W99200F

MPEG1 Video Encoder
Data Sheet

Rev 1.0

For additional information or questions, please contact:
Hung-Ming Wang

Multimedia Product Marketing & Planning Dept.
TEL: 886-3-5790666 Ext. 6135
FAX: 886-3-5796139
Email : hmwang0@winbond.com.tw

This product is jointly developed with **Zapex Research Ltd.**
Web: http://www.zapex.co.il

Winbond Electronics Corp.

Table of Contents

Table of Figures

Table of Tables

1. General Description

The W99200F is a **real time** MPEG1 / Motion JPEG video encoder. It can encode QSIF and SIF resolution pictures in a maximum picture rate of 30 pictures/sec.

Figure 1-1 shows a typical system configuration based on the W99200F. This system can be used to capture analog video (from handycam or VCR), to encode it in MPEG1 or M-JPEG and to send it to a host. The host bridge directly provides signals for PCI bus, parallel port or generic bus without any glue logic outside the chip. The only memory component needed is a single 1M × 16 bit SDRAM (that serves both as a frame buffer and as an external FIFO). The W99200F includes a glueless interface to popular video decoder devices. Thus, with only three components (W99200F, SDRAM and a video decoder) a complete MPEG1 video capturing and encoding system can be built.

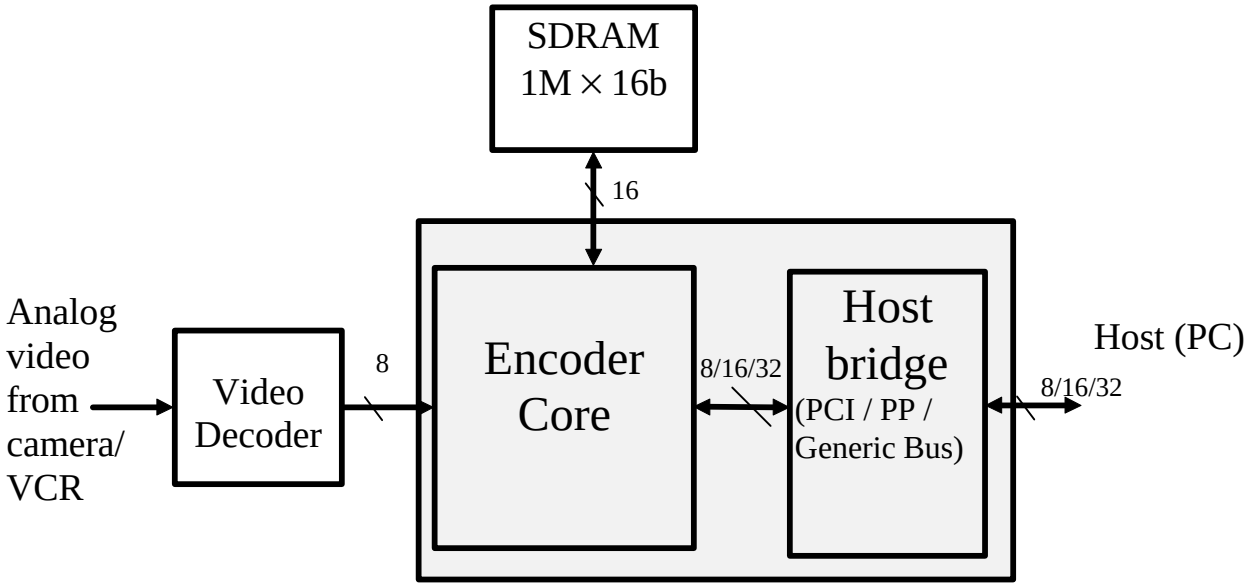


Figure 1-1: A video encoding system based on the W99200F chip

The W99200F has ten modes of operations:

1. **Real time live video encoding**. In this mode the W99200F gets the live video from the video decoder, decimates it, encodes it in MPEG1 or JPEG formats and sends it to the host. Frame sizes supported are: SIF and QSIF.

2. **Real time live video pass through**. In this mode the W99200F gets the live video from the video decoder, decimates it and sends it to the host as YCbCr 4:2:0 video stream. Frame sizes supported are: SIF and QSIF. Frame rate is selectable from full rate (25/30 frames/seconds) down to one frame per two seconds.

3. **Live video snap shot.** In this mode the W99200F captures a single frame from the incoming live video and stores it into the SDRAM . The frame is filtered by adaptive filter in order to de-interlace it. Frame sizes supported are: FULL (YCbCr 4:2:0), SIF and QSIF.

4. **Single frame encoding**. In this mode the W99200F encodes picture already stored in the SDRAM as a MPEG1 I frame or as a JPEG frame and sends it to the host. Frame sizes in this mode are: every size from 32x32 pixels up to 720×576 pixels in a resolution of 16×16 pixels (MB).

5. **Write frame mode**. In this mode the host can write a single frame in YCbCr 4:2:0 format to the SDRAM. Frame sizes in this mode are: every size from 32x32 pixels up to 720×576 pixels in a resolution of 16×16 pixels (MB).

6. **Read frame mode**. In this mode the host can read a single frame already stored in the SDRAM in YCbCr 4:2:0 format. Frame sizes in this mode are: every size from 32x32 pixels up to 720×576 pixels in a resolution of 16×16 pixels (MB).

7. **SDRAM write mode**. In this mode the W99200F enables the host to write any data to the SDRAM. The W99200F is only used as a bridge between the host and the SDRAM.

8. **SDRAM read mode**. In this mode the W99200F enables the host to read any data from the SDRAM. The W99200F is only used as a bridge between the host and the SDRAM.

9. **Write internal memories mode**. In this mode the W99200F enables writing to internal RAMs and registers. This is done in order to increase the encoder controllability and programmability.

10. **Read internal memories mode**. In this mode the W99200F enables reading of internal RAMs/ROMs and registers. This is done in order to increase the encoder observability.

Pay attention that by:
- activating modes 3+4 above the W99200F enables the host to capture a single frame from a live video in MPEG1 or JPEG formats.
- Activating modes 3+6 above the W99200F enables the host to capture a single frame from a live video in an uncompressed (YCbCr 4:2:0) format.
- Activating modes 5+4 above the W99200F enables the host to send uncompressed frame and get it back compressed.

The W99200F produces high quality MPEG1 / M-JPEG video by including a unique rate control algorithm and a dedicated hardware for most of the compression parts such as:

- High quality decimation filters.
- Half-pel accuracy motion estimation with a large search window. Distance between anchor frames ('m') up to 4.
- High precision DCT + Quantization.
- Entropy encoding.
- Automatic 3:2 and scene change detection.

In addition, W99200F Supports two interfaces:

1. Supports the VCD decoder interface (8051-like) to provide the function that VCD bitstreams can be sent from Generic bus, PCI or Parallel port host and pass through W99200F to VCD decoder.

2. Supports audio data input interface to provide the function that audio PCM data or compressed bitstreams can pass through W99200F and be sent to the host. There are two modes:
    - without external audio FIFO: 3-pins Serial data interface.
    - with external audio FIFO: 8-bits parallel FIFO data interface.

2. Features

2.1 PCI Interface

- 32-bit PCI bus interface which supports bus target mode and bus master mode write cycle for high bit rate output.
- provides a set of configuration registers to achieve the purpose of totally software initialization and configuration.
- programs encoder configuration registers and receives encoder status.
- output compressed bitstreams to host.
- monitors the input frames before and during encoding.
- operates up to 33 MHz.

2.2 Parallel Port Interface

- Supports Plug & Play using Nibble mode reverse channel transfer.
- Supports industrial standard SPP/EPP/ECP interface for configuration register programming and encoder status receiving.
- Supports industrial standard EPP/ECP interface for reverse data transfer to host.
- output compressed bitstreams to host.
- Monitors the selected input frames before and during encoding.

2.3 Generic Bus Interface

- Supports 32-bit bus interface.
- programs encoder configuration registers and receives encoder status.
- operates up to 54 MHz.

2.4 VCD Decoder Interface

- Support the VCD decoder interface (8051-like) to provide the function that VCD bitstreams can be sent from PCI or Parallel port host and pass through W99200F to VCD decoder.
- Glueless interface between W99200F and VCD decoder.
- W99200F and VCD decoder share the single 16M bits SDRAM.

2.5 Audio Bitstream Input Interface

- Supports audio data input interface to provide the function that audio PCM data or compressed bitstreams can pass through W99200F and be sent to PC.

- Mode I: without external audio FIFO. Support I2S interface to get bitstreams. Glueless interface between audio codec and W99200F.
- Mode II: with external audio FIFO. Support 8-bits parallel FIFO data interface. The size of external FIFO can be up to 256K bytes.

2.6 Live Video Encoding

- Real time (29.97-NTSC, 25-PAL) frames/seconds IBP MPEG1 encoder.
- Real time (29.97-NTSC, 25-PAL) frames/seconds M-JPEG encoder
- Encoding in SIF and QSIF resolutions.
- Video input format is selectable to one of: CCIR601 NTSC, CCIR601 PAL and SQUARE NTSC. Cropping and decimation are done according to the output formats (See Table 2-1).

	Input Video Format	Input Video Resolution	Output Video Resolution (4:2:0)				
			FULL	SIF		QSIF	
			Cropping	Cropping	Decimation	Cropping	Decimation
1.	CCIR601 NTSC	720 x 480	704 x 480	704 x 480	352 x 240	704 x 448	176 x 112
2.	CCIR601 PAL	720 x 576	704 x 576	704 x 576	352 x 288	704 x 576	176 x 144
3.	SQUARE NTSC	640 x 480	-	-	320 x 240	640 x 448	160 x 112

Table 2-1 : Input and output live video formats

- Decimation filters are

	Luminance		Chrominance	
	Horizontal	Vertical	Horizontal	Vertical
FULL	-	adaptive field averaging	-	adaptive field averaging + 2 tap 2:1 decimation
SIF	17 tap 2:1 decimation	adaptive field averaging + 3 tap 2:1 decimation	10 tap 2:1 decimation	drop field + 10 tap 2:1 decimation
QSIF	21 tap 4:1 decimation	drop field + 17-tap 2:1 decimation	9 tap 4:1 decimation	drop field + 9-tap 4:1 decimation

Table 2-2: Decimation FIR filter

- Supports I, P and B frames and a programmable GOP structure (m and n values):

m	n	Frame Sequence	Comments
1	1	IIIIII..IIII	
1	2 <= n <=60	IPPPP..PI	
2	2	BIBIBIBI..BI	
2	2 <= n <=60	BIBPBPBP..BI	n grows in steps of 2: 2,4,...,60
3	3 <= n <= 60	BBIBBPBBP..BBI	n grows in steps of 3: 3,6,...,60
4	4 <= n <= 60	BBBIBBBPBBBP..BBBI	n grows in steps of 4: 4,8,...,60

Table 2-3: supported GOP structures

- Bit rate is selected via the host interface. Supported bit rate range (in steps of 400 bits/second):

Frame size	MPGE1 "I" frame (bits/second)	MPEG1 "IBP" (bits/second)
SIF	1.5 M - 9.0 M	192 K - 6.0 M
QSIF	0.5 M - 3.0 M	64 K- 2 .0 M

Table 2-4: supported bit rate range

- Three bit rate policies are supported for MPEG1 encoding:

  - Constant bit rate: stuffing included when needed.
  - Maximum bit rate: no stuffing.
  - Variable bit rate: constant quality.

  Note: For M-JPEG encoding, only the last policy (variable bit rate) is supported.

- An on chip motion estimation unit to support a Search Window (SW) with Half Pixel accuracy. SW sizes:

		SIF	QSIF
Search Technique (full-pel)		2 stage logarithmic	Exhaustive
Search Window	P frame	±41.5 (H) - ±19.5 (V)	±31.5 (H) - ±15.5 (V)
	B frame	±25.5 (H) - ±15.5 (V)	±15.5 (H) - ±15.5 (V)

Table 2-5: motion estimation algorithm and search window

- Automatic Inverse Telecine for 29.97 frames/second video input (Optional).
- Automatic scene change control (Optional.)
- Automatic time code generation, initial value programmed by the host. Drop Frame Flag is supported for 29.97 frames/seconds video.
- Insertion of a user defined bit-map into the encoded video with two level alpha-blending (Optional).
- Insertion of Time-stamp into the MPEG1 bit-stream (Optional).
- Skipping B-frames by repeating last P-frame for low-bitrate encoding (Optional).
- Host selectable encoding parameters:
  - Encoding picture size: SIF/QSIF
  - Encoding Format: MPEG1 / M-JPEG
  - Bit rate policy: constant, maximum, variable.
  - GOP structure: m, n.
  - Half pixel control: on/off.
  - B start sequence: on/off .
  - Open/Close GOP.
  - Inverse telecine control: on/off.
  - Scene change control: on/off
  - Initial time code.
  - Headers frequency: Sequence header, GOP header, Slice header.
  - VBV size and initial fullness.

- Status:
  - FIFO status
  - Video decoder status
  - Encoding status (available once per frame):

    Frame time code.

    Frame type: I,B,P.

    Frame bit budget

    VBV level.

    First frame in a new scene.

2.7 Live Video Pass Through

- Passes YCbCr 4:2:0 video from the video decoder to the host
- Frame resolution supported are SIF and QSIF.
- Cropping and decimation according to Table 2-1.
- Frame rate selectable by the host.
- Insertion of a user defined bit-map into the incoming video with two level alpha-blending (Optional).

2.8 Live Video Snap Shot

- Capture a single frame from the video decoder on the SDRAM in YCbCr 4:2:0 format.
- Frame resolution supported are FULL, SIF and QSIF.
- Cropping and decimation according to Table 2-1.
- Insertion of a user defined bit-map into the incoming video with two level alpha-blending (Optional).

2.9 Single Frame Encoding

- Encoding single picture in MPEG1 I frame or in JPEG.
- Source picture stored on the SDRAM as YCbCr 4:2:0 in any resolution which is aligned to MB borders from 2×2 MBs (32×32 pixels) up to 45×36 MBs (720×576 pixels).
- Encoded picture in the same resolution as the input picture.

2.10 Write / Read a frame

- A frame previously stored in the SDRAM (e.g., by Live Video Snap Shot) can be read out to the host in YCbCr format.
- The host can write a frame into the SDRAM in YCbCr format. This frame can be encoded later on by using single frame encoding mode.

2.11 Write / Read from / to SDRAM

- The W99200F enables the host to write/read any data to/from the SDRAM. This may serve the system to enlarge its memory.

2.12 Write / Read from /to internal memories

- The W99200F enables the host to write/read part of its internal memories. This is done in order to enable future modifications to the encoder and by that increases its programmability.

2.13 System Consideration

- Glueless interface to popular video decoders. Host can program the video decoder interface.
- I^2C protocol supported for programming the video decoder.
- Automatic detection of "stop" condition in the incoming video (time-out on the video decoder "sync" signals).
- Only one component of memory is needed: 1M × 16 bit SDRAM .
  256K Bytes (256K × 8 bit) video FIFO is implemented in the SDRAM.
- W99200F program size is 1536 bytes, to be loaded by the system host at initial configuration time.
- Minimum latency from the incoming video to the output bit stream.
- Perfect Audio/Video synchronization support by stream and by a special pin. PTS(Presentation Time Stamp) for stream support and OSYNC# pin for active video.
- provides programmable I/O pins for system control.

2.14 Technology

- 3.3 voltage operation
- Power consumption (Maximum) : 0.9 W
- Built-in power-saving mode
- Operating frequency: 54MHz
- 160-pin PQFP

3. Pin Configuration

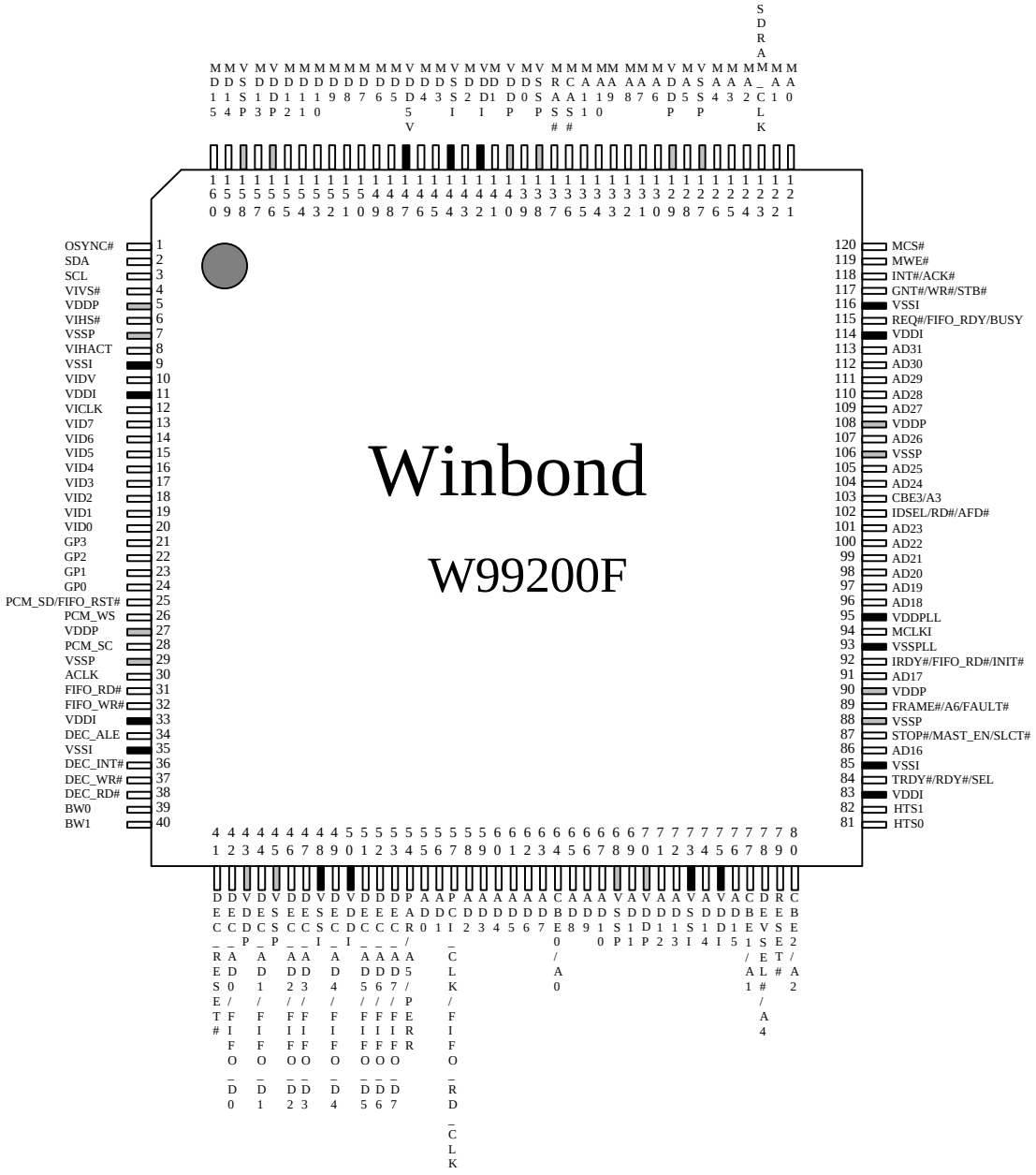


Figure 3-1: Pin Out

4. Pin Description

4.1 Miscellaneous and System Operation Signals (42 pins)

pin name	type	pin description
MCLKI	I	Main clock input. An appropriate choice for system operation is 27 MHz. W99200F uses an internal PLL to generate main operating clock. The frequency of main operating clock is two times that of MCLKI.
HTS[1:0]	I	Host type select. 00, means 32-bit PCI bus is selected, 01, means ECP/EPP parallel port is selected, 10, means generic bus is selected, 11, means test mode to tri-state all output signals.
GP[3:0]	I/O	Programmable general purpose input/output pins.
VDD5V (1)		5 voltage power supply.
VDDPLL (1)		4.1 voltage power supply for PLL circuit.
VSSPLL (1)		analog ground for PLL circuit.
VDDI (7), VDDP (9)		3.3 voltage power supply
VSSI (7), VSSP (9)		0 voltage ground

4.2 Host Bus Signals (50 pins)

pin name	type	pin description
RESET#	I	System reset, active low.
PCI_CLK/ FIFO_RD_CLK	I	In PCI bus, PCI bus clock, up to 33 MHz. In Generic bus, fifo read clock, up to 54MHz.
AD[31:0]	I/O	In PCI bus, 32-bit multiplexed system address and data bus. In Generic bus, Data bus. In Parallel Port, AD[7:0] is used as PD[7:0], parallel port data bus.
FRAME#/ A[6]/ FAULT#	I/O	In PCI bus, Cycle Frame, active Low. In Generic bus, address bus bit 6. Input only. In Parallel Port, fault, active low. Output only.
PAR/ A[5]/ PERR	I/O	In PCI bus, parity is even parity across address bus and CBE, required by all PCI agents. In Generic bus, address bus bit [5]. Input only. In Parallel Port, paper error, active high. Output only.
DEVSEL#/ A[4]	I/O	In PCI bus, Device Select, active Low. In Generic bus, address bus bit [4]. Input only.
CBE[3:0]/ A[3:0]	I/O	In PCI bus, Bus Command and Byte Enable defines bus command and valid byte. In Generic bus, address bus bits [3:0]. Input only.
TRDY#/ RDY#/ SEL	I/O	In PCI bus, Target Ready, active Low. In Generic bus, access operation is finished, active low. Output only.

		In Parallel Port, select, active high. Output only.
IRDY#/ FIFO_RD#/ INIT#	I/O	In PCI bus, Initiator Ready, active Low. In Generic bus, FIFO read signal, active low. Input only. In Parallel Port, initial, active low. Input only.
STOP#/ MAST_EN/ SLCT#	I/O	In PCI bus, Transaction Vstop, active Low. In Generic bus, master mode enable. Enables bus ownership request, active high. Input only. In Parallel Port, select in, active low. Input only.
IDSEL/ RD#/ AFD#	I	In PCI bus, Initialization Device Select is used as chip select in configuration transaction cycles. Active high. In Generic bus, read enable, active low. In Parallel Port, auto line feed, active low.
REQ#/ FIFO_RDY/ BUSY	O	In PCI bus, Bus Request indicates to the bus arbiter that the W99200F requires the use of PCI bus. Active low. In Generic bus, Fifo data is ready, active high. In Parallel Port, not ready to receive data, active high.
GNT#/ WR#/ STB#	I	In PCI bus, Bus Grant indicates to the W99200F that the PCI bus is granted. Active low. In Generic bus, write enable, active low. In Parallel Port, strobe signal, active low.
INT#/ ACK#	O	In PCI and Generic bus, Interrupt Request. Active low. In Parallel Port, acknowledge, active low.
BW[1:0]	I	In Generic bus, Bus Width, 00: 8 bits, 01: 16 bits, 10: not used, 11: 32 bits.

4.3 DRAM Interface Signals (33 pins)

pin name	type	pin description
SDRAM_CLK	O	SDRAM clock. Its frequency is the twice of MCLKI. It can programmably be tri-state. An appropriate frequency for system operation is 54 MHz.
MA[11:0]	O	SDRAM address bus. It can programmably be tri-state.
MD[15:0]	I/O	SDRAM data bus. It can programmably be tri-state.
MCS#	O	SDRAM chip select, active low. It can programmably be tri-state.
MRAS#	O	SDRAM row address select, active low. It can programmably be tri-state.
MCAS#	O	SDRAM column address select, active low. It can programmably be tri-state.
MWE#	O	SDRAM write enable, active low. It can programmably be tri-state.

4.4 Video Decoder Interface Signals (14 pins)

pin name	type	pin description
VICLK	I	Video Input Clock. Free running clock. Allowed frequency: 24.54MHz -

		36MHz.
VID[7:0]	I	Video Input Data. VID[7:0] for Y/Cb/Cr[7:0].
VIDV	I	Video Input Data Valid. Pixel data is sampled only if this signal is high (default). Polarity is programmable via register 0x19.
VIHACT	I	Video Input Horizontal Active pixels identifier. When high (default) indicates the active period of a video line. During VBI there may be an activation of this signal. However, No pixel data is sampled during a VBI window. When active during an active video line, pixel data is sampled with every rising edge of VICLK, where VIDV is active. Polarity is programmable via register 0x19.
VIHS#	I	Video Input Horizontal Sync signal. Falling edge (Default) indicates a new line. Signal polarity is programmable via register 0x19.
VIVS#	I	Video Input Vertical Sync signal. Falling edge (default) indicates a new field. Signal polarity and line offset are programmable via registers 0x18 and 0x19.
OSYNC#	O	Video Out of Sync signal. It is used to indicate the input video data is out of sync. While it is active, system should stop to latch audio data. Active low.

4.5 I^2C Interface Signals (2 pins)

pin name	type	pin description
SCL	I/O	I^2C clock
SDA	I/O	I^2C data

4.6 Decoder Interface Signals (13 pins)

pin name	type	pin description
DEC_RESET#	O	Decoder reset signal, active low.
DEC_AD[7:0]/ FIFO_D[7:0]	I/O	In decoding mode, they are used for decoder address/data bus. Output for multiplexed 8-bit address bus and data bus. Input for 8-bit data bus. In encoding mode, they are FIFO parallel data input.
DEC_ALE	O	Decoder address latch enable for 8051-series uP, active high.
DEC_RD#	O	Decoder read enable, active low.
DEC_WR#	O	Decoder write enable, active low.
DEC_INT#	I	Decoder interrupt request input, active when an interrupt event is triggered, active low.

4.7 Audio Data Interface Signals (6 pins)

pin name	type	pin description

ACLK	**I**	**Audio master clock. Its frequency is programmable. The frequency should be the multiple of 44.1KHz. The default frequency is 11.2896MHz. It is used to generate PCM_SC and PCM_WS signals.**
PCM_SC	**O**	**Audio codec clock.**
PCM_WS	**O**	**Audio codec clock.**
PCM_SD/ FIFO_RST#	**I/O**	**In I^2S interface mode, serial data. Input. In FIFO interface mode, it is used to reset FIFO. Active low. Output.**
FIFO_RD#	**O**	**In I^2S interface mode, it is not used. In FIFO interface mode, FIFO read enable. Active low.**
FIFO_WR#	**O**	**In I^2S interface mode, it is not used. In FIFO interface mode, FIFO write enable. Active low.**

5. System Diagram

W99200F is a single-chip real time MPEG-1 and Motion JPEG Video Encoder for low-cost PC-based video capture and encoding. It is companioned with software MPEG-1 audio encoder, A/V mixing and authoring tools. It has two kinds of H/W configuration: (1) external box with connection to PC parallel port, (2) PCI add-on card. It provides modes of Pre-view, Real-time encode, Monitoring and I-frame Edit.

Figure 5-1 shows a complete video encoding/decoding system. This system contains the following parts:

1. W99200F.
2. Video decoder chip.
3. VCD decoder chip. (Glueless interface to Winbond W9925QF and W9926QF.)
4. Audio codec.
5. TV encoder.
6. One 1M x 16 bit SDRAM component. This SDRAM is shared by W99200F and VCD decoder.

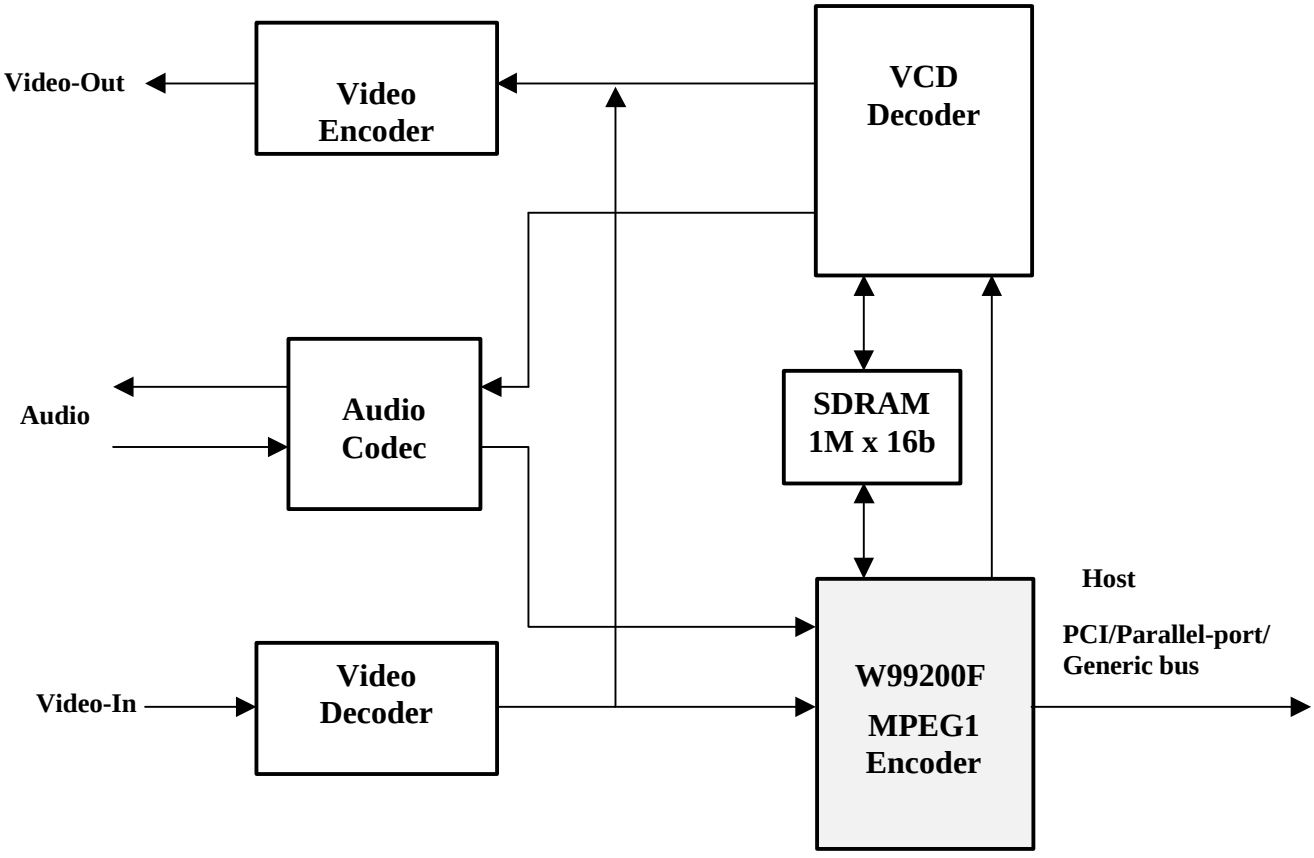


Figure 5-1: W99200F A/V system

Figure 5-2 shows a simple PC based capturing and encoding system which is based on the W99200F. Such a system contains the following parts:

1. W99200F
2. Video decoder chip that captures the analog live video from the camera or VCR and converts it to a digital YCbCr 4:2:2 bit stream (glueless interface to popular video decoders, such as: Philips SAA7110/11/12/13/14, BrookTree Bt827/9, Samsung KSO122/127, etc.).
3. One 1M × 16b SDRAM component. This SDRAM is used as a frames buffer and as the video bit stream FIFO.

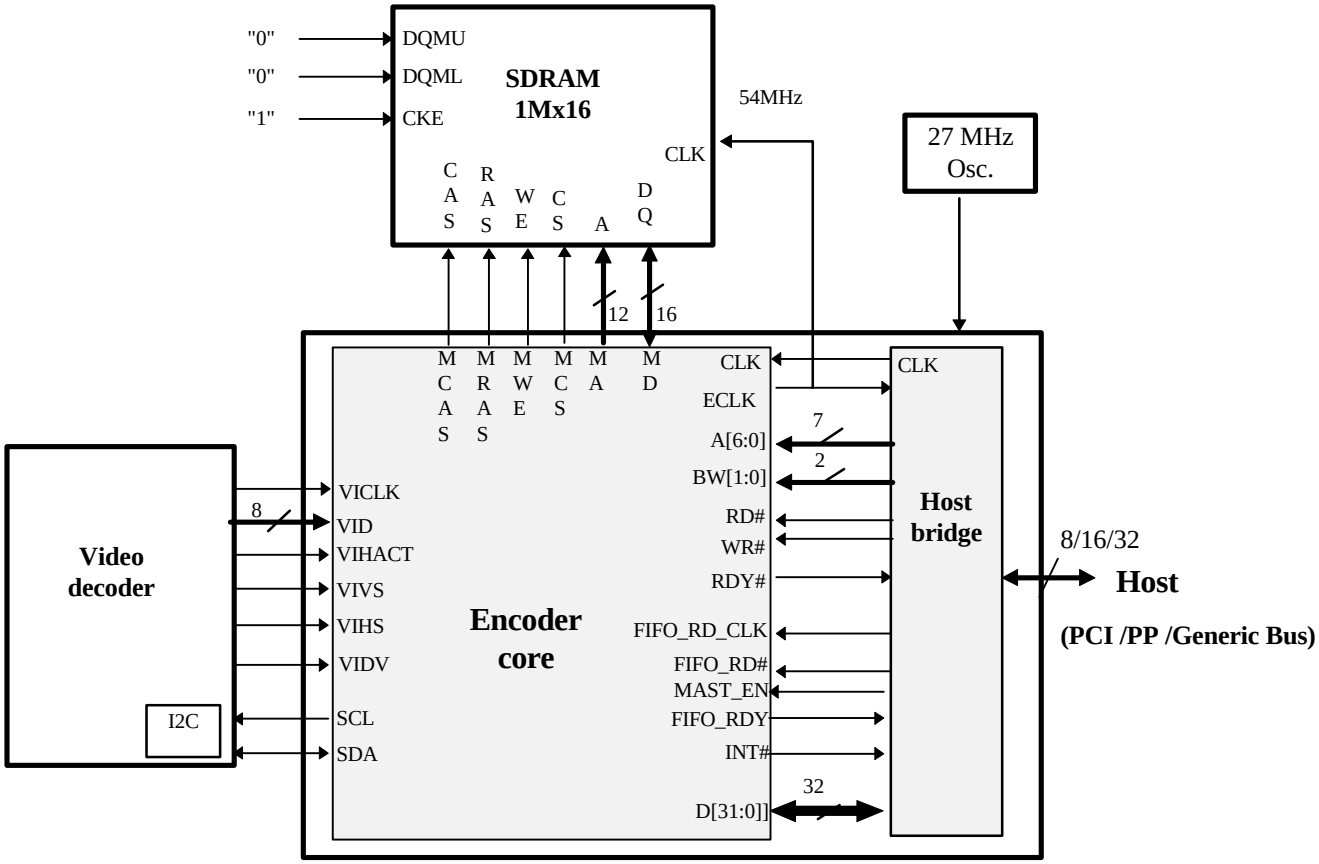


Figure 5-2: W99200F video system

6. Block Diagram

This chapter describes the functionality of the main units of the W99200F. Figure 6-1 shows the block diagram of the W99200F. The following sections describe the functionality of each block.

