DATA SHEET

ENHANCED PCI BUS MULTIMEDIA CONTROLLER

FEATURES

- Glueless interface to PCI bus (PCI spec. 2.1 compliant).
- Minimum interface to JPEG decoders (e.g. ZR36050 & ZR36016), MPEG1 decoders (e.g., ZR36110), video digitizers (e.g., SAA7110, SAA7111) and video encoders (e.g., SAA7188, MD207).
- Bi-directional DMA transfer of compressed data up to 11M bytes/sec.
- DMA transfer of video and overlay information.
- Support for fast still image compression and decompression.
- Smooth image down-scaler (up to 5-tap horizontal filter).
- On-chip pixel accurate masking.
- YUV-to-RGB converter with quantization noise reduction by error diffusion.
- Video output: 15- and 16-bit RGB pixel formats, as well as 24-bit (packed and unpacked), and YUV 4:2:2.
- Hardware support for non-contiguous JPEG code buffers.
- Graceful recovery from extreme bus latencies both on video and code transfers.
- Choice of emulated interlaced video display, or single field display, to eliminate motion artifacts.
- Hardware support for simple, cost effective frame grabbing.
- I^2C bus master port.
- Plug & Play support.
- 208-pin PQFP package.
- Accompanying software includes Video For Windows (VFW) drivers for Windows 3.1 and Windows 95.

APPLICATIONS

- High quality video and audio capture/playback and editing boards for PCI systems.
- Multimedia/Graphics subsystems using a secondary PCI bus.
- PCI motherboards with multimedia capability.
- JPEG/MPEG1 solutions for PowerPC and Macintosh PCI systems.

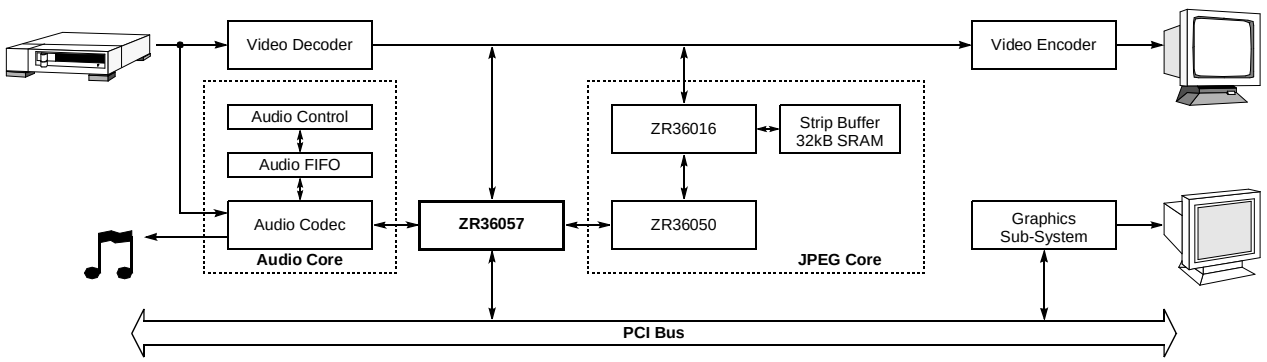


Figure 1. Block Diagram of a Typical Motion JPEG System for PCI

Enhanced PCI Bus Multimedia Controller

1.0 INTRODUCTION

1.1 The ZR36057

The ZR36057 is a PCI adapter intended for multimedia applications on PCI systems. It supports high rate code (compressed data stream) transfer between the system memory and JPEG or MPEG processors. Simultaneously to the code transfer, the ZR36057 captures digital video, such as decompressed MJPEG (Motion JPEG), MPEG, or the output of a video digitizer/ decoder, and creates a scaled video window in the graphics display memory.

The ZR36057 provides the host software with full control over a large number of non-PCI multimedia devices such as: ZR36050, ZR36016 (Zoran Motion JPEG Codec chip set), audio codecs, ZR36100/ZR36110 (Zoran MPEG decoders), etc, as well as any number of I2C devices, such us video digitizers, video encoders, etc.

The ZR36057 interfaces directly to the PCI bus. As a bus master, it transfers data (e.g, MPEG or JPEG code) to or from the system memory, and writes digital video pixels to the graphics display memory. As a bus target, the ZR36057 reflects the host accesses onto a microcontroller-type 8-bit “Guest Bus”.

The ZR36057 has a special “Still Transfer” port by means of which the host software writes (or reads) digitized video, as RGB pixels, from the system memory to (or from) the video bus. This path enables fast transfer of still images to be compressed (or decompressed) by the JPEG chip set.

1.2 JPEG System Overview

Figure 1 depicts an example of an MJPEG add-on board.

The ZR36057 supports 4 basic JPEG modes of operation:

- Motion Video Compression.
- Motion Video Decompression.
- Still Image Compression.
- Still Image Decompression.

1.2.1 Motion Video Compression

The video decoder directs the video in YUV 4:2:2 format, and the video synchronization signals, to the video input port of the ZR36016. The video is also transferred to the video encoder for display on a TV monitor and simultaneously to the Video Front End of the ZR36057. The ZR36057 can optionally down-scale the video, convert it to RGB, and transfer the pixels using DMA to the display memory of the host PC. In parallel, the ZR36016 performs the raster-to-block operation needed by the JPEG algorithm and transfers the video in block order to the ZR36050 for compression. The ZR36050 drives the code stream to the Codec Front End of the ZR36057, which transfers the compressed video fields using DMA to a system memory buffer allocated by the host.

1.2.2 Motion Video Decompression

In Motion Video Decompression, the ZR36057 transfers the code stream from system memory via the ZR36057 Codec Front End to the ZR36050, using DMA. The ZR36050 decompresses the JPEG code and transfers the decompressed blocks to the ZR36016. The ZR36016 performs the block-to-raster conversion and drives the video to the video encoder to be displayed on a TV monitor. The ZR36016 video output is driven simultaneously to the Video Front End of the ZR36057 to be processed, as in the capture mode, and transferred using DMA to the PC display memory.

1.2.3 Still Image Compression

In Still Image Compression mode an image bitmap is written by the host, pixel by pixel, through the PCI bus to the ZR36057. The ZR36057 transfers the pixels through its video bus port to the video input port of the ZR36016. After the first strip of 8 video lines is filled, the ZR36016 starts performing the raster-to-block operation, sending the blocks to the ZR36050. The ZR36057 generates and drives the required video synchronization signals for the ZR36016. The ZR36050 compresses the video blocks and drives the code stream to the ZR36057. The code stream is transferred using DMA to the host memory as in Motion Video Compression.

1.2.4 Still Image Decompression

In Still Image Decompression mode, the ZR36057 fetches the code stream from system memory using DMA, as in Motion Video Decompression. The ZR36050 reads the compressed data from the ZR36057, decodes it and send the expanded blocks to the ZR36016, which drives the video to the video port of the ZR36057. From there the host software reads it out to system memory, pixel by pixel.

Note that still image decompression can also be accomplished by configuring the ZR36057 in Motion Video Decompression mode, and transferring the decompressed video to a contiguous buffer in system memory instead of the display memory. Since this has a speed advantage over Still Image Decompression mode, it is the preferred method for most applications.

1.2.5 Notations and Conventions

External signals:	Capital letters (e.g., IDSEL)
Active-low mark [1]:	Overbar (e.g., $\overline{\text{DEVSEL}}$)
Internal function units:	capital (non-bold) letters (e.g., VFE)
Buses:	XXmsb_index..lsb_index (e.g., AD31..0)
Register fields:	XXmsb_index:lsb_index (e.g., Mode27:16)
Register types:	R - read only RC - read-clear. Writing ‘1’ clears the register bit. RS - read-set. Writing ‘1’ sets the register bit to ‘1’. RW - read-write (contents of write can be read back) W - write only (contents of read are meaningless)
Numbers:	Unmarked numbers are decimal (e.g., 365, 23.19). Hexadecimal numbers are marked with a ‘0x’ prefix (e.g., 0xB000, 0x3). Binary numbers are marked with a ‘b’ suffix (e.g., 010b, 0000110100011b).

1. In this document, an overbar is used to denote active low signals. In other documents referenced herein, such as the PCI specifications, the # suffix notation is often used instead. The two forms of notation are interchangeable. Thus, for example, $\overline{\text{DEVSEL}}$ is equivalent to DEVSEL#.

2.0 ARCHITECTURAL OVERVIEW

The ZR36057 architecture contains two main data paths, the video path and the code path. The incoming video is processed along the video path and transferred to the graphics display memory using PCI DMA bursts.

The ZR36057 Video Front End samples the video bus within a programmable active field window, defined with repect to the video synchronization signals. An optional vertical and horizontal smooth scale down can be applied, in order to support variable image sizes and variable PCI video data rate. The scaled video stream can be converted to various RGB formats. The converted pixels are packed and stored in a 256-byte Video FIFO, organized as 64 32-bit doublewords. The stored video pixels are read from the Video FIFO and transferred to the graphics display memory according to a display masking map controlled by and stored inside the ZR36057.

The Code path is bidirectional. The data flow direction depends on the mode of operation. The code stream (MPEG or JPEG) is transferred between system memory and the internal Code FIFO of the ZR36057 using PCI DMA bursts. The ZR36057 controls the transfer and addressing in both directions. The Code FIFO size is 640 bytes, organized as 160 doublewords.

In JPEG Compression modes the ZR36057 Codec Front End fills the Code FIFO. From the Code FIFO the code is transferred to the system memory, field by field.

In JPEG Decompression modes the code stream flows in the opposite direction, from the system memory to the ZR36057 Code FIFO. The Codec Front End reads out the Code FIFO byte by byte onto the Code Bus.

In MPEG Playback mode, the code stream is transferred to the ZR36057 Code FIFO from the system memory. The code bytes are read from the Code FIFO out to the Guest Bus.

The ZR36057 video and the code paths operate simultaneously while the ZR36057 arbitrates the PCI bus requests for each process.

Besides managing the Video and Code paths, the ZR36057 bridges the host CPU to peripheral devices (known as Guests). Using a dedicated handshaking mechanism (the “PostOffice” mechanism), host accesses to an internal ZR36057 register are reflected to the Guest Bus in order to enable indirect host read and write operations to the Guests.

The ZR36057 contains a dedicated “Still Transfer” port which enables data flow between the PCI interface and the Video Front End. Using a specific handshake protocol, the host software may transfer digitized video (RGB pixels) from the system memory to the Video bus, and vice versa. This path enables very fast transfer of still video images to be compressed or decompressed by the JPEG codec.

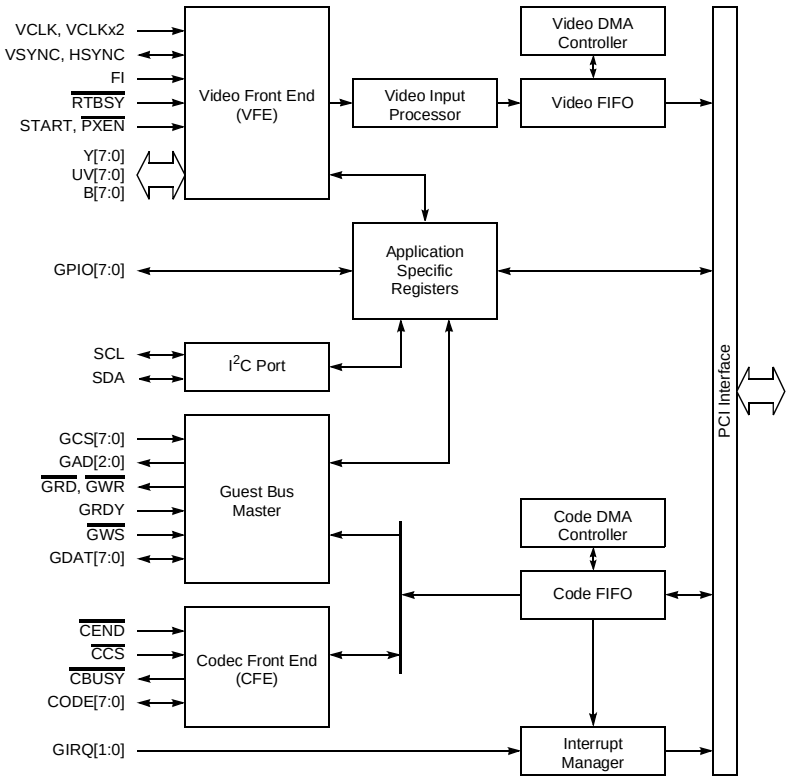


Figure 2. ZR36057 Block Diagram

3.0 PIN DESCRIPTIONS

Symbol	Type [1]	Direction	Description
PCI Interface (48 pins)			
AD[31:0]	3-state	I/O	Multiplexed address and data bus pins.
C/$\overline{BE}$[3:0]	3-state	I/O	Bus commands or byte enables.
PAR	3-state	I/O	Even parity bit for AD31..0 and C/$\overline{BE}$[3:0].
$\overline{FRAME}$	3-state*	I/O	PCI cycle frame.
$\overline{TRDY}$	3-state*	I/O	PCI target ready indicator.
$\overline{IRDY}$	3-state*	I/O	PCI initiator ready.
$\overline{STOP}$	3-state*	I/O	Indicates a target request to stop the current data transfer.
$\overline{DEVSEL}$	3-state*	I/O	PCI device select, indicates that the target has decoded its address.
IDSEL	I	I	PCI initialization device select. Used as a chip-select to the ZR36057's configuration space.
$\overline{REQ}$	3-state	O	PCI bus request.
$\overline{GNT}$	3-state	I	PCI bus grant.
PCICLK	I	I	PCI clock.
$\overline{PCIRST}$	I	I	PCI reset. When active, all ZR36057 output pins are tri-stated. A low to high transition puts the ZR36057 into its power-on reset state. Minimum active low duration is 3 PCI clocks.
$\overline{INTA}$	open drain	O	PCI interrupt request A. A low level on this signal requests an interrupt from the host.
Digital Video Bus Interface (32 pins)			
Y[7:0]/R[7:0]	3-state	I/O	Luminance/Red video lines.
UV[7:0]/G[7:0]	3-state	I/O	Chrominance/Green video lines.
B[7:0]	3-state	I/O	Blue video lines.
VCLKx2	I	I	Double frequency video bus clock.
VCLK	I	I	Digital video bus clock. Used as a qualifier to VCLKx2. Must be synchronized to VCLKx2.
HSYNC	3-state	I/O	Digital video bus horizontal sync.
VSYNC	3-state	I/O	Digital video bus vertical sync.
FI	I	I	Digital video bus field indicator (top/bottom).
$\overline{PXEN}$	O	O	Active low Pixel Enable output to the ZR36016.
$\overline{RTBSY}$	I	I	Active low Strip Memory Overflow/Underflow signal from the ZR36016.
START	O	O	Active high Start process output to the ZR36016.
GuestBus Interface (25 pins)			
$\overline{GCS}$[7:0]	O	O	Active low chip-select output to guest bus devices.
GADR[2:0]	O	O	Address outputs to guest bus devices.
GDAT[7:0]	3-state	I/O	Guest data bus .
$\overline{GRD}$	O	O	Active low read output to guest bus devices.
$\overline{GWR}$	O	O	Active low write output to guest bus devices.
GRDY	I	I	Active high "guest ready" input.
$\overline{GWS}$	I	I	Guest Wait-State indication. Assertion of this active-low input allows the guest device to extend the GuestBus write (or read) cycle until it is capable of latching-in (or providing) the data.
GIRQ[1:0]	I	I	Positive-edge-sensitive interrupt request inputs from one or two of the guest bus slave devices.
CodecBus Interface (11 pins)			
CODE[7:0]	3-state	I/O	Code Bus connected to the ZR36050.
$\overline{CEND}$	I	I	Active low End of field process indication from the ZR36050.
$\overline{CCS}$	I	I	Active low Code Bus active cycle signal from the ZR36050.
$\overline{CBUSY}$	O	O	Active low Code FIFO Busy indication to the ZR36050.
I2C Bus Interface (2 pins)			
SDA	OD	I/O	I^2C bus data

3.0 PIN DESCRIPTIONS (CONTINUED)

Symbol	Type [1]	Direction	Description
SCL	OD	I/O	I^2C bus clock (qualifies for a single master operation only).
General Purpose Programmable Inputs/Outputs (8 pins)			
GPIO[7:0]	3-state	I/O	General purpose input/output pins. After hardware or software reset all 8 pins are configured as inputs. Their logical levels are reflected as register bits. Any of the pins can be configured as output. In this state its logical level is driven by a register bit. These pins may be used to monitor or control various board-level functions.
Test Pins (2 pin)			
ENID	Test	I	Used for IDD test. In normal operation must be connected to GND.
TEST	Test	I	Test pin used in test mode only. In normal operation must be connected to GND.
Power (80 pins)			
GND	ground		Ground (50 pins).
VDD	power		Power supply (5V) (30 pins).

1. I - standard input-only • O - standard active driver • 3-state - bi-directional I/O pin • 3-state* - a special type of 3-state, as defined in the PCI spec. May be driven by only one PCI agent at any time • OD - open drain, may be shared by multiple drivers, as a wired-or.

4.0 FUNCTIONAL OVERVIEW

The ZR36057 multimedia controller performs the following functions:

- Interfacing to a YUV 4:2:2 digital video bus (e.g., Phillips SAA7110 or SAA7111).
- Video DMA channel for burst transfers of video pixels.
- Independent horizontal and vertical downscaling of the input image, with optional horizontal filtering.
- Optional sync mastering with configurable HSYNC and VSYNC pulse widths and polarity.
- Conversion of the YUV 4:2:2 digital video input into one of the following pixel formats: YUV 4:2:2, RGB 5,6,5, RGB 5,5,5 or RGB 8,8,8 (packed or unpacked).
- Overlay support: any number of video pixels can be masked off, letting the corresponding graphics pixels appear instead of them.
- Frame grabbing.
- Two display modes: emulation of the interlaced input video, or a single field display.
- Bidirectional Code DMA transfer with support for fragmented code buffers.
- Control of the ZR36050/ZR36016 Motion JPEG chip set.

The ZR36057 supports digital video in CCIR 601 or square pixel formats, following either the NTSC or PAL video standard. Other non-standard input schemes are supported as well.

The functional description below follows the block diagram.

4.1 Digital Video Path

4.1.1 Digital Video Front End (VFE)

The VFE samples the incoming YUV 4:2:2 video data and sync signals with a flexible sampling scheme, that makes it compatible with a wide variety of digital video sources. The digital input video can be cropped. The input resolutions supported by the VFE range from 32x32 to 1023x1023, in increments of one pixel.

4.1.2 Video Input Processor

The chroma components of the video data are upsampled to YUV 4:4:4 format. All components are horizontally filtered. Five filtering schemes are implemented, with different parameters for chrominance and luminance samples. Horizontal and vertical downscaling is available if required. The vertical downscaling can be optimized for live video or Motion JPEG playback, in which each field is independent, or for the output of an MPEG-1 decoder such as the ZR36100 or ZR36110, which duplicates fields to produce its interlaced video output.

4.1.3 Pixel Formatting

The filtered and scaled video is converted to the desired color space and packed according to the selected pixel format. YUV 4:2:2, 24-bit RGB (packed or unpacked) and 15- and 16-bit RGB are supported. An error diffusion algorithm can be applied to the RGB 5,5,5 and 5,6,5, in order to eliminate quantization artifacts on the output image.

4.2 Video DMA Controller

4.2.1 Pixel Bursts

The packed pixels are transferred directly to the display memory (or to the system memory), using PCI DMA bursts. Both Little

and “Gib” Endian formats are supported where applicable (Refer to the PCI Multimedia Design Guide, Revision 1.0).

4.2.2 Display Modes

The display mode can be configured to either emulated interlaced video (both input fields are displayed simultaneously on the non-interlaced monitor) or single field display. The latter is appropriate for motion artifact elimination when displaying live or decompressed Motion JPEG video.

4.2.3 Frame Grabbing

The ZR36057 can grab video frames (scaled or non scaled), or fields, in any of the pixel formats listed above, directly into system memory, eliminating the need for memory on the add-in board.

4.2.4 Overlay Control

Graphics overlay is supported, in that display memory areas that are “owned” by graphics applications, may not be loaded with video pixels, allowing true windowing and overlay. The software driver prepares a masking map of the video rectangle, and the ZR36057 uses this map for masking decision, when transferring the pixels to the display memory.

4.3 Host Control/Communication Services

4.3.1 Application-Specific Registers (ASRs)

The name “application-specific” distinguishes these registers from the PCI configuration space registers. These memory mapped registers provide the host software with full control over the operation of the ZR36057. The ZR36057 claims a contiguous space of 4 KBytes in system memory.

