ADVANCE INFORMATION

JPEG IMAGE COMPRESSION PROCESSOR

FEATURES

- Implements JPEG Baseline image compression and expansion, including:
  - DCT/IDCT operations
  - Quantization
  - Variable length coding/decoding
- Full support of the JPEG Baseline standard, including:
  - Bit and byte stuffing
  - JPEG markers including restart (RST), application (APP), and comment (COM)
- JPEG Lossless compression and expansion
- DMA/SLAVE bus interface
- Motion video (30 frames/sec) compression/expansion capability for CCIR resolution (720 x 480)
- "Fast Preview" option
  - Preview of "thumbnail" version of images (up to 25x faster)
- Bit rate control option
  - Guarantees compressed image file size
- Low cost solution
  - Low cost single chip
  - Support for inexpensive memories
  - Requires minimal host intervention
- TTL compatible
- 27 and 21 MSamples/sec data-rate
- Standby mode for very low power consumption
- 100-pin plastic quad flat-pack (PQFP) packaging

APPLICATIONS

- Computer and multimedia add-in boards
- Full-motion video compression/expansion
- Digital still cameras and peripherals
- Security and industrial systems
- Videophones and color FAX machines
- Color printers and scanners
- Fixed bit rate image transmission devices
- Cost-sensitive image compression systems

GENERAL DESCRIPTION

The ZR36050 is a high-speed JPEG Image Compression Processor that performs the algorithm specified by the JPEG Baseline and JPEG Lossless standards for high-quality image compression and expansion of continuous-tone color or monochrome images. The ZR36050 performs Discrete Cosine Transform (DCT), quantization and variable-length encoding for image compression (coding), and the corresponding inverse operations for expansion (decoding).

In the JPEG Baseline encoding operation, the ZR36050 performs the DCT operation on 8 x 8 blocks of image data, converting image data into its spatial frequency components, and quantizes them using a user defined "quantization table."

Because the human visual system is less sensitive at the higher spatial frequencies, these higher frequency components can be quantized more coarsely than the lower-frequency components, with negligible effect on image quality.

The coarser quantization of high-frequency coefficients results in long strings of zero valued quantized coefficients, when the 8x8 blocks are scanned in zigzag order. The scanned coefficients are characterized in terms of their nonzero values and the zero run lengths. As a result, a long string of zeroes is coded as a single number. The ZR36050 then performs Huffman coding using user-defined Huffman tables, whereby bit patterns of different lengths code the nonzero values (values that occur frequently use the shortest codes; while those that infrequently occur use the longest codes). These techniques greatly reduce the amount of memory needed to store an image.

In the decoding operation, the compressed data is decoded (the inverse of the Huffman and the zigzag modified-run-length coding), and dequantized. A 2-D inverse Discrete Cosine Transform is performed on the DCT coefficients, resulting in an expanded image.

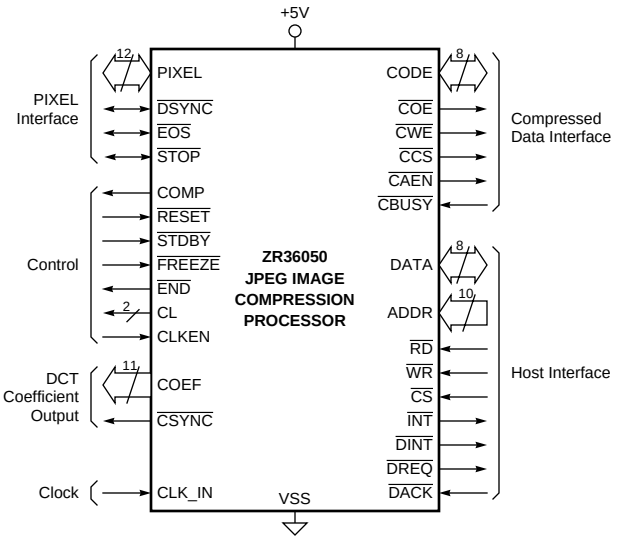


Figure 1. ZR36050 Logical Pinout

This document was created with FrameMaker 4.0.4

In addition to the JPEG Baseline, the ZR36050 supports a subset of JPEG Lossless standard. The ZR36050 performs one dimensional differential prediction followed by variable-length encoding for JPEG Lossless compression, and the corresponding inverse operations for JPEG Lossless expansion.

The ZR36050 maintains full compatibility with the JPEG Baseline standard. The unique ability to perform bit rate control. Bit rate control capability allows the user to preset the size of a compressed image file. This capability is important because without bit rate control, the size of a compressed image is highly data dependent for a given set of quantization tables (images with fine detail generate considerably larger files than files generated from smooth images).

The ability to perform bit rate control is critical for applications where predictable file sizes for compressed images is desired, or where the time allocated to transmit an image across a communications network is fixed. The compressed image file size is constrained to be no greater than a user specified target, and is typically kept within a range of 95% to 100% of this target. The bit rate control feature relies on a two pass algorithm for its operation.

The ZR36050 has the ability to generate a “thumbnail” version of an image for “Fast Preview.” This thumbnail image is a 1/64 scale version of the image, and is generated up to 25 times faster than full image expansion. The thumbnail image is generated from the JPEG Baseline compressed data, and eliminates the need for a separately encoded and stored thumbnail image. This feature is particularly useful for previewing large databases of images.

The ZR36050 operates as a dedicated processor requiring only minimal host intervention. The host processor controls the operation of the device by writing parameter values into the ZR36050’s Internal Memory. Once initialized, the ZR36050 operates continuously until it has completed the compression or expansion of the image. Since the ZR36050 fully complies with the JPEG Baseline standard, the compressed data bit-stream generally requires no intervention by the host. Full JPEG capability also allows for the interchange of files created by other JPEG imaging systems with files generated by systems using the ZR36050.

The ZR36050 is useful for a wide range of motion and still video applications. For example, a typical multimedia application (30 seconds of video at 10 frames/sec and 320 x 240 resolution) would require 69 Mbytes of storage in uncompressed form. With compression using the ZR36050, the requirement can be reduced to 2.9 Mbytes, making storage feasible on a personal computer hard disk. Similarly, for digital still camera applications, the memory requirement can be reduced from a 22 Mbyte hard disk to a 1 Mbyte memory card for storage of twenty 768 x 480 compressed images.

The ZR36050 is fabricated with an advanced low-power CMOS technology, making it suitable for use in low-power, cost-sensitive applications. The device is available in a 100-pin Plastic Quad Flat Pack (PQFP).

Table 1. Signal Description[1, 2]

Signal	Type[3]		Description
	Encode	Decode	
VCC	S	S	+5 volt Power supply. All VCC pins must be connected to +5V.
VSS	S	S	Ground. All VSS pins must be connected to GND.
CLK_IN	I	I	Data Transfer Clock. Provides data transfer timing for the device. All timing is referenced to the rising edge of this clock.
$\overline{\text{RESET}}$	I	I	Reset. This active-low input signal resets all the internal controls and places the ZR36050 in the Idle state. $\overline{\text{RESET}}$ can be activated only when CLKEN is asserted and must remain active for a minimum of four CLK_IN cycles. The STATUS_0, INT_REQ_0, and INT_REQ_1 register bits are reset by this signal. The STATUS_1 bits except the END bit are reset; the END bit is set. $\overline{\text{RESET}}$ initializes the ZR36050 to the compression mode, and activates $\overline{\text{END}}$, $\overline{\text{STOP}}$ and COMP. $\overline{\text{RESET}}$ can be activated during the Standby state; in this case the device draws normal current as long as $\overline{\text{RESET}}$ is active.
$\overline{\text{STDBY}}$	I	I	Standby. This active-low input signal places the ZR36050 in the Standby state. If CLKEN is active, only the internal clock circuit consumes power. If CLKEN is inactive in the Standby state, the device power consumption is further reduced. The ZR36050 should be switched to the Standby state only when it is in the Idle state: after activation of a $\overline{\text{RESET}}$ and prior to loading the Internal Memory, or after the ZR36050 issues an $\overline{\text{END}}$. If CLKEN is active, then $\overline{\text{STDBY}}$ should be deasserted at least four CLK_IN cycles before accessing the Internal Memory. $\overline{\text{RESET}}$ can be activated during the Standby state, only when CLKEN is active. Reading from or writing to the Internal Memory during the Standby state is prohibited.
CLKEN	I	I	Clock Enable. This active-high input signal enables the data transfer clock CLK_IN, and the internal PLL that generates an internal double-frequency clock. When inactive, this signal reduces power further in the Standby state by deactivating the internal clock. The frequency of CLK_IN must be stable before CLKEN is activated. Furthermore, 5000 CLK_IN cycles are required for the PLL to stabilize, after CLKEN has been activated and before the device is ready for operation. If the frequency of CLK_IN is changed without turning off the power, then CLKEN must be reactivated. When $\overline{\text{STDBY}}$ is high, this pin should also be high. For systems in which the 5000 CLK_IN recovery time is not significant, the $\overline{\text{STDBY}}$ and CLKEN pins can be tied together to the external standby signal.
$\overline{\text{FREEZE}}$	I	I	Freeze. This active-low input signal freezes all chip operations. $\overline{\text{FREEZE}}$ is sampled on the rising edge of CLK_IN. Immediately after $\overline{\text{FREEZE}}$ is sampled, all buses float and all activities of the ZR36050 are frozen in their current state. All activities resume normally following the deassertion of $\overline{\text{FREEZE}}$.
$\overline{\text{END}}$	O	O	End Of Process. This active-low output signal indicates the normal end of an encoding or decoding process. If an encoding process ends because of an overflow, $\overline{\text{END}}$ is not activated. $\overline{\text{END}}$ is activated after activation of $\overline{\text{RESET}}$ and at the completion of an encoding or decoding process. It stays activated until a GO command is issued or the STATUS_1 register in the Internal Memory is read.

1. The DATA, CODE, PIXEL, and COEF buses have internal pull-downs that provide 50 microamps of pull-down current at 0.4 volts.
2. The control pins: $\overline{\text{DSYNC}}$, $\overline{\text{EOS}}$, $\overline{\text{STOP}}$, $\overline{\text{END}}$, CL, $\overline{\text{CSYNC}}$, $\overline{\text{COE}}$, $\overline{\text{CWE}}$, $\overline{\text{CCS}}$, $\overline{\text{CAEN}}$, $\overline{\text{INT}}$, $\overline{\text{DINT}}$, $\overline{\text{DREQ}}$ and COMP, have internal pull-up devices that provide 50 microamps at 2.4 volts. These pull-ups are turned on only when $\overline{\text{STDBY}}$ is active but $\overline{\text{RESET}}$ is inactive. When $\overline{\text{STDBY}}$ is active together with $\overline{\text{RESET}}$, the above control pins float.
3. I = Input, O = Output, B = Bidirectional, S = Supply.

Table 1. Signal Description[1, 2] (Continued)

Signal	Type[3]		Description
	Encode	Decode	
CL(1-0)	-	O	Color. During the decoding mode, the CL output signals designate the index of the color component that is being decoded. 00 - First MCU component. 01 - Second MCU component. 10 - Third MCU component. 11 - Fourth MCU component or Idle. In the JPEG Baseline mode, the designation changes with the falling edge of the first $\overline{\text{DSYNC}}$ of the component or with the falling edge of $\overline{\text{EOS}}$. In the Fast Preview and Lossless decoding modes, CL is active together with the $\overline{\text{DSYNC}}$ which precedes the DC coefficient value and the pixel data, respectively. CL is undefined in the encoding mode.
$\overline{\text{STOP}}$	O	I	Stop Sending/Receiving. This active-low bidirectional signal is an input in encoding and output in decoding modes. $\overline{\text{STOP}}$ is used for the following purposes: At the start of and during an encoding operation, $\overline{\text{STOP}}$ is an output signal indicating that the ZR36050 is not ready to receive image data. $\overline{\text{STOP}}$ is activated when the ZR36050 is in the Idle state, and when reading or processing Internal Memory parameters during the encoding modes. In the encoding mode, $\overline{\text{STOP}}$ is an output signal indicating that three of the ZR36050's four internal coefficient buffers are full. In this case, $\overline{\text{STOP}}$ is output 42 CLK_IN cycles prior to the last image data sample of the current block that is being input. If $\overline{\text{STOP}}$ remains active until the next $\overline{\text{DSYNC}}$ is due, then the system must not input the next $\overline{\text{DSYNC}}$ and the image data block. $\overline{\text{STOP}}$s that are deactivated prior to the next $\overline{\text{DSYNC}}$ can be ignored. The system can resume inputting the next image data block immediately after $\overline{\text{STOP}}$ is deactivated. In the Lossless encoding mode, the system must stop inputting data within three CLK_IN cycles of activation of $\overline{\text{STOP}}$ to prevent overflow. In the decoding mode, $\overline{\text{STOP}}$ is an input signal that notifies the ZR36050 that it should not assert the next $\overline{\text{DSYNC}}$ and consequently delay output of the next decoded image data block at the end of the current block. In this case, $\overline{\text{STOP}}$ must be activated at least 24 CLK_IN cycles before the last image sample of the current block that is being output and must remain active at least until the end of the current block. Once $\overline{\text{STOP}}$ is deactivated, the ZR36050 outputs the next $\overline{\text{DSYNC}}$ followed by its corresponding image data block, at least 17 CLK_IN cycles after deactivation of $\overline{\text{STOP}}$. In the Lossless decoding and Fast Preview modes, when $\overline{\text{STOP}}$ is activated or deactivated, the ZR36050 stops or resumes delivering image data after 2 CLK_IN cycles.
PIXEL(11-0)	I	O	Pixel bus. This 12-bit unsigned bidirectional bus is used for the following purposes: In the encoding modes, the most significant 8 bits are used to carry the input image data. The remaining 4 bits are "don't care" and can be left as unconnected pins. In the Lossless encoding mode, it is a 12-bit input bus. If fewer than 12 bits are required, then the most significant bits of the bus are used to carry the input image data. In the decoding mode, the most significant 8 bits are used to carry the output image data. The least significant 4 bits are forced to "0". In the Lossless decoding mode, this is a 12-bit output bus. If fewer than 12 bits are used, then the most significant bits of the bus are used to carry the output image data and the unused bits are forced to "0". In the Fast Preview mode, it is an output bus carrying the 11-bit unsigned DC coefficient values on the most significant 11 bits of the bus. The least significant bit is forced to "0". The input/output data in the encoding and decoding modes is ordered in row-by-row scanned 8x8 blocks. In the Lossless encoding and decoding modes, the image data is scanned row by row.