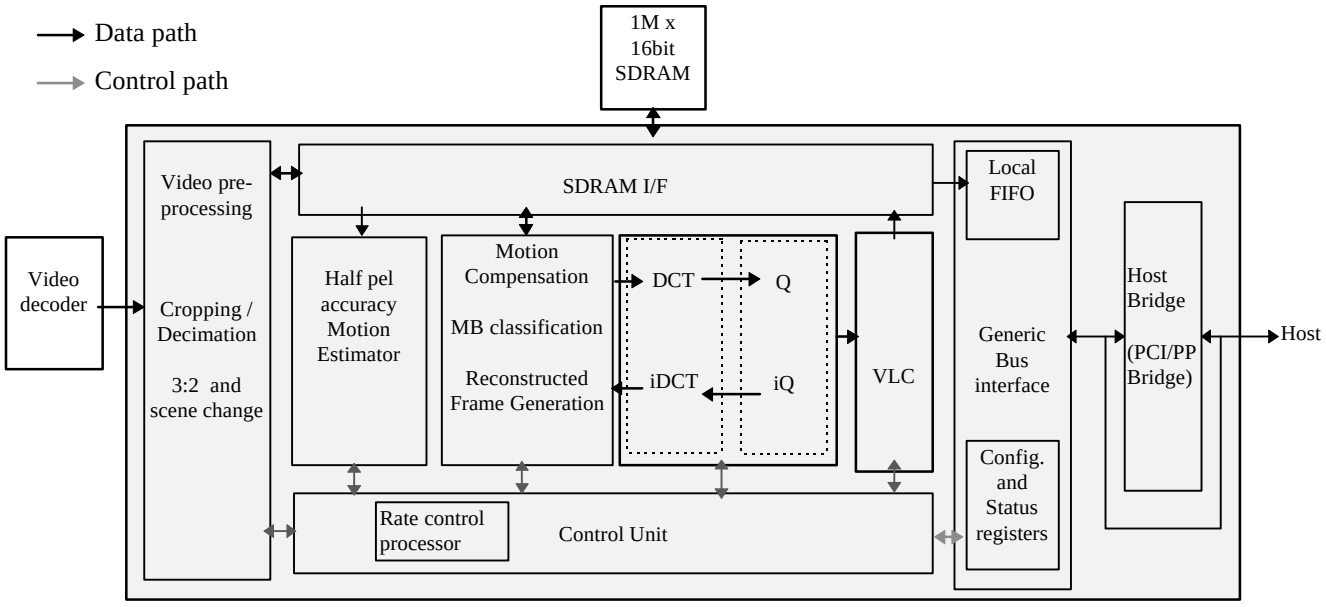


Figure 6-1: W99200F block diagram

6.1 Video pre processing

The video input interface is responsible for the following three major activities:

- Decimation and cropping.
- Telecine detection.
- Scene change detection.

6.1.1 CCIR to SIF Decimation.

This block gets the 4:2:2 video input from the video decoder and performs decimation to the requested frame size according to Table 2-2.

6.1.2 Telecine Detection

MPEG-1 provides means to encode film originated NTSC video stream by specifying the frame rate in the Video Sequence header. The W99200F detects an NTSC video stream generated by a 3:2 pulldown process, and convert it back into film rate stream, dropping the redundant fields. In that case the W99200F indicates a 23.976 frame/sec rate in the Video Sequence header.

6.1.3 Scene Change Detection

A video stream may have scene changes within. An ordinary scene change is associated with a significant difference between the last frame of the previous scene and the first frame of the new scene. The W99200F detects that difference, and recommends to encode the new scene's first as a first frame of a new group of pictures (either an I frame or a "closed GOP" B frame), thus improving the encoded video quality.

6.2 Memory (SDRAM) Interface

SDRAM interface block is responsible to the write and read transactions to the SDRAM as well as refresh cycles. It does not arbitrate the SDRAM accesses (priority determination), but rather translates read and write commands with its addresses to SDRAM legal sequences.

6.3 Motion Estimation unit

The Motion Estimation unit finds a Minimum SAD for every Template MB (TMB) in "P" and "B" frames. The Motion estimation works in full-pel and half-pel resolution. The full-pel search is done according to Table 6-1. The half-pel search is done on the eight half-pel vectors which surrounds the best full-pel motion vector.

		SIF	QSIF
Search Technique		2 stage logarithmic	Exhaustive
Search Window	P frame	±41 (H) - ±19 (V)	±31 (H) - ±15 (V)
	B frame	±25 (H) - ±15 (V)	±15 (H) - ±15 (V)

Table 6-1: Full-pel motion estimation algorithm and search window

6.4 Motion Compensation

Motion compensation block. This block performs the following tasks:

- Calculate the Intra SAD
- Calculate the interpolated SAD ("B" frame)
- Do the MB classification (fwd, bwd, itp, zero-motion, etc)
- Calculate the motion compensated MB: the differences between the original MB and its predictor ("P" and "B" frames), according to rate-control unit MB classification decision.
- Calculate the reconstructed frame ("I" and "P" frames). In SIF encoding calculate the reconstructed decimated (QSIF) Luma parts (for ME).

6.5 DCT/Q/iQ/iDCT

DCT/iDCT block performs the Discrete Cosine Transform and the inverse Transform. Q/iQ block performs the quantization and the inverse quantization according to the MPEG1 / M-JPEG standards.

6.6 Variable Length Coding

Variable Length Coding (VLC) block is responsible for the following:

1. Generate MPEG1 bit stream when encoding is MPEG1.

2. Generate M-JPEG bit stream when encoding is M-JPEG.

The produced bit stream is written into the SDRAM FIFO.

6.7 Control Unit

Control unit controls all W99200F activity. It is responsible for:

1. Synchronization between the different blocks.
2. Time code generation.

6.8 Rate Control Processor

The rate control processor is responsible for controlling the Frame/MB bit budget to meet the required MPEG1 stream bit rate. There are three supported modes of operations:

1. Constant bit rate. The RCP is responsible for shuffling the bit budget among frames and MB according to complexity of the picture or a portion of a picture. It is responsible for the VBV control, zero stuffing, etc.
2. Maximum bit rate. The same as above, with no zero stuffing.
3. Constant quality. I.e., constant quantization factor. In that case there is no need to handle a VBV. Also, no stuffing is needed.

The Rate-control is also performs MB classification.

6.9 Generic Bus Interface

Generic Bus Interface is the unit that performs the generic bus protocol and is connected to the host bridge. The generic bus interface also includes the local FIFO and the configuration and status registers.

6.9.1 Local FIFO

Local FIFO is a small FIFO from which the output bit stream is read to the host port.

6.9.2 Configuration and status registers

Through these registers the host controls the W99200F.

6.10 Host Bridge

Host Bridge is the unit that decodes host bus signals and transfer host operation to generic bus signals. It includes PCI interface, parallel port interface, decoder interface and audio input interface.

7. Functional Description

7.1 Host Bus Interface

The host bus interface has built for interfacing with three kinds of host bus, which are:
1. 32-bit PCI bus
2. parallel port
3. generic bus

7.1.1 32-bit PCI Bus

The PCI block provides an easy interface with PCI bus such that no extra bridge components are needed. The W99200F provides two operating modes in PCI bus: bus target and bus master. When acting as a bus target, the W99200F responds to I/O-mapped, memory-mapped and configuration access cycles. While playing the role of a bus master, the W99200F will issues the memory write command to move the bitstream data to system main memory. There are four types of access cycles, including:

- Configuration cycle,
- Target mode I/O read/write cycle,
- Target mode memory read cycle,
- Master mode memory write Cycle (Initiated by W99200F while it acts as a bus master).

7.1.1.1 Configuration Cycle:

Based on the PCI 2.1 local bus definition, a set of configuration registers are built in the W99200F to achieve the purpose of total software initialization and configuration. The description of configuration registers which are used in the W99200F is given as followings:

31		16 15	0	
Device ID		Vendor ID		00h
Status		Command		04h
Class Code			Revision ID	08h
BIST	Header Type	Latency Timer	Cash Line Size	0Ch
Base Address Register				10h 14h 18h 1Ch 20h 24h
Cardbus CIS Pointer				28h
Subsystem ID		Subsystem Vendor ID		2Ch
Expansion ROM Base Address				30h
Reserved				34h
Reserved				38h
Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	3Ch

Figure 7-1: Layout of PCI Configuration

The following section is the description of each register in the configuration space.

- **Vender ID :** This field identifies the manufacturer of the device. It's a read-only register. For Winbond Electronics Corporation, the Vender ID is 1050h.
- **Device ID :** This field identifies the particular device and it is read-only. For the W99200F, the Device ID is 9922h.
- **Revision ID :** This field specifies a device specific revision identifier and it is read-only. The ID number of this revision is 03h.
- **Header Type :** This byte identifies the layout of bytes 10h through 3Fh in configuration space and also whether or not the device contains multiple function. For the W99200F**,** bit 7 is set to 0 to specifies that the W99200F is single function. Bit[6:0] are set to 00h to specify that the layout of bytes 10h through 3Fh is as shown in Figure 7-1. This register is read-only.
- **Class Code :** This field is used to identify the generic function of device. This register is broken into three byte size field. The upper byte is a base class code, the middle is a sub-class code and the lower byte identifies a specific register-level programming interface. For the W99200F**,** the register is set to 040000h to identify the W99200F is a Multimedia device. This is also a read-only register.
- **Command :** The Command register provides coarse control over the device's ability to generate and respond to PCI cycles. When a 0 is written into this register, the device is logically disconnected from the PCI bus for all accesses except configuration accesses. The definition of each bit in this register is given in Table 7-1.
- **Status :** The Status register is used to record status information for PCI bus related events. The definition of this register is given in Table 7-2. For this register, a bit is reset whenever the register is written and the written data in the corresponding bit location is a 1.

- **Base Address :** This register is used to record the base address of the device. When reading register 10h, the returned values are 1 for bit 0 and 0 for bit 1, it means that the W99200F is mapped into I/O space. When reading register 14h, the returned values are "0000" for bit[3:0], it means that the W99200F is mapped to memory space. This memory space is used to read video FIFO data. When reading register 18h, the returned values are "0000" for bit[3:0], it means that the W99200F is mapped to another memory space. This memory space is used to read audio FIFO data.

  For example:
  (1) I/O space:
  - Step1: The host issues a configuration write command to PCI bridge and address is 0x10, data is 0xffffffff.
  - Step 2: The host issues a configuration read command to PCI bridge and address is 0x10. W99200F will send data 0xfffffff1. It means W99200F needs a 16-byte IO space. (Please refer to PCI spec for detail.)
  - Step 3: The host issues a configuration write command to PCI bridge and address is 0x10, data is , for example, 0x80000. (This address is allocated by OS and called as IO space base address.) Then W99200F will be mapped into IO space and its address is 0x80000 ~ 0x8000f.

    Actually, W99200F is only mapped into 3 double-word IO space. They are, for example, 0x80000, 0x80004 and 0x80008. We call them as AIR, DPR and ODPR in sequence. We will detailedly describe them in the section "I/O Access Cycle".

  (2) memory space:
  - Step1: The host issues a configuration write command to PCI bridge and address is 0x14 (or 0x18), data is 0xffffffff.
  - Step 2: The host issues a configuration read command to PCI bridge and address is 0x14 (or 0x18). W99200F will send data 0xfffff000. It means W99200F needs a 4K-byte memory space. (Please refer to PCI spec for detail.)
  - Step 3: The host issues a configuration write command to PCI bridge and address is 0x14 (or 0x18), data is , for example, 0x2000000. (This address is allocated by OS and called as memory space base address.) Then W99200F will be mapped into memory space and its address is 0x2000000 ~ 0x2000fff.

    We will detailedly describe it in the section "Target mode memory read cycle".
-
- **CardBus CIS Pointer :** This optional register is used by those devices that want to share silicon between CardBus and PCI. For W99200F, the returned value is 0 while reading this register.
- **Subsystem Vendor ID and Subsystem ID :** This is also a read only register. When reading these two register, the returned values are always zero.
- **Expansion ROM Base Address :** It is a read only register and the read value is always kept at zero.
- **Cache Line Size :** This register specifies the system cache line size in unit of 32-bits word. This register is set to 0 to ignore PCI cache support lines. At reset this field should be set to 0.
- **Latency Timer :** This register is readable/writeable register. After hardware reset, it is set to zero and the system BIOS must write an appropriate value to the register.
- **BIST :** The BIST(Built in self test) register is also read-only and the returned value is always zero for W99200F.
- **Interrupt Line :** This is a 8-bit register used to communicate interrupt line routing information. The register is read/write-able. The value in this register tells which input of the system interrupt controller the device's interrupt pin is connected to.
- **Interrupt Pin :** The interrupt pin register tells which interrupt pin the device uses. This register is read only. For the W99200F**,** the returned value would be 1.
- **Min_GNT :** The value of this register specifies how long a burst period the device needs assuming a clock rate of 33 MHz. For the W99200F, this is a read-only register and the returned value is FFh.
- **Max_LAT :** This register is used for specifying how often the device needs to gain access to the PCI bus. For W99200F, this is also a read-only register and the returned value is 01h.

Bit Location	Description
0	Controls a device's response to I/O space access. A value of 0 disables the device response. A value of 1 enables the response. This is a read/write bit and the default value is 0.
1	Controls a device's response to memory space accesses. A value of 0 disables the device response. A value of 1 enables the response. This is a read/write bit and the default value is 0.
2	Controls a device's ability to act as a master of the PCI bus. It is a read/write bit. The default value is 0.
3	Controls the device's action on Special Cycle operation. This bit is read-only and its value is set to 0.
4	Memory Write and Invalidate Enable. This bit is read-only. Its value is 0.
5	VGA Palette Snoop. A read-only bits. A '0' is set to this bit.
6	This is a read only bit and the returned value is always 0. It means that the W99200F will not do the parity check.
7	This bit is read-only and fixed at zero. It means that the W99200F does not do address/data stepping.
8	This is an enable bit for the SERR# driver. A 0 value disables the SERR# driver. A 1 value enables the SERR# driver. After hardware reset, this bit is fixed at 0.
9	Fast Back-to-Back Enable. This is a read-only bit. The returned value is 0. It means that the W99200F doesn't do fast back-to-back transaction in master mode.
10 - 15	Reserved

Table 7-1: Command Register Bits

Bit Location	Description
0 - 4	Reserved
5	This optional read-only bit is fixed at 0 to indicate that the W99200F is capable of running at 33 MHz.
6	This optional read-only bit is set to zero for the W99200F. It means the W99200F does not support User Definable Features.
7	This is a read-only bit and returned value is 1. It means that the W99200F is capable of accepting the fast back-to-back transaction in bus target mode
8	The bit is read only. The returned value is 0.
9 - 10	DEVSEL timing. The two bits encode the timing of DEVSEL#. These bits are read only and indicates the slowest time that a device asserts DEVSEL# for any bus command except Configuration Read and Write. For W99200F, these bits are set to be 01b for medium timing.
11	Signaled Target Abort. This bit must be set by a target when it terminates a transaction with target-abort.
12	Receive Target Abort. While the W99200F operates as a bus master, this bit will be set whenever its transaction is terminated with target-abort.
13	Received Master Abort. If the W99200F acts as a bus master, this bit must be set whenever its transaction is terminated with master abort. It is readable/writeable.
14	Signaled System Error. The returned value is always zero.
15	Detected Parity Error. This bit must be set by the device whenever it detects a parity error, even if parity handling is disabled.

Table 7-2: Status Register Bits

The W99200F is a target of the configuration command (read or write) only when its IDSEL pin is high and AD[1:0] are 00 during the address phase of the configuration command. Internal addressing of the 64-DWORD configuration space is done by AD[7:2], and C/BE[3:0]# indicates the bus command. This timings of configuration read and write command are shown in Figure 7-2 and Figure 7-3 respectively. The mark xxxx in IDSEL means an invalid state. The detail description of timing will be given in the description of timing of IO access.

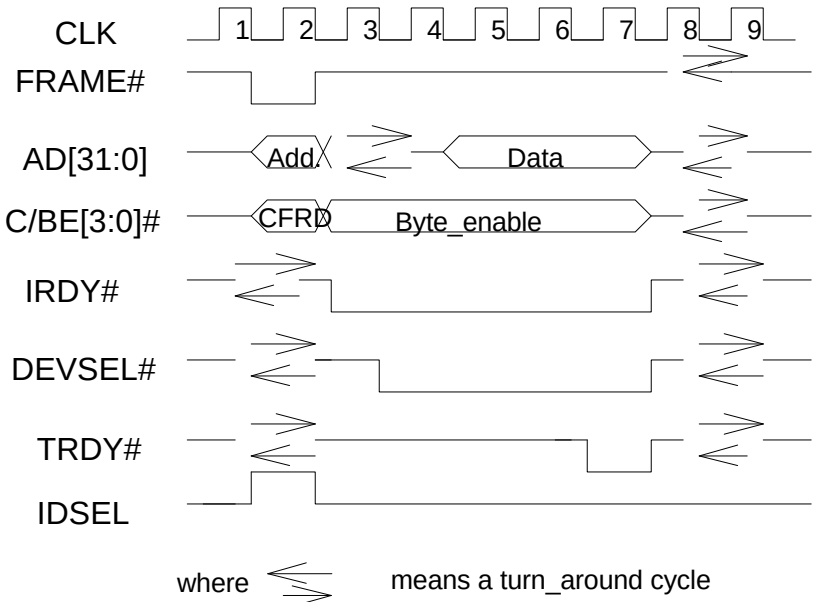


Figure 7-2: PCI Configuration Read Timing

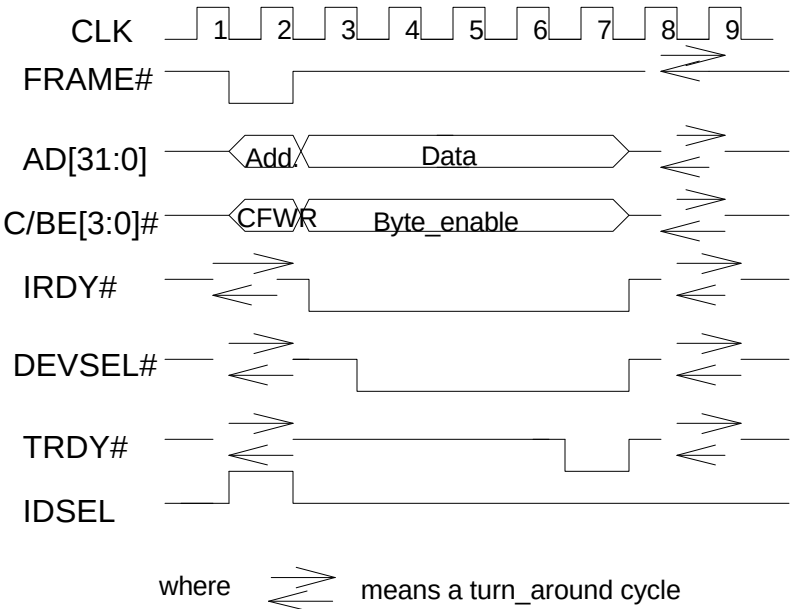


Figure 7-3: PCI Configuration Write Timing

7.1.1.2 I/O Access Cycle:

The host can access the internal control, status registers, local SDRAM by using a set of Address Index Port Register and Data Port Register. All the above access cycles are I/O-mapped access cycles. The timing diagrams of I/O mapped accessing are illustrated in Figure 7-4 and Figure 7-5 for I/O read and I/O write respectively. In Figure 7-4, a Read transaction starts with an address phase which occurs when FRAME# is asserted for the first time and it occurs on clock2 in this figure. FRAME# is deasserted on clock 3 to indicated that the current transaction is single mode. During address phase, AD[31:0] contain a valid address and C/BE[3:0]# contain a valid bus command. The first clock of data phase is clock 3. During data phase, C/BE[3:0]# indicate which byte lanes are involved in the current data phase. DEVSEL# is active on clock 4 to inform host that a device (W99200F) has been selected in this data transaction. The whole data transaction completes when data are transferred which occurs when both IRDY# and TRDY# are asserted on the same clock edge. In this figure, the data transfer completes on clock 7. While the data transfer is done, all signals including FRAME#, IRDY#, TRDY#,DEVSEL# are all deasserted. The notation "⇐" in Figure 7-4 means a turnaround cycle which is necessary for all signal that may be driven by more than one agent. Figure 7-5 illustrates a Write transaction. The Write transaction is similar to a Read transaction except no turnaround cycle is required following the address phase in AD[31:0].

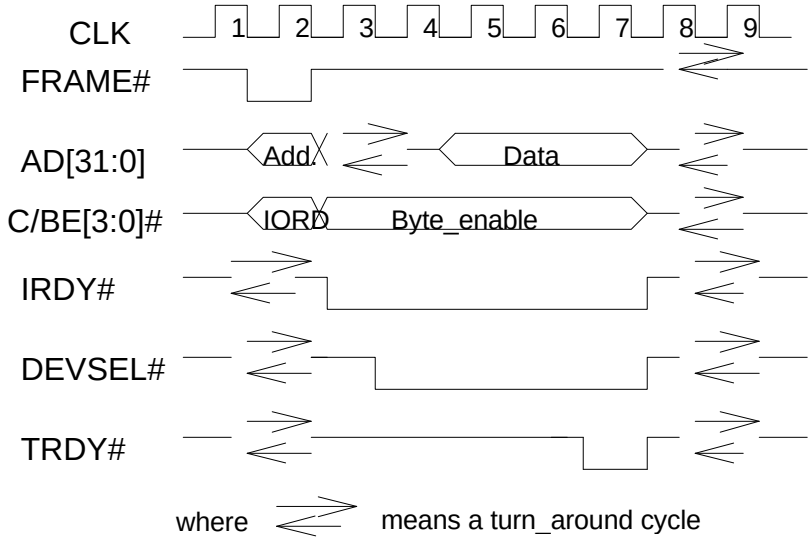


Figure 7-4: PCI I/O Read Timing

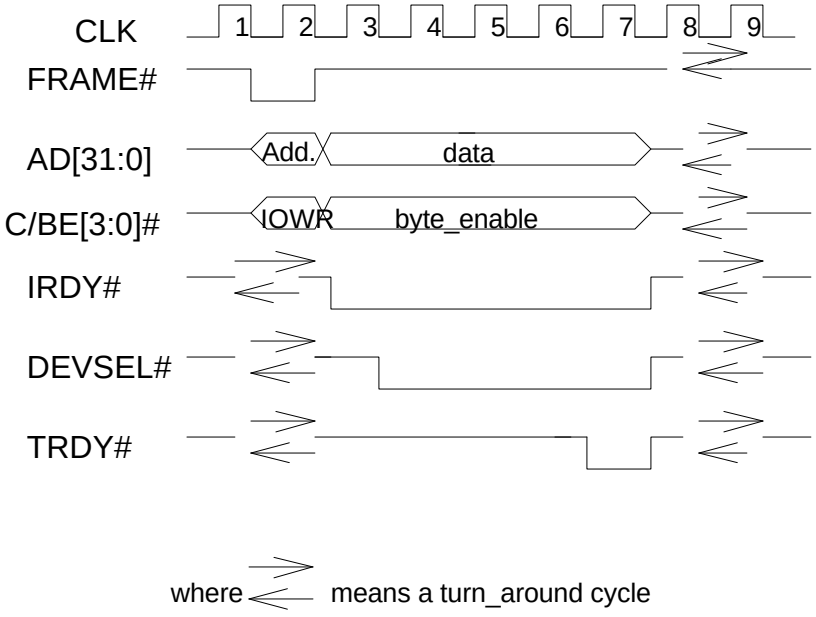


Figure 7-5: PCI I/O Write Timing

7.1.1.3 Target Mode Memory Read Cycle:

The host can read internal Vdata_out FIFO by memory read cycles. The timing diagram of memory read cycle is illustrated in Figure 7-6. A read transaction starts with an address phase which occurs when FRAME# is asserted for the first time and it occurs on clock 2 in this figure. FRAME# is deasserted on clock 3 to indicated that the current transaction is single mode. During address phase, AD[31:0] contain a valid address and C/BE[3:0]# contain a valid bus command. The first clock of

data phase is clock 3. During data phase, C/BE[3:0]# indicate which byte lanes are involved in the current data phase. DEVSEL# is active on clock 4 to inform host that a device (W99200F) has been selected in this data transaction. The whole data transaction completes when data are transferred which occurs when both IRDY# and TRDY# are asserted on the same clock edge. In this figure, the data transfer completes on clock 5, 6 and 7. While the data transfer is done, all signals including FRAME#, IRDY#, TRDY#,DEVSEL# are all deasserted. The notation "  " in Figure 7-6 means a turnaround cycle which is necessary for all signals that may be driven by more than one agent.

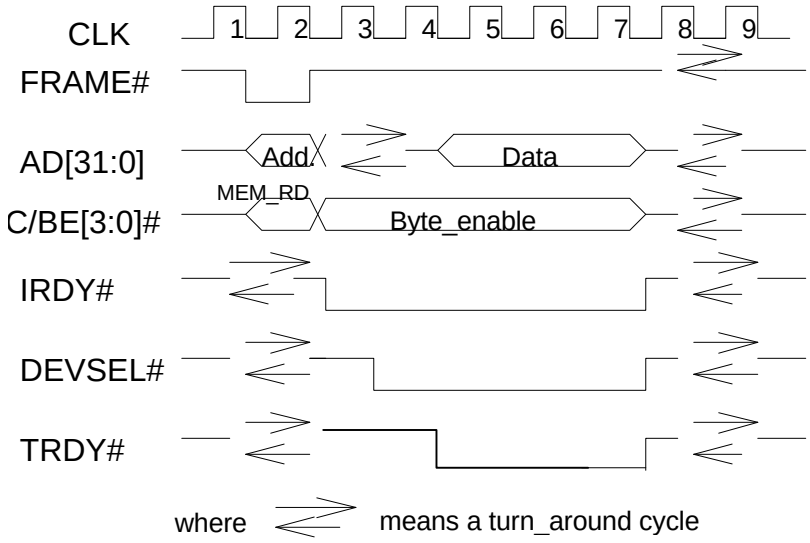


Figure 7-6: PCI Target Mode Memory Read Timing

For example, suppose that the memory mapping space of W99200F is 0x2000000 ~ 0x2000fff, the host can read the data from internal Vdata_out register by the following procedure:

After initialization, while the host receives "FIFO ready" interrupt request, the host first reads the value of "Vthreshold" register. The host should read up to threshold level units of data from video Vdata_out FIFO (or audio FIFO). The host issues the first memory read command, address = 0x2000000, to get the first data of Vdata_out (or Audio_out) register. Issuing the second memory read cycle, address = 0x2000004, to get the second data. And issuing the 3rd memory read cycle, address = 0x2000008, to get the 3rd data, and so on. The host may also issue burst memory read command to read data from Vdata_out (or Audio_out) register. The burst length will be decided by PCI control chip (PC core logic chip). But it is equal or less to threshold level. When the memory address reaches up to 0x2000ffc, the host should issue the memory read cycle, address = 0x2000000, in the next time. Note that the maximal size of memory mapping space must be multiple of threshold level DWORD. That means some region of W99200F memory mapping space (it is fixed to 4K bytes.) may not be used by the host.

Note that PCI bridge doesn't support burst cycle to access memory mapping space II (Audio_out register).

Actually, the target mode memory read cycles with address located at W99200F memory mapping space I will all be translated to FIFO_RD# cycles in generic bus by PCI bridge. And the target mode memory read cycles with address located at W99200F memory mapping space II will all be translated to RD# cycles in generic bus and A[6:0]=0x67.

7.1.1.4 Master Mode Memory Write Cycle:

In 32-bit PCI bus, the input frames are sent to Host via PCI bus master mode. In PCI bus master mode, W99200F plays the role of a bus master and asserts the memory write command to output bitstream data from Vdata_out FIFO in burst mode. The timing of this type of accessing is shown in Figure 7-7. In this mode, the W99200F asserts REQ# to request the ownership of bus to write the frame data to host. If the GNT# is detected active, it means that the request of bus ownership has been granted by the bus arbiter and the W99200F will pull FRAME# to low, deassert REQ#, put the address of target of current access on AD[31:0] and memory write command on C/BE[3:0]# to initiate the memory write bus transaction. IRDY# will be active at the rising at next cycle to indicates that the W99200F is ready to send the data and byte enable will also be put on C/BE[3:0]# to indicates which byte lane is desired simultaneously. The pointed target will receive the data while both TRDY# and IRDY# are low. Data transfer will be continued until FRAME# is driven to high to indicate that the last DWORD will be transferred at next cycle. Whole bus transaction will be completed when the last DWORD data is written.

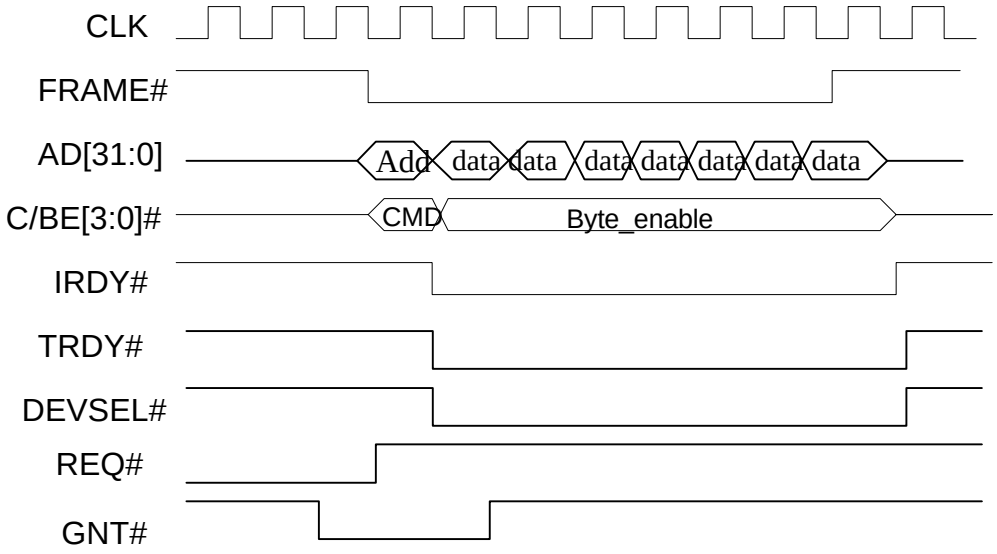


Figure 7-7: PCI Memory Write Timing of Bus Master in Burst Mode

The procedure of master mode memory write cycle:

1. After initialization, the host programs BBSAR0, BBSAR1, BBS0 and BBS1 and sets BB_RDY0=1. Then set BB_RDY1=1.
2. The host sets MASTER_EN=1.
3. If FIFO_RDY =1, PCI bridge asserts REQ# to PCI bus.
4. While GNT# is low, FRAME# and IRDY# are high, PCI bridge issues master mode memory burst write command to PCI bus. Burst length is equal to threshold level. The address is equal to BBSAR0. This command will move the data from Vdata_out FIFO to Buffer0 in the system main memory.
5. Goto step 3 and the address is increased 4* threshold level, until Buffer0 is full.
6. While Buffer0 is full, PCI bridge automatically sets BB_FULL0 "1", and BB_RDY0 "0". And PCI bridge asserts INT# to the host.
7. At that time, if BB_RDY1=1, PCI bridge starts to move the data from Vdata_out FIFO to Buffer1.

8. In the mean time, the host responds INT# by reading BSTR. If the host finds BB_FULL0 =1 and BB_FULL1 =0, the host programs new BBSAR0 and BBS0. Then the host sets BB_RDY0=1 and clears BB_FULL0. The host starts to move out the data of the original Buffer0 to other storages, for example, hard disk.
9. While Buffer1 is full, PCI bridge automatically sets BB_FULL1 "1", and BB_RDY1 "0". And PCI bridge asserts INT# to the host.
10. At that time, if BB_RDY0=1, PCI bridge starts to move the data from Vdata_out FIFO to Buffer0.
11. In the mean time, the host responds INT# by reading BSTR. If the host finds BB_FULL1 =1 and BB_FULL0 =0, the host programs new BBSAR1 and BBS1. Then the host sets BB_RDY1=1 and clears BB_FULL1. The host starts to move out the data of the original Buffer1 to other storages, for example, hard disk.
12. Goto step 6.