4.3.2 GuestBus

Host software control over non-PCI devices, such as a Motion JPEG codec, an MPEG decoder, a video encoder, etc., is done through the ZR36057’s GuestBus. Host accesses to these “guest” devices, mapped as application specific registers inside the ZR36057, are output as GuestBus cycles. Such accesses can either use the PostOffice handshaking protocol, or the Code DMA Controller. The first method is adequate for commands, configuration data, etc., while the second method provides a faster channel, and is intended for continuous transfer of data such as a compressed bitstream.

4.3.3 PostOffice Handshaking Protocol

The ZR36057 PostOffice handshaking protocol, implemented over the GuestBus, allows host accesses to relatively slow guest devices, with no degradation of the PCI bus performance.

4.3.4 Still Transfer Mechanism

The ZR36057 supports a dedicated Still Transfer port, by means of which the host writes (reads) image pixels in JPEG Still Image Compression (Decompression) mode. The Still Transfer port is mapped inside the ASR area, and a special controller interconnects this port to the ZR36057’s extended video bus (24 bits RGB). The Still Transfer handshake protocol enables high rate pixels transfer between the system memory and the JPEG processor via the ZR36057.

4.3.5 I^2C Port

A software-driven I^2C port allows controlling of I^2C devices.

4.3.6 Interrupt Manager

Interrupt requests associated with several internal and external conditions are sent to the host via the PCI bus (using $\overline{\text{INTA}}$). Selection of interrupt originators is programmable.

4.4 Code DMA Controller

The ZR36057 includes a DMA channel for transferring data between the system memory and a selected device on the Code Bus or GuestBus. Two configurations are supported:

- MPEG mode. The data flow is unidirectional, from the system memory, to the ZR36057’s GuestBus.
- JPEG mode. The data flow is bidirectional, and the direction is determined by the selected sub-mode. In JPEG Compression, the data flows from the Codec bus to the system memory. In JPEG Decompression, the data flows from the system memory to the Codec bus.

4.4.1 MPEG Mode

In MPEG mode, the data flows from the system memory to the ZR36057’s GuestBus. Typically, this would be a compressed bitstream, to be decompressed by a device such as an MPEG decoder attached to the GuestBus. Other examples are sampled audio (WAV data), MIDI token stream, etc. Temporary latencies on the PCI bus or the GuestBus are handled without loss of data.

The GuestBus master simultaneously serves the PostOffice accesses and the code DMA transfers: DMA transfers are viewed as the “main” task of the GuestBus master, while any number of PostOffice requests may occasionally interrupt the DMA traffic.

The DMA controller supports both auto-initialized (cyclic) block transfers, or single block transfers. The size of the destination block in main memory can be selected out of several possible sizes, ranging from 8 KBytes up to 256 KBytes. The destination block may also be virtually split into several sub-blocks, allowing the ZR36057 to interrupt the host when a sub-block has been transferred. This feature provides the software with a means of optimizing the refill accesses according to the application requirements and the disk performance.

4.4.2 JPEG Modes

In the JPEG modes, the data flows between the system memory and the ZR36057’s Codec bus. In JPEG Compression, from the

Codec bus to the system memory; in JPEG Decompression, from the system memory to the Codec bus.

The JPEG code data inside the system memory is structured within code buffers. Each code buffer may contain a compressed field or a frame (2 fields), in accordance with a user configurable register bit.

The ZR36057 supports four code buffers, defined dynamically in a dedicated table (the Code Buffer Table) in the system memory. The actual memory of each code buffer may be fragmented. The content of each entry in the code buffer table is a pointer to a secondary Fragment Table. The fragment table contains the pointers to the allocated memory chunks (fragments).

In JPEG Compression mode, the host software builds and updates the code buffer table and the fragment tables according the memory allocated by the operating system. On every JPEG field/frame process, the ZR36057 Code DMA Controller reads the fragment table pointer of the current buffer from the code buffer table, and then fills up the buffer fragments one by one.

In JPEG Decompression mode, the host software builds the code buffer table and fragment tables, and fills the fragments with the compressed field or frame. On every JPEG process, the ZR36057 Code DMA Controller retrieves the code fragments one by one, and directs them to the Codec bus.

5.0 INTERFACES

5.1 PCI Bus Interface

In general, the ZR36057 is compatible with the PCI 2.1 specifications. As a bus master, it may initiate two types of data transfer over the PCI bus:

- Memory Write (PCI command 0111b), from the ZR36057's Video FIFO buffer to the display memory (or main memory), and from ZR36057's Code FIFO buffer to the system memory.
- Memory Read Line (PCI command 1110b), from system memory to the ZR36057's Mask Buffer[1] and from system memory to the ZR36057's Code FIFO buffer.

As a bus target, the ZR36057 responds to the following types of transfer:

Memory Read	(0110b)
Memory Read Line	(1110b)
Memory Read Multiple	(1100b)
Memory Write	(0111b)
Configuration Read	(1010b)
Configuration Write	(1011b)

All other PCI commands are ignored.

Memory Read Line and Memory Read Multiple are handled exactly like Memory Read.

Normally, as a slave, the ZR36057 is intended to be accessed with single data phase cycles. However, multiple phase bursts are supported. When the ZR36057 is accessed in a burst it increments its internal (offset) address such that each data phase is routed to/from the next address location (in double-words). The ZR36057 supports byte enables, such that an access to explicit bytes is possible.

The error reporting signals, $\overline{\text{SERR}}$ and $\overline{\text{PERR}}$, are not included in the ZR36057: as a multimedia device it is only required to report parity errors through the PCI status register.

The ZR36057 uses the $\overline{\text{INTA}}$ PCI interrupt request line.

5.2 Digital Video Interface

The ZR36057 interfaces to a wide spectrum of digital video devices. The Video Interface is bidirectional and two video pixel flows are supported:

- The incoming video is sampled and directed via the ZR36057's video input processor to the graphics display memory or system memory.
- The video pixels are transferred to or from the system memory using a dedicated mechanism (the Still Transfer mechanism).

The ZR36057 supports two sync signal source configuration options:

- External sync - the sync signals are driven by the external video source.
- Internal sync - the sync signals are generated internally and mastered by the ZR36057.

In JPEG Motion Video Compression and Decompression modes as well as in MPEG mode, the Video Interface transfers the incoming video to the ZR36057's video input processor. The sampling of the video stream is performed according to the video clocks and sync signals. In JPEG Motion Video Compression mode and in MPEG mode, the synchronization source should be external. In JPEG Motion Video Decompression mode, the synchronization source can be either external or internal.

In JPEG Still Image Compression and Decompression modes, the ZR36057 uses a dedicated mechanism (the Still Transfer mechanism) as a means for the host software to transfer pixels to or from the compression module. The Video Interface masters

1. Not shown in the block diagram.

the extended 24-bit video bus and the synchronization signals in order to drive the pixels to the ZR36016 or get them from it, as appropriate.

The following four subsections detail the four basic functions of the Video Interface:

- Sampling the incoming video.
- Generating the synchronization signals.
- Pixel transfer in Still Image Compression.
- Pixel transfer in Still Image Decompression.

5.2.1 Sampling The Incoming Video

The ZR36057's Video Front End (VFE) interfaces to a standard YUV 4:2:2 video bus. It samples the Y7..0, UV7..0, HSYNC and VSYNC with every other positive edge of VCLKx2. The valid positive edge (out of every two consecutive ones), which is the one used for sampling, is "marked" by VCLK, i.e., VCLK is used as a clock qualifier. The qualifying polarity of VCLK is configured by the host. This scheme makes the ZR36057 compatible with a wide range of digital video sources and immune to board-level parasitic delays. VCLKx2 (positive edges) is used internally in the video processing pipeline.

The VFE generates a field indication signal targeted to some internal video processing units. There are two alternative ways of generating the field indication. With devices that output a field indication, the VFE uses the FI input as an indicator of the current field identity. The interpretation of the logical level of FI (top or bottom field) is configured by the host. With devices that do not provide such an indication, the VFE infers the field identity from the relationship of HSYNC to VSYNC.

The VFE can capture square pixel and CCIR-601 formats, or user defined formats, within the limitation of its parameters. The maximum theoretical total input resolution is 1023 pixels/line x 1023 lines per frame. Cropping of the input image is possible by proper configuration of the VFE parameters.

Table 1 lists the Video Front End parameters. The host software needs to configure these parameters according to the timing parameters of the video source (e.g., SAA7110, SAA7111, ZR36100, etc.) and the required cropping. Note that these parameters relate to the input video, and not to the destination video window.

Table 1: Video Front-End Parameters

Parameter	Description
VStart	Number of lines (HSYNCs) from the active edge (positive or negative, according to VSPol) of VSYNC to the first line to be sampled.
HStart	Number of pixel clocks in a line from the active edge of HSYNC until the first pixel to be sampled.
VEnd	Number of lines (HSYNCs) from the active edge (positive or negative, according to VSPol) of VSYNC to the last line to be sampled.

Table 1: Video Front-End Parameters

Parameter	Description
HEnd	Number of pixel clocks in a line from the active edge of HSYNC until the last pixel to be sampled.
ExtFI	This one bit parameter indicates whether the video source provides a field indication signal.
HSPol	The HSYNC polarity. HStart and HEnd are counted from the active edge of HSYNC. '1' means that HStart, HEnd will be counted from the negative edge of HSYNC.
VSPol	The VSYNC polarity. VSstart and VEnd are counted from the active edge of VSYNC. '1' means that VStart, VEnd will be counted from the negative edge of VSYNC.
TopField	Top Field Interpretation. If field indication is derived from the FI input signal (see ExtFI), TopField indicates the interpretation of the FI signal: TopField='1' - FI high indicates the top field (default value). TopField='0' - FI low indicates the top field. If field indication is derived internally from HSYNC and VSYNC, TopField indicates the interpretation of the level of HSYNC as sampled by the active edge of VSYNC: TopField='1' - HSYNC high indicates the top field (default value). TopField='0' - HSYNC low indicates the top field.
VCLKPol	Polarity of VCLK as a data qualifier. If VCLKPol=1 the video input is sampled with those positive edges of VCLKx2 that correspond to VCLK=1. If VCLKPol=0, the video input is sampled by those positive edges of VCLKx2 that correspond to VCLK=0.

5.2.2 Synchronization Signal Generation

The ZR36057 supports internal generation of the video synchronization signals. In this mode the ZR36057 generates and drives VSYNC and HSYNC signals.Using software programmable parameters, the ZR36057 can generate various video synchronization signal formats.

Table 3 lists the sync signal parameters. The host software configures those parameters according the mode of operation and the video peripheral devices used.

The polarity of the sync signals is determined by the VSPol and HSPol parameters (Table 2)

Table 2: Synchronization Signal Parameters

Parameter	Meaning
FrmTot	Total number of lines per frame (e.g., in NTSC: 525)
LineTot	Total number of pixel clocks per line (e.g., in CCIR NTSC: 858)
VsyncSize	The length of the VSYNC signal, measured in lines.
Hsync Start	The point in the scan line at which the HSYNC signal should be asserted.

The ZR36057 Video Interface drives out the video synchronization signals synchronized with the negative edge of VCLKx2 (every other edge). The VCLK input is used as a phase qualifier.

The timing of the rising and the falling edges of VSYNC with respect to the HSYNC signal are:

- Odd fields - The edges of VSYNC occur in the middle of the non-active portion of HSYNC.
- Even fields - The edges of VSYNC occur in the middle of the active portion of HSYNC.

5.2.3 ZR36057 Connection To The ZR36016

Figure 3 shows the recommended connections between the **ZR36057** and the ZR36016.

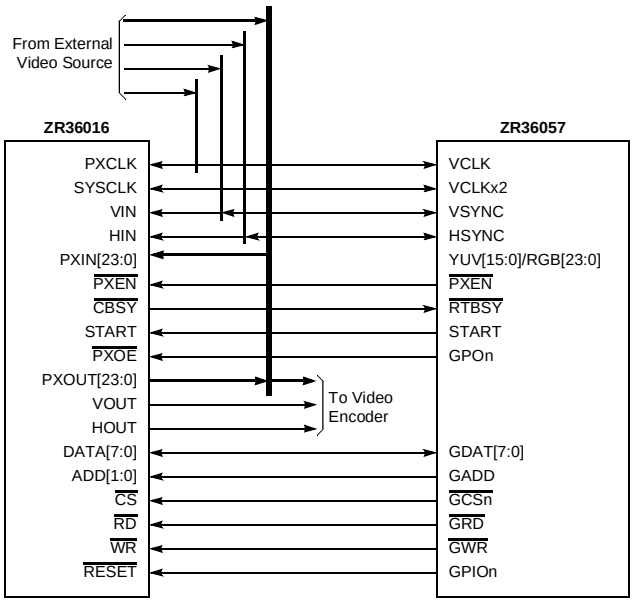


Figure 3. Recommended ZR36057 to ZR36016 Connections

In Motion Video Compression, the YUV video and synchronization signals are driven by the external video source (for example, a SAA7110) to the inputs of the ZR36057 and ZR36016. The ZR36057 disables the PXOUT bus of the ZR36016 by deasserting $\overline{\text{PXOE}}$, in order to prevent bus contention.

In Motion Video Decompression, the synchronization signals are driven by the ZR36057 (in sync master mode) or by an external sync generator, associated with the video source. The decompressed digital video is transferred from the ZR36016 PXOUT bus to the ZR36057 video interface. The video bus of the external video source, and if necessary its sync signals, must be 'tri-stated' using software control.

In Still Image Compression, the video bus as well as the sync signals are driven by the ZR36057. The ZR36057 deasserts the PXOE input of the ZR36016. The video bus of the external video source (such as the video decoder of Figure 1), and if necessary its sync signals, must be tri-stated using software control.

In Still Image Decompression mode, the sync signals are driven by the ZR36057. Decompressed video is transferred from the ZR36016's PXOUT bus to the ZR36057's Video Interface. In this mode of operation, the video bus and sync signals of an external video source (such as the video decoder of Figure 1) must be tri-stated.

Table 3 defines the ZR36016 and the ZR36057 parameters which define the portion of the field to be processed (the active portion) and the type of field to begin with.

Table 3: Parameters Defining The Active Portion Of A Field

Parameter	Meaning
NAX	The number of pixels to be skipped, counted from the active edge of HSYNC.
PAX	The number of active pixels in a line.
NAY	The number of lines to be skipped counted from the active edge of VSYNC.
PAY	The number of active lines in a field.
Odd_Even	The type of the first field to be processed. '1' means an Odd field, '0' means an Even field.

5.2.3.1 START Signal Control

The START Signal is used in order to start the Compression/ Decompression process at the correct time for the desired field type.

5.2.3.2 $\overline{\text{PXEN}}$ Signal Control

The ZR36057 drives the $\overline{\text{PXEN}}$ input of the ZR36016. While $\overline{\text{PXEN}}$ is deactivated, the ZR36016 does not sample the PXIN bus, nor the HIN and VIN inputs. Also, it freezes the PXOUT bus, and HOUT and VOUT.

In order to keep the ZR36057 and the ZR36016 synchronized, deactivation of $\overline{\text{PXEN}}$ by the ZR36057 also 'holds' the horizontal counting and processing inside the ZR36057.

In Motion Video Compression mode, $\overline{\text{PXEN}}$ is activated continuously.

In Motion Video Decompression mode, $\overline{\text{PXEN}}$ is asserted by then ZR36057 after latching the rising edge of $\overline{\text{RTBSY}}$, indicating that the first strip is ready in the ZR36016 strip memory.

In Still Image Compression mode, the image is transferred to the ZR36057 by the host software pixel by pixel. $\overline{\text{PXEN}}$ is activated only when a pixel is ready to be sent out from the ZR36057 Video Interface to the ZR36016 video input.

In Still Image Decompression mode, the image is read from the ZR36057 by the host software pixel by pixel. $\overline{\text{PXEN}}$ is activated to get the next pixel from the ZR36016 only after the previous pixel has been read by the host.

PXEN can be de-asserted dynamically (in the middle of a process) in response to several different conditions. The dynamic de-assertion is enabled independently by three configuration bits - VFIFO_FB, CFIFO_FB, RTBSY_FB. The host software is allowed to set those bits only while the ZR36057 is the sync master.

When VFIFO_FB is set, if the Video FIFO is close to overflow status, PXEN is de-asserted to hold the ZR36016 video output until the Video FIFO is emptied. When CFIFO_FB is set, if the Code FIFO underflows, and CBUSY is asserted, PXEN is de-asserted to hold the ZR36016 video output until the Code FIFO is filled. When RTBSY_FB is set, if RTBSY is asserted by the ZR36016 during the active portion of the field, PXEN is de-asserted until RTBSY is de-asserted.

5.2.3.3 RTBSY Signal Control

The ZR36057's RTBSY input is connected to the ZR36016's CBSY output. The ZR36057 uses RTBSY to detect overflow and underflow conditions in the ZR36016 strip memory.

In Motion Video Compression mode, RTBSY asserted at the trailing edge of HSYNC, which indicates a strip buffer overflow, stops the compression process for a field period until the beginning of the following field.

In Motion Video Decompression mode, de-assertion of RTBSY is used to decide when to initiate the decompression process.

In all modes, when RTBSY_FB is set, RTBSY is checked by the ZR36057 at the following times:

- Before the trailing edge of VSYNC, which triggers the start of a new field process in the ZR36016,
- During the active portion of the field.

The ZR36057 de-asserts PXEN if RTBSY is asserted at those times. PXEN is asserted again after RTBSY is de-asserted.

5.2.4 Pixel Transfer In Still Image Compression Mode

The image is transferred by the host software to the ZR36057's extended 24-bit video bus. The host writes the pixels one by one to a dedicated register (the Still Transfer register) in the ZR36057. Each pixel is synchronized with the video clock and transferred to the Video Interface port.

Two modes of host access are supported:

- The host verifies, by polling, the availability of the Still Transfer register before each pixel write access. Typically, this mode would be used after writing the first pixel of each line, before writing the remainder of the line.
- The host configures the WaitState parameter and writes the pixels continuously to the Still Transfer register. The ZR36057 de-asserts the PCI TRDY signal every host access with the timing specified by the WaitState parameter. Typically, this mode would be used to transfer the pixels of a line after the first.

The Still Transfer write protocol and the associated internal mechanism of the ZR36057 are described below:

- The host writes a pixel using a single data phase memory write cycle. The pixel is latched in the Still Transfer register using the PCI clock.
- The Still_Bsy bit is set.
- The incoming pixel is synchronized with the Video Interface clock. The PXEN output signal is asserted and the pixel is driven out on the correct video clock phase.
- Feedback from the Video Interface resets the Still_Bsy bit, indicating that the video port is empty and the following host write is permitted.

5.2.5 Pixel Transfer In Still Image Decompression Mode

The image is transferred from the ZR36057's extended 24-bit video bus to system memory. The host software reads the pixels one by one from the Still Transfer register. After each pixel is fetched from the video port, it is synchronized with the PCI clock, ready to be read by the software.

Before each read access, the host software should check the Still_Bsy bit to verify availability of data in the register. The register contents are valid only if the Still_Bsy bit is 0. Note that by configuring the WaitState parameter, it is possible to ensure that the Still Transfer register will be valid every read access, making it unnecessary to check the Still_Bsy bit.

The protocol for a still image decompression read operation, and the ZR36057's behavior, are as follows:

- The ZR36057 checks the Still_Bsy bit. If it is 1, meaning that the previous valid pixel was read by the host, the ZR36057 fetches a new pixel from the ZR36016.
- PXEN is asserted, causing a new pixel to be driven onto the video bus.
- The new pixel is synchronized with the PCI clock and latched in the Still Transfer register.
- Still_Bsy is reset to 0, to indicate that a new valid pixel is available.
- The host software reads the pixel by means of a memory read access to the Still Transfer register, and Still_Bsy is set to 1.

5.3 GuestBus Interface

The ZR36057 masters a generic MCU-style bus intended to concurrently host up to eight slave devices (referred to as "guests"). The bus consists of 8 data lines (GDAT[7:0]), 3 address lines (GADR[2:0]), 8 active-low chip-select lines (GCS***[7:0]***), read and write signals (GRD, GWR), and a wait-state insertion line (GWS). The bus also includes two interrupt-request inputs (GIRQ[1:0]) and one status/acknowledge input (GRDY). Three types of data transfers are possible on the guest bus. One is a code-write cycle, initiated by the code DMA controller of the ZR36057, targeted to one of the guests, configured a priori for

such write cycles[2]. The second is a GO command to the ZR36050 (configured a priori as one of the guests) in JPEG mode. The third is a PostOffice cycle, initiated by the host software, targeted to any one of the eight guests; in particular, PostOffice cycles are used to program the ZR36016 and ZR36050.