1. The DATA, CODE, PIXEL, and COEF buses have internal pull-downs that provide 50 microamps of pull-down current at 0.4 volts.
2. The control pins: $\overline{\text{DSYNC}}$, $\overline{\text{EOS}}$, $\overline{\text{STOP}}$, $\overline{\text{END}}$, CL, $\overline{\text{CSYNC}}$, $\overline{\text{COE}}$, $\overline{\text{CWE}}$, $\overline{\text{CCS}}$, $\overline{\text{CAEN}}$, $\overline{\text{INT}}$, $\overline{\text{DINT}}$, $\overline{\text{DREQ}}$ and COMP, have internal pull-up devices that provide 50 microamps at 2.4 volts. These pull-ups are turned on only when $\overline{\text{STDBY}}$ is active but $\overline{\text{RESET}}$ is inactive. When $\overline{\text{STDBY}}$ is active together with $\overline{\text{RESET}}$, the above control pins float.
3. I = Input, O = Output, B = Bidirectional, S = Supply.

Table 1. Signal Description[1, 2] (Continued)

Signal	Type[3]		Description
	Encode	Decode	
$\overline{\text{DSYNC}}$	I	O	Data Synchronization. This active- low signal is an input in encoding and output in decoding modes. In the encoding modes, $\overline{\text{DSYNC}}$ marks the start of an 8x8 image data block and should appear as an input one CLK_IN before the first image data of a block. In the decoding modes, $\overline{\text{DSYNC}}$ is output one CLK_IN before the first image data sample of a block. The width of $\overline{\text{DSYNC}}$ is one CLK_IN cycle. In the Fast Preview mode, and the Lossless encoding and decoding modes, this signal precedes each image data sample.
$\overline{\text{EOS}}$	I	O	End Of Scan. This active-low signal is an input in encoding modes. $\overline{\text{EOS}}$ indicates the last image data sample of each scan entering the ZR36050. In encoding modes, $\overline{\text{EOS}}$ must be input regardless of the $\overline{\text{STOP}}$ signal. $\overline{\text{EOS}}$ is an output signal in the decoding mode. It is generated together with the last image data sample of each scan leaving the ZR36050. In this case, $\overline{\text{DSYNC}}$ will not be issued. In the Fast Preview and Lossless decoding modes, $\overline{\text{EOS}}$ is output within 64 CLK_IN cycles after the last sample of a scan. It is merely used in as an indication of the completion of the current process without having any timing significance. In decoding mode, $\overline{\text{EOS}}$ is output regardless of the $\overline{\text{STOP}}$ signal. The width of $\overline{\text{EOS}}$ is one CLK_IN cycle.
COEF(10-0)	O	O	Coefficient Bus. This 11-bit output bus is used to transfer DCT coefficients out of the device in the encoding and decoding modes. The DCT coefficients are output in column-major order. This bus is not used in the Fast Preview and Lossless encoding and decoding modes.
$\overline{\text{CSYNC}}$	O	O	Coefficient Synchronization. This active-low signal indicates the beginning of an 8x8 DCT coefficient block. In the encoding and decoding modes, this signal is generated by the ZR36050. It is asserted one CLK_IN cycle before the first coefficient of a block is placed on the COEF bus by the ZR36050. The width of $\overline{\text{CSYNC}}$ is one CLK_IN cycle. $\overline{\text{CSYNC}}$ is not used in the Fast Preview and Lossless encoding and decoding modes.
CODE(7-0)	O	I	Code. In Master mode Compressed Data Transfer, this 8-bit bidirectional bus is used to read the compressed data from or write to the Compressed Data Memory. In the 16-bit Slave and DMA modes, this bus is used as an extension of the DATA bus.
$\overline{\text{COE}}$	-	O	Compressed Data Memory Read. This active-low output signal acts as a read pulse from the ZR36050 to the Compressed Data Memory. $\overline{\text{COE}}$ goes active 0.5 CLK_IN cycles after the start of a read cycle and remains active until the end of the read cycle. The CODE bus is latched on the rising edge of $\overline{\text{COE}}$.
$\overline{\text{CWE}}$	O	-	Compressed Data Memory Write. This active-low output signal acts as a write pulse from the ZR36050 to the Compressed Data Memory. $\overline{\text{CWE}}$ goes active 0.5 CLK_IN cycles after the start of a write cycle and remains active until the end of the write cycle.
$\overline{\text{CCS}}$	O	O	Compressed Data Memory Chip Select. This active-low output signal acts as a chip select signal from the ZR36050 to the Compressed Data Memory. $\overline{\text{CCS}}$ goes active at the start of a read or write cycle and remains active throughout the cycle. $\overline{\text{CCS}}$ remains active continuously in back to back read or write cycles. The length of a read or write cycle can be from one to eight CLK_IN periods.
$\overline{\text{CAEN}}$	O	O	Address Counter Enable. This active-low output signal can be used to advance an external Compressed Data Memory address counter.

1. The DATA, CODE, PIXEL, and COEF buses have internal pull-downs that provide 50 microamps of pull-down current at 0.4 volts.
2. The control pins: $\overline{\text{DSYNC}}$, $\overline{\text{EOS}}$, $\overline{\text{STOP}}$, $\overline{\text{END}}$, CL, $\overline{\text{CSYNC}}$, $\overline{\text{COE}}$, $\overline{\text{CWE}}$, $\overline{\text{CCS}}$, $\overline{\text{CAEN}}$, $\overline{\text{INT}}$, $\overline{\text{DINT}}$, $\overline{\text{DREQ}}$ and COMP, have internal pull-up devices that provide 50 microamps at 2.4 volts. These pull-ups are turned on only when $\overline{\text{STDBY}}$ is active but $\overline{\text{RESET}}$ is inactive. When $\overline{\text{STDBY}}$ is active together with $\overline{\text{RESET}}$, the above control pins float.
3. I = Input, O = Output, B = Bidirectional, S = Supply.

Table 1. Signal Description[1, 2] (Continued)

Signal	Type[3]		Description
	Encode	Decode	
$\overline{\text{CBUSY}}$	I	I	Compressed Data Memory Busy. This active-low input signal indicates that the Compressed Data Memory is busy. During the encoding modes, $\overline{\text{CBUSY}}$ active means that the ZR36050 cannot write to the Compressed Data Memory. During the decoding modes, $\overline{\text{CBUSY}}$ active means that the ZR36050 cannot read from the Compressed Data Memory. If $\overline{\text{CBUSY}}$ is activated at least one CLK_IN prior to the beginning of a read or write cycle, then the next read or write cycle will not be performed. The minimum width of $\overline{\text{CBUSY}}$ is one CLK_IN cycle.
ADDR(9-0)	I	I	Internal Memory Address. This 10-bit input bus is used to address the Internal Memory of the ZR36050.
DATA(7-0)	B	B	Internal Memory Data Bus. This 8-bit bidirectional bus is used to read from or write to the Internal Memory of the ZR36050. In the 16-bit Slave and DMA modes, the CODE bus is used as an extension of the DATA bus.
$\overline{\text{RD}}$	I	I	Read. This active-low input signal acts as a read pulse from the host to the ZR36050.
$\overline{\text{WR}}$	I	I	Write. This active-low input signal acts as a write pulse from the host to the ZR36050. The DATA bus is latched on the rising edge of $\overline{\text{WR}}$.
$\overline{\text{CS}}$	I	I	Chip Select. This active-low input signal acts as a chip select signal from the host to the ZR36050.
$\overline{\text{INT}}$	O	O	Interrupt. This active-low output signal notifies the host that one of the STATUS bits, except for DATRDY, is set. $\overline{\text{INT}}$ is reset by reading the relevant STATUS register, by activation of $\overline{\text{RESET}}$, or by a GO command.
$\overline{\text{DINT}}$	O	O	Data Ready Interrupt. In Slave mode Compressed Data Transfer, this active-low output signal notifies the host that the DATRDY bit in the STATUS_1 register is set. $\overline{\text{DINT}}$ is reset by reading the STATUS_1 register, by activation of $\overline{\text{RESET}}$, or by reading or writing the compressed data in the Compressed Data Input/Output register at address 30H of the ZR36050 Internal Memory with $\overline{\text{CS}}$ active.
$\overline{\text{DACK}}$	I	I	DMA Acknowledge. This active-low input signal is used in DMA mode Compressed Data Transfer. $\overline{\text{DACK}}$ is an acknowledgment pulse from the DMA controller to the ZR36050. It must be active during the entire Read or Write cycle.
$\overline{\text{DREQ}}$	O	O	DMA Request. This active-low output signal is used in DMA mode Compressed Data Transfer. $\overline{\text{DREQ}}$ is the DMA request from the ZR36050 to the host. The ZR36050 does not output a $\overline{\text{DREQ}}$ until the $\overline{\text{DACK}}$ signal to the previous $\overline{\text{DREQ}}$ has been deactivated.$\overline{\text{DREQ}}$ is deactivated by $\overline{\text{RESET}}$, or by $\overline{\text{DACK}}$.
COMP	O	O	Compress/Expand. This output signal provides an indication of the current operating mode of the ZR36050. When it is high, the ZR36050 is in the encoding mode; when it is low, the ZR36050 is in the decoding mode. The mode and the state of COMP are changed when the MODE register in the Internal Memory is read by the ZR36050 after a GO command is issued by the host. One CLK_IN cycle after COMP changes state, $\overline{\text{EOS}}$, $\overline{\text{STOP}}$, and $\overline{\text{DSYNC}}$ change directions. Activation of $\overline{\text{RESET}}$ sets COMP high.

1. The DATA, CODE, PIXEL, and COEF buses have internal pull-downs that provide 50 microamps of pull-down current at 0.4 volts.
2. The control pins: $\overline{\text{DSYNC}}$, $\overline{\text{EOS}}$, $\overline{\text{STOP}}$, $\overline{\text{END}}$, CL, $\overline{\text{CSYNC}}$, $\overline{\text{COE}}$, $\overline{\text{CWE}}$, $\overline{\text{CCS}}$, $\overline{\text{CAEN}}$, $\overline{\text{INT}}$, $\overline{\text{DINT}}$, $\overline{\text{DREQ}}$ and COMP, have internal pull-up devices that provide 50 microamps at 2.4 volts. These pull-ups are turned on only when $\overline{\text{STDBY}}$ is active but $\overline{\text{RESET}}$ is inactive. When $\overline{\text{STDBY}}$ is active together with $\overline{\text{RESET}}$, the above control pins float.
3. I = Input, O = Output, B = Bidirectional, S = Supply.

FUNCTIONAL OVERVIEW

Figure 2 is a functional block diagram of the ZR36050 JPEG Image Compression Processor. The ZR36050 consists of two major processing units, the DCT Unit and the Encoding/Decoding Unit, with their associated buffers, an Internal Memory for data exchange with the host processor, special internal storage for tables, and three bus interfaces.

During a compression operation, image data flows in through the Pixel Interface and compressed data flows out through either the Compressed Data Interface or the Host Interface. The direction of data flow is reversed during an expansion operation.