7.1.1.5 I/O Registers

7.1.1.5.1 IO Register Decoding

For the 32-bit PCI bus, based on the PCI Local Bus specification 2.1, the W99200F provides a set of configuration registers to achieve the purpose of total software initialization and configuration. By using these configuration registers, system will allocate proper resources to the W99200F, like I/O port base address and interrupt line, etc. The user can get these resource information by reading the corresponding configuration registers and using the I/O base address allocated by system BIOS as the base address of I/O register. The W99200F will decode the base address at the address phase of each IO-mapped bus transaction to determine whether to response or not.

7.1.1.5.2 I/O Registers Mapping

There are three I/O registers which are DWORD addressable in a 32-bit PCI host bus configuration. The corresponding I/O addresses and their type are specified as follows:

(1) DWORD address configuration for PCI bus:

Register Name	Address	Type
AIR	IOAR	read/write
DPR	IOAR+4h	read/write
ODPR	IOAR+8h	read

IOAR will be decoded in address phase of bus transaction to identify whether W99200F is the target of current bus transaction or not and the values, 04h, or 08h will be used to select which I/O port is accessed in current transaction. The content of IOAR is stored in the configuration register whose address is 10h in the configuration space.

Address Index Register (AIR)

AIR[6:0]
register index

Address Index Register, AIR, is used to specify the index value of internal register. When PCI host access AIR, AIR[6:0] are equal to the data phase of AD[6:0], and AD[31:7] are not used.

Data Port Register (DPR)

DPR
bit[31:0] or bit[7:0]

Data Port Register, DPR, is the register data port. When the host accesses DPR, the internal register indexed by AIR is read or written. When the internal register is 8-bit wide, DPR[7:0] are equal to the data phase of AD[7:0] and AD[31:8] are not used. When the internal register is 32-bit wide, DPR[31:0] are equal to the data phase of AD[31:0].

Output-FIFO Data Port Register (ODPR)

ODPR
bit[31:0]

Output-FIFO Data Port Register, ODPR, is the output port of the Output-FIFO. Host can read this register to get the bitstream data.

For example:

1. If the host wants to write data to "Vdata_in " register (index is 0x01), the host should first issue a IO write cycle, address = 0x80000 and data = 0x01. This cycle is used to set AIR as 0x01. Then the host issues a IO write cycle, address = 0x80004 and data = (the data that the host wants to write.) If the host wants to continuously write data to "Vdata_in " register, the host only needs issue a IO write cycle, address = 0x80000 and data = 0x01, once. And then it can continuously issue a IO write cycle, address = 0x80004 , many times. That means the value of AIR will keep the current value until it is written again.

2. If the host wants to read "Vthreshold" register value (index is 0x10), the host should first issue a IO write cycle, address = 0x80000 and data = 0x10. This cycle is used to set AIR as 0x10. Then the host initiates a IO read cycle, address = 0x80004. PCI bridge will send the data of "Vthreshold" register to the host.

3. If the host wants to get data from "Vdata_out " register, it should issue a IO read cycle, address = 0x80008. PCI bridge will send the data of "Vdata_out " register to the host. Note that it is not necessary to set AIR value before reading ODPR register.

7.1.2 Parallel Port Bus Interface:

The W99200F supports industrial standard SPP/EPP/ECP for PC interface. It also supports device ID in Nibble Mode reverse channel transfer. But W99200F doesn't support the data transfer using Nibble mode. The following interface signals, according to IEEE P1284 standard, are redefined in SPP/EPP/ECP modes:

SIGNAL (DRIVEN)	SPP	EPP	ECP
NSTB# (Host)	nStrobe	nWrite	HostClk
NAFD# (Host)	nAutoFd	nDStrb	HostAck
NINIT# (Host)	nInit	nInit	nReverseRequest
NSLCT# (Host)	nSelectIn	nAStrb	1284 Active

NACK# (Peripheral)	nAck	Intr	PeriphClk
BUSY (Peripheral)	Busy	nWait	PeriphAck
PERR (Peripheral)	PError	(Not used)	nAckReverse
NFAULT# (Peripheral)	nFault	(Not used)	nPeriphRequest
SEL (Peripheral)	Select	(Not used)	Xflag

Table 7-3: Parallel Port Signals

7.1.2.1 Negotiation

The interface is always initialized to the SPP mode. Host may reinitialize the interface at any time by asserting nInit (NINIT#) low in conjunction with nSelectIn (NSLCT#) low. The W99200F returns to the Compatibility Mode Idle phase after initialization.

A 1284 compliant host negotiates with the W99200F to verify that the W99200F is P1284 compliant. Upon the verification the host will request a communication mode for the W99200F. The W99200F will acknowledge the communication mode request based on its capabilities and execute or reject the request as appropriate. This is accomplished by the host placing an extensibility request value on the data bus during the negotiation phase. Table 7-4 shows the communication modes supported by the W99200F.

MODE DEFINITION	EXTENSIBILITY REQUEST VALUE
Request EPP Mode	0100 0000
Request ECP Mode	0001 0000
Request Device ID Using Nibble Mode Rev Chan Transfer	0000 0100

Table 7-4: Parallel Port Communication Modes

7.1.2.2 Device ID

The Device ID is a length field followed by a string of ASCII characters defining W99200F characteristics and capabilities. The Device ID will be sent from the W99200F to the host using the Nibble Mode, which is required by the P1284.

The W99200F Device ID contains four keys: MANUFACTURER, MODEL, CLASS and DESCRIPTION. Currently implemented ID string is shown below:

x′00′, x′66′,
MFG:Winbond;
MDL:W99200F;
CLS:MEDIA;
DES:Winbond's VIDEO ENCODER driver can not be found in the system;

7.1.2.3 Device Addressing

The bi-directional SPP/EPP/ECP protocal describes two basic types of 8-bit information transfers: data read/write operations and address read/write operations. Table 7-5 specifies the control signals used by each mode to perform address/data cycle and to indicate reverse data flow.

SIGNAL	SPP	EPP	ECP
Address Strobe	NSTB (NAFD=0)	NSLCT	NSTB (NAFD=0)
Data Strobe	NSTB (NAFD=1)	NAFD	NSTB (NAFD=1)
Reverse Channel	Not supported	Implicit	NINIT=0

Table 7-5: Parallel Port Read/Write Signals

7.1.2.4 Standard Parallel Port (SPP)

1.In SPP mode, the host can write internal registers through the command/data transfer. With command transfer the host can issue the register address A[6:0]. With data transfer the host can deliver the 8-bits data.
2. The command and data transfer in SPP mode is differentiated by ″nAutoFd″. ″nAutoFd″ low is command transfer, whereas ″nAutoFd″ high is data transfer.
3. To write register in SPP mode, host should issue two bytes transfer :
   SPP-command-transfer, with PD[6:0] = register address[6:0].
   SPP-data-transfer, PD[7:0] transfer the 8bits register data.

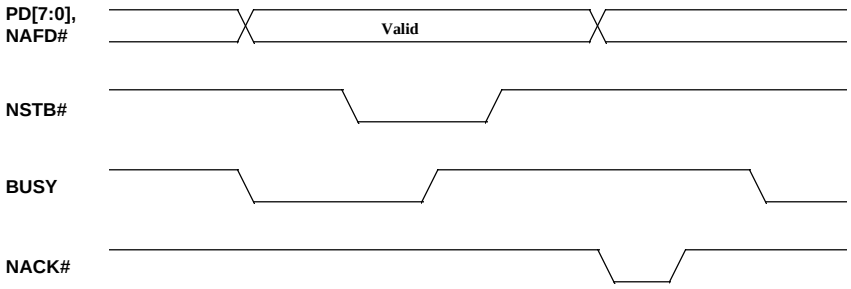


Figure 7-8: Standard Parallel Port (SPP) Timing

7.1.2.5 Enhanced Parallel Port (EPP)

1. The host should go through the negotiation process to enter the EPP mode.
2. The host can do several bytes of write/read data command between two write-address command.
3. The EPP process:
   to enter EPP mode:
      Negotiate-to-EPP
   to write register:
      Write-Address, with PD[6:0] = register address[6:0].
      Write-Data, PD[7:0] transfer the 8bits register data.
      * With register address = 0x01, host can write the fifo data.

to read register:
Write-Address, with PD[6:0] = register address[6:0].
Read-Data, PD[7:0] transfer the 8bits register data. .
to read fifo: N= number of byte to be read
Write-Address, with PD[6:0] = 0x00.
Read-Data, repeat N times.
* N= number of byte to be read

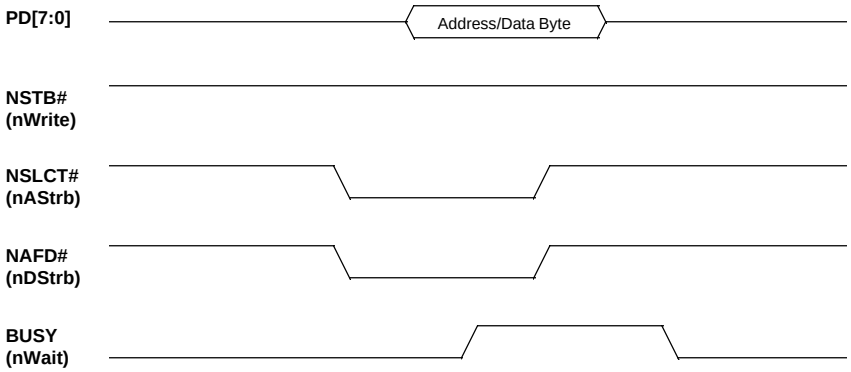


Figure 7-9: EPP Data or Address Read Timing

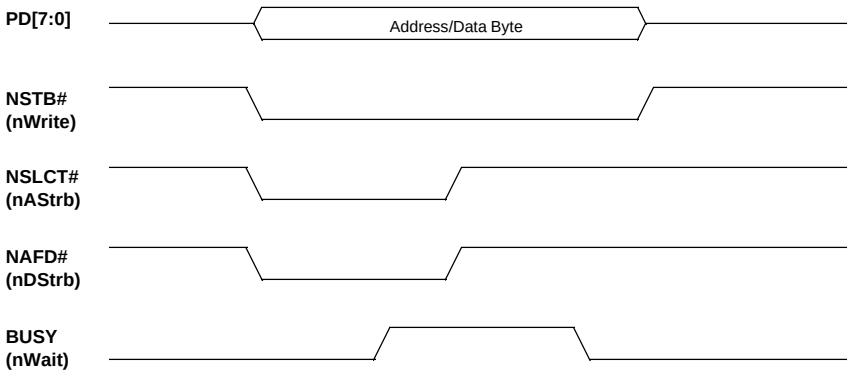


Figure 7-10: EPP Data or Address Write Timing

7.1.2.6 Extended Capabilities Port (ECP)

1. Host should go through the negotiation process to enter the ECP mode.
2. In ECP mode, host can access internal registers through the command/data transfer. With command transfer host can issue the register address A[6:0]. With data transfer host can read/write the 8-bits data.

3. The command and data transfer in ECP mode is differentiated by ″nAutoFd″. ″nAutoFd″ low is command transfer, whereas ″nAutoFd ″ high is data transfer.
4. After one byte of command transfer, host can deliver several bytes of data. The issued register address will keep effective till next command transfer happens. This might be used when writing fifo-in register.

5. Here list the process,

to enter ECP mode:
Negotiate-to-ECP
SetUp

to write register:
Forward-command, with PD[6:0] = register address[6:0].
Forward-data, PD[7:0] transfer the 8bits register data.
* With register address = 0x01, host can write the fifo data.

to read register:
Forward-command, with PD[6:0] = register address[6:0].
Forward-to-Reverse
Reverse, PD[7:0] transfer the 8bits register data. .

to read video fifo:
Forward-command, with PD[6:0] = 0x00 (address of ″Video Data Out Register″).
Forward-to-Reverse
Reverse, repeat N times.
(N is video threshold level.)

to read audio fifo:
Forward-command, with PD[6:0] = 0x67 (address of ″Audio Data Out Register″).
Forward-to-Reverse
Reverse, repeat N times.
(N is audio threshold level.)

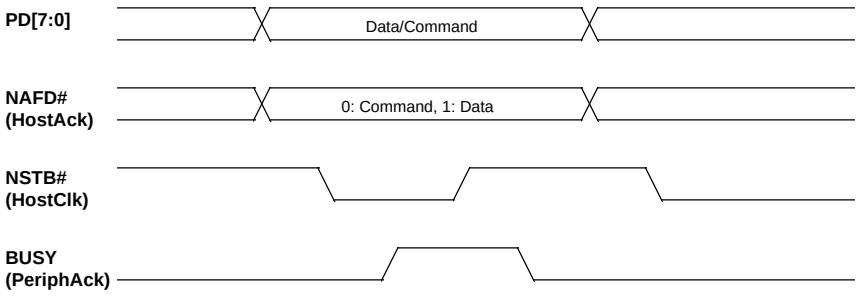


Figure 7-11: ECP Forward Timing

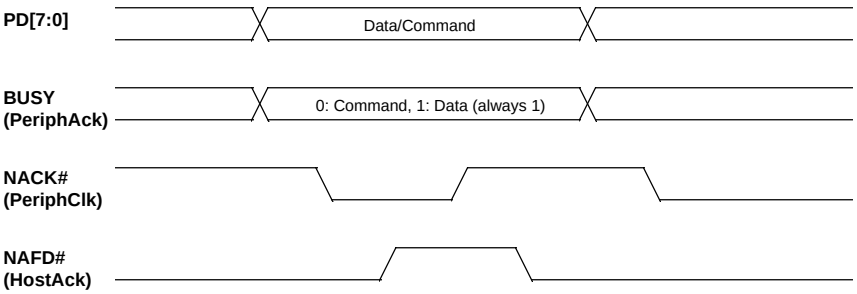


Figure 7-12: ECP Reverse Timing

7.1.3 Generic bus interface:

The generic bus is a 32 bits data bus that can operate in 32-bits, 16-bits or 8-bits wide modes. Bus width is selected by the BW[1:0] signals:

1. BW[1:0] = 00. 8 bits wide
2. BW[1:0] = 01. 16 bits wide
3. BW[1:0] = 10. Not allowed
4. BW[1:0] = 11. 32 bits wide

The generic bus includes 7 bits address bus, A[6:0], supporting an address space of 128 registers. These registers can be divided into three groups:

1. Command, configuration and status registers. These registers are 8 bits wide and are read and written through the 8 l.s.b of the data bus. Read and Write transactions to these registers are synchronized to the 54MHz clock.
2. Vdata_in register. This write only register is used to write pixels into the W99200F in forward data transactions. The width of this register changes according to the BW[1:0] signals. Write transactions to this register are synchronized to the 54MHz clock.
3. Vdata_out register. This read only register is used to read data in reverse data transfer transactions. This register is the out port of the on chip FIFO. The width of this register changes according to the BW[1:0] signals. Reading transactions from this register are synchronized to a special clock (FIFO_RD_CLK). This enables the host bridge to work as a bus master and to read the data out from the W99200F in the maximum rate of the external (e.g., PCI) bus. When the bus master mode is enabled (MAST_EN) the W99200F indicates the host bridge that data is ready in the FIFO (FIFO_RDY), the host bridge then requests the PCI bus and when it gets it, it reads the data from the FIFO by using the FIFO_RD# signal.

Registers	Clock signal	Write signal	Read signal	Register width
Vdata_out	FIFO_RD_CLK	—	FIFO_RD#	according to BW[1:0]
Vdata_in	CLK	WR#	—	according to BW[1:0]
Configuration	CLK	WR#	RD#	8 or 32 bits
Command	CLK	WR#	—	8 bits
Status	CLK	—	RD#	8 bits

Table 7-6: Generic Bus Read and Write Signals from Internal Registers

FIFO_RDY signal:

When the host bridge is a master bridge (MAST_EN signal is on) the W99200F notifies the bridge whenever threshold amount of data is ready in the FIFO by activating the FIFO_RDY signal. The bridge can then request the bus and transfer the data from the W99200F FIFO towards the host memory. After the first data is being read by the HOST, FIFO_RDY is deactivated until the HOST finishes reading a threshold amount of data. In order to work in full rate, the FIFO_RD_CLK can be the host bus (e.g., PCI) clock. In this case there is no need for the FIFO ready interrupt and it can be disabled.

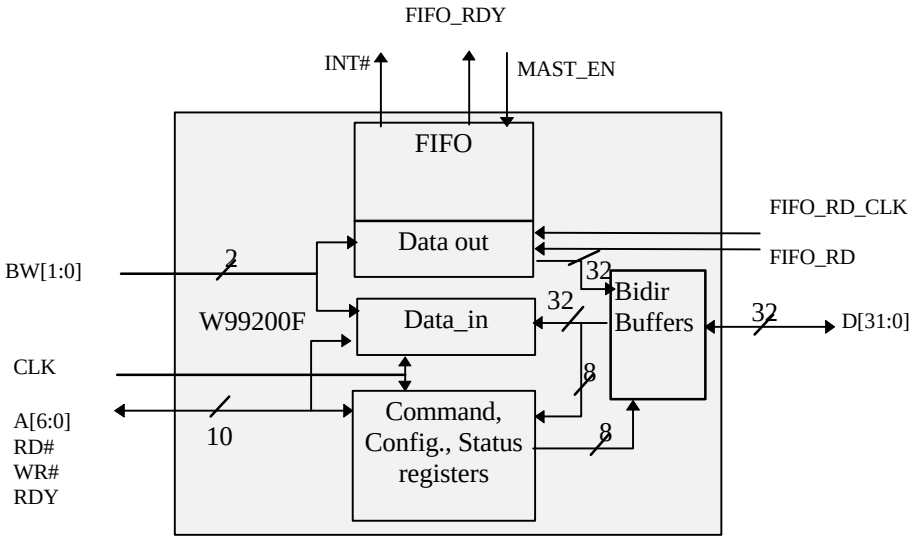


Figure 7-13: Generic bus signals

Figure 7-14 to Figure 7-19 show the timing diagrams on the host bus.
(H) : means this signal is driven by the host .
(C) : means this signal is driven by the encoder.

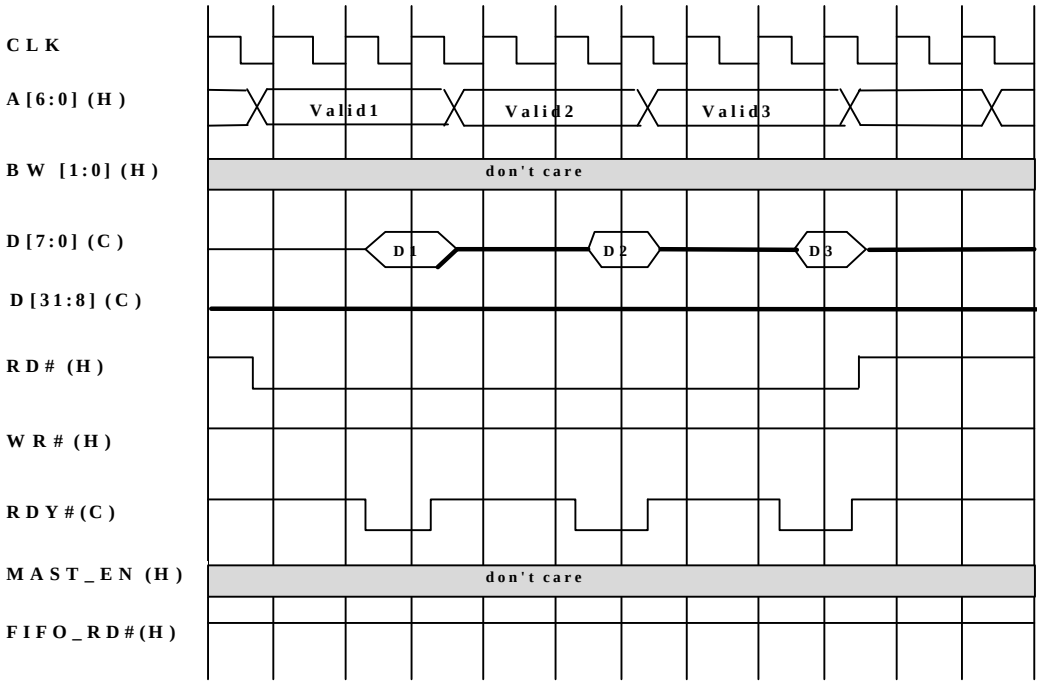


Figure 7-14: Read cycles from configuration and status registers

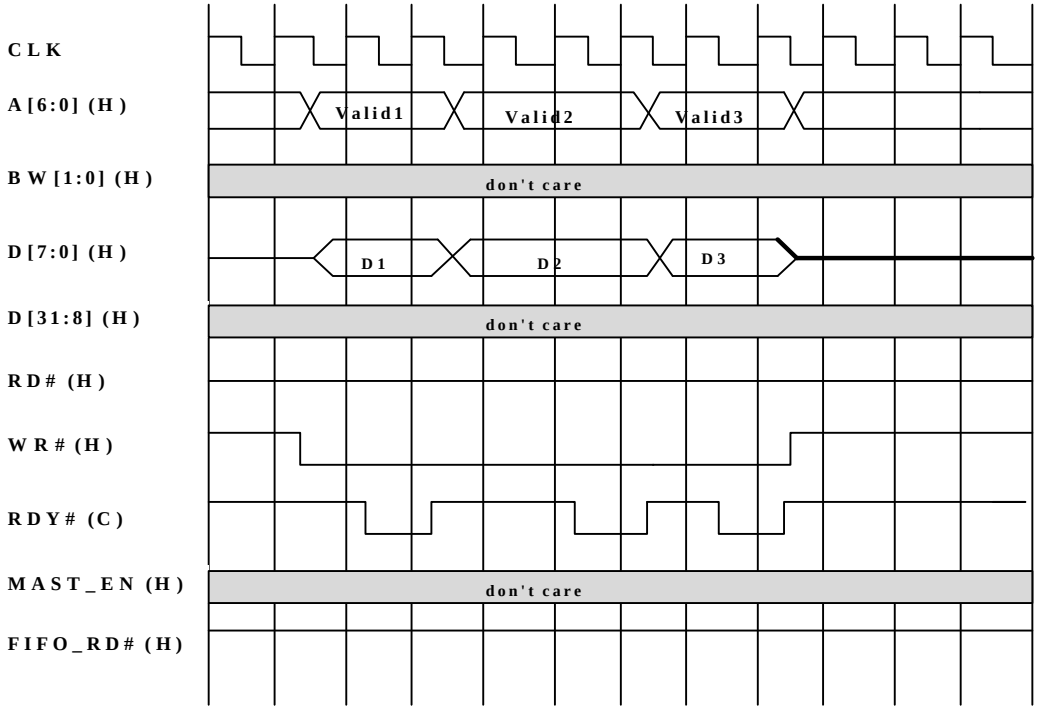


Figure 7-15: Write cycles to command and configuration registers

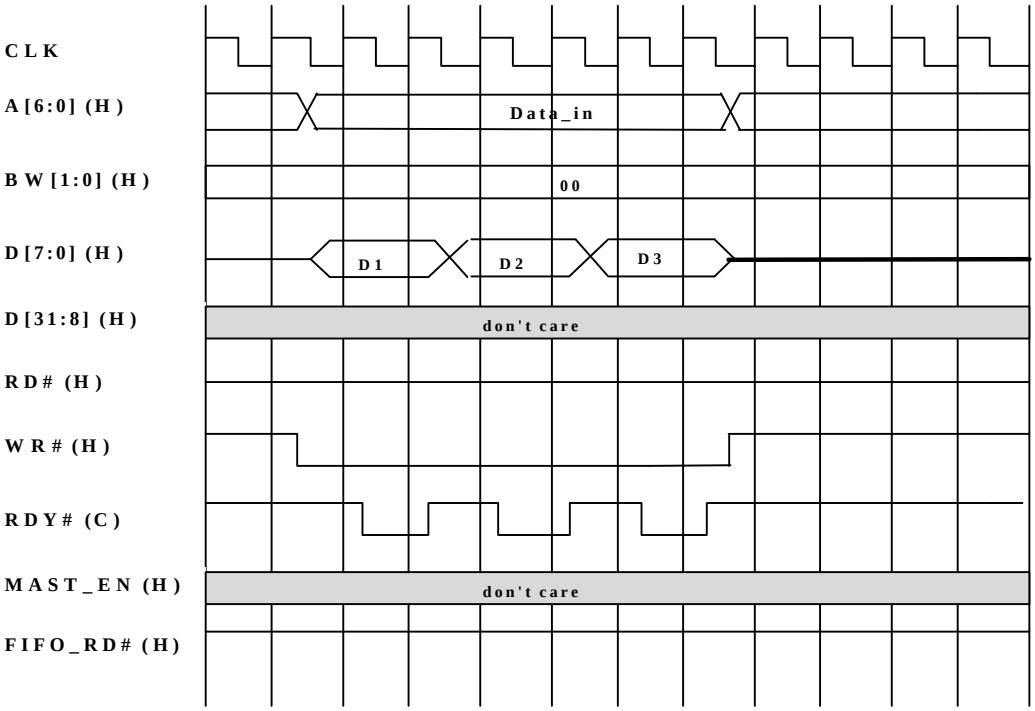


Figure 7-16: Write cycles to Vdata_in register, BW[1:0]=00

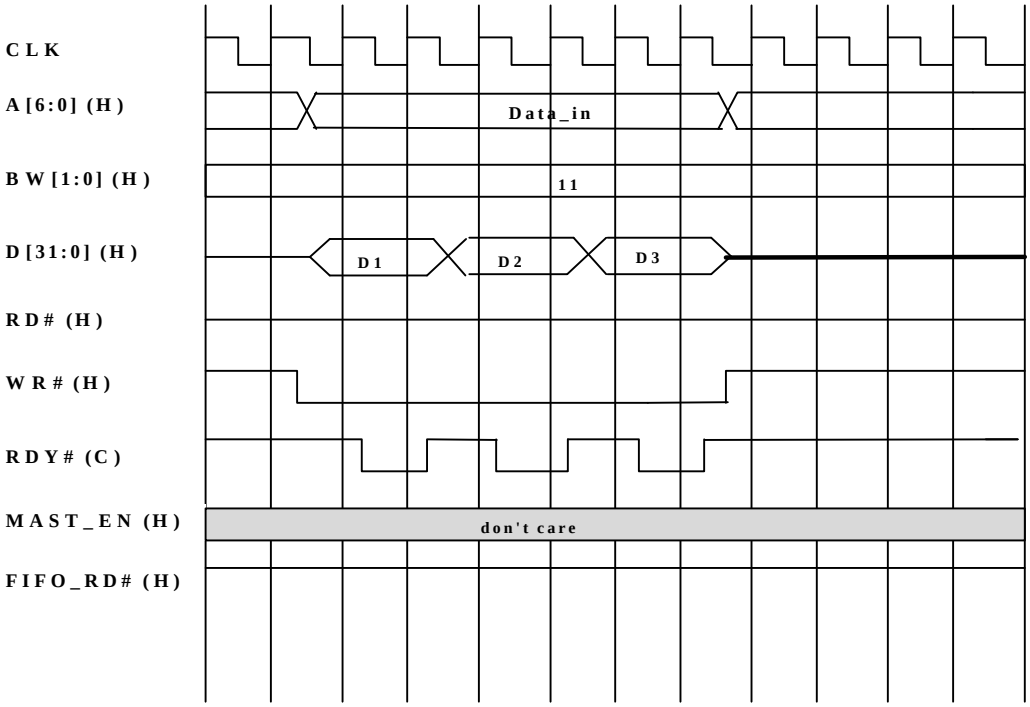


Figure 7-17: Write cycles to Vdata_in register, BW[1:0]=11

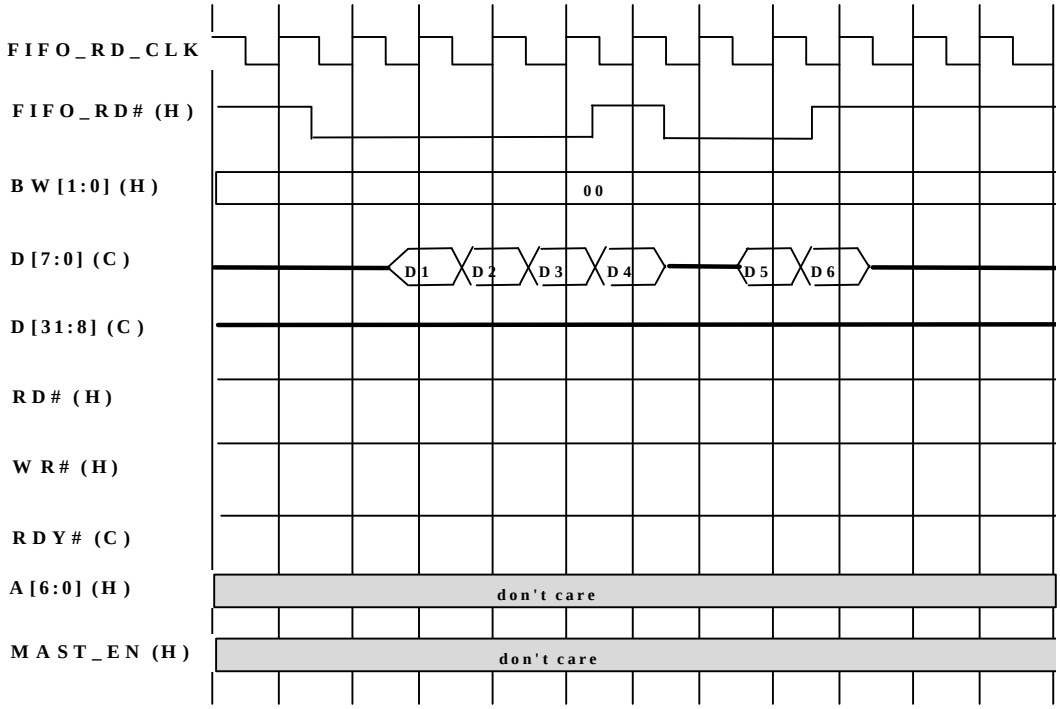


Figure 7-18: Read cycles from Vdata_out register, BW[1:0]=00

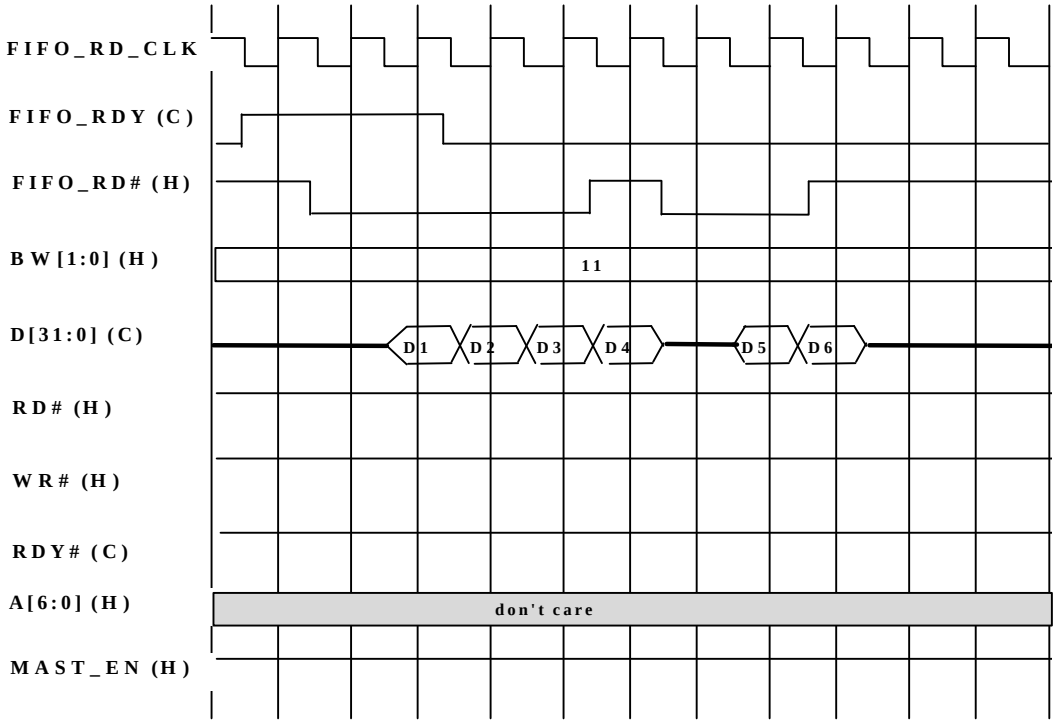


Figure 7-19: FIFO_RDY and Read cycles from Vdata_out register, BW[1:0]=11

7.2 VCD Decoder Interface

W99200F supports the VCD decoder interface (8051-like) to provide the function that VCD bitstreams can be sent from PCI or Parallel port host and pass through W99200F to VCD decoder.