5.3.1 Flexible GuestBus Timing

Different guest devices may have different bus timing requirements. In order to meet these requirements and still master the GuestBus efficiently, the ZR36057 has two timing parameters for each guest:

T_{gdurN} is the duration of a $\overline{GWR}$ or $\overline{GRD}$ signal when accessing guest N.

T_{grecN} is the minimum recovery time in which $\overline{GRD}$ and $\overline{GWR}$ must be non active after the rising edge of the previous access (read or write) to guest N.

T_{gdur} and T_{grec} are configured by the host in units of PCI clock (3,4,12 or 15 PCI clocks are the possible values).

Additional timing parameters are given in section 14.0 “AC Timing Specifications”.

5.3.2 Code-Write Operations

Code-Write cycles are initiated by the GuestBus master if all of the conditions below are met:

- The CFIFO is not empty.
- A PostOffice request is not pending.
- The GRDY input is high (‘1’).

A code-write cycle consists of reading one code byte from the CFIFO and writing it to the guest selected by the host for code-write cycles, to the register address configured by the host. The timing parameters of the code-write cycle are those programmed by the host for this specific guest device. Note that the same parameters apply for PostOffice accesses to this guest.

5.3.3 Doubleword to Bytes Mapping in Code-Write Operations

The code is read in doublewords from main memory, and transferred in bytes to the guest device. The ordering of the bytes is such that the least significant byte of the doubleword is the first one to be sent over the GuestBus, and the most significant byte is the last one.

5.3.4 PostOffice Operations

When the PostOffice pending bit (POPen) in the PostOffice register is set to ‘1’, the GuestBus master completes the current code-write cycle (if such is executed), and executes a PostOffice cycle, even when the Code FIFO is not empty. The type of the cycle (read or write) is determined by the PostOffice direction bit (PODir). The identity of the targeted guest and its specific register are also specified by the PostOffice register. In both read and write cycles the timing parameters of the cycle are those configured by the host for the targeted guest. Upon completion of a PostOffice cycle, the pending bit is reset to ‘0’ by the ZR36057.

PostOffice Write: The GuestBus master transfers the least significant 8 bits of the PostOffice register out on the bus.

PostOffice Read: The GuestBus master reads from the specified target and writes the input byte into the least significant 8 bits of the PostOffice register (POData)

5.3.5 GuestBus Wait States

Slow guests that are equipped with a “bus hold” output can force a code-write or PostOffice GuestBus cycle to be extended by one or more additional PCI clocks, by asserting the $\overline{GWS}$ signal right after the GuestBus master asserts the $\overline{GRD}$ or $\overline{GWR}$ signal. $\overline{GWS}$ is sampled with the PCI clock that follows the assertion of $\overline{GWR}$ or $\overline{GRD}$. When $\overline{GWS}$ is sampled high again, the cycle is completed. If T_{gdur} of the accessed guest has already expired since the assertion of $\overline{GRD}$ or $\overline{GWR}$, the next PCI clock will latch the data (in case of a read) and $\overline{GRD}$ (or $\overline{GWR}$) will be deasserted. The recovery portion of the cycle will then take place, and the cycle will be completed.

Insertion of wait states is possible during both code-write and PostOffice cycles.

The maximum number of PCI clock cycles allowed for $\overline{GRD}$ or $\overline{GWR}$, including wait-states, is 64. If a guest holds the cycle until this limit expires, the GuestBus master aborts the cycle. If the cycle was a PostOffice one, the PostOffice time-out bit of the PostOffice register is set to ‘1’, and the PostOffice pending bit is cleared. If the cycle was a code-write (or code-read, if viewed from the PCI side), the code-write time-out flag (CodTime) is set to ‘1’.

Figure shows two examples of GuestBus cycles. The upper one is a write to guest 0, register 0, followed by a read from guest 0, register 5. Note that for guest 0 T_{gdur0}=3 and T_{grec0}=4. The lower example shows a read from guest 2, register 1, with 3 wait-states inserted by the guest.

Notice that the assertions of GADR and $\overline{GCS}$ are done together. The assertion of $\overline{GRD}$ and $\overline{GWR}$ is done one PCI clock after the assertion of GADR and $\overline{GCS}$. The de-assertion of $\overline{GRD}$ and $\overline{GWR}$ is done one PCI clock before the de-assertion of GADR and $\overline{GCS}$.

2. A typical choice for guest configured a priori for the code-write cycles would be a decompression device, such as the ZR36100/ ZR36110, in the ZR36057’s MPEG mode.

5.4 PostOffice Handshaking Protocol

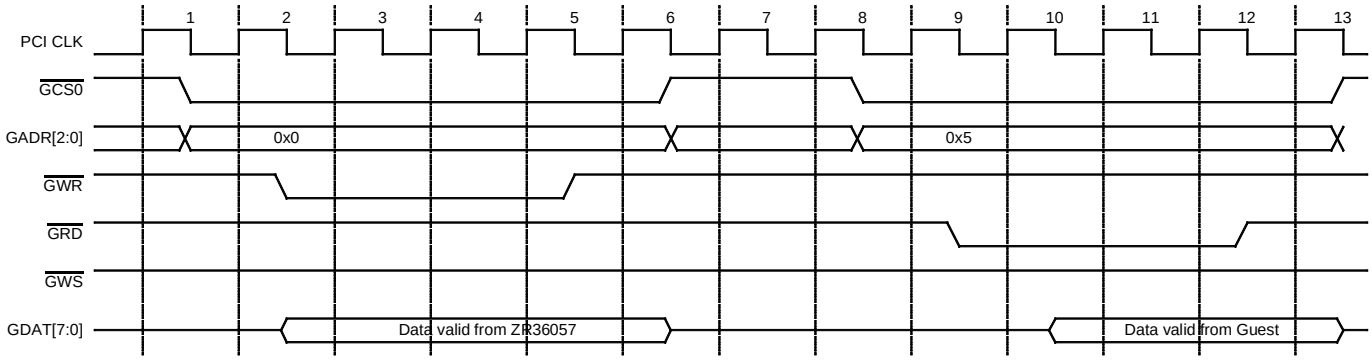
Reading data from or writing data to any of the ZR36057 guests using the PostOffice mechanism requires the host software to follow the handshaking protocol described below. The main idea is that the host has to poll the PostOffice request pending bit in order to confirm the availability of the GuestBus and verify the validity of the data contained in the PostOffice data byte. In general, host accesses to the PostOffice register may change the PostOffice pending bit, as explained below. Thus, the host software must ensure that accesses to the PostOffice register are governed by a central routine. For example, independent accesses to the PostOffice register both from an interrupt service routine and the main processor task(s), or from more than one task in a multitasking environment, might cause a deadlock, unless explicit protection measures are taken.

5.4.1 Host Writes to a Guest Device

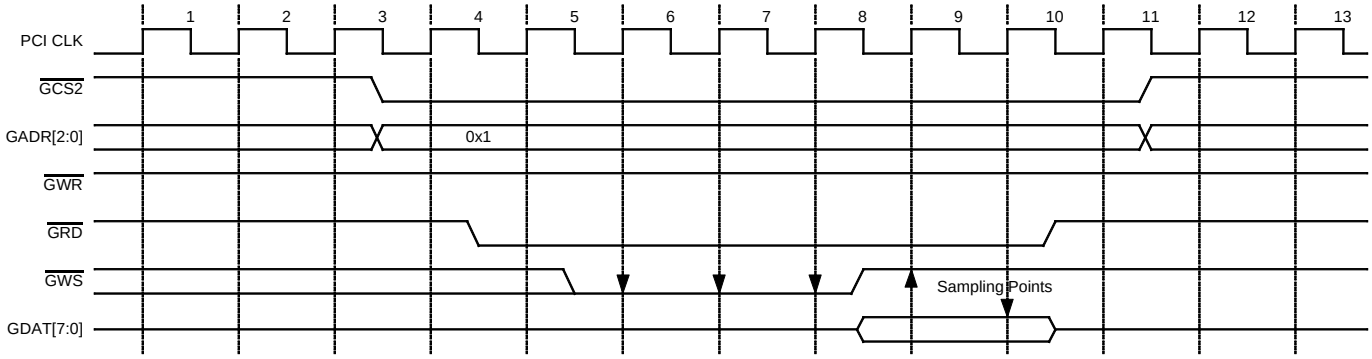
- The host reads the PostOffice register, and checks the PostOffice pending bit. If this bit is '1', the write cycle cannot be executed now, because the GuestBus master is busy executing a previous PostOffice read or write request. Once this bit is '0', the write request can be made.
- The host writes a full doubleword to the PostOffice register, containing the data byte to be sent to the guest, the guest's identity (0,...,7), the specific guest register (0,...,7), and an indication that this is a write request (direction bit = 1). As a result of writing to the PostOffice data byte, the PostOffice pending bit is set to '1'.
- The ZR36057 completes the current code-write cycle, if one is being executed, and, before executing the next code-write cycle (if one is needed), it executes the pending PostOffice request. At the completion of the GuestBus write cycle it clears the request pending bit.
- The host may read the PostOffice register, to verify that the pending bit is '0', meaning that the write request has been completed.
- Note that in multiple (back-to-back) PostOffice operations the host has to poll the request pending bit only once between two requests, since reading this bit zero indicates both that the previous request has been completed and that the next request can be made.

5.4.2 Host Reads from a Guest Device

- The host reads the PostOffice register, and checks the PostOffice pending bit. If this bit is '1', the read cycle cannot be executed now, because the GuestBus master is busy executing a previous PostOffice read or write request. Once this bit is '0', the read request can be made.



Example 1



Example 2

Figure 4. Two Examples of GuestBus Cycles

- The host writes a full doubleword to the PostOffice register, containing the guest's identity (0,..,7), the specific guest register (0,...,7), and an indication that this is a read request (direction bit = 0). The data portion of the doubleword is meaningless, but should be set to a byte of zeros. As a result of writing to the PostOffice data byte, the PostOffice pending bit is set to '1'.
- The ZR36057 completes the current code-write cycle, if one is being executed, and before executing the next code-write cycle (if one is needed), it executes the pending PostOffice request. It transfers the byte read from the guest to bits 7...0 of the PostOffice register. At the completion of the GuestBus read cycle it clears the request pending bit.
- The host may read the PostOffice register, to verify that the pending bit is '0', meaning that the read request has been completed and the data portion of the PostOffice register is the result.
- Note that in multiple (back-to-back) PostOffice operations the host has to poll the request pending bit only once between two requests, since reading this bit zero indicates both that the previous request has been completed and that the next request can be made.

5.5 Codec Bus Interface

The Codec Front End (CFE) interfaces to the ZR36050 JPEG Codec to transfer code (compressed data) to or from the ZR36057. The CFE is designed to operate with the ZR36050 configured in Code Bus Master mode. In Compression mode, the CFE receives the code stream from the codec and transfers it to the Code FIFO. In Decompression mode, the CFE transfers the code stream from the Code FIFO to the codec in response to the codec's read requests.

The CFE uses the CODE[7..0] bus to transfer the code using the handshaking signals $\overline{CCS}$ and $\overline{CBUSY}$, synchronized to VCLKx2. The ZR36057 supports the highest ZR36050 code bus transfer rate, one clock cycle per byte transfer (ZR36050 CFIS parameter = 0).

5.5.1 Compression Mode Code Transactions

The functional timing diagram for compression is shown in Figure 5.

In this example, two code bytes are transferred from the ZR36050 to the ZR36057. The falling edge of $\overline{CCS}$ designates the start of the cycle. The data is driven by the ZR36050 at the rising edge of VCLKx2. The ZR36057's Codec Interface samples the data with VCLKx2 enabled by $\overline{CCS}$.

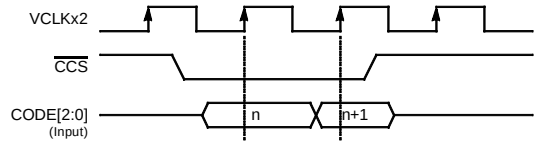


Figure 5. Compression Mode Code Transactions

5.5.2 Decompression Mode Code Transactions

The functional timing diagram for decompression is shown in Figure 6.

In this example, two code bytes are transferred from the ZR36057 to the ZR36050. The falling edge of $\overline{CCS}$ designates the start of the cycle. The data is driven by the ZR36057. The ZR36050 samples the data with the rising edge of $\overline{COE}$ (note that $\overline{COE}$ is not an input to the ZR36057; it is mentioned here only for the completeness of the description).

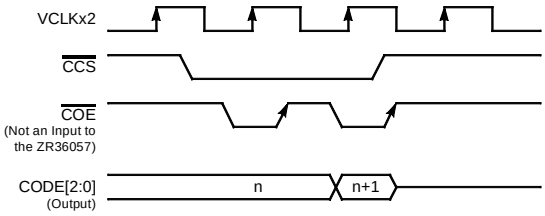


Figure 6. Decompression Mode Code Transactions

5.5.3 Transaction Termination

The ZR36050 is the master of the code transactions in Compression and Decompression modes. Any number of bytes can be transferred in a contiguous burst transaction. The ZR36057 uses the ZR36050's $\overline{CBUSY}$ to block CODE bus transactions, in order to prevent overflow of the Code FIFO in compression or underflow in decompression. Figure 7 shows the functional timing diagram for CODE bus transaction termination.

The ZR36050 examines $\overline{CBUSY}$ one clock cycle prior to the beginning of each access cycle. The ZR36057 asserts $\overline{CBUSY}$ one VCLKx2 cycle before the beginning of the last desired code transfer cycle.

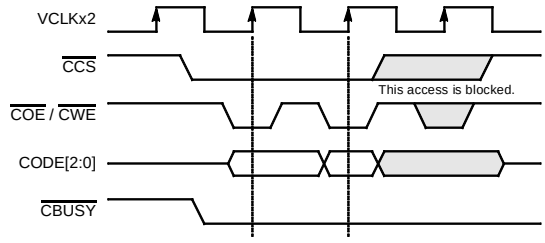


Figure 7. Transaction Termination

5.5.4 Connecting The ZR36057 To The ZR36050

Figure 8 shows the recommended connection of the ZR36057 to the ZR36050.

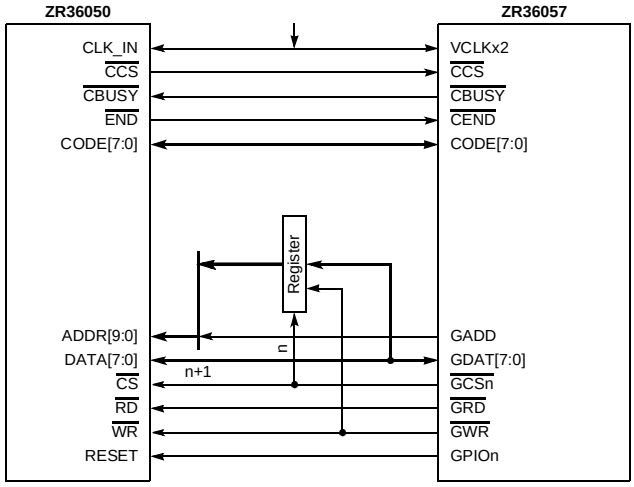


Figure 8. ZR36057 to ZR36050 Connections

5.5.4.1 ZR36050 Host Interface

The host interface of the ZR36050 is controlled by the ZR36057 as one of the guests. The recommended guest timing parameters are Tdur=12, Trec=3. An external register must be added in order to support access to the entire ZR36050 internal memory space. This external register is controlled by the ZR36057 as an additional guest.

One basic host access to the ZR36050 internal memory via the ZR36057 is performed using two successive guest bus accesses. The first is to the external address register during which the GDAT bus holds (for example) the 8 Lsb's of the ZR36050 internal memory address. The second is the command itself during which the GDAT bus holds the data to be written to the ZR36050 internal memory and the 2 Msb's of the address are driven by GADD[1..0].

5.5.4.2 $\overline{CEND}$ Input Signal

Assertion of $\overline{END}$ by the ZR36050 indicates the end of a field process in compression or decompression. This is latched as $\overline{CEND}$ in the Codec Interface.

5.5.4.3 The ZR36050 GO Command

The GO command signals the ZR36050 to start a compression or decompression process. The GO command is performed automatically by the ZR36057, by writing to address 0x00h of the ZR36050 internal memory. The host software is responsible for pre-loading this address to the external address register after the completion of ZR36050 initialization.

The ZR36057 uses the JPEGuestID and the JPEGuestReg parameters for the GO command guest address.

In Motion Video Compression mode, the first GO command is initiated by the host after finishing the initialization of all the peripheral devices. After the first field, the GO command is executed after the code of the previous field has been transferred to the system memory.

In Decompression mode, the GO command is executed only after the Code FIFO has been filled from the relevant system memory code buffer.

Important Note:

The automatic GO command is enabled by the Go_En register bit, in the JPEG Mode and Control Register. If the host software tries to execute a PostOffice cycle while Go_En is set to '1' (enabled), the PostOffice cycle may not be executed correctly. Therefore, Go_En must be disabled during PostOffice accesses. The following is the sequence of actions required for a Post Office access or series of accesses:

- Disable the Go_En bit.
- Wait at least 15 PCI clock cycles to allow the currently executing GO command, if there is one, to be completed.
- Execute the PostOffice access, or the series of accesses.
- Enable the Go_En bit (at least for 5 PCI clocks).

If an internal GO command is requested while Go_En is disabled, it is stored internally, and executed when Go_En is enabled again.

Note that there is a trade-off between the "chunk" size of a series of PostOffice accesses and the delay imposed on the ZR36050 GO command. Using a large chunk size reduces the overhead resulting from the need to toggle Go_En, but delaying the GO command too much can prevent the next field process from executing correctly.

5.5.4.4 $\overline{CCS}$ Input Signal

The ZR36050 $\overline{CCS}$ signal designates the start of a code transaction. It stays active until the end of the transaction. In Compression mode, the ZR36057 uses $\overline{CCS}$ to enable the sampling of the incoming code stream. In Decompression mode, it uses $\overline{CCS}$ to enable the drive of the code stream.

5.6 I2C Bus Interface

The I^2C port of the ZR36057 consists of a clock signal, SCK, and data signal, SDA. Both have two possible levels: active low or passive tri-state. This configuration lets the ZR36057 be the only master of an I^2C clock. Both lines must be pulled-up externally. By accessing the I^2C application-specific register bits appropriately, the host software can generate valid I^2C start and stop conditions, write address and write or read data one bit at a time.

5.7 General Purpose I/O Pins

The ZR36057 has 8 general purpose I/O pins, fully controlled by the host. Each of these pins can be separately configured as

input or output. When configured as an output, the host is able to force the level of a pin through its corresponding register bit.

5.8 Interrupt Requests

The ZR36057's interrupt manager connects to the various conditions that may generate an interrupt request, enables or disables them as specified in the Interrupt Control Register, and drives the $\overline{\text{INTA}}$ output. It stores the corresponding status bits in the Interrupt Status Register, and clears the status bits per host instructions.

The **ZR36057** can associate any one of the following events with an interrupt request:

- A positive edge on the GIRQ1 input pin.
- A positive edge on the GIRQ0 input pin.
- MPEG mode - the code memory buffer pointer passing one of its report points.
- JPEG modes - successful completion of a JPEG field or frame process (in compression or decompression).

Each one of these events can be separately enabled or disabled through the corresponding bit in the Interrupt Control Register. An additional global enable bit enables or disables all interrupts.

When an interrupt-associated event occurs, two things happen:

- The corresponding bit in the Interrupt Status Register is set.
- If the interrupt is enabled, and the interrupts are enabled globally, then the $\overline{\text{INTA}}$ open-drain output pin is asserted to its active-low level.

Both the status bit(s) and $\overline{\text{INTA}}$ remain active, until the host clears those status bits that are currently set. This is done by writing a '1' to those bits. When the host does that, the $\overline{\text{INTA}}$ output signal returns to its passive, tri-state level.

If the host attempts to clear any of the Interrupt Status Register bits at the same time that the interrupt logic attempts to set it (because of an interrupt event), the set operation has priority over the clear operation.

6.0 VIDEO INPUT PROCESSOR

6.1 Horizontal Filter

Prior to a significant horizontal down scaling of the input image, it is advisable to apply one of the possible horizontal filters. The filter type is selected through the HFilter parameter.