Pixel Interface

In compression of an image, the source of the image data samples is typically a strip, field, or frame buffer (see Figure 3, for example). The external image buffer control logic writes the image samples into the ZR36050 via the Pixel Interface, which operates synchronously, transferring each sample in on the rising edge of the clock. A synchronizing signal, driven by the buffer control logic, indicates to the Pixel Interface when valid samples are being presented to it.

In JPEG Baseline compression, each color component of the image must be partitioned into blocks of 8 x 8 samples. The image data enters the Pixel Interface one component block at a time, each block starting with the top left sample, scanned by rows, and ending with the bottom right sample. As required by the JPEG Baseline standard, the blocks are grouped into a repeating pattern of Minimum Coded Units, or MCU's. For example, in an interleaved scan of Y, U, and V components, with 4:2:2 subsampling, the MCU consists of two blocks of Y, one block of U, one block of V, and image blocks enter the device in the following order:

$$Y_0\ Y_1\ U_0\ V_0\ Y_2\ Y_3\ U_1\ V_1\ Y_4\ Y_5\ U_2\ V_2\ ....$$

where each component designator (Y, U or V) represents a block of 64 samples, and the subscript indicates the block index.

On the other hand, in JPEG Lossless compression, the image components enter the Pixel Interface in a normal raster. In the above example, the MCU then consists of two Y samples, one U sample and one V sample.

When outputting expanded image data, the Pixel Interface drives the synchronizing signal and the pixel data bus.

DCT Unit and Coefficient Buffers

In JPEG Baseline compression, the DCT Unit transforms each component block into 64 DCT coefficients, and writes the coefficients into the next available DCT Coefficient Block Buffer. At the same time, it outputs the DCT coefficients on the Coefficient Bus. In expansion, it performs the inverse DCT, whenever a complete block of DCT coefficients is available in the buffer.

Because JPEG Lossless compression uses a spatial domain algorithm, when the ZR36050 operates in Lossless mode, the DCT Unit and Coefficient Block Buffers are bypassed. There is also a Fast Preview mode (described later) for expansion of JPEG Baseline compressed data, in which the Inverse DCT Unit and Coefficient Buffers are also bypassed.

Encoding/Decoding Unit, Code Buffer

The Encoding/Decoding Unit performs the remainder of the JPEG compression and also formats the compressed data — including the insertion of JPEG bit and byte stuffing, JPEG markers, marker segments and their associated parameters. The compressed data conforms completely to the JPEG format and no reformatting or additional parameter insertion is needed.

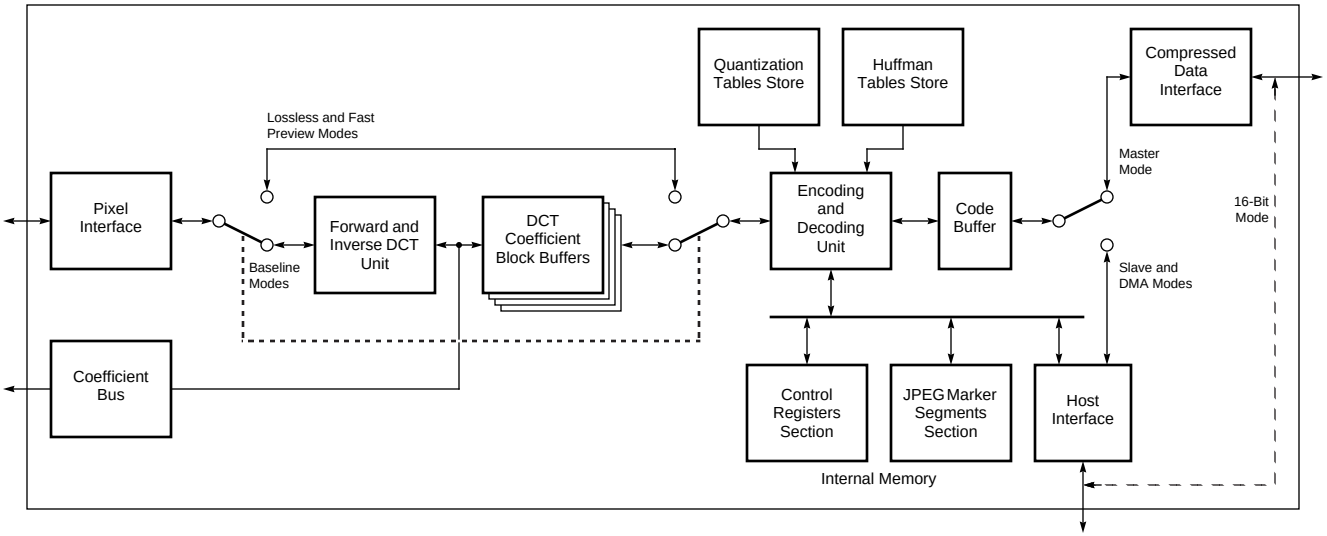


Figure 2. ZR36050 Functional Block Diagram

In JPEG Baseline compression, the Encoding Unit reads the DCT coefficients from the Block Buffers in the JPEG zig-zag sequence and quantizes them using the tables in the Quantization Table Store. The Quantization Table Store holds up to four quantization tables. Each component of the image is assigned one of the quantization tables; this assignment is specified in the JPEG frame header.

The Encoding Unit encodes the difference of the quantized DC coefficients of the current and previous blocks, using one of the DC tables in the Huffman Table Store. After accumulating runs of zero-valued AC coefficients, it encodes the zero run lengths and the non-zero AC coefficients using the AC tables in the Huffman Table Store. The Huffman Table Store has space for two DC Huffman tables and two AC Huffman tables. Each component of a scan is assigned to one DC table and one AC table; this assignment is specified in the JPEG scan header.

Quantization tables are not used in Lossless compression. In this mode of operation, the Encoding Unit performs one-dimensional, horizontal predictive coding and Huffman coding of the samples, similar to that of the DC coefficients in Baseline compression.

When it has generated a Huffman code, or when it is transferring the marker segment data, the Encoding Unit writes the compressed data into the Code Buffer. From there, the compressed data is transferred out of the device, either via the Compressed Data Interface, or via the Host Interface.

The procedure for JPEG Baseline or Lossless image expansion is the inverse of the corresponding compression procedure When expanding an image, the Encoding/Decoding Unit detects and decodes all the markers and marker segment parameters included in the compressed data. The host does not need to extract or decode parameters, such as tables, from the JPEG compressed data file, since this is done automatically.

Compressed Data Interface

The Compressed Data Interface is the fastest means of transferring the compressed data into or out of the ZR36050, therefore it could be used in a motion video compression application, (Refer to the example shown in Figure 3). Since it can optionally be configured to operate with up to seven internally generated wait states, it is also suitable for use with a slow compressed data store, such as a memory card. When the Compressed Data Interface is being used, it transfers data in a Master mode, driving the access control signals to an external auto-incrementing (e.g. FIFO) memory device.

Host Interface

If the ZR36050 is configured to transfer the compressed data via the Host Interface, it can do so in one of two submodes. Slave or DMA modes. The principal function of the Host Interface is to allow the host to access the Internal Memory. This access is required in order to program the operating mode of the device, specify the JPEG marker segments and their parameters for compression, initiate the encoding or decoding operation and read the status of the device.

Internal Memory

The Internal Memory is partitioned into a Control Registers Section and a JPEG Marker Segments Section.

The Control Registers Section contains the various configuration registers, status registers for interaction with the host, and informational registers that provide feedback to the host after the completion of an operation by the ZR36050.

The Marker Segments Section is where the host writes the contents of the JPEG marker segments before initiating compression of an image or changing markers between the frame and scan marker, or between scans. After an expansion, it

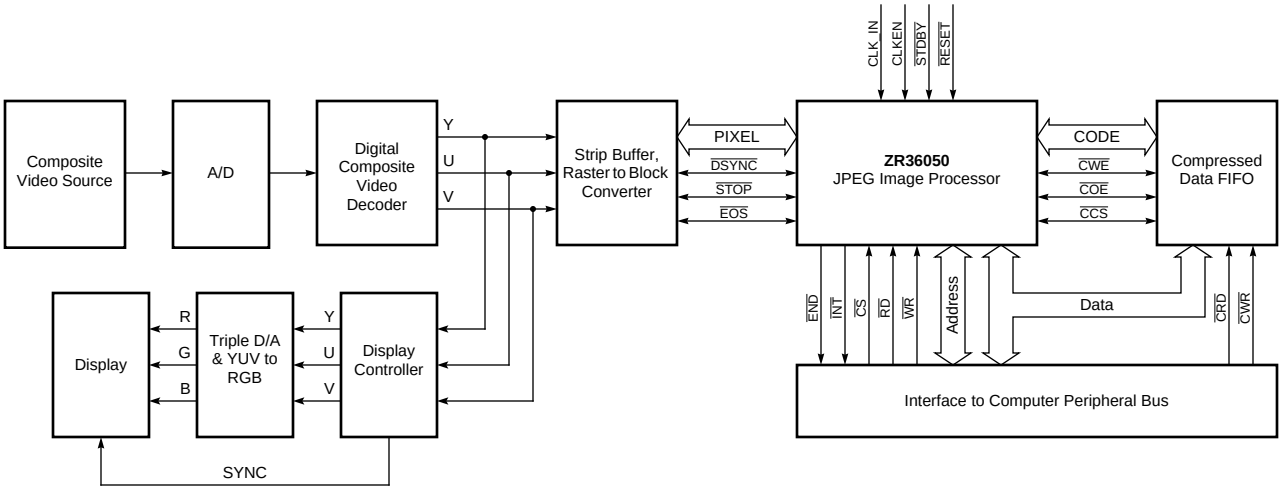


Figure 3. Typical Motion JPEG Compression/Expansion System Configuration

contains the marker segments that the Encoding/Decoding Unit extracted from the compressed data.

The Internal Memory space also contains a write-only virtual register (at address 0) by means of which the host issues the GO command to start operation. In Slave mode Compressed Data Transfer, the host reads the compressed data from or writes it into, a Compressed Data Input/Output register. That is also mapped into the Internal Memory space.

PROGRAMMING THE ZR36050

The host configures the processing units and interfaces of the ZR36050 for the desired operation by means of the Control Registers and Marker Segments sections of the Internal Memory.

Many of the parameters that determine the proper operating configuration of a JPEG encoder or decoder are embedded in the marker segments that are included with JPEG compressed image data. For example, the JPEG frame header specifies the number of data units (blocks or samples) of each image component in the MCU, and the scan header specifies the number of image components in the scan. This information is essential for the Encoding/Decoding Unit to perform the quantization and Huffman coding correctly. The ZR36050 decodes these parameters automatically from marker segments in JPEG format. In compression, these marker segments must be written in the Marker Segments Section of Internal Memory by the host, before it initiates the compression. In expansion, the marker segments are part of the compressed data; the Encoding/Decoding Unit copies them from the Code Buffer into Internal Memory before decoding the parameters.

The basic operating mode and interface configuration are specified in the MODE and HARDWARE control registers. For an image expansion operation, these and the INT_REQ registers are the only ones that need to be initialized. Other control registers, such as the OPTIONS and MARKERS_EN registers, are used only in compression operation or Table Preload modes.

In addition to the frame (SOFn) and scan (SOSn) header marker segments, the Internal Memory has space allocated for the Define Restart Interval (DRI), Define Number of Lines (DNL), Define Quantization Tables (DQT), Define Huffman Tables (DHT), Application (APP) and Comment (COM) marker segments, which can optionally be included with the compressed image data in compression. The MARKERS_EN register provides the means by which the host can specify which of the optional marker segments are to be included in the compressed data.

For further details on programming, refer to the section “INTERNAL MEMORY FORMAT.”

STATUS AND INTERRUPTS

The ZR36050 has two status registers, that are readable by the host and provide feedback on the various events occurring during an encoding or decoding operation. Associated with the status registers are two interrupt enable registers, the bits of which correspond to those of the status registers. If the host sets an interrupt enable bit, the ZR36050 will interrupt it when the corresponding status bit becomes active and stops processing. The host can then take appropriate action before commanding the ZR36050 to continue processing (by means of the GO command).

The STATUS_0 and INT_REQ_0 register bits correspond to the names of JPEG marker segments. In a decoding operation, the ZR36050 sets a STATUS_0 bit when the corresponding marker segment has been extracted from the compressed data stream and written in the Internal Memory. If the INT_REQ_0 bit was also set, the host can read out the marker segment content before continuing.

The APP and COM bits of STATUS_0 have a special mechanism associated with them, that makes it possible for the host to read out a segment of any length, even though there are only 64 bytes allocated to each of the APP and COM segments in Internal Memory. If the corresponding interrupt is enabled, and the segment is longer than 64 bytes, the ZR36050 stops after writing each 64 byte portion in Internal Memory, and waits for a GO command. The last portion may have fewer than 64 bytes.