For example W9926QF SVCD decoder:

1. Glueless interface between W99200F and VCD decoder.
2. W9926QF needs 10 address ports, so W99200F will reserve address 0x70 ~ 0x78 and 0x7C for W9926QF.
3. W9926QF build an intelligent I/O address setup mechanism. The host can access W9926QF in the next procedure:
   - writes "1" into bit 7 (SDRAM_TRI) of PCR register to tri-state the SDRAM interface of W99200F.
   - writes "0" into bit 5 (DE_TRI0) of PCR register to de-assert DEC_WR# and DEC_RD#.
   - writes "0" into bit 6 (DE_TRI1) of PCR register to de-assert DEC_RESET#.
   - writes the ASCII code "!" into index 0x70 register of W99200F.
   - repeat writing index 0x70 register of W99200F the ASCII code "N", "A", "5", "1", "7", "9", "!" four times.
   - the port address, 0x70, will be latched into AIR0 of W9926QF.
   - The index 0x70~0x78 and 0x7C registers of W99200F will be mapped to the port AIR0, AIR1, ... IADPR0 of W9926QF. (Refer to the data sheet of W9926QF.)
4. W99200F and VCD decoder share the single 16M bits SDRAM.
5. In W9926QF SVCD decoder, SDRAM interface signals can be tri-state by asserting RESET#, WR# and RD# active.
6. W99200F do "AND" function between VCD decoder's INT# and other interrupt requests, then send it to PC.

7.3 Audio Data Input Interface (AIN)

Audio data input interface is responsible to get audio codec data and sent them to the host. Then the host can compress audio bitstream and mix the video compressed and audio compressed bitstream. The W99200F provides two modes of audio data input interface:

7.3.1 Without external audio FIFO

W99200F supports 3-pins I2S interface.

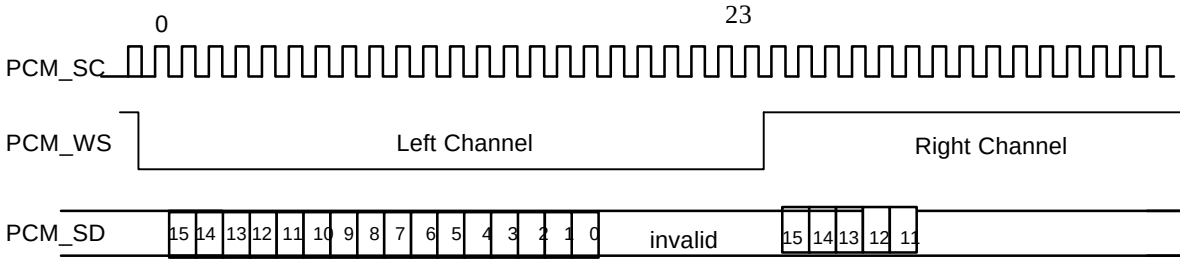


Figure 7-20: I2S Format

Figure 7-20 shows an I^2S format, which is characterized as PCM_SC, PCM_SD MSB first and every word MSB appeared one clock period after the PCM_WS changes, and data latch at the rising edge of PCM_SC. The waveform of PCM_SC and

PCM_WS are generated by ACLK input and can be programmably set by the host. (Refer to "Audio Clock Divider Register".)

There is a 128x32 FIFO in W99200F chip.

7.3.2 With external audio FIFO

Figure 7-21 shows the waveform of external parallel FIFO data interface.

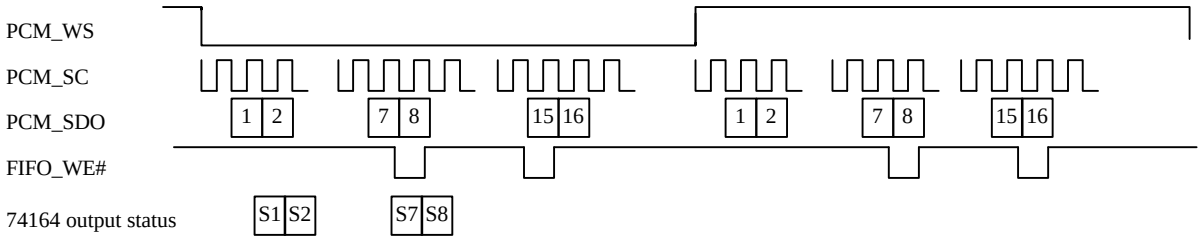


Figure 7-21: Parallel FIFO Data Interface

W99200F can supports an external audio FIFO. The size of FIFO can be 4KB, 16KB, 64KB and 256KB. It can be set by the host at the initialization.

7.4 Video Decoder Interface

The W99200F incoming video timing is derived from ITU-R BT.656. Video data format is 4:2:2, and is input as 8 bit pixel data order Cb,Y,Cr,Y. The supported resolutions are shown in Table 7-7. Unlike ITU-R-656, where the video timing signals are coded within the video stream, the W99200F gets those timing signals as separate pins, see Table 7-8.

Data rate is either 13.5MHz for ITU-R BT.601 resolutions, or 12.27MHz for Square-pixel, NTSC resolution. However, the allowed frequency of the clock accompanies the data is 24.54MHz (For Square pixel) to 36MHz. Therefore, only when a data qualifier signal (VIDV) is active, a pixel data is sampled by the W99200F.
The above is an example of the W99200F's flexibility, which enables a glueless connection to many available video decoders.

Input Resolution	Frame Rate [Hz]
720 x 480 (ITU-R BT.601, NTSC)	29.97
720 x 576 (ITU-R BT.601, PAL)	25
640 x 480 (Square Pixel, NTSC)	29.97

Table 7-7: Supported input video resolutions

Pin name	Pin Type	Pin Description

VICLK	I	Video Input Clock. Free running clock. Allowed frequency: 24.54MHz - 36MHz
VID[7:0]	I	Video Input Data.
VIHS#	I	Video Input Horizontal Sync signal. Falling edge (Default) indicates a new line. Signal polarity is programmable.
VIVS#	I	Video Input Vertical Sync signal. Falling edge (default) indicates a new field. Signal polarity and line offset are programmable. Pin can also be configured as Field ID(FID) input. Falling edge (default) indicates an odd field.
VIHACT	I	Video Input Horizontal Active pixels identifier. When high (default) indicates the active period of a video line. During VBI there may be an activation of this signal. However, No pixel data is sampled during a VBI window. When active during an active video line, pixel data is sampled with every rising edge of VICLK, where VIDV is active. Polarity is programmable.
VIDV	I	Video Input Data Valid. Pixel data is sampled only if this signal is high (default). Polarity is programmable.

Table 7-8: Video Input interface - Pin Description

7.4.1 Video timing signals

The W99200F uses VIHS#, VIVS#, VIHACT and VIDV to synchronize the video data. VIHS# is used to increment the line counter. VIHACT and VIDV are used to sample internally the active video data of the active video lines (Active video lines for different resolutions are shown in Figure 7-22, Figure 7-23, and Figure 7-24). VIVS# is used to identify the input field and to reset the line counter.
Figure 7-22, Figure 7-23, and Figure 7-24 present the video timing signals for each of the supported video input formats. In general, the number of lines in the video input active region may be larger than required by the W99200F. However, that video input active region must envelope the W99200F's video active region. For example, in ITU-R BT.601, NTSC format, there are 486 active video lines, while the W99200F requires only 480 lines, see Figure 7-22.
In ITU-R BT.601 resolutions, there are 720 active pixels in a video line. The 352 active pixels in SIF resolution are 2:1 decimated from the central 704 active pixels in the original line. The additional 8 pixels on each side are cropped.
In Square pixel resolution, the original 640 pixel/line are decimated to 320. In that case, the pixels at the edges are multiplied to create an artificial line with more pixels, which is used by the horizontal decimation filter. The same applies for vertical decimation processes of all input resolutions.

7.4.1.1 Vstart/stop encoding

When W99200F receives a 'start encoding' command it waits to the first odd field indicated by a combination of VIVS# and VIHS#. That field is the first field to be encoded.
W99200F indicates out-of-sync in either one of the following two cases:

1. Wrong line length - Number of active pixels between two successive VIHS# is not equal to N, where N = 720 -for PAL/NTSC, 640 - for Square pixel.
2. Wrong field indication - VIVS# indicates a top field, when bottom field is expected, or indicates bottom field, when top field is expected.
3. Early VIVS# - VIVS# is detected too early, i.e. not in the exact line it was supposed to be detected.

4. Late VIVS# - VIVS# is not detected within the expected line.
5. Fast Video source -The incoming video rate is too high for the encoding machine . If the analogue video source is more than 0.3% faster than nominal video rate (VICLK frequency relative to MCLKI frequency is >3000ppm faster).

In each of the above cases, W99200F ignores the frame during which the out-of-sync was received or detected, and encodes and outputs all the previous frames. In LVE and LVPT modes the W99200F tries to get synchronized again by waiting for the next odd field indicated by a combination of VIVS# and VIHS#. In LVSS mode the W99200F stops when out-of-sync is received.

7.4.2 Programmable registers

In order to enable glueless connection to most of video decoders on the market, some default parameters which are not common in all video decoders may be adjusted, by programming them via the registers shown in Table 7-9.

Description	Register name	Bit #	Available values
VIVS# Odd Field indication Offset	Vin_offset	3:0	Signed integer (-8..7) indicates the offset (in lines) of odd field indication, from the required line location. In PAL, the minimum value is: -6. Default: 0x0.
VIVS# Even Field indication Offset	Vin_offset	7:4	Signed integer (-8..0) for NTSC and (-7..0) for PAL indicates the offset (in lines) from the required line location of even field indication. Default: 0x0.
VIHS# Polarity	Vin_cntl	0	0 (default): Active low. Falling edge is used to indicate a new video line. 1: Active high. Rising edge is used to indicate a new video line.
VIVS# Polarity	Vin_cntl	1	0: If VIVS# is VS (vin_cntl[6]==0) - Falling edge marks the beginning of a new field. else (VIVS# if FID) (vin_cntl[6]==1) - Falling edge marks the beginning of an odd field. 1: If VIVS# is VS (vin_cntl[6]==0) - Rising edge is used to indicate a new video field else (VIVS# if FID) (vin_cntl[6]==1) - Rising edge marks the beginning of an odd field.
VIHACT Polarity	Vin_cntl	2	0: Active low. When low - indicates the active region in a line. 1 (default): Active high. When high - indicates the active region in a line.
VIDV Polarity	Vin_cntl	3	0: Active low. When low - indicates a valid pixel to be sampled. 1 (default): Active high. When high - indicates a valid pixel to be sampled.
VIHS# sampled level	Vin_cntl	4	0 (default): Odd field is indicated by VIHS# high sampled with VIVS#, Even field is indicated by VIHS# low sampled with VIVS#. 1: Odd field is indicated by VIHS# low sampled with VIVS#, Even field is indicated by VIHS# high sampled with VIVS#.
VIVS# input select	Vin_cntl	6	0: (default) VIVS# input is vertical sync signal 1: VIVS# input is field ID signal (FID)
Blanked Video	Vin_picture	0	0 (default): No Video lines are blanked.

Lines			1: Last 16 rows in FULL resolution, last 8 rows in SIF resolution or last 4 rows in QSIF resolution are blanked.

Table 7-9: Video input programmable parameters

Note: All video timing signals are shown in the following figures with their default polarity.

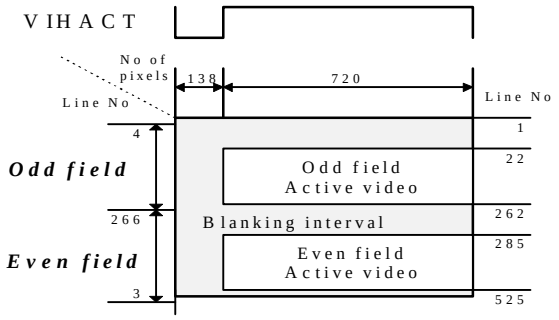


Figure 7-22: Active video for ITU-R BT.601, NTSC

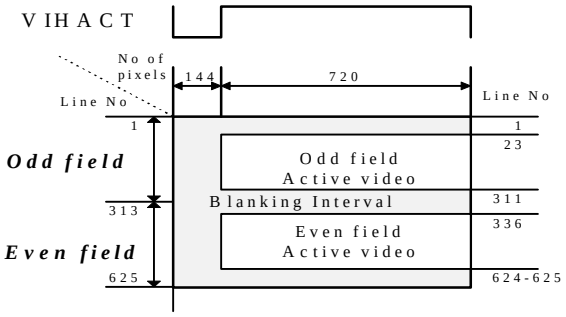


Figure 7-23: Active video for ITU-R BT.601, PAL

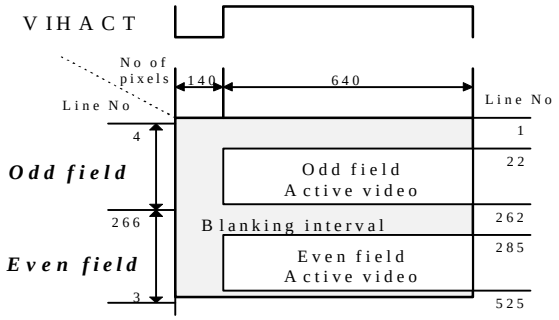
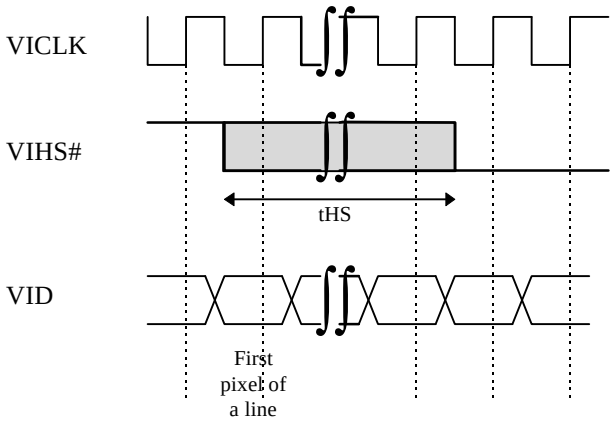


Figure 7-24: Active video for SQUARE, NTSC



- 0 ≤ tHS ≤ 32 [VICLK cycles]

Figure 7-25: VIHS# falling edge timing

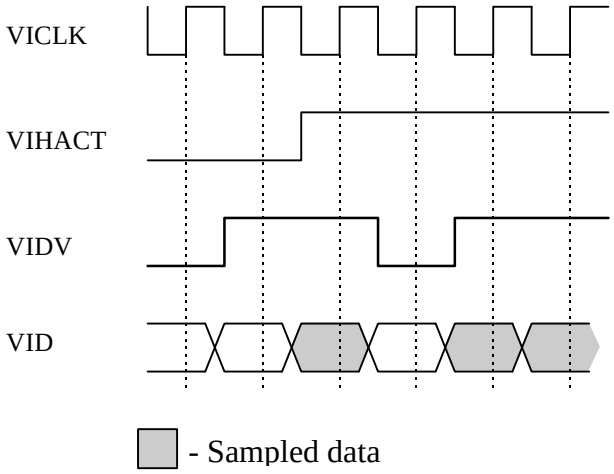
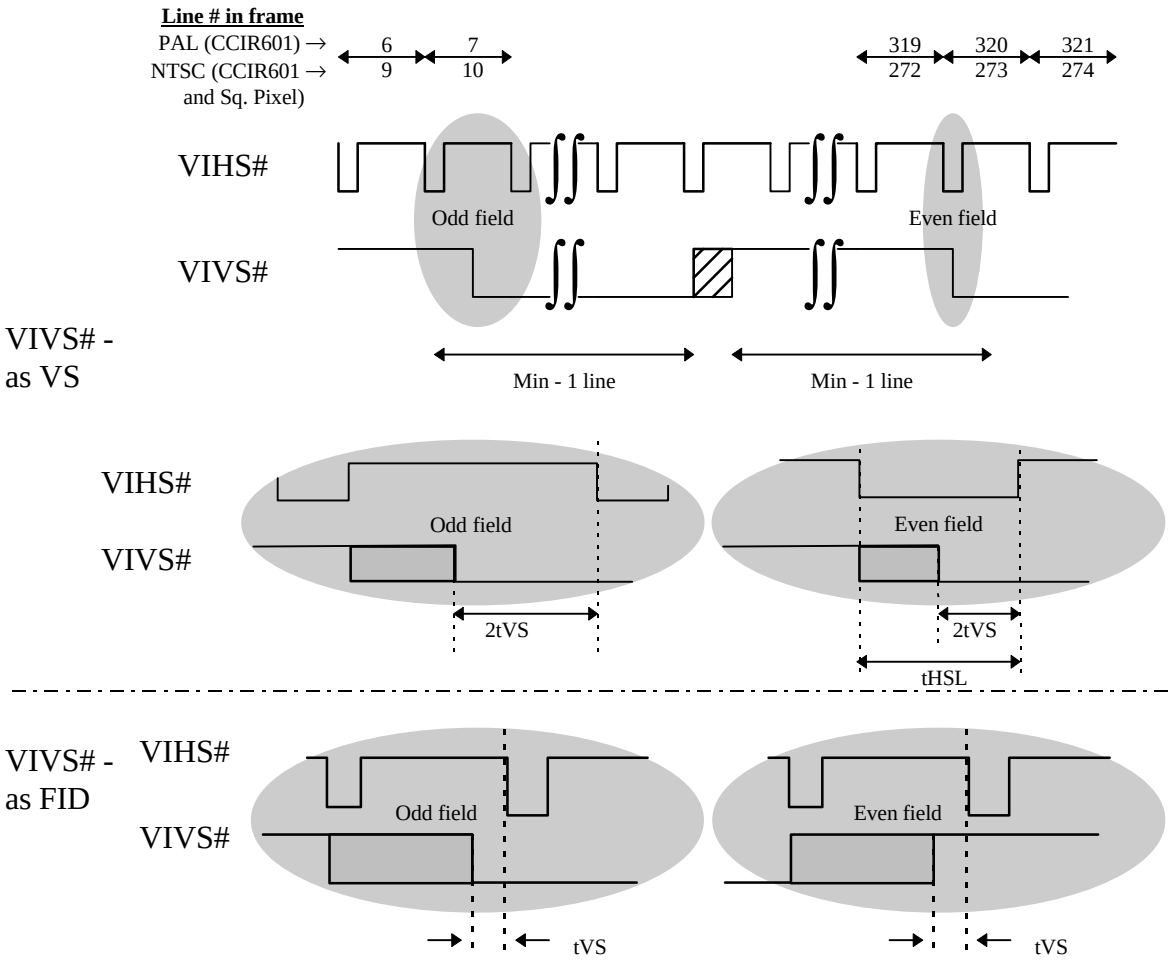


Figure 7-26: VID sampling with VIDV



2 ≤ tVS [VICLK cycles] 4 ≤ tHSL [VICLK cycles] ≤ 512

VIVS# falling edge offset from the required line location is programmable (Register 0x18). For example: PAL Odd field VIVS# falling edge, occurs in line no. 10, should be offset programmed of +3.

Figure 7-27: video timing signals

7.5 I²C interface

Most video decoders can be programmed using I²C bus. The W99200F can function as an I²C bus master and by that enables the host to program the video decoders. The W99200F is responsible to pass through the I²C stream forward, to the component, which forms the slave on the I²C bus, and backward, to the host, which forms the master on the I²C bus. All transactions on the I²C bus are initiated by the host, the W99200F is only used as a bridge.

7.5.1 I²C protocol overview

The bus physically consists of 2 active wires and a ground connection. The active wires, SDA and SCL, where SDA is the Serial Data line and SCL is the Serial Clock line. In general, I²C is a multi-master bus, where the master for a particular dialog is the one who initiated that dialog. However, in our case, the W99200F is always the only master, while the video decoder is the slave.

When the master wants to write to its slave, it will issue a START condition. Then it sends the device address it wants to access, together with a Write indication. This takes 8 clock pulses. If the address matches the slave address, it will produce a response called ACKNOWLEDGE condition. After that acknowledge is received by the master, the data can be written to the slave. After each 8 cycles of data, an acknowledge condition is produced by the slave. When the master completes the write sequence, it issues a STOP condition.
When the master wants to read from the slave, it will issue a START condition. Then it sends the device address it wants to access, together with a Read indication. This takes 8 clock pulses. After acknowledge being received by the master, the data can be read from the slave. After each 8 cycles of data, an acknowledge condition may be produced by the master. If such an acknowledge is issued the slave will send another 8 bits of data, otherwise the sequence will be terminated.

7.5.2 W99200F I²C mechanism

The W99200F is used as a master on the I²C bus on which a video decoder and maybe other devices are slaves. Only the standard mode (100KHz) is supported. The W99200F actually performs a parallel to serial (Host to I²C) and serial to parallel (I²C to Host) conversion. All parallel data received from the host is stored in W99200F registers before being transferred on the I²C bus.
The detailed protocol between the W99200F and the host with regard to the I²C bus is described in 7.5.4.

7.5.3 I²C control and status registers

Table 7-10: I²C commands

Description	Register name	Available values	
I²C Data	I2c_data	Data to/from I²C bus written to this register	
I²C START command	I2c_start	Writing to this address initiates an I²C START condition on the I²C bus	
I²C STOP command	I2c_stop	Writing to this address initiates an I²C STOP condition on the I²C bus	
I²C Read Acknowledge	I2c_rack	Writing to this address initiates an ACKNOWLEDGE condition on the I²C bus	
I²C Ready (I2C_RDY)	Vint_source	Bit 6	I2c interrupt status register. Should be read whenever an active INT# is accompanied with high I2C_RDY bit in the ***Vint_source*** register
I²C success (I2C_SUCC)	I2c_status	Bit 0	When high, indicates that either the last data transfer to the slave has been acknowledged, or the last data transfer from the slave has been completed. The host is then expected to activate either one of the following actions: 1. Activate I2C_START bit. (In read/write process) 2. Activate I2C_STOP bit. (in read/write process) 3. Write data to I2C_DATA register. (In write process) 4. Read data from I2C_DATA register. (In read process) When low, indicates that the W99200F has detected a NOT acknowledge condition issued by the slave receiver (In write process). The host is then expected to activate the I2C_STOP bit, to terminate the transaction. It then may repeat the last transaction.

			This register is self cleared when read by the host.

7.5.4 I²C protocol via the host interface

Write sequence:

After Video_reset : host waits 200 micro-seconds before an I²C transaction, while W99200F issues a STOP condition on the I²C bus.

In configuration time : Host enables the I2C_RDY bit in the ***Vint_enable*** register.

1. Host reset I2C by writing I2C_RDY bit to ***Vint_clear*** .
2. Host writes to ***I2c_start***.
3. Host writes to ***I2c_data*** register the device address to be connected to. LSB is written with '0' to indicate a WRITE sequence.
4. W99200F issues a START condition on the I²C bus.
5. W99200F writes the content of ***I2c_data*** register on the I²C bus in 8 consecutive SCL cycles.
6. IF NOT (W99200F receives acknowledge on the I²C bus)
   - W99200F issues a STOP condition on the I²C bus.
   - W99200F activates INT#, accompanied with activating I2C_RDY bit in the ***Vint_source*** register. I2C_SUCC bit in ***I2c_status*** register is low.
   - Host reads '0' in I2C_SUCC bit.
   - Host writes '1' to I2C_RDY bit in the ***Vint_clear*** register, in order to clear the active interrupt
   - END.

   ELSE
7. W99200F activates INT#, accompanied with writing '1' to I2C_RDY bit in the ***Vint_source*** register, and to I2C_SUCC bit in ***i2c_status*** register
8. Host reads '1' in I2C_SUCC bit.
9. Host writes '1' to I2C_RDY bit in the ***Vint_clear*** register, in order to clear the active interrupt.
10. Host writes a data byte to ***I2c_data*** register.
11. W99200F writes the content of ***I2c_data*** register on the I²C bus in 8 consecutive SCL cycles.
12. IF NOT (W99200F receives acknowledge on the I²C bus)
    - W99200F issues a STOP condition on the I²C bus.
    - W99200F activates INT#, accompanied with activating I2C_RDY bit in the ***Vint_source*** register. I2C_SUCC bit is not set .
    - Host reads '0' in I2C_SUCC bit.
    - Host writes '1' to I2C_RDY bit in the ***Vint_clear*** register, in order to clear the active interrupt .

- END.

ELSE

13. W99200F activates INT#, accompanied with writing '1' to I2C_RDY bit in the ***Vint_source*** register, and to I2C_SUCC bit in ***i2c_status*** register
14. Host reads '1' in I2C_SUCC bit.
15. Host writes '1' to I2C_RDY bit in the ***Vint_clear*** register, in order to clear the active interrupt.
16. Host can:
    - Perform write of another byte → Go to step 10 of the write sequence.
    - Perform repeated start following by a read sequence → Go to step 2 of read sequence

    ELSE
17. host writes to ***I2c_stop***
18. W99200F issues a STOP condition on the I^2C bus.
19. W99200F activates INT#, accompanied with writing '1' to I2C_RDY bit in the ***Vint_source*** register,.
20. Host writes '1' to I2C_RDY bit in the ***Vint_clear*** register, in order to clear the active interrupt .
21. End

Read sequence:

After Video_reset : host waits 200 micro-seconds before an I^2C transaction, while W99200F issues a STOP condition on the I^2C bus.

In configuration time : Host enables the I2C_RDY bit in the ***Vint_enable*** register.

1. Host reset I2C by writing I2C_RDY bit to ***Vint_clear*** .
2. Host writes to ***I2c_start***.
3. Host writes to ***I2c_data*** register the device address to be connected to. LSB is written with '1' to indicate a READ sequence.
4. W99200F issues a START condition on the I^2C bus.
5. W99200F writes the content of ***I2c_data*** register on the I^2C bus in 8 consecutive SCL cycles.
6. IF NOT (W99200F receives acknowledge on the I^2C bus)
    - W99200F issues a STOP condition on the I^2C bus.
    - W99200F activates INT#, accompanied with activating I2C_RDY bit in the ***Vint_source*** register. I2C_SUCC bit is not activated.
    - Host reads '0' in I2C_SUCC bit.
    - Host writes '1' to I2C_RDY bit in the ***Vint_clear*** register, in order to clear the active interrupt.
    - END

ELSE

7. W99200F receives 8 cycles of data on the I^2C bus.
8. W99200F writes the received data to ***I2c_data*** register.
9. W99200F activates INT#, accompanied with writing '1' to I2C_RDY bit in the ***Vint_source*** register, and to I2C_SUCC bit in ***i2c_status*** register
10. Host reads '1' in I2C_SUCC bit.
11. Host writes '1' to I2C_RDY bit in the ***Vint_clear*** register, in order to clear the active interrupt.
12. Host reads the data byte from ***I2c_data*** register.
13. IF NOT (host writes to ***I2c_stop***)
    - Host writes to ***I2c_rack*** register.
    - W99200F issues an ACKNOWLEDGE signal on the I^2C bus.
    - Go to step 7.

    ELSE
14. W99200F issues a STOP condition on the I^2C bus.
15. W99200F activates INT#, accompanied with writing '1' to I2C_RDY bit in the ***Vint_source*** register.
16. Host writes '1' to I2C_RDY bit in the ***Vint_clear*** register, in order to clear the active interrupt.
17. End

7.6 Audio & Video synchronization

W99200F support 3 different means for getting perfect A/V sync.
The A/V sync problems comes from :

1. A/V start point :
   One time sync resolution, and thus not a real sync issue. Resolved by a proper start to Audio/Video at approximately the same time. The video however, can still be out of sync (see item 3).
2. Clock drift :
   The video source, The video encoder (W99200F) and the Audio codec have three different clocks. Each one of these clocks can have 100PPM tolerance. Together, we have 200PPM possible distance between the clocks that is accumulated to .7 sec distance of A/V over an hour of capture.
3. Video out of sync:
   The audio stream is continuous and uninterrupted. The video stream is sync controlled and hence can get out of sync. One example is the point in time where the actual Vsync arrives (e.g.; when operating the VCR only AFTER start of capture) . Another example are possible Vsync jumps (due to TV zapping or bad video source) . In the out of sync event, the captured video is stopped until the next valid sync, the audio captured at this time needs to be dropped.
4. Telecine frame-rate changes
   When using the telecine option in NTSC, W99200F produces a stream combined of several MPEG1 video sequences with changing frame-rates (29.97->23.976->29.97). This process is introducing time-base changes (Increase or decrease, depending on the actual place in the sequence that the sequence change took place). These time-base changes are Video only and they need to be compensated by an appropriate audio drop/duplicate.

7.6.1 Clock drift sync

In order to fully resolve this A/V sync source, genlocking of the Video/Audio and encoding is a must. There are two possible system configurations :

- Genlocking audio codec to Video source

Based on SAA7112 digital video decoder + Any audio codec. SAA7112 has an audio clock output AMCLK that is fully synchronized to ICLK (27MHZ Video HSYNC genlocked clock). In this case The Video clock is genlocked to the Video source Syncs (including its potential drift and jitter) and so is the Audio clock. The W99200F SCR (System Clock Reference) for generating PTS (Presentation Time Stamp) must be taken from the SAA7112 ICLK , this can be done by programming the Vin_cntl register bits 5-6. The SAA7112 can drive the main W99200F MCLKI clock with the LLC 27MHZ pin.

- Genlocking audio codec to W99200F MCLKI

In the case of all typical video decoders that do not have the audio clock output (e.g.; the Brooktree video decoder) the synchronization can be achieved by genlocking the Audio codec clock and the W99200F main clock (MCLKI 27MHz). The genlocking can be done by using a Microclock device. The W99200F SCR (System Clock Reference) for generating PTS (Presentation Time Stamp) must be taken from the MCLKI (W99200F main 27MHZ clock), this can be done by programming the Vin_cntl register bits 5-6. The genlocking of the clocks must be accomplished by SW PTS support in the system MUX that will compensate for audio/video drifts (see the PTS MUX in next section).

Note : In the case were the audio capture is done on another board and hence clocks genlocking is not possible. A/V sync drift problem will always occur.

7.6.2 Video out of sync

The video-source is expected to be stable and report a VSYNC (Vertical-Synchronization signal) in a constant time-periods. However, some events may cause the VSYNC to flicker for a while - for example, zapping between TV channel or bad VHS cassette. In such cases the period between successive video frames may differ from the expected period as implied by the frame-rate.

There are two W99200F solutions for video out of sync :

- System MUX PTS based fix

The PTS (Presentation Time Stamp) is sampled according to MCLK/VICLK (see previous section on clock drift). There is a known PTS period for a frame (e.g.,3600 for PAL). The System-Mux is required to identify the cases of Vsync not in place, and take actions such as dropping or duplicating some audio data to keep the audio synchronized to the video stream.

- OSYNC# based fix

The output pin OSYNC# is high ONLY WHEN A VALID VIDEO IS SAMPLED. This makes it very easy to use it as an enable signal to the audio capture (e.g.; by controlling the Audio FIFO write enable). This solution is suitable in cases that the system MUX cannot be modified to support PTS.

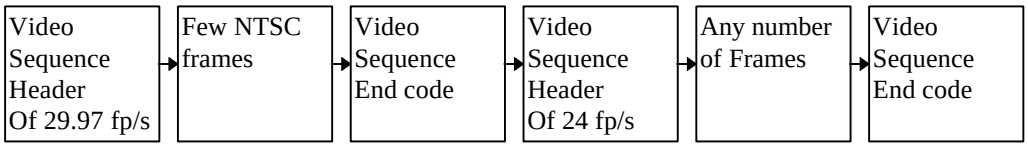
Note : In the case were the audio capture is done on another board, OSYNC# based fix is not possible and System MUX PTS must be used.

7.6.3 Telecine frame-rate fix

This problem occurs only in NTSC while using the Telecine option on. The only solution in this case is to use a system mux PTS based fix like in the out of sync case. If the system MUX cannot be modified to support PTS, the Telecine option must be configured OFF , otherwise, A/V sync problems will happen.

The W99200F supports a unique feature of Telecine-detection. "Telecine" is the process by which a cinema-film source of 24 frames-per-second is translated to NTSC 30 frames-per-second. Naturally, the "Telecine" process involves duplication of

video-data. The telecine-detection feature detects the original 24 fp/s pattern within a NTSC source, and allows the MPEG encoder to encode only those original frames, dropping the duplicated frames.
When the telecine-detector is turned ON (allowed only for NTSC) and a telecine pattern is detected, the MPEG video stream produced is a concatenation of MPEG video-sequences. Specifically, the stream may look like -



Moreover, if the source input stops following the telecine pattern (for example, in the case of TV commercials embedded with in the Film source) - The telecine detector stops the 24 fp/s video-sequence and starts a new NTSC video-sequence. The System-Mux is required to follow the changes of frame-rate by changing the PTS/DTS and dropping/duplicating audio frames.

7.7 W99200F support for A/V synchronization by the System-Mux

The W99200F allows putting time-stamps within the video stream to help the System-Mux in keeping the A/V synchronization. The time-stamp is given as a 33 bits counter in units of 90 KHz clock and put after each frame-header as a user-data field. Timer counting is reset to zero when START is issued. The time-stamp represent the moment in which the first active pixel of the frame was captured i.e. it is actually the capture-time.

The format of the time-stamp is -

4 bytes of User-Data-Code (0x000001B2).
2 bytes of time-stamp identifier (0x0207).
5 bytes of time-stamp itself at the format of -

- bit pad "00000" - 5 bit
- Time-stamp[32..30] - 3 bit
- marker bit "1" - 1 bit
- Time stamp[29..15] - 15 bit
- marker bit "1" - 1 bit
- Time stamp[14..0] - 15 bit

Normally, the time-stamp is rolling forward at a constant rate, implied by the source-video frame-rate. However, the following cases may cause the time-stamp to shift forward -

- Within the telecine sequence - Within the 24 fp/s sequence, a single frame out of any 5 source frames is dropped by the W99200F. Therefore, the distance between two successive frames is either the expected distance of around 1/30 (actually, 1/29.97) of a second or a double distance of around 1/15 of a second.
- The telecine transition point - At which a single video frame might have been dropped, therefore the time-stamp distance may be again 1/15 of a second instead of the normal 1/30 of a second.
- Out-of-sync events - Some frames may be lost due to an out-of-sync event. Note that the W99200F assures that out-of-sync may only cause shifting the time-stamp forward.