HFilter = 0	Filter 1: No luminance filter, 3-tap pre-interpolation filter of chrominance.
HFilter = 1	Filter 2: 3-tap luminance filter, 3-tap pre-interpolation chrominance filter.
HFilter = 2	Filter 3: 4-tap luminance filter, 4-tap chrominance filter.
HFilter = 3	Filter 4: 5-tap luminance filter, 4-tap chrominance filter.
HFilter = 4	Filter 5: 4-tap luminance filter, 4-tap chrominance filter.

6.2 Horizontal/Vertical Downscaler

The horizontal and vertical down scalers are independent of each other. The horizontal scaling ratio is configured through the HorDcm parameter. HorDcm indicates the number of pixels to drop out of every 64 consecutive pixels. HorDcm ranges from 0 to 63, where 0 represents the no scaling configuration (1:1 input to output ratio).

The vertical scaling ratio is configured through the VerDcm parameter. VerDcm indicates the number of lines to drop out of every 64 consecutive lines. VerDcm ranges from 0 to 63, where 0 represents the no scaling configuration (1:1 input to output ratio).

The vertical downscaler can operate in two ways. If DupFld=0 it treats the top and bottom fields the same way. If DupFld=1 it uses different line dropping topologies for the top and the bottom fields, such that if the fields are equal (one field is actually duplicated, like the output of most MPEG-1 decoders), then the total loss of information is minimized. For example, when the video source is a video digitizer, it is recommended to apply DupFld=0, and when the video source is the ZR36110, and the CCIR size is down scaled by half or more, it is recommended to apply DupFld=1.

6.3 Color Space Converter

The color space converter converts the YUV input to RGB format. The YUV2RGB parameter determines the type of conversion:

YUV2RGB = 00b	no conversion, output format is YUV 422.
YUV2RGB = 01b	conversion to RGB 8,8,8 (24-bit output)
YUV2RGB = 10b	conversion to RGB 5,6,5 (16-bit)
YUV2RGB = 11b	conversion to RGB 5,5,5 (15-bit)

When the 15- or 16-bit RGB format is selected, it is advisable to apply the error diffusion option, in order to eliminate false contours from the output image. This option is selected by the ErrDif parameter (1 turns the error diffusion option on, while 0 turns it off).

7.0 VIDEO OUTPUT CONTROL

The ZR36057 outputs the video pixels over the PCI bus, using DMA bursts, initiated and controlled by the ZR36057's Video DMA Controller. In order to enable the DMA Controller, the Master Enable bit of the PCI configuration space must be set to '1', and the VidEn bit in the Video Display Configuration Register must also be set to '1'. Once VidEn=='1', the software is not allowed to change registered parameters that are involved in the video processing. The register description (See "Application-Specific Registers (ASRs)" on page 25) specifies the conditions under which each parameter is allowed to be modified.

The ZR36057 transfers the video to a rectangle in the display (or system) memory, defined by a base address for each field (MaskTopBase, MaskBotBase), an inter-line stride (DispStride), and the rectangle height (VidWinHt) and width (VidWinWid). Obviously, these parameters must be provided by the host prior to enabling the Video DMA Controller.

7.1 Display Modes

The ZR36057 can either display both fields, emulating the interlaced input, or only the top field. The latter option has the advantage of reducing the motion artifacts that might be exhibited when interlaced video is diplayed on a non-interlaced monitor. The parameter that controls the display mode is DispMod.

By a proper configuration of the display base addresses it is also possible to display two fields (from either one or two separate video sources) on two separate rectangles (video windows).

7.2 Frame Grabbing

The ZR36057 has a special mode for capturing video frames (or fields) and storing them in system memory. This mode is invoked by setting the SnapShot parameter to '1'. When in this mode, every time the host switches the FrameGrab bit from '0' to '1', the ZR36057 downloads a frame (or a field, if DispMod==0), to memory.

Following is an example of a flow of actions intended to grab one frame. The example assumes that the vertical sync is used as an interrupt source (by externally tying VSYNC to GIRQ0 or GIRQ1), and that prior to grabbing the frame, the ZR36057 operates in the "normal" continuous scheme of live video display.

- Through a push-button click in the application GUI the user triggers a frame grabbing request.
- The host sets SnapShot=1. The ZR36057's Video DMA Controller waits for the next VSYNC and then freezes the live display. Since now SnapShot=1 and FramGrab=0, video parameters can be changed (even without VidEn=0; refer to section 12.0 "Application-Specific Registers (ASRs)").
- The host sets new addresses in VidTopBase and VidBotBase. These addresses point to main memory. DispStride is also given a new value. If needed, other video parameters can be changed now, e.g., pixel format, etc.
- The host sets FrameGrab=1. The ZR36057 waits until the next VSYNC and then transmits two consecutive fields to main memory.
- After the second of the two fields is completed, FrameGrab is cleared by the ZR36057.
- When the host senses (after constant polling or polling inside VSYNC-triggered interrupts) that FrameGrab=0 again, it sets the old addresses back to VidTopBase and VidBotBase. DispStride is given back its old value. The remainder of the previous video parameters can be restored now.
- The host sets SnapShot=0, putting the ZR36057 back into the continuous video display mode.
- With the next VSYNC the ZR36057's Video DMA Controller resumes "normal" live display operation.

7.3 Output Pixel Organization

The output pixel format is determined by the following parameters: YUV2RGB, Pack24 (applicable only to RGB 8,8,8 format), and LittleEndian (applicable to all formats, excluding the 24-bit packed). Following are the bit organizations of the different pixel formats when a video doubleword is transferred over the PCI bus:

Table 4: YUV 4:2:2 Pixel Format

	Bits			
Endian-ness	**31...24**	**23...16**	**15...8**	**7...0**
Little Endian	Y17...0	V07...0	Y07...0	U07...0
Gib Endian	V07...0	Y17...0	U07...0	Y07...0

Table 5: RGB 5,5,5 Pixel Format

	Bits			
Endian-ness	**31...24**	**23...16**	**15...8**	**7...0**
Little Endian	0,R14...0, G14...3	G12...0, B14...0	0,R04...0, G04...3	G02...0, B04...0
Gib Endian	G12...0, B14...0	0,R14...0, G14...3	G02...0, B04...0	0,R04...0, G04...3

Table 6: RGB 5,6,5 Pixel Format

	Bits			
Endian-ness	**31...24**	**23...16**	**15...8**	**7...0**
Little Endian	R14...0, G15...3	G12...0, B14...0	R04...0, G05...3	G02...0, B04...0
Gib Endian	G12...0, B14...0	R14...0, G15...3	G02...0, B04...0	R04...0, G05...3

Table 7: 24-bit Unpacked Pixel Format

Endian-ness	Bits			
	31...24	23...16	15...8	7...0
Little Endian	0x0	R7...0	G7...0	B7...0
Gib Endian	B7...0	G7...0	R7...0	0x0

In the 24-bit packed format the first active pixel in a line is always packed as indicated in the “First” row of Table 8. From then on the byte organization is as described by the table.

Table 8: 24-bit Packed Pixel Format

Bus Cycle	Bits			
	31...24	23...16	15...8	7...0
First	B17...0	R07...0	G07...0	B07...0
Second	G27...0	B27...0	R17...0	G17...0
Third	R37...0	G37...0	B37...0	R27...0

8.0 GRAPHICS OVERLAY

The ZR36057’s Video DMA Controller is capable of masking off (i.e., not transmitting) pixels that are marked by ‘0’ in a masking map prepared and maintained by the driver software. The masking feature, referred to as overlay or clipping, is turned on by setting the OvlEnable parameter to ‘1’. As long as OvlEnable=’0’, all pixels within the selected portion of the image are transferred to destination. The masking map is a one bit deep map of the video rectangle. Its location in system memory is defined by a pair of base addresses (one for each field - MaskTopBase, MaskBotBase), and an inter-line stride (MaskStride). The width of the map must be doubleword aligned. Thus, the line size is:

int((VidWinWid+31) >> 5)

and the number of lines in the map is:

2*VidWinHt, if DispMod==0, or VidWinHt, if DispMod==1.

In order to match the 0/1 values of the map to their corresponding pixels in the video rectangle, the map must follow the format given in Table 9.

Table 9: Bit, Byte and Doubleword Organization of the Masking Map

DWORD	...1	0			
Byte	...0	3	2	1	0
Bit	7....0	31...24	23...16	15... 8	7...0
Pixel index in line	39...32	31...24	23...16	15...8	7...0

9.0 JPEG CODE TRANSFER

The data flow direction depends on the JPEG mode of operation. In Motion Video and Still Image Compression, the code is transferred from the CODE bus to the system memory. In Motion Video and Still Image Decompression, the code is transferred from the system memory to the CODE bus.

The compressed data in system memory is structured within code buffers. Each code buffer may contain a compressed field or frame (two fields) as specified by a user-configurable bit.

The ZR36057 supports four code buffers, defined dynamically in a dedicated table, the Code Buffer Table, in the system memory. Before starting a new JPEG process, the host must load the physical address of the code buffer table into the I_STAT_COM_PTR register of the ZR36057.

The actual memory allocated to each code buffer by the operating system may be fragmented. The content of each entry in the code buffer table is a pointer to a secondary Fragment Table. The fragment table contains the pointers to the allocated memory chunks.

Figure 9 provides a graphical description of the data structure in the system memory.

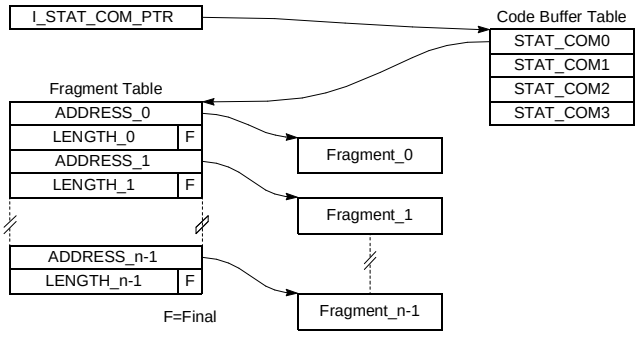


Figure 9. JPEG Code Data Structure In System Memory

9.1 The Code Buffer Table

The Code Buffer Table consists of four STAT_COM (“status or command”) entries, one for each code buffer. The interpretation of each STAT_COM entry is determined by its l.s. bit, the STAT_BIT. If STAT_BIT=’0’, the content of the entry is the “com-

mand" information, written by the host, for the next available buffer. Table 10 shows the interpretation for this case.

Table 10: STAT_COM entry content, when STAT_BIT=0

Bit	Content	Description
31:2	FRAG_TAB_PTR[31:2]	The address pointer to the fragment table of the next buffer to be used. The 2 l.s. bits of the address must be '0' - that is, the address must be doubleword aligned.
1	FRAG_TAB_PTR [1]	Must to be '0'.
0	STAT_BIT	Must be '0' to indicate that the table entry is a command.

If the STAT_BIT='1', the content of the entry is the "status" information, written by the ZR36057, of the most recently processed field or frame. Table 11 shows the interpretation for this case.

Table 11: STAT_COM entry content, when STAT_BIT=1

Bit	Content	Description
31:24	F_CNT[7:0]	The serial number, modulo 256, of the most recent field or frame. Used in compression only.
23		Reserved, set to '0'.
22:1	F_LENGTH[21:0]	The length of the most recently compressed field or frame in bytes, used in compression only. The length is doubleword aligned, so bits 2..1 are always '0'.
0	STAT_BIT	'1', indicating that this table entry is a status written by the ZR36057.

9.2 Fragment Table

The Fragment Table defines the structure of the allocated memory for each code buffer. The table has N entries for N allocated fragments. Each entry holds two doublewords:

- first doubleword - the physical address of the fragment (must be doubleword aligned),
- second doubleword - the fragment length in doublewords (must be doubleword aligned).

The l.s. bit of the second doubleword is the FINAL bit, indicating (when '1') that the current fragment entry defines the last fragment of the buffer.

9.3 JPEG Compression Modes

The sequence of actions performed by the host software and the ZR36057 is as follows:

- The host allocates the first four code buffers and writes out the code buffer table and the fragment table.
- The host loads the ZR36057's I_STAT_COM_PTR register with the base address of the code buffer table.
- After the start of the compression process, the ZR36057 starts filling the fragments of the first code buffer with the compressed data of the first field or frame.
- After successfully completing transfer of the compressed data of a field or frame, the ZR36057 writes the status information back to the STAT_COM entry of the current buffer. It sets the STATUS_BIT in the STAT_COM entry, declaring its content as status. It then issues an interrupt, and starts the next field or frame process.
- If a new code buffer is unavailable at the beginning of a field or frame process (that is, if the STATUS_BIT of the next buffer entry is '1'), the ZR36057 polls the STAT_COM entry until the buffer is available.
- When the host receives the interrupt, it checks the STAT_COM entries in the code buffer table. For each entry whose STAT_BIT='1', the host checks F_CNT to determine whether any fields/frames were dropped. It records the code buffer information, updates the fragment table and STAT_COM entry of the code buffer, and returns it by resetting STAT_BIT to '0'.

Several scenarios might cause a field or frame process to be unsuccessful:

- The allocated code buffer memory was smaller than the actual field/frame code volume (the ZR36057 fills the final fragment, but the field/frame code transfer is not complete).
- A ZR36016 strip memory overflow occurs during the field/frame compression (might be caused by very large PCI bus latency)

After an unsuccessful field/frame process in compression, the ZR36057 does not issue an interrupt, and restarts with the next field or frame.

9.4 JPEG Decompression Modes.

The sequence of actions performed by the host software and the ZR36057 is as follows:

- The host allocates the first four code buffers and writes out the code buffer table and the fragment table.
- The host fills the allocated fragments with the code to be decompressed.
- The host loads the ZR36057's I_STAT_COM_PTR register with the base address of the code buffer table.
- After the start of the decompression process, the ZR36057 starts reading the code of the first field or frame from the code buffer fragments.
- After completing decompression of the field or frame, the ZR36057 sets the STATUS_BIT in the STAT_COM register,

declaring that this buffer was already decompressed, issues an interrupt, and starts the next field/frame process. It reads the next STAT_COM entry from the code buffer table. If the buffer is available, that is, the STAT_BIT='0' indicating it contains the code of a new field/frame, the ZR36057 starts transferring its fragments for decompression.

- If a new code buffer is unavailable at the beginning of a field or frame process (that is, if the STATUS_BIT of the next buffer entry is '1'), the ZR36057 repeats decompression of the last available code buffer.
- When the host receives the interrupt, it checks the STAT_COM entries in the code buffer table. For each entry whose STAT_BIT='1', the host loads compressed data into a new code buffer and updates its fragment table, then returns the buffer by resetting its STAT_BIT to '0'.

Important Note:

In Motion Video Decompression mode, if the ZR36057 starts a new fragment when the Code FIFO is exactly full, the Code FIFO can sometimes overflow.

There are two possible ways to deal with this problem:

1. Copying the fragmented buffer into a single-fragment buffer before playing it back.

Implementing this alternative may cause some performance degradation due to the additional memory-to-memory copying.

2. Configuring the Code FIFO threshold to be low enough.

If the Code FIFO threshold is low enough, the Code FIFO will never reach fullness. The maximum length of a single PCI burst is limited, and after the Code FIFO gets above its threshold, the PCI bus is not requested again.

The recommended parameters of the ZR36057 for this work-around option are:

- The Triton bit (in the Video Display Configuration Register) should be 0. This bit causes the ZR36057's $\overline{\text{REQ}}$ signal to be de-asserted immediately after the bus was granted. Typically (on machines with Triton PCI chip set), this causes the $\overline{\text{GNT}}$ signal to be de-asserted. The ZR36057's burst length is then determined by the Latency Timer parameter.
- Master Latency Timer = 48 (in the PCI Configuration Space Register.
- JPEGCodTrshold (in the JPEG Code FIFO Threshold Register = 20.

10.0 RESET

There are three means of resetting the *ZR36057*. One is a hardware reset, which is applied through the $\overline{\text{PCIRST}}$ input, the second is a software reset, which is applied through the SoftReset register bit. The third is the JPEG Process Reset asserted by the P_reset register bit.

10.1 Hardware Reset

The hardware reset signal $\overline{\text{PCIRST}}$ resets the internal state machines in the ZR36057 and loads all registers with their default states. The reset state of the PCI interface pins is as defined by the PCI specifications (2.1). The reset state of the other output/bidirectional signals is as follows.

The GPIO[7:0] lines are all inputs after reset. If required for system purposes, they can be pulled high or low through 1 K external resistors to have fixed values on reset.

As long as $\overline{\text{PCIRST}}$ is asserted, the following signals are tri-stated: GDAT[7:0], GADR[2:0], $\overline{\text{GCS}}$[3:0], $\overline{\text{GWR}}$, $\overline{\text{GRD}}$, SDA, SCL. Once the $\overline{\text{PCIRST}}$ input is deasserted, these signals go to their software reset condition (as does the entire device).

A hardware reset asserts (clears) the SoftReset bit in the system register. After the hardware reset is over, the *ZR36057* will be in software reset condition until the SoftReset bit is deasserted.

10.2 Software Reset

There are two ways in which the ZR36057 can go into the software reset condition: one is right after hardware reset (i.e., upon the low to high transition of $\overline{\text{PCIRST}}$), the other is by clearing the SoftReset bit.

While in software reset, all registers and state machines in the device are reset to their default values/states, except the SoftReset bit itself, and the PCI Interface (including the PCI configuration space registers).

The device continues to respond according to the PCI Specification and can be the target of a PCI transfer targeted at the ASRs (Application Specific Registers) or PCI Configuration Space. While in software reset the device will not initiate any PCI transfers, because all DMA channels are disabled.

After the SoftReset bit is deasserted, all registers retain their default values, all DMA channels remain disabled and all ASRs are programmable according to their “normal” modification conditions.

Hence, the initialization of the device (loading all registers with the values required for the specific application) must start with setting SoftReset to ‘1’, otherwise new ASR values will not be latched in.

10.3 JPEG P_reset

The P_reset bit resets all of the JPEG related state machines and controls in the ZR36057.

The P_reset bit enables re-configuration of all JPEG parameters with no effect on the other functions of the device.

Before starting a new JPEG process, and while loading new JPEG parameters, the P_reset bit must be asserted.

11.0 PCI CONFIGURATION SPACE REGISTERS

The PCI specification requires that a PCI device include a configuration register space, a set of 256 8-bit configuration registers. The first 64 bytes make up the configuration header, predefined by the specification and the remainder are application specific. These registers allow device relocation, device independent system address map construction and automatic configurations. The configuration registers are accessed by the host software through two consecutive doubleword I/O accesses to physical addresses 0x0CF8 and 0x0CFC, respectively. The host PCI bridge is responsible for translating these accesses to a PCI configuration cycle, including the assertion of the IDSEL input. The ZR36057 then responds to these cycles. This section details the ZR36057's PCI configuration registers. PCI configuration accesses to ZR36057 configuration addresses that are not explicitly described here return zeros (in reads).

Table 12: ZR36057 PCI Configuration Space Registers

Address Offset	Bits	Type	Description
0x00	31:16	R	Device ID. Hardwired to 0x6057.
	15:0	R	Vendor ID. Hardwired to 0x11DE.
0x04	31	RC	Parity Error Detected. This bit is set when a parity error is detected, regardless of the Parity Error Response bit.
	30	R	System Error Signaled. Hardwired to '0'.
	29	RC	Master Abort Detected. This bit is set when a master-abort condition has been detected.
	28	RC	Target Abort Detected. This bit is set when a target-abort condition has been detected.
	27	RC	Target Abort Signaled. When the ZR36057 terminates a transaction as a target (e.g., due to wrong address parity) it sets this bit.
	26:25	R	$\overline{\text{DEVSEL}}$ Timing. Hardwired to '00' ("fast" timing, i.e., $\overline{\text{DEVSEL}}$ is asserted before rising edge of clock three within a cycle).
	24	R	Data Parity Reported. Hardwired to '0'.
	23	R	Fast Back-to-Back Capability. Hardwired to '0'.
	22:16	R	Reserved. Returns zeros.
	15:10	R	Reserved. Returns zeros.
	9	R	Fast Back-to-Back Enable. Hardwired to '0'.
	8	R	System Error Enable. Hardwired to '0'.
	7	R	Wait Cycle (Stepping) Enable. Hardwired to '0'.
	6	R	Parity Error Response. Hardwired to '0'.
	5:3	R	Unused. Hardwired to '0'.
	2	RW	Master Enable. When this bit is set to '1' the ZR36057 can operate as a bus master. Default is '0'.
	1	RW	Memory Access Enable. When this bit is set to one the device responds to PCI memory accesses. Default value is '0'.
	0	R	I/O Access Enable. Hardwired to '0'.
0x08	31:8	R	Class Code. Returns 0x040000 (Multimedia Video Device)
	7:0	R	Revision ID. Hardwired to 0x01
0x0C	31:24	R	Unused. Return zeros.
	23:16	R	Header Type. Returns zeros.
	15:8	RW	Master Latency Timer. The number of PCI clocks that limit ZR36057-initiated bursts in case $\overline{\text{GNT}}$ is deasserted by the bus arbiter during the ZR36057-initiated burst. The 3 LS bits are read-only zeros. The default value is 0x00.
	7:0	R	Unused. Returns zeros.
0x10	31:12 11:0	RW R	Memory Base Address. This value determines the base address of the ZR36057 as a memory-mapped device. The ZR36057 occupies a range of 4096 bytes out of the memory map: Bits 11:0 are hardwired to '0'. The default value of all other bits is '0'.
0x14 to 0x3B		R	Hardwired to '0'.

