
PIXCLK	I	Video Pixel Clock
System Signals		
VRST#	O	VIP Module Reset
VIRQ#	I - OD	Interrupt Request - Open Drain (CMOS) Open Collector, Level Sensitive

2 Theory of Operation

2.1 Note on Compatibility

VIP will be extended in the future to support more demanding applications. All register flags that are marked “reserved” in this specification must be tied to zero. This will ensure software compatibility with future extensions.

2.2 The Video Bus

The video bus is a unidirectional bus with an 8-bit YCrCb bus and a PIXCLK. See Chapter 8 for details on the video format. On power up no VIP slave device should drive this bus until activated by the graphics chip. See timing diagram for more details.

2.3 VIP Host Bus

2.3.1 VIP Host Signals

Four pins are required for host port transfers, VIPCLK, HAD[1:0] and HCTL.

VIPCLK is the host port clock, specified from 25-33MHz. VIPCLK can be from any source. A good example is the PCI system clock.

HAD[1:0] is a two wire bus, and is used to transfer commands, addresses and data between the master (graphics controller) and the slave devices.

HCTL is a shared control pin. It is driven by the master to initiate and terminate data transfers. It is driven by the slave to terminate and add wait states to data transfers. Because it is a shared control signal, special attention must be given to its generation to avoid bus conflicts. The master pulls HCTL high when no one drives it.

2.3.2 Terminology

2.3.2.1 Cycle and Phase

A VIP host bus transfer consists of a number of *phases*. Each phase consists of a number of *cycles*.

A *cycle* refers to a particular VIPCLK clock period.

A *phase* refers to a group of one or more cycles that constitute a functional part of a complete transfer. There are a total of seven phase types, which are explained in detail below. Note that phase definitions are provided to facilitate explanation of the Host Port operation and do not necessarily represent actual states required by either the master or slave state machines. For example, the Retry, Wait and Data phases are similar and might be implemented as a single flow. However they are separated in the following discussion since they have subtle differences.

2.3.2.2 The Symbolic Signals

HCTL is the only physical control pin for the Host Bus. It is shared between the master and the slaves. Three symbolic signals can be indicated as follows:

Symbolic Signal	Explanation
VFRAME	VFRAME is the symbolic name for the signal generated by the master to frame a bus transfer. The master drives VFRAME high to start a bus transfer and low during the first clock of a Wait or Data phase to end the transfer.
DTACK#	DTACK# is the symbolic name for the signal driven by the slave onto HCTL during the second cycle of a Retry, Wait or Data phase to indicate its ability to accept the next data byte.
VSTOP#	VSTOP# is a symbolic name for the signal driven by the slave onto HCTL during the first cycle of a Retry, Wait or Data phase to end the transfer.

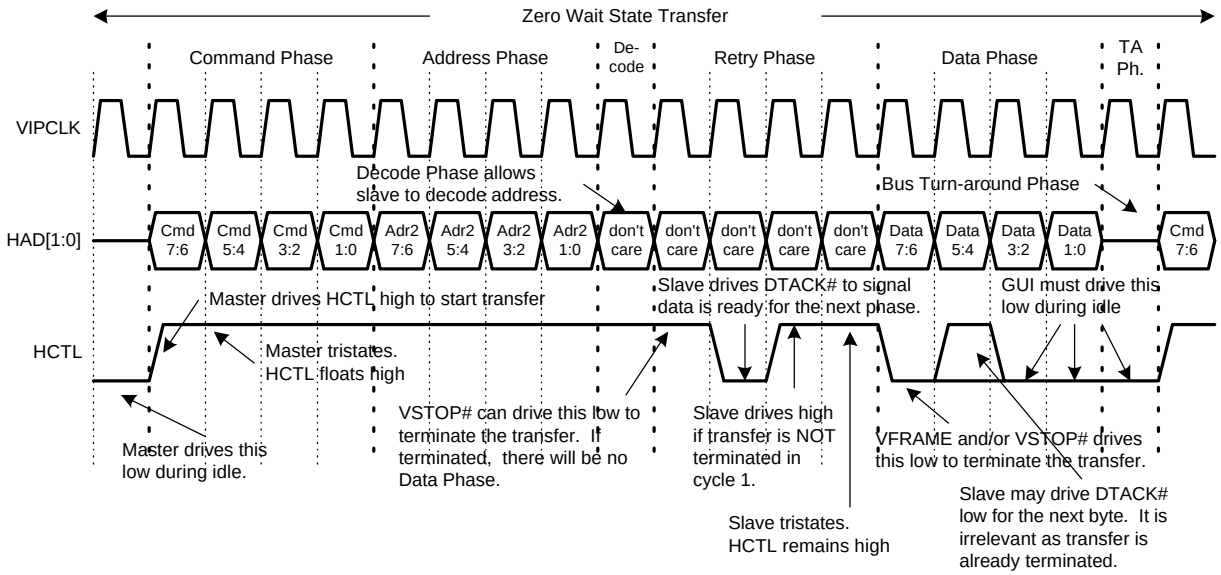
2.3.2.3 The VIP Phases

Phase	Explanation
Command	All Host Port transfers start with a Command phase. The 8-bit Command/Address byte is multiplexed onto HAD[1:0] during the Command phase. The command byte selects between devices, read, and write cycles, register or FIFO transfers and contains the most significant four bits of the register address.
Address	During register transfers the Command phase is followed by the Address Extension phase. The least significant 8-bits of the VIP Register Address are multiplexed onto HAD[1:0] during the Address Extension phase. This phase is not present during FIFO transfers.
Decode	Following the Command or Command/Address phase(s), a one clock delay is required to allow slave devices to decode the address and determine if they are able to respond within the 1 wait phase requirement for active operation.
Retry	The four clock cycles immediately following the Decode phase constitute the Retry phase. During the Retry phase, the slave indicates its desire to terminate the operation without transferring any data (retry), add a Wait phase or transfer the first byte of data. When the slave asserts VSTOP#, the Transfer ends with the Retry phase. When the slave asserts DTACK#, the Retry phase is followed by a Data phase. When the slave neither terminates the transfer nor accepts the byte, the Retry phase is followed by a Wait phase.
Wait	During the second cycle of a Decode, Retry or Wait phase, the slave indicates its ability to transfer the next byte of data by driving HCTL low. When the slave does not drive HCTL low and the transfer is not terminated, the current phase is followed by a Wait phase. During Wait phases, the current owner (master for writes, slave for reads) continues to drive the HAD bus however no data is transferred. The slave is allowed to add one wait phase per byte to register accesses without compromising system timing. Additional wait phases are not prevented but overall system reliability may be compromised.
Data	When HCTL is de-asserted during cycle 1 of a Retry, Wait or Data phase, the current phase is followed by a Data phase. Data is transferred between master and slave devices during data phases, multiplexed onto HAD[1:0].
TA (Turn Around)	Immediately following the last transfer phase of a read transfer, a one cycle delay is required giving the slave time to tri-state the HAD bus. The master is free to begin a new bus transfer, driving HAD and HCTL immediately following the TA phase.

2.3.3 VIP Transfer Example

The following timing diagram illustrates a standard zero-wait-state single byte register transfer. It is intended as an overview of the bus operation, and not a complete protocol specification. For detail protocol rules please refer to Section 5. The transfer starts when HCTL is driven high by the graphics chip. This starts the Command Phase of the transfer in which the command byte is issued by the graphics chip. It is followed by the Address Phase – the second address byte. (If it were a FIFO transfer, there would be no second address byte).

The Decode Phase is followed by the Retry Phase. HCTL is active during this phase, which provides the signal flags for VFRAME, DTACK#, and VSTOP#. HAD[1:0] is different between a read and a write cycle during the Retry Phase. For a write cycle, the graphics chip continues to drive HAD[1:0], however, its value is undefined. During the Data Phase the graphics chip drives the data onto the bus. For a read cycle, the graphics chip will get off the bus during the Retry Phase and allow the selected slave to start driving the bus for the Data Phase.



2.3.4 Hardware Polling

VIP supports a hardware-polling scheme. A slave may support hardware polling ports. These are typically the FIFO ports, such as the ports for MPEG2 compressed data. When the slave is not ready to handle data it asserts VSTOP# without asserting DTACK#. This results in no data transfer.

The use of bus transfer initiation for polling has the one key advantage: if the slave is ready, the bus cycle initiated is used for data transfer immediately. There is no need to terminate polling, examine the status, and then start the data transfer. Hardware polling can significantly reduce the software overhead.

Sometimes the graphics chip may need to poll more than one output port. For instance, if an MPEG2 is playing DVD while the video decoder is capturing VBI data. The graphics chip should do polling on both the compressed data port of the MPEG2 decoder and the VBI data port of the video decoder.

2.3.5 Timeout Condition

Timeout cycles provide software with a mechanism for determining what devices are connected to the VIP

bus. Software tries to read the Vendor ID register in each of the possible VIP target devices. When no device responds to the request within a pre-determined number of clocks, the VIP master terminates the bus transfer with a timeout and returns the value "0xFF" for each of the requested bytes. When the device is present, the Vendor ID is returned to the initialization software. As in the PCI specification, the value "0xFFFF" is an invalid Vendor ID. See Timeout timing diagram for details.

After initialization, timeout cycles are indications of error conditions. A timeout occurs on any data phase where the target doesn't complete the transfer (by asserting DTACK# or VSTOP#) within a pre-determined number of clocks. The transfer is considered to be completed by the VIP master and will not be retried. The value returned on a read is undefined. VIP masters may choose to return "0xFF" for any timeout but are only required to return this value for requests to the Vendor ID register.

The timeout period is measured from the first phase that samples DTACK# or VSTOP#. This can be the Retry Phase, when no data has been transferred, or a Data Phase, when some data has been transferred in a burst transfer. The timeout period is programmable. The maximum is 16 VIP phases, including the first sample phase.

2.3.6 Command/Address Bytes

The first byte issued by the graphics chip during an access transfer is a Command/Address Byte. It contains the following bits:

Address Bit	Name	Description
7	DEVSEL1	Device Select
6	DEVSEL0	Support up to 4 devices
5	R/W#	1=Read; 0=Write
4	F/R#	1=FIFO Acc.; 0=Register Acc.
3	UA3	
2	UA2	Upper Address Bus
1	UA1	
0	UA0	

The second byte contains the lower eight bits of the register address ADR[7:0].

2.3.7 Device Select Types

The two DEVSEL bits allow up to four devices. Two device types are pre-defined as follows:

DEVSEL1	DEVSEL0	Device Type
0	0	MPEG (MPEG1 or MPEG2)
0	1	Video Decoder
1	0	User-define
1	1	User-define

2.3.8 FIFO vs. Register Access

Both FIFO and register accesses follow the same bus protocol with one exception -- no second address byte for FIFO access. Even though wait states are support for both FIFO and register accesses, an efficient slave should support zero-wait-state burst transfers.

UA[3:0] is the FIFO address, providing 16 FIFO ports per device. UA[3:0]+ADR[7:0] is the register address, providing 4,096 locations per device. All addresses, including both FIFO and register, refer to byte locations. All transfers are byte-oriented, and can be terminated at any byte boundary.

2.3.9 Burst Transfer

Burst mode transfer is implicit in all VIP transfers. As long as the graphics chip asserts VFRAME, consecutive bytes are transferred until terminated either by the graphics chip or the slave. It is up to the graphics chip designer to define the maximum burst length. If a register access is pending the graphics chip should terminate a long burst transfer to minimize the latency.

In a burst FIFO transfer no address increment is assumed. In a burst register transfer the address increments in an ascending order. The first address always refers to the least significant byte of a register address range.

2.3.10 Predefined FIFO Ports

The location of the status registers and FIFO ports are predefined as follows. A VIP system has a maximum of four FIFO DMA channels. A typical implementation including a MPEG and a video decoder will use three channels: MPEG compressed video, MPEG compressed audio, and VBI from the video decoder.

Location	FIFO Port
0	Status 0
1	Status 1
2	User Defined
3	User Defined
4	FIFO A
5	FIFO B
6	FIFO C
7	FIFO D
8-15	User Defined

2.3.11 Status 0

Bit	Flag	Description
0	VIRQ	This flag reflects the state of the VIRQ# pin. A “1” indicates that the VIRQ# pin is asserted (low). An access does not clear this flag.
1	User Define	
2	User Define	These flags are defined by the VIP device manufacturer
3	User Define	
4	DREQA	DMA Request for FIFO A
5	DREQB	DMA Request for FIFO B
6	DREQC	DMA Request for FIFO C
7	DREQD	DMA Request for FIFO D

2.3.12 Status 1

Bit	Flag	Description
0	PRESENT A	1=FIFO A exists for this device
1	R/W# A	1=FIFO A is a Read Port; 0=Write
2	PRESENT B	1=FIFO B exists for this device
3	R/W# B	1=FIFO B is a Read Port; 0=Write
4	PRESENT C	1=FIFO C exists for this device
5	R/W# C	1=FIFO C is a Read Port; 0=Write
6	PRESENT D	1=FIFO D exists for this device
7	R/W# D	1=FIFO D is a Read Port; 0=Write

2.4 VBI Data

VIP is designed to support concurrent MPEG and Video Decoder VBI Capture. In this concurrent mode the MPEG decoder occupies the video port. Sliced VBI data from the video decoder must be transferred through the host port. ITU-R-656 header codes for VBI data must be preserved in the transfer. A VIP-compliant graphics chip must be able to read a minimum of 16 bytes for every 16uS. A VIP-compliant video decoder must be able to operate within the 16byte/16us requirement without losing data.

2.5 Power Down Mode

The host CPU can notify the graphics chip to enter the power down mode. The graphics chip will stop VIPCLK after it is in the idle state and that there is no active DMA transfer. The graphics chip will have the option of tri-stating HCTL and letting it flow high. HCTL must be driven low by the graphics chip at least 100ns before it re-activates VIPCLK. All VIP slaves must support the stopping of VIPCLK.

2.5.1 Power States

Four power states are defined below:

Power State	Description
P0	**Fully on (REQUIRED).** V Port devices must support this state. This is the "normal" operating state and is also the state that is always entered after a reset.
P1	**Some power savings.** P1 is an optional power state. This is used to save some power while providing very low turn-on latency. This state should consume less power than P0 and should be ready for operation within less than 10mS. Power and clock will be supplied in this state.
P2	**More power savings - clock may or may not be removed.** P2 is an optional power state. This is used to save significant power while providing moderate turn-on latency. This state should consume less power than P1 (if P1 is implemented) and should be ready for operation within less than 1 Second. Context should be retained. Power will be supplied but the clock may or may not be active in this state. The device must be ready to respond to host accesses to the configuration space when clock is supplied. The video port must be inactive.

P3	**Fully off - power may or may not be removed (REQUIRED).** This is a required state. In this power state, the device should consume minimal power. The turn-on latency is similar to that at reset. Power and clock may or may not be active in this state. The device must be ready to respond to host accesses to the configuration space when power and clock are supplied. Context need not be preserved and it is suggested that an internal reset be generated when the device is programmed back to the D0 state.

2.5.2 Separate Power State Hazards

This specification assumes the graphics and VIP device power are either both enabled or both disabled together (within 1mS). This would likely be the case where VIP is implemented as a daughter card on the graphics board. If this is not the case, e.g., a separate VIP ribbon-cable board, the OS must ensure that the power to the buses that contain both the graphics and VIP devices are in the same state at all times. If it is impossible to maintain the same power state, the system designer must ensure that damage will not occur to either device if one is powered without the other.

2.5.3 Power Management Programming Interface

Power management functions are implemented through the Command and Status registers in Configuration Space. For details please refer to the section on VIP Device Configuration Space.

2.6 Plug-and-Play

2.6.1 VIP Power Up Detection

Graphics vendors may implement several different video interface buses in their graphics controllers in order to preserve backward compatibility with previous graphics controllers. A controller which implements the VIP bus as well as a previous version of a video interface bus needs to indicate to the Windows driver which bus the physical board layout supports.

All VIP slave devices must drive Connector A pin Z5 (VIP: HAD[0] Pin) low while VRST# is active. In the feature connector mode this pin is ESYNC#. A VIP-compliant graphics chip must pull this pin high when nobody drives it. After reset, a VIP slave must tri-state this pin to allow it to float high. This allows the graphics chip to detect the presence of VIP devices and activate the VIP mode.

2.6.2 VIP Slave Devices

VIP slave devices must boot up with the ability to respond to VIP configuration register accesses by the graphics chip. When the PC boots, the VIP-enabled Windows driver attempts to read the configuration space of device 0, device 1, device 2, and device 3. If an access results in timeout, the graphics chip assumes the device is absent. The graphics chip identifies these devices and reports their presence to the Plug and Play configuration manager. Once the devices are registered, the normal Plug and Play driver installation proceeds for the devices.

2.6.3 Legacy 656 Devices

VIP can coexist with legacy video devices that are controlled via I^2C. These devices need to power on with their outputs tri-stated so that they don't interfere with the querying of the VIP Slave Devices. Once

the software has queried the VIP Slave devices, the Windows driver will utilize the I^2C bus associated with VIP in the particular graphics chip implementation, and access the supported legacy decoders.

The method for identifying legacy devices through I^2C is beyond the scope of VIP. The implementation is up to the graphics chip vendors.

2.6.4 Interrupt Request

A VIP-enabled graphics chip driver will request an interrupt during initialization if it detects the presence of VIP device(s) on power up. This is based on the assumption that all MPEG decoders need interrupts, and it is very likely that a VIP-enabled PC contains a MPEG decoder.

2.6.5 VIP Device Configuration Space

Every VIP slave device must support the following configuration registers. Each register consists of two bytes. The lower byte address always refers to the least-significant byte of the register.

Register Address	Register	I/O Type	Description
001:000	Vendor ID	Read Only	This field identifies the manufacturer of the device. Valid vendor identifiers are allocated by the PCI Special Interest Group to ensure uniqueness.
003:002	Device ID	Read Only	This field identifies the particular device, and is allocated by the vendor.
005:004	Subsystem Vendor ID	Read Only	This field identifies the subsystem manufacturer, e.g. the board manufacturer. It is up to the device manufacturer to implement ways for the subsystem vendor to provide this ID. An example is power-up resistor strapping.
007:006	Subsystem Device ID	Read Only	This field is similar to the Subsystem Vendor ID. However, it refers to the device ID of the subsystem vendor. It is up to the device manufacturer to implement ways for the subsystem vendor to provide this ID.
009:008	Command	R/W	This register will contain command flags for various functions. For now only power management command flags are defined. It will be extended in the future.
00B:00A	Status	Read Only	This register contains various status and capability flags. For now only power management status flags are defined. It will be extended in the future.
00D:00C	Revision ID	Read Only	This register specifies a device specific revision identifier. The value is chosen by the vendor. Zero is an acceptable value. This field should be viewed as a vendor defined extension to the Device ID

2.6.5.1 Command Register

Bit	Field	Description
1:0	Power_State	A read to this field reports the current operating power state (P0 to P3). A write to this field sets the device into a new power state. If software tries to set an unsupported state, the transaction should complete normally but the state should be left unchanged.
15:2	Reserved	

2.6.5.2 Status Register

Bit	Field	Description
0	P1_Support	"1" indicates that power state P1 is supported.
1	P2_Support	"1" indicates that power state P2 Is supported.
15:2	Reserved	

3 Arbitration Scheme Analysis

3.1 Overview

A straightforward round robin scheme is presented as an example arbitration scheme. This scheme is provided to illustrate required system timing, and as proof that the timing constraints can be met. Other schemes exist and may be more optimal for a particular system. The actual implementation of VIP host port arbitration is left to the graphics chip designer subject to the real time constraints of the system.

In this scheme the graphics chip Host port arbitration implements a round robin polling scheme, selecting between PCI requests, VBI transfers and MPEG compressed bitstreams. Minimum burst sizes are chosen to ensure required bandwidth however they may be shorter when there is less data to be transferred.

3.2 Timing Assumptions

1. Register accesses occur with a maximum of one wait phase (4 cycles) per byte.
2. Time critical data streams such as MPEG bitstreams and VBI data are mapped into FIFO space and occur without wait states when the slave is able to accept or provide the data.
3. Hardware support is provided in the graphics chip for MPEG compressed bitstream and sliced VBI data transfers.
4. The Host port master does not prevent the slave from adding additional wait phases but reliable system operation may be compromised.
5. The graphics chip must be able to transfer VBI data at a rate of 16 bytes/16 us when the VBI source has data to transfer.
6. The VBI source must be polled at least once within each 16 us period to prevent loss of VBI data.
7. The graphics chip must be able to transfer compressed MPEG bitstreams at an average rate of 2 Mbytes per second.