The System-Mux mission is to collect the audio and video and setup PTS (Presentation-Time-Stamp) and DTS (Decoding-time-stamp) for both video and audio, allowing synchronized display of both of them under given restrictions. VCD standard requires that the PTS and DTS will be **locked** to quanta of the current video frame-rate. When the System-Mux advances the PTS and DTS in this constant manner, and the real-video is effected by events which cause shifting forward in the actual capture rate - The System-Mux is required to compensate for those events. The W99200F time-stamp serves as a hint to

detect the starting point of the video-stream, the telecine effects and the gap between successive frames before and after an out-of-sync event. Specifically, based on the time-stamp the System-Mux may -

1. Synchronize the audio to the first video-frame - Assuming START was issued for both audio source and W99200F at the same time - the time-stamp of the first frame in **display** order may be used to synchronize or even drop any audio sample that was captured beforehand. For the System-Mux convenience, it may use the first frame in **encoding** order and get the relevant time-stamp using a simple arithmetic based on the time-reference MPEG field.
2. Synchronize audio and video by duplicating or dropping some audio packets according to the actual video capture time as implied by the time-stamp, compared to the given PTS. Naturally, it is recommended to set up a minimal threshold under which the time-stamp and PTS differences are tolerable and does not effect the stream.

Both telecine and Out-of-sync cases are handled correctly according to the last rule. Within a telecine sequence, the PTS advances by 1/24 of a second, while the time-stamp advances by either 1/30 or 1/15 of a second. After each 5 frames - the PTS and the time-stamp accumulate a 1/6 of a second. Assuming a tolerance level for PTS and time-stamp difference - the telecine will not have any frequent effect on the synchronization.
The Clock drift problem (if not resolved by audio codec genlocking to Video source) will also be resolved by the PTS synchronization (only in the case that the Audio codec and W99200F MCLK are genlocked and the PTS is sampled by MCLK).
Note : These guidelines describe the potential usage of the W99200F time-stamp. If required, Winbond will provide a detailed guidelines for generating the DTS & PTS of the muxed stream and dropping/duplicating audio packets. A Reference MUX C- source code is available.

7.8 Reset

The W99200F enters into reset state in five cases:

1. Hardware reset when activating (low) the RESET# pin
2. Total software reset when writing to the Processor Control Register (Reg 0x66) bit 1.
3. Video software reset when writing to the Video_reset register (Reg 0x04).
4. Audio input interface software reset when writing to the AFCR0 (Reg 0x68) bit 6.
5. Audio FIFO software reset when writing to the AFCR0 (Reg 0x68) bit 5.

In cases 1, 2 and 3, the following occurs in video unit of encoder:

- All external registers get their reset value
- All internal state machine are reseted

Only in cases 1 and 2, the following occurs in video unit of encoder:

- All output signals are inactive or in 3-state.
- SDRAM content is unknown.
- Internal memories content is unknown.

7.8.1 Hardware Reset

The W99200F remains in reset state until the pin RESET# goes high.
In power up RESET# should be low (active) for at least 250 microseconds after CLK signal is stable.
There should be at least 10 CLK cycles between RESET# rising (trailing) edge to first host transaction.

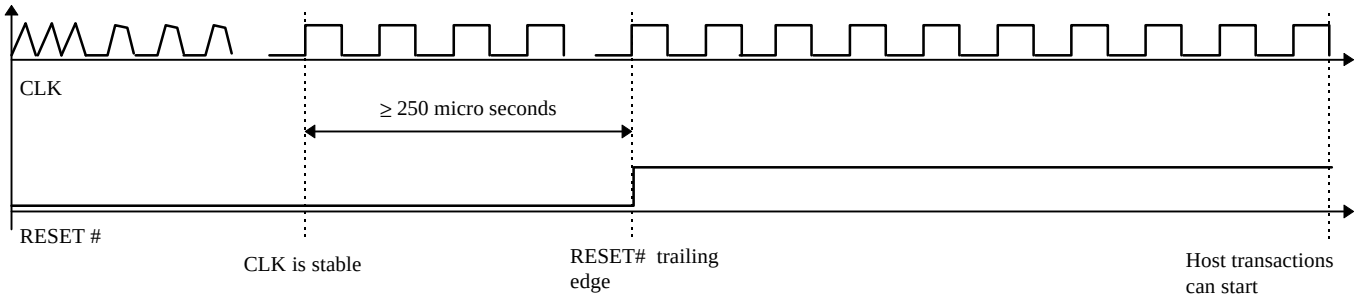


Figure 7-28: CLK and RESET# after power up

7.8.2 Total Software Reset (SRST)

After total software reset, some registers of Bridge unit get their reset value. See the register description for detail.
And after total software reset, most circuits of W99200F get their reset value, except the circuits of the host interface.

7.8.3 Software Video Reset (Video_reset)

The W99200F can get the Video_reset in any stage during operation.
Video_reset does not change the content of the SDRAM.
There should be at least 10 CLK cycles between Video_reset to first host transaction.
After Video_reset, the host must wait 200 micro-seconds before an I^2C transaction, while W99200F issues a STOP condition on the I^2C bus.

7.8.4 Audio Input Interface Software Reset (AIN_RST)

Audio input interface software reset will reset AIN unit. After AIN_RST, some registers of AIN unit get their reset value. See the register description for detail.
And after AIN_RST, most circuits of AIN unit get their reset value, except:
(1) The circuit of the generic bus interface,
(2) The circuit of audio clock generator.

7.8.5 Audio FIFO Software Reset (FIFO_RESET)

This software reset is only used for chip debug. The user is prohibited to use this function.
After FIFO_RESET, the read and write address of audio FIFO, including internal and external, are both reseted to "0".

7.9 Initialization

7.9.1 Power Up and Video only Encoding Mode

After power up, the W99200F is in the encoding mode. In order to normal operation, the host should program W99200F according to the below procedures:

1. Writes "0" into BY_PASS (the bit 2 of PCR). It makes the internal operating clock source got from internal PLL.
2. Adequately programs video unit according to the type of operation mode.

7.9.2 Audio and Video Encoding Mode

After power up, if we want W99200F operating on audio and video encoding mode, the host should program W99200F according to the below procedures:

1. Writes "0" into BY_PASS (the bit 2 of PCR). It makes the internal operating clock source to be got from internal PLL.
2. Programs video unit according to the type of operation mode.
3. Writes the adequate configuration values into AFCR0, AFCR1 and ADIVR.
4. Issues AIN_RST. (the bit 6 of AFCR0).

7.9.3 The Change from Encoding Mode to Decoding Mode

If we want to change operating mode from encoding mode to decoding mode, the host should program W99200F according to the below procedures:

1. Writes "1" into SDRAM_TRI (the bit 7 of PCR). It makes the SDRAM interface of W99200F to be tri-stated.
2. Writes "0" into DE_TRI0 (the bit 5 of PCR) to deassert DEC_WR# and DEC_RD#.
3. Writes "0" into DE_TRI1 (the bit 6 of PCR) to deassert DEC_RESET#.

7.9.4 The Change from Decoding Mode to Encoding Mode

If we want to change operating mode from decoding mode to encoding mode, the host should program W99200F according to the below procedures:

1. Writes "1" into DE_TRI0 and DE_TRI1 (the bits 5 and 6 of PCR) to assert DEC_WR#, DEC_RD# and DEC_RESET#. It makes the output pins of W9926QF to be tri-stated.
2. Writes "0" into SDRAM_TRI (the bit 7 of PCR).
3. Issues SRST (total software reset) (the bit 0 of PCR).
4. Adequately programs video unit and AIN unit.

7.9.5 Restarts the Same Operation Mode

In the encoding mode, if we want to abort the current operation and restart the same operation, the host should program W99200F according to the below procedures:

1. Issues Video_reset to reset video unit.
2. Issues AIN_RST to reset AIN unit. It is not necessary to issue A_STOP to AIN.
3. Reads ALDR0 (Reg 0x6C). It is used to reset audio FIFO overflow counter.
4. Adequately programs video unit.
5. It is not necessary to program AIN unit again.
6. Issues Vstart and A_START again.

7.10 Operation Modes

W99200F has twelve modes of operations:

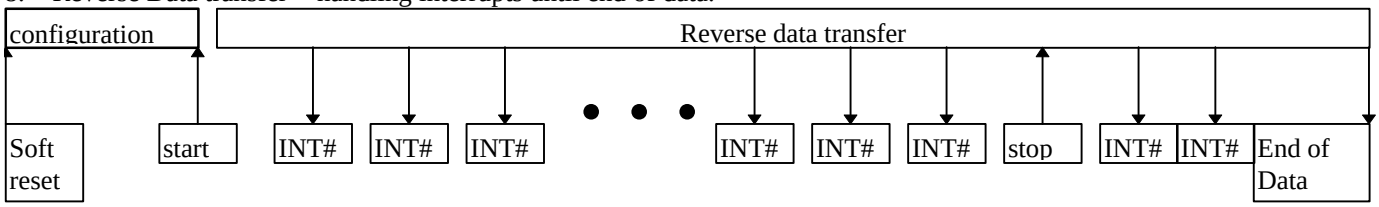
- Real time live video encoding mode
- Real time live video pass through mode
- Live video snap shot mode
- Single frame encoding mode
- Write frame mode

- Read frame mode
- SDRAM write mode
- SDRAM read mode
- Write internal memories mode
- Read internal memories mode
- Audio bitstreams input mode
- VCD bitstreams pass through mode

7.10.1 Live Video Encoding

In live video encoding MPEG1 the following steps should be performed:

1. Video_reset
2. Write internal memories - Initialize memories and registers in the W99200F.
3. Write SDRAM bit-map (optional)
4. Configuration
5. Vstart command
6. Reverse Data transfer + handling interrupts
7. Vstop command
8. Reverse Data transfer + handling interrupts until end of data.



7.10.1.1 Write internal memories

Before starting of Live Video Encoding, part of the internal memories and registers should be initialized. This is done by a write internal memories work mode.

7.10.1.2 Live Video Encoding - Configuration

In the configuration stage the host writes to the relevant configuration registers. These registers should be set prior to encoding. Table 7-11 describes the configuration registers which are relevant to live video encoding - MPEG1. Table 7-12 describes the configuration registers which are relevant to live video encoding - M-JPEG. The registers which are not mentioned are not important and shall not be programmed.

register	Comments
Vint_enable	Enable the relevant interrupts.
Vthreshold	Picks the threshold level.
Vwork_mode	Bits 4:0 = 0x0.
Video_format	Picks SIF, QSIF. Picks picture format: NTSC, PAL, SQUARE
Venc_cntl	Picks bit rate policy, half pixel ME on/off, 'B' start, Closed GOP, Inverse Telecine, Scene change
Vframe_pattern	Picks 'm' and 'n' values
Vbit_rate_m, Vbit_rate_l	Picks the bit rate according to Table 2-4. Ignored if Venc_cntl[1:0] == 0x2
Vbv_size	Picks the VBV size. Ignored in variable bit rate
Vbv_initial	Picks the VBV initial fullness. Ignored in variable bit rate
Vquality	Picks the stream quality. Ignored in constant or maximum bit rate

Vslice_header	Picks the slice header frequency in the bit stream.
Vgop_header	Picks the GOP header and sequence header frequencies in the bit stream.
Vit_hour	Initial time code hours field.
Vit_minute	Initial time code minutes filed.
Vit_second	Initial time code seconds filed.
Vit_frame	Initial time code frame number filed.
Vin_offset	according to video decoder
Vin_cntl	according to video decoder
Vin_picture	according to VCR. Enabling/Disabling bit-map insertion.
Vmem_select	0x00,0x01,0x02,0x03,0x04,0x0C for loading W99200F program

Table 7-11: Configuration registers in live MPEG1 encoding

register	Comments
Vint_enable	Enable the relevant interrupts.
Vthreshold	Picks the threshold level
Vwork_mode	Bits 4:0 = 0x10.
Video_format	Picks SIF, QSIF. Picks picture format: NTSC, PAL, SQUARE
Vframe_pattern	Picks the distance between two encoded frames.
Vquality	Picks the stream quality.
Vin_offset	according to video decoder
Vin_cntl	according to video decoder
Vin_picture	according to VCR. Enabling/Disabling bit-map insertion.
Vmem_select	0x00,0x01,0x02,0x03,0x04,0x0C for loading W99200F program

Table 7-12: Configuration registers in live M-JPEG encoding

7.10.1.3 Live Video Encoding - Reverse Data Transfer

- Data transfer starts when the "Vstart" register is being written. The first frame to be encoded is the first incoming odd field (indicating by a combination of VIVS# and VIHS#) that begins after this register is written.
- During the reverse data transfer the host gets interrupts (if enabled) whenever the internal FIFO passes the threshold level. The host should respond to these interrupts by reading up to threshold level units of data (bytes, words or double-words according to BW) from the FIFO. In case of MAST_EN active the FIFO_RDY signal rises when the FIFO is ready and the host bridge can read the FIFO content and write it on the host bus.
- During the reverse data transfer the host can read the W99200F status registers. Reading the FIFO status register, Vfifo_status, is especially important and gives the host indication about the FIFO fullness condition.
- The W99200F stops getting a new frame in one of the following cases:
    1. Vstop register is written.
    2. FIFO overflow.

  In cases 1 the W99200F ignores the incoming frame during which the event occurs and encodes previous frames. In case of FIFO overflow the W99200F stops encoding. In all 2 cases the host can keep reading FIFO content.
- When the last data is read from the FIFO the "End of data" interrupt is active (if enabled, otherwise host the host should poll the Vint_source register). This completes the data transfer.

7.10.1.4 Live Video Encoding - Data Format

7.10.1.4.1 MPEG1 Encoding

MPEG1 bit stream format is as defined in ISO/IEC 11172-2 standard.

MPEG Video Stream characteristics:

- A single video-sequence unless automatic inverse telecine is activated, for which another video-sequence may be initialized for each modification of the input format from a 29.97 frames/second stream to a 23.976 frames/second stream and vice-versa.
- According to a user-selected n,m values, a constant frame-pattern is used, unless a scene-change is detected.
- The first frame in a GOP is a B-picture unless m<2. The usage of a B-frame at the first GOP within a session is user-selected.
- Repeating frequency of any of the MPEG headers (sequence header, GOP header, slice header) is user-selected.
- In case of a scene change detection a new GOP is opened and if the user asked for repeating sequence headers then also a sequence header is inserted into the bit-stream.
- No D-pictures.
- Quantization uses the default matrixes for both inter and intra.

User data in MPEG1 stream:

The VCD standard permits an insertion of user data into the picture header according to MPEG1 standard. The VCD standard requires that this data will be composed of fields, where each field has the following format:

Byte #	data
1	T = field type
2	L = field length
3	field data byte #1
.	.
L	field data byte #(L-2)

Table 7-13: User data format in VCD 2.0

W99200F utilizes the picture user data section for the following data:

1. Time-stamp information. The time stamp represent the actual capturing time of the frame. It is a 33 bit number represent the time in 90KHz cycles from the "start" command (writing to Vstart register) until the capturing time of first active line_of the picture being encoded. Time stamp may be used by the MPEG1 sys-mux for video-audio synchronization. Time stamp insertion is optional according to Venc_cntl register bit 7. If enabled time stamp is included in each frame. Time stamp appears in the user data in 5 bytes (40 bits) format according to the following table:

bit pad "00000"	5 bit
Time stamp[32..30]	3 bit
marker bit "1"	1 bit
Time stamp[29..15]	15 bit
marker bit "1"	1 bit
Time stamp[14..0]	15 bit
Total	40 bit

2. Supporting Scan information. Scan information is a special field type used by the VCD 2.0 to give information on the sectors where the next and previous I-frames are (for trick modes like: fast-forward, fast-rewind, etc). This information appears only in I-frames and can only be written by an authoring tool that actually writes to the CD. The length of the scan data field is 14 bytes. The W99200F supports scan information by including 14 bytes of user data in 'I' frames. The

authoring tool can farther replace these bytes by the Scan-data. The VBV fullness is being calculated by taking these bytes into account. If enabled Scan information appears in each 'I' frame.

3. Various information on the encoded sequence. This information appears in the first 'I' frame after sequence headers and it describes the characteristics of this sequence. It is 40 bits data according to the following figure:

Bit 0 New Scene	Bit 1 reserved	Bit 2 reserved	Bit 3 reserved	...	Bit 39 reserved

Figure 7-29: Sequence information in picture user data

When the new scene bit is on (Bit-0) then the following sequence is a result of a scene change. This information may be used by a farther editing tool for editing. Other bits are reserved for future use.

The following table summarize the user data :

Frame type	User data if Time stamp disabled	User data if Time stamp enabled	
1st 'I' frame after sequence header	0x000001B2 - User data start code 0x01 - Field type 0x07 - Field length <5 bytes Sequence info>	0x000001B2 - User data start code 0x02 - Field type 0x07 - Field length < 5 bytes TS > 0x03 - Field type 0x07 - Field length 0xFFFFFFFFFF 0x01 - Field type 0x07 - Field length <5 bytes Sequence info>	This section can be replaced by scan information
Other 'I' frames	None	0x000001B2 - User data start code 0x02 - Field type 0x07 - Field length < 5 bytes TS > 0x03 - Field type 0x07 - Field length 0xFFFFFFFFFF	This section can be replaced by scan information
'P' or 'B' frames	None	0x000001B2 - User data start code 0x02 - Field type 0x07 - Field length < 5 bytes TS >	

Table 7-14: User data format in W99200F

7.10.1.4.2 M-JPEG Encoding

M-JPEG bit stream is sent as a sequence of JPEG pictures (according to ISO/IEC 10918-1). There are no headers or any other data between the JPEG pictures.

JPEG Video Stream characteristics:

- Baseline-sequential process.
- A single scan of 3 components interleaved.

- Interchange format.
- Quantization is based on 62 different sets of constant matrixes corresponding for 31 levels of quality multiplied by {LUMA, CHROMA}.

7.10.1.4.3 Bit map insertion

A user defined bit-map can be combined into the incoming video (e.g., for creating a company Logo). To achieve that the host should first prepare the bit-map using Write SDRAM mode and then enable the bit map insertion by setting Vin_picture[1] bit. The bit-map insertion can be switch on/off in a frame by frame basis.

7.10.1.5 MPEG1 / MJPEG bits order on the data bus (D) according to bus-width

Lets assume that MPEG1 / MJPEG produces 32 bits according to the following order:

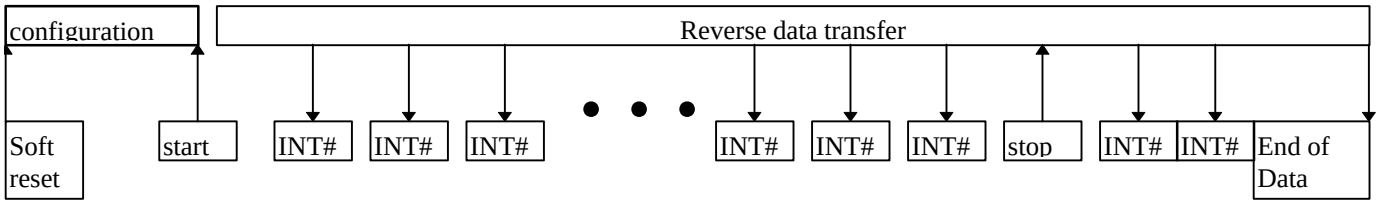
- b0 - ^{1}st bit that was produced
- b1 - ^{2}nd bit that was produced
- b31 - ^{32}nd bit that was produced

a) When sending this information on the data bus working in 8-bit mode:
^{1}st byte to be sent: D[7:0] == {b0,b1,b2,....,b7}
^{2}nd byte to be sent: D[7:0] == {b8,b9,b10,...,b15}
^{3}rd byte to be sent: D[7:0] == {b16,b17,...,b23}
^{4}th byte to be sent: D[7:0] == {b24,b25,....,b31}
b) When sending this information on the data bus working in 16-bit mode
^{1}st word to be sent: D[15:0] == {b8,b9,..,b15,b0,b1,...,b7}
^{2}nd word to be sent: D[15:0] == {b24,b25,..,b31,b16,b17,..,b23}
c) When sending this information on the data bus working in 32-bit mode
D[31:0] == {b24,b25,..,b31,b16,b17,..,b23,b8,b9,..,b15,b0,b1,...,b7}

7.10.2 Live Video Pass Through

In live video pass through the following steps should be performed:

1. Video_reset
2. Write SDRAM bit-map (optional).
3. Configuration
4. Vstart command
5. Reverse Data transfer + handling interrupts
6. Vstop command
7. Reverse Data transfer + handling interrupts until end of data



7.10.2.1 Live Video Pass Through - Configuration

In the configuration stage the host writes to the relevant configuration registers. These registers should be set prior to encoding. Table 7-15 describes the configuration registers which are relevant to live video pass through. The registers which are not mentioned are not relevant and shall not be programmed.

register	Comments
Vint_enable	Enable the relevant interrupts.
Vthreshold	Picks the threshold level
Vwork_mode	Bits 4:0 = 0x1.
Video_format	Picks picture size: SIF, QSIF. Picks picture format: NTSC,PAL, SQUARE
Vframe_pattern	Picks the distance between two frames.
Vin_offset	according to video decoder
Vin_cntl	according to video decoder
Vin_picture	according to VCR. Enabling/Disabling bit-map insertion.
Vmem_select	0x00,0x01,0x02,0x03,0x04,0x0C for loading W99200F program

Table 7-15: Configuration registers in live video pass through

7.10.2.2 Live Video Pass Through - Reverse Data Transfer

- Data transfer starts when the "Vstart" register is being written. The first frame to be transferred is the first incoming odd field (indicating by a combination of VIVS# and VIHS#) that begins after this register is written.
- During the reverse data transfer the host gets interrupts (if enabled) whenever the internal FIFO passes the threshold level. The host should respond to these interrupts by reading up to threshold_level units of data (bytes, words or double-words according to BW) from the FIFO. In case of MAST_EN active the FIFO_RDY signal rises when the FIFO is ready and the host bridge can read the FIFO content and write it on the host bus.
- During the reverse data transfer the host can read the W99200F status registers. Reading the FIFO status register, Vfifo_status, is especially important and gives the host indication about the FIFO fullness condition.
- The W99200F stops getting new frames in one of the following cases:
    1. Vstop register is written.
    2. FIFO overflow.

  In all cases above the W99200F ignores the incoming frame during which the event occurs. Previous store frame can still be read by the host.
- When the last data is read from the FIFO the "End of data" interrupt is active (if enabled, otherwise host the host should poll the Vint_source register). This completes the data transfer.

7.10.2.3 Live Video Pass Through - Data Format

In live video pass through data (4:2:0 YCbCr) comes according to the following order:
1. "start_frame" flag (4 bytes): 0x00, 0x00, 0x00, 0x00
2. Luma (Y) component of the entire frame in a raster scan order.
3. Chroma (Cb) component of the entire frame in a raster scan order
4. Chroma (Cr) component of the entire frame in a raster scan order
5. "end_frame" flag (4 bytes): 0xFF, 0xFF, 0xFF, 0xFF

Pixels format is according to ITU-R 601 (especially 0x00 and 0xFF values are not allowed)

Table 7-16 shows the number of bytes per frame that is transmitted in all supported pictures formats. The host is responsible (by selecting the distance between frames) that total bit rate will not exceed the bus bandwidth.

Picture format and size	Number of bytes per frame			
	Luma	Cr	Cb	Total (including "start"+"end")
NTSC - SIF	84,480	21,120	21,120	126,728
NTSC - QSIF	19,712	4,928	4,928	29,576

PAL - SIF	101,376	25,344	25,344	152,072
PAL - QSIF	25,344	6,336	6,336	38,024
SQUARE - SIF	76,800	19,200	19,200	115,208
SQUARE - QSIF	17,920	4,480	4,480	26,888

Table 7-16: Number of bytes per frame in Live video pass through

7.10.2.3.1 Bit map insertion

A user defined bit-map can be combined into the incoming video (e.g., for creating a company Logo). To achieve that the host should first prepare the bit-map using Write SDRAM mode and then enable the bit map insertion by setting Vin_picture[1] bit. The bit-map insertion can be switch on/off in a frame by frame basis.

7.10.2.4 Pixels order on the data bus (D) according to bus-width

Lets assume pixels are organized in a frame according to the following order:
top line left corner (LUMA pixels): Y0, Y1, Y2, Y3, Y4,.....

a) Working in 8 bit mode the pixels are read according to the following order:
   D[7:0] = Y0 - 1st transaction
   D[7:0] = Y1 - 2nd transaction
   D[7:0] = Y2 - 3rd transaction
   D[7:0] = Y3 - 4th transaction
b) Working in 16 bit mode:
   D[15:0] = {Y1 Y0} - 1st transaction
   D[15:0] = {Y3 Y2} - 2nd transaction
c) Working in 32 bit mode:
   D[31:0] = {Y3 Y2 Y1 Y0}

7.10.3 Live Video Snap Shot

Live Video Snap shot captured a single frame from the incoming video and stores it into the SDRAM. If the successive operation is Single Frame Encoding or Read Frame, the content of the SDRAM remains unchanged unless hardware reset was applied.

In live video snap shot the following steps should be performed:

1. Video_reset
2. Write SDRAM bit-map (optional).
3. Configuration
4. Vstart command
5. Wait to end of data

7.10.3.1 Live video Snap Shot - Configuration

In the configuration stage the host writes to the relevant configuration registers. These registers should be set prior to encoding. Table 7-17 describes the configuration registers which are relevant to live video snap shot. The registers which are not mentioned are not relevant and shall not be programmed.

register	Comments
Vint_enable	Enable the relevant interrupts.
Vthreshold	Picks the threshold level
Vwork_mode	Bits 4:0 = 0x2.
Video_format	Picks picture size: FULL, SIF, QSIF. Picks picture format: NTSC,PAL, SQUARE
Vin_offset	according to video decoder

Vin_cntl	according to video decoder
Vin_picture	according to VCR. Enabling/Disabling bit-map insertion.
Vmem_select	0x00,0x01,0x02,0x03,0x04,0x0C for loading W99200F program

Table 7-17: Configuration registers in live video snap shot

7.10.3.1.1 Bit map insertion

A user defined bit-map can be combined into the snap-shot picture (e.g., for creating a company Logo). To achieve that the host should first prepare the bit-map using Write SDRAM mode and then enable the bit map insertion by setting Vin_picture[1] bit.

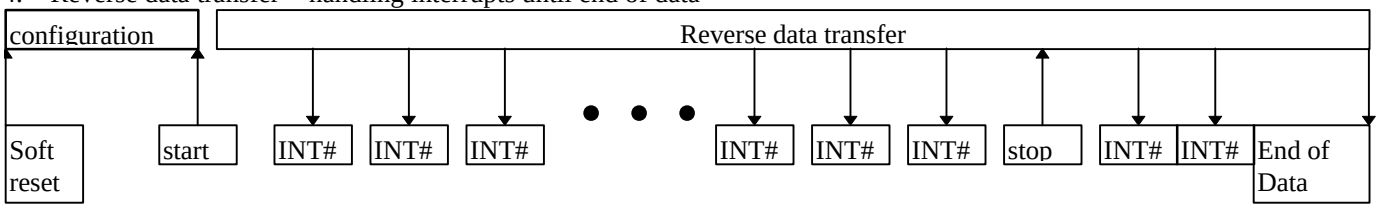
7.10.3.2 Pixels order on the data bus (D) according to bus-width

See 7.10.2.4

7.10.4 Single Frame Encoding

In single frame encoding the following steps should be performed:

1. Video_reset
2. Configuration
3. Vstart command
4. Reverse data transfer + handling interrupts until end of data



7.10.4.1 Single Frame Encoding - Configuration

Table 7-18 describes the configuration registers which are relevant to single frame encoding MPEG1 and M-JPEG. The registers which are not mentioned are not important and shall not be programmed.

register	Comments
Vint_enable	Enable the relevant interrupts.
Vthreshold	Picks the threshold level
Vwork_mode	Bits 3:0 = 0x3. Picks MPEG1 or JPEG
Video_format	Picks picture format: NTSC, PAL, SQUARE and by that the frame rate
Vquality	Picks the picture quality
Vsize_h	Picks the frame horizontal size.
Vsize_v	Picks the frame vertical size.
Vslice_header	Picks the slice header frequency in the bit stream.
Vit_hour	Initial time code hours field.
Vit_minute	Initial time code minutes filed.
Vit_second	Initial time code seconds filed.
Vit_frame	Initial time code frame number filed.

Table 7-18: Configuration registers in Single frame encoding

7.10.4.2 Single Frame Encoding - Reverse Data Transfer

- Reverse data transfer starts after start register is being written.
- During the reverse data transfer the host gets interrupts (if enabled) whenever the internal FIFO passes the threshold level. The host should respond to these interrupts by reading up to threshold_level units of data (bytes, words or double-words according to BW) from the FIFO. In case of MAST_EN active the FIFO_RDY signal rises when the FIFO is ready and the host bridge can read the FIFO content and write it on the host bus.
- When the last data is read from the FIFO the "End of data" interrupt is active (if enabled, otherwise host the host should poll the Vint_source register). This completes the data transfer.

7.10.4.3 Single Frame Encoding - Data Format

The reverse data format depends on the encoding mode: MPEG1 'I' frame or JPEG:

1. In MPEG1 'I' frame the data is being sent in the following way:
    - sequence header ()
    - group_start_code
    - time_code (not relevant 25 bits = 0x0001000)
    - closed_gop (1 bit = 1)
    - broken_link (1 bit = 0)
    - picture()
    - sequence_end_code

2. In JPEG the data is as defined at 7.10.1.4.2.

7.10.4.4 MPEG1 / MJPEG bits order on the data bus (D) according to bus-width

See 7.10.1.5

7.10.5 Write frame mode

Write frame mode writes a single frame and stores it into the SDRAM. If the successive operation is Single Frame Encoding or Read Frame, the content of the SDRAM remains unchanged unless hardware reset was applied.

In Write frame mode the following steps should be performed:

1. Video_reset
2. Configuration
3. Vstart command
4. Forward Data transfer according to data format
5. Vstop command
6. Read the data error bit (Vint_source [3]) in order to determine whether the forward data transfer succeeded. Data error should be 0.

7.10.5.1 Configuration

register	Comments
Vwork_mode	Bits 4:0 = 0x4.
Vsize_h	Frame size horizontal
Vsize_v	Frame size vertical

Table 7-19: Configuration registers in Write frame mode

7.10.5.2 Data format

1. Luma component of the entire frame in raster scan order.
   Size (in bytes): 256 × (size_h) × (size_v)
2. Chroma (Cb) component of the entire frame in raster scan order.
   Size (in bytes): 64 × (size_h) × (size_v)
3. Chroma (Cr) component of the entire frame in raster scan order.
   Size (in bytes): 64 × (size_h) × (size_v)

Note: Pixels value are limited according to ITU-R 601 recommendation:
1. *Luma (Y) pixels should be in the range of 0x10 (=16) to 0xEB (=235)*
2. *Croma pixels (Cb,Cr) should be in the range of 0x10 (=16) to 0xF0 (=240)*

7.10.5.3 Pixels order on the data bus (D) according to bus-width

See 7.10.2.4.

7.10.6 Read frame mode

In Read frame mode the following steps should be performed:
1. Video_reset
2. Configuration
3. Vstart command
4. Reverse Data transfer

7.10.6.1 Configuration

register	Comments
Vint_enable	Enable the relevant interrupts.
Vthreshold	Picks the threshold level
Vwork_mode	Bits 4:0 = 0x5
Vsize_h	Frame horizontal size
Vsize_v	Frame vertical size

Table 7-20: Configuration registers in Read frame mode

7.10.6.2 Reverse data transfer

- Data transfer starts when the "start" register is being written.
- During the reverse data transfer the host gets interrupts (if enabled) whenever the internal FIFO passes the threshold level. The host should respond to these interrupts by reading up to threshold_level units of data (bytes, words or double-words according to BW) from the FIFO. In case of MAST_EN active the FIFO_RDY signal rises when the FIFO is ready and the host bridge can read the FIFO content and write it on the host bus.
- When the last data is read from the FIFO the "End of data" interrupt is active (if enabled, otherwise host should poll the Vint_source register). This completes the data transfer.

7.10.6.3 Read Frame mode - Data Format

1. "start_frame" flag (4 bytes): 0x00, 0x00, 0x00, 0x00
2. Luma (Y) component of the entire frame in a raster scan order.
3. Chroma (Cb) component of the entire frame in a raster scan order
4. Chroma (Cr) component of the entire frame in a raster scan order
5. "end_frame" flag (4 bytes): 0xFF, 0xFF, 0xFF, 0xFF

Pixels format is according to ITU-R 601 (especially 0x00 and 0xFF values are not allowed)

7.10.6.4 Pixels order on the data bus (D) according to bus-width

See 7.10.2.4.