Table 12: ZR36057 PCI Configuration Space Registers

Address Offset	Bits	Type	Description
0x3C	31:24	R	Max_Lat - Hardwired to 0x10 (i.e., 4 µs). This value indicates for the operating system how often the device needs access to the PCI bus.
	23:16	R	Min_Gnt - Hardwired to 0x2 (i.e., 0.5us). Indicates to the operating system the minimum length of a burst.
	15:8	R	Interrupt pin. Hardwired to 0x1, indicating that $\overline{\text{INTA}}$ is used.
	7:0	RW	Interrupt Line. These bits indicate the interrupt line that is being used (e.g., IRQ10 ==> 0xA, etc.). Default value is 0xA.

12.0 APPLICATION-SPECIFIC REGISTERS (ASRS)

The ZR36057 application-specific registers (ASRs) are memory-mapped. Their base address is configured by the host into PCI configuration address 0x10. The ZR36057 claims a contiguous range of 4K bytes of memory. PCI memory-read accesses to addresses (within the 4K range) that are not explicitly described in this section return zeros.

The ASRs can be accessed in any byte combination.

The column Mod of the following tables defines the conditions under which each parameter of the ASRs is allowed to be modified by the host software. The following abbreviations are used:

Abbreviation	Description
all	This parameter may be modified on the fly, i.e. any time.
res	This parameter is set once after a reset of the ZR36057, no modifications allowed during operation.
vid	This parameter may be modified if either VidEn = '0' or SnapShot = '1' and FrameGrab = '0'.
cod	This parameter may be modified if CodReadEn = '0'.
snap	This parameter may be modified if SnapShot = '1'.
jpg	This parameter may be modified if P_reset = '0'.

Note that after a hard or soft reset VidEn = '0', CodReadEn = '0', and P_reset = '1'.

12.1 Video Front End Horizontal Configuration Register

This 32 bit register contains the horizontal configuration parameters of the video source.

Address Offset: 0x000

Bit	Type	Mod	Description
31	R		Reserved. Returns zero.
30	RW	vid	**HSPol** - HSYNC Polarity. HStart and HEnd are counted from the active edge of HSYNC. '1' - negative edge of HSYNC. '0' - positive edge of HSYNC (default value).

Address Offset: 0x000 (Continued)

Bit	Type	Mod	Description
29:20	R		Reserved. Returns zero.
19:10	RW	vid	**HStart** - Horizontal Start Offset. Number of pixel clocks in a line from the active edge of HSYNC until the first pixel to be sampled. Default value is 0x001.
9:0	RW	vid	**HEnd** - Horizontal End Offset. Number of pixel clocks in a line from the active edge of HSYNC until the last pixel to be sampled. Default value is 0x3FF.

12.2 Video Front End Vertical Configuration Register

This 32 bit register contains the vertical configuration parameters of the video source.

Address Offset: 0x004

Bit	Type	Mod	Description
31	R		Reserved. Returns zero.
30	RW	vid	**VSPol** - VSYNC Polarity. VStart and VEnd are counted from the active edge of VSYNC. '1' - negative edge of VSYNC. '0' - positive edge of VSYNC (default value).
29:20	R		Reserved. Returns zero.
19:10	RW	vid	**VStart** - Vertical Start Offset. Number of lines from the active edge of VSYNC until the first line to be sampled. Default value is 0x001.
9:0	RW	vid	**VEnd** - Vertical End Offset. Number of lines from the active edge of VSYNC until the last line to be sampled. Default value is 0x3FF.

12.3 Video Front End, Scaler and Pixel Format Register

This register contains the video front end configuration (byte 3), video scaler (bytes 2-1) and pixel formatter (byte 0) parameters

Address Offset: 0x008

Bit	Type	Mod	Description
31:27	R		Reserved. Returns zero.
26	RW	vid	**ExtFI** - External Field Indication. '1' - the video source provides an FI signal. '0' - field indication is derived from HSYNC and VSYNC (default value).
25	RW	vid	**TopField** - Top Field Interpretation. If field indication is derived from the FI input signal (see ExtFI), then this parameter defines which level of FI indicates the top field: '1' - FI high indicates the top field (default value). '0' - FI low indicates the top field. If field indication is derived internally from HSYNC and VSYNC, then this parameter defines which level of HSYNC indicates the top field on the active edge of VSYNC: '1' - HSYNC high indicates the top field (default value). '0' - HSYNC low indicates the top field.
24	RW	vid	**VCLKPol** - Video Clock Polarity. When set to '0' (default value), the video interface inputs are sampled by the positive edge of VCLKx2 that is qualified by VCLK = '0'. The video interface outputs (other than $\overline{\text{PXEN}}$) are driven at the negative edge of VCLKx2 which is qualified by VCLK = '1'. When set to '1', the video interface inputs are sampled by the positive edge of VCLKx2 that is qualified by VCLK = '1'. The video interface outputs (other than $\overline{\text{PXEN}}$) are driven at the negative edge of VCLKx2 which is qualified by VCLK = '0'.
23:21	RW	vid	**HFilter** - Horizontal Filter Selection. The following horizontal filters can be selected: 000b - Filter 1 = no luminance, 3-tap chrominance 001b - Filter 2 = 3-tap luminance, 3-tap chrominance 010b - Filter 3 = 4-tap luminance, 4-tap chrominance 011b - Filter 4 = 5-tap luminance, 4-tap chrominance 100b - Filter 5 = 4-tap luminance, 4-tap chrominance 101b - 111b will result in the default filter. Default value 000b.

Address Offset: 0x008 (Continued)

Bit	Type	Mod	Description
20	RW	vid	**DupFld** - Duplicate Field. This parameter has an effect on the vertical decimation of a frame in respect to the video source. The result is less loss of information and less distortion. '1' - indicates top and bottom fields are equal. '0' - indicates an interlaced video source, top and bottom fields are different (default value).
19:14	RW	vid	**HorDcm** - Horizontal Decimation Ratio. This parameter defines the number of pixels to be dropped out of every 64 consecutive pixels. 000000b - no horizontal decimation (default value).
13:8	RW	vid	**VerDcm** - Vertical Decimation Ratio. This parameter defines the number of lines to be dropped out of every 64 consecutive lines. 000000b - no vertical decimation (default value).
7	R		Reserved. Returns zero.
6	RW	vid	**DispMod** - Display Mode. '1' - single field. '0' - emulated interlaced video (default value).
5	R		Reserved. Returns zero.
4:3	RW	vid	**YUV2RGB** - YUV to RGB Conversion. This parameter defines the pixel output format: 00b - YUV 4:2:2, 01b - RGB 8,8,8, 10b - RGB 5,6,5 (default value), 11b - RGB 5,5,5.
2	RW	vid	**ErrDif** - Error Diffusion. This parameter has an effect only on RGB 5,6,5 and 5,5,5 output formats. '1' - error diffusion is active '0' - simple truncation of RGB 8,8,8 is executed (default value).
1	RW	vid	**Pack24** - 24 Bit Packed Format. This bit is applicable only if YUV2RGB = 01b. '1' - RGB 8,8,8 is packed such that it takes 3 PCI cycles to transfer 4 pixels. '0' - one pixel per PCI cycle is transferred (default value).
0	RW	vid	**LittleEndian** - Little Endian Format flag. '1' - pixel layout on PCI is little endian (default value). '0' - pixel layout on PCI is gib endian.

12.4 Video Display "Top" Register

This register contains the doubleword base address of the top field.

Address Offset: 0x00C

Bit	Type	Mod	Description
31:2 1:0	RW R	vid	**VidTopBase** - Video Top Field Base Address. This is the destination starting address for the top field. Default value is 0xFFFFFFFC. Bits 1..0 are hardwired to 00b.

12.5 Video Display "Bottom" Register

This register contains the doubleword base address of the bottom field.

Address Offset: 0x010

Bit	Type	Mod	Description
31:2 1:0	RW R	vid	**VidBotBase** - Video Bottom Field Base Address. This is the destination starting address for the bottom field. Default value is 0xFFFFFFFC. Bits 1..0 are hardwired to 00b.

12.6 Video Stride, Status and Frame Grab Register

This register contains parameters for display addressing (bytes 2-3), status of VFIFO (byte 1) and frame grab control (byte 0).

Address Offset: 0x014

Bit	Type	Mod	Description
31:18 17:16	RW R	vid	**DispStride** - Display Stride. This register defines the address increment in bytes to be added to the address of the last pixel of a display line, to generate the address of the next consecutive display line. If the address difference between two consecutive display lines is zero (i.e, they are physically consecutive) than DispStride should be set (by the driver software) to zero (if DispMode=1) or to the display line size in bytes (if DispMode=0). Default value is 0xFFFC. Bits 1..0 are hardwired to 00b.
15:9	R		Reserved. Returns zero.
8	RC	all	**VidOvf** - Video FIFO Overflow flag. This bit is asserted by the Video FIFO server when an overflow of the Video FIFO occurs. This bit is cleared when the host tries to write '1' to it. In case of concurrent accesses to this bit, it remains '1'. '1' - a VFIFO overflow occurred. '0' - no overflow (default value).
7:2	R		Reserved. Returns zero.

Address Offset: 0x014

Bit	Type	Mod	Description
1	RW	all	**SnapShot** - Frame Grab Mode. If this bit is asserted the ZR36057 goes into frame grab mode. When deasserted continuous display of video is resumed. '1' - frame grab mode. '0' - continuous display mode (default value).
0	RS	snap	**FrameGrab** - Frame Grabbing Command/ Status. When this bit is asserted by the host and SnapShot is asserted, the ZR36057 will transfer the next two fields (indicated by the VSYNC signal) to memory. At the end of the second field this bit will be cleared internally, indicating that the frame grabbing has been completed and video transfer has been stopped. In case of concurrent accesses to this bit, the result is '0'. '1' - start frame grabbing. '0' - frame grabbing completed (default value).

12.7 Video Display Configuration Register

This register contains the configuration parameters for the video display.

Address Offset: 0x018

Bit	Type	Mod	Description
31	RW	all	**VidEn** - Video Display Enable. If this bit is cleared by the host, video write DMA transfers are disabled. When enabled, the video DMA controller operates normally. '1' - normal video transfer mode. '0' - video write transfers disabled (default value).
30:24	RW	vid	**MinPix** - Minimum Number of doublewords. This parameter defines a threshold value. When the number of doublewords inside the Video FIFO has reached this value a video-write burst is requested. Default value is 0x0F. Range 0x01 - 0x3C. The l.s. bit of this field (bit 24) is also used to configure the motherboard PCI bridge type, so the actual resolution of MinPix is limited by the $\overline{\text{Triton}}$ bit. When the $\overline{\text{Triton}}$ bit is '1', only odd values of MinPix are supported. When the $\overline{\text{Triton}}$ bit is '0', only even values are supported.
24	RW	vid	**$\overline{\text{Triton}}$** - PCI Bridge Controller type. This parameter configures the PCI $\overline{\text{REQ}}$ behavior to match the motherboard PCI bridge characteristics. '0' - Intel 'Triton' Bridge Controller. The $\overline{\text{REQ}}$ assertion and de-assertion conditions are modified accordingly. '1' - Other PCI Bridge Controllers. Default value is '1'. This bit is also used as the l.s. bit of MinPix.

Address Offset: 0x018 (Continued)

Bit	Type	Mod	Description
23:22	R		Reserved. Returns zero.
21:12	RW	vid	**VidWinHt** - Video Window Height. This register defines the number of lines that should be displayed by the ZR36057. If DispMod = 0, VidWinHt is half the number, if DispMod = 1, it is the entire number of display lines. Default value is 0x0F0.
11:10	R		Reserved. Returns zero.
9:0	RW	vid	**VidWinWid** - Video Window Width. This register defines the width of the video window in number of pixels. Default value is 0x3FF.

12.8 Masking Map "Top" Register

This register contains the DWORD base address of the top masking map.

Address Offset: 0x01C

Bit	Type	Mod	Description
31:2 1:0	RW R	vid	**MaskTopBase** - Masking Map Top Base Address. This is the source starting address of the top field for the masking map read transfers. Default value is 0xFFFFFFFC. Bits 1..0 are hardwired to 00b.

12.9 Masking Map "Bottom" Register

This register contains the DWORD base address of the bottom masking map.

Address Offset: 0x020

Bit	Type	Mod	Description
31:2 1:0	RW R	vid	**MaskBotBase** - Masking Map Bottom Base Address. This is the source starting address of the bottom field for the masking map read transfers. Default value is 0xFFFFFFFC. Bits 1..0 are hardwired to 00b.

12.10 Overlay Control Register

This register contains the parameters controlling overlay (byte 1) and masking map addressing (byte 0).

Address Offset: 0x024

Bit	Type	Mod	Description
31:16	R		Reserved. Returns zero.
15	RW	vid	**OvlEnable** - Overlay Enable flag. When enabled the masking information in the video mask is evaluated to allow overlay of other windows or graphics. When disabled the video window is always on top. '1' - overlay enabled, '0' - overlay disabled (default value).
14:8	R		Reserved. Returns zero.
7:0	RW	vid	**MaskStride**. This register defines the address increment in DWORDs that is needed to get from the end of a mask line to the beginning of the next. If the address difference between two consecutive mask lines in main memory is zero (i.e, they are physically consecutive) then MaskStride should be set (by the driver software) to zero (if DispMode=1) or the mask line size in DWORDs (if DispMode=0). Default value is 0xFF.

12.11 System, PCI and General Purpose Pins Control Register

This register contains the software reset bit (byte 3), a PCI control parameter (byte 2) and the General Purpose Pins direction parameter (byte 0).

Address Offset: 0x028

Bit	Type	Mod	Description
31:25	R		Reserved. Returns zero.
24	RW	all	**SoftReset** - Software Reset. This bit is asserted by the host to reset the ZR36057. If this bit is set to '0' all resettable registers in the device will be reset to their default value, except: - the SoftReset bit, - the PCI interface. The device continues to respond according to the PCI Specification and can be the target of a PCI transfer targeted at the ASRs or config. space. The transfer (single- or burst-write) asserting this bit will not be terminated abnormally. The device will not initiate any PCI transfers during reset, because all DMA channels are disabled. This register continues to be programmable. During reset, actually, only this bit of the ASRs can be modified by the host (e.g. turning reset off). A power on (hardware) reset also asserts this bit. After the hardware reset is over, a large portion of the ZR36057 (see above) will remain in reset mode until the SoftReset bit is deasserted. After this bit is deasserted, all registers retain their default values, all DMA channels remain disabled and all ASRs are programmable in accordance with their modification condition. '1' - no reset, '0' - reset (default value after power on).
23:19	R		Reserved. Returns zero.
18:16	RW	all	**WaitState** - PCI Wait State Control. This parameter defines the number of wait states inserted by the PCI slave logic. During each PCI transfer cycle to the ZR36057, the device will de-assert $\overline{\text{TRDY}}$ according to this value. Default value 000b.
15:8	R		Reserved. Returns zero.
7:0	RW	res	GenPurDir - General Purpose Pins Direction. These eight bits define the direction of the GPIO7..0 pins, respectively. A '1' defines the corresponding pin as an input, a '0' as an output. Default value is 0xFF (all inputs).

12.12 General Purpose Pins and GuestBus Control Register (I)

This register contains the values for General Purpose outputs (byte 3) and timing parameters for the first four Guests (bytes 1-0).

Address Offset: 0x02C

Bit	Type	Mod	Description
31:24	RW	all	**GenPurIO** - General Purpose Input/Output. The function of this register depends on the setting of GenPurDir. For each pin configured as input (default): - reading this bit will return the current value of the input pin. - writing to an input has no meaning, no change of that bit. For each pin configured as output: - reading this bit will return the value on the corresponding output pin. (If there is no external short circuit, this is the last value written by the host.) - writing will change the output to the value specified. Default value is 0xF0.
23:16	R		Reserved. Returns zero.
15:14	RW	res	duration time for guest 3: 00b - Tdur3 = 3 PCI clocks (default value), 01b - Tdur3 = 4 PCI clocks, 10b - Tdur3 = 12 PCI clocks, 11b - Tdur3 = 15 PCI clocks.
13:12	RW	res	recovery time for guest 3: 00b - Trec3 = 3 PCI clocks (default value), 01b - Trec3 = 4 PCI clocks, 10b - Trec3 = 12 PCI clocks, 11b - Trec3 = 15 PCI clocks.
11:8	RW	res	duration and recovery times for guest 2 (same structure as defined for guest 3 above).
7:4	RW	res	duration and recovery times for guest 1 (same structure as defined for guest 3 above).
3:0	RW	res	duration and recovery times for guest 0 (same structure as defined for guest 3 above).

12.13 MPEG Code Source Address Register

This register contains the DWORD base address for MPEG mode code DMA transfers.

Address Offset: 0x030

Bit	Type	Mod	Description
31:2 1:0	RW R	cod	**CodMemBase** - MPEG Code Memory Base Address. This is the source starting address for the code-read DMA transfers. Default value is 0xFFFFFFFC. Bits 1..0 are hardwired to 00b.

12.14 MPEG Code Transfer Control Register

This register contains status and control bits and configuration parameters for code DMA transfers.

Address Offset: 0x034

Bit	Type	Mod	Description
31	R		Reserved. Returns zero.
30	RC	all	**CodTime** - MPEG Code-Write Timeout flag. This bit is set to '1' by the GuestBus master if a code-write cycle on the GuestBus lasts more than 64 PCI clocks. This might happen when the accessed guest holds $\overline{GWS}$ low too long. It is cleared ('0') by the host writing a '1'. Priority is given to the GuestBus master in case of concurrent accesses to this bit. '1' - a code-write cycle has timed out. '0' - no timeout occurred (default value).
29	R		**CEmpty** - Code FIFO Empty. This bit reflects the status of the internal code buffer. When the buffer is empty this bit shows one, otherwise zero. '1' - Buffer is empty (default value), '0' - Buffer is not empty.
28	RW	cod	**CFlush** - Code FIFO Flush. This bit is used by the host to reset the internal code buffer. When it is asserted remaining code in the buffer is lost. After it is deasserted the buffer is empty and ready to receive data. '1' - flush internal code buffer (default value), '0' - normal operation of buffer.
27:23	R		Reserved. Returns zero.
22:20	RW	cod	**CodGuestID** - Code Guest Identification. These three bits select the guest to be accessed for MPEG code DMA transfers. Default after reset is 000b.
19	R		Reserved. Returns zero.
18:16	RW	cod	**CodGuestReg** - MPEG Code Guest Register. Register indication of accessed guest for MPEG code DMA transfers. Within each guest up to eight registers can be addressed. Default after reset is 000b.
15	R		Unused. Returns zero.
14:12	RW	cod	**CodMemSize** - MPEG Code Memory Buffer Size. This value determines the size of the contiguous memory buffer allocated by the host for storage of MPEG compressed data: 000b - 8 kbyte, 001b - 16 kbyte, 010b - 32 kbyte, 011b - 64 kbyte (default value), 100b - 128 kbyte, 101b - 256 kbyte. 110b - 111b will result in the default value.
11	R		Reserved. Returns zero.