The APP and COM bits of STATUS_0 are also used, with a similar mechanism, to include APP or COM segments longer than 64 bytes in the compressed data. In this case, the host programs the APP or COM bit of the MARKERS_EN register, writes the first 64 byte portion of the segment in the Internal Memory, and enables the interrupt in the INT_REQ_0 register. When the ZR36050 has transferred the portion to the Code Buffer, it stops as before, if the length parameter indicated a segment longer than 64 bytes. The host can then write the next 64 byte portion in the Internal Memory, and repeat the procedure until the whole segment is transferred. The ZR36050 does not stop after transferring the last portion of 64 or fewer bytes.

The STATUS_1 and INT_REQ_1 register bits indicate miscellaneous conditions: readiness for data or new optional marker segments, end of process, and overflow conditions. A special bit in STATUS_1, consisting of the logical OR of all the bits of STATUS_0, permits the host to determine the type of event that caused an interrupt, by reading only the STATUS_1 register.

CODE VOLUME CONTROL

The JPEG Baseline encoding algorithm does not inherently provide a means of achieving a target code volume for a compressed image (or an equivalent target average coding rate in bits per pixel, or target compression ratio). In fact, since the standard specifies that the whole image frame is encoded using the same set of quantization tables and Huffman coding tables, it provides no built-in mechanism to modify the coding rate once encoding of an image has started.

One way to achieve a target code volume in compression of a particular image is to iterate the compression, using different quantization tables each time, until the desired target is achieved. The Scale Factor (SF register in the Internal Memory) allows the host to modify the quantization tables by changing a single parameter. In operation, the ZR36050 multiples the quantization tables, specified in the DQT segment of Internal Memory, by the Scale Factor before storing them in the Quantization Tables Store.

When it is activated with appropriate values of the Scale Factor and an Allocation Factor (AF register), the unique Bit Rate Control feature of the ZR36050 ensures that the actual code volume achieved is equal to or slightly less than the target code volume. Moreover, if the Scale Factor is close enough to the correct value, the Bit Rate Control operates without affecting the compressed image quality. The ZR36050 has a Statistical Pass mode of operation, in which it performs a preliminary pass over the image data, and determines appropriate values of the Scale Factor and Allocation Factor for use in a Compression Pass with Bit Rate Control. The host specifies the target code volume to be used in the computation of the Scale Factor and Allocation Factor, in the TCV_DATA register. There is also a mode of operation in which the ZR36050 performs a Statistical Pass followed automatically by a Compression Pass with Bit Rate Control. For more details on these modes of operation, see the “ENCODING MODES” section.

The code volume overflow detection option (activated by the OVF bit of the OPTIONS register) is useful if Bit Rate Control is not used, or if the Scale Factor and Allocation Factor contain inappropriate values. With this option selected, the ZR36050 continuously compares the accumulated code volume with a target limit specified in the TCV_NET register. If the code volume exceeds the limit at any time during a compression pass, the TCVOVF status register bit is set, and the interrupt pin is activated to notify the host. Since this is considered a destructive event, the ZR36050 aborts the compression pass and goes Idle.

COMPRESSED DATA FORMAT

The ZR36050 supports encoding and decoding of all three compressed data format classes defined by the JPEG standard:

- The full interchange format, in which the specifications of all the tables used for encoding the image are included with the compressed image data, in their marker segments. This format is used when the compressed image must be decoded by a JPEG decoder that has no knowledge of the tables used.
- The abbreviated format for compressed data. This format contains the compressed image data and the frame and scan headers, but where some or all of the tables are omitted.
- The abbreviated tables-only format. This format contains marker segments with table specifications but no compressed image data.

The two abbreviated formats complement each other in applications based on the ZR36050. For example, when a sequence of image frames, such as a video clip, are compressed, the same Huffman tables are used for all the images in the sequence. If the compressed sequence is to be sent to a JPEG decoder that has no prior knowledge of the tables, it is most economical for the encoder to create initially, an abbreviated format compressed data-less image, containing only the Huffman tables, followed by a sequence of compressed images in which the Huffman tables are omitted. To expand the image sequence, the JPEG decoder first decodes the tables-only data, thus installing the Huffman tables, which are subsequently used in the decoding of all the images in the sequence.

Compressed data generated by the ZR36050 fully complies with the JPEG standard and includes all headers and marker segments necessary to allow it to be decoded by a compliant JPEG decoder. The JPEG standard allows virtually unlimited permutations in the order and repetition of optional marker segments. As a practical matter, there are some restrictions on the permutations and repetitions in compressed data generated by the ZR36050:

- The order of optional marker segments is fixed, as follows: APP, COM, DRI, DQT, DHT.
- Only one instance of each optional marker segment can be inserted before each SOS, or the SOF, marker.
- The frame header can specify up to eight image components.
- A compressed image can contain up to eight scans.

A compressed image to be decoded by the ZR36050 must be in JPEG standard format. The ZR36050 recognizes and decodes all the following marker types: SOI, SOF (SOF0 or SOF3), SOS, APP, COM, DRI, RST, DQT, DHT, DNL, EOI. All marker segments other than JPEG Baseline marker segments, i.e., markers followed by a length parameter, are disregarded and do not cause any error. Markers without a following length parameter, except for SOI, RST and EOI, will cause unpredictable behavior. A marker can be prefixed by any number of FF bytes.

In its decoding modes, the ZR36050 can expand any JPEG Baseline compressed image. The finite size of the Internal

Memory, however, does impose some restrictions, that have only a very minor effect on the applicability of the device:

- The number of image components specified in the frame header must be eight or fewer.
- A DHT marker segment must have a length of 420 or fewer bytes (excluding the DHT marker), otherwise the tables may be decoded incorrectly into the Huffman Table Store.

  A JPEG Baseline DHT segment has a length of only 418 bytes, so this is not normally a problem. The restriction, however, stems from the fact that the standard does not explicitly disallow the repetition of the same tables in a DHT segment.
- A DQT marker segment of any length is decoded correctly into the Quantization Tables Store. If, however, its length (excluding the DQT marker) is greater than 262 bytes, it may overwrite the DHT, APP, and COM segments in Internal Memory, affecting the ability of the host to read out the contents of these segments correctly.

  A JPEG Baseline DQT segment has a maximum length of 262 bytes. The restriction again stems from the fact that the standard does not explicitly disallow the repetition of the same tables in a DQT segment.

In addition to the above restrictions, to be decoded correctly, the frame header of a JPEG Lossless compressed image must specify a precision of 12 or fewer bits, and only horizontal subsampling. The scan header must select a type 1 predictor (one-dimensional horizontal), and can specify at most two different Huffman tables.

OPERATING MODES

The host sets the operating mode of the ZR36050 by programming the MODE register. Nine distinct modes can be selected, falling into two categories:

- six **encoding modes**, involving compression and associated functions. The encoding modes are: JPEG Baseline Compression Pass, Auto Bit Rate Control, Statistical Pass, Compression Pass with Bit Rate Control, Tables-only Pass, and Tables Preload.
- three **decoding modes**, involving expansion: JPEG Baseline Expansion, Fast Preview, and Tables Preload.

Two additional modes, the Lossless compression and expansion modes of operation, are not distinguished from the JPEG Baseline Compression Pass and Expansion, respectively, in the programming of the MODE register. Rather, the ZR36050 enters the JPEG Baseline or Lossless mode based on the SOF (Start Of Frame) marker. If the marker found in internal memory (compression) or the compressed data (expansion) is SOF0 (FFC0), the ZR36050 configures itself for JPEG Baseline operation. Otherwise, if the marker is SOF3 (FFC3), it configures itself for Lossless operation.

Auto Bit Rate Control, Statistical Pass, and Compression Pass with Bit Rate Control are relevant only to JPEG Baseline compression, and Fast Preview is useful only with JPEG Baseline compressed data. These modes have no meaning for Lossless operation.

ENCODING MODES

JPEG Baseline Compression Pass

The Compression Pass performs the Baseline encoding operation on the input image component samples. During a Compression Pass, the ZR36050 reads the JPEG marker segment information written by the host in the Marker Segments Section of Internal Memory, and uses it to determine the MCU configuration, and includes the compulsory and optional marker segments (selected by the MARKERS_EN register) in the compressed data stream. Note that, if the DQT marker segment is enabled, the ZR36050 first multiplies the quantization tables specified in the DQT segment of Internal Memory by the Scale Factor, and stores the scaled tables in the Quantization Table Store. The quantization tables included in the compressed data are the same as the stored (scaled) tables. At the completion of the JPEG Baseline Compression Pass, the ZR36050 calculates a New Scale Factor (NSF) and saves it in the SF Internal Memory register. The NSF can be used in the next encoding operation or the host can overwrite it by its own Scale Factor.

Statistical Pass

In the Statistical Pass, the ZR36050 performs the computations for JPEG Baseline encoding of the image, with the initially specified Scale Factor, but without transferring any data to the Code Buffer. It accumulates the code volume and a total activity measure. Based on the Target Code Volume (TCV_DATA register), it calculates the Allocation Factor and a new Scale Factor at the end of the pass. It writes the new Scale Factor in the SF register, in place of the initial Scale Factor, and the Allocation Factor, Accumulated Code Volume, and Total Activity measure, in their respective registers (SF, AF, ACV, and ACT), where the host can access them if needed.

Compression Pass with Bit Rate Control

This mode allows the user to ensure a compressed data volume equal to or slightly less than the Target Code Volume. Before encoding each block, the ZR36050 computes a measure of the block activity, and allocates a code volume to the block based on the activity and the Allocation Factor (AF register). During encoding of the block, if the accumulated code volume for the block exceeds the allocation, the ZR36050 truncates the code for the block. The code is also truncated if it exceeds the Maximum Block Code Volume specified in the MBCV register. Aside from the bit rate control, this mode is the same as a JPEG

Baseline Compression Pass, and the resulting compressed data is fully JPEG compatible. At the completion of the Compression Pass with or without Bit Rate Control, the ZR36050 calculates a New Scale Factor (NSF) and saves it in the SF Internal Memory register. The NSF can be used in the next encoding operation or the host can overwrite it by its own Scale Factor.

Auto Bit Rate Control

In this mode the ZR36050 performs a Statistical Pass followed automatically (without host intervention) by a Compression Pass with Bit Rate Control. The ZR36050 computes the new Scale Factor and the Allocation Factor at the end of the Statistical Pass, and rescales the quantization tables by the new scale factor at the start of the Compression Pass. The DQT marker must be enabled for Auto Bit Rate Control to work correctly.

Tables-Only Pass

In this mode, the ZR36050 generates compressed data in the JPEG abbreviated format for table specification. The abbreviated format compressed data contains only the SOI marker, quantization and/or Huffman tables specifications (DQT and/or DHT marker segments), optional APP, COM marker segments, and the EOI marker. The content of the MARKERS_EN register specifies which marker segments are to be included in the abbreviated format data. The Pixel Interface is inoperative in this mode.

Tables Preload for Encoding

Prior to encoding a sequence of images with the same quantization and/or Huffman tables, this mode is used to preload the tables. The DQTI and DHTI bits of the MARKERS_EN register specify which tables to preload. If DQTI is set, the ZR36050 multiplies the quantization tables, specified in the DQT segment of Internal Memory by the Scale Factor, and stores the scaled tables in the Quantization Tables Store. If DHTI is set, it decodes the Huffman tables specifications from the DHT segment of Internal Memory, where they are specified in accordance with the JPEG syntax, and stores the decoded tables in the Huffman Tables Store, for use in compressing the images. In this mode, the Pixel Interface and the Code Buffer are inoperative.

JPEG Lossless Compression

In a Compression Pass, if the ZR36050 finds the SOF3 frame marker in the Internal Memory, it switches to the JPEG Lossless Compression mode. JPEG Lossless compression uses a spatial algorithm, so the DCT Unit is bypassed in this mode. No quantization is performed, so the Quantization Tables Store is not used. The ZR36050 encodes the image samples using the JPEG one-dimensional horizontal prediction method and Huffman coding. Up to two Huffman tables are allowed in JPEG Lossless compression. The sample precision can be from 2 to 12 bits. Horizontally subsampled components are supported and no point transform is performed (the point transform parameter in the scan header marker segment is ignored).

DECODING MODES

JPEG Baseline Expansion

In Expansion mode, the ZR36050 reads compressed data and expands it using the inverse of the JPEG Baseline encoding algorithm. It stores the marker segments extracted from the compressed data in the Marker Segments Section of Internal Memory, where the host can access the information. When it encounters a DQT or DHT marker segment, it decodes the segment and stores the tables in the Table Stores. It expands any subsequent compressed image data using the stored tables. If the compressed data contains multiple instances of a marker segment, a new segment will overwrite the previous segment of the same type in the internal memory. This mode is also used to decode abbreviated format tables-only compressed data.

Fast Preview

In this mode, the ZR36050 decodes only the DC coefficients from the JPEG Baseline compressed image data, and outputs them, after level-shifting to form unsigned DC values, via the Pixel Interface. Since only one sample is output for each 8 x 8 block, the result is a thumbnail version of the image, scaled down by a factor of eight horizontally and vertically. It is generated up to 25 times faster than full image expansion. The inverse DCT computation is bypassed in this mode.

Tables Preload for Decoding

The ZR36050 reads the quantization and/or Huffman tables from Internal Memory and stores them in the Table Stores for decoding images in the JPEG abbreviated data-only format. The DQTI and DHTI bits of the MARKERS_EN register specify which tables to preload. If DQTI is set, the ZR36050 multiplies the quantization tables by the Scale Factor and stores the scaled tables in the Quantization Tables Store. If DHTI is set, it decodes the Huffman tables specifications from the DHT segment of the Internal Memory, where they are specified in accordance with the JPEG syntax, and stores the decoded tables in the Huffman Tables Store. In this mode, the Pixel Interface and the Code Buffer are inoperative.

JPEG Lossless Expansion

In Expansion mode, if the ZR36050 detects the SOF3 frame marker in the compressed data, it switches to the JPEG Lossless Expansion mode. The DCT Unit is bypassed in this mode, and the quantization tables are not used. The compressed image is decoded by the inverse of the method used in the Lossless Compression mode. As in Lossless Compression, up to two Huffman tables are allowed. The sample precision can be from 2 to 12 bits. Horizontally subsampled components are supported and no point transform is performed (the point transform parameter in the scan header marker segment is ignored. Therefore if a lossless bitstream with point transform other than zero is input into the ZR36050, it will be decoded as if the point transform is zero).

QUANTIZATION AND HUFFMAN TABLES

The tables used by the ZR36050 to encode the image data are always the ones that reside in the Quantization and Huffman Table Stores. The host does not, however, write the table data directly into these Table Stores, but in the Marker Segments Section of Internal Memory. Therefore, before it can encode or decode actual image data, the ZR36050 must first have preloaded the table data in the Table Stores.

In encoding, this can be done as part of the Statistical Pass or Compression Pass (Baseline or Lossless), if the appropriate bits of the MARKERS_EN register are set. Alternatively, it can be accomplished as a separate operation, in the Tables Preload for Encoding mode.

In decoding, if the marker segments preceding the compressed image data contain table specifications, the ZR36050 decodes these marker segments and preloads the tables in the Table Stores, and decodes the image using these tables. Otherwise, it uses the tables that pre-exist in the Table Stores, either decoded from a previous image expansion, or preloaded using the Tables Preload for Decoding operation.

Note that the same Huffman Table Store is used both in encoding and decoding. However, the internal format of the tables is different. Therefore, after a switch from encoding to decoding, or vice versa, the Huffman Table Store is invalid, and Huffman tables must be preloaded before image data can be encoded or decoded. The Quantization Table Store remains valid. In encoding, however, if the Scale Factor is changed, the Quantization Table Store is clearly invalidated and must be preloaded before the new Scale Factor can take effect.

NON-OPERATING STATES

The ZR36050 has four states in which it does not process data: Idle, Waiting, Standby, and Freeze.

The host can access the Internal Memory in the Idle and Waiting states, but not in the Standby and Freeze states.

Idle

The ZR36050 enters the Idle state after these events:

- de-activation of $\overline{\text{RESET}}$
- activation of $\overline{\text{END}}$ at the end of an encoding or decoding process
- abort of encoding, due to code volume overflow, or DCT Coefficient Buffer(s) overflow.

It remains Idle until the host issues the GO command (write to address 0 of Internal Memory), to initiate a new encoding or decoding process.

Waiting

In this state, the ZR36050 has stopped processing during encoding or decoding, and is waiting for a response from the host or system control circuitry in order to continue. It enters a Waiting state in one of these conditions:

- $\overline{\text{INT}}$ is activated, i.e., a Status Register bit is set and the corresponding Interrupt Request Register bit was enabled. The ZR36050 continues processing after the host issues the GO command.
- During encoding, if the $\overline{\text{DSYNC}}$ or $\overline{\text{EOS}}$ input signal is not active together with the last sample of a block being input, the ZR36050 finishes processing the current block and enters a Waiting state. It resumes normal operation when $\overline{\text{DSYNC}}$ is next activated.

Standby

When the device is in the Idle state, it can be switched to the Standby state to conserve power, by activation of $\overline{\text{STDBY}}$. In Standby, if CLKEN is inactive, power consumption is further reduced. After $\overline{\text{STDBY}}$ is deactivated, the host must wait at least 4 CLK_IN cycles before the first access to Internal Memory. After reactivation of CLKEN, a recovery time of 5000 CLK_IN cycles is required before any access is allowed. If $\overline{\text{RESET}}$ is activated during Standby, the device will draw normal power as long as it remains active.

Freeze

If the $\overline{\text{FREEZE}}$ input is activated, the ZR36050 freezes its operation by disabling the internal processing clock. It returns to normal operation on the next CLK_IN after $\overline{\text{FREEZE}}$ is deactivated.

ENCODING OPERATION

Figure 4 is an overview of the event sequence for typical operation of one of the encoding modes. The example shown is a Compression Pass, with compressed data output in Master mode via the Compressed Data Interface. For simplicity, the example contains one scan, and the interrupt request is assumed to be enabled only for the END status condition.

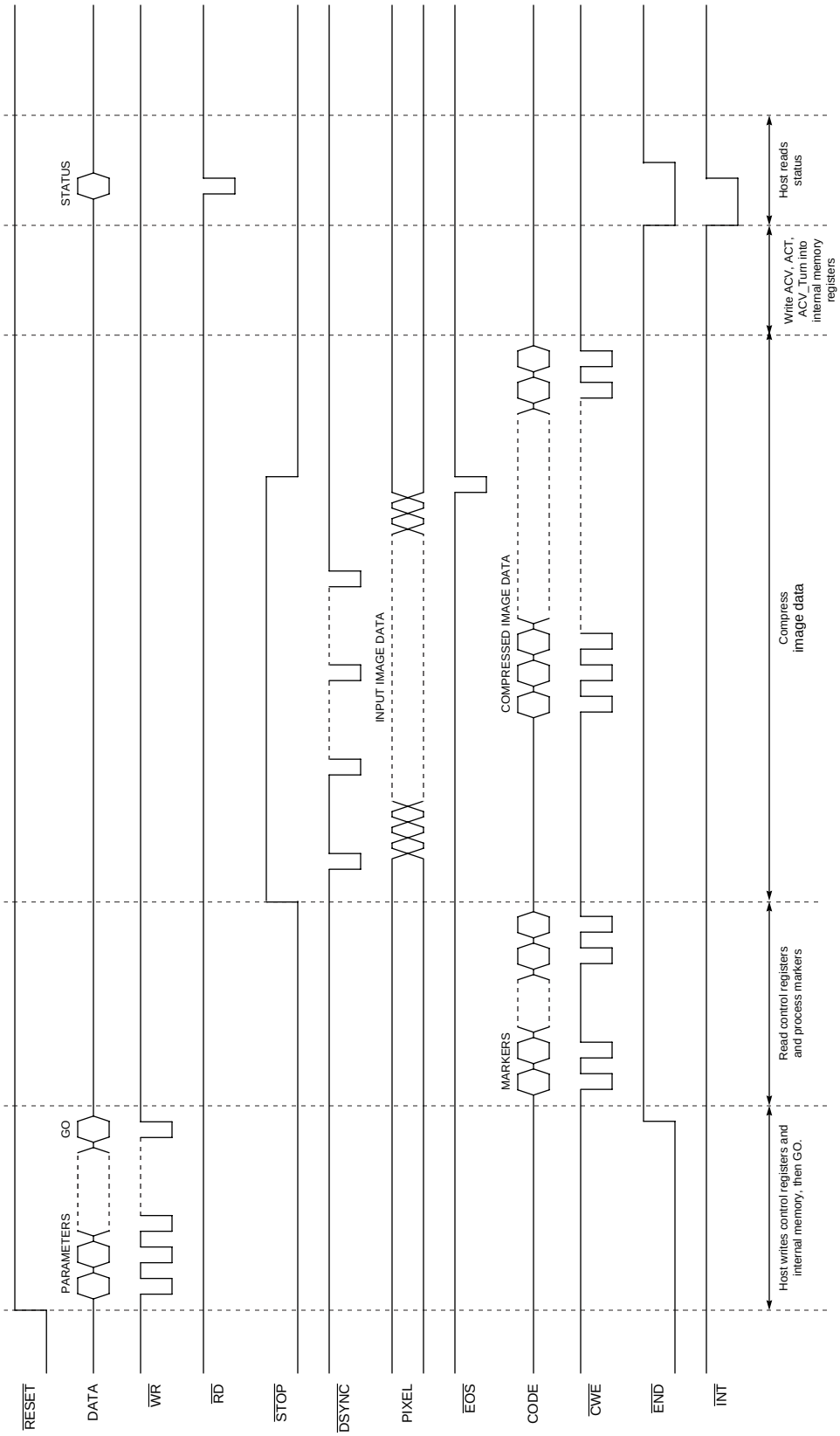


Figure 4. Event diagram for a typical encoding sequence: Compression Pass, one scan, compressed data output in master mode, interrupt request set only for END

Any encoding mode operation starts in the same way, regardless of the specific operating mode selected:

- Initially, after $\overline{\text{RESET}}$ is deactivated, the ZR36050 is in the Idle state, awaiting the GO command, with the $\overline{\text{END}}$ and $\overline{\text{STOP}}$ signal pins active. Alternatively, it could be in the Idle state as a result of the completion of a previous operation.
- Before the host starts the ZR36050 by issuing a GO command, the Control Registers must be initialized to their appropriate states, and the Marker Segments Section of Internal Memory must contain the JPEG marker segments needed for the desired operation. If the registers and marker segments are correctly configured from a previous operation, no action is required. Otherwise, the host can write the initialization data in the registers and marker segments at any time while the device is in the Idle state.
- After the host issues the GO command, the ZR36050 clears the END status bit, reads the Control Registers and initializes itself for the selected operating mode and interface configuration.

The continuation of the operating sequence from this point depends on the operating mode selected.

When a status (STATUS_0 or STATUS_1) register bit becomes active, the ZR36050 normally continues processing unless the corresponding interrupt request enable (INT_REQ_0 or INT_REQ_1) register bit was also set by the host. If it was set, the ZR36050 activates the $\overline{\text{INT}}$ output signal, stops processing and waits for the host to intervene. $\overline{\text{INT}}$ is deactivated when the host reads the relevant status register, and the ZR36050 continues processing after the host issues a GO command. If the TCVOVF or DATOVF status bits are set, the ZR36050 aborts processing and goes into the Idle state. The TCVOVF and DATOVF bits in the INT_REQ_1 register must be set to “1” at the beginning of an encoding operation.

Compression Pass

- The ZR36050 processes the SOF marker segment and the optional marker segments selected by the MARKERS_EN register, and transfers them to the compressed data after the SOI marker.

  If the frame marker is SOF3, the ZR36050 switches to Lossless encoding operation.
- Before processing the SOS marker, the ZR36050 sets the RFM status bit. If the corresponding interrupt request is enabled, the ZR36050 activates $\overline{\text{INT}}$ and stops processing. After receiving the GO command, it re-reads the INT_REQ_(1,2) and MARKERS_EN registers and transfers the specified marker segments to the compressed data. If more that one scan is required, as specified in the OPTIONS register, the ZR36050 sets RFM as above before processing each SOS marker, and re-reads MARKERS_EN after receiving the GO command. The host can use this mechanism to insert optional marker segments before each scan, if required. Also, since the Internal Memory has space for only four SOS marker segments, the same mechanism is used if more than four scans are required. In this case, when the ZR36050 stops before the fifth scan, the host can update the scan marker segments in the internal memory before continuing.

  If the MARKERS_EN register specifies an APP (or COM) segment, and the desired segment length is more that 64 bytes, it can be transferred in 64 byte sections if the APP (or COM) interrupt request bit is enabled in the INT_REQ_0 register. When initializing the Internal Memory, the host writes the first 64 bytes of the segment. Each time the Encoding Unit has completed transfer of a 64 byte section to the Code Buffer, the ZR36050 sets the APP (or COM) status bit, activates $\overline{\text{INT}}$, and waits for the host to write the next section in the Internal Memory. After transferring the last section of 64 or fewer bytes, the ZR36050 continues processing without setting the status bit or activating the interrupt.
- When the ZR36050 processes the SOS marker for the first scan, it calculates the number of blocks (or samples for Lossless compression) of each component of the scan in the MCU, and transfers the scan marker segment to the Code Buffer. When it is ready to accept image data, it sets the RFD status bit, and deactivates $\overline{\text{STOP}}$. The system can then start supplying image data for the first scan.
- The ZR36050 processes the image data and outputs the compressed data. In JPEG Baseline compression, the DCT Unit computes the DCT of each block, and the Encoding Unit quantizes the transformed samples and Huffman codes the quantized samples. In Lossless compression, the Encoding Unit calculates the prediction for each sample, and Huffman codes the predicted value. It writes the compressed data in the Code Buffer, from where it is sent to the compressed data interface or the host interface, as specified in the HARDWARE control register.

  In a Compression Pass with Bit Rate Control, the ZR36050 also computes the activity measure of each block before encoding it, and allocates a variable number of compressed data bits to the block, based on the Allocation Factor but not exceeding the number specified in the MBCV register. If the encoded block requires more bits than the allocated number, the ZR36050 truncates the compressed data for the block in a manner that preserves JPEG Baseline compatibility.

  If the OVF bit of the OPTIONS register is set, and the accumulated code volume exceeds the limit specified by the TCV_NET register at any time during the encoding, the Encoding Unit stops writing compressed data into the Code Buffer, and the ZR36050 aborts the Compression Pass completely, sets the TCVOVF status bit, and goes into the Idle state.
- When the system activates the $\overline{\text{EOS}}$ input, indicating the last data sample of the scan, the ZR36050 activates $\overline{\text{STOP}}$.
- At the end of the first scan only, if the DNL bit of the MARKERS_EN register is set, the Encoding Unit transfers the DNL marker segment from Internal Memory to the Code Buffer. DNL is disregarded in all other scans.
- If more that one scan is required, as indicated by the OPTIONS register, the ZR36050 sets the RFM status bit and compresses the second scan.
- After completing the last scan, the ZR36050 calculates a New Scale Factor (NSF) and writes the NSF, Accumulated Code Volume, Truncated Accumulated Code Volume (if Bit Rate Control is used), and Total Activity in the respective

registers, appends an EOI marker to the compressed data, sets the END status bit, activates $\overline{\text{END}}$, and goes into the Idle state.

Statistical Pass

Operation of the Statistical Pass is similar to a Compression Pass, except that the ZR36050 does not output compressed data. If they are set, the APP, COM, DRI, DQT, DHT, and DNL bits of the MARKERS_EN register are ignored in a Statistical Pass. The OVF bit of the OPTIONS register is also ignored. A Statistical Pass proceeds as follows:

- The ZR36050 processes the SOF marker segment, and sets the RFM status bit.
- The ZR36050 processes the SOS marker for the first scan, sets the RFD status bit, and deactivates $\overline{\text{STOP}}$. The system can then start supplying image data for the first scan.
- The ZR36050 processes the image data, until the $\overline{\text{EOS}}$ input indicates the last data sample of the scan, and activates $\overline{\text{STOP}}$.
- If more that one scan is required, as indicated by the OPTIONS register, the ZR36050 sets the RFM status bit and processes the second scan.
- After the last scan, the ZR36050 computes the new Scale Factor and the Allocation Factor.
- At the end of a stand-alone Statistical Pass, the ZR36050 writes the new Scale Factor, the Allocation Factor, the Accumulated Code Volume, and the Total Activity in the respective registers, sets the END status bit, activates $\overline{\text{END}}$, and goes into the Idle state.

  If the Statistical Pass is part of an Auto Bit Rate Control operation, the ZR36050 continues immediately with the Compression Pass, using the new Scale Factor, and writes the ACV_NET, ACT, and ACV_TRUN into their appropriate registers in the Internal Memory at the end of the Compression Pass.

Tables-Only Pass

- The ZR36050 processes the SOF marker segment and any of the DQT, DHT, APP, and COM optional marker segments that are selected by the MARKERS_EN register. The ZR36050 transfers the selected optional marker segments to the compressed data after an SOI marker.

  If an APP or COM segment is longer than 64 bytes, it can be transferred by the means described for the Compression Pass.
- The ZR36050 appends an EOI marker to the compressed data, sets the END status bit, activates $\overline{\text{END}}$, and goes into the Idle state.

Tables Preload for Encoding

- The ZR36050 preloads the tables specified by the DQTI and DHTI bits of the MARKERS_EN register. If DQTI is set, the ZR36050 reads the quantization tables specified in the DQT marker segment of the Internal Memory, multiplies them by the Scale Factor, and stores them in the Quantization Table Store. If DHTI is set, the ZR36050 decodes the Huffman tables specified in the DHT segment of the Internal Memory, and stores the decoded tables in the Huffman Table Store.
- It sets the END status bit and activates the $\overline{\text{END}}$ signal, and goes into the Idle state.

DECODING OPERATION

Figure 5 is an overview of the event sequence for typical decoding operation. The example shown is for full Expansion, with compressed data input in DMA mode. For simplicity, a single scan is shown and it is assumed that no interrupt request is enabled.

A decoding operation starts as follows:

- Initially, after $\overline{\text{RESET}}$ is deactivated, the ZR36050 is in the Idle state, awaiting the GO command, with the $\overline{\text{END}}$ and $\overline{\text{STOP}}$ signal pins active. Alternatively, it could be in the Idle state as a result of the completion of a previous operation.
- Before the host starts the ZR36050 by issuing a GO command, the Control Registers must be initialized to their appropriate states. If the registers are correctly configured from a previous decoding operation, no action is required. Otherwise, the host can write the initialization data in the registers at any time while the device is in the Idle state.
- After the host issues the GO command, the ZR36050 clears the END status bit, reads the Control Registers and initializes itself for the selected operating mode and interface configuration.

The continuation of the operating sequence from this point depends on the operating mode selected.

When a status register bit becomes active, the ZR36050 continues processing unless the corresponding interrupt request register bit was also set by the host. If it was set, the ZR36050 activates the $\overline{\text{INT}}$ output signal, stops processing and waits for the host to intervene. $\overline{\text{INT}}$ is deactivated when the host reads the status register, and the ZR36050 continues processing after the host issues a GO command.

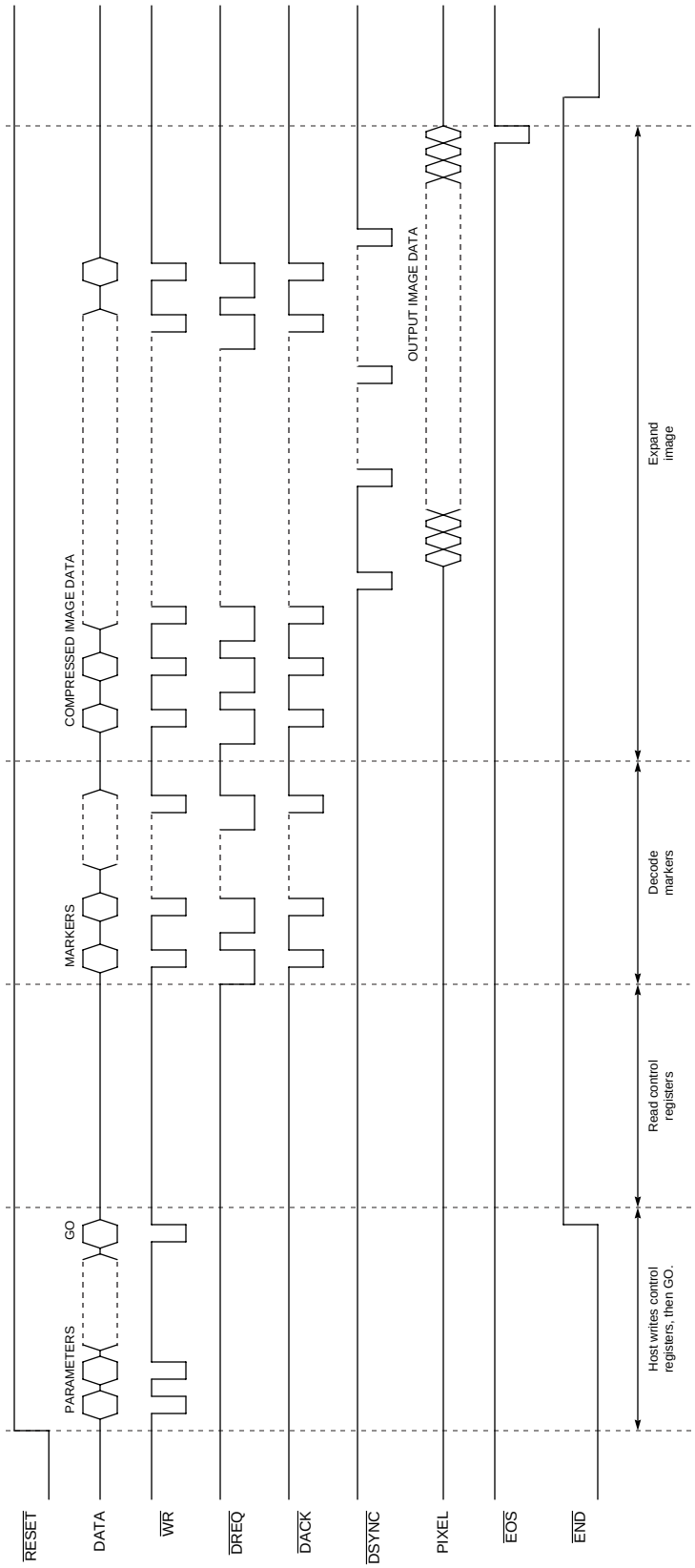


Figure 5. Event diagram for a typical decoding sequence: one scan, compressed data input in DMA mode, no interrupt requests set

Expansion (Full or Fast Preview)

- The ZR36050 reads compressed data from the Compressed Data Interface or the Host Interface, as selected by the HARDWARE register, into the Code Buffer.

  Whenever it encounters a marker segment, it writes the segment in the appropriate location in Internal Memory, decodes the information contained in the segment and sets the appropriate marker bit in the STATUS_0 register. If the corresponding interrupt request is enabled, the ZR36050 stops processing the compressed data, activates $\overline{INT}$, and waits for the host to restart processing by issuing the GO command.

  If an APP (or COM) marker is encountered, and the length of the segment is greater than 64 bytes, the host can read it out by a method similar to that used in encoding, if the APP (or COM) interrupt request is enabled. The ZR36050 writes 64 bytes at a time (or fewer in the last section) in the internal memory, sets the APP (or COM) status bit, and waits for the host to intervene and read the data. If the interrupt request bit was not set, the ZR36050 writes the first 64 bytes of the segment into internal memory and continues processing, discarding the remainder of the segment.
- If the frame marker is SOF3, the ZR36050 switches to JPEG Lossless decoding.
- The Zr36050 decodes the SOS marker segment, and determines the number of blocks (JPEG Baseline) or samples (JPEG Lossless) of each component of the scan (in the MCU). The ZR36050 then sets the RFD status bit when it has completed processing the markers and the Pixel Interface is ready to send out the first image samples.
- The ZR36050 decodes the first scan of the compressed image and outputs the expanded image samples via the Pixel Interface. It outputs an $\overline{EOS}$ signal together with the last sample of the scan, and sets the RFM status bit at the same time.
- If there is more than one scan, the ZR36050 repeats the scan decoding until all scans have been processed.
- The ZR36050 sets the END status bit, activates the $\overline{END}$ signal, and goes to the Idle state.

Note: At the end of a decoding process, the ZR36050 performs up to two extra read cycles. Therefore the system must allow the ZR36050 to read up to two Compressed Data Memory bytes after the EOI marker. These bytes may be any value, including "FF".

Tables Preload for Decoding

- The ZR36050 preloads the tables specified by the DQTI and DHTI bits of the MARKERS_EN register. If DQTI is set, the ZR36050 reads the quantization tables specified in the DQT marker segment of the Internal Memory, multiples them by the Scale Factor, and stores them in the Quantization Table Store. If DHTI is set, the ZR36050 decodes the Huffman tables specified in the DHT segment of the Internal Memory, and stores the decoded tables in the Huffman Table Store.
- It sets the END status bit and activates the $\overline{END}$ signal, and goes into the Idle state.

FREEZE OPERATION

The $\overline{FREEZE}$ signal freezes all ZR36050 operations. It is sampled on the rising edge of CLK_IN. Once $\overline{FREEZE}$ is sampled, all buses float immediately, and the activities of the ZR36050 are frozen in their current state. The output control signals that are activated on the same rising edge of CLK_IN that samples $\overline{FREEZE}$ will be activated but frozen in their new states.

However, the activation of the control signals that become active with 0.5 DCLK delay with respect to rising edge of CLK_IN (for example, CRD, CWR) will be delayed until $\overline{FREEZE}$ is deactivated. Figures 6 and 7 show an example of Freeze and Normal operation for compressed Data memory write cycle with CFIS=01 (2 CLK_IN cycles).

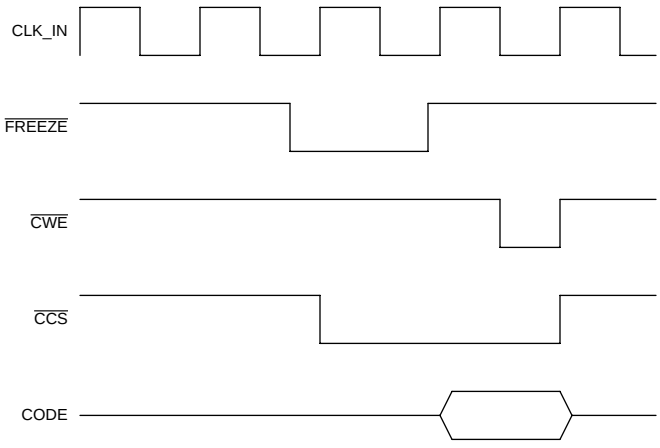


Figure 6. Freeze Operation for Compressed Data Memory Write Cycle with CFIS=01

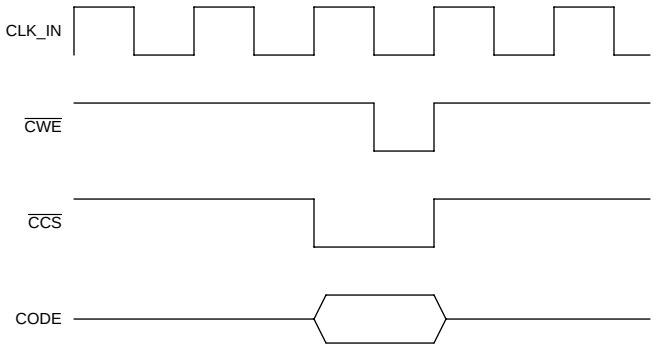


Figure 7. Normal Operation for Compressed Data Memory Write Cycle with CFIS=01

INTERFACES

In the diagrams illustrating operation of the interfaces, an arrow linking signal transitions indicates the causal relationship of the transitions. An arrow terminating on a circle designates the point at which the indicated signal is sampled.

Host Interface

The Host Interface is used to access the Internal Memory (Control Registers and JPEG Marker Segments) of the ZR36050. It can also optionally be used to transfer the compressed data to and from the ZR36050. There are 4 categories of host interface bus cycle:

- Internal Memory read and write by the host
- Interrupt acknowledgment
- Slave mode Compressed Data Transfer
- DMA mode Compressed Data Transfer

Note that, although the Host Interface behaves as an asynchronous interface, its control signals are internally synchronized to CLK_IN, and CLK_IN must be toggling and enabled by CLKEN in order for the Host Interface to operate.

Internal Memory Read and Write

Internal Memory read and write by the host is always 8 bits wide, using the DATA, ADDR, $\overline{CS}$, $\overline{RD}$, and $\overline{WR}$ signal pins.

Figure 8 and Figure 9 show read and write cycles, respectively. A bus cycle starts when the host drives $\overline{CS}$ low, and ends when $\overline{CS}$ goes high. ADDR and $\overline{CS}$ must be stable throughout the cycle and must overlap the $\overline{RD}$ or $\overline{WR}$ pulse. ADDR is sampled with the falling edge of $\overline{RD}$ or $\overline{WR}$.

In a read cycle, the Host Interface drives the DATA bus when $\overline{CS}$ and $\overline{RD}$ are both active.

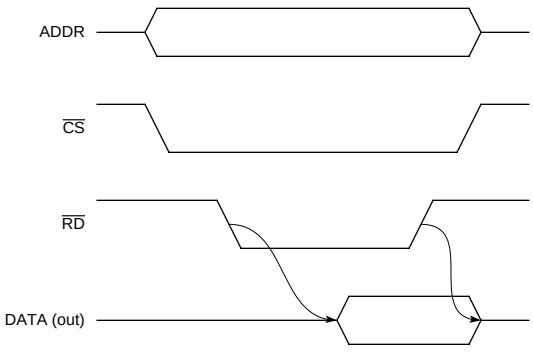


Figure 8. Parameter Read by Host

In a write cycle the Host Interface latches the data on the rising edge of $\overline{WR}$. Data must be valid before the trailing edge of $\overline{WR}$; it need not be valid before the leading edge. Note that the GO command is a write of arbitrary data to address 0.

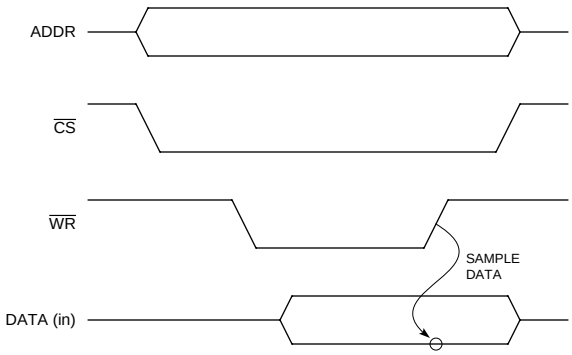


Figure 9. Parameter Write by Host

Interrupt Acknowledgment Cycle

The Host Interface activates $\overline{INT}$ when one of the STATUS_0 or STATUS_1 register bits (with the exception of the DATRDY bit) becomes active, and the corresponding bit in INT_REQ_0 or INT_REQ_1 is set. When the host reads the status register containing the active bit, or gives the GO command, the status register is cleared and the Host Interface deactivates $\overline{INT}$.

Figure 10 shows activation of $\overline{INT}$, and its deactivation in response to a status read.

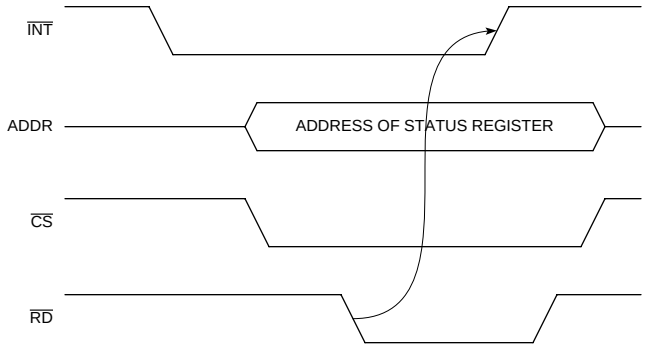


Figure 10. Interrupt Acknowledgment by Read of the Status Register

Slave Mode Compressed Data Transfer

In Slave mode, the host reads the compressed data from the memory-mapped Compressed Data register in encoding, or writes it into this register in decoding. Compressed data transfers can be 8 or 16 bits wide, as specified by the BSWD bit in the HARDWARE register. In a 16 bit transfer, the CODE bus acts as an extension of the DATA bus. The lower numbered byte (the byte that would have been transferred earlier in 8 bit transfers) is read or written on the CODE or DATA bus, depending on the BELE bit in the HARDWARE register. In 16-bit transfers, if the last transfer of compressed data requires only one byte, the ZR36050 appends a FF byte to complete the 16 bits in encoding, and in decoding, the host can append an arbitrary byte.

In encoding, the ZR36050 indicates that 8 or 16 bits of compressed data are ready to be read out, by setting the DATRDY bit in the STATUS_1 register. If the corresponding bit in INT_REQ_1 is set, the Host Interface activates $\overline{DINT}$. When the host reads the Compressed Data register, the ZR36050 clears DATRDY and deactivates $\overline{DINT}$, as shown in Figure 11.

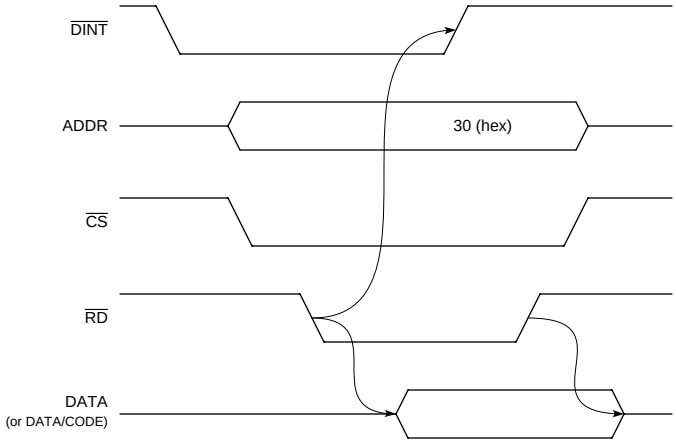


Figure 11. Slave Read of Compressed Data in Encoding

In decoding, active $\overline{DINT}$ indicates that the ZR36050 is ready to receive the next 8 or 16 bits of compressed data. When the host writes the data in the Compressed Data register, the ZR36050 clears DATRDY and deactivates $\overline{DINT}$, as shown in Figure 12.

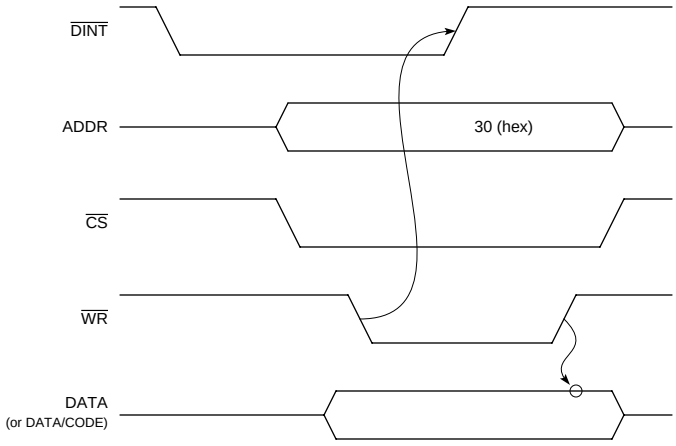


Figure 12. Slave Write of Compressed Data in Decoding

Note that if the DATRDY bit in the INT_REQ_1 register is not set, the Host Interface does not activate $\overline{DINT}$. The host can, however, still determine whether new compressed data is ready or needed, by polling the STATUS_1 register to determine the state of DATRDY.

DMA Compressed Data Transfer

In DMA mode, the Host Interface requests a DMA Compressed Data Transfer by activating $\overline{DREQ}$, and the host system's DMA controller acknowledges the transfer with $\overline{DACK}$. In encoding, the DMA controller reads the data by activating $\overline{RD}$. In decoding, the Host Interface latches the data on the trailing edge of $\overline{WR}$. The ZR36050 does not output a $\overline{DREQ}$ until the $\overline{DACK}$ signal to the previous $\overline{DREQ}$ has been deactivated. In both cases $\overline{DREQ}$ is deactivated by $\overline{RESET}$, or by $\overline{DACK}$.

In the encoding modes, the ZR36050 outputs $\overline{END}$ only after the EOI marker (FFD9) has been read and its corresponding $\overline{DACK}$ signal has been deactivated.

The transfers can be 8 or 16 bits wide, as specified by the BSWD bit in the HARDWARE register. In 16 bit transfers, the CODE bus acts as an extension of the DATA bus. The lower numbered byte (the byte that would have been transferred earlier in 8 bit transfers) is read or written on either the CODE or DATA bus, depending on the BELE bit in the HARDWARE register. In 16-bit transfers, if the last transfer of compressed data requires only one byte, the ZR36050 appends a FF byte to complete the 16 bits in encoding, and in decoding, the host can append an arbitrary byte.

Figure 13 and Figure 14, respectively, show DMA Compressed Data Transfer cycles in encoding and decoding.

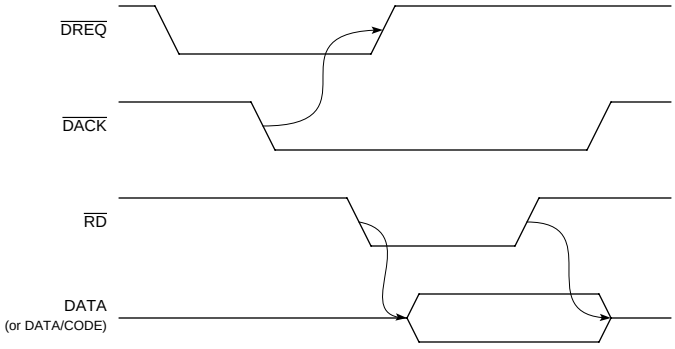


Figure 13. DMA Read of Compressed Data in Encoding

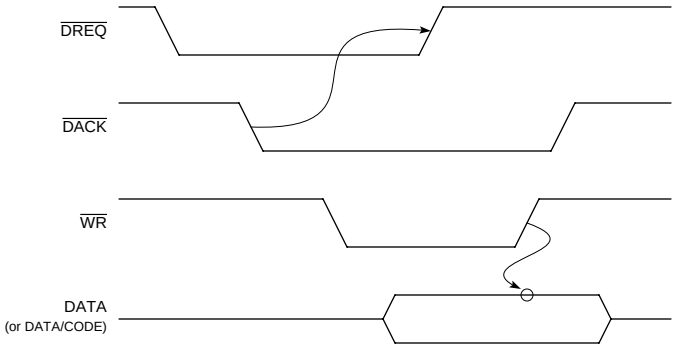


Figure 14. DMA Write of Compressed Data in Decoding

Compressed Data Interface

When the MSTR bit of the HARDWARE register is set, the ZR36050 transfers compressed data using the Compressed Data Interface, directly accessing the external compressed data memory in a Master mode. Data transfers are 8 bits wide, using the CODE bus and the control signals $\overline{CCS}$, $\overline{COE}$, $\overline{CWE}$, and $\overline{CAEN}$.

The $\overline{CBUSY}$ input can be used by the external compressed data memory control logic, to suppress read or write access by the Compressed Data Interface when the memory is temporarily unavailable. $\overline{CBUSY}$ is sampled one CLK_IN prior to the beginning of a bus cycle, and its state determines whether the bus cycle is executed or suppressed.

The Compressed Data Interface does not provide the address signals to the compressed data memory. The memory must appear to the interface as a FIFO. $\overline{CAEN}$, which is activated on each access, can be used by the memory control logic to enable an address counter.

The cycle time of master mode transfers can be from one to eight CLK_IN cycles, and is specified by the CFIS field of the HARDWARE register. A new bus cycle can begin immediately after the end of the previous cycle.

A bus cycle starts when the Compressed Data Interface activates $\overline{CCS}$. $\overline{CCS}$ is stable throughout the bus cycle, and $\overline{CAEN}$ is active during the last CLK_IN cycle of the bus cycle. In back-to-back bus cycles, $\overline{CCS}$ is active continuously.

In a read cycle, executed during decoding, $\overline{COE}$ goes active 0.5 x CLK_IN after the beginning of the cycle, and remains active until the end of the cycle. The memory must drive data on the CODE bus while $\overline{COE}$ is active. The Compressed Data Interface latches the data on the trailing edge of $\overline{COE}$.

In a write cycle, executed during encoding, $\overline{CWE}$ goes active 0.5 x CLK_IN after the beginning of the cycle, and remains active until the end of the cycle. The Compressed Data Interface drives CODE with valid data for the duration of the bus cycle.

When the Compressed Data Interface is inactive, the CODE bus floats, and is pulled low by the internal pull-down devices. The interface control signals are driven high.

Figure 15 and Figure 16 show examples of read and write cycles, with CFIS = 001 (2 CLK_IN cycles per bus cycle). In these examples, $\overline{CBUSY}$ is sampled first inactive, to enable a bus cycle, then active to suppress the start of the next bus cycle. The dashed lines depict the start of a bus cycle that could have been executed if $\overline{CBUSY}$ had been sampled active.

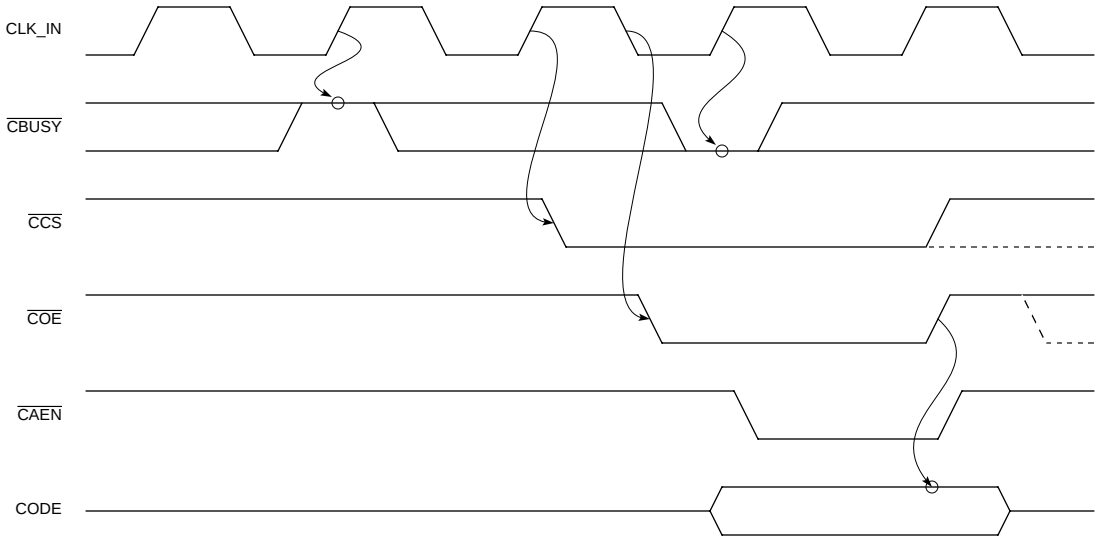


Figure 15. Master Mode Compressed Data Read in Decoding, Shown for Bus Cycle Time of Two CLK_IN

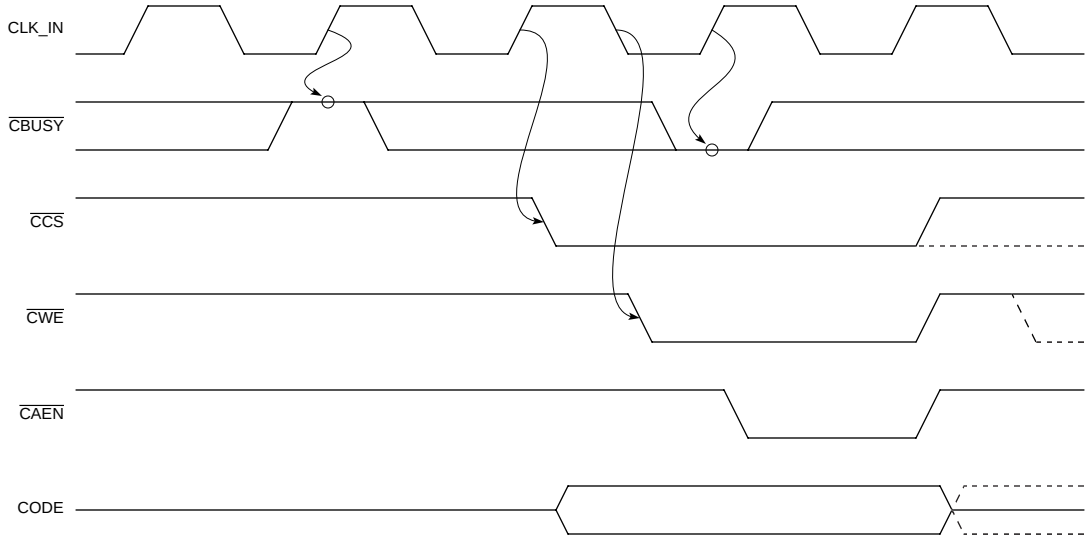


Figure 16. Master Mode Compressed Data Write in Encoding, Shown for Bus Cycle Time of Two CLK_IN

Pixel Interface

The pixel interface operates synchronously in all modes of operation, transferring an image data sample in or out every CLK_IN, when enabled to do so. The DSYNC and STOP signals control the flow of data on the PIXEL bus, DSYNC indicating which clock cycles contain valid data, and STOP acting as a request to halt data transfer for an arbitrary time interval. An additional signal, EOS, indicates the end of a scan. In encoding modes, PIXEL, DSYNC and EOS are inputs and STOP is an output, whereas in decoding modes, PIXEL, DSYNC and EOS are outputs and STOP is an input.

In the JPEG Baseline encoding and decoding modes, image data is transferred on the PIXEL bus a block at a time, and DSYNC indicates the beginning of a block. DSYNC is active for one CLK_IN cycle, preceding the first sample of a block, and is always followed by a burst of 64 image samples.

In the Fast Preview decoding mode, and in JPEG Lossless encoding and decoding, where image data is transferred in a line by line raster, DSYNC is active in the CLK_IN cycle preceding each sample.

JPEG Baseline Encoding

In JPEG Baseline encoding, the Pixel Interface samples the most significant eight bits of the PIXEL bus.

Figure 17 shows two consecutive blocks of image samples being input at the beginning of a scan, or after a break in data transfer. Note that the figure shows the second block immediately following the first, however, this is not necessarily the case. The first sample of the second block could be separated from the last sample of the first block by any number of clocks, by delaying the activation of DSYNC.

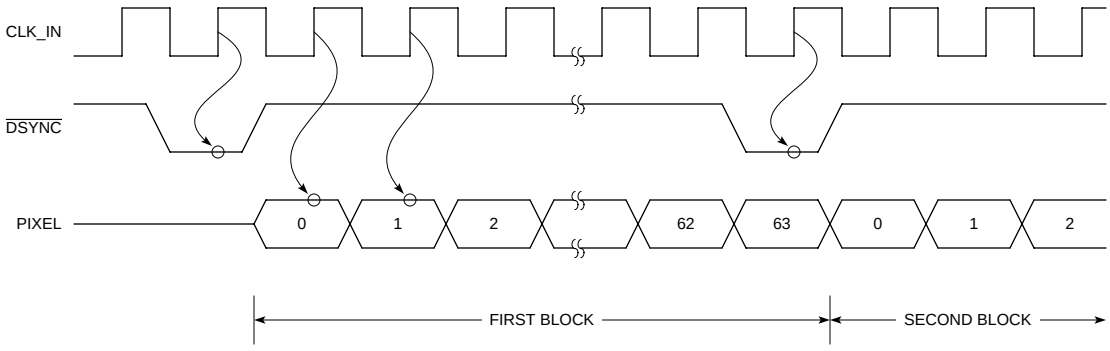


Figure 17. Beginning of a Scan in JPEG Baseline Encoding

Figure 18 shows activation of $\overline{STOP}$ by the Pixel Interface, and the required response of the external system logic in halting the flow of samples. The Pixel Interface activates $\overline{STOP}$ when three of the four Coefficient Buffers are full, so the system must respond by halting after the last sample of the current block.

The system must activate $\overline{EOS}$ together with the last image sample of a scan, as shown in Figure 19.

JPEG Baseline Decoding

In JPEG Baseline decoding, the Pixel Interface places the data samples on the most significant eight bits of the PIXEL bus and zeros the other four bits.

Figure 20 and Figure 21 show the beginning and end of a scan, respectively, in JPEG Baseline decoding.

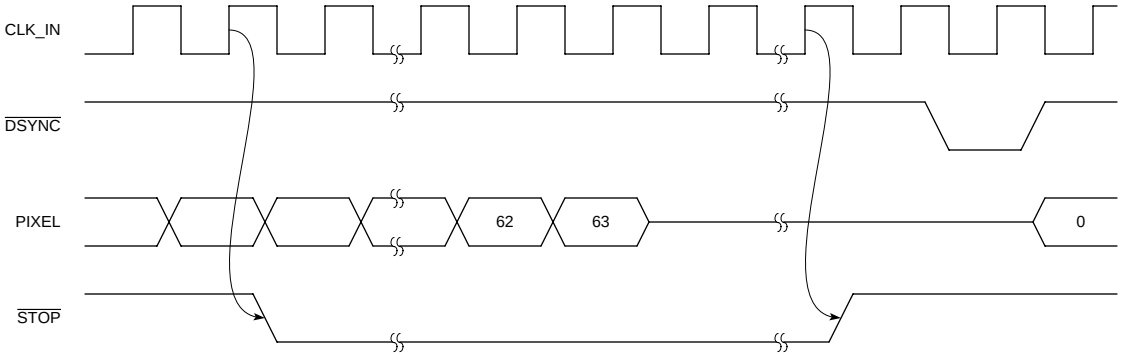


Figure 18. System Response to $\overline{STOP}$ in JPEG Baseline Encoding

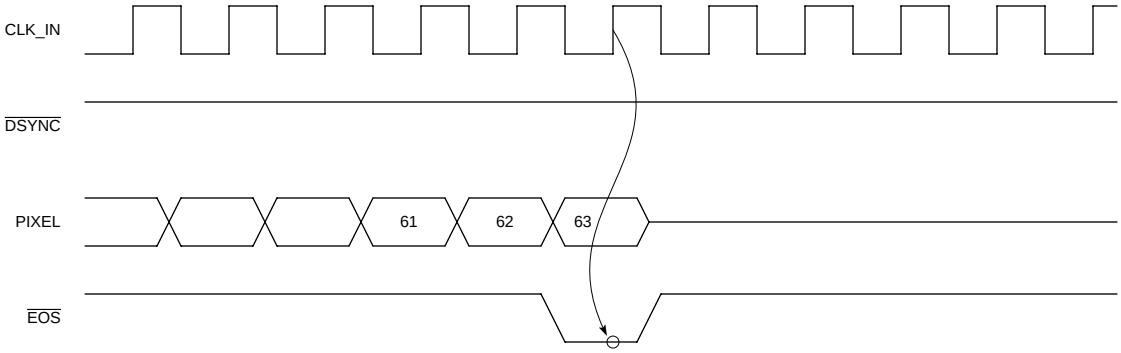


Figure 19. Activation of $\overline{EOS}$ at end of Scan in JPEG Baseline Encoding