7.10.7 SDRAM write mode

The W99200F enables the host to write any data to the SDRAM. This can only be done when no other working mode is active. If the successive operation is Read SDRAM mode, the content of the SDRAM remains unchanged unless hardware reset was applied.
The SDRAM is divided into 256 blocks. Each block contains 8 SDRAM rows which are 8192 Bytes (8 rows × 2 banks × 256 columns × 2 Bytes).
In SDRAM write mode the following steps should be performed:

1. Video_reset
2. Configuration
3. Vstart command
4. Forward Data transfer
5. Vstop command

7.10.7.1 Configuration

register	Comments
Vwork_mode	Bits 4:0 = 0x6.
Vmem_select	Selects the SDRAM part to be written

Table 7-21: Configuration registers in SDRAM write mode

7.10.7.2 Forward data transfer

- Forward data transfer starts after Vstart register is being written.
- During the data transfer the host writes any data to the Vdata_in register. The W99200F assembles 8 bytes and writes them to the SDRAM (Less than 8 bytes are not written). The host can write any amount of data (in integer multiples of 8 bytes) up to 8192 bytes into a single SDRAM block.
  Note: The host can not start writing to any address other than the block first address.
- Data transfer ends when Vstop register is being written.

7.10.7.3 Bit map insertion

A special case of SDRAM write mode is inserting a user-defined bit-map into the incoming video for example in order to insert the company's Logo (in LVE, LVPT or LVSS modes). Bit map size is 6 MBs (96 pixels) width and 2 MBs (32 pixels) height in all frame sizes: FULL, SIF and QSIF. Bit map location inside the picture is user selectable.
The bit-map is organized as a Y,Cb,Cr 4:2:0 pixels where each pixel can have a value of 0 or *ITU*-R601 values (i.e., Luma (Y) pixels should be in the range of 0x10 (=16) to 0xEB (=235), Croma pixels (Cb,Cr) should be in the range of 0x10 (=16) to 0xF0 (=240)). If the pixels has the value 0 it means that this pixel is transparent and the original frame pixel should be put instead.
The bit map should be written into page 0xDF in the SDRAM (Vmem_select = 0xDF).
The bit-map can be put into the incoming video in LVE, LVPT or LVSS mode. The bit-map content (page 0xDF in the SDRAM) remains unchanged in all working modes unless hardware reset is applied.

MB 0,0	MB 1,0	MB 2,0	MB 3,0	MB 4,0	MB 5,0
MB 0,1	MB 1,1	MB 2,1	MB 3,1	MB 4,1	MB 5,1

Figure 7-30: Bit map divided to MBs

Data format is:

- The data is written into the SDRAM in the following MB order - (0,0) , (0,1) , (1,0) , (1,1) ... (5,0) , (5,1)
- Each MB is written as 256 LUMA pixels followed by 64 Cb pixels, 64 zeros, 64 Cr pixels and 64 zeros (total is 512 bytes per MB).
- Order inside each MB color component is raster-scan.
- 12 MBs are 6144 bytes, after which two bytes are written for the bit-map position inside the frame:
  - Horizontal position of the top-left corner in 16 pixels resolution
  - Vertical position of the top-left corner in 8 pixels resolution

  The position should be such that the whole bit-map area does not exceed the frame border according to the frame size
- Each value (x) of the bit-map is mixed with the incoming video frame by using x whenever x>0 or using the frame original value whenever x=0.

Using the bit map insertion is optional and is enabled by Vin_picture [1]. The Vin_picture[1] can be changed during LVE and LVPT modes in order to switch on/off the bit map insertion only on selected frames.

7.10.8 SDRAM read mode

The W99200F enables the host to read any data from the SDRAM. This can only be done when no other working mode is active. The SDRAM is divided into 256 blocks. Each block contains 8 SDRAM rows which are 8192 Bytes (8 rows × 2 banks × 256 columns × 2 Bytes).

In SDRAM read mode the following steps should be performed:

1. Video_reset
2. Configuration
3. Vstart command
4. Reverse Data transfer
5. Vstop command (optional)

7.10.8.1 Configuration

register	Comments
Vint_enable	Enable FIFO ready and end of data interrupts.
Vthreshold	Picks the threshold level
Vwork_mode	Bits 4:0 = 0x7
Vmem_select	Selects the SDRAM part to be read

Table 7-22: Configuration registers in SDRAM write mode

7.10.8.2 Reverse data transfer

- Data transfer starts when the "Vstart" register is being written.

- During the reverse data transfer the host gets interrupts (if enabled) whenever the internal FIFO passes the threshold level. The host should respond to these interrupts by reading up to threshold_level units of data (bytes, words or double-words according to BW) from the FIFO. In case of MAST_EN active the FIFO_RDY signal rises when the FIFO is ready and the host bridge can read the FIFO content and write it on the host bus.
- When the last data is read from the FIFO the "End of data" interrupt is active (if enabled, otherwise host the host should poll the Vint_source register). This completes the data transfer.
- In case the host wants to stop the data transfer before end of data, it can write into the Vstop register.

7.10.9 Internal memories write mode

This mode enables the host to initialize the W99200F internal memories and registers. This is done for three purposes:

1. Load the W99200F 1536 bytes of program at initialization time. This load is done before live video encoding, live video pass-through and live video snap-shot. Enabling the host to perform this initialization increases the programmability of the W99200F.
2. Check the internal memories during production tests.
3. Check the units connected to the internal memories during production tests.

In order to run this work mode the host should do the following:

1. Video_reset
2. Configuration :
   - Vwork_mode = 0x8,
   - Vmem_select - selects the memory to be written
3. Vstart command
4. Forward Data transfer to Vdata_in register. Exactly 256 bytes should be written.
5. In case there is a need to initialize more than one memory steps 2-4 above should be repeated

7.10.10 Internal memories read mode.

This mode enables the host to read the W99200F internal memories and registers. This is done in order to increase the observability of the encoder.
In order to run this work mode the host should do the following:

1. Configuration
2. Vstart command
3. Reverse Data transfer

7.10.10.1 Configuration

register	Comments
Vint_enable	Enable FIFO ready and end of data interrupts.
Vthreshold	Picks the threshold level
Vwork_mode	Bits 4:0 = 0x9
Vmem_select	Selects the internal memory to be read

Table 7-23: Configuration registers in SDRAM write mode

7.10.10.2 Reverse data transfer

- Data transfer starts when the "Vstart" register is being written.
- During the reverse data transfer the host gets interrupts (if enabled) whenever the internal FIFO passes the threshold level. The host should respond to these interrupts by reading up to threshold_level units of data (bytes, words or double-words according to BW) from the FIFO. In case of MAST_EN active the FIFO_RDY signal rises when the FIFO is ready and the host bridge can read the FIFO content and write it on the host bus.

- When the last data is read from the FIFO the "End of data" interrupt is active (if enabled, otherwise host the host should poll the Vint_source register). This completes the data transfer.

7.10.11 Audio bitstreams input mode

This mode enables the host to read the audio bitstreams from W99200F. This is done in order to get perfect A/V synchronization during encoding.
In order to run this work mode the host should do the following:

1. Configuration
2. AIN software reset. (AIN_RST)
3. Vstart command (A_START)
4. Vstop command (A_STOP)

7.10.11.1 Configuration

register	Comments
AFCR0	OSYNC_EN=1, AF_OV_EN=1, AF_EOD_EN=1, AF_R_EN=1, A_THRES[1:0]= adequate value.
AFCR1	EX_FIFO and EX_FIFO_TYPE[1:0] are equal to adequate values.
ADIVR	set the adequate value.

Table 7-24: Configuration registers in Audio bitstreams input mode

7.10.11.2 Audio bitstreams transfer

- audio bitstream transfer starts when writing "1" into the "A_START".
- During the audio bitstream transfer the host gets interrupts (if enabled) whenever the internal audio FIFO passes the threshold level. The host should response to these interrupts by reading up to threshold_level units of data from the audio FIFO.
- During the audio bitstream transfer the host gets interrupts (if enabled) whenever the audio FIFO is overflow. The host should respond to these interrupts by reading the ALDR0 and ALDR1 to get the number of audio lost data.
- After the host issues A_STOP and when the last data is read from the FIFO the "End of data" interrupt is active (if enabled, otherwise the host should poll the Vint_source register). This completes the data transfer.

7.10.12 VCD bitstreams pass through mode

This mode enables the host to send the VCD bitstreams to VCD decoder pass through W99200F.
In order to run this work mode the host should do the following:

1. Configuration W99200F
2. Configuration VCD decoder

7.10.12.1 the sequence of W99200F configuration

register	Comments
PCR	SDRAM_TRI=1
PCR	DE_TRI0=0
PCR	DE_TRI1=0

Table 7-25: Configuration sequence in VCD bitstreams pass through mode

7.10.12.2 VCD decoder configuration

Refer to W9925QF or W9926QF data sheet for detail.

7.10.12.3 Interrupt dealing

During VCD bitstreams pass through mode, W99200F can pass through VCD decoder interrupt requests to the host. The host should respond to these interrupts. Refer to W9925QF or W9926QF data sheet for detail.

7.11 Dealing with interrupt sources

There are 4 kinds of interrupt sources in the W99200F:

1. Video interrupt.
2. Audio interrupt.
3. PCI bridge interrupt.
4. VCD decoder interrupt.

7.11.1 Video interrupt

There are 7 video interrupt sources:

1. Video end of data - Available in working modes with reverse data transfer to inform the host that no more data is going to be produced. Thus, this "session" is ended.
2. Video FIFO ready - Available in working modes with reverse data transfer to inform the host that data in the W99200F local FIFO passed the threshold level and is ready for the host to read.
3. Video FIFO overflow - Available in Live video encoding and in live video pass-through. In live video encoding it informs the host that SDRAM FIFO is full (with 256K bytes of MPEG1/M-JPEG bit stream) and the W99200F was not able to store new produced data. In live video pass-through it informs the host that four unread frames are stored in the SDRAM and there is no place to store more frames. In both cases the W99200F stops working, the already stored data is unchanged but new data is not stored any more.
4. Video data error - In write frame mode: The number of bytes that were written to the W99200F does not match the expected number of bytes.
5. Vin out of sync - Available in Live video encoding, Live video pass-through and Live video snap shot. Informs the host that the Video decoder is out-of-sync. In this case the W99200F drops the frame that the out of sync is received on and then tries to get synchronized again. In case of out of sync in LVSS mode, W99200F stops.
6. VBV underflow - Available in Live video encoding - MPEG1 constant bit rate. Informs the host that VBV underflow occurs. The W99200F keeps working .
7. I2c ready - In I2C transactions informs the host that the transaction ended. This is the only interrupt available in the configuration stage.

When an interrupt condition occurs the relevant bit in the Vint_source register is set. The host can poll this register in order to be informed on the interrupt sources. All interrupts can be enabled or disabled by using Vint_enable register. When an interrupt condition occurs and if the interrupt is enabled the INT# signal is activate (low).

The host can clear the interrupt by writing to the relevant bit in Vint_clear register, by doing that the interrupt is cleared from the Vint_source register. If this interrupt source is the only active and enabled interrupt the INT# signal is deactivated.

Video FIFO ready interrupt which is usually the most frequent interrupt is also cleared when the host read the Vint_rd_clear register. This is done in order to save the host the need to access two registers: Vint_source and Vint_clear. The Vint_rd_clear replaces these two accesses.

The next sections discuss the recommended way of treatment in the different kinds of interrupts:

7.11.1.1 Video end of data

In all video working modes video data transfer ends when the video End of Data bit is set. Because the video FIFO is usually been read in "chunks" of data (according to bus width and threshold level), the W99200F stuffs with zeros after the last valid data. The host should clear the end-of-data interrupt and then "clean" the stuffing zeros. This can be done in the following way:

1. In Live video encoding or single frame encoding MPEG1 - All data that comes after "sequence_end_code" (See standard)
2. In live video encoding or single frame encoding JPEG - All data that comes after the last "end of image" (See standard) code.
3. In live video pass through or read frame from SDRAM - All data that comes after the last "end_of_frame" flag (=0xFFFFFFFF).

7.11.1.2 Video FIFO ready

The W99200F includes 256K bytes video FIFO that buffers the produced data. The FIFO threshold can be configured and an interrupt is assigned (if enabled) whenever data in the FIFO passed threshold level condition. The host should clear the FIFO ready interrupt and read threshold level data elements from the FIFO. This can be done by the host using DMA transactions.
For example if BW = 00 (8 bits) and threshold level is 0x4 (=16) then whenever there are 16 bytes ready in the FIFO an interrupt or FIFO_RDY occurs. The host should respond to this interrupt by reading these 16 bytes from the FIFO. If produced bit rate is 2M bits/second = 0.25M bytes/second then interrupt frequency is ~15,000 times a second (every 64 micro second).

7.11.1.3 Video FIFO Overflow

If a video FIFO overflow error occurs, the W99200F stops working. The host should clear the FIFO overflow interrupt, it may keep reading the FIFO until End of Data indication .

In order to prevent FIFO overflow it is recommended that the FIFO will be close to empty most of the time. When working in live video encoding or pass through modes, the host should check the FIFO condition by reading the FIFO status register. FIFO fullness is given in this register. In case that FIFO fullness is getting too high the host can do the following:

1. In live video encoding MPEG1 constant or maximum bit rate, host should stop the encoding process and restart it with a lower bit rate demand.
2. In live video encoding M-JPEG or MPEG1 variable bit rate, the host can lower the quality value written in the Vquality register and by that lower the rate of the generated bit stream.
3. In live video pass through the host should stop the W99200F, enlarge the distance between the transferred frames and then restart the process.

7.11.1.4 Video data error

Video data error may be active in write frame mode in order to inform the host that the amount of data written to the W99200F (between "Vstart" and "Vstop" commands) is different from the expected. The host should repeat the write frame mode and check this bit again..

7.11.1.5 Vin out-of-sync

In case of out of sync in the incoming video an interrupt is issued (if enabled). The W99200F stops getting new frames but completes encoding (or sending) the ones that were already captured. The host should clear the Vin out-of-sync interrupt, read the already stored data and wait for the End of Data indication. The user should then check and fix potential problems in the incoming video signal and restart the encoding/monitoring process.

7.11.1.6 VBV underflow

VBV underflow is NOT a fatal error problem . It is more a warning message. The host should clear the interrupt and may keep on working .

7.11.1.7 I2C ready

In case of i2c ready interrupt the host should clear the interrupt and read the i2c_status register in order to determine whether the transaction succeeded or not. If the transaction not succeeded the host can repeat the last transaction.

7.11.2 Audio interrupt

There are 3 audio interrupt sources:

1. Audio end of data - Available in audio bitstream input mode. It is used to indicate that the host has read the last byte in the audio bitstream and no more audio data is going to be produced.
2. Audio FIFO ready - Available in audio bitstream input mode. It informs the host that data in the W99200F audio FIFO passed the threshold level and is ready for the host to read.
3. Audio FIFO overflow - Available in audio bitstream input mode. It informs the host that audio FIFO is overflow. In this case W99200F doesn't stop working, but W99200F stops to write new audio data into FIFO. While the host has read data from FIFO, that means FIFO is not empty, W99200F starts to write current audio data into FIFO.

When an interrupt condition occurs the relevant bit in the "Audio FIFO Interrupt Source Register" is set. The host can poll this register in order to be informed on the interrupt sources. All interrupts can be enabled or disabled by using "Audio FIFO Control Register 0". When an interrupt condition occurs and if the interrupt is enabled the INT# signal is activate (low).

The host can clear the interrupt by writing to the relevant bit in "Audio FIFO Interrupt Clear Register", by doing that the interrupt is cleared from the "Audio FIFO Interrupt Source Register". If this interrupt source is the only active and enabled interrupt the INT# signal is deactivated.

The next sections discuss the recommended way of treatment in the different kinds of interrupts:

7.11.2.1 Audio end of data

Audio data transfer ends when the audio End of Data bit is set. Because the audio FIFO is usually been read in "chunks" of data (according to bus width and audio threshold level), the host may not read all data in audio FIFO and lose a small amount of audio data. The number of lost audio data is less than threshold level.

7.11.2.2 Audio FIFO ready

The audio FIFO threshold can be configured and an interrupt is assigned (if enabled) whenever data in the FIFO passed threshold level condition. The host should clear the FIFO ready interrupt and read threshold level data elements from the FIFO. After the host clears audio FIFO ready interrupt, this interrupt is masked until the host reads the amount of audio threshold level data from audio FIFO.

For example if threshold level is 0x01 (=256 bytes) then whenever there are 256 bytes ready in the FIFO an interrupt occurs. The host should respond to this interrupt by reading these 256 bytes from the FIFO.

7.11.2.3 Audio FIFO overflow

If an audio FIFO overflow error occurs, the W99200F doesn't stop working. The host should read ALDR0 and ALDR1 registers to know how many audio data are lost. This information is helpful for A/V synchronization. The host should clear the FIFO overflow interrupt.

7.11.3 PCI bridge interrupt

PCI bridge interrupt occurs only in PCI master mode. It informs the host that the video bitstream buffer is full. It is available in the video reverse data transfer. See 7.1.1.4 for detail.

7.11.4 VCD decoder interrupt

W99200F can pass through VCD decoder interrupt requests to the host. See W9925QF or W9926QF data sheet for detail.

8. Control And Status Registers

Table 8-1 describes the W99200F register set. There are five groups of registers:

1. Data register including 4 registers:
   - Vdata_out register (register 0x00) - This *Read Only* register is used to transfer video data from W99200F in reverse data transaction.
   - Vdata_in register (register 0x01) - This *Write only* register is used to transfer video data to W99200F in forward data transactions.
   - Audio_out register (register 0x67) - This *Read Only* register is used to transfer audio data from W99200F in reverse data transaction.
   - I2c_data register which is used to write/read data into/from the video decoder through the i2c bus.

2. Command registers - These are *Write only* registers which are used to reset the W99200F, to start/stop data processing and to start/stop and acknowledge I^2C transfer. All command registers initiate the relevant command when any data is written into them. After the write transaction the registers are ready for a new command.

3. Interrupt registers - These registers are used to activate various source of interrupts, to read interrupt status and to clear active interrupt sources.

4. Configuration registers - 8/32 bit wide registers which should be set prior to data transferring. Most of the video configuration registers (except for the Vquality and Vin_picture registers) should not be changed during data transfer.

5. Status registers *Read Only* registers which could be read anytime to check the FIFO status or other information on the process being held.

Table 8-1: W99200F registers

index	Group	Name	Description
0x00	Data	Vdata_out	In reverse transactions: FIFO output port. In generic bus host type, reading this register is done by the FIFO_RD# signal.
0x01	Data	Vdata_in	In forward transactions: the data to be written into the chip
0x02	Data	I2c_data	Data to be written or read to/from the video decoder through the i^2c bus.
0x03	Data	Reserved	
0x04	Command	Video_reset	Writing to this address puts the W99200F video unit into reset condition
0x05	Command	Vstart	Writing to this address starts data transfer transactions
0x06	Command	Vstop	Writing to this address stops data transfer transactions
0x07	Command	I2c_start	Writing to this address initiates an I^2C START condition on the I^2C bus
0x08	Command	I2c_stop	Writing to this address initiates an I^2C STOP condition on the I^2C bus
0x09	Command	I2c_rack	Writing to this address initiates an ACKNOWLEDGE condition on the I^2C bus
0x0A-0x0B	Command	Reserved	
0x0C	Interrupts	Vint_enable	Enables or disables the W99200F interrupts.
0x0D	Interrupts	Vint_source	Gives information on status/problems that when enabled cause interrupts.
0x0E	Interrupts	Vint_clear	clears active interrupts
0x0F	Interrupts	Vint_rd_clear	reads interrupts status and clears FIFO ready interrupts

0x10	Configuration	Vthreshold	FIFO threshold.
0x11	Configuration	Vwork_mode	Selects the working mode and encoding format,
0x12	Configuration	Video_format	Selects the picture size and picture format.
0x13	Configuration	Venc_cntl	Selects bit rate policy, half pixel on/off, B start on/off, Open/Closed GOP, Inv. Telecine on/off, Scene change on/off
0x14	Configuration	Vframe_pattern	Selects the frame pattern ('m' and 'n' values)
0x15	Configuration	Vbit_rate_m	Selects the Encoding bit rate - most significant byte
0x16	Configuration	Vbit_rate_l	Selects the Encoding bit rate - list significant byte
0x17	Configuration	Vbv_size	Selects VBV size
0x18	Configuration	Vbv_initial	Selects VBV initial fullness.
0x19	Configuration	Vquality	Selects the constant quality.
0x1A	Configuration	Vslice_header	Selects the slice header frequency
0x1B	Configuration	Vgop_header	Selects the GOP and sequence header frequencies in MPEG1
0x1C	Configuration	Vit_hour	Initial Time Code - Hours
0x1D	Configuration	Vit_minute	Initial Time Code - Minutes
0x1E	Configuration	Vit_second	Initial Time Code - Seconds
0x1F	Configuration	Vit_frame	Initial Time Code - Frame numbers
0x20	Configuration	Vin_offset	Video decoder sync offset
0x21	Configuration	Vin_cntl	Video decoder control
0x22	Configuration	Vin_picture	Input video picture control
0x23	Configuration	Vmem_select	Selects the memory part to be written/read in write/read SDRAM and internal memories modes
0x24	Configuration	Vsize_h	In single frame encoding selects the frame horizontal size
0x25	Configuration	Vsize_v	In single frame encoding selects the frame vertical size
0x26-0x2F	Configuration		Reserved
0x30	Status	Vfifo_status	Gives the FIFO fullness status
0x31	Status	I2c_status	I^2C last transaction status
0x32	Status	Vframe_count	Counter that counts incoming frames.
0x33	Status	Vpic_status	Last encoded frame: frame type, scene change
0x34	Status	Vbv_level_m	VBV level after last encoded frame most significant byte
0x35	Status	Vbv_level_l	VBV level after last encoded frame least significant byte
0x36	Status	Venc_bit_m	Last encoded frame bit budget most significant byte
0x37	Status	Venc_bit_l	Last encoded frame bit budget least significant byte
0x38	Status	Vct_hour	Current (last frame) Time Code - Hours
0x39	Status	Vct_minute	Current (last frame) Time Code - Minutes
0x3A	Status	Vct_second	Current (last frame) Time Code - Seconds
0x3B	Status	Vct_frame	Current (last frame) Time Code - Frame numbers
0x3C	Status	Vtemp_ref_m	Last encoded frame temporal reference (msb)
0x3D	Status	Vtemp_ref_l	Last encoded frame temporal reference (lsb)
0x3E-0x3F	Status		Reserved
0x40-0x5F	Testing		Reserved
0x60	Status	IDR0	Identification Register 0
0x61	Status	IDR1	Identification Register 1
0x62	Status	IDR2	Identification Register 2
0x63	Status	GPIR	General Purpose Input Register

0x64	Configuration	GPOR	General Purpose Output Register
0x65	Configuration	GPCR	General Purpose Control Register
0x66	Configuration	PCR	Processor Control Register
0x67	Data	Audio_out	Audio Data Out Register
0x68	Configuration	AFCR0	Audio FIFO Control Register 0
0x69	Configuration	AFCR1	Audio FIFO Control Register 1
0x6A	Interrupt	AFISR	Audio FIFO Interrupt Source Register
0x6B	Interrupt	AFIRR	Audio FIFO Interrupt Clear Register
0x6C	Status	ALDR0	Audio Lost Data Register 0
0x6D	Status	ALDR1	Audio Lost Data Register 1
0x6E	Configuration	ADIVR	Audio Clock Divider Register
0x6F	Configuration	PPCR	Parallel Port Control Register
0x70-0x7F			Access to VCD decoder
0x80	Configuration	BBSAR0	Bitstream Buffer 0 Starting Address Register
0x81	Configuration	BBSAR1	Bitstream Buffer 1 Starting Address Register
0x82	Configuration	BBSR	Bitstream Buffer Size Register
0x83	Configuration	PCICR	PCI Control Register
0x84	Configuration	BBSTR0	Bitstream Buffer 0 Status Register
0x85	Configuration	BBSTR1	Bitstream Buffer 1 Status Register
0x86	Status	BBFR	Bitstream Buffer Fullness Register

Restrictions

1. ***Addresses which are specified as reserved should not be written. Writing any value to these addresses may cause unexpected results***
2. ***Some of the registers have reserved bits. Reserved bits should always be written as '0'. Writing '1' to the reserved bits is illegal and may cause unexpected results***
3. ***Some of the registers have predefined value ranges. Writing out of range values is illegal and may cause unexpected results.***
4. ***The video configuration registers (except for the Vquality and Vin_picture registers) must not be changed during data transfer: from writing "Vstart" command until the last byte was read from FIFO. Changing any configuration register during data transfer is illegal and may cause unexpected results.***

Vdata_out(0x00)

The Vdata_out register is used for reverse data transfer between the W99200F and the host. This address is the output port of the FIFO. The host should access the Vdata_out register only when the W99200F indicates (by an interrupt or by FIFO_RDY) that the data in the FIFO passed the (programmable) threshold level. Consecutive read operations from this registers results in different data.
The Vdata_out register can be 8, 16 or 32 bits according to the BW[1:0] signals.
In generic bus host type (HTS=2), reading this register is done by the FIFO_RD signal and the transactions are synchronized to FIFO_RD_CLK clock.

Vdata_in (0x01)

In forward transactions this register is used by the host to write the YCbCr picture into the W99200F.
The Vdata_in register can be 8, 16 or 32 bits according to the BW[1:0] signals.

I2c_data (0x02)

This register is used to write/read the data to be sent to / to be read from the video decoder over the I^2C bus. For a detailed explanation on the I^2C protocol see section 7.4.

Video_reset (0x04)

This is a write only register. Any data written to it puts the W99200F video unit into reset condition.

Vstart (0x05)

This is a write only register. Any data written to it starts the W99200F operation (according to the work mode) . In live video encoding, live video pass through and live video snap shot modes the first frame to be encoded (transferred) is the first incoming odd field (indicating by a combination of VIVS# and VIHS#) that begins after this register is written.

Vstop (0x06)

This is a write only register. Any data written to it indicates the W99200F to stop data transfer transactions. In live video encoding and live video pass through modes, W99200F ignores the incoming frame during which the Vstop register was written and encodes (transferred) all the previous frames..

I2c_start (0x07)

This is a write only register. Any data written to it initiates an I^2C START condition on the I^2C bus.

I2c_stop (0x08)

This is a write only register. Any data written to it initiates an I^2C STOP condition on the I^2C bus.

I2c_rack (0x09)

This is a write only register. Any data written to initiates an ACKNOWLEDGE condition on the I^2C bus.

Vint_enable (0x0C)

Reset value: 0x00
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	I2C ready	VBV underflow	Vin Out-Of-sync	Video data error	Video FIFO Overflow	Video FIFO Ready	Video End of Data
0	0	0	0	0	0	0	0

This configuration register is used to enable the different video interrupt sources of the W99200F. All interrupts are disabled by default.

- Bit 0 is the "Video end of data" bit. If this bit is set the INT# signal will be active after the host reads the last data element. End of data completes the reverse data transfer session. .
- Bit 1 is the "Video FIFO ready" bit. If this bit is set the INT# signal will be active when the number of available bytes in the FIFO passed the threshold level as defined in the Vthreshold register.
- Bit 2 is the "Video FIFO overflow" bit. If this bit is set the INT# signal will be active when an overrun error occurs (i.e., the host failed to read the FIFO before it gets full and a new data can not be written into it).
- Bit 3 is the "Video data error" bit. This bit is set in write frame mode to inform the host that the amount of data received by the W99200F is different from the expected (according to Vsize_h, Vsize_v registers).
- Bit 4 is the "Vin out of sync" bit. If this bit is set the INT# signal will be active when the video decoder gets out of sync.
- Bit 5 is the "VBV underflow" bit. If this bit is set the INT# signal will be active whenever a VBV underflow occurs. This bit can be active only in live video encoding, MPEG1 constant bit rate mode.
- Bit 6 is the "I2c ready" bit. If this bit is set the INT# signal will be active whenever last i2c transaction was ended.
- Bit 7 - reserved

Notes:

1. *More than one interrupt sources can be active together. It is the software responsibility to prioritized the interrupts.*
2. *When changing the Vint_enable register the INT# signal may change (activate, deactivate) according to register change and Vint_source value.*

Vint_source (0x0D)

Reset value: 0x40
(Read only)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	I2C ready	VBV underflow	Vin Out-Of-sync	Video data error	Video FIFO Overflow	Video FIFO Ready	Video End of Data
0	1	0	0	0	0	0	0

This register should be read by the host when it receives an interrupt in order to decide which interrupt service routine will be executed or from time to time for polling bits. Only interrupts that are enabled can generate the INT# signal, but even if an interrupt is disabled the bit is set when the related event occurs. There may be more than one interrupt sources active together, in this case the host should execute the interrupt routines according to their priority (determined by software).

- Bit 0 is the "Video end of data" bit. One in this bit indicates that the host has read the last byte in the bit stream and no more bytes are going to be produced.
- Bit 1 is the "Video FIFO ready" bit. One in this bit indicates that the number of new bytes in the FIFO passed the threshold level.
- Bit 2 is the "Video overflow" bit. One in this bit indicates that an overflow error occurs, thus the host failed to read the FIFO (Data register) before it gets full and an unread data was overwritten by a new one. When overflow error occurs the FIFO is blocked for writing .
- Bit 3 is the "Video data error" bit. One in this bit indicates that an error occurs during forward transactions in write frame mode. The number of bytes that were written to the W99200F doesn't match the expected number of bytes.
- Bit 4 is the "Vin out-of-sync" bit. This bit is on if the video decoder is out of sync.

- Bit 5 is the "VBV underflow" bit. This bit is on if a VBV underflow occurred. This bit is active only in live video encoding, MPEG1, constant bit rate.
- Bit 6 is the "I2c ready" bit. One in this bit indicates that the last I2c transaction ended and the i2c is ready for a new transaction. The host should then read the i2c status register in order to determine whether the transaction succeeded or not.

Vint_clear (0x0E)

(Write only)

This register is used to clear active interrupt sources. The structure of this register is equal to the Vint_enable register. Writing '1' to any bit in this register clears the associated event. Clearing an interrupt source by the Vint_clear register clears also the status bit associated to that interrupt (See Vint_source register) regardless whether this interrupt is enabled or not. The INT# signal is deactivated following clearing all enabled and active interrupt sources.

After clearing FIFO ready status, it is mask until the host reads the amount of threshold data elements from the Vdata_out register.

Vint_rd_clear (0x0F)

Reset value: 0x40
(Read only)

This register, exactly like the Vint_source register, gives the interrupt status with the only difference that it clears the FIFO ready status. This is done in order to save transactions between the host and the W99200F associated with the relative large amounts of FIFO ready interrupts. Pay attention that reading this register clears the FIFO ready status bit regardless whether this interrupt is enabled or not.

After clearing FIFO ready status, it is mask until the host reads the amount of threshold data elements from the Vdata_out register.

Vthreshold (0x10)

Reset value: 0x04
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	Reserved	Threshold Level [5]	Threshold Level [4]	Threshold Level [3]	Threshold Level [2]	Threshold Level [1]	Threshold Level [0]
0	0	0	0	0	1	0	0

The Vthreshold register enables the host to set the fullness level of the video FIFO. When FIFO level is above threshold:

1. The "video FIFO ready" bit in the Vint_source register is set.
2. If the "Video FIFO ready" interrupt is enabled an interrupt is issued
3. If MAST_EN signal is active the FIFO_RDY signal goes active in generic bus host type.

- Bits 5:0 - Vthreshold level in groups of 4 bus width data. Allowed values depends on BW[1:0] :
  - If BW[1:0]=0 (8 bit bus) allowed values are 0x1 - 0x3C.
  - If BW[1:0]=1 (16 bit bus) allowed values are 0x1 - 0x1E.
  - If BW[1:0]=3 (32 bit bus) allowed values are 0x1 - 0x0F

Examples:

a) BW [1:0] = 00 (8 bits) and Vthreshold=0x3C. The FIFO is ready when there are at least 240 bytes available in it

b) BW [1:0] = 01 (16 bits) and Vthreshold=0x8. The FIFO is ready when there are at least 32 words (64 bytes) available in it

c) BW [1:0] = 11 (32 bits) and Vthreshold=0x2. The FIFO is ready when there are at least 8 double-word (32 bytes) available in it

- Bit 7:6 - reserved

Vwork_mode (0x11)

Reset value: 0x00 (Live Video Encoding - MPEG1)
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	Reserved	Reserved	Encoding format	Working mode [3]	Working mode [2]	Working mode [1]	Working mode [0]
0	0	0	0	0	0	0	0

- Bits 3:0 - Working mode:
  - 0 - Live video encoding
  - 1 - Live video pass through
  - 2 - Live video snap shot
  - 3 - Single frame encoding mode
  - 4 - Write frame mode
  - 5 - Read frame mode
  - 6 - SDRAM write mode
  - 7 - SDRAM read mode
  - 8 - Internal memories write mode
  - 9 - Internal memories read mode
  - A-F - Reserved
- Bit 4 - Encoding format (in Live Video Encoding or Single Frame Encoding modes):
  - 0 - MPEG1
  - 1 - JPEG

Video Format (0x12)

Reset value: 0x05 (Square Pixel NTSC, SIF)
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]

Reserved	Reserved	Reserved	Reserved	Picture Format [1]	Picture Format [0]	Picture Size [1]	Picture Size [0]
0	0	0	0	0	1	0	1

This register selects the input and output video format

- Bits 1:0 - Picture size:
  - 0 - Full
  - 1 - SIF
  - 2 - QSIF
  - 3 - Other. Picture size is defined in Vsize_h, Vsize_v registers.
- Bits 3:2 - Picture format:
  - 0 - CCIR601 NTSC
  - 1 - Square pixel NTSC
  - 2 - CCIR601 PAL
  - 3 - Reserved

Notes:
1. *If bits Vwork_mode [3:0] == 0 ,1 (Live video encoding and pass through) then bits 1:0 should be 1 or 2 (SIF or QSIF).*
2. *If bit Vwork_mode [3:0] == 3 (Single frame encoding) then bits 3:2 define the frame rate field in the MPEG1: NTSC - 29.97 frames/second PAL - 25 frames/second*

Venc_cntl (0x13)

Reset value: 0x0C
(No time stamp, auto scene change -off, auto inverse telecine - off, open GOP, half pixel enable, 'B' start on, constant bit rate)
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Time Stamp Enable	Auto Scene Change	Auto Inverse Telecine	Closed GOP	'B' Start	Half Pixel ME search Enable	Bit Rate Policy [1]	Bit Rate Policy [0]
0	0	0	0	1	1	0	0

This register defines control parameters that are only considered in live video encoding MPEG1 format. In all other cases this register is ignored.