Address Offset: 0x034 (Continued)

Bit	Type	Mod	Description
10:8	RW	cod	**CodMemStep** - MPEG Code Memory Report Step. In MPEG mode, this value determines the amount of code data, in quanta of 8 kbytes, after which the ZR36057 notifies its position (within the buffer) to the host, by requesting an interrupt. For proper operation, the buffer size must be greater than or equal to the report step size: 000b - IRQ every 8 kbytes, 001b - IRQ every 16 kbytes (default value), 010b - IRQ every 32 kbytes, 011b - IRQ every 64 kbytes, 100b - IRQ every 128 kbytes, 101b - IRQ every 256 kbytes. 110b - 111b will result in the default value.
7	RW	all	**CodReadEn** - MPEG Code-Read Enable. In MPEG mode, if this bit is cleared by the host, the code-read transfers are stopped. The current code-read pointer retains its value. When this bit is set to '1', the ZR36057 resumes code-read transfers. Default value is '0'.
6:4	R		Reserved. Returns zero.
3:1	RW	cod	**CodTrshld** - MPEG internal code FIFO threshold. In MPEG mode, if the fullness of the FIFO drops below this threshold value (in DWORDs) and CodReadEn is set to '1', a code-read burst is requested. Default value is 0x6.
0	RW	cod	**CodAutoEn** - MPEG Code-Read Auto Re-initialize Enable. In MPEG mode, if this bit is cleared, every time the code memory pointer reaches the end of the allocated space (i.e. CodMemBase plus CodMemSize) the code-read transfer is stopped. If the bit is set the code memory pointer is reinitialized at the end of the allocated space and code-read transfers run in a cyclic manner. Default value is '0'.

12.15 MPEG Code Memory Pointer Register

This register contains the pointer to the code memory of the host in MPEG mode.

Address Offset: 0x038

Bit	Type	Mod	Description
31:16	R		Unused. Returns zero.
15:0	RW	cod	**CodMemPoint** - MPEG Code Memory Pointer. In MPEG mode, this register reflects the current position of the code memory pointer within the range of the allocated host memory. The value represents the number of DWORDs from the base address. Writing to this register by the host can be used for reset or moving the pointer inside the code memory space, only if CodReadEn is deasserted. A value pointing outside the memory size should not be used. Default value is zero.

12.16 Interrupt Status Register

This register contains the status of the interrupt sources.

Address Offset: 0x03C

Bit	Type	Mod	Description
31	R		Reserved. Returns zero.
30	RC	all	**GIRQ1** - **GIRQ1** Input Pin. A '1' indicates that a guest requested an interrupt on the GIRQ1 input pin. This bit is cleared when the host tries to write '1' to it. In case of concurrent accesses to this bit, it remains '1'. Default value is '0' (no IRQ).
29	RC	all	**GIRQ0** - **GIRQ0** Input Pin. A '1' indicates that a guest requested an interrupt on the GIRQ0 input pin. This bit is cleared when the host tries to write '1' to it. In case of concurrent accesses to this bit, it remains '1'. Default value is '0' (no IRQ).
28	RC	all	**CodRepIRQ** - MPEG Code Report Step Interrupt Request. A '1' indicates that the code memory buffer pointer, in MPEG mode, has passed a report step. This bit is cleared when the host tries to write '1' to it. In case of concurrent accesses to this bit, it remains '1'. Default value is '0' (no IRQ).

Address Offset: 0x03C (Continued)

Bit	Type	Mod	Description
27	RC	all	**JPEGRepIRQ** - JPEG Report Interrupt Request. A '1' indicates that a JPEG field/frame process has ended. In Compression modes, a field compression has ended and its data has been transferred to the allocated code buffer in system memory. In Decompression modes, a field/frame decompression has ended and its corresponding code buffer can be re-loaded by the software. This bit is cleared when the host tries to write '1' to it. In case of concurrent accesses to this bit, it remains '1'. Default value is '0' (no IRQ).
26:0	R		Reserved. Returns zero.

12.17 Interrupt Control Register

This register contains the control byte for the interrupt handling.

Address Offset: 0x040

Bit	Type	Mod	Description
31	R		Reserved for future interrupt source. Returns '0'.
30	RW	all	**GIRQ1En** - **GIRQ1** Enable. When enabled and IntPinEn is set to '1', each positive edge of the GIRQ1 input will generate an interrupt request on the PCI Bus $\overline{\text{INTA}}$ output pin. When cleared, GIRQ1 continues to reflect the corresponding interrupt pin. '1' - GIRQ1 enabled, '0' - GIRQ1 disabled (default value).
29	RW	all	**GIRQ0En** - GIRQ0 Enable. When enabled and IntPinEn is set to '1', each positive edge of the GIRQ0 input will generate an interrupt request on the PCI Bus $\overline{\text{INTA}}$ output pin. When cleared, GIRQ0 continues to reflect the corresponding interrupt pin. '1' - GIRQ0 enabled, '0' - GIRQ0 disabled (default value).
28	RW	all	**CodRepIrqEn** - MPEG Mode Code Report Step Interrupt Enable. When enabled and IntPinEn is set to '1', an interrupt request will be generated on the PCI Bus $\overline{\text{INTA}}$ output pin each time the code memory buffer pointer passes a report step. When CodRepIrqEn is cleared, CodRepIrq continues to reflect the internal report step interrupt request. '1' - interrupt request enabled, '0' - disabled (default value).

Address Offset: 0x040 (Continued)

Bit	Type	Mod	Description
27	RW	all	JPEGRepIRQEn - JPEG Report Interrupt Request Enable. When enabled and IntPinEn is set to '1', an interrupt request will be generated on the PCI Bus $\overline{INTA}$ output pin after the end of each JPEG field/frame process. When JPEGRepIRQEn is cleared, JPEGRepIRQ continues to reflect the JPEG process status. '1' - interrupt request enabled, '0' - disabled (default value).
26:25	R		Reserved. Returns zero.
24	RW	all	***IntPinEn*** - $\overline{INTA}$ Pin Enable. When cleared, none of the events that may cause an interrupt request on the PCI Bus $\overline{INTA}$ pin is enabled. Nevertheless the interrupt status register continues to reflect all interrupt input pins and internal interrupt requests. '1' - every interrupt request is passed onto the PCI Bus, '0' - $\overline{INTA}$ disabled (default value).
23:0	R		Reserved. Returns zero.

12.18 I²C-Bus Register

This register contains the control bits of the I^2C Bus.

Address Offset: 0x044

Bit	Type	Mod	Description
31:2	R		Unused. Returns zero.
1	RW	all	**SDA** - I^2C SDA Line. When the host writes '0' to this bit, the SDA output signal goes low. When the host writes '1', SDA goes into tri-state. When the host reads this bit it reflects the current level on the SDA pin. Default value is '1'.
0	RW	all	**SCL** - I^2C SCL Line. When the host writes '0' to this bit, the SCL output signal goes low. When the host writes '1', SCL goes into tri-state. When the host reads this bit it reflects the current level on the SCL pin. Default value is '1'.

12.19 PostOffice Register

This register contains the status (byte 3), control (byte 2) and data (byte 0) parameters for PostOffice transfers.

Address Offset: 0x200 - 0x2FF

Bit	Type	Mod	Description
31:26	R		Reserved. Returns zero.
25	R	all	**POPen** - PostOffice Request Pending flag. This bit is set internally to '1' when the host writes to the PostOffice data byte. It is cleared when a PostOffice cycle is completed or a PostOffice time-out occurred. In case of concurrent accesses to this bit, the result is '0'. '1' - PostOffice request is pending. '0' - PostOffice request is not pending (default value).
24	RC	all	**POTime** - PostOffice Time-out flag. This bit is set to '1' by the GuestBus master if a PostOffice cycle on the GuestBus lasts more than 64 PCI clocks. This might happen when the accessed guest holds $\overline{GWS}$ low for too long. It is cleared ('0') by the host writing a '1'. In case of concurrent accesses to this bit, it remains '1'. '1' - PostOffice cycle has timed out. '0' - no time-out occurred (default value).
23	RW	all	**PODir** - PostOffice Direction flag. This bit defines the direction of the PostOffice operation: '0' - Read (host reads guest). '1' - Write (host writes to guest). Default after reset is '1'.
22: 20	RW	all	**POGuestID** - PostOffice Guest Identification. These three bits select the guest to be accessed. They determine which of the $\overline{GCS}$ pins will be asserted in the requested PostOffice cycle. Up to eight guests can be identified. Default after reset is 000b.
19	R		Reserved. Returns zero.
18:16	RW	all	**POGuestReg** - PostOffice Guest Register. Register indication of accessed guest. Within each guest up to eight registers can be addressed. The POGuestReg bits determine the register address that will be presented on the GADR2:0 lines in the requested PostOffice cycle. Default after reset is 000b.
15:8	R		Reserved. Returns zero.
7:0	R, W	all	**POData** - PostOffice Data. An eight-bit register containing the data being transferred during PostOffice reads and writes. Default after reset is 00000000b.

12.20 JPEG Mode and Control

This register contains the JPEG Mode configuration and optional control bits.

Address Offset: 0x100

Bit	Type	Mod	Description
31	RW	all	**JPG**- JPEG/MPEG mode selection. This bit selects between the two code DMA controller modes. '1' - JPEG Mode. '0' - MPEG Mode. Default value is '0'
30 : 29	RW	jpg	**JPGMode** - JPEG Sub-Modes Selection. These two bits configure the JPEG sub-mode. 11b - Motion Video Compression. 10b - Motion Video Decompression. 01b - Still Image Compression. 00b - Still Image Decompression. Default value is 11b
28 : 7	R		Reserved. Returns zero.
6	RW	jpg	**RTBSY_FB** - RTBSY Feed-Back. This control bit enables the ZR36057 to de-assert dynamically (during a process) the $\overline{\text{PXEN}}$ signal if $\overline{\text{RTBSY}}$ is detected in the active area of the field. '1' - Enable $\overline{\text{PXEN}}$ de-assertion if $\overline{\text{RTBSY}}$ is detected. '0' - Disable $\overline{\text{PXEN}}$ de-assertion. Default value is '0'
5	RW		Go_en - The enable bit of the ZR36050 GO command cycle. The bit, when '1', enables the GuestBus Master to perform a JPEG GO cycle. During the JPEG GO cycle the ZR36057 assumes that the correct address (0x00h) is pre-latched in an external address register. It is the host's responsibility to perform the write operation to load the address. The host must de-assert Go_en whenever it accesses the ZR36050 (using PostOffice) in a middle of a JPEG process, or changes the address latched in the external register. The host must also de-assert Go_en before PostOffice pseudo write-through burst cycles. In this case, the host also has to wait at least 0.5 microseconds before initiating the burst (to allow the JPEG GO cycle to complete, if one was started). Default value is '0' (Go not enabled).
4	RW	jpg	**SyncMstr** - Sync Signals Master. This bit configures the ZR36057 as a sync master. This configuration is allowed in all JPEG modes except Motion Video Compression '1' - The ZR36057 is the sync signal master. '0' - The sync signals are driven from an external video source. Default value is '0'

Address Offset: 0x100 (Continued)

Bit	Type	Mod	Description
3	RW	jpg	**Fld_per_buff** - Number of Fields Per Code Buffer. This bit reflects the system memory code buffer structure in JPEG Compression and Decompression modes. '1' - The code buffer contains one code field. '0' - The code buffer contains two consecutive code fields, one code frame. Default value is '0'
2	RW	jpg	**VFIFO_FB** - VFIFO Feed-Back. This control bit enables the ZR36057 to de-assert dynamically (during a process) the $\overline{\text{PXEN}}$ signal according to the status of the Video FIFO. '1' - Enable $\overline{\text{PXEN}}$ de-assertion if the pixel buffer is close to overflow. '0' - Disable $\overline{\text{PXEN}}$ de-assertion if the pixel buffer is close to overflow. Default value is '0'
1	RW	jpg	**CFIFO_FB** - CFIFO Feed-Back. This control bit enables the ZR36057 to de-assert dynamically (during a process) the $\overline{\text{PXEN}}$ signal according to the status of the Code FIFO. '1' - Enable $\overline{\text{PXEN}}$ de-assertion if the code buffer is close to overflow/underflow. '0' - Disable $\overline{\text{PXEN}}$ de-assertion if the code buffer is close to overflow/underflow. Default value is '0'
0	RW	jpg	**Still_LitEndian** - Still image pixel Little Endian format selector. This control bit defines the pixel format in Still Image Compression and Decompression. '1' - The pixel format is Little Endian. '0' - The pixel format is Gib Endian. Default value is '1'.

12.21 JPEG Process Control

This register contains the JPEG process control.

Address Offset: 0x104

Bit	Type	Mod	Description
31: 8	R		Reserved. Returns zero.
7	RW	all	**P_reset** - Process Reset. This bit is asserted by the host in order to reset the ZR36057 JPEG-related state machines. The bit must be asserted at the beginning of a JPEG process. While it is asserted, all of the JPEG process parameters may be configured by the host. '1' - No reset. '0' - Reset. Default value is '1'.
6	R		Reserved, Returns zero.
5	RW		**CodTrnsEn** - JPEG Code Transfer Enable. This bit enables the code transfer between the internal code buffer and the system memory in all of the JPEG modes. '1' - Code transfer is enabled. '0' - code transfer is disabled. Default value is '0'.

Address Offset: 0x104

Bit	Type	Mod	Description
4 : 1	R		Reserved, Returns zero.
0	RW	all	Active - This command bit is asserted by the host in order to initiate a JPEG process. '1' - Active is asserted. '0' - Active is deasserted. Default value is '0'.

12.22 Vertical Sync Parameters

This register contains the VSYNC parameters to be generated by the **ZR36057** as a sync master.

Address Offset: 0x108

Bit	Type	Mod	Description
31: 24	RW		Reserved. Returns zero.
23: 16	RW	jpg	**VsyncSize** - VSYNC signal length. The VSYNC length is measured in lines. Default value is 0x06.
15 : 0	RW	jpg	**FrmTot** - Frame total size. The total number of lines per frame. This parameter must be an odd number. Default Value is 0x020D (525, for NTSC)

12.23 Horizontal Sync Parameters

This register contains the HSYNC parameters to be generated by the **ZR36057** as a sync master.

Address Offset: 0x10C

Bit	Type	Mod	Description
31 : 16	RW	jpg	**HsyncStart** - HSYNC signal Start point. The point in the line (measured in number of VCLKs) at which HSYNC should be asserted. Default value is 0x0280 (640, for square pixel NTSC).
15 : 0	RW	jpg	**LineTot** - Line total size. The total number VCLKs in a line. Default Value is 0x030C (780, for square pixel NTSC)

12.24 Field Horizontal Active Portion

This register contains the horizontal parameters of the active portion of the processed field.

Address Offset: 0x110

Bit	Type	Mod	Description
31 : 16	RW	jpg	**NAX** - The first pixel in a line to be processed. Counted from the active edge of HSYNC. Default value is 0x0000.
15 : 0	RW	jpg	**PAX** - The number of pixels to be processed in a line. Default Value is 0x0280 (640, for square pixel NTSC)

12.25 Field Vertical Active Portion

This register contains the vertical parameters of the active portion of the processed field.

Address Offset: 0x114

Bit	Type	Mod	Description
31 : 16	RW	jpg	**NA**Y - The first line in a field to be processed. Counted from the active edge of VSYNC. Default value is 0x000A.
15 : 0	RW	jpg	**PAY** - The number of lines to be processed in a field. Default Value is 0x00F0 (240, for NTSC)

12.26 Field Process Parameters

This register contains the general parameters of the field process.

Address Offset: 0x118

Bit	Type	Mod	Description
31 : 1	R		Reserved. Returns zero.
0	RW	jpg	**Odd_Even** - First field type. The type of the first field to be processed. Odd type is defined as the field in which the active edge of VSYNC is asserted during the active portion of the horizontal line. Even type is defined as the field in which the active edge of VSYNC is asserted during the active portion of HSYNC. '1' - The first field is Odd. '0' - The first field is Even. Default value is '1'.

12.27 JPEG Code Base Address

This register specifies the base address of the code buffer table.

Address Offset: 0x11C

Bit	Type	Mod	Description
31 : 0	RW	jpg	**I_STAT_COM_PTR** - The memory address of the code buffer table. Default value 0xFFFFFFFC.

12.28 JPEG Code FIFO Threshold

This register specifies Code FIFO threshold in JPEG mode.

Address Offset: 0x120

Bit	Type	Mod	Description
31 : 8	R		Reserved, Returns zero.
7 : 0	RW	jpg	**JPEGCodTrshld** - JPEG code FIFO Threshold. In Compression, if the fullness level of the Code FIFO (in doublewords) goes above this threshold the PCI bus is requested. In Decompression, if the fullness level of the Code FIFO (in doublewords) goes under this threshold the PCI bus is requested. Default value is 0x50h

12.29 JPEG Codec Guest ID

This register contains the ZR36050 Guest ID for the JPEG GO command cycle.

Address Offset: 0x124

Bit	Type	Mod	Description
31 :7	R		Reserved, Returns zero.
6 : 4	RW	jpg	**JPEGuestID** - JPEG Codec Guest ID. These three bits define the guest port to which the ZR36050 is connected, in order to perform the JPEG GO command cycle. Default value is 100b
3	R		Reserved, Returns zero.
2 : 0	RW	jpg	**JPEGuestReg** - JPEG Codec Register. These three bits define the ZR36050 internal register address for the JPEG GO command. Default value is 000b

12.30 *GuestBus* Control Register (II)

This register contains the timing parameters of an additional 4 guests (the first 4 guests' parameters are defined in 12.12 "General Purpose Pins and GuestBus Control Register (I)").

Address Offset: 0x12C

Bit	Type	Mod	Description
31 : 16	R		Reserved, Returns zero.
15 : 14	RW	res	Duration time for guest 7: 00b - $Tdur_7$ = 3 PCI clocks (default value), 01b - Tdur7 = 4 PCI clocks, 10b - Tdur7 = 12 PCI clocks, 11b - Tdur7 = 15 PCI clocks.
13 :12	RW	res	Recovery time for guest 7: 00b - Trec7 = 3 PCI clocks (default value), 01b - Trec7 = 4 PCI clocks, 10b - Trec7 = 12 PCI clocks, 11b - Trec7 = 15 PCI clocks.
11 : 8	RW	res	Duration and recovery time of guest 6 (same structure as defined for guest 7 above)
7 : 4	RW	res	Duration and recovery time of guest 5 (same structure as defined for guest 7 above)
3 : 0	RW	res	Duration and recovery time of guest 4 (same structure as defined for guest 7 above)

12.31 "Still Transfer" Register

This register is used for data and control in Still Image Compression or Decompression.

Address Offset: 0x140

Bit	Type	Mod	Description
31/7	R	all	**Still_Bsy** - Still Transfer Busy indication bit. The bit indicates to the host whether the "Still Transfer" register is available to write the next pixel in Still Image Compression mode. When this bit is '0', the register is available. The bit location depends on the Still_LitEndian configuration bit in the JPEG Mode and Control register. If Little Endian format was selected then the bit location is 31. If Gib Endian format was selected then the bit location is 7. When Still Image Compression mode is selected (and after P_reset was de-asserted), the bit is '0'. After the host writes a pixel, the bit is set to '1'. The bit remains '1' until the pixel is synchronized to the video clock and going to be driven out from the ZR36057 video port. Then when the register is available again, the bit is reset to '0'. When Still Image Decompression mode is selected (and after P_reset was de-asserted), the bit is '1'. After the ZR36057 fetches a new pixel, and the pixel is ready for the host to read it, the bit is reset to '0'. It is set to '1' again after the pixel was read. '1' - The register is not available. '0' - The register is available. Default value is '0'.
31 : 0	RW	all	Still Transfer Pixel register. The byte order (RGB or BGR) is defined by the Still_LitEndian configuration bit in the JPEG Mode and Control register. If Little Endian format was selected, the byte order is: R7..0 on bits 23..16. G7..0 on bits 15..8. B7..0 on bits7..0. If Gib Endian format was selected, the byte order is: R7..0 on bits 15..8. G7..0 on bits 23..16. B7..0 on bits 31..24.