3.3 Timing Analysis

3.3.1 Round Robin Loop Timing

Up to 4 bytes can be transferred during each PCI request slot. This insures that at least one PCI data phase can be completed each time the PCI port is serviced. Given the latency of a Host port register transfer, it is assumed that writes and reads are posted or retried on the PCI bus. Assuming one wait phase per byte, each PCI transfer has an overhead of 14 (Command, Address, Dec, Retry, TA) + 8 (Wait, Data) * the number of bytes to be transferred. The maximum time required for a dword read is:

14 + 8 * 4 = 46 VIPCLKs = 1.84us (VIPCLK=25MHz)

VBI transfers have a maximum burst length of 8 bytes. The overhead is 10 clocks (Command, Dec, Retry, TA) + 4 clocks/byte. The maximum time for a VBI transfer is:

10 + 4 clocks/byte * 8 bytes = 42 VIPCLKS = 1.68 us.

MPEG transfers have a maximum burst length of 16 bytes. Although the Turn-Around phase is not required for Host write transfers, it is included here to be conservative. The overhead is 10 clocks (clocks (Command, Dec, Retry, TA) + 4 clocks/byte. The maximum time for an MPEG transfer is:

10 + 4 clocks/byte * 16 bytes = 74 VIPCLKS = 2.96 us.

The maximum time to complete the loop when all requestors need serviced is the sum of the maximum latencies is:

1.84 us + 1.68 us + 2.96 us = 6.48 us.

3.3.2 VBI Transfer Requirements

The VBI device gets polled once every 6.48us and transfers up to 8 bytes per request. This allows 16 bytes to be transferred every 13 us, within the VIP specification.

3.3.3 MPEG Transfer Timing

Up to 16 bytes of compressed bitstream can be transferred every 6.48 us. This provides a minimum guaranteed bandwidth of 16/6.48 = 2.48 Mbytes / sec.

3.3.4 PCI Timing

The worst case PCI latency is about 6.48uS per dword transfer in this round-robin example. There are times when it is desirable to reduce this latency. For example, host access to the Status Register 0 during the interrupt handler should incur low latency. One simple way is to have a low latency VIP Host port configuration register that allows the interrupt handler to disable MPEG bitstream transfers while it reads the status registers.

This reduces the loop latency to 1.84 us + 1.68 us = 3.52 us. Status registers will actually be much faster since they are mapped into FIFO space and are only byte transfers.

Since there will be many loop periods where the PCI bus is not requesting access to VIP Target space, there will be opportunities for the MPEG bitstream transfer to catch up.

4 Mechanical Specification

4.1 VIP Connectors

VIP uses two dual row 0.1" center connectors -- Connector A and B. Connector A is a 26-pin connector, which is the same as the standard Feature Connector. Connector B is a 14-pin connector consisting of the power pins, ground, VRST#, and I^2S. On the master side, Connector A is male, and Connector B is female. This applies to either a graphics adapter card or the motherboard. Slave modules will have the opposite mating connectors. Connector B on the slave side should be shrouded to prevent the danger of wrong insertion. Connector A can be configured as a standard Feature Connector, VIP, or Connector A (video) of VMI 1.4. On power up, the graphics chip (master) is configured in the standard feature connector mode. All VIP slaves must be disabled and tri-stated on power up.

VIP CONNECTOR A (26-pin Dual Row Header, 0.100 in. centers)						
Standard Feature Connector		**VIP Mode**		**Standard Feature Connector**		**VIP Mode**
Pin #	**Signal Name**	**Signal Name**		**Pin #**	**Signal Name**	**Signal Name**
Z1	Ground	Ground		Y1	P0	VID[0]
Z2	Ground	Ground		Y2	P1	VID[1]
Z3	Ground	Ground		Y3	P2	VID[2]
Z4	EVIDEO#	HAD[1]		Y4	P3	VID[3]
Z5	ESYNC#	HAD[0]		Y5	P4	VID[4]
Z6	EDCLK#	HCTL		Y6	P5	VID[5]
Z7	N/C	SCL		Y7	P6	VID[6]
Z8	Ground	Ground		Y8	P7	VID[7]
Z9	Ground	Ground		Y9	DCLK	PIXCLK
Z10	Ground	Ground		Y10	BLANK#	VIPCLK
Z11	Ground	Ground		Y11	HSYNC	N/C
Z12	N/C	VIRQ#		Y12	VSYNC	N/C
Z13	N/C	SDA		Y13	Ground	Ground

VIP CONNECTOR B (14-pin Dual Row Header, 0.100 in. centers)						
Standard Feature Connector		**VIP Mode**		**Standard Feature Connector**		**VIP Mode**
Pin #	**Signal Name**	**Signal Name**		**Pin #**	**Signal Name**	**Signal Name**
Z1	-	+3.3V		Y1	-	+3.3V
Z2	-	+3.3V		Y2	-	+3.3V
Z3	-	Ground		Y3	-	Ground
Z4	-	+5V		Y4	-	+5V
Z5	-	+12V		Y5	-	VRST#
Z6	-	SCLK		Y6	-	Ground
Z7	-	LRCLK		Y7	-	PCMDATA

4.2 I²C Bus

I²C is not part of the VIP specification. The I²C bus signals SCL and SDA, which are shown in the Connector A signal table as pin Z7 and Z13, are merely recommendations. There are many variations of I²C implementations today, which can lead to incompatibility. We strongly recommend the Varying Voltage implementation with up to 5V support. This offers maximum compatibility among many I²C devices.

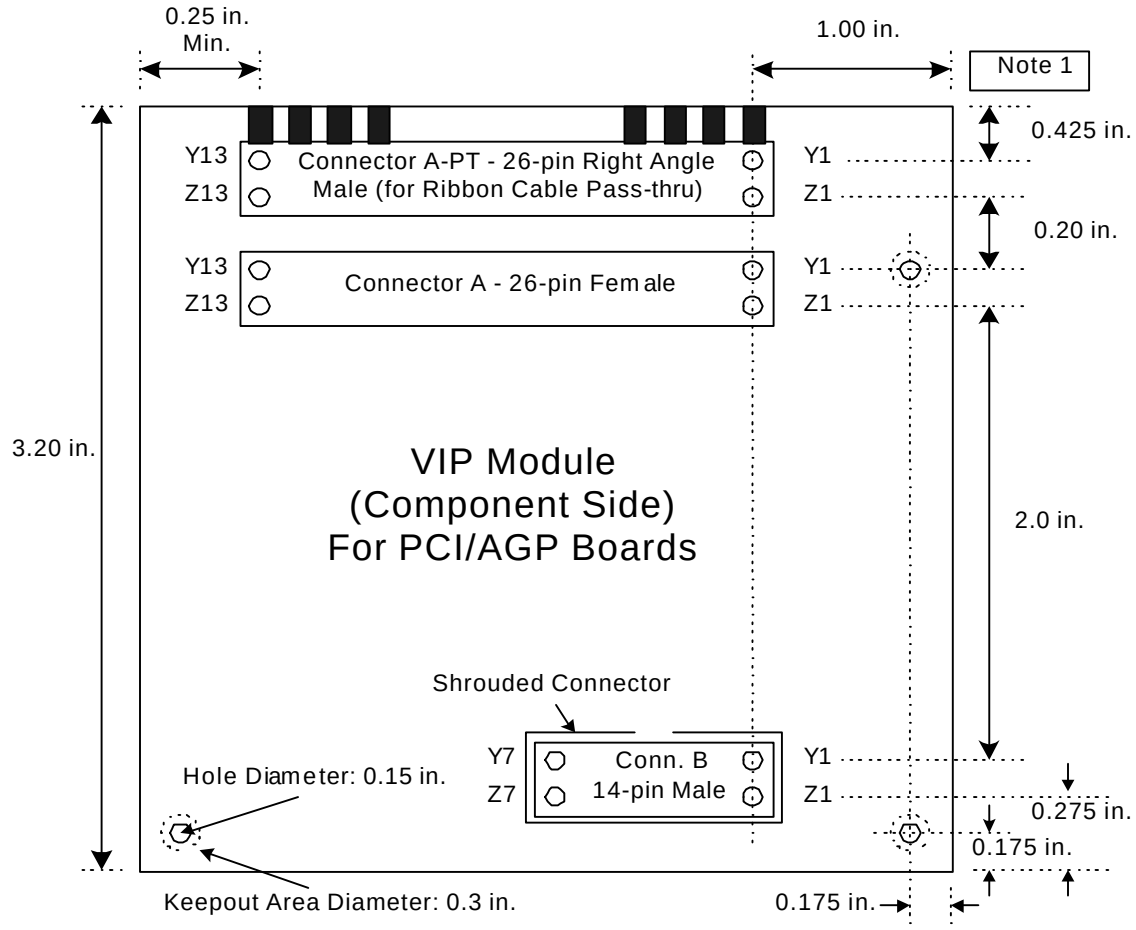
4.3 I²S Digital Audio Output

A standard I²S audio port is included as part of Connector B. On power up no VIP slave can drive this I²S port until activated by software. Software must make sure that at any time only one VIP slave drives this bus. This I²S can be digitally mixed with the sound subsystem.

4.4 Ribbon Cable Specification

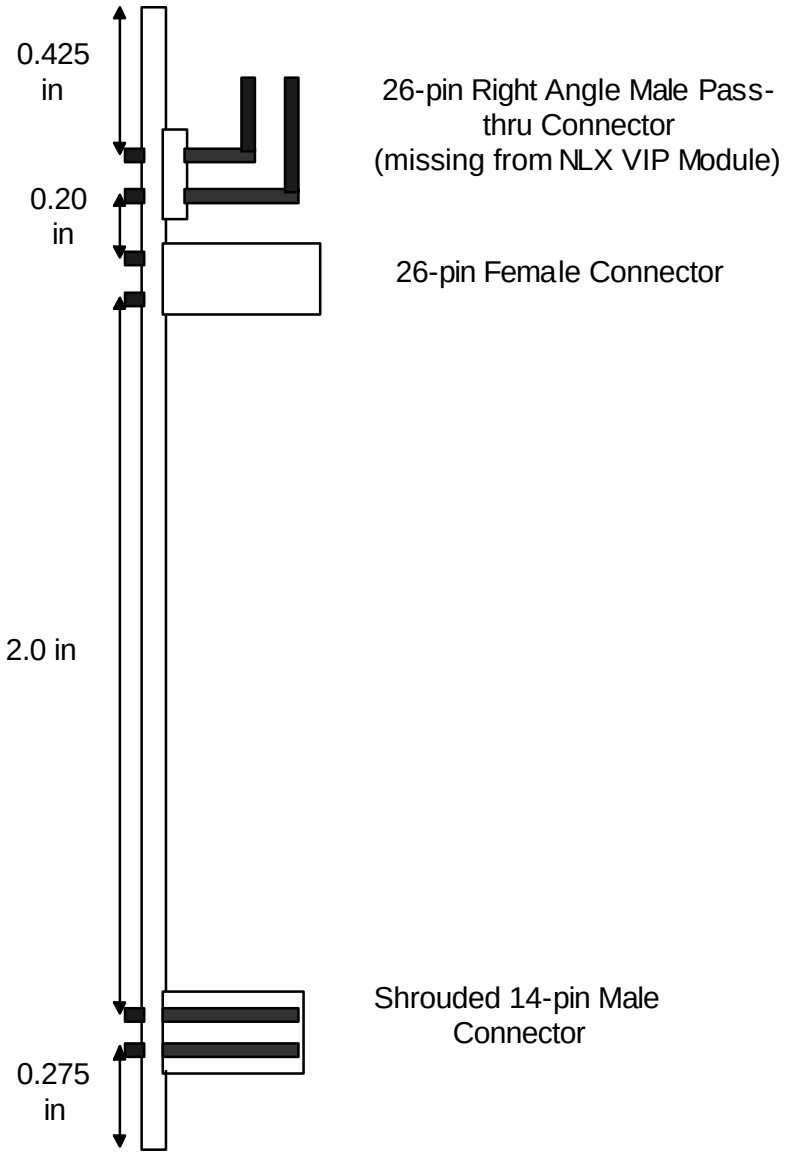
VIP supports the same ribbon cable as specified in the VESA STANDARD VGA PASS-THROUGH CONNECTOR SPECIFICATION (Document VS890803).

4.5 VIP Module Mechanical Examples (PCI or AGP)

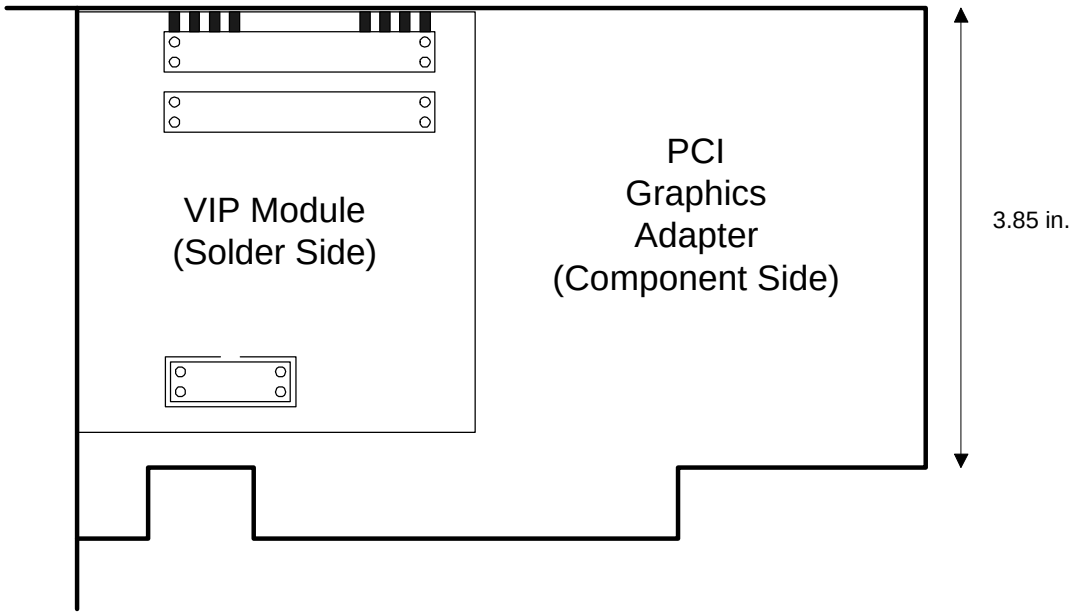


Note 1. This edge of the board is where the IO connectors of the VGA adapter are usually located. Care must be used in placing components in this area to avoid interference with the VGA card connectors.

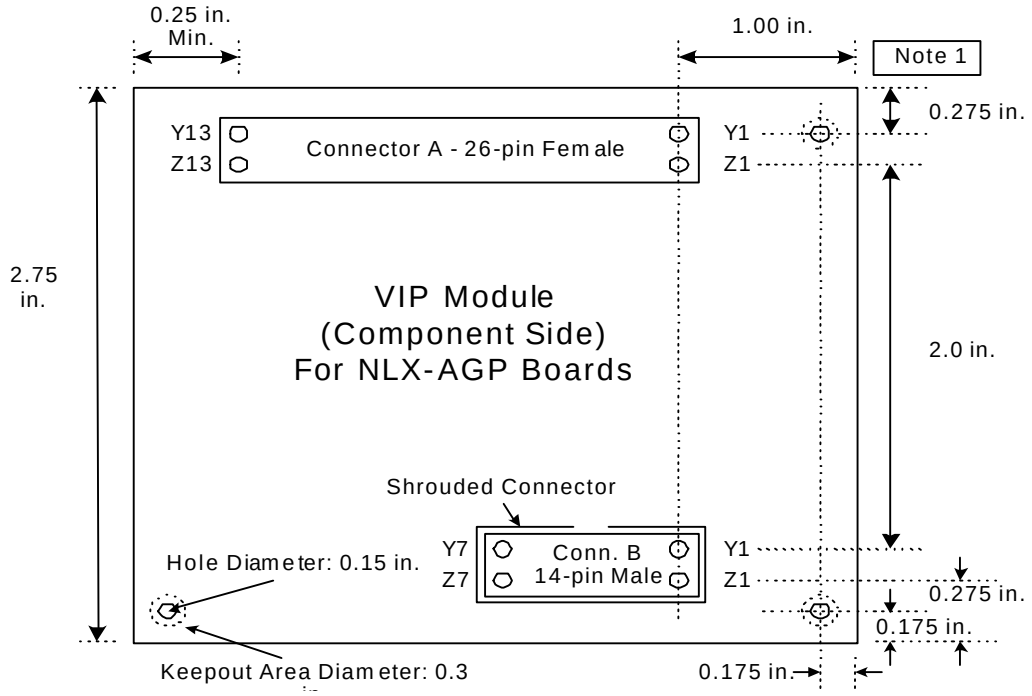
4.6 VIP Module Side View (PCI or AGP)



4.7 Graphics Adapter/VIP Module Attachment Example

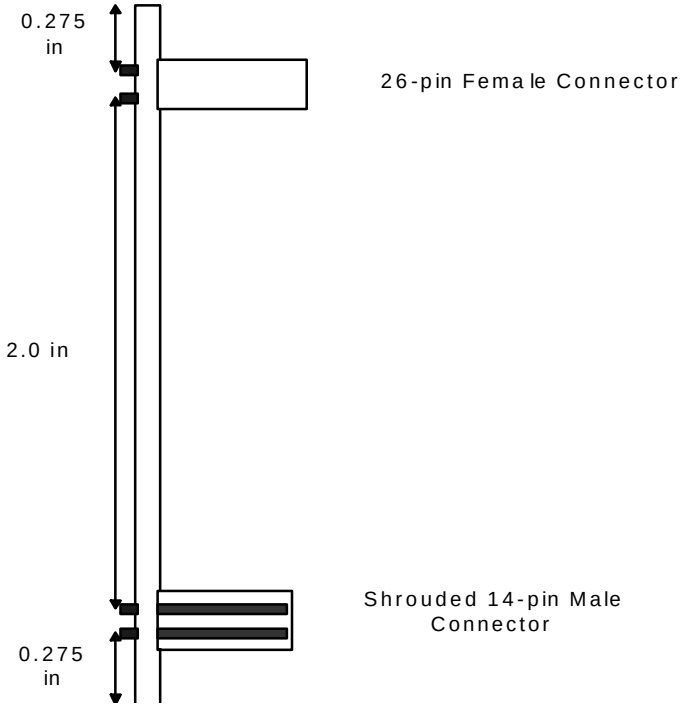


4.8 VIP Module Mechanical Examples (NLX AGP)



Note 1. This edge of the board is where the IO connectors of the VGA adapter are usually located. Care must be used in placing components in this area to avoid interference with the VGA card connectors.

4.9 VIP Module Side View (NLX AGP)



4.10 Component Height Considerations

VIP is designed as an OEM standard, and not a retail compatible standard. It is assumed that each OEM has complete control over both the graphics and the video subsystems. It is the OEM's responsibility to ensure that there is no component height conflict between the two subsystems. This decision was made to give OEMs maximum implementation flexibility.

5 Protocol Rules

5.1 Retry, Wait and Data Phases

1 Retry, Wait and Data Phases are always four clocks long.

The definition of the HCTL bus depends on the clock cycle, which are defined by the following rules:

2 Cycle 1 - VFRAME, VSTOP#

2.1 VIP devices drive HCTL low during cycle 1 to end the transfer at the end of the current phase. They leave HCTL tri-stated to continue transferring data.

2.2 VIP devices are not permitted to drive HCTL high during cycle 1 of a Retry, Wait or Data phase.

2.3 The slave can terminate the transfer during any Retry, Wait or Data phase.

2.4 Except for timeout conditions, the master can only terminate transfers during Data phases.

2.5 When a timeout condition occurs, the master can terminate a bus transfer during a Wait phase. Transfers terminating due to a timeout are not repeated.

2.6 The master is not permitted to terminate a bus transfer during a Retry phase.

3 Cycle 2 - DTACK#

3.1 VIP devices may only drive HCTL low or leave it tri-state in cycle 2, they are not permitted to drive HCTL high during cycle 2.

3.2 HCTL is driven low in cycle 2 by the slave to indicate that it is ready to accept the next data transfer.

3.3 When HCTL is driven low in cycle 1, it is ignored in cycle 2 and the transfer terminates at the end of the current phase.

3.4 When the transfer is not terminated during cycle 1 and the slave does not drive HCTL low during cycle 2, the transfer is extended by adding a Wait phase.

3.5 When the master terminates the transfer by driving HCTL low during cycle 1, it continues to drive HCTL low in cycle 2 internally while externally it is tri-stated.

3.6 When the slave terminates the transfer by driving HCTL low during cycle 1, it continues to drive HCTL low in cycle 2.