- Bits 1:0 - bit rate policy
  0 - Constant bit rate: The bit rate defined in Vbit_rate_m:Vbit_rate_l registers is constant. Attempt is made to prevent VBV underflow and overflow (an interrupt is set if VBV underflow happens) and stuffing is implemented in order to satisfy these conditions (See also ISO/IEC 11172-2 Annex C)
  1 - Maximum bit rate: Like constant bit rate but stuffing is not implemented. Thus, the actual bit rate may be lower than the value written in Vbit_rate_m:Vbit_rate_l registers. VBV underflow and overflow are not prevented.

2 - Variable bit rate - Bit rate is variable. Vbit_rate_m:Vbit_rate_l registers are ignored. The host selects the video quality and the bit rate depends on this selection and the incoming video.
3 - Reserved.

Note : If bit Vwork_mode[4] == 1(JPEG), bit [1:0] should be 2 (Bit rate policy - Variable).

- Bit 2: half pixel on:
  0 - Half pixel search is disabled
  1 - Half pixel search is enabled
- Bit 3: B start on:
  0 - Start an encoding session with 'I' frame
  1 - Start an encoding session with 'B' frames
  Notes:
  1. *This bit is ignored if "m" value is less than 2.*
  2. *This bit effects only the first encoded frame in a session (after start command). Other MPEG1 sequences in a session start always with 'B'-start.*
- Bit 4: Closed GOP
  0 - Use open GOP structures (thus, the GOP has been encoded with motion vectors pointing to the previous GOP)
  1 - Use closed GOP structures (thus, the GOP has been encoded without motion vectors pointing to the previous GOP)
  Note: This bit is ignored if "m" value is less than 2.
- Bit 5: Inv. Telecine on
  0 - Automatic inverse telecine detection is disabled.
  1 - Automatic inverse telecine detection is enabled. When enabled the W99200F tries to detect duplicate fields in the sequence caused by the telecine 3:2 mechanism. Upon detection the W99200F starts a new MPEG1 sequence with a frame rate of 23.976 frames/second. The W99200F keeps tracking of the 3:2 sequence and switches back to 29.75 frame rate sequence if it detects that the 3:2 pattern is not kept any more. Thus, enabling this bit may cause switching between MPEG1 sequences of 29.75 and 23.976 frames/second.
  Note: This bit is ignored if picture format is PAL.
- Bit 6: Scene change on
  0 - Automatic scene change detection is disabled.
  1 - Automatic scene change detection is enabled. When enabled the W99200F tries to detect scene changes in the sequence. In a result to a scene change the W99200F starts a new closed GOP. A new sequence header is also put unless Vgop_header[7:4] ≠ 0 .
- Bit 7: Time stamp enabled. When this bit is on the W99200F provides time stamp (the time of the actual capturing) on every encoded frame. The time-stamp is provided in the user-data section of the Picture header in MPEG1 bit-stream.

Vframe_pattern (0x14)

Reset value: 0x04 (m=1 ; n=1)
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
n/m [5]	n/m [4]	n/m [3]	n/m [2]	n/m [1]	n/m [0]	m-1 [1]	m-1 [0]
0	0	0	0	0	1	0	0

This register has two functions:

1. In Live video Encoding MPEG1 format: Selects the frame pattern (m, n values)
   - Bits 1:0- "m-1" value. Number of 'B' frames between two anchor frames ('I' or 'P').
     0: No 'B' frames
     1: One 'B' frame between two anchor frames
     2: Two 'B' frames between two anchor frames
     3: Three 'B' frames between two anchor frames
   - Bits 7:2 - "n/m" value. Distance between two 'I' frames divided by the distance between two anchor frames. For example if "m-1" equals 0x2 and "n/m" equals 0x05 then the distance between two 'I' frames is 15 and a frame pattern looks like this: BBIBBPBBPBBPBBP . 'n' should not be more than 60. Allowed values are 0x01 - 60/m

2. In Live video encoding - MJPEG and Live video pass through:
   - Bits 1:0 - Reserved
   - Bits 7:2 - The distance between two output frames. Allowed values are 0x1 - 0x3F (=63). It is the host responsibility to verify that the channel bandwidth is large enough to pass the live video sequence.

Vbit_rate_m (0x15), Vbit_rate_l (0x16)

Reset value: 0x0A00 (=1024 kbps)
(R/W)

These two registers select the output bit rate (in units of 400 bits/second) in Live video encoding MPEG1 mode. Allowed values are:

Frame size	MPGE1 'I' frame Live video encoding		MPEG1 'IBP' Live video encoding	
	Bit rate (Kbps)	Registers values (hex)	Bit rate (Kbps)	Registers values (hex)
SIF	1536 - 9216	0F00 - 5A00	192 - 6144	01E0 - 3C00
QSIF	512 - 3072	0500 - 1E00	64 - 2048	00A0 - 1400

Vbit_rate_m includes the 8 most significant bits and Vbit_rate_l includes the 8 least significant bits of the bit rate register.

Notes:
1. *The W99200F may exceed the target bit-rate, in these cases VBV underflow conditions happen.*
2. *The actual produced bit rate depends on the bit rate policy :*
   - *In case of variable bit rate policy the Vbit_rate registers are ignored.*
   - *In case of constant or maximum bit rate policies, the actual produced bit rate depends on the VBV size and it can not be more than VBV size multiplied by the number of frames per second..*

Vbv_size (0x17)

Reset value: 0x14 (= 327,680 bits).
(R/W)

The Vbv_size is an eight bit register that defines the minimum size of the VBV (Video Buffering Verifier) buffer needed to decode the MPEG1 sequence. It is defined as: B = 16384 × Vbv_size Where B is the minimum VBV buffer size in bits required to decode the sequence (See also ISO/IEC 11172-2, Annex C). Allowed values are: 0x01 - 0x7F

Note: The Vbv_size register is ignored unless all the following conditions are TRUE:
1. *Working mode is: Live video encoding*
2. *Encoding format is: MPEG1*
3. *Bit_rate policy is: constant bit rate OR maximum bit rate.*

Vbv_initial (0x18)

Reset value: 0x13
(R/W)
The Vbv_initial is an eight bit register that defines the initial fullness of the VBV (Video Buffering Verifier) buffer needed to decode the MPEG1 sequence. It is defined as: IF = 16384 × Vbv_initial Where IF is the initial fullness of the VBV buffer in bits (See also ISO/IEC 11172-2, Annex C). Allowed values are: 0x01 - 0x7F(Vbv_initial must be equal or less to Vbv_size register).

Note: The Vbv_initial register is ignored unless all the following conditions are TRUE:
1. *Working mode is: Live video encoding*
2. *Encoding format is: MPEG1*
3. *Bit_rate policy is: constant bit rate OR maximum bit rate.*

Vquality (0x19)

Reset value: 0x08
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	Reserved	Reserved	Quality [4]	Quality [3]	Quality [2]	Quality [1]	Quality [0]
0	0	0	0	1	0	0	0

This register selects the quality of the encoded M-JPEG bit stream and the encoded MPEG1 bit stream if the bit rate policy is variable.

- Bits 4:0 - MPEG1 encoding video quality:
  - 0x00 - Reserved
  - 0x01 - Highest quality
  - 0x1F - Lowest quality
- Bits 7:5 - Reserved.

Notes:
1. *The Vquality register is valid if one of the following conditions is TRUE, otherwise it is ignored:*
   - *Working mode is: Single frame encoding.*
   - *Working mode is: Live video encoding AND Encoding format is: M-JPEG .*
   - *Working mode is: Live video encoding AND Encoding format is: MPEG1 AND Bit_rate policy is: variable.*

2. *When working in "constant quality policy" the produced bit rate is variable and depends on the encoded video. It is the software responsibility to ensure that the produced bit rate will not exceed the bus bandwidth.*
3. *The Vquality register (together with the Vin_picture register) can be changed during data transfer. When working in M-JPEG or constant Quality (variable bit rate) MPEG1, the software should check the FIFO status from time to time. In case that the FIFO is getting full the software can change the quality value to prevent overflow condition. Changing the Vquality is taking place only in frame boundaries.*

Vslice_header (+ Repeat_B) (0x1A)

Reset value: 0x01 (Slice Header every row, Repeat 'B' frame - off)
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Repeat 'B' frames	Reserved	Slice Header [5]	Slice Header [4]	Slice Header [3]	Slice Header [2]	Slice Header [1]	Slice Header [0]
0	0	0	0	0	0	0	1

This six+1 bits register defines the slice header (slice_start_code, quantizer_scale) frequency in the MPEG1 bit stream. Slice header frequency is given in rows of MBs. Allowed values are from 1 to the number of MB rows in the picture according to the picture format and size. The Repeat_B is a bit economy feature. It allows repeating of B-Frames instead of full encoding and by this trading-off bits for actual new frames per second. The number of repeated frames equals to the B-frames number (i.e., m-1)

- Bits 5:0 - Slice header frequency
- Bit 7 - Repeat B frames.
  0 : Normal encoding
  1 : Repeat B-frames, no actual B-frames encoding.

Vgop_header (0x1B)

Reset value: 0x01 (GOP Header every 'I' frame, No repeating sequence Headers)
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Sequence Header [3]	Sequence Header [2]	Sequence Header [1]	Sequence Header [0]	GOP Header [3]	GOP Header [2]	GOP Header [1]	GOP Header [0]
0	0	0	0	0	0	0	1

This eight bit register defines the GOP header (including the fields: group_start_code, time_code, closed_gop, broken_link) and Sequence header frequencies in the MPEG1 bit stream.

- Bits 3:0 - GOP header frequency
  The GOP header frequency is defined as the distance in 'I' frames between two GOP headers (e.g., if Vgop_header = 0x04 it means that GOP header appears every four 'I' frames). Allowed values are: 0x1-0xF
- Bits 7:4 - Sequence header frequency
  The Sequence header frequency is defined as the distance in GOP headers between two sequence headers (e.g., if Seq_header = 0x04 it means that sequence header appears every four GOP headers frames). Allowed values are: 0x0-0xF
  The value 0x0 means that only necessary sequence headers appears in the bit stream. (i.e., only in new sequence start)

Notes:
1. *Activating the inverse telecine or the scene change options may change temporally the distance between GOP headers in the bit stream.*
2. *Activating the inverse telecine or the scene change options may change temporally the distance between sequence headers in the bit stream.*

Vit_hour (0x1C)

Reset value: 0x00
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	Reserved	Reserved	Hours [4]	Hours [3]	Hours [2]	Hours [1]	Hours [0]
0	0	0	0	0	0	0	0

This register is used to initiate the value to the time code - hours field. The register should be written before encoding starts. This register is one out of four time code registers.

- bits 4:0 - Time code hours - Allowed values 0x00 - 0x17 (=23)
- bits 7:5 - reserved

Vit_minute (0x1D)

Reset value: 0x00
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	Reserved	Minutes [5]	Minutes [4]	Minutes [3]	Minutes [2]	Minutes [1]	Minutes [0]
0	0	0	0	0	0	0	0

This register is used to initiate the value to the time code - minutes field. The register should be written before encoding starts. This register is one out of four time code registers.

- bits 5:0 - Time code minutes - Allowed values 0x00 - 0x3B (=59).
- bits 7:6 - reserved

Vit_second (0x1E)

Reset value: 0x00
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	Reserved	Seconds [5]	Seconds [4]	Seconds [3]	Seconds [2]	Seconds [1]	Seconds [0]
0	0	0	0	0	0	0	0

This register is used to initiate the value to the time code - seconds field. The register should be written before encoding starts. This register is one out of four time code registers.

- bits 5:0 - Time code seconds - Allowed values are 0x00 - 0x3B (=59).
- bits 7:6 - reserved

Vit_frame (0x1F)

Reset value: 0x00
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	Reserved	Reserved	Frames [4]	Frames [3]	Frames [2]	Frames [1]	Frames [0]
0	0	0	0	0	0	0	0

This register is used to initiate the value to the time code - frame number field. The register should be written before encoding starts. This register is one out of four time code registers. Frame numbering is from 0-29 in a frame rate of 29.97 frames/sec ,0-24 in a frame rate of 25 frames/second and 0-23 in a frame rate of 23.976 frames/second.

In case of 29.97 frames/sec encoding the W99200F implements the drop frame mechanism in order to compensate on the differences between the source frame rate of 29.97 frames/second and the time code rate of 30 frames/second. In this mechanism frame numbers 0 and 1 at the start of each minute, except minutes 0, 10 ,20, 30, 40 and 50 are omitted from the count (e.g., time code is changed from 01:00:59:29 to 01:01:00:02).

- bits 4:0 - Time code frame numbers - Allowed values as written above.
- bits 7:5 - reserved

Vin_offset (0x20)

Reset value: 0x00
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
EFO [3]	EFO [2]	EFO [1]	EFO [0]	OFO [3]	OFO [2]	OFO [1]	OFO [0]

0	0	0	0	0	0	0	0

- Bits 3:0 - Odd Field Offset (OFO) :
  Signed integer (-8..0) indicates the offset (in lines) of odd field indication, from the default line location (default line location: line 7 in PAL, line 10 in NTSC). In PAL, the minimum value is -6.
- Bits 7:4 - Even Field Offset (EFO) :
  Signed integer (-8..0) for NTSC and (-7..0) for PAL indicates the offset (in lines) from the default line location of even field indication (default line location: line 320 in PAL, line 273 in NTSC).

Note: Maximum difference between Odd and Even offsets is :± 4. ABS(Odd_Offset - Even_Offset) £ 4.

Vin_cntl (0x21)

Reset value: 0x0C
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	VIVS# input select	SCR generation clock	VIHS# sampled level	VIDV polarity	VIHACT polarity	VIVS# polarity	VIHS# polarity
0	0	0	0	1	1	0	0

- Bit 0 - VIHS# polarity:
  0: Active low. Falling edge is used to indicate a new video line (default).
  1: Active high. Rising edge is used to indicate a new video line.
- Bit 1 - VIVS# polarity:
  0: If VIVS# is VS (vin_cntl[6]==0) - Falling edge marks the beginning of a new field.

  else (VIVS# if FID) (vin_cntl[6]==1) - Falling edge marks the beginning of an odd field.

  1: If VIVS# is VS (vin_cntl[6]==0) - Rising edge is used to indicate a new video field

  else (VIVS# if FID) (vin_cntl[6]==1) - Rising edge marks the beginning of an odd field.
- Bit 2 - VIHACT polarity:
  0: Active low. When low - indicates the active region in a line
  1: Active high. When high - indicates the active region in a line (default)
- Bit 3 - VIDV polarity:
  0: Active low. When low - indicates a valid pixel to be sampled
  1: Active high. When high - indicates a valid pixel to be sampled (default).
- Bit 4 - VIHS# sampled level
  0: (default): Odd field is indicated by VIHS# high sampled with VIVS#, Even field is indicated by VIHS# low sampled with VIVS#.
  1: Odd field is indicated by VIHS# low sampled with VIVS#, Even field is indicated by VIHS# high sampled with VIVS#.
- Bit 5 - SCR generation clock

  0: viclk . Specifically supports Phillips like video decoders. PTS is genlocked to Video HSYNC.

1: iclk . Suitable for Brooktree like video decoders. PTS is independent of the Video source.

- Bit 6 - VIVS# input select

  0: (default) VIVS# input is vertical sync signal

  1: VIVS# input is field ID signal (FID)

Vin_picture (0x22)

Reset value: 0x00
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Bitmap insertion	Blanked video lines
0	0	0	0	0	0	0	0

- Bit 0 - Blanked Video Lines
  0 :(default): No Video lines are blanked.
  1: Last 16 rows in FULL resolution, last 8 rows in SIF resolution or last 4 rows in QSIF resolution are blanked
- Bit 1 - Bit-map insertion is on.
  0 (default): No bit-map insertion
  1: A pre-defined bit map is combined with the input video.

Notes:

1) This register is only valid in Live Video Encoding, Live Video pass-through and Live Video snap shot modes.
2) Bit 1 (bit map insertion) can be changed during LVE and LVPT modes in order to enable switching on/off the bit-map insertion.. Bit 0 of this register should not be changed after the configuration stage. It is the host responsibility to insure that only bit 1 is changed (e.g., by read modify write command). Changing the bit takes effect only in frame boundaries

Vmem_select (0x23)

Reset value: 0x00
(R/W)

This register has two functions:

1. In Write/Read to/from SDRAM it selects the SDRAM part to be written or read. The SDRAM is divided into 256 pages 4K x 16 bit each. Allowed values are 0x00 - 0xFF.
2. In Write/Read internal memories mode this register selects the memory or internal registers to be written/read in data chunks of 256 bytes.

Vsize_h (0x24)

Reset value: 0x2C
(R/W)

In Write frame, Read frame and Single frame encoding modes this register gives the horizontal size of the frame to be encoded in units of 16 pixels. Available numbers are from 0x02 - 0x2D (=45).

In other working modes this register is ignored

Vsize_v (0x25)

Reset value: 0x1E
(R/W)

In Write frame, Read frame and Single frame encoding modes this register gives the vertical size of the frame to be encoded in units of 16 pixels. Available numbers are from 0x02 - 0x24 (=36).

In other working modes this register is ignored

Vfifo_status (0x30)

Reset value: 0x00
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	Reserved	Reserved	Reserved	Full [3]	Full [2]	Full [1]	Full [0]
0	0	0	0	0	0	0	0

- Bits 3:0 are FIFO fullness indication. These bits are valid only in live video encoding mode
  0x0 - FIFO is less than 1/16 full.
  0x1 - FIFO is between 1/16 to 2/16 full
  0x2 - FIFO is between 2/16 to 3/16 full
  ...
  0xE - FIFO is between 14/16 to 15/16 full
  0xF - FIFO is more than 15/16 full
- Bits 7:4 are reserved

I2c_status (0x31)

Reset value: 0x00
(R/W)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	I2C Succeeded

0	0	0	0	0	0	0	0

- Bit 0 - i2c_succeeded:
  0 - I2c last transaction not succeeded.
  1 - I2c last transaction succeeded. In read operation the data is ready in the i2c_data register.
- Bits 7:1 - Reserved

Vframe_count (0x32)

Reset value: 0x00
(Read only)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Valid status data	Frame Count [6]	Frame Count [5]	Frame Count [4]	Frame Count [3]	Frame Count [2]	Frame Count [1]	Frame Count [0]
0	0	0	0	0	0	0	0

This register counts incoming frames and uses as semaphore for other status registers. It has two fields:
- Bits 6:0 - Frame count. This field is incremented every incoming frame. It counts from 0x00 to 0x7F and then starts again.
- Bit 7 - Status registers are valid. - When this bit is on it means that the status registers: Vpic_status, VBV_level, Venc_bit, Vct_hour, Vct_minute, Vct_second, Vct_frame and Vtemp_ref are synchronized to the same frame. Otherwise they might have information from mixed frames. This bit is only valid in Live Video encoding MPEG1.

Note: This register is valid only in Live Video Encoding and Live Video Pass Through modes.

Vpic_status (0x33)

Reset value: 0x00
(Read only)

[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
Reserved	Reserved	Reserved	Reserved	New Sequence	New Scene	Frame Type [1]	Frame Type [0]
0	0	0	0	0	0	0	0

This register includes information on the last encoded frame (Valid only in Live video encoding MPEG1).
- Bits 1:0 - Last encoded frame type:
  0x0 - Last encoded frame was not coded frame.
  0x1 - Last encoded frame was 'I' frame
  0x2 - Last encoded frame was 'P' frame
  0x3 - Last encoded frame was 'B' frame

- Bit 2 - Last encoded frame was the first encoded frame in a new scene. .

- Bit 3 - Last encoded frame was the first encoded frame in a sequence which has different frame rate from the previous one (switching in and out from 29.75 to 23.97 frames/second)
- Bits 7:4 Reserved.

Vbv_level_m (0x34), Vbv_level_l (0x35)

Reset value: 0x0000
(Read only)

These registers provide the VBV value after the last encoded frame. The value is given in units of 64 bits truncated toward zero.
The contents of the register is valid only in live encoding mode MPEG1 format - constant bit rate

Venc_bit_m (0x36), Venc_bit_l (0x37)

Reset value: 0x0000
(Read only)

These registers provide the number of bits that the last frame was encoded in. It is given in units of 16 bits truncated toward zero.
The contents of the register is valid only in live encoding mode MPEG1 - constant bit rate

Vct_hour (0x38)

Reset value: 0x00
(Read only)

This register provides the time code (hours field) of the last encoded gop. Its structure is identical to Vit_hour. This register is updated once every GOP .
The content of the register is valid in live encoding MPEG1 mode.

Vct_minute (0x39)

Reset value: 0x00
(Read only)

This register provides the time code (minutes field) of the last encoded GOP. Its structure is identical to Vit_minute. This register is updated once every GOP.
The content of the register is valid in live encoding MPEG1 mode.

Vct_second (0x3A)

Reset value: 0x00
(Read only)

This register provides the time code (seconds field) of the last encoded GOP. Its structure is identical to Vit_second. This register is updated once every GOP .
The content of the register is valid in live encoding MPEG1 mode.

Vct_frame (0x3B)

Reset value: 0x00
(Read only)

This register provides the time code (frames field) of the last encoded GOP. Its structure is identical to Vit_frame This register is updated once every GOP.
The content of the register is valid in live encoding MPEG1 mode.

Vtemp_ref_m (0x3C), Vtemp_ref_l (0x3D)

Reset value: 0x00
(Read only)

This 10bit register include the last encoded frame temporal reference relative to the Time code.

Register 0x6F is only used in Parallel port.
Registers 0x80 ~ 0x86 are only used in PCI host.

Identification Register 0 (IDR0)

index: 0x60
reset value: 0x00
(Read only)

bit(7:0)
device ID
00h

Identification Register 1 (IDR1)

index: 0x61
reset value: 0x92
(Read only)

bit(7:0)
device ID
92h

Identification Register 2 (IDR2)

index: 0x62
reset value: 0x09
(Read only)

bit(7:4)	bit(3:0)
revision ID	device ID
0h	9h

IDR2, IDR1 and IDR0 contain the revision ID and device ID of W99200F for system identification. The revision ID number of this version is 0h and device ID number is 99200h.

General Purpose Input Register (GPIR)

index: 0x63
reset value: 0x00
(Read only)

bit(7:4)	bit(3)	bit(2)	bit(1)	bit(0)
not used	GPI3	GPI2	GPI1	GPI0
0h	0	0	0	0

This register provides general purpose I/O signals for board-level control, hardware reset only.
GPI0: read only, used to reflect the immediate state of GP0 pin.
GPI1: read only, used to reflect the immediate state of GP1 pin.
GPI2: read only, used to reflect the immediate state of GP2 pin.
GPI3: read only, used to reflect the immediate state of GP3 pin.

General Purpose Output Register (GPOR)

index: 0x64
reset value: 0x00
(R/W)

bit(7:4)	bit(3)	bit(2)	bit(1)	bit(0)
not used	GPO3	GPO2	GPO1	GPO0
0h	0	0	0	0

This register can only be reseted by hardware reset.

GPO0: read-/write-able, connected to the output of GP0 pin.
GPO1: read-/write-able, connected to the output of GP1 pin.
GPO2: read-/write-able, connected to the output of GP2 pin.
GPO3: read-/write-able, connected to the output of GP3 pin.

General Purpose Control Register (GPCR)

index: 0x65
reset value: 0x00
(R/W)

bit(7:4)	bit(3)	bit(2)	bit(1)	bit(0)
not used	GPC3	GPC2	GPC1	GPC0
0h	0	0	0	0

This register can only be reseted by hardware reset.

GPC0: read-/write-able, used to determine the I/O state of GP0 pin. GPC0=0, GP0 pin is in input state, while GPC0=1, GP0 pin is in output state.
GPC1: read-/write-able, used to determine the I/O state of GP1 pin. GPC1=0, GP1 pin is in input state, while GPC1=1, GP1 pin is in output state.
GPC2: read-/write-able, used to determine the I/O state of GP2 pin. GPC2=0, GP2 pin is in input state, while GPC2=1, GP2 pin is in output state.
GPC3: read-/write-able, used to determine the I/O state of GP3 pin. GPC3=0, GP3 pin is in input state, while GPC3=1, GP3 pin is in output state.

Processor Control Register (PCR)

index: 0x66
reset value: 0x64
(R/W)

bit(7)	bit(6)	bit(5)	bit(4:3)	bit(2)	bit(1)	bit(0)
SDRAM_TRI	DE_TRI1	DE_TRI0	(not used)	BY_PASS	PD	SRST
0	1	1	0	1	0	0

This register can only be reseted by hardware reset.

SDRAM_TRI is used to tri-state W99200F SDRAM interface signals.
SDRAM_TRI=0, disables the tri-state function of SDRAM interface signals.
SDRAM_TRI=1, enables the tri-state function of SDRAM interface signals

DE_TRI1 and DE_TRI0 are used to tri-state VCD decoder output signals in order to release the control of SDRAM.
While DE_TRI1 is active, W99200F will assert DEC_RESET# active.
DE_TRI1=0, disables to force DEC_RESET# active.
DE_TRI1=1, enables to force DEC_RESET# active.

While DE_TRI0 is active, W99200F will assert DEC_WR# and DEC_RD# active.
DE_TRI0=0, disables to force DEC_WR# and DEC_RD# active.
DE_TRI0=1, enables to force DEC_WR# and DEC_RD# active.

BY_PASS is used to control the source of main clock, hardware reset only.
BY_PASS =0, means the internal operating clock is derived from PLL output.
BY_PASS =1, means the internal operating clock is derived from the input MCLK directly.

PD is used to control power down mode of W99200F for power saving, hardware reset only.
PD=0, disable power down mode,
PD=1, enables power down mode to block encoder operating clock.

SRST, total software reset, is used to reset whole chip.
SRST=0, disables software reset,
SRST=1, enables software reset.

Audio Data Out Register (Audio_out)

index: 0x67
reset value: 0x00
(Read only)

The Audio_out register is used to transfer audio data to the host. This address is the output port of the audio FIFO. The host should access the Audio_out register only when the W99200F indicates (by an interrupt or by polling) that the data in the audio FIFO passed the audio threshold level.
The Audio_out register can be 8, 16 or 32 bits according to the BW[1:0] signals.
In generic bus host type (HTS=2), reading the register is done by the RD# signal.

Audio FIFO Control Register 0 (AFCR0)

index: 0x68

reset value: 0x00
(R/W)

bit(7)	bit(6)	bit(5)	bit(4)	bit(3)	bit(2)	bit(1:0)
OSYNC_EN	AIN_RST	FIFO_RESET	AF_OV_EN	AF_EOD_EN	AF_R_EN	A_THRES(1:0)
0	0	0	0	0	0	0

This register can only be reseted by hardware reset and total software reset.

A_THRES[1:0] are used to set the threshold level of audio FIFO. When audio FIFO level is above threshold:

1. The "AF_R" bit in the "audio interrupt source" register will be set "1".
2. If the "AF_R_EN" bit is enabled, an interrupt is issued.

A_THRES[1:0]
=00, the threshold level is 128 bytes,
=01, the threshold level is 256 bytes,
=10, the threshold level is 384 bytes,
=11, the threshold level is 508 bytes,

Note: A_THRES[1:0]=11, is only used in the EX_FIFO=1 mode.

AF_R_EN is used to enable the interrupt while AF_R (audio FIFO data ready) is active.
AF_R_EN=0, disable,
AF_R_EN=1, enable.

AF_EOD_EN is used to enable the interrupt while AF_EOD (audio FIFO end of data) is active.
AF_EOD_EN=0, disable,
AF_EOD_EN=1, enable.

AF_OV_EN is used to enable the interrupt while AF_OV (audio FIFO overflow) is active.
AF_OV_EN =0, disable,
AF_OV_EN =1, enable.

FIFO_RESET is used to reset internal and external FIFO content. While FIFO_RESET is active, the output pin FIFO_RST# is active too. FIFO_RESET is active high.
Note that FIFO_RESET don't affect ALDR1 and ALDR0.

AIN_RST (AIN software reset) is used to reset AIN unit. AIN_RST will reset all registers, state machine and FIFO in AIN unit, except AFCR0 and AFCR1. AIN_RST is active high.

OSYNC_EN is used to enable the function that AIN unit will stop to latch the audio input data while OSYNC# is active. It is used for A/V synchronization.
OSYNC_EN=0, AIN is not controlled by OSYNC#.
OSYNC_EN=1, enable the function.

Audio FIFO Control Register 1 (AFCR1)

index: 0x69
reset value: 0x00
(R/W)

bit(7:5)	bit(4)	bit(3)	bit(2)	bit(1:0)
reserved	A_START	A_STOP	EX_FIFO	EX_FIFO_TYPE(1:0)
0h	0	0	0	0

This register can only be reseted by hardware reset and total software reset.

A_START is audio "start" command. Writing "1" into this bit will start AIN operation.
This bit is "write only". While the host reads this bit, "0" is returned in any time.

A_STOP is audio "stop" command. Writing "1" into this bit, AIN will stop to get audio data into FIFO. The remainder in audio FIFO is still valid. The host should read all data in audio FIFO.
This bit is "write only". While the host reads this bit, "0" is returned in any time.

EX_FIFO is used to indicate whether there is an external FIFO or not.
EX_FIFO=0, without external FIFO,
EX_FIFO=1, with external FIFO.

EX_FIFO_TYPE(1:0) are used to indicate the type of external FIFO.
=00, external FIFO is 4K x8.
=01, external FIFO is 16K x8.
=10, external FIFO is 64K x8.
=11, external FIFO is 256K x8.

Note : The user should not write "1" into bit [7:5].

Audio FIFO Interrupt Source Register (AFISR)

index: 0x6A
reset value: 0x0
(Read only)

bit(7:3)	bit(2)	bit(1)	bit(0)
not used	AF_OV	AF_EOD	AF_R
0	0	0	0

AF_R is "audio FIFO data ready". AF_R=1 indicates that the number of data in the audio FIFO exceeds the threshold level.

AF_EOD is "audio FIFO end of data". AF_EOD=1 indicates that the host has read the last byte in the audio bitstream and no more audio data is going to be produced.

AF_OV is "audio FIFO overflow". AF_OV=1 indicates that audio FIFO is overflow. While the number of lost audio data is more than 1K bytes, AF_OV will be active and AIN stops to write audio data into FIFO.
While EX_FIFO=0, AF_OV=1 means that internal FIFO is overflow.
While EX_FIFO=1, AF_OV=1 means that external FIFO is overflow.

Audio FIFO Interrupt Clear Register (AFIRR)

index: 0x6B
reset value: 0x0
(Write only)

bit(7:3)	bit(2)	bit(1)	bit(0)

not used	AF_OV_CLR	AF_EOD_CLR	AF_R_CLR
0	0	0	0

AF_R_CLR is used to clear active AF_R interrupt source. Writing "1" to this bit clears interrupt request and AF_R status regardless whether AF_R_EN is enabled or not. After the host clears AF_R, AF_R is masked until the host reads the amount of audio threshold data from Audio_out register.

AF_EOD_CLR is used to clear active AF_EOD interrupt source. Writing "1" to this bit clears interrupt request and AF_EOD status regardless whether AF_EOD_EN is enabled or not.

AF_OV_CLR is used to clear active AF_OV interrupt source. Writing "1" to this bit clears interrupt request and AF_OV status regardless whether AF_OV_EN is enabled or not.

Audio Lost Data Register0 (ALDR0)

index: 0x6C
reset value: 0x0
(Read only)

bit(7:0)
ALD[7:0]
0

Audio Lost Data Register1 (ALDR1)

index: 0x6D
reset value: 0x0
(Read only)

bit(7:0)
ALD[15:8]
0

ALDR1 and ALDR0 can only be reseted by hardware reset and total software reset.

ALDR1 and ALDR0 are used to record the number of lost audio data after audio FIFO is overflow. SW driver can read these two registers and do A/V synchronization. The unit of ALD is in 1K bytes. For example, if ALD[15:0]=2, that means the number of lost audio data is in the range: 2K bytes ~ 3K bytes.

Actually, we have 24-bits counter in AIN. The unit of counter is in DWORD of audio data. When the host reads ALDR0, AIN will latch the maximal significant 16 bits of the counter into ALD[15:0] and simultaneously clear the counter and then start to count from value "0". **So the host should first read ALDR0 and then read ALDR1.**

Audio Clock Divider Register (ADIVR)

index: 0x6E
reset value: 0xF9
(R/W)

bit(7:3)	bit(2:0)
ADIV1[4:0]	ADIV0[2:0]
1Fh	1h

This register can only be reseted by hardware reset and total software reset.

This register is used to indicate the frequency factor between ACLK, PCMCLK and PCMWS.
ADIV1[4:0] is used to indicate the frequency factor between PCMCLK and PCMWS.