13.0 ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

Storage Temperature .....65°C to +150°C
Supply Voltage (V_{DD}) .....-0.5 V to +6.0 V
DC Output Voltage ..... 0.5 V to V_{DD} + 0.5
DC Input Voltage ..... -0.5 V to V_{DD} + 0.5
DC Output Current .....-100 mA to +100 mA
DC Input Current .....-100 mA to +100 mA
Max Power Dissipation ..... 1.6W @ 33 MHz

NOTE: Stresses above these values may cause permanent device failure. Functionality at or above those limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGE

Temperature.....0°C to +70°C
Supply Voltage ..... 4.75 V to 5.25 V

DC CHARACTERISTICS

Table 13: Digital Inputs

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input low voltage		0.8	V	
V_{IH}	Input high voltage	2.0		V	
V_{IIL}	Input low voltage (SDA, SCL)		0.3 x V_{DD}	V	
V_{IIH}	Input high voltage (SDA, SCL)	0.7 x V_{DD}		V	
I_{LI}	input leakage current		± 10 [1]	µA	
C_{IN}	input capacitance		8	pF	

1. Input Pullup Leakage Current for SDA is -200 µA.

Table 14: Digital Outputs

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{OL}	Output low voltage		0.4	V	
V_{OH}	Output high voltage	3.8		V	
I_{LO}	Output leakage current		±10	µA	
C_{OUT}	Output capacitance		8	pF	

Table 15: PCI Bus

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input low voltage		0.8	V	
V_{IH}	Input high voltage	2.0		V	
V_{OL}	Output low voltage		0.55	V	
V_{OH}	Output high voltage	2.4		V	
I_{LIH}	Input leakage current		±70	µA	V_{IN} = 2.7/0.5 V
C_{PCICLK}	PCICLK input capacitance	5	12	pF	

14.0 AC TIMING SPECIFICATIONS

14.1 PCI Bus Timing

Table 16: PCI Bus Timing

Symbol	Parameter	Min	Max	Unit	
t_{PCI}	PCICLK period	30		ns	
t_{PH}	PCICLK high	12		ns	
t_{PL}	PCICLK low	12		ns	
t_{A3}	PCICLK slew rate	1	4	V/ns	
t_{A4}	PCICLK to signal valid delay - bussed signals	2	11	ns	
t_{A5}	PCICLK to signal valid delay - point-to-point	2	12	ns	
t_{A6}	Float to active delay	2	11	ns	
t_{A7}	Active to float delay		28	ns	
t_{A8}	Input setup time to PCICLK - bussed signals	7		ns	
t_{A9}	Input setup time to PCICLK - point-to-point	10		ns	
t_{A10}	Input hold time from PCICLK	0		ns	
t_{A11}	Unoaded output rise/fall time	1	5	V/ns	Between 0.4 V and 2.4 V

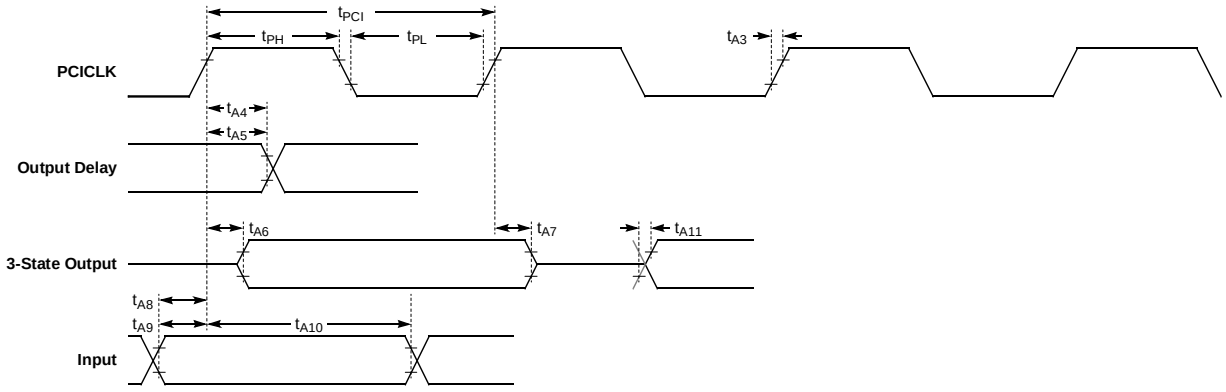


Figure 10. Video Bus Timing

14.2 Video Bus Timing

Table 17: Video Bus Timing

Symbol	Parameter	Min	Max	Unit	
t_{V2P}	VCLKx2 period	31	45	ns	40% to 60% duty cycle
t_{V2T}	VCLKx2 rise/fall transition		5	ns	
t_{VT}	VCLK rise/fall transition		5	ns	
t_{VIS}	Video bus input setup	11		ns	
t_{VIH}	Video bus input hold	0		ns	
t_{VCLKH}	VCLK hold	-2		ns	

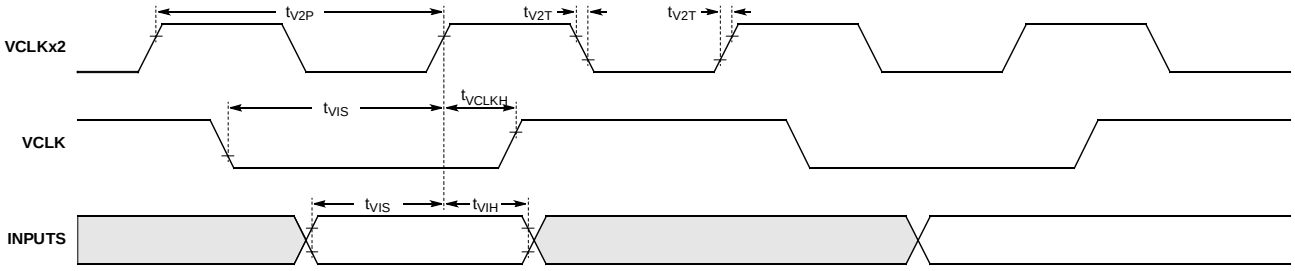


Figure 11. Video Bus Timing

Table 18: Video Bus Outputs

Symbol	Parameter	Min	Max	Unit	
t_{VBO}	Video bus output delay	1	16	ns	Typical output Load 50pf
t_{VBH}	Video bus output hold (all signals except $\overline{PXEN}$)	1		ns	
t_{VPO}	$\overline{PXEN}$ output delay		17	ns	
t_{VPH}	$\overline{PXEN}$ output hold	2		ns	

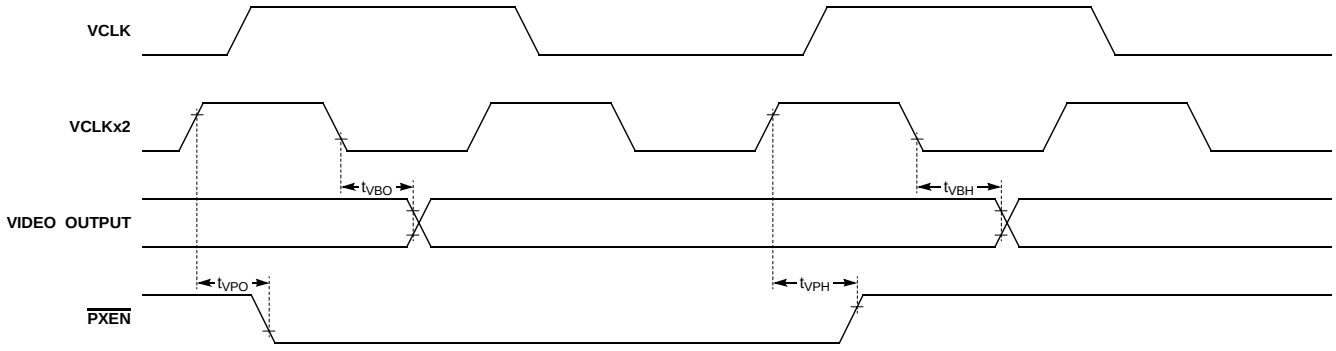


Figure 12. Video Bus Timing – Control and Data Output Signals

14.3 GuestBus Timing

Table 19: GuestBus Interface Timing

Symbol	Parameter	Min	Max	Unit	
t_{CARW}	$\overline{GCS}$n / GADR2..0 setup	16		ns	
t_{RWCA}	$\overline{GCS}$n / GADR2..0 hold	16		ns	
t_{RWDUR}	$\overline{GWR}$ / $\overline{GRD}$ low	90		ns	Determined by parameter T_{dur} and signal $\overline{GWS}$
t_{RWREC}	$\overline{GWR}$ / GRD high	90		ns	Determined by parameter T_{rec}
t_{PD}	Propagation delay. PCICLK to output signal	1	16	ns	Typical output load 70 pF
t_{GDH}	Write data hold. $\overline{GWR}$ / GRD to data float		16	ns	Typical output load 70 pF
t_{GRS}	Read data setup	11		ns	
t_{GRH}	Read data hold	0		ns	
t_{GIRQ}	GIRQ1..0 high / low	$2 * t_{PCI}$		ns	
t_{GWS}	Wait state setup. $\overline{GWS}$ to $\overline{GWR}$/$\overline{GRD}$	$(t_{DUR} -1) * t_{PCP} - t_{PD} - t_{GRS}$		ns	
t_{GWL}	Wait state low [1]	t_{PCI}		ns	
t_{GWH}	Wait state hold. $\overline{GWS}$ to $\overline{GWR}$/$\overline{GRD}$	$t_{PCP} + t_{PD} + t_{GRS}$		ns	

1. For each period of t_{PCI} the duration of the transfer will be extended for one PCI clock period.

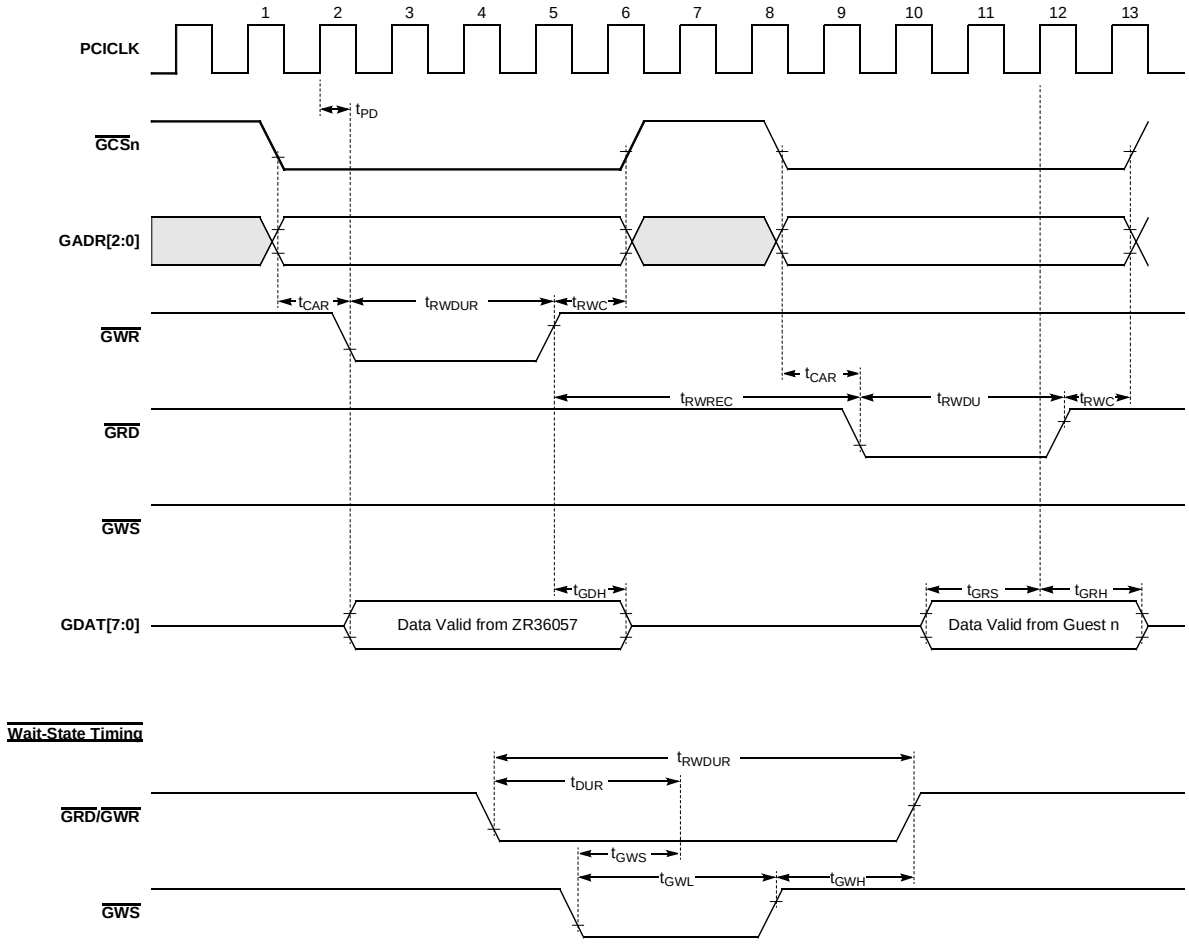


Figure 13. GuestBus Timing

14.3.1 Codec Bus Interface Timing

Table 20:

Symbol	Parameter	Min	Max	Unit	
t_{CCS}	$\overline{CCS}$ setup	10		ns	
t_{CCH}	$\overline{CCS}$ hold	0		ns	
t_{CPD}	Code output propagation delay	3	9	ns	Typical load 15pF
t_{CHD}	Code output hold delay	1		ns	
t_{CIS}	Code input setup	9		ns	
t_{CIH}	Code input hold	0		ns	
t_{CBO}	$\overline{CBUSY}$ output propagation delay	2	16	ns	Typical load 15pF
t_{CBH}	$\overline{CBUSY}$ output hold delay	2		ns	
t_{CES}	$\overline{CEND}$ setup	9		ns	
t_{CEH}	$\overline{CEND}$ hold	1		ns	

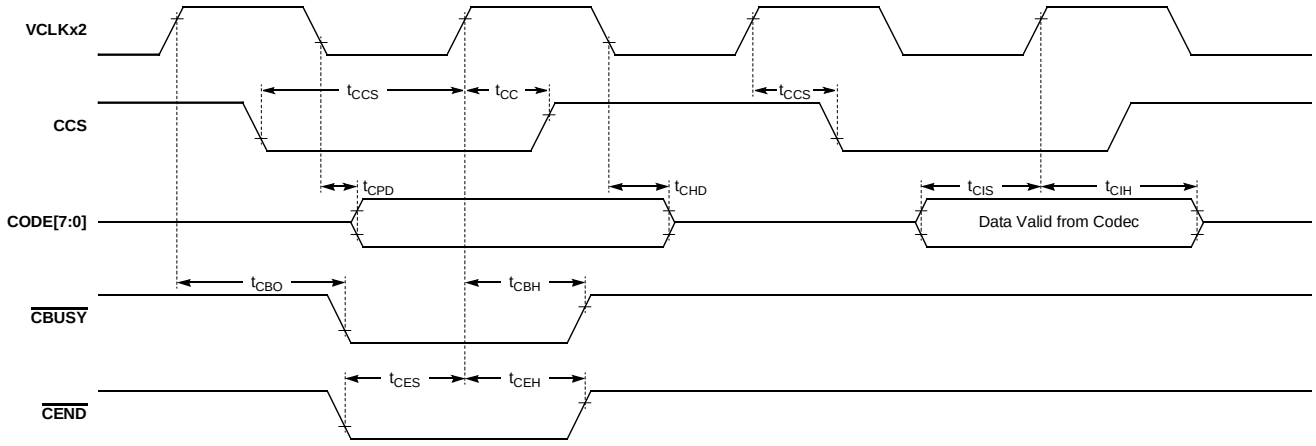
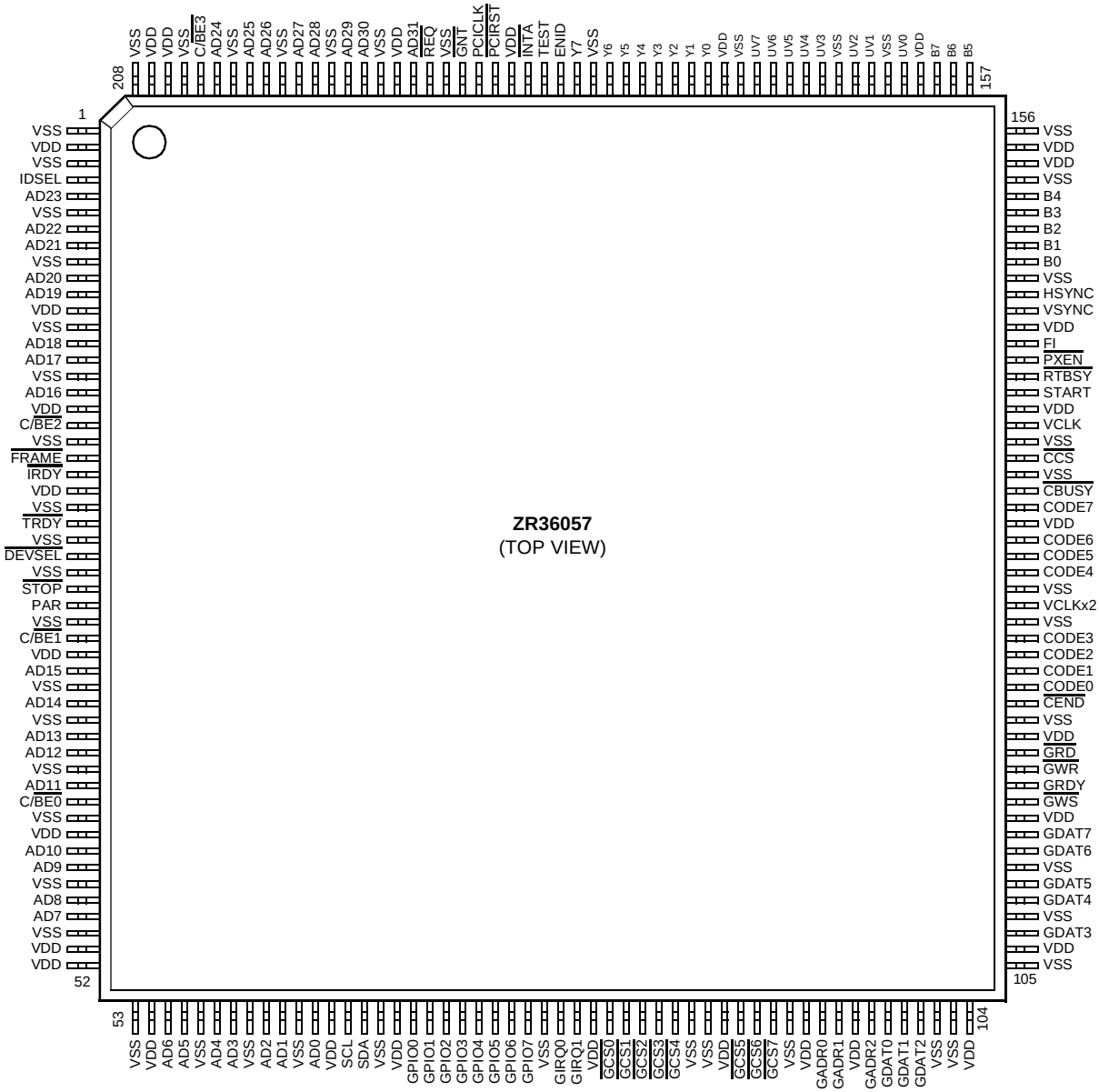