3.7 VIP master is always tri-stated during cycle 2. However, internally it must drive HCTL low in cycle 2. This is to prevent the slight possibility of bus conflict in case the master drives HCTL low in cycle 3. Bus conflict can occur if the master transition from high to low at the beginning of cycle 3 while it turns on the output driver. The temporary high value can conflict with the slave, which may be in the low state and is tri-stating its output driver at the beginning of cycle 3.

4 Cycle 3 - Recovery

4.1 During cycle 3, the master may drive HCTL low or tri-state. The slave may drive HCTL high or tri-state.

4.2 If the transfer is terminated in cycle 1(by the master or the slave), the master drives HCTL low during cycle 3.

4.3 When the transfer is terminated during cycle 1 and the slave drives HCTL low in cycle 2, it tri-states HCTL in cycle 3. The slave must continue to drive a low value during the tri-state turn off period.

4.4 When the transfer is not terminated and the slave drives HCTL low during cycle 2, it drives HCTL high in cycle 3 prior to tri-stating it in cycle 4.

5 Cycle 4 - Tri-state

5.1 During cycle 4 the slave never drives HCTL.

5.2 When the slave drives HCTL high in cycle 3, it continues to drive high during the output driver turn-off time.

5.3 When the transfer has been terminated (by either the master or slave), the master drives HCTL low in cycle 4.

5.2 HCTL Tri-state Rules

In general, when a master or slave device transitions from driving HCTL to tri-state, it is required to maintain the same driving condition during the output driver turn-off time to prevent HCTL from being left in an undetermined state. Specific examples include but are not limited to:

1 The master is required to continue driving a high value (while its output driver is turned off) during the second cycle of a command phase to insure that the high value is maintained during tri-state.
2 A slave device is required to continue driving HCTL low (while its output driver is turned off) during cycle 3 of a Retry, Wait or Data phase when it drives HCTL low in cycle 2 and the transfer is terminated.
3 A slave device is required to continue driving a high value (while its output driver is turned off) during the cycle 4 of a Retry, Wait or Data phase when it drives HCTL high during cycle 3.

5.3 HAD Tri-state Rules

1 In general, a Turn-Around phase is always required when switching devices on the HAD bus.
2 The master tri-states the HAD bus no later than the cycle 3 of the Retry phase when the requested operation is a VIP slave read.
3 During read transfers, the slave may begin driving HAD on the cycle 4 of the Retry phase unless it terminates the transfer with a Retry (by driving HCTL low in cycle 1).
4 The slave is not allowed to drive the HAD bus when it terminates the transfer during the Retry phase.
5 During Wait phases, the current owner (master for writes, slave for reads) continues to drive the HAD bus, however, no data is transferred.
6 The master is not required to insert a Turn-Around phase following a Retry, even if the retried operation is a slave read.
7 Any read transfer that is not terminated during the Retry phase must follow the last data transfer with a one-clock turn-around phase. The slave tri-states its HAD driver during the turn-around phase.
8 The turn-around phase is optional when the transfer is a write and following read transfers that are terminated by the slave during the Retry phase.

5.4 The Idle Phase

Any cycle that is not part of a transfer is an Idle Phase. The Master must drive HCTL low during the Idle

Phase. After system reset, the Master must first enter the Idle Phase by driving HCTL low.

5.5 The Command Phase

1 Cycle 1 – CMD1

1.1 The VIP Master starts a transfer by driving HCTL high for one cycle.

1.2 The Master outputs to HAD[1:0] the most significant two bits of the Command Byte CMD[7:6].

1.3 The Slaves should tri-state HCTL during this cycle.

2 Cycle 2, 3, 4 – CMD2, CMD3, CMD4

2.1 The VIP Master tri-states HCTL during these cycles. HCTL is held high by the pull-up circuit.

2.2 The Master outputs to HAD[1:0] the remaining six bits of the Command Byte, CMD[5:4], CMD[3:2], CMD[1:0], respectively.

2.3 The Slaves should continue to tri-state HCTL during these cycles.

5.6 The Address Phase

The Address Phase exists for a register transfer only. It immediately follows the Command Phase.

1 Cycle 1 – ADR1

1.1 The Master outputs to HAD[1:0] the most significant two bits of the Address Byte ADR[7:6].

1.2 The Master and the Slaves should tri-state HCTL during this cycle.

2 Cycle 2, 3, 4 – ADR2, ADR3, ADR4

2.1 The VIP Master and Slaves tri-state HCTL during these cycles. HCTL is held high by the pull-up circuit.

2.2 The Master outputs to HAD[1:0] the remaining six bits of the Address Byte, ADR[5:4], ADR[3:2], ADR[1:0].

5.7 The Decode Phase

1 The Decode Phase is a one cycle phase to give the Slaves enough time to decode the Command Byte and/or the Address Byte. It follows either the Command Phase (for FIFO transfer), or the Address Phase (for Register Transfer).

2 The Master may either drive or tri-state HAD[1:0]. Both the Master and the Slave should tri-state HCTL.

6 Timing Diagrams

In the following timing diagrams, the HCTL signal is shown as the combination of two conceptual signals, HCTL_master and HCTL_slave. HCTL_master defines the behavior of the master driven HCTL while HCTL_slave demonstrates slave driven behavior. The effect of the two together, along with the required pullup resistor is shown as HCTL.

The first diagram is explained in detail. Since the basic timing is very similar for register and FIFO accesses, the discussion of the remaining diagrams is focused on their differences.

6.1 Figure 1: Register read, 1 byte, no wait states, master terminated

6.1.1 Command Phase

The master initiates the transfer by driving HCTL high during cycle 1 of the Command phase. During cycle 2, the master tri-states its HCTL driver while continuing to drive it high. This prevents HCTL from being driven low during the drivers turn off time. A pull-up resistor or active bus keeper keeps HCTL high during periods when no VIP device is driving it. The Command phase is used to transfer the Command/Address byte, two bits per clock, starting with the most significant bits.

6.1.2 Address Phase

Since this example is a register transfer the Command phase is followed by an Address phase. During this phase, the least significant 8-bits of the register address are multiplexed onto HAD[1:0].

6.1.3 Decode Phase

A one cycle Decode phase is inserted to give slave devices time to decode the address before entering the Retry phase. This allows the slave to terminate the sequence without transferring data. During this phase, the HAD bus is undefined.

6.1.4 Retry Phase

The Retry phase is the data decision phase for the first byte to be transferred.

During cycle 1, the slave does not drive driving HCTL low, indicating that it intends to complete the transfer.

During cycle 2, the slave drives HCTL low, indicating that it will provide the requested byte without any additional delay. The data is transferred during the following Data phase.

During cycle 3 the slave drives HCTL high in preparation for tri-stating it in cycle 4. In cycle 4 the slave tri-states its HCTL driver while continuing to drive a high value. This is indicated by the “tri-state, 1” designation in the diagram. The slave must continue to drive 1 to avoid HCTL being pulled down during the time required to tri-state its output driver.

The data on HAD is undefined during Retry phases, however the master must is tri-state its HAD drivers no later than cycle 3, giving the slave an extra clock to enable its drivers before entering the Data phase.

6.1.5 Data Phase

The first byte is transferred during the data phase while HCTL is used to make a decision for the next byte. In cycle 1 the master drives HCTL low, indicating that it wants to end the transfer after the current Data phase.

Since the slave has not had a clock to respond to the master's request, it is allowed to drive HCTL low in cycle 2. Since the master has already indicated its intention to terminate the transfer, the value driven on HCTL during cycle 2 is ignored.

In cycle 3 the slave responds to the master terminating the transfer by tri-stating its HCTL driver. The slave continues to drive low while tri-stating to avoid conflicting with the master. This is indicated by the "tri-state, 0" designation on the diagram. The master continues to drive HCTL low during the remainder of the Data Phase 5 (clocks 2-4).

6.1.6 Turn-Around Phase

A one clock delay is required between the last data transfer and the start of the next command phase to allow the slave to tri-state its HAD pins.

6.2 Figure 2: FIFO burst write, master terminated

Figure 2 illustrates a two-byte FIFO burst write cycle, which is terminated by the master. It is similar to Figure 1 but because the transfer is to a FIFO port, the Address Phase is eliminated. This reduces FIFO transfer overhead by four clocks per transfer.

During write operations the master continues to drive the HAD bus during the Retry phase although the data is undefined. In cycle 2 of the retry phase the slave indicates its ability to complete the first transfer without wait states causing the Retry phase to be followed by a Data phase.

During the first Data phase, the master leaves HCTL tri-state, indicating that there is more data to transfer. Since the slave doesn't drive HCTL during clock 1 (indicating it wants to terminate) and drives HCTL low during clock 2, an additional Data phase is inserted and a second byte is written to the slave.

During the second Data phase, the master drives HCTL low to terminate the transfer. This causes the transfer to terminate after the second Data phase.

Since the transfer is a write operation, the Turn-Around phase is not required and the master is free to initiate another transfer on the cycle following the second Data phase.

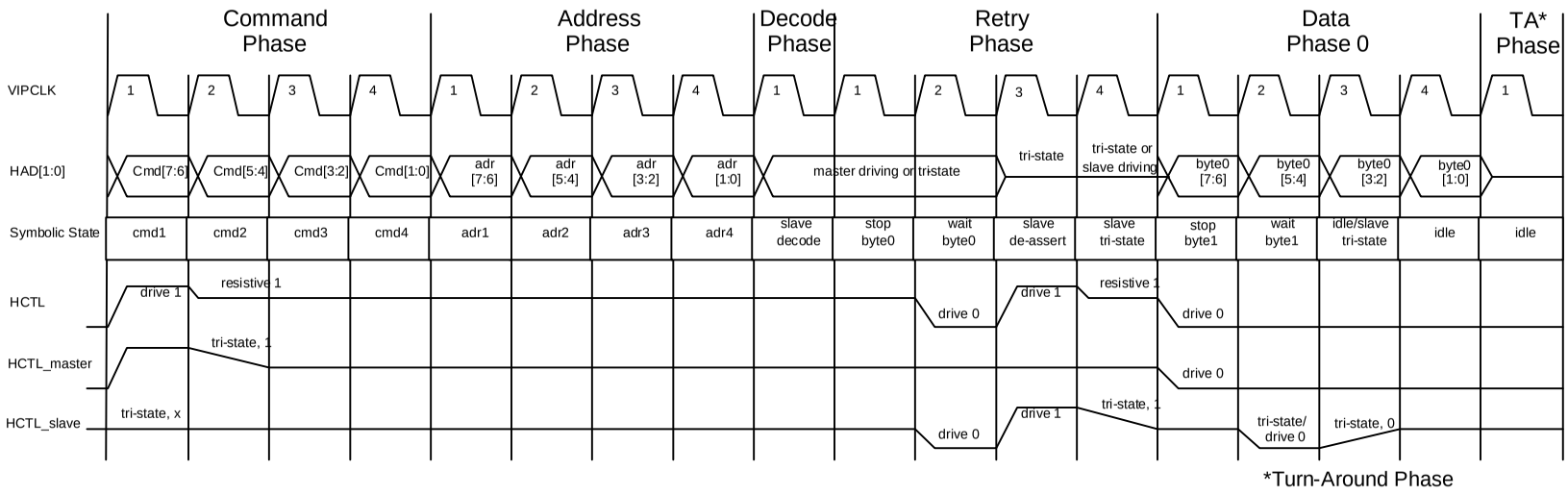


Figure 1. One byte register read, no wait states, master terminated

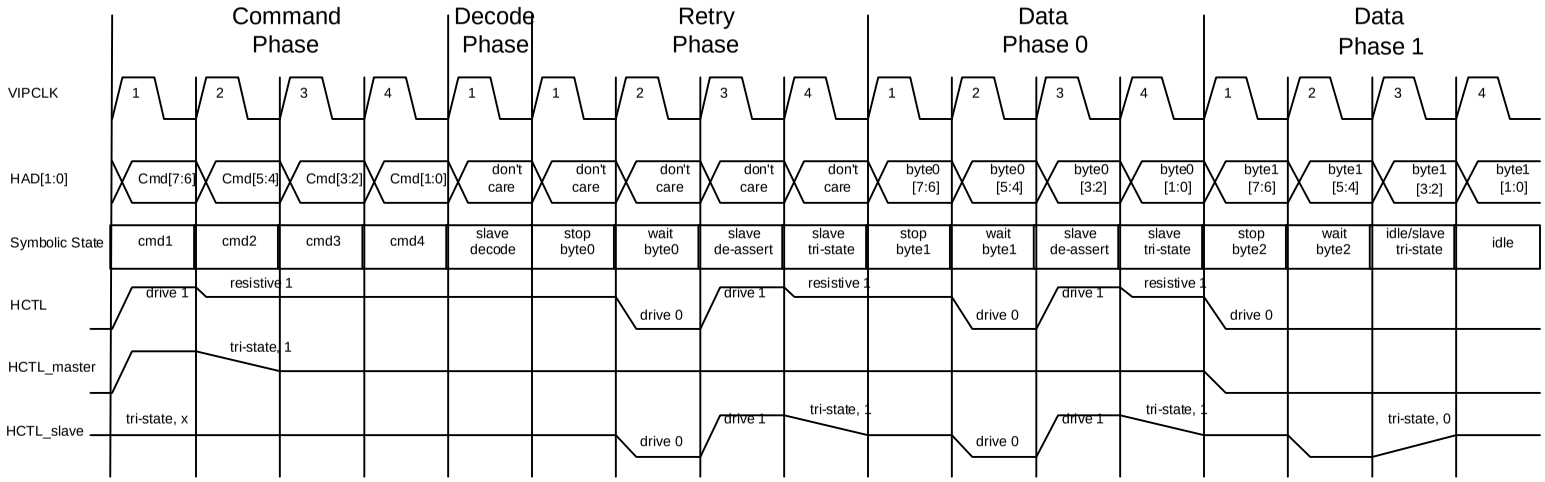


Figure 2. Two-byte FIFO write, master terminated

6.3 Figure 3: FIFO write, retry (slave terminated)

Figure 3 illustrates a FIFO write to a port that is unable to take the transfer. This is an example of the hardware polling mechanism. Since the slave is unable to accept more data, it terminates the cycle by driving HCTL low during cycle 1 of the Retry phase.

It also drives HCTL low during cycle 2 because the master cannot respond until cycle 3.

In cycle 3 the slave tri-states its HCTL pin (while continuing to drive low) as indicated by the “tri-state, 0” designation. In cycles 3 and 4 the master drives HCTL low.

6.4 Figure 4: FIFO read, retry (slave terminated)

Figure 4 is an example of a FIFO read when the selected port does not have any data to transfer. The timing is similar to Figure 4 with the exception of the state of the HAD bus.

In cycle 3 of the Retry phase the master tri-states the HAD bus in anticipation of the slave driving read data. The slave does not actually drive HAD because it terminates during the Retry phase. This saves the overhead for the Turn-Around phase when hardware polling a FIFO read port.

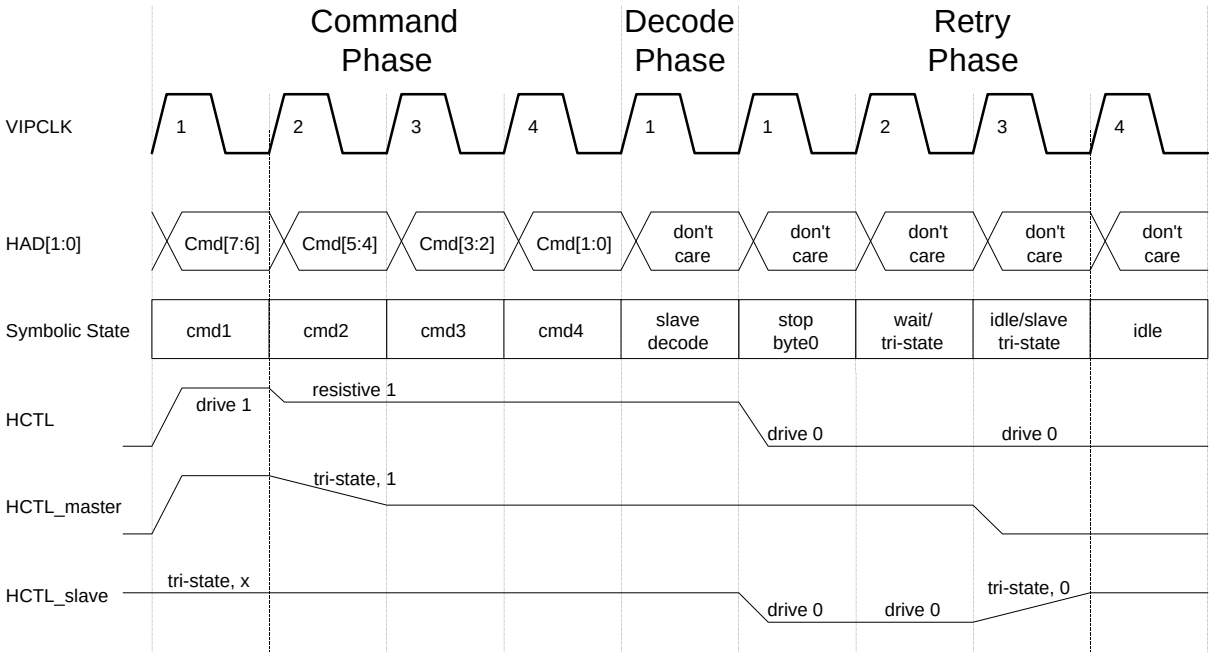


Figure 3. FIFO write - Retry

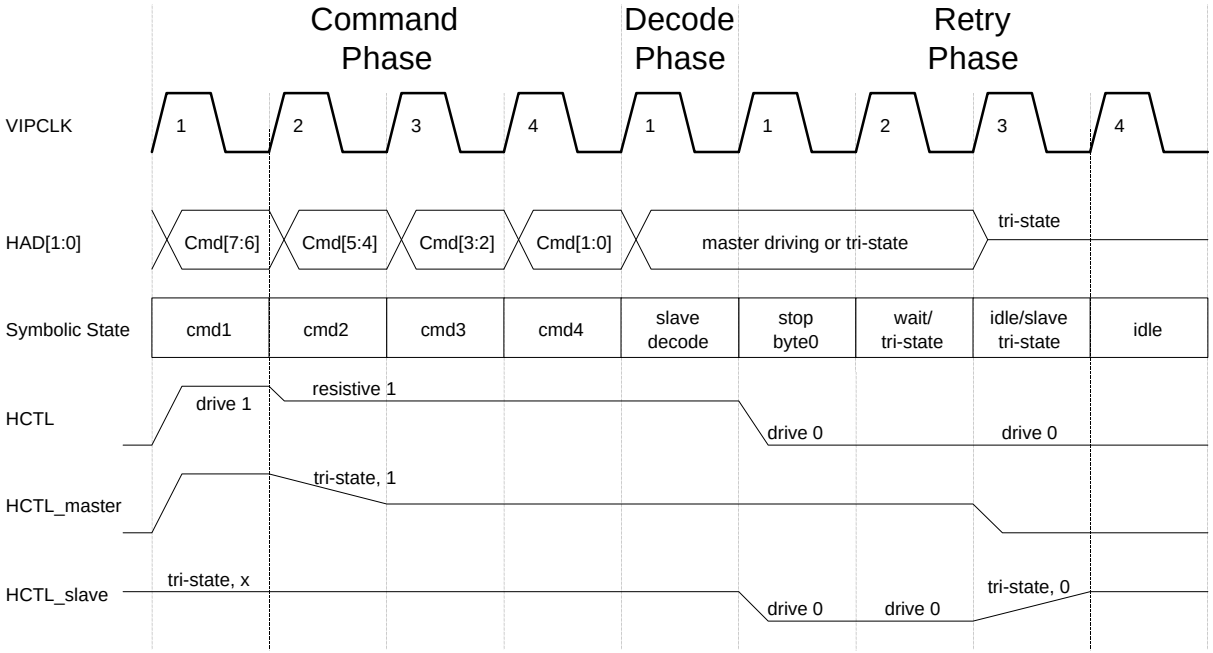


Figure 4. FIFO read - Retry

6.5 Figures 5 and 6: Register Accesses with wait states, master and slave termination

Figure 5 illustrates a register read with wait states terminated by the slave. Figure 6 illustrates a register write with wait states terminated simultaneously by the master and slave.

In cycle 2 of the Retry phase HCTL is not driven by the slave, leaving it in the high state. This causes the master to follow the Retry phase with a Wait phase.

During cycle 2 of the Wait phase the slave drives HCTL low, telling the master to transition to the Data phase.

When HCTL is not driven in cycle 2 of Data phase 0, it is followed with another Wait phase. Again the slave drives HCTL low during cycle 2 of the Wait causing the transition to Data phase 1.

On cycle 1 of Data phase 1 the slave drives HCTL low to terminate the transfer following Data phase 1. Although the master has additional data to transfer in Figure 6, it recognizes that the slave wants to terminate the transfer and drives HCTL low during cycles 3 and 4. In Figure 7, both the master and slave terminate the transfer by driving HCTL low during cycle 1 of Data phase 1.

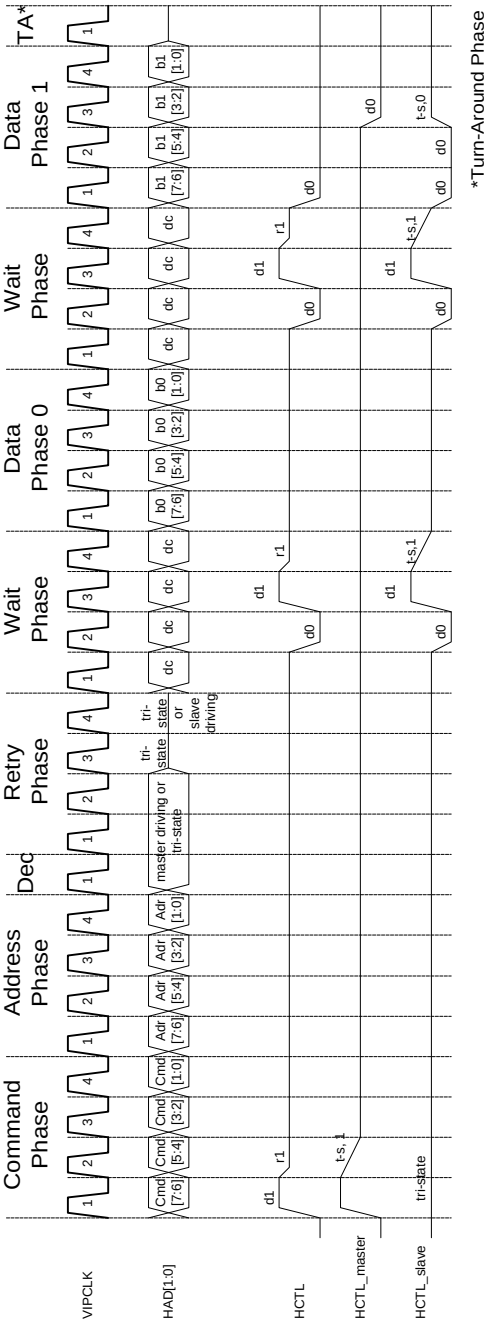


Figure 5. Register read with wait phase, slave terminated

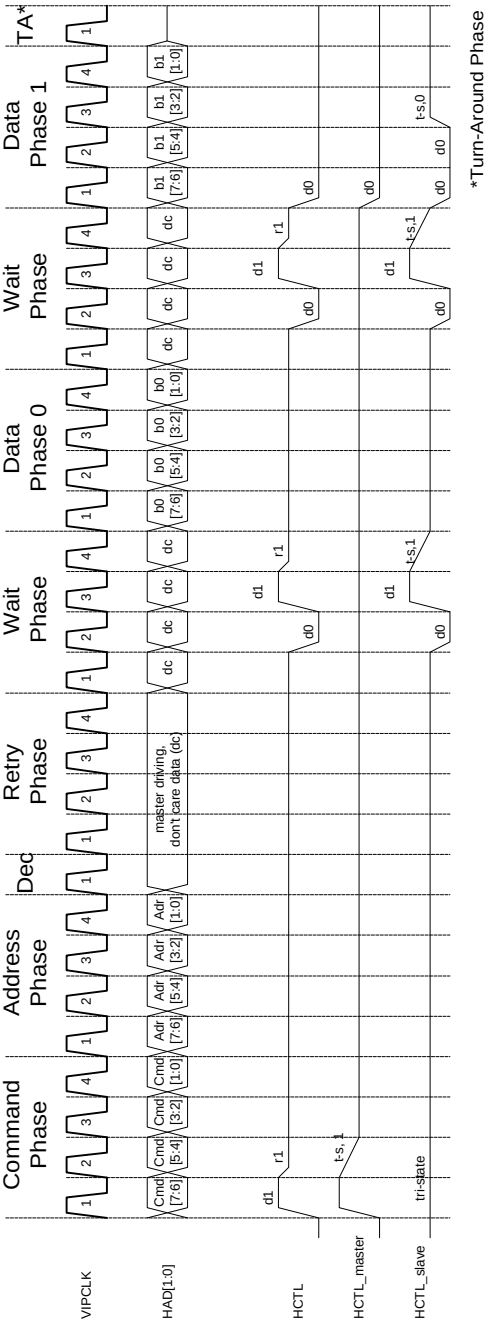


Figure 6. Register write with wait phase, master and slave terminated

6.6 Figures 7 and 8: FIFO burst transfers, master and slave termination

Figure 7 illustrates a four-byte FIFO read, terminated by the slave. Figure 8 illustrates a seven-byte FIFO write, terminated simultaneously by the master and slave. These examples are included for completeness however they do not illustrate any new conditions.

6.7 Figures 9 and 10. Timeout Cycles

Figures 9 and 10 illustrate Timeout timing for register and FIFO accesses. This is the only condition where the master is permitted to terminate a transfer during a Wait phase. When the timeout period is exceeded without the slave accepting or terminating the transfer, the master drives HCTL low during cycle 1 of the Wait phase to terminate the transfer. This is an error condition that should be reported to the driver. Like other read transfers, the master is required to follow the last Wait phase with a Turn-Around phase. The transfer is not repeated.

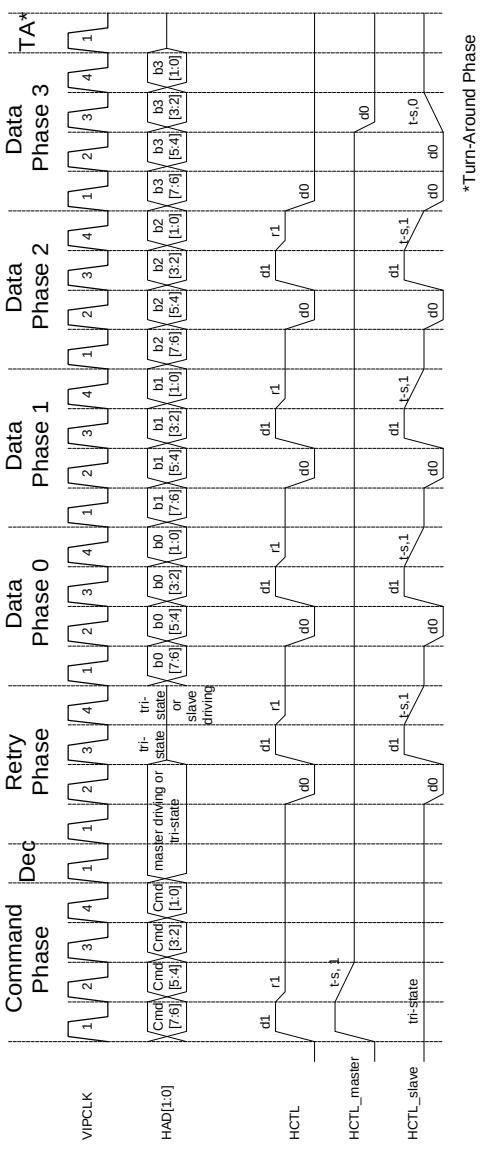


Figure 7. FIFO burst read, slave terminated

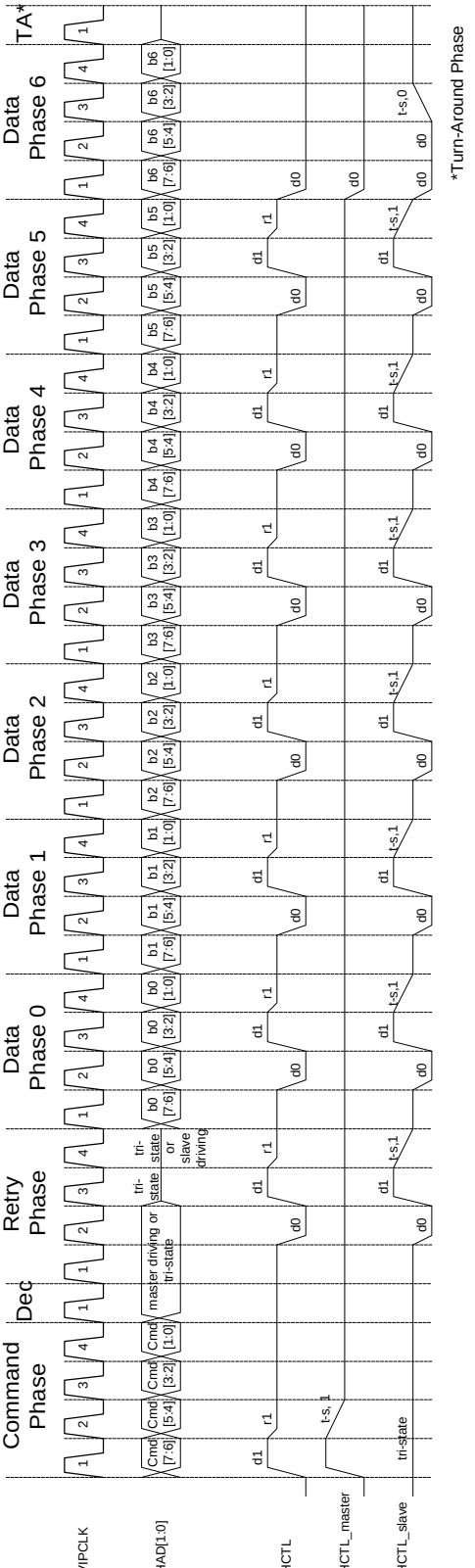


Figure 8. FIFO burst write, master and slave terminated

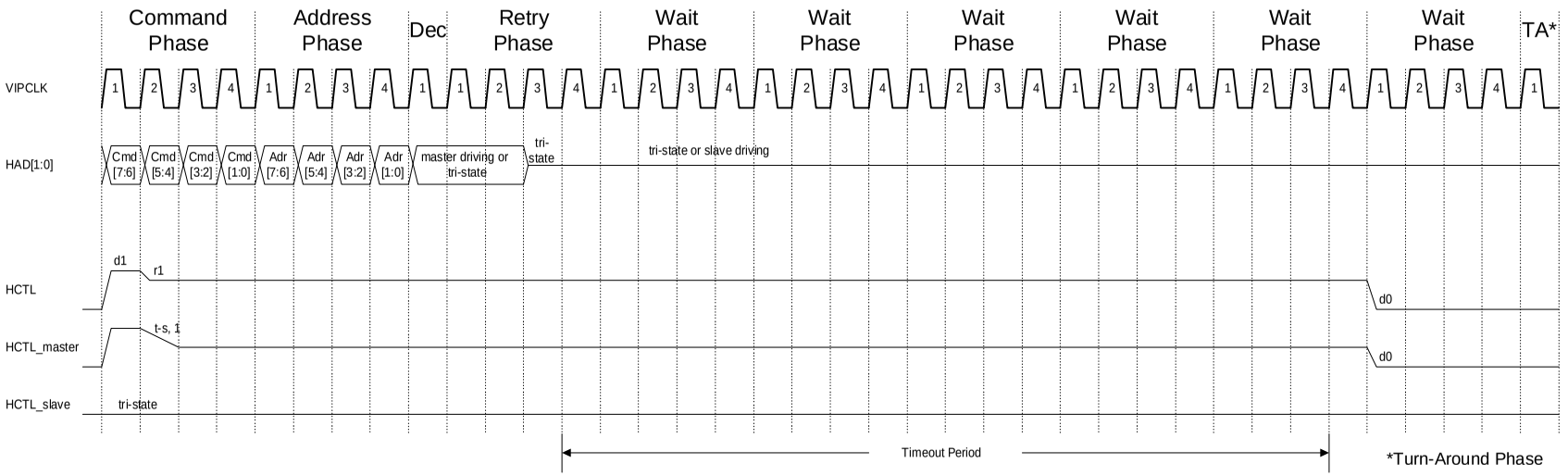


Figure 9. Register read, timeout

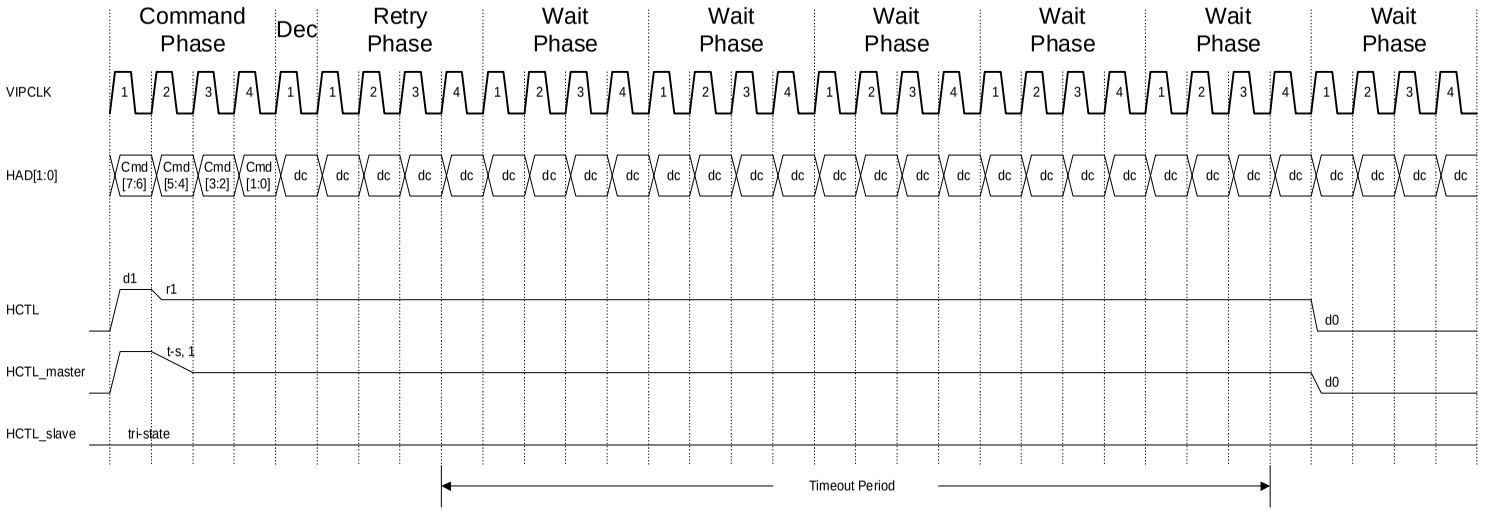
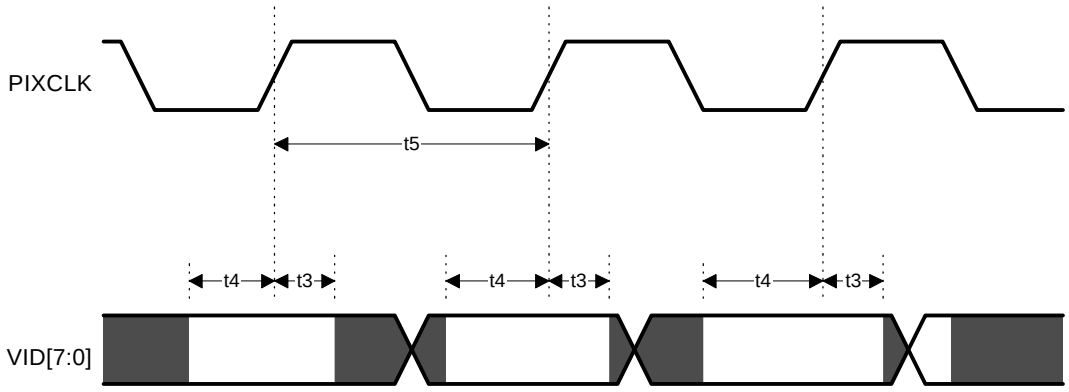


Figure 10. FIFO write, timeout

6.8 VIP Video Port



	Description	MIN (ns)	MAX (ns)
t3	VID hold after PIXCLK HIGH	0	-
t4	VID setup to PIXCLK HIGH	5	-
t5	Cycle Time	33	-

6.8.1 YUV (YCbCr) Byte Ordering

1st byte	2nd byte	3rd byte	4th byte	5th (next DWORD)	6th byte	7th ...
U[7..0]	Y0[7..0]	V[7..0]	Y1[7..0]	U[7..0]	Y0[7..0]	V[7..0]
Cb[7..0]	Y0[7..0]	Cr[7..0]	Y1[7..0]	Cb[7..0]	Y0[7..0]	Cr[7..0]

7 Electrical Specification

Upon power-up, the VIP module output, and other on-board VIP devices should be tristated until a request from the motherboard signals the module to begin driving the bus. The VIP module may continue to drive the bus until a request from the motherboard signals the module to tristate its outputs.
Maximum load capacitance per pin should not exceed 70pF (I^2C Port excluded). This is based on the assumption that each device has 15pF, the ribbon cable is 10pF (15x4+10=70pF).
All signals are 3.3V-compliant. 5V tolerance is not required.

7.1 VIPCLK - DC Requirements

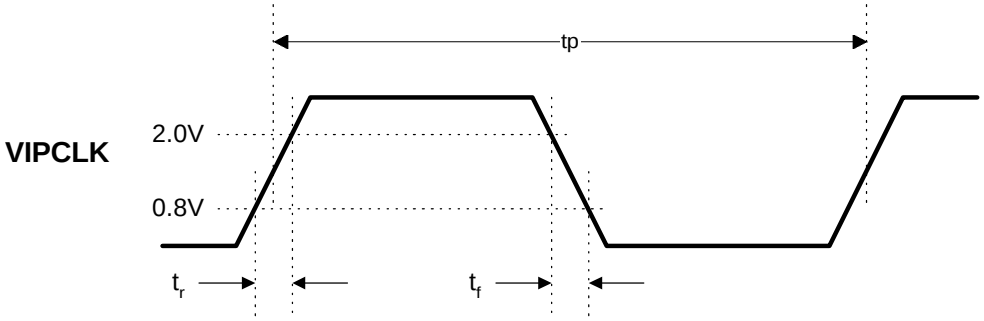
Symbol	Description	Condition	Min	Max	Units
Voh	Driver high level output voltage	Iout=Ioh(min)	2.4		Volts
Vol	Driver low level output voltage	Iout=Iol(min)	-0.5	0.4	Volts
Ioh(DC)	Driver high level output current	Vout-Voh(min)	-12		mA
Iol(DC)	Driver low level output current	Vout=Vol(max)	12		mA
Ioh(AC)	Switching to high output current	0<Voh<1.4*	-75		mA
Iol(AC)	Switching to low output current	Vol>2.2V	80		mA
Vih	Receiver high level input voltage		2	3.9	Volts
Vil	Receiver low level input voltage		-0.5	0.8	Volts
Iih	Receiver high level input current	Vih=2.4		100	uA
Iil	Receiver low level input current	Vol=0.8V		-100	uA

*AC Switching currents are specified as a design aid in choosing the clock driver circuit. The number given here is a minimum peak value reached during the switching time of CLK. In this case the transition time includes the time to transition from one logic level to another.

7.2 All signals except VIPCLK - DC Requirements

Symbol	Description	Condition	Min	Max	Units
Voh	Driver high level output voltage	Iout=Ioh(min)	2.4		Volts
Vol	Driver low level output voltage	Iout=Iol(min)	-0.5	0.4	Volts
Ioh(DC)	Driver high level output current	Vout-Voh(min)	-4		mA
Iol(DC)	Driver low level output current	Vout=Vol(max)	6		mA
Ioh(AC)	Switching to high output current	0<Voh<1.4	-16		mA
Iol(AC)	Switching to low output current	Vol>2.2V	32		mA
Vih	Receiver high level input voltage		2	3.9	Volts
Vil	Receiver low level input voltage		-0.5	0.8	Volts
Iih	Receiver high level input current	Vih=2.4		+/-200	uA
Iil	Receiver low level input current	Vol=0.8V		+/-200	uA

7.3 AC Requirements



Description	MIN (ns)	MAX (ns)
Tr Slew Rate	-	2.5
Tf Slew Rate	-	2.5
VIPCLK Cycle to Cycle Jitter		200ps
Tp VPICLK Period	30	40
VIPCLK Duty Cycle (at 1.5V)	40%	60%
VIPCLK to Output	-	11
Data to VIPCLK Setup Time	5	-
VIPCLK to Data Hold Time	0	-
HCTL low before VIPCLK active after power down	100	-

Test condition: 70pf, including probe capacitance, between output and ground. Additionally there is a 500 ohms load between output and ground.

8 VIP Video Format

8.1 Overview

VIP video port is based on a subset of ITU-R-656, with extensions that are suited for PC applications. While this section describes ITU-R-656 as it relates to VIP, it does not cover ITU-R-656 in its entirety. For more details or further clarification please refer to the ITU-R-656 Specification.

The VIP Video Port transports various types of real time signal streams:

- active video, represented in digital baseband components YUV (Cb-Y-Cr-Y-), i.e. the " visible" part of a video signal, as defined by an acquisition window, this signal can be scaled in three dimensions: horizontal, vertical, field/frame rate,
- decoded (sliced) VBI data of separately selected VBI lines,
- "raw" ADC samples of digital CVBS "composite video" during selected VBI lines,
- digital audio, one or multiple stereo channels, PCM coded,
- other real time capture related data.

The active video data, as selected by an acquisition definition as active pixels per line and active lines per field, represent the basic signal stream.

These signals are interleaved into one single stream. They are synchronized and separated by unique ITU-R-656 "header" codes. Active video and raw ADC samples are enveloped by SAV and EAV codes. Sliced VBI data and digital audio stream are transported as ANC (ancillary) data blocks during horizontal or vertical blanking time.

The following table shows the VIP video control codes:

Event description	**Inserted header and 'raster' reference code**			
	Task A		Task B	
Field ID	Odd	Even	Odd	Even
Next pixel is **first** pixel of any active "acquired" line	FF-00-00- **8** *0*	FF-00-00- **C** *7*	FF-00-00- **0** *E*	FF-00-00- **4** *9*
(Bits TFVH rrrr[1])	**1000** *0000*	**1100** *0111*	**0000** *1110*	**0100** *1001*
Previous pixel was **last** pixel of any active line, but not last line,	FF-00-00- **9** *D*	FF-00-00- **D** *A*	FF-00-00- **1** *3*	FF-00-00- **5** *4*
(Bits TFVH rrrr)	**1001** *1101*	**1101** *1010*	**0001** *0011*	**0101** *0100*
Previous pixel was **last** pixel of the last active line only,	FF-00-00- **B** *6*	FF-00-00- **F** *1*	FF-00-00- **3** *8*	FF-00-00- **7** *F*
(Bits TFVH rrrr)	**1011** *0110*	**1111** *0001*	**0011** *1000*	**0111** *1111*
Dummy Byte Between SAV and EAV	00	Dummy Byte. Empty clock cycle carries no data. Don't capture; don't increment pointer. (Note that "00" within SAV-EAV is not allowed in ITU-R-656.)		
Ancillary data block – VBI Data	00-FF-FF- DID- SDID- NN- IDID-IDID- ..... CS FB	Preamble DID = ITU-data type, SDID = secondary identifier, NN = Data Count = number of D words IDID = internal data identifier .... VBI data … CS = check sum FB = Fill Byte		
Ancillary data block – Audio Data	00-FF-FF- DID- BN- NN- ..... CS FB	Preamble DID = ITU-data type, BN = Block Number, NN = Data Count = number of D words .... Audio data … CS = check sum FB = Fill Byte		

[1] ITU-R-656 Protection Code. "Don't care" for a VIP receiver.

8.2 SAV and EAV codes for active video

ITU-R-656 specifies the following codes for active video:

Luminance Y:
black = 10 hex = 16 dec, white = EB hex = 235 dec.

Chrominance Cb, Cr:
no color = 80 hex = 128 dec, 100% saturation = 10..F0 hex = 16..240 dec.

Active video is sampled as YUV 4:2:2, and is transmitted as a byte serial stream of Cb-Y-Cr-Y- ... .

The codes 00 hex and FF hex are prohibited as video sample codes, and reserved for header and synchronization purposes. The codes from 01...0F hex and F1...FE hex are reserved as signal processing margin. They should not be clipped.

Each video line is enveloped by a pair of SAV (Start of Active Video) and EAV (End of Active Video) reference codes. The SAV and EAV codes consist of a 4-byte sequence of **FF-00-00-RP**. The fourth byte RP carries the raster reference information. The reference byte RP is split up in actual reference information (upper nibble R), and four reserved bits for future extensions (lower nibble P). To maintain backward compatibility, these reserved bits can be "don't cares" in VIP1.1. These reserved bits were defined in the ITU-R-656 specification as Hamming coded error protection and correction bits P3..P0. They were needed because ITU-R-656 was designed for long cable connections among video processing equipment.

T	F	V	H	*Reserved*	*Reserved*	*Reserved*	*Reserved*

The bits in the reference byte RP of the SAV and EAV code sequences (header).

According to ITU-R-656, the upper nibble R of the reference byte contains the three "reference" bits F, V, and H. They carry the following video timing / raster information:

H =	horizontal blanking:	0 = active line,	1 = horizontal blanking.
V =	vertical blanking:	0 = active video,	1 = vertical blanking,
F =	field ID:	0 = odd,	1 = even,
T =	task bit:	0 = task B,	1 = task A.

To maintain compatibility with ITU-R-656, the T, F, and V bits can only be changed in the EAV code. VIP slaves that do not output scan lines during vertical blanking must create at least one dummy scan line to provide the necessary EAV code in order to convey the updated T, F, and V bits.

The Task Bit T is an extension for VIP video streams to distinguish two different tasks, e.g. two different video streams, or different destinations in the frame buffer. In ITU-R-656 the MSB (T-bit) is fixed to 1. A VIP slave may choose not to use the Task Bit. However, a VIP-compliant graphics chip must support the Task Bit. The ability to store streams with different task bit in different areas of the frame buffer is a requirement for the graphics chip.

In VIP's active video stream, a single code 00 hex - that does not belong to any header sequence - is used to mark an empty cycle, i.e. a clock cycle that does not carry any valid video sample. In VIP applications, the number of pixels per line is not restricted to 720, but could be any number. This is an extension to the ITU-R-656 recommendation. It can be used to support scaled video and the different sampling scheme of SQP (square pixel).

VIP requires, as a minimum, that the active video lines, as defined by an "active video acquisition window", are enveloped by appropriate SAV and EAV codes:
All active, selected, valid lines start with a SAV code of H = 0 = active, V = 0 = active.
All, but the last, active lines end with an EAV code of H = 1 = inactive, V = 0 = active.

The last active line ends with an EAV code with H = 1 = inactive, V = 1 = inactive.
The unique combination of V = 0 = active in SAV and V = 1 = inactive in EAV in the same line marks the end of a field, and acts like a vertical sync. The related DMA or memory pointer can be reset.

"Empty" video lines, i.e. lines that are not selected by the active video acquisition window, do not appear on the VIP video bus. During blanking periods no specific video sample values need to appear on the VIP video bus. That saves room to stream other data types on the same wires.

Raw ADC samples of selected VBI lines (VBI bypass range, "raw samples acquisition window") can be transported in a way similar to active video, not as multiplexed Cb-Y-Cr-Y- ..., but as a single stream of ADC samples. SAV and EAV codes are built according to above stated rules.

The Task (T) Bit differentiates between two video streams, or different destinations in the frame buffer. The ability to store streams with different task bits in different frame buffer areas is a requirement for a VIP graphics chip. The usage can vary depending on the device type. T=1 is always the default primary video stream. T=0 indicates a secondary video stream, or raw VBI sample data.

For a VIP1.1 video stream, when T=0, the VIP master must capture all the data within SAV/EAV to the VBI buffer. This is true regardless of the V Bit value. The VIP slave device driver can access the captured data.

For a VIP2 video stream, the Task Bit is strictly used to differentiate between two video streams. Within each video stream, when V=1 (blanking), the receiving device must capture all the data within SAV/EAV to its VBI buffer.

8.3 Ancillary data blocks with ANC header

ITU-R-656 supports the transport of data other than video as ancillary data blocks during horizontal or vertical blanking intervals only. Such an ancillary data block is headed by a 6-byte ANC header sequence of **00-FF-FF-DID-SDID-NN-**. The first three bytes, **00-FF-FF-,** represent the preamble to the ANC header. **DID** and **SDID** are "data identifier" byte and "secondary data identifier" byte. The use of their codes is regulated by ITU or SMPTE organization. The **NN** byte specifies the length of the ancillary data block to follow measured in number of D-words (4-byte blocks). The data in the data block has no code restriction. Any code value may occur during the ancillary data block, i.e. also 00 and FF are valid data codes inside an ancillary data block. The SAV/EAV- and ANC-header detection circuitry has to be disabled for that time of (n+1) D-words. The ancillary data block does not have a specific end marker, but is followed by a check sum byte. The DID, SDID and NN bytes as well as the check sum byte are error protected in itself.[1]

VIP supports the use of ANC data to transport:

Decoded (sliced) VBI data, from a hardware data decoder, or

Digital audio streams (optional), usually correlated with captured video stream.

Ancillary data blocks can be inserted only during blanking times, i.e. only after a video EAV code. Only entire ANC data blocks can be inserted, and only as an un-interrupted data stream. The transmission of an ancillary data block has to be completed before the next active video line can start with a SAV header.

[1] A VIP source of ANC data blocks should provide the error protecting codes. A VIP receiver, e.g. VGA controller, is not required to check for their integrity in hardware. But the entire data stream, including the error protecting bytes, should be made available (captured) to the CPU system for optional software error checking.

8.4 Sliced VBI data

Depending on the VBI data standard, there are up to 48 bytes encoded in a video line (e.g. 2 bytes per line for closed caption, 32 bytes per line for NABTS and Intercast). This is much less data than raw VBI transfer in which 1500 raw ADC-VBI samples per line need to be collected and transported to CPU. A hardware VBI data slicer reduces significantly the required frame buffer amount and/or the real-time PCI load demand.

If a hardware slicer is used, it is unlikely, that the raw VBI-ADC sample capture is used. Then the Task Bit T of the SAV/EAV codes may be used to support two independent video sources or destinations, e.g. in a toggle mode field by field.

A hardware VBI data slicer may select and process input signals independent of the signal source for “active video“. The sources for captured/displayed video and for the (VBI) data slicer may be asynchronous. Ancillary data blocks with sliced VBI data may appear on the bus between lines of active video anywhere during the active video field.

The following table shows the internal bits and corresponding byte values of the ANC header codes, as used for the ancillary data blocks of sliced VBI data.

ANC header and ANC data block : in bits and bytes, D-words

0	0	0	0	0	0	0	0			00	ANC
1	1	1	1	1	1	1	1			FF	pre--
1	1	1	1	1	1	1	1			FF	amble
NEP	EP	0	1	0	**did2**	**did1**	**did0**	**DID**	ANC Identifier	DID	
NEP	EP	pr5	pr4	pr3	pr2	pr1	pr0	SDID	programmable	SDID	first
NEP	EP	**n5**	**n4**	**n3**	**n2**	**n1**	**n0**	**NN**	Data count	NN	D-
idid07	idid06	Idid05	idid04	idid03	idid02	idid01	idid00	IDID	Internal Data	IDID0	word
idid17	iiid14	Idid15	idid14	idid13	idid12	idid11	idid10	IDID	Identifier	IDID1	
d7	d6	d5	d4	d3	d2	d1	d0	1.data	byte	data	
								2.data	byte	data	to
								:	:	:	
								:	:	:	be
								:	:	:	
								:	:	:	stored
								:	:	:	
								:	:	:	
								:	:	:	
								:	:	:	n+1st
d7	d6	d5	d4	d3	d2	d1	d0	n.data	last data byte	data	= last
NEP	EP	cs05	cs04	cs03	cs02	cs01	cs00	CS	check sum		D-
								FB	Fill Byte		word

Legend: EP = even parity on lower 6-bits
NEP = negated even parity bit

ITU-R-656 specifies eight DID codes for user application (without obligation for registration):

50h, 91h, 92h, 53h, 94h, 55h, 56h, 97h.

The upper two bits represent parity, and inverted parity bits, for error protection. Only the three LSBs of the DID-byte need to be investigated by the receiving VGA controller.

VIP uses the available lower three bits of DID in the following way:

I2	I1	I0	DID	Event in source stream for data slicer		
0	0	0	50	start of first, odd field,	dummy ANC block	optional
0	0	1	91	sliced data of VBI lines of first field	ANC data block	VBI data
0	1	0	92	end of nominal VBI of first field	dummy ANC block	optional
0	1	1	53	Sliced data of line 24 to end of first field	ANC data block	full field data
1	0	0	94	start of second, even field,	dummy ANC block	optional
1	0	1	55	Sliced data of VBI lines of second field	ANC data block	VBI data
1	1	0	56	end of nominal VBI of second field	dummy ANC block	optional
1	1	1	97	Sliced data of line 24 to end of second field	ANC data block	full field data

The DID codes '91' and '55' transports sliced data of regular VBI. The DID codes of '53' and '97' transport sliced data of full field data encoding.

For a dummy ANC block, the SDID and "nn" bytes are required. The "nn" byte, which is zero, is the last byte of the block.

The optional DID codes of "start of field" and "end of nominal VBI" are defined for special ANC "timing" headers, that will not be followed by any data block to be stored. They are supposed to trigger buffer swapping or DMA operations. The "end of nominal VBI" code can be used by the VGA controller, e.g., to initialize an interrupt to get the local buffer moved to CPU or system memory. Such data move should be finished before the next "start of field" code occurs, or a second buffer has to be allocated. Please note that signal source and timing of the hardware data slicer may be asynchronous to active video stream accompanied by SAV and EAV codes.

The SDID code is a "don't care" for the receiving VIP Master. The VIP Master needs to capture but not decode the SDID code.The use of SDID is up to the disposal of the software application. The N-byte defines the number of D-words to follow. N is coded in the lower 6-bits, covering a range of 0 to 63 D-words. The upper two bits carry parity and inverted parity. For the special "dummy" DID-timing codes, the N-bits show 0.

A VIP compliant VGA controller, receiving the ANC data block, is supposed to catch the 00-FF-FF- preamble of the ANC header, to check then the DID code, to decide what to do. If DID is not a special interrupt / timing code, the VGA controller has to look out for N (number of D-words), but store (N+1) D-words[1], beginning after the DID-byte, i.e. with the SDID byte. The extra D-word is formed by the four bytes following the DID-byte: SDID, N-byte, and two "internal data identifier" bytes. The other D-words to be stored contain the sliced VBI data, including the check sum byte. If the amount of recovered (sliced) data does not match with D-word boundary, it gets filled up with fill values. The VIP Master captures but need not decode the checksum and/or the fill bytes

8.5 Digital Audio PCM

In parallel to the captured video stream, digital audio originated of the same source may be captured through the VIP video bus. Multiple stereo channels could be supported. A stereo channel with, e.g., 2 * 16-bits PCM, sampled with 48 kHz, produces about 12 bytes per video line. This amount of data is

[1] If a continuos buffer mode is applied (no line pitch), it is recommended to store the entire ANC header including preamble, to ease restructuring of the collected data by software.

treated as ancillary data, and fits easily into horizontal blanking interval of the video stream.

The description in this section is preliminary. It is based on, and represents a subset of, a draft of the ITU task group 11/2 under progress. It may need to be revised after that ITU draft gets final status.

Up to 4 digital audio PCM channels, i.e. two stereo pairs, are bundled into a group. Each group is represented by a specific ANC-DID code. Up to 4 groups, i.e. up to 16 channels can be encoded as ancillary data into the horizontal blanking intervals of the video stream.

The transmission of the audio samples is spread over the video field, depending on their sample rate, to achieve near equal ANC data load for all line blanking periods. The collected samples per line form an Audio Data Packet. The packets of each group are transported by means of a specific ANC-DID code.

Four ANC-DID codes are reserved for the four Audio groups:

BF h, 7F h, 7E h and BE h.

9 VIP2 Extensions

9.1 Overview

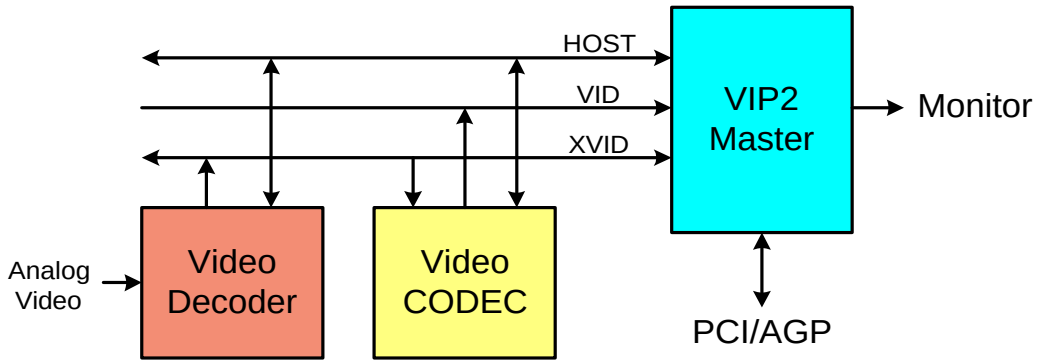
This section contains the VIP2 extensions. These extensions enhance the bandwidth and capability of both the video and the host port for future applications. The philosophy behind these extensions is to maintain backward compatibility to VIP1.1 and offers a low-cost/low-complexity enhancement path. By definition, a VIP2 device must be fully compliant with VIP1.1. VIP1.1 is described in Section 1 through 8 of this Specification.

With these extensions, the peak host port bandwidth will be 264M-Bit/s and the peak video bandwidth will be 150MByte/s.

VIP2 specifies three levels of capabilities: Level I, II, and III. This level system offers flexibility in implementation that covers a wide range of applications. All levels support the same host port, but differ in the video port capabilities. By definition, a higher level device must support all the lower level capabilities. Level I supports an 8-bit/75MHz video port (VID), and targets HD0 applications. Level II supports a 16-bit/75MHz video port (VID and XVID), and targets HD1 applications. The enhanced video port is intended for more than HDTV support. The higher bandwidth enables a new generation of traditional VIP slaves (e.g. video decoder, DVD decoder) to offer value-add video processing features such as de-interlacing and scaling.

Level III supports two separate 8-bit video ports (VID and XVID). In addition, XVID can be programmed as an output from the VIP2 master. This is a significant architectural enhancement, enabling VIP2 to support new applications. Here are some examples:

1. Video co-processor or de-interlacer that receives video from the graphics chip via XVID, then outputs the processed video back to the graphics chip via VID.
2. XVID can be programmed as a TV output port that interfaces to an external video encoder. XVID runs up to 80MHz, and can support 800x600x60Hz in YCrCb format.
3. A video compression/decompression (codec) chip that receives an input video stream from XVID. The video stream is either from an external VIP video decoder or from the VIP Master. The compressed stream is output through the VIP host port, and the on-the-fly decompressed video stream is output to the VIP Master via VID. This architecture is illustrated in the following diagram:



9.2 VIP2 Signal Description

Signal	I/0 Type (From VIP Master Side)	Description
Host Port		
VIPCLK	O	VIP Host Clock (25-33MHz)
HAD[7:0]	I/O	Host Address/Data Bus (4X mode)
HCTL	I/O	Host Control; this includes the symbolic signals of VFRAME, DTACK#, and VSTOP#
Video Port		
VID[7:0]	I	VIP Video Data
VIDB, VIDA	I	Least Significant two bits in the optional 10-bit video mode.
PIXCLK	I	Video Pixel Clock
XVID[7:0]	I or O	Extended VIP Video Data (2^{nd} video stream)
XVIDB, XVIDA	I or O	Least Significant two bits in the optional 10-bit video mode.
XPIXCLK	I or O	Extended Video Pixel Clock (Level III Only)
System Signals		
VRST#	O	VIP Module Reset
VIRQ#	I - OD	Interrupt Request - Open Drain (CMOS) Open Collector, Level Sensitive

9.3 The Host Port

The data bus width is extended from the current two bits to a maximum of 8-bits. Two levels of extensions are possible: 4-bit, and 8-bit. A VIP2 master **MUST** support the Standard 2-bit, 4-bit and 8-bit modes. A VIP2 slave, on the other hand, can choose to support 4-bit, 8-bit, or the Standard 2-bit mode only.

The clock frequency and the protocol rules remain unchanged, which should greatly simplify the implementation of these extensions.

It is **STRONGLY RECOMMENDED** for a PC-based VIP2 Master to support a minimum of one PCI bus mastering DMA channel for access to and from the FIFO ports of the slaves. DMA bus-mastering significantly reduces the CPU overhead, offers much higher latency tolerance, and provides a much more reliable system environment for interfacing to a wide variety of VIP applications.

On power up, all devices with extensions must operate in the Standard 2-bit compatibility mode. It is the responsibility of the graphics device driver to configure the slave devices into the extended modes. Each slave must add the following bits to its Status and Command registers.

9.3.1 HOST_CAP[1:0]

For the Status Register (address 00B:00A), add HOST_CAP[1:0] to bits 3 and 2. HOST_CAP[1:0] has the following meaning:

HOST_CAP[1:0]	Highest Host Mode
00	Standard (2-bit)
01	4-bit
10	8-bit
11	Reserved

HOST_CAP[1:0] allows the graphics chip to query the slave's host port highest capability. While Standard 2-bit Mode support is mandatory for all the slaves, an 8-bit slave does not need to support the 4-bit mode.

9.3.2 XHOST_ON

For the Command Register (address 009:008), add XHOST_ON to bit 2. Once the graphics device driver determines the highest capability of a slave, it can switch the slave into its extended mode by writing a "1" to the XHOST_ON bit. Extended mode operations with a VIP slave start immediately after the current transfer. Once set, this flag cannot be reset without a system reset or power down.

9.3.3 MY_IRQ

For the Status 0 register (FIFO location zero), add MY_IRQ to bit 1. A "1" indicates that the interrupt pin of the queried device is active. This gives the master a quick way to determine the source of interrupt.

9.3.4 VIP2 Registers

9.3.4.1 The Command Register (009:008)

Bit	Field	Description
1:0	Power_State	A read to this field reports the current operating power state (P0 to P3). A write to this field sets the device into a new power state. If software tries to set an unsupported state, the transaction should complete normally but the state should be left unchanged.
2	XHOST_ON	Turn on Extended Host Mode (VIP2 Only)
15:3	Reserved	

9.3.4.2 The Status Register (00B:00A)

Bit	Field	Description
0	P1_Support	"1" indicates that power state P1 is supported.
1	P2_Support	"1" indicates that power state P2 Is supported.
3:2	HOST_CAP[1:0]	Host Capability (VIP2 Only)
15:4	Reserved	

9.3.4.3 Reserved Registers

For a VIP2 slave, register locations 00F:00E are reserved for future use. They should return zeros when read.

9.3.4.4 Status 0 (FIFO Port 0)

Bit	Flag	Description
0	VIRQ	1 = The VIRQ# pin is asserted (Source unknown). An access does not clear this flag.
1	MY_IRQ	1= THIS device asserts its interrupt pin. An access does not clear this flag. (VIP2 Only)
2	User Define	These flags are defined by the VIP device manufacturer
3	User Define	
4	DREQA	DMA Request for FIFO A
5	DREQB	DMA Request for FIFO B
6	DREQC	DMA Request for FIFO C
7	DREQD	DMA Request for FIFO D

9.3.5 Extended Data Transfer

The extended data signals are HD[7:2]. It still takes 4 cycles to complete a command/address or data transfer phase, but with 2X and 4X, two and four bytes are transferred, respectively. The order of data transfer is as follows:

9.3.5.1 Standard Mode

Cycle	HAD[1:0]
1	Byte0[7:6]
2	Byte0[5:4]
3	Byte0[3:2]
4	Byte0[1:0]

9.3.5.2 4-bit Mode

Cycle	HAD[1:0]	HAD[3:2]
1	Byte0[7:6]	Byte1[7:6]
2	Byte0[5:4]	Byte1[5:4]
3	Byte0[3:2]	Byte1[3:2]
4	Byte0[1:0]	Byte1[1:0]

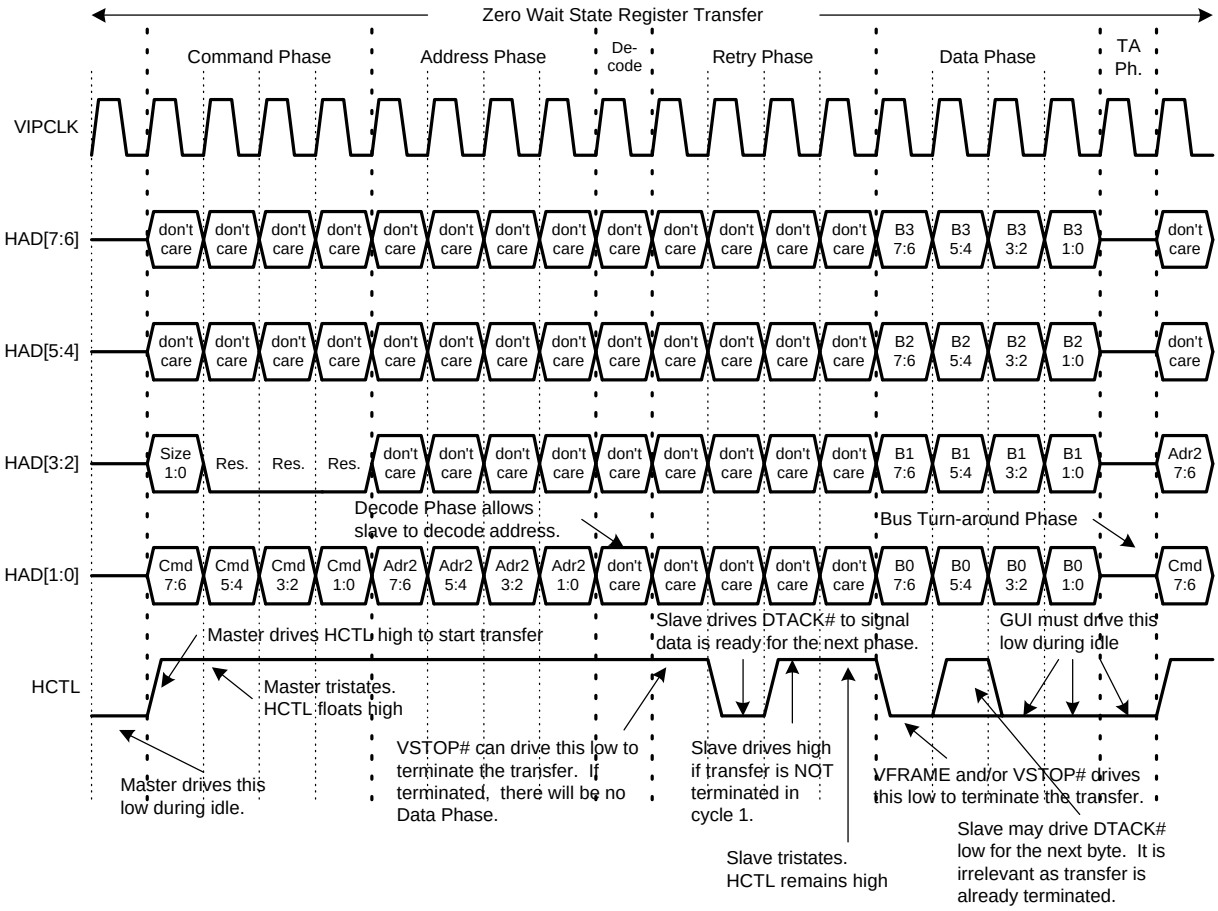
9.3.5.3 8-bit Mode

Cycle	HAD[1:0]	HAD[3:2]	HAD[5:4]	HAD[7:6]
1	Byte0[7:6]	Byte1[7:6]	Byte2[7:6]	Byte3[7:6]
2	Byte0[5:4]	Byte1[5:4]	Byte2[5:4]	Byte3[5:4]
3	Byte0[3:2]	Byte1[3:2]	Byte2[3:2]	Byte3[3:2]
4	Byte0[1:0]	Byte1[1:0]	Byte2[1:0]	Byte3[1:0]

In the event that fewer bytes are involved in a transfer, the higher portion of the HAD bus will be undefined during the entire phase. For example, if only one byte is transferred in the 8-bit mode, the data byte will appear at HAD[1:0]. No valid data will appear at HAD[7:2].

In standard VIP register access, a second address phase is needed as the four address bits in the first Command byte is insufficient. This second byte requirement applies to extended transfers as well to ensure compatibility.

All extended transfers, including both FIFO and register accesses, must place the SIZE[1:0] bits on HAD[3:2] during the first (command) phase. The SIZE[1:0] are explained later in this section. As in VIP1.1, all extended Register Transfers include the second address phase, while extended FIFO transfers do not have the second address phase. The following is an example of an 8-bit Register Transfer:



9.3.6 The SIZE[1:0] Bits

It is desirable to maintain the byte granularity nature of VIP. One reason is that DVD data is inherently byte-aligned. To maintain byte granularity, the graphics chip must be able to indicate the number of bytes per each data phase. This is accomplished by the transfer size indication, SIZE[1:0]. SIZE[1:0] defines the following:

SIZE[1:0]	Byte(s) Transfer
00	1
01	2
10	3
11	4

The graphics chip outputs the SIZE[1:0] bits on the most significant two bits of HAD[3:2] during the Command Phase. For multi-byte burst transfers, SIZE[1:0] applies to **EACH** data phase. For the 4-bit mode, only SIZE[0] is meaningful.

For a particular data phase, if the slave cannot complete **ALL** the bytes specified by SIZE[1:0], it must terminate the transfer without transferring a single byte. The graphics chip will then retry the transfer on a byte-by-byte basis. Since odd-byte transfer only occurs during a boundary condition, its performance impact should be negligible.

9.4 The Video Port

The video port will be extended in two ways: double the data width to 16-bit, and increase the maximum PIXCLK to 75MHz. Also, three levels of implementation are defined. The highest level includes two independent video ports of which one can be programmed for either direction.

At 75MHz, the extended video mode cannot transfer through the ribbon cable using the pin-out and connector in Section 9.6. For backward compatibility it is possible that a VIP system may consists of ribbon cable modules and extended video modules. Buffers will be needed to cut off the ribbon cable connection during high speed extended video transfer. It is the OEM's responsibility to ensure data integrity during extended video transfer.

VIP2 supports the optional high quality 10-bit video mode. A graphics chip is not required to support the 10-bit mode. Slaves that support the 10-bit mode must also support the 8-bit mode. The least significant two bits are designated as VIDA (or XVIDA) and VIDB (or XVIDB), where VIDA is the least significant bit.

A VIP2 master must support both the VIP1.1 and VIP2 video modes. A VIP2 slave, on the other hand, can support either or both of the video modes. If a VIP2 slave chooses to support the VIP1.1 Video Mode, VIP1.1 must be the power-up default video mode.

On power up the default VIP1.1 8-bit video mode is assumed. When a slave is selected and is ready to output video data, it must communicate with the graphics driver through software to determine the operating video mode.

9.4.1 Extended Video Mode

In the extended video mode, video data is transferred across the 16-bit bus, but control codes such as SAV, EAV, and ANC headers are still transferred across the lower 8-bit bus only. During 16-bit video transfer the first byte is sent across VID[7:0], while the second byte is transferred across XVID[7:0]. To maintain compatibility with some existing 16-bit devices, the data order of Cb-Y-Cr-Y (8-bit) will be changed to Y-Cb-Y-Cr (16-bit). This will result in the Y values on VID[7:0]and the Cx values on XVID[7:0].

Since video data is 16-bit per pixel, there should not be data alignment problems. Skip data ("00") during active video must appear in 16-bit as well so as to preserve the 16-bit data alignment.

Ancillary data will be transferred in the 16-bit mode as well, although the ANC header is always on VID[7:0].

9.4.2 Video Flags

In VIP1.1, the traditional video flags, T, F, and V can only change in the EAV code. This restriction is lifted in VIP2. In VIP2, T, F, and V can change in either the EAV or the SAV code. However, valid values must appear no later than the SAV of the first scan line of the next region (active video or VBI, for the same task). During Vertical Blanking (V=1), there must be a minimum of one SAV/EAV scan line.

9.4.3 New Video Flags

Two new video flags are defined that allow the VIP slave to communicate field/frame-specific information to the graphics chip during the video stream output. These flags are embedded in the lower nibble (Nibble P) of the SAV or EAV header. Nibble P was defined as the hamming code in the original ITU-R-656, but they have been redefined in VIP as the reserved bits. These video flags allow the graphics chip to handle Bob and Weave, as well as 3:2 pull-down in hardware. At the earliest, the video flags can be valid during the EAV of the last active video line, i.e. when the V bit transitions from 0 to 1., The graphics chip must continually update the video flags as valid flags may appear anytime afterward. Valid video flags must appear no later than the SAV of the first active video line.

Bit 0 of the VIP2 Nibble P, EXT_FLAG, indicates whether there are extra flag bytes to follow. Extra bytes can only occur during EAV, not SAV.

The following shows Nibble P with the new video flags:

Bit	Flag	Description
3	NON_INT	"1" indicates that the video is from a non-interlaced source. "0" indicates that the video is from an interlaced source. Typically this flag is embedded in a MPEG2 stream. Note that this flag refers to the nature of the source material, not the output format.
2	REPEAT	"1" indicates that the current field is a repeat field. This occurs during 3:2 pull-down. This flag enables a VIP master to drop the repeat field in the weave mode.
1	Reserved	
0	EXT_FLAG	"1" indicates that extra flag byte(s) follow this EAV

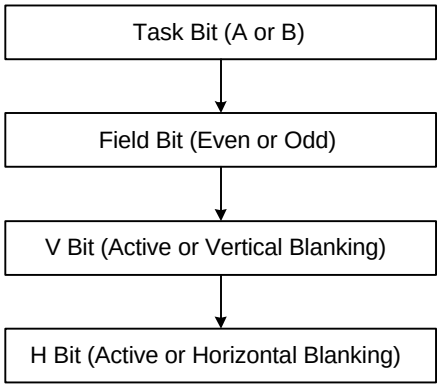
9.4.4 Extra Flag Bytes

Bit	Flag	Description
7	EXT_FLAG#	Inverse of Bit 0 - prevents byte from being a 00 or FF
6-1	User-defined	These flags are to be captured by the graphics chip and can be retrieved by the slave. The associated video reference needs to be maintained. The implementation of the video reference is up to the graphics chip vendor.
0	EXT_FLAG	"1" indicates that extra flag byte(s) follow this byte (chaining)

9.4.5 The Task Bit

In VIP1.1, the Task Bit is used to indicate VBI data within a video stream. In VIP2, the Task Bit is strictly used to differentiate between two video streams. In VIP2, V=1 indicates the vertical blanking period, but also indicates the VBI data region. When V=1, the receiving device must capture all the data within SAV/EAV to its VBI buffer. A VIP2 video stream, regardless of its direction, must adhere to the Task Bit definition of this section.

The following diagram illustrates the hierarchy of a video stream:



According to the diagram above, each task has its own set of Field, V, and H bits. The receiving device must maintain separate sets of flags for each stream (task). The two streams can be interleaved in one of two modes: Line or Field. In Line Mode, the Task Bit can change from one line to another. It is the responsibility of the streaming device to ensure that the correct line-by-line duration is maintained. In Field Mode, the Task Bit can only change after a complete field has been output.

9.5 VIP2 Level System

To avoid confusion with the various configurations a VIP2 master must conform to one of the following levels:

Level	Video Port	Max. PIXCLK	Host Port
I	8-bit	75MHz	Standard, 4-bit, and 8-bit
II	16-bit	75MHz	Standard, 4-bit, and 8-bit
III	Dual-8-bit. XVID[7:0] Must be bi-directional	75MHz for VID, 80MHz for XVID	Standard, 4-bit, and 8-bit

A higher level master must support all the lower level features. For example, a Level III master must support 16-bit and 8-bit video modes.

A VIP2 Slave, on the other hand, can choose to support a subset of the features. However, the slave device vendor must indicate the minimum required level for the VIP2 master. For example, a slave must specify a minimum Level II master if it has a 16-bit video port and a 2X host port.

9.5.1 Level III Dual Stream Mode

A VIP2 Level III master must support the Dual Stream Mode. In this mode two independent streams, VID and XVID can be active simultaneously. Both streams must be of the same version, and can either be

VIP1.1 or VIP2 compliant. The direction of VID is fixed as in VIP1.1 – Slave-to-Master. However, the direction of XVID can be programmed for either direction. In addition, the direction of XPIXCLK is also programmable and can be programmed independent of the direction of XVID. As an example, XVID can be an output from the VIP2 master, while its pixel clock, XPIXCLK, can be an input signal into the VIP2 master. Frequently this is the preferred configuration for a video encoder.

With the possibility of TaskA and TaskB for each video stream, a maximum of four video data sources can be active simultaneously.

9.5.2 XVID Video Encoder Support

As an output stream from the VIP2 master, XVID can be used as a video encoder output. To support the graphics resolution of 800x600 at 60Hz, the XVID pixel clock is extended from a maximum of 75MHz to 80MHz. This enables the support of the new flicker video encoders that perform tasks such as flicker reduction and scaling. Note that the VIP2 master must first convert the pixels from RGB to YcrCb, and the output must follow the VIP video format as described in Chapter 8, *VIP Video Format.* During initialization, the VIP2 master must inform the slave through software that it is outputting a non-interlaced video stream. The slave will ignore the Field flag in Nibble P of the video stream.

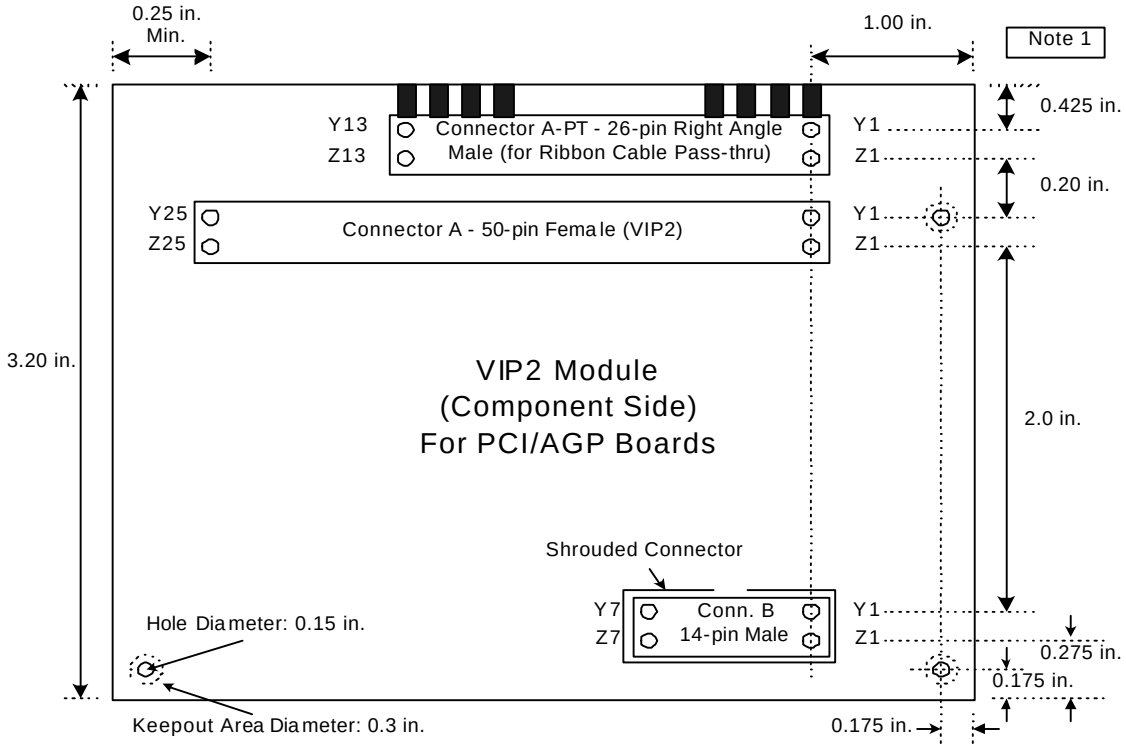
9.6 The Extended Pin Assignment

EXTENDED VIP CONNECTOR A (50-pin Dual Row Header, 0.100 in. centers)						
Standard Feature Connector		**VIP Mode**		**Standard Feature Connector**		**VIP Mode**
Pin #	**Signal Name**	**Signal Name**		**Pin #**	**Signal Name**	**Signal Name**
Z1	Ground	Ground		Y1	P0	VID[0]
Z2	Ground	Ground		Y2	P1	VID[1]
Z3	Ground	Ground		Y3	P2	VID[2]
Z4	EVIDEO#	HAD[1]		Y4	P3	VID[3]
Z5	ESYNC#	HAD[0]		Y5	P4	VID[4]
Z6	EDCLK#	HCTL		Y6	P5	VID[5]
Z7	N/C	SCL		Y7	P6	VID[6]
Z8	Ground	Ground		Y8	P7	VID[7]
Z9	Ground	Ground		Y9	DCLK	PIXCLK
Z10	Ground	Ground		Y10	BLANK#	VIPCLK
Z11	Ground	Ground		Y11	HSYNC	N/C
Z12	N/C	VIRQ#		Y12	VSYNC	N/C
Z13	N/C	SDA		Y13	Ground	Ground
Z14	-	KEY (Ground)		Y14	-	KEY (Ground)
Z15	-	HAD[2]		Y15	-	HAD[3]
Z16		HAD[4]		Y16		HAD[5]
Z17	-	HAD[6]		Y17	-	HAD[7]
Z18	-	VIDA		Y18	-	VIDB
Z19	-	Ground		Y19	-	Ground
Z20	-	XPIXCLK		Y20	-	XVIDA
Z21	-	XVIDB		Y21	-	XVID[0]
Z22	-	XVID[1]		Y22	-	XVID[2]
Z23	-	XVID[3]		Y23	-	Ground

EXTENDED VIP CONNECTOR A (50-pin Dual Row Header, 0.100 in. centers)						
Standard Feature Connector		**VIP Mode**		**Standard Feature Connector**		**VIP Mode**
Pin #	**Signal Name**	**Signal Name**		**Pin #**	**Signal Name**	**Signal Name**
Z24	-	XVID[4]		Y24	-	XVID[5]
Z25	-	XVID[6]		Y25	-	XVID[7]

The 14-pin Connector B for VIP2 remains unchanged from VIP1.1.

9.7 VIP2 Module Mechanical Example



Note 1. This edge of the board is where the IO connectors of the VGA adapter are usually located. Care must be used in placing components in this area to avoid interference with the VGA card connectors.

The pin assignment and the above mechanical example are intended for add-in card implementation with no ribbon cable. A different pin-out and/or mechanical design may be suitable for other implementations, such as motherboard design.

10 Appendix A: DTV Formats

10.1 References

This appendix contains diagrams of the various DTV formats. It is provided as a reference, and is not part of the VIP requirement. The information is derived from the following standards:

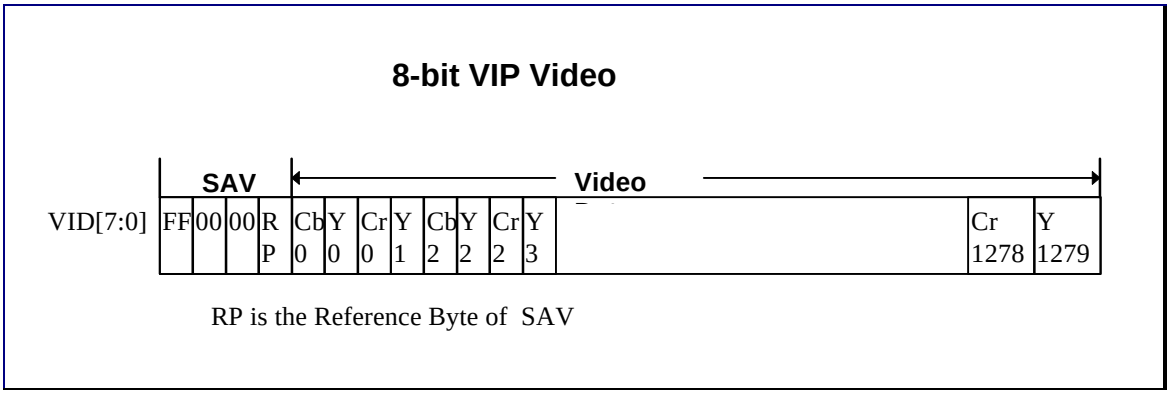
- ATSC Standard A/53 (1995), ATSC Digital Television Standard.
- ITU-R BT.601-4 (1994), Encoding parameters of digital television for studios.
- ITU-R BT.656-3 (1995), Interfaces for Digital Component Video Signals in 525-line and 625-line television systems operating at the 4:2:2 level of Recommendation ITU-R BT.601 (Part A).
- SMPTE 272M (1994), Standard for television, Formatting AES/EBU Audio and Auxiliary Data into Digital Video Ancillary Data Space
- SMPTE 274M (1995), Standard for television, 1920 x 1080 Scanning and Interface.
- SMPTE 291M Standard for Television - Ancillary Data Packet and Space Formatting.
- SMPTE 296M Standard for television, 1280 x 720 Scanning and Interface.
- SMPTE 259 (1993), Standard for television, 10-bit 4:2:2 Component and 4fsc NTSC Composite Digital Signals — Serial Digital Interface.

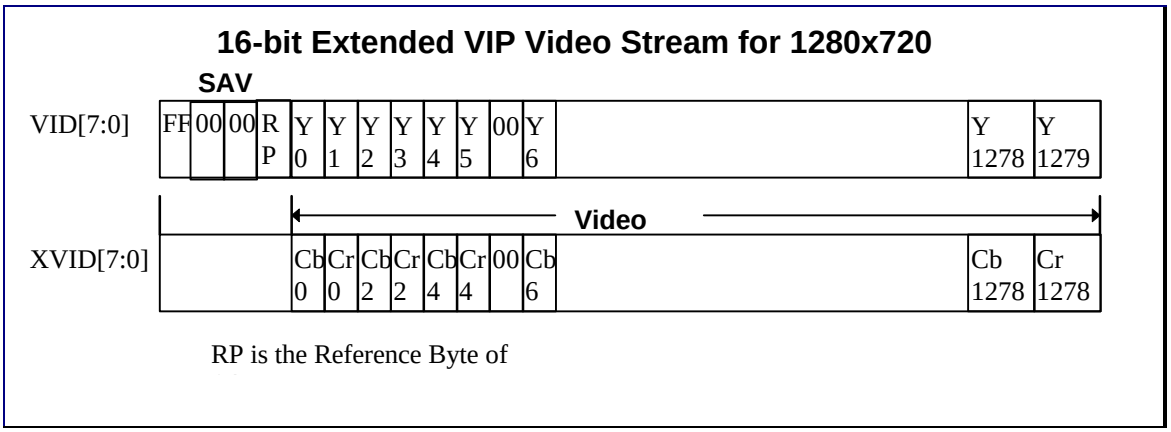
10.2 DTV Resolutions

This table shows some examples of digital TV standards that apply to standard TV resolution (NTSC, PAL), HD0, and HDTV resolution. Shown also are the related recommended VIP implementations.

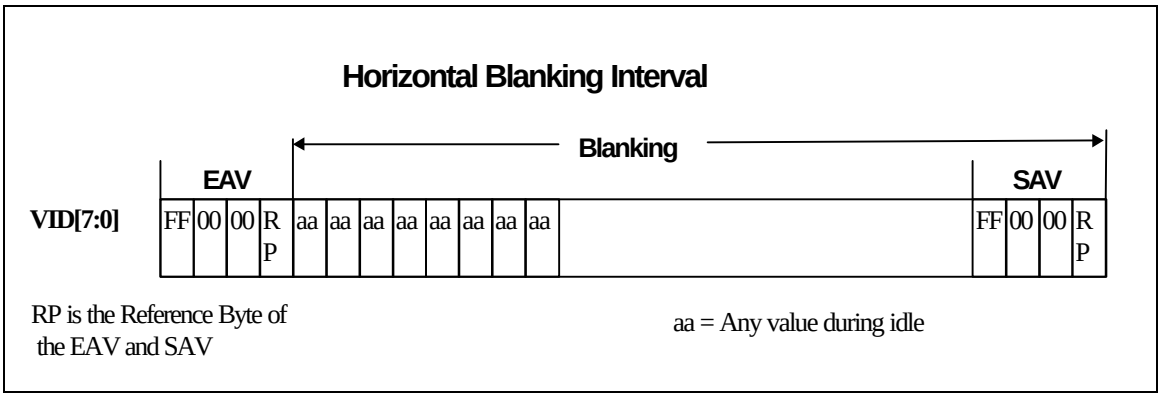
Resolution / picture size			**SDTV / DVD**			**HD0/ ATSC**			**HDTV**	
		NTSC	PAL	Movie	480P	576P	720P	720P	1080I	1080P
Pixel/line		720	720	704	704	704	1280	1280	1920	1920
Line/frame		486	576	<576	480	576	720	720	1080	1080
Rate Hz		60 I	50 I	24 P	60 P	50 P	24 P	60 P	60 I	< 30 P
Pixel rate on native display	MHz	13.5	13.5	*12.9 n.a*	27	27	*29.7 n.a.*	74.25	74.25	*74.25*
VIP Video Port										
YUV422 video, average data rate	MB/s	21	21	20	41	41	45	111	124	124
VIP PIXCLK	bits @ MHz	8 27	8 27	8 27	8 / 54 or 16 / 27	8 / 54 or 16 / 27	8 / 54 or 16 / 27	16 75	16 75	16 75
Video Mode	level	VIP1	VIP1	VIP1	I or II	I or II	I or II	II	II	II
Max clock	MHz	27	27	27	75	75	75	75	75	75
VIP Host Port										
Transport stream (MPEG) data rate	Mb/s	10	10	10	19	19	19	38	38	38
Recommended VIP Host Port	mode pins	Std. 2	Std. 2	Std. 2	2X 4	2X 4	2X 4	4X 8	4X 8	4X 8
Host Port Mb/s	max	66	66	66	133	133	133	266	266	266

10.3 8-bit VIP Video Stream Formats

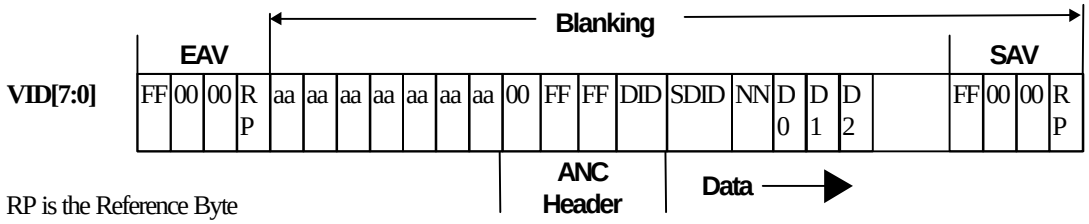




10.3.1 Horizontal Blanking Interval and Ancillary Data Formats



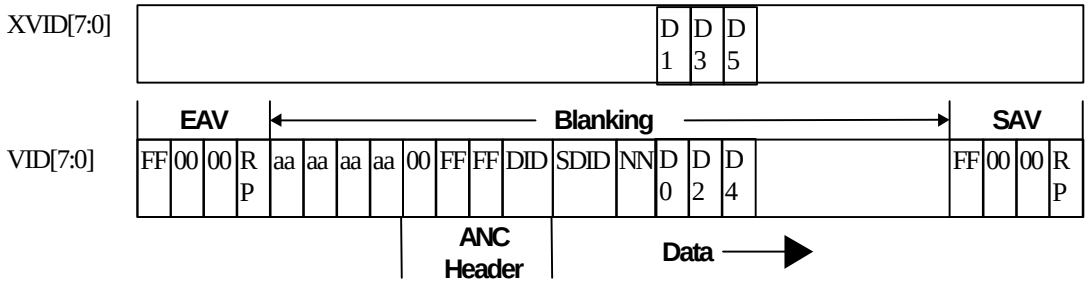
Ancillary Data in Horizontal Blanking Interval



RP is the Reference Byte of the EAV and SAV

aa = Any value during idle

16 Bit Ancillary Data in Horizontal Blanking Interval



RP is the Reference Byte of SAV/EAV

aa = Any value during idle

10.4 DTV Display Formats

10.4.1 Summary of EAV Transitions for Video Format Protocols

Table 1 EAV F and V-bit Transitions by Line Number

Transition	NTSC	PAL	480P	720I	720P	1080I
V-Blank Start for Field 1 (V=1) *	1	624	1	749	746	1124
Field Identification for Field 1 (F=0)	4	1	na	1	na	1
V-Blank End for Field 1 (V=0) if optional Active Video	10	6	na	na	na	na
V-Blank End for Field 1 (V=0) standard Active Video	20	23	43	13	26	21
Line for Closed Caption data	21					
V-Blank Start for Field 2 (V=1) *	264	311	na	373	na	561
Field Identification for Field 2 (F=1)	266	313	na	376	na	564
V-Blank End for Field 2 (V=0) if optional Active Video	273	319	na	na	na	na
V-Blank End for Field 2 (V=0) standard Active Video	283	336	na	389	na	584
Line for Extended Data Services	284					

* Video Flags must be valid in these EAV's

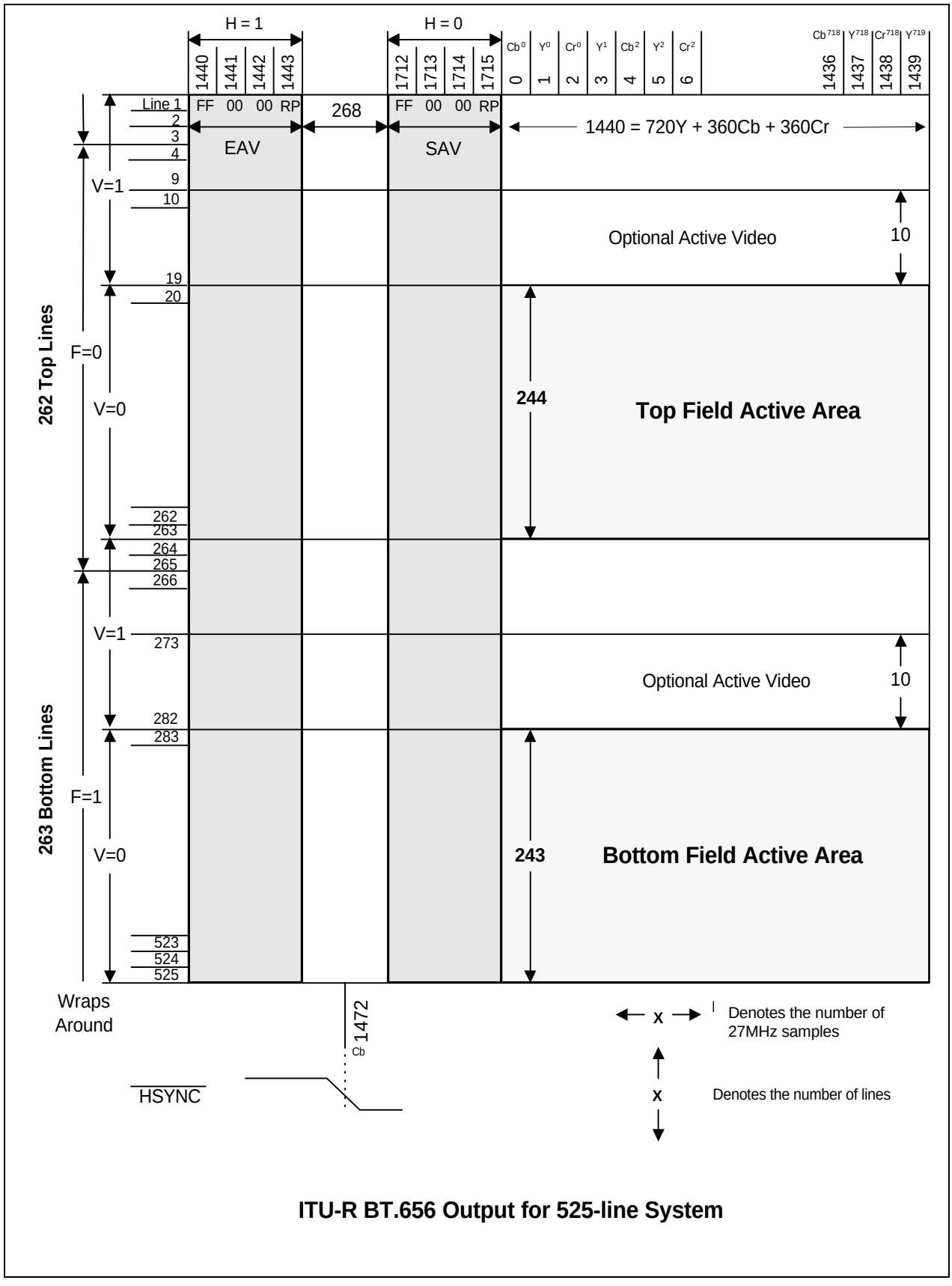
10.4.2 Protocol for 525-line(NTSC) Interlace Video

This diagram shows the ITU-R BT.656-3 standard. The video data is transferred in 8-bit mode at 27MHz. Pixel data is in YUV422 format. Total horizontal line time per line is 63.5us. The horizontal timing point (0_H) takes place on clock 1472.

State change for F and V-bits of EAV	Line Number
V-Blank Start for Field 1 (V=1)	1
Field Identification for Field 1 (F=0)	4
V-Blank End for Field 1 (V=0) if optional Active Video	10
V-Blank End for Field 1 (V=0) standard Active Video	20
V-Blank Start for Field 2 (V=1)	264
Field Identification for Field 2 (F=1)	266
V-Blank End for Field 2 (V=0) if optional Active Video	273
V-Blank End for Field 2 (V=0) standard Active Video	283

The VIP Extended Video Flags are valid only in the EAV for line 1 and 264.

The following diagram shows the data protocol.



ITU-R BT.656 Output for 525-line System

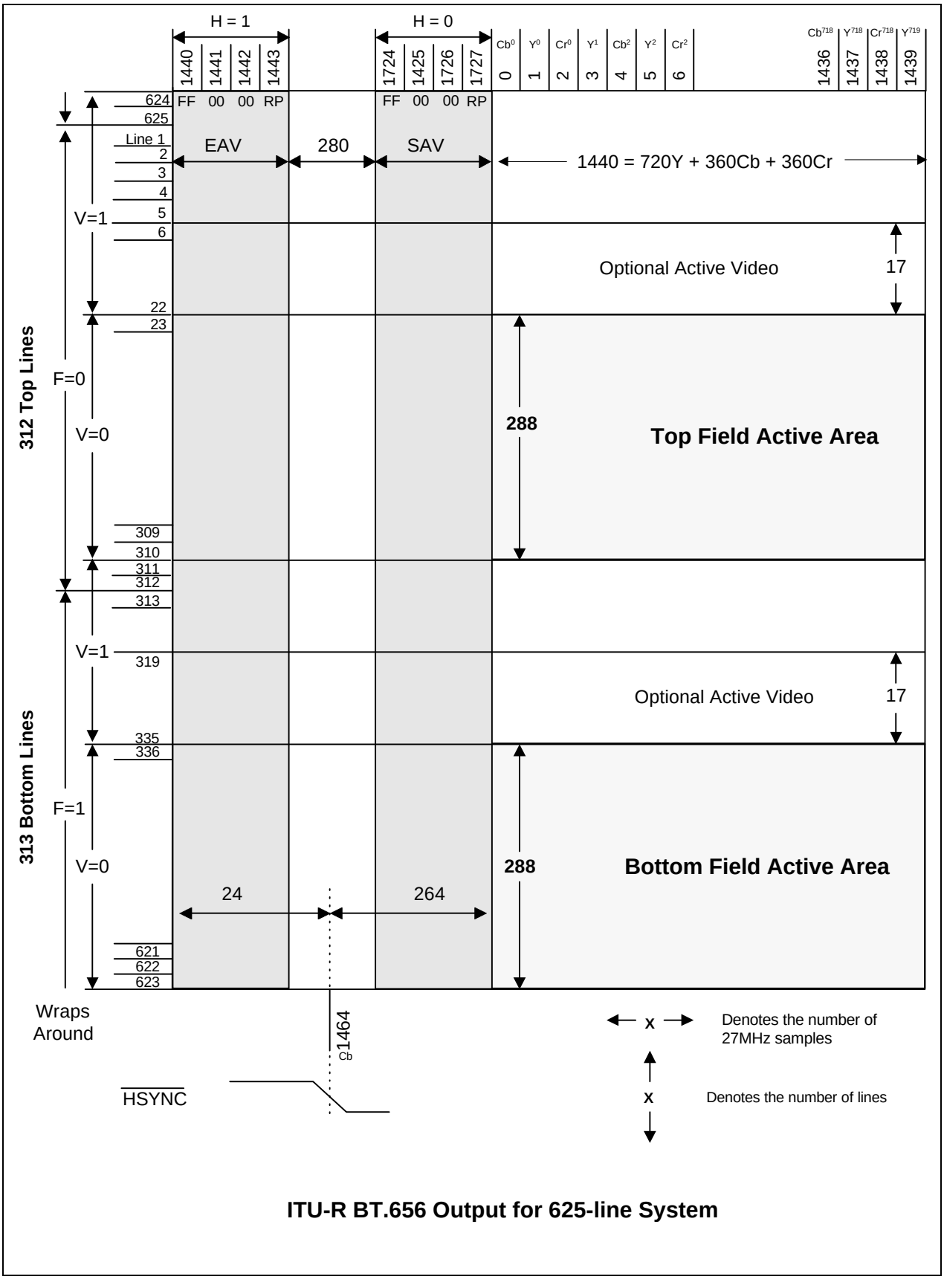
10.4.3 Protocol for 625-line(PAL) Interlace Video

This protocol is based on the ITU-R BT.656-3. The video data is transferred in 8-bit at 27MHz. Pixel data is in YUV422 format. Total horizontal line time per line is 64us. The horizontal timing point (0_H) takes place on clock 1464.

State change for F and V-bits of EAV	Line Number
V-Blank Start for Field 1 (V=1)	624
Field Identification for Field 1 (F=0)	1
V-Blank End for Field 1 (V=0) if optional Active Video	6
V-Blank End for Field 1 (V=0) standard Active Video	23
V-Blank Start for Field 2 (V=1)	311
Field Identification for Field 2 (F=1)	313
V-Blank End for Field 2 (V=0) if optional Active Video	319
V-Blank End for Field 2 (V=0) standard Active Video	336

The VIP Extended Video Flags are valid only in the EAV for line 624 and 311.

The following diagram shows the data protocol.



ITU-R BT.656 Output for 625-line System

10.4.4 Protocol for 704x480 Progressive Video

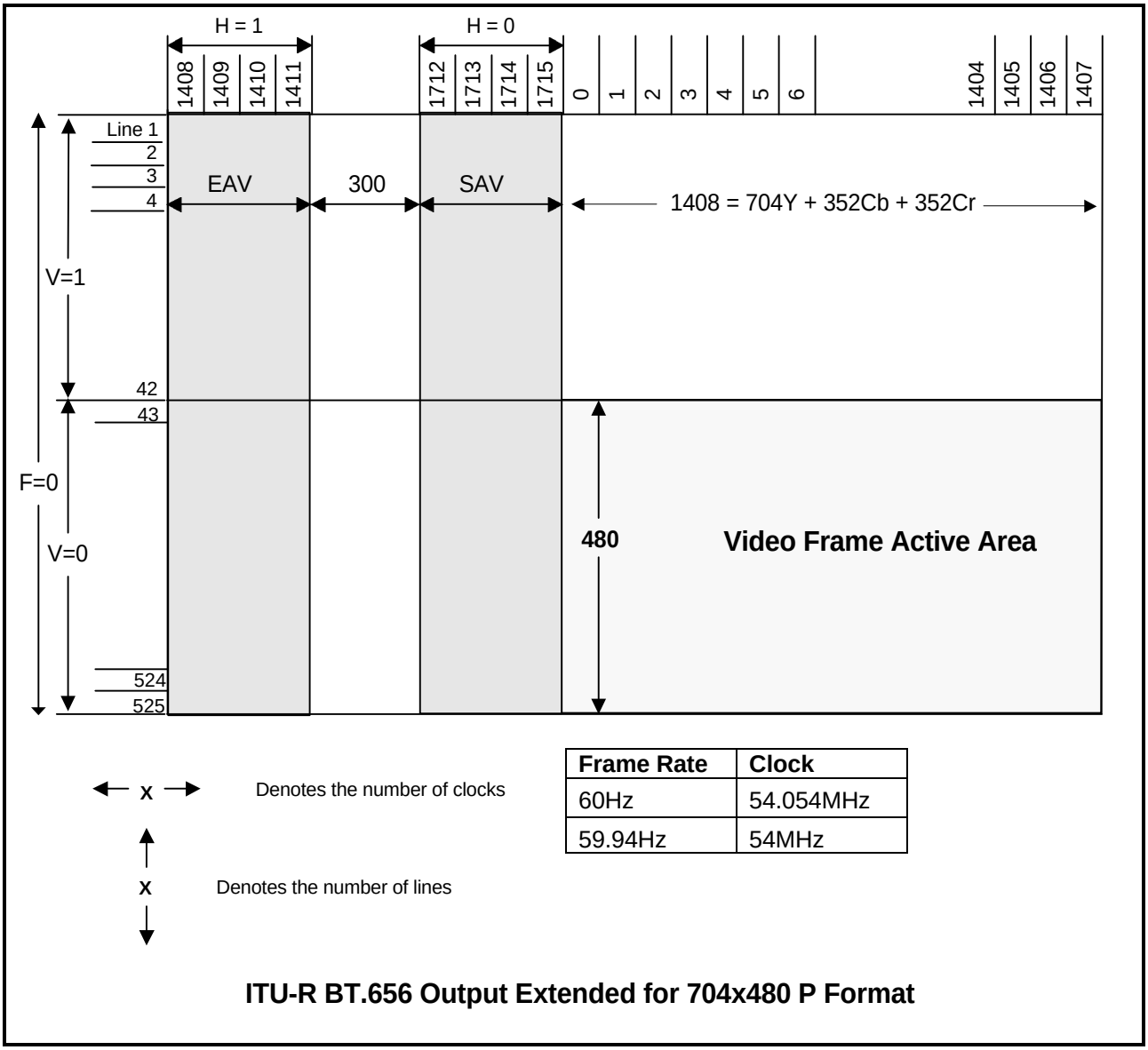
This protocol is based on the ITU-R BT.656-3. The video data is transferred in 8-bit mode at the 54MHz or 54.054MHz. Pixel data is in YUV422 format. The specification for this format had not been released at the time this document was drafted, check for later specification.

State change for V-bit of EAV	Line Number
V-Blank Start for Frame (V=1)	1
V-Blank End for Frame (V=0)	43

The Field Identification bit (F) is fixed at 0 for this format.

The VIP Extended Video Flags are valid only in the EAV for line 1.

The following diagram shows the data protocol.



Frame Rate	Clock
60Hz	54.054MHz
59.94Hz	54MHz

ITU-R BT.656 Output Extended for 704x480 P Format

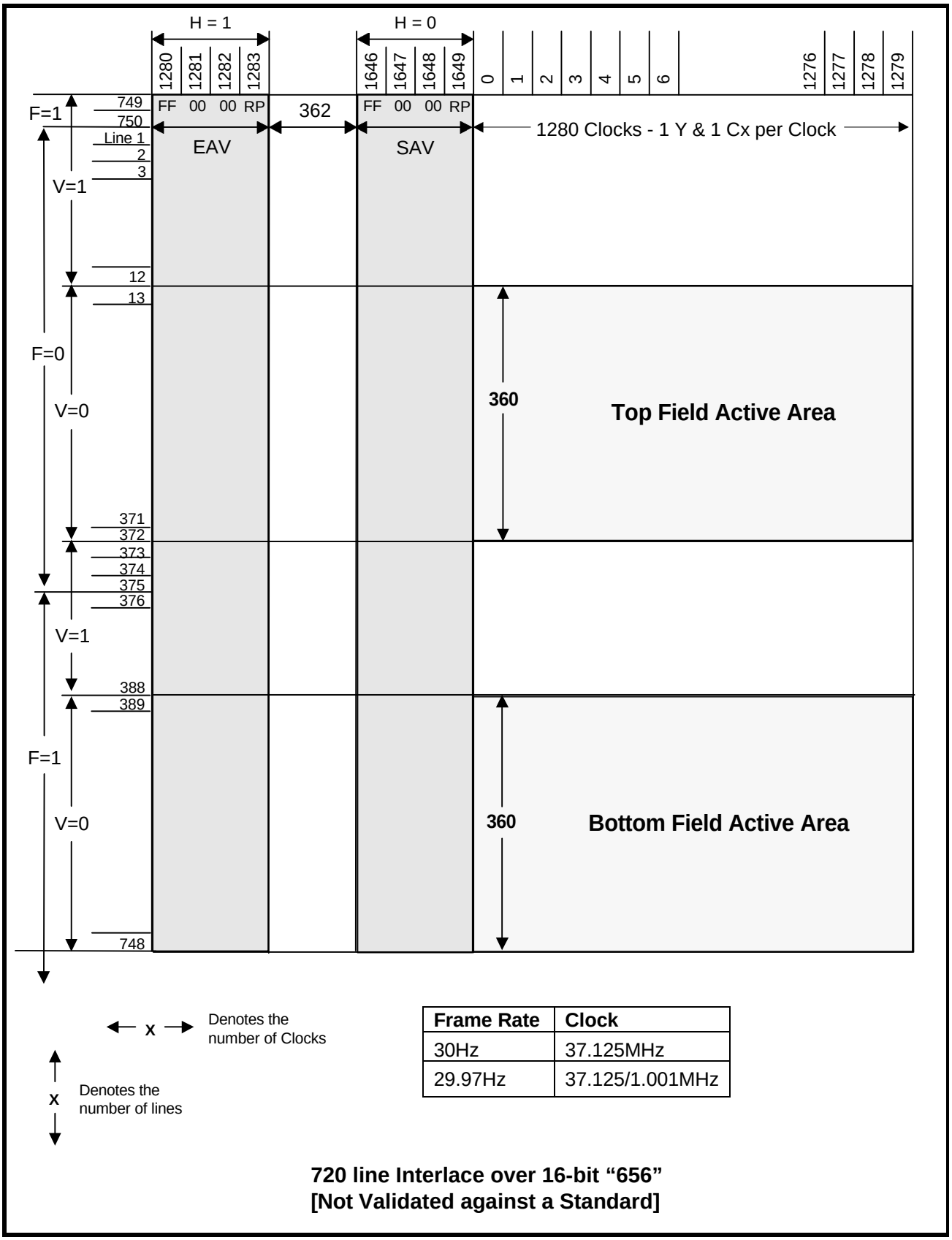
10.4.5 Protocol for 1280x720 Interlace Video

This protocol is based on the SMPTE 296M Standard and the ITU-R BT.656-3. The video data is transferred in 16-bit mode. Pixel data is in YUV422 format. Total horizontal line time per line is 44.44us. The horizontal timing point(0_H) takes place on clock ???. The version of 296M used for this definition did not specify an interlaced version of this resolution, check for newer versions or other specification for final word.

State change for F and V-bits of EAV	Line Number
V-Blank Start for Field 1 (V=1)	749
Field Identification for Field 1 (F=0)	1
V-Blank End for Field 1 (V=0)	13
V-Blank Start for Field 2 (V=1)	373
Field Identification for Field 2 (F=1)	376
V-Blank End for Field 2 (V=0)	389

The VIP Extended Video Flags are valid only in the EAV for line 749 and 373.

The following diagram shows the data protocol.



Frame Rate	Clock
30Hz	37.125MHz
29.97Hz	37.125/1.001MHz

720 line Interlace over 16-bit "656"
[Not Validated against a Standard]

10.4.6 Protocol for 1280x720 Progressive Video

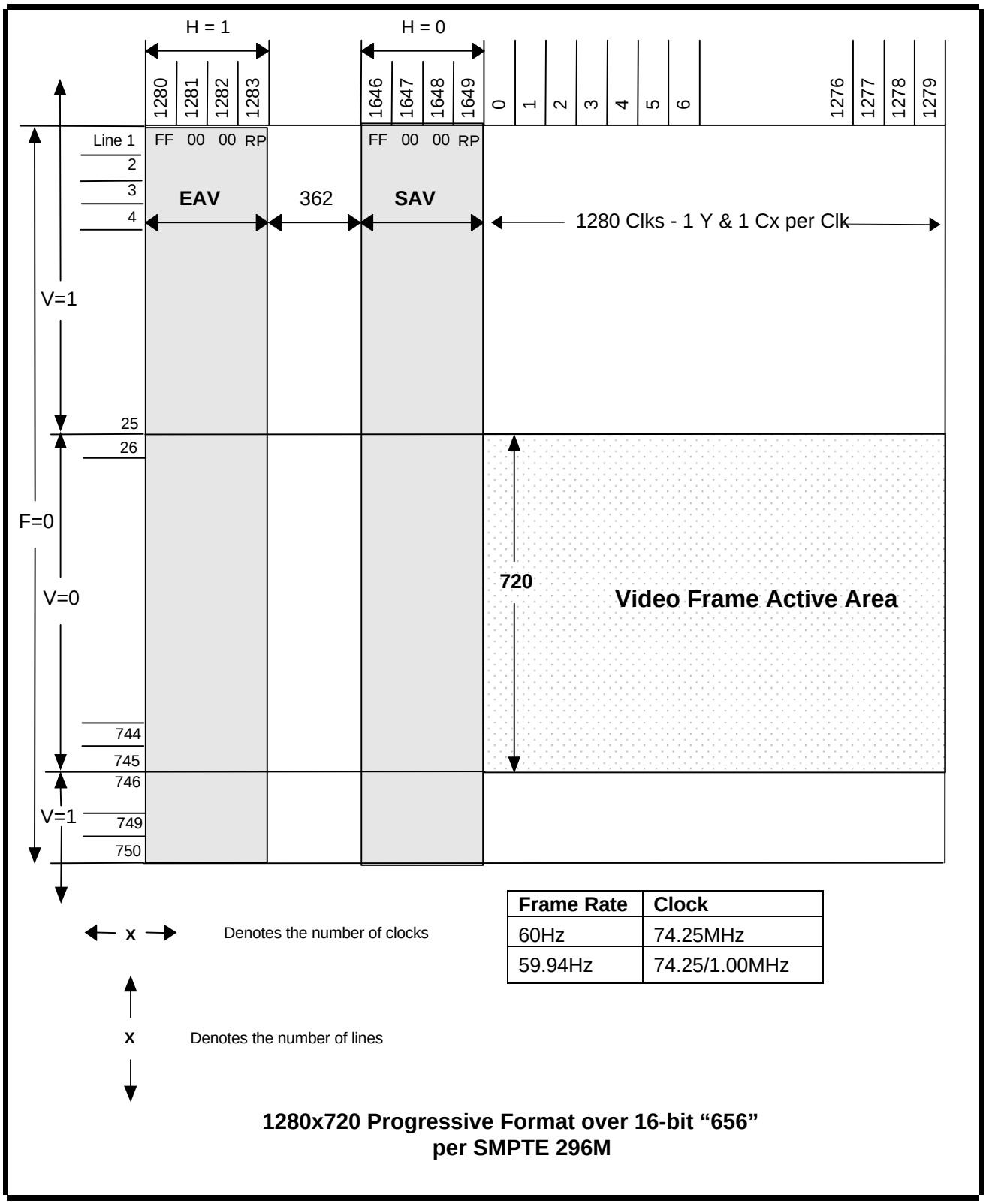
This protocol is based on the SMPTE 296M Standard and the ITU-R BT.656-3. The video data is transferred in 16-bit mode. Pixel data is in YUV422 format. Total horizontal line time per line is 22.2us. The horizontal timing point (0_H) takes place on clock 1390.

State change for V-bit of EAV	Line Number
V-Blank Start for Frame (V=1)	746
V-Blank End for Frame (V=0)	26

The Field Identification bit (F) is fixed at 0.

The VIP Extended Video Flags are valid only in the EAV for line 746.

The following diagram shows the data protocol.



Frame Rate	Clock
60Hz	74.25MHz
59.94Hz	74.25/1.00MHz

1280x720 Progressive Format over 16-bit "656" per SMPTE 296M

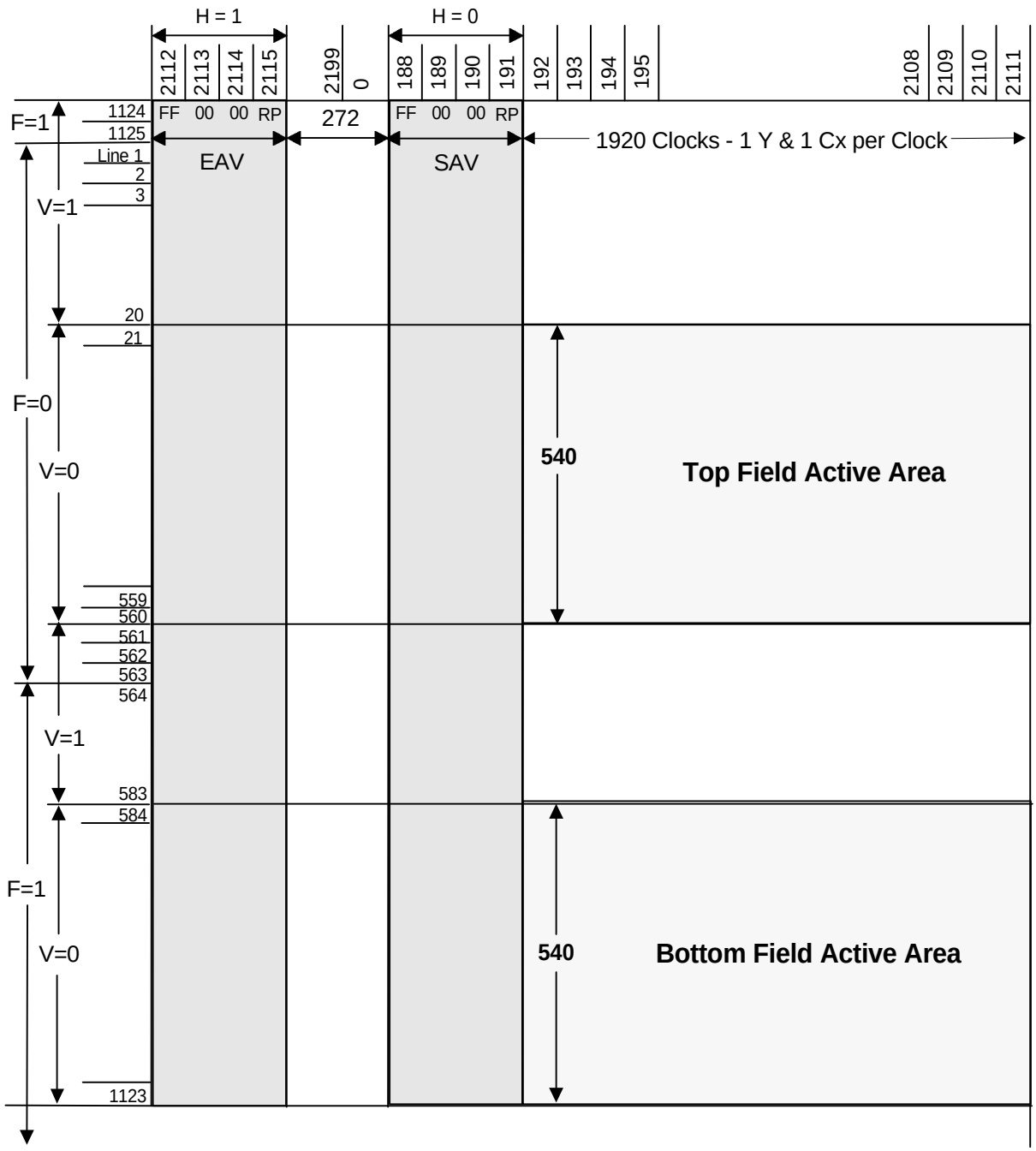
10.4.7 Protocol for 1920x1080 Interlaced Video

This protocol is based on the SMPTE 274M Standard and the ITU-R BT.656-3. The video data is transferred in 16-bit mode. Pixel data is in YUV422 format. Total horizontal line time per line is 29.63us. The horizontal timing point (0_H) takes place on clock 0. Note in this format video data pixels are not numbered from 0.

State change for F and V-bits of EAV	Line Number
V-Blank Start for Field 1 (V=1)	1124
Field Identification for Field 1 (F=0)	1
V-Blank End for Field 1 (V=0)	21
V-Blank Start for Field 2 (V=1)	561
Field Identification for Field 2 (F=1)	564
V-Blank End for Field 2 (V=0)	584

The VIP Extended Video Flags are valid only in the EAV for line 1124 and 561.

The following diagram shows the data protocol.



Frame Rate	Clock
30Hz	74.25MHz
29.97Hz	74.25/1.001MHz

1080 line Interlace over 16-bit "656"
Based on SMPTE 274M

10.4.8 Protocol for Data Packet Mode

Data Packet Mode provides a mechanism to quickly transfer large amounts of data to the system memory. This could be broadcast data packets, compressed video, audio or some other streaming data type.

This mode is an extension to SMPTE 291M Specification and the VESA VIP Ancillary Data Specification. In this mode only Ancillary Data Packets will be sent, no SAV, EAV or Video data. Valid data will be determined only by detection of the Ancillary Data Header and the packet size is determined by the data count. The data contained in the packet can be any value from 00 to FF. Between valid data packets any value other than 00 and FF should be sent on the bus, the capture engine will ignore these values. The mode will support both 8- and 16-bit mode.

10.4.8.1 8-bit Data Packet Format

Hex Value	Function
00	ANC Header
FF	“ “
FF	“ “
xx	DID Code
xx	SDID Code
xx	Data Count 0-63 DWORDS(4 bytes)
xx	Data byte 0
xx	Data byte 1
xx	Checksum

10.4.8.2 16-bit Data Packet Format

Hex Value Vid[7:0]	Hex Value Vid[15:8]	Function
00		ANC Header
FF		“ “
FF		“ “
xx		DID Code
xx		SDID Code
xx		Data Count (0-63) of DWORDS(4 bytes)
xx	xx	Data Word 0
xx	xx	Data Word 1
xx		Checksum