(The frequency of PCMCLK) / (The frequency of PCMWS) = 2 x (ADIV1[4:0] +1).

ADIV0[2:0] is used to indicate the frequency factor between ACLK and PCMCLK.

(The frequency of ACLK) / (The frequency of PCMCLK) = 2 x (ADIV0[2:0] +1).

PCMWS is fixed at 44.1KHz. That means the default frequency of PCMCLK is 2.8224MHz and ACLK should be 11.2896MHz.

Parallel Port Control Register (PPCR)

index: 0x6F
reset value: 0x03
(R/W)

bit(7:2)	bit(1:0)
not used	MPW[1:0]
0	3

This register is only used in parallel port mode. Hardware reset only.
To control the glitch filter of parallel port interface. To prevent from noise glitch make the interface fail. The parallel port interface signals' pulse width shorter than the "minimum pulse width", will be filter out as noise glitch.
These three bits define the "minimum pulse width", hardware reset only.

MPW[1:0]= 0, no filtering.
1, minimum pulse width = 1 x mclk period.
2, minimum pulse width = 2 x mclk period.
3, minimum pulse width = 3 x mclk period.
(1x mclk period = 18.5 ns)

Bitstream Buffer 0 Starting Address Register (BBSAR0)

index: 0x80
reset value: 0
(R/W)

Bit(31:0)
Bitstream Buffer 0 Starting Address
0

This register is only used in PCI master mode to record the starting address of Bitstream Buffer 0. This Bitstream Buffer is allocated on system main memory to store the encoded bitstream data. While the data in the Vdata_out FIFO pass the threshold level and the data in Bitstream Buffer 1 is full, W99200F will get the bitstream data from Vdata_out FIFO and write to Bitstream Buffer 0 which starting from the specified address using master memory write command. This register must be set before the master mode enable bit (MAST_EN) is enabled and the setting value can not be changed during operation. The default value is 0 after hardware or software reset.

Bitstream Buffer 1 Starting Address Register (BBSAR1)

index: 0x81
reset value: 0
(R/W)

Bit(31:0)
Bitstream Buffer 1 Starting Address
0

This register is only used in PCI master mode to record the starting address of Bitstream Buffer 1. This Bitstream Buffer is allocated on system main memory to store the encoded bitstream data. While the data in the Vdata_out FIFO pass the threshold level and the data in Bitstream Buffer 0 is full, W99200F will get the bitstream data from Vdata_out FIFO and write to Bitstream Buffer 1 which starting from the specified address using master memory write command. This register must be set before the master mode enable bit (MAST_EN) is enabled and the setting value can not be changed during operation. The default value is 0 after hardware or software reset.

Bitstream Buffer Size Register (BBSR)

index: 0x82
reset value: 0xFFFFFFFF
(R/W)

bit(31:16)	bit(15:0)
Bitstream Buffer 1 Size	Bitstream Buffer 0 Size
FFFFh	FFFFh

This register is only used in PCI master mode.

Bitstream Buffer Size is used to record the size of Bitstream Buffer. They are kept at 64k DWORD after hardware or software reset. The unit of the buffer size is DWORD and it must be multiple of threshold level DWORD.

PCI Control Register (PCICR)

index: 0x83
reset value: 0x0
(R/W)

bit(31:9)	bit(8:6)	bit(5:4)	bit(3:2)	bit(1)	bit(0)
not used	reserved	ABS_ORDER	BS_ORDER	MASTER_EN	BB_INT_EN
0h	0	0	0	0	0

Bit[1:0] are only used for PCI master mode.

BB_INT_EN is used to enable interrupt request while BB_FULL0 or BB_FULL1 is active. After hardware or software reset, it keeps at "0".
BB_INT_EN=1, enable interrupt request,
BB_INT_EN=0, disable interrupt request.

MASTER_EN is used to enable or disable the PCI master mode for Vdata_out fifo transfer.
Only useful in PCI host. Will be reset by hardware reset and software reset.
MASTER_EN=0 disables the capability of PCI bus master mode for Vdata_out fifo transfer,
MASTER_EN=1 enables the capability of PCI bus master mode for Vdata_out fifo transfer.

BS_ORDER[1:0] is used to specify the byte order of bitstream fifo output bitstream.
Only useful in PCI host. Will be reset by hardware reset and software reset.

BS_ORDER=00, the byte order of bitstream fifo output data will be as following:

PCI_AD(31:24)	PCI_AD(23:16)	PCI_AD(15:8)	PCI_AD(7:0)
FIFO_data byte3	FIFO_data byte2	FIFO_data byte1	FIFO_data byte0

BS_ORDER=01, the byte order of bitstream fifo output data will be as following:

PCI_AD(31:24)	PCI_AD(23:16)	PCI_AD(15:8)	PCI_AD(7:0)
FIFO_data byte2	FIFO_data byte3	FIFO_data byte0	FIFO_data byte1

BS_ORDER=10, the byte order of bitstream fifo output data will be as following:

PCI_AD(31:24)	PCI_AD(23:16)	PCI_AD(15:8)	PCI_AD(7:0)
FIFO_data byte0	FIFO_data byte1	FIFO_data byte2	FIFO_data byte3

BS_ORDER=11, the byte order of bitstream fifo output data will be as following:

PCI_AD(31:24)	PCI_AD(23:16)	PCI_AD(15:8)	PCI_AD(7:0)
FIFO_data byte1	FIFO_data byte0	FIFO_data byte3	FIFO_data byte2

bit(31:2)	bit(1:0)
not used	ABS_ORDER
0h	0h

ABS_ORDER[1:0] is used to specify the byte order of audio fifo output bitstream.
Only useful in PCI host. Will be reset by hardware reset and software reset.

ABS_ORDER=00, the byte order of audio fifo output data will be as following:

PCI_AD(31:24)	PCI_AD(23:16)	PCI_AD(15:8)	PCI_AD(7:0)
FIFO_data byte3	FIFO_data byte2	FIFO_data byte1	FIFO_data byte0

ABS_ORDER=01, the byte order of audio fifo output data will be as following:

PCI_AD(31:24)	PCI_AD(23:16)	PCI_AD(15:8)	PCI_AD(7:0)
FIFO_data byte2	FIFO_data byte3	FIFO_data byte0	FIFO_data byte1

ABS_ORDER=10, the byte order of audio fifo output data will be as following:

PCI_AD(31:24)	PCI_AD(23:16)	PCI_AD(15:8)	PCI_AD(7:0)
FIFO_data byte0	FIFO_data byte1	FIFO_data byte2	FIFO_data byte3

ABS_ORDER=11, the byte order of audio fifo output data will be as following:

PCI_AD(31:24)	PCI_AD(23:16)	PCI_AD(15:8)	PCI_AD(7:0)
FIFO_data byte1	FIFO_data byte0	FIFO_data byte3	FIFO_data byte2

Note : The user should not write "1" into bit[8:6].

Bitstream Buffer0 Status Register (BSTR0)

index: 0x84
reset value: 0x0
(R/W)

bit(31:2)	bit(1)	bit(0)
not used	BB_FULL0	BB_RDY0
0h	0	0

BB_RDY0=1 means that the Bitstream Buffer 0 is ready for W99200F writing bitstream data on it. While the data in Bitstream Buffer 0 is full, W99200F will simultaneously set BB_FULL0 "1" and BB_RDY0 "0". After hardware or software reset, it keeps at "0".

BB_FULL0=1 means that the data in Bitstream Buffer 0 is full. While BB_INT_EN is active and BB_FULL0 is "high", W99200F will assert INT# to notify host and host must move out the data in Bitstream Buffer 0. If BB_INT_EN is not active, host must poll BB_FULL0 value. The hosts should writes BB_FULL0 "1" to clear this bit. After hardware or software reset, it keeps at 0.

Bitstream Buffer1 Status Register (BSTR1)

index: 0x85
reset value: 0x0
(R/W)

bit(31:2)	bit(1)	bit(0)
not used	BB_FULL1	BB_RDY1
0h	0	0

BB_RDY1=1 means that the Bitstream Buffer 1 is ready for W99200F writing bitstream data on it. While the data in Bitstream Buffer 1 is full, W99200F will simultaneously set BB_FULL1 "1" and BB_RDY1 "0". After hardware or software reset, it keeps at "0".

BB_FULL1=1 means that the data in Bitstream Buffer 1 is full. While BB_INT_EN is active and BB_FULL1 is "high", W99200F will assert INT# to notify host and host must move out the data in Bitstream Buffer 1. If BB_INT_EN is not active, host must poll BB_FULL1 value. The host should writes BB_FULL1 "1" to clear this bit. After hardware or software reset, it keeps at "0".

Bitstream Buffer Fullness Register (BBFR)

index: 0x86
reset value: 0x0
(Read)

bit(31:17)	bit(16:0)
not used	Bitstream Buffer Fullness
0h	0h

This register is only used in PCI master mode.
It is used to record the fullness of Bitstream Buffer used currently. The unit of the fullness size is DWORD. The value range of this register is from 0 to 64K DWORD.

9. Electrical Characteristics

9.1 Absolute Maximum Rating

Symbol	Parameter	Min	Max	Unit
Vdd-Vss	DC power supply	3.0	3.6	V
Vin	Input Voltage	Vss-0.3	5.5	V
Tj	Operating Temperature	0	70	° C
Ta	Ambient Temperature		43	° C
Tst	Storage Temperature	-55	+150	° C

Table 9-1: Absolute Maximum Rating

9.2 DC Characteristics

Vdd-Vss=3.3V±10% , Ta=25° C, MCLK=54MHz

Symbol	Parameter	Test Condition	Min	Typ	Max	Unit
Vdd	Operating Voltage		3.0	3.3	3.6	V
Idd	Operating Current			275		mA
Ipwdn	Power Down Current	Power Down		13		mA
Ilk	Input Leakage		-10		+10	uA
Vol	Output Voltage Low				0.4	V
Voh	Output Voltage High		2.4			V
Vih	Input Voltage High		2		5.5	V
Vil	Input Voltage Low		-0.3		0.8	V

Table 9-2: DC Characteristics

9.3 AC Characteristics

9.3.1 System Clock Input Waveform

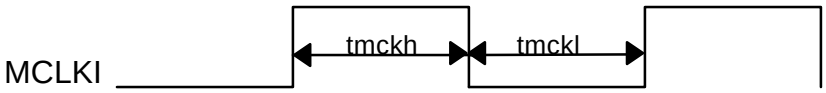


Figure 9-1: System Clock Input Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
tmclk	Main Clock period	35	37		ns	
tmckh	Main clock high time	15	18.5		ns	
tmckl	Main clock low time	15	18.5		ns	

Table 9-3: System Clock Input Timing

9.3.2 Host Bus Interface

9.3.2.1 PCI Signal Timing

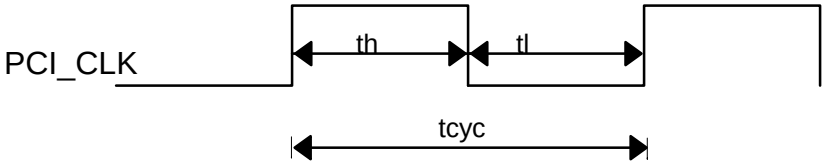


Figure 9-2: PCI Clock Input Timing

Symbol	Parameter	Min	Typ	Max	Unit	Notes
tcyc	PCI clock cycle time	30		∞	ns	
th	PCI clock high time	11			ns	
tl	PCI clock low time	11			ns	

Table 9-4: PCI Clock Input Timing

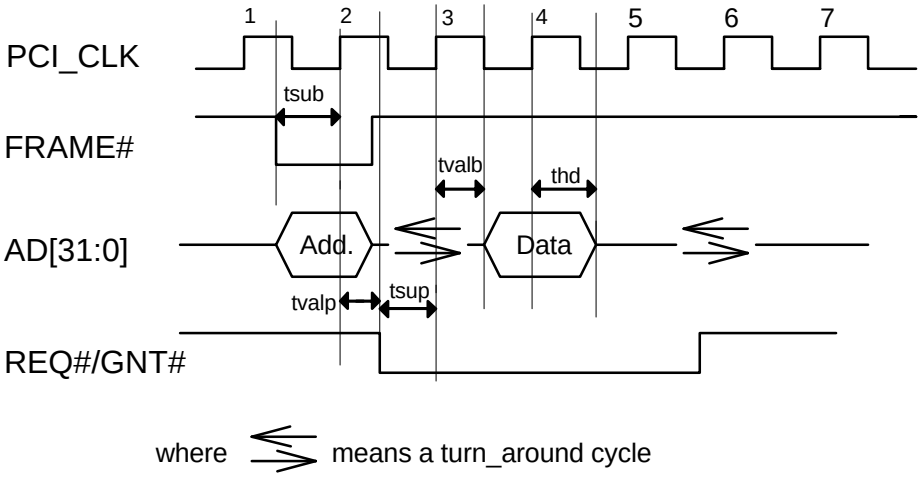


Figure 9-3: PCI Operation

Symbol	Parameter	Min	Max	Unit	Note
tsup	PtP signal set up time to PCI_CLK	10,12		ns	(2)
tsub	Bused signal set up time to PCI_CLK	7		ns	(1)
tvalp	PCI_CLK to PtP signal valid delay	2	12	ns	
tvalb	PCI_CLK to bused signal valid delay	2	11	ns	
thd	Input hold time to PCI_CLK	0		ns	

Table 9-5: PCI Operation Timing

Note:

(1) C/BE[3:0]#, AD[31:0], FRAME#,IRDY#,TDRY#,STOP#,DEVSEL# are all bus signal. These signals share the same AC timing.
(2) REQ# and GNT# are PtP (point to point) signals and have different output valid delay and setup time. REQ# has a setup of 12, GNT# has a setup of 10.

9.3.2.2 Standard Parallel Port (SPP) Timing

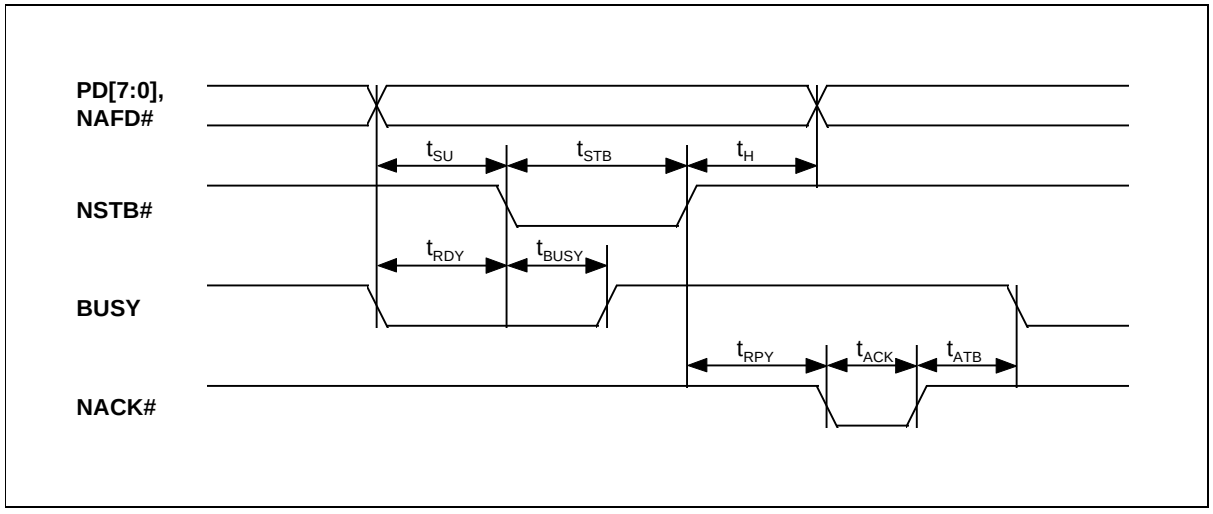


Figure 9-4: Standard Parallel Port (SPP) Timing

Symbol	Parameter	Min	Typ	Max	Unit
t_{SU}	Data Setup Time	500			ns
t_{STB}	Strobe Pulse Width	500			ns
t_{H}	Data Hold Time	500			ns
t_{RDY}	Busy Inactive to Strobe Active	0			ns
t_{BUSY}	Strobe Active to Bust Active			500	ns
t_{RPY}	Strobe Inactive to Ack Active	0			ns
t_{ACK}	Ack Pulse Width	500		10000	ns
t_{ATB}	Ack Inactive to Busy Inactive	0			ns

Table 9-6: Standard Parallel Port (SPP) Timing

9.3.2.3 Enhanced Parallel Port (EPP) Timing

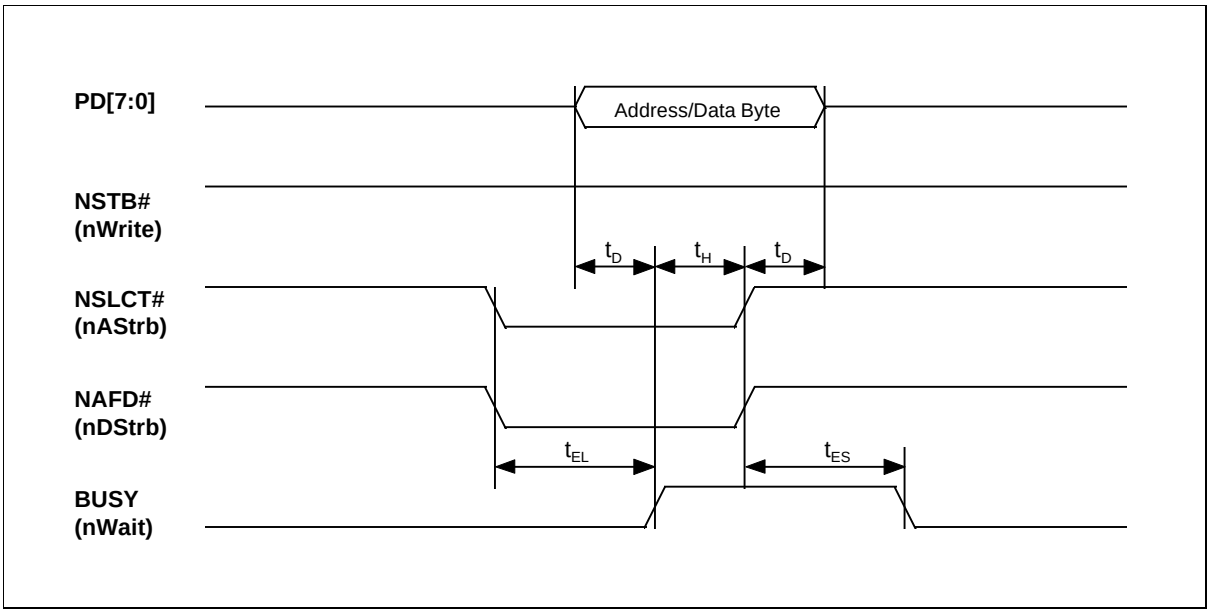


Figure 9-5: EPP Data or Address Read Timing

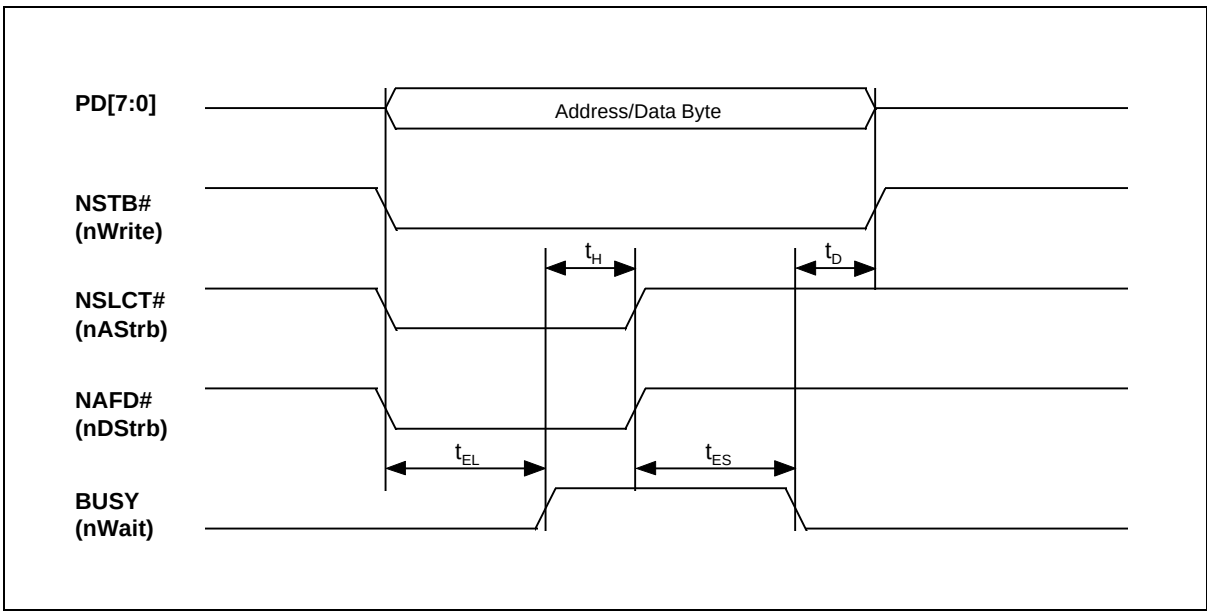


Figure 9-6: EPP Data or Address Write Timing

Symbol	Parameter	Min	Typ	Max	Unit
t_D	Data Setup Time	0			ns

t_H	Host Response Time	0		1	s
t_{EL}	Long Response Time	0		10	us
t_{ES}	Short Response Time	0		125	ns

Table 9-7: EPP Timing

9.3.2.4 Extended Capabilities Port (ECP) Timing

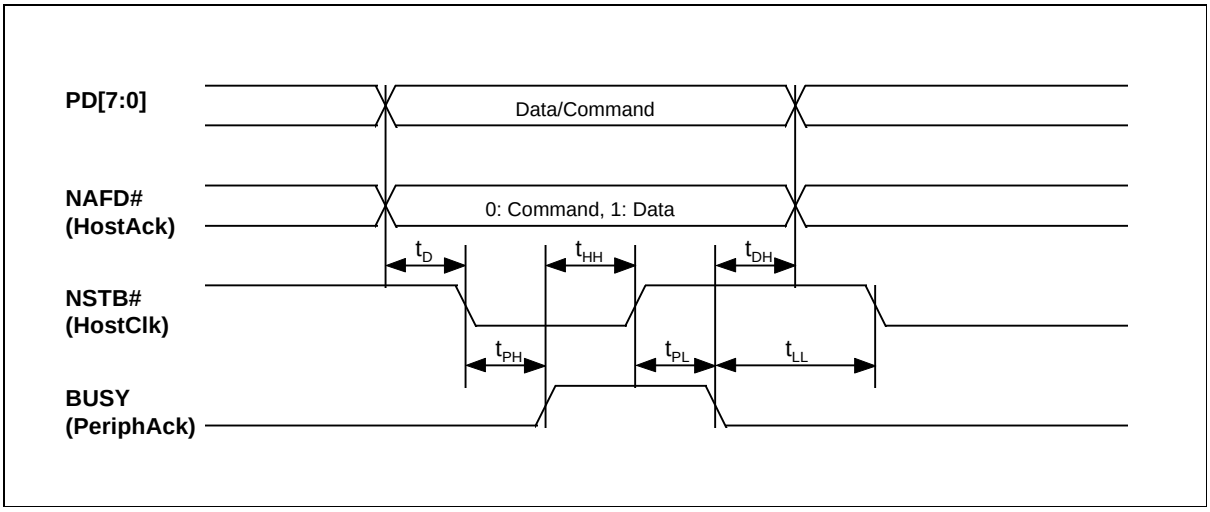


Figure 9-7: ECP Forward Timing

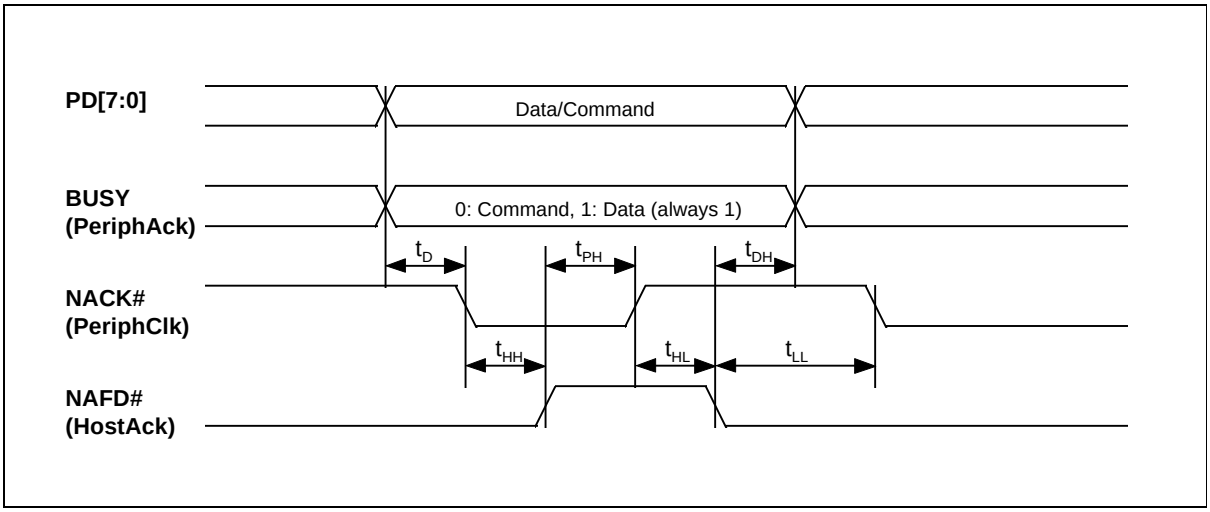


Figure 9-8: ECP Reverse Timing

Symbol	Parameter	Min	Typ	Max	Unit

t_D	Data Setup Time	0			ns
t_{PH}	Peripheral Response High Time	75			ns
t_{HH}	Host Response High Time	0		1	s
t_{PL}	Peripheral Response Low Time	0		35	ms
t_{HL}	Host Response Low Time	0		1	s
t_{DH}	Data Hold Time	0			ns
t_{LL}	Low to Low Response Time	0			ns

Table 9-8: ECP Timing

9.3.2.5 Generic Bus Signal Timing

Measuring Conditions: Recommended Operating Conditions. C_L=2[pF]					
Symbol	**Parameter**	**Min**	**Max**	**Unit**	**Notes**
t0	FIFO_RD_CLK Frequency		54	MHz	
t1	FIFO_RD_CLK Period	18.5		ns	
t2	FIFO_RD_CLK High Time	7		ns	
t3	FIFO_RD_CLK Low Time	7		ns	
t4	FIFO_RD_CLK Rise Time		1.5	ns	
t5	FIFO_RD_CLK Fall Time		1.5	ns	
t6	FIFO_RD_CLK Stability		±200	ps	
t7	A Setup Time	12		ns	MCLKI reference
t8	D Setup Time	6		ns	MCLKI reference
t9	BW, MAST_EN, RD#, WR# Setup Time	10		ns	MCLKI reference
t10	A, BW, D, MAST_EN, RD#, WR# Hold Time	2		ns	MCLKI reference
t11	D, INT#, RDY# Valid Delay	2	10	ns	MCLKI reference
t12	FIFO_RD# Setup Time	10		ns	FIFO_RD_CLK reference
t13	FIFO_RD#	2		ns	FIFO_RD_CLK reference
t14	FIFO_RDY	2	10	ns	FIFO_RD_CLK reference

Table 9-9: Generic Bus Timing

9.3.3 SDRAM Interface

Measuring Conditions: Recommended Operating Conditions. C_L=50[pF]					
Symbol	**Parameter**	**Min**	**Max**	**Unit**	**Notes**
t15	MDQ Setup Time	3		ns	SDRAM_CLK reference
t16	MDQ Hold Time	2		ns	SDRAM_CLK reference
t17	MDQ, MCS#, MRAS#, MCAS#, MWE#, MA Valid Delay	2.5	13	ns	SDRAM_CLK reference

Table 9-10: SDRAM Interface Timing

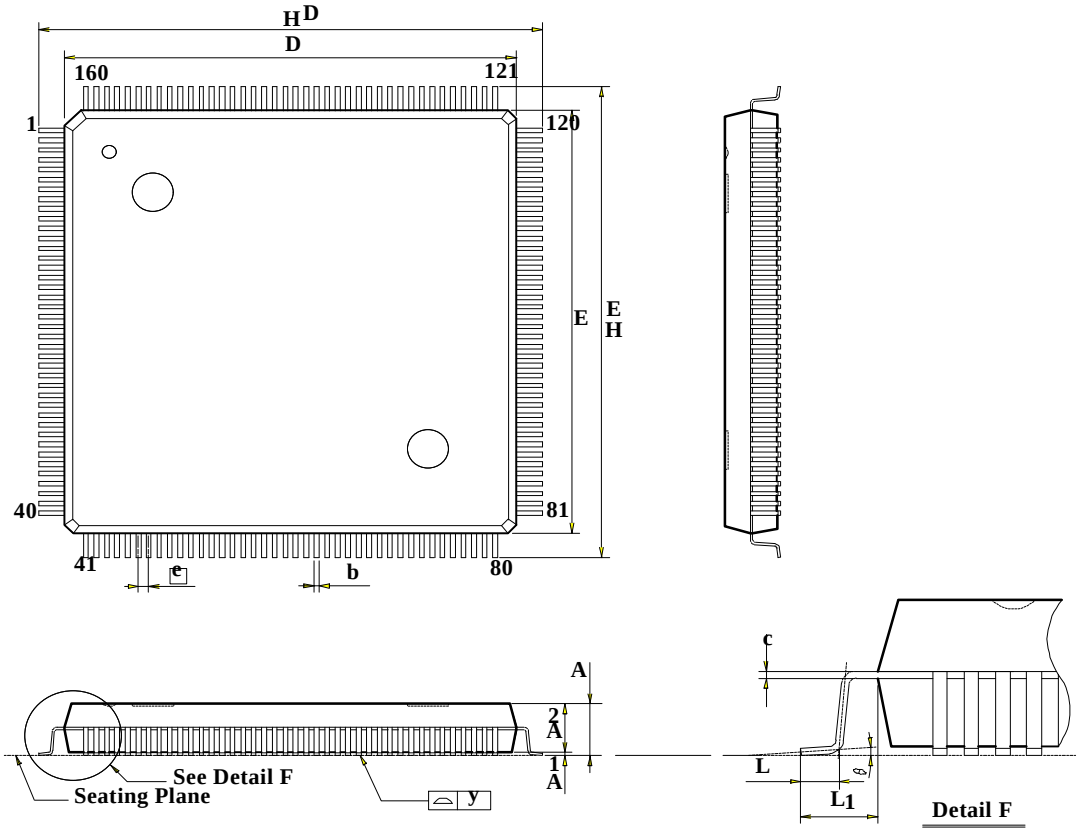
9.3.4 Video Decoder Interface

Measuring Conditions: Recommended Operating Conditions. C_L=50[pF]					
Symbol	**Parameter**	**Min**	**Max**	**Unit**	**Notes**
t18	VICLK Frequency	13.5	27	MHz	
t19	VICLK Period	37	74	ns	
t20	VICLK High Time	4		ns	Measured between 2.0V crossing points
t21	VICLK Low Time	4		ns	Measured between 0.8V crossing points
t22	VICLK Rise Time		1.5	ns	
t23	VICLK Fall Time		1.5	ns	
t24	VICLK Stability[1,2]		±200	ps	
t25	VID, VIDV, VIHACT, VIHS#, VIVS# Setup Time	5		ns	VICLK reference
t26	VID, VIDV, VIHACT, VIHS#, VIVS# Hold Time	1		ns	VICLK reference
I2C interface pins					
t27	SCL frequency		100	KHz	
t28	SCL, SDA rise and fall time		300	ns	

Table 9-11: Video Decoder Interface Timing

[1]The amount of jitter present must be accounted for as a component of VCLK skew between devices.
[2]Measured on rising edge of adjacent VCLKs at 1.5V.

10. Package Specification



Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	0.145	—	—	3.68
A1	0.004	—	—	0.10	—	—
A2	0.122	0.127	0.132	3.10	3.23	3.36
b	0.010	0.012	0.016	0.25	0.30	0.40
c	0.004	0.006	0.010	0.10	0.15	0.25
D	1.097	1.102	1.107	27.87	28.00	28.13
E	1.097	1.102	1.107	27.87	28.00	28.13
e	0.020	0.026	0.032	0.50	0.65	0.80
HD	1.216	1.228	1.240	30.90	31.20	31.50
HE	1.216	1.228	1.240	30.90	31.20	31.50
L	0.023	0.031	0.039	0.60	0.80	1.00
L1	0.055	0.063	0.071	1.40	1.60	1.80
y	—	—	0.004	—	—	0.10
θ	0°	—	12°	0°	—	12°

Note:

1. Dimension D & E do not include interlead flash.
2. Dimension b does not include dambar protrusion/intrusion .
3. Controlling dimension : Millimeter
4. General appearance spec. should be based on final visual inspection spec.

Figure 10-1: Package Specification

CORPORATE HEADQUARTERS:
NO. 4, Creation Rd. III
Science-Based Industrial Park
Hsinchu, Taiwan, R.O.C.
TEL: 886-3-5770066
FAX: 886-3-5796139

INFORMATION CONTACTS:
Hung Ming Wang
Multimedia Product Marketing & Planning Dept.
TEL: 886-3-5790666 Ext. 6135
Email : hmwang0@winbond.com.tw

Note: All data and specifications are subject to change without notice.