Figure 14. Codec Interface Timing

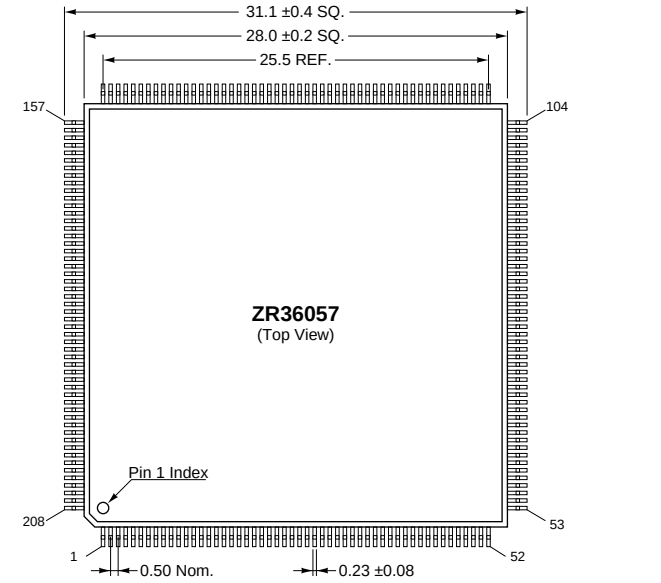
15.0 MECHANICAL DATA

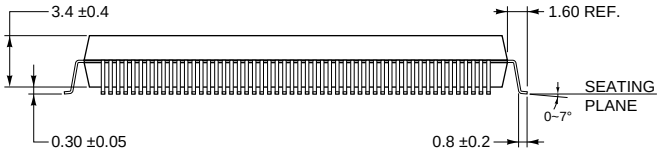
15.1 Pinout

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	VSS	27	DEVSEL	53	VSS	79	GIRQ0	105	VSS	131	CODE6	157	B5	183	TEST
2	VDD	28	VSS	54	VDD	80	GIRQ1	106	VDD	132	VDD	158	B6	184	INTA
3	VSS	29	STOP	55	AD6	81	VDD	107	GDAT3	133	CODE7	159	B7	185	VDD
4	IDSEL	30	PAR	56	AD5	82	GCS0	108	VSS	134	CBUSY	160	VDD	186	PCIRST
5	AD23	31	VSS	57	VSS	83	GCS1	109	GDAT4	135	VSS	161	UV0	187	PCICLK
6	VSS	32	C/BE1	58	AD4	84	GCS2	110	GDAT5	136	CCS	162	VSS	188	GNT
7	AD22	33	VDD	59	AD3	85	GCS3	111	VSS	137	VSS	163	UV1	189	VSS
8	AD21	34	AD15	60	VSS	86	GCS4	112	GDAT6	138	VCLK	164	UV2	190	REQ
9	VSS	35	VSS	61	AD2	87	VSS	113	GDAT7	139	VDD	165	VSS	191	AD31
10	AD20	36	AD14	62	AD1	88	VSS	114	VDD	140	START	166	UV3	192	VDD
11	AD19	37	VSS	63	VSS	89	VDD	115	GWS	141	RTBSY	167	UV4	193	VSS
12	VDD	38	AD13	64	AD0	90	GCS5	116	GRDY	142	PXEN	168	UV5	194	AD30
13	VSS	39	AD12	65	VDD	91	GCS6	117	GWR	143	FI	169	UV6	195	AD29
14	AD18	40	VSS	66	SCL	92	GCS7	118	GRD	144	VDD	170	UV7	196	VSS
15	AD17	41	AD11	67	SDA	93	VSS	119	VDD	145	VSYNC	171	VSS	197	AD28
16	VSS	42	C/BE0	68	VSS	94	VDD	120	VSS	146	HSYNC	172	VDD	198	AD27
17	AD16	43	VSS	69	VDD	95	GADR0	121	CEND	147	VSS	173	Y0	199	VSS
18	VDD	44	VDD	70	GPIO0	96	GADR1	122	CODE0	148	B0	174	Y1	200	AD26
19	C/BE2	45	AD10	71	GPIO1	97	VDD	123	CODE1	149	B1	175	Y2	201	AD25
20	VSS	46	AD9	72	GPIO2	98	GADR2	124	CODE2	150	B2	176	Y3	202	VSS
21	FRAME	47	VSS	73	GPIO3	99	GDAT0	125	CODE3	151	B3	177	Y4	203	AD24
22	IRDY	48	AD8	74	GPIO4	100	GDAT1	126	VSS	152	B4	178	Y5	204	C/BE3
23	VDD	49	AD7	75	GPIO5	101	GDAT2	127	VCLKx2	153	VSS	179	Y6	205	VSS
24	VSS	50	VSS	76	GPIO6	102	VSS	128	VSS	154	VDD	180	VSS	206	VDD
25	TRDY	51	VDD	77	GPIO7	103	VSS	129	CODE4	155	VDD	181	Y7	207	VDD
26	VSS	52	VDD	78	VSS	104	VDD	130	CODE5	156	VSS	182	ENID	208	VSS


ZR36057
(TOP VIEW)
1
52
53
104
105
156
157
208
VSS VDD VDD VSS C/BE3 AD24 VSS AD25 AD26 VSS AD27 AD28 VSS AD29 AD30 VSS VDD AD31 REQ VSS GNT PCICLK PCIRST VDD INTA TEST ENID Y7 VSS Y6 Y5 Y4 Y3 Y2 Y1 Y0 VDD VSS UV7 UV6 UV5 UV4 UV3 VSS UV2 UV1 VSS UV0 VDD B7 B6 B5
VSS VDD VSS IDSEL AD23 VSS AD22 AD21 VSS AD20 AD19 VDD VSS AD18 AD17 VSS AD16 VDD C/BE2 VSS FRAME IRDY VDD VSS TRDY VSS DEVSEL VSS STOP PAR VSS C/BE1 VDD AD15 VSS AD14 VSS AD13 AD12 VSS AD11 C/BE0 VSS VDD AD10 AD9 VSS AD8 AD7 VSS VDD VDD
VSS VDD VDD VSS B4 B3 B2 B1 B0 VSS HSYNC VSYNC VDD FI PXEN RTBSY START VDD VCLK VSS CCS VSS CBUSY CODE7 VDD CODE6 CODE5 CODE4 VSS VCLKx2 VSS CODE3 CODE2 CODE1 CODE0 CEND VSS VDD GRD GWR GRDY GWS VDD GDAT7 GDAT6 VSS GDAT5 GDAT4 VSS GDAT3 VDD VSS
VSS VDD AD6 AD5 VSS AD4 AD3 VSS AD2 AD1 VSS AD0 VDD SCL SDA VSS VDD GPIO0 GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO7 VSS GIRQ0 GIRQ1 VDD GCS0 GCS1 GCS2 GCS3 GCS4 VSS VSS VDD GCS5 GCS6 GCS7 VSS VDD GADR0 GADR1 VDD GADR2 GDAT0 GDAT1 GDAT2 VSS VSS VDD

15.2 Dimensions





Dimensions in millimeters

Appendix A: *ZR36100* - *ZR36057* Interface

Figure 15 describes a recommended interconnection between the ZR36057 and the *ZR36100*/*ZR36110* host bus.

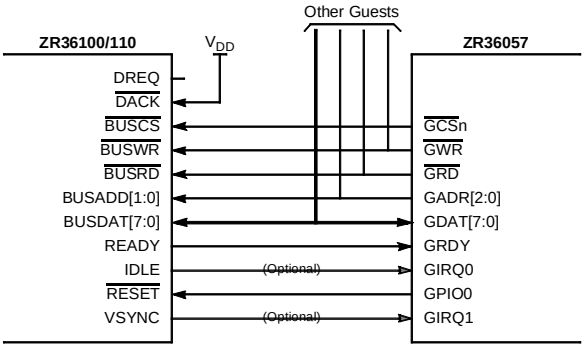


Figure 15. ZR36057 - ZR36100/110 Basic Interconnection

ZR36100 Reset

Any of the software controlled GPI/O pins (configured as output) of the ZR36057 may be used as a RESET input of the decoder. The software then directly manipulates the RESET signal through the corresponding register bit. Since the default configuration of the GPIO pins after reset is input, a pull down resistor should be applied to the *ZR36100* RESET input.

Mapping the *ZR36100* on the ZR36057's GuestBus

The driver software must map the *ZR36100* on the GuestBus: the *ZR36100*'s guest ID number (0,1,2, or 3) must be configured as a code-write target. The proper timing parameters (tdur = 3, to ensure 82ns, trec = 4, to ensure 100ns) of the *ZR36100* must be loaded to the GuestBus control register (address 0x030). The *ZR36100* occupies only four 8-bit registers out of the eight registers dedicated to each guest.

ZR36100 Initialization

The initialization consists of loading the *ZR36100* microcodes and parameters. This is done using the PostOffice mechanism. The host interface of the *ZR36100* must be set to 8 bit, Intel format, I/O only. The BSLN parameter should be set to 2 or 4, for efficient operation.

On-Line Commands and Status

On-line command writes and status reads are also done using the PostOffice mechanism.

Bitstream Transfer

Some preparations must be done prior to triggering the *ZR36100* with a go command.
The host must allocate a contiguous code buffer in the system memory. Bitstream retrieved from the MPEG source is stored in this buffer. The ZR36057 reads data from this buffer in a DMA fashion and transfers it, through the CFIFO, to the MPEG decoder. There are several possible sizes of the memory buffer. The host must inform the ZR36057 of the buffer address, size and "report step". After the code buffer in memory is allocated, reported to the ZR36057, and filled up for the first time, a *ZR36100* go command can be issued. Immediately after this, the DMA code-read cycles must be enabled by setting the DMA Code-Read Enable bit to '1'. The ZR36057 then starts fetching data from the main memory buffer using cyclic addressing. Whenever it passes a "report step" it initializes an interrupt request. Within the interrupt service routine the host should check the current position of the ZR36057 Code Memory Buffer Pointer, and decide weather it should refresh an old portion of the buffer with new data from the MPEG source. Once the coded data arrives at the CFIFO, the GBM unit starts writing it over the GuestBus to the *ZR36100*.

Appendix B: MD207/MD208 - *ZR36057* Interface

This appendix suggests the basic interconnection between the ZR36057 and the MD207/208 video encoder. Naturally, when these two devices are connected together there must be a third device, mastering the YUV bus. Figure describes a basic interconnection between the ZR36057 and the MD207/208, with an arbitrary YUV source. This minimum example does not use the graphics overlay capability of the MD207/208

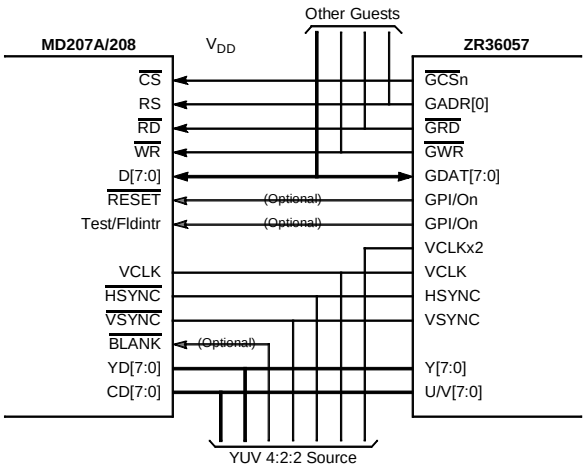


Figure 16. ZR36057 - MD207/208 Basic Interconnection

MD207/208 Reset

Software controlled usage of the $\overline{\text{RESET}}$ input of the MD207/208 is optional. Generally, it is more efficient to connect it to the power-up reset of the circuit, and control the device through the software reset register bit of the MD207/208. If a hardware reset is needed in the design, any of the software controlled GPI/O pins (configured as output) of the ZR36057 may be used as a $\overline{\text{RESET}}$ signal. The software then directly manipulates the $\overline{\text{RESET}}$ signal through the corresponding register bit of the ZR36057. Since the default configuration of the GPI/O pins after reset is input, a pull down resistor should be applied to the MD207/208 $\overline{\text{RESET}}$ input.

Mapping the MD207/208 on the *ZR36057*'s GuestBus

The software driver must map the MD207/208 on the GuestBus. The proper timing parameters (t_{dur} = 12, t_{rec} = 15) of the MD207/208 must be loaded to the GuestBus control register.

Reading/writing one byte from/to the MD207/208 requires two GuestBus cycles: in the first cycle the address (index) of the internal MD207/208 register is written, in the second one the data byte is read/written. The RS (register select) input of the MD207/208 is used to distinguish between the two types of cycles. Connecting this pin to the ZR36057's GADR0 virtually creates two MD207/208 registers at the level of the GuestBus: when GADR0 is low (even registers), the MD207/208 expects address to be transferred on its D7:0 bus, when GADR0 is high (odd registers) data is output or input on these lines. Another way is to connect GADR2 to RS.

Sync Polarity

Since, unlike the ZR36057, the sync polarity of the MD207/208 is not programmable, then, depending on the YUV 4:2:2 source, it might be necessary to invert the HSYNC and VSYNC of the MD207/208.

Vertical Interpolation with the MD208

Pin 8 is the only one that is different between the MD207A and the MD208. While in the MD207A it is a test pin, normally connected to ground, the MD208 uses this input to switch its internal vertical interpolation mechanism on and off. When this mechanism is on, one field out of every pair is vertically interpolated and the interpolated lines are the ones sent out. Since this operation is not always desired (e.g., in high resolution still pictures of VideoCD 2.0) it must be controlled by the software. The natural way to obtain this control is using one of the GPI/Os. It is better to pull this pin down in order for the same layout to support both the MD207A and the MD208.

Appendix C: Fitting the Input Size to the Required Display Window

The ZR36057 can crop the input video and scale it down to match any display size required by different applications, as long as the required size is not larger than the original input. This appendix provides some programming guidelines for proper setting of the ZR36057 parameters involved in this process.

For better understanding, a typical example is detailed along with the general explanations.

It is assumed that the driver software "knows" the following basic parameters about the incoming video:

Wt	Total width of the input field (i.e., in CCIR NTSC Wt = 858).
Wa	Active width of the input field (i.e., in CCIR NTSC Wa = 720).
Ht	Total height of the input frame (i.e., in CCIR NTSC Ht = 525).
Ha	Active height of the input frame (i.e., in CCIR NTSC Ht = 480).

It is assumed that Ha is an even number. It is also assumed that from knowing the video input format, the driver knows how to set the following ZR36057 parameters, such that the entire active portion of the video input would have been sampled:

HSPol	The polarity of HSYNC, as defined in the ZR36057 data sheet.
VSPol	The polarity of VSYNC, as defined in the ZR36057 data sheet.
HStart'	The number of pixels, from the active edge of HSYNC, after which the ZR36057 starts to sample the input.
HEnd'	The number of pixels, from the active edge of HSYNC, after which the ZR36057 stops sampling the input.

The following equation connects HStart', HEnd', and Wa:

$$Wa = HEnd' - HStart' + 1$$

In the example, Hend' = 841, HStart' = 122.

VStart'	The number of lines, from the active edge of VSYNC, after which the ZR36057 starts to sample the input.
VEnd'	The number of lines, from the active edge of VSYNC, after which the ZR36057 stops sampling the input.

The following equation connects VStart', VEnd', and Ha:

$$Ha/2 = VEnd' - VStart' + 1$$

In the example, Vend' = 249, VStart' = 10.

The driver receives from the application software the parameters that define the size of the rectangle, on the monitor, that should be filled with video pixels. This is referred to as the destination rectangle:

VidWinWid	Width of the video rectangle.
VidWinHt	If DispMode=1 (single field display) this is the height of the video rectangle. If DispMode=0 (emulation of interlaced video) this is half of the height of the video rectangle (i.e., the height of one destination field). Note that if DispMode=0 the vertical size of the video rectangle must be an even number. This limitation might conflict with applications that will require a video window of a specific, odd vertical size. Such conflicts will be resolved by the driver software, which in this case will build a rectangle one line shorter than required, and then fill-in the missing line (a-priori) with a background color.

Obviously, in many cases the size of the video rectangle to be displayed is different (smaller) than the size of the active video input sampled by the ZR36057.

In the example: VidWinWid = 597, VidWinHt = 199. (Assuming DispMod = 0, this means that the target "video window" in the example is 597x398).

In such cases, out of the parameters above, the driver software must determine the following ZR36057 parameters, such that the quality of the displayed video is optimal in the sense that the decimation factors will be as small as possible and the portions of the input that are "cut out" (cropped) will also be as small as possible.

hcrop1	The number of pixels, that will be dropped from the beginning of the active line. hcrop1 is not really a ZR36057 parameter, but it is useful to obtain HStart:
	HStart = HStart' + hcrop1.
hcrop2	The number of pixels, that will be dropped from the end of the active line. hcrop2 is not really a ZR36057 parameter, but it is useful to obtain HEnd:
	HEnd = HEnd' - hcrop2.
vcrop1	The number of lines, that will be dropped from the beginning of the active field. vcrop1 is not really a ZR36057 parameter, but it is useful to obtain VStart:
	VStart = VStart' + vcrop1.
vcrop2	The number of lines, that will be dropped from the end of the active field. vcrop2 is not really a ZR36057 parameter, but it is useful to obtain VEnd:
	VEnd = VEnd' - vcrop2.
HorDcm	Ratio of horizontal decimation. A number of HorDcm pixels will be dropped out of every consecutive 64 pixels in an input line. HorDcm/64 is the horizontal decimation factor.
VerDcm	Ratio of vertical decimation. A number of VerDcm pixels will be dropped out of every consecutive 64 lines in an input field. VerDcm/64 is the vertical decimation factor.
HFilter	The horizontal filter through which the input is passed. The filter is selected according to the horizontal decimation factor that is first determined.

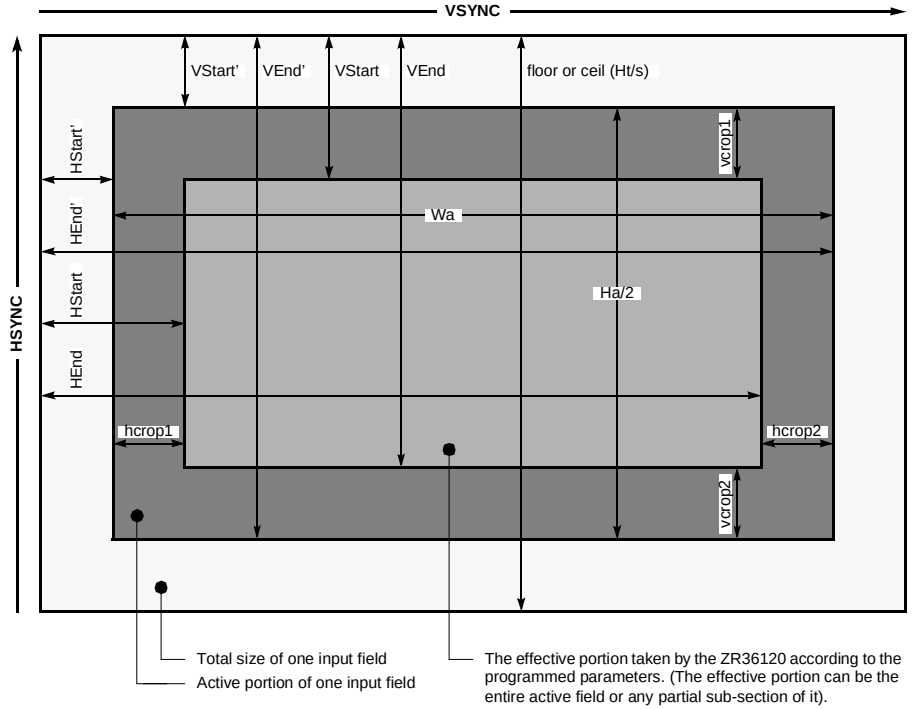


Figure 17. Input Image Parameters

Calculating the Horizontal Parameters:

X and We denote two temporary variables:

X = ceil (VidWinWid*64/Wa)
We = floor (VidWinWid*64/X)
HorDcm = 64 - X
hcrop1 = 2*floor((Wa-We)/4)
hcrop2 = Wa - We - hcrop1

HStart and Hend are then calculated from hcrop1 and hcrop2.

In the example:

X = ceil(597*64/720) = 54
We = floor(597*64/54) = 707
HorDcm= 64 - 54 = 10
hcrop1 = 2*floor((720-707)/4) = 6
hcrop2 = 720-707 - 6 = 7
HStart = HStart' + hcrop1 = 122 + 6 = 128
HEnd = HEnd' - hcrop2 = 841 - 7 = 834
(834 - 128 + 1 = 707)

The ZR36057 will actually sample-in 707 pixels from every line. The first 6 and the last 7 active pixels will be cut out.

Calculating the Vertical Parameters

(Regardless of DispMod!)

Y and He denote two temporary variables:

Y = ceil (VidWinHt*64*2/Ha)
He = floor (VidWinHt*64/Y)
VerDcm = 64 - Y
vcrop1 = floor((Ha/2 - He)/2)
vcrop2 = Ha/2 - He - vcrop1

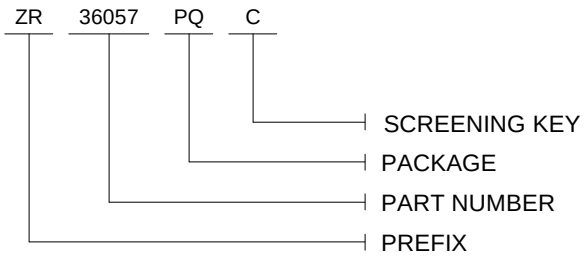
VStart and VEnd are then calculated from vcrop1 and vcrop2.

In the example:

Y = ceil(199*64*2/480) = 54
He = floor(199*64/54) = 235
VerDcm= 64 - 54 = 10
vcrop1 = floor(((480/2) - 235) / 2) = 2
vcrop2 = (480/2) - 235 - 2 = 3
VStart = VStart' + vcrop1 = 10 + 2 = 12
VEnd = VEnd' - vcrop2 = 249 - 3 = 246
(246 - 12 + 1 = 235)

The ZR36057 will actually sample-in 235 lines from every field. The first 2 and the last 3 active lines of every field will be cut out.

ORDERING INFORMATION



PACKAGE

PQ - Plastic Quad Flat Pack (EIAJ)

SCREENING KEY

C - 0°C to +70°C (V_{CC} = 4.75V to 5.25V)

SALES OFFICES

- ***U.S. Headquarters***
  Zoran Corporation
  2041 Mission College Blvd
  Santa Clara, CA 95054 USA
  Telephone: 408-986-1314
  FAX: 408-986-1240

- ***Israel Operations***
  Zoran Microelectronics, Ltd.
  Advanced Technology Center
  P.O. Box 2495
  Haifa, 31024 Israel
  Telephone: 972-4-8545-777
  FAX: 972-4-8551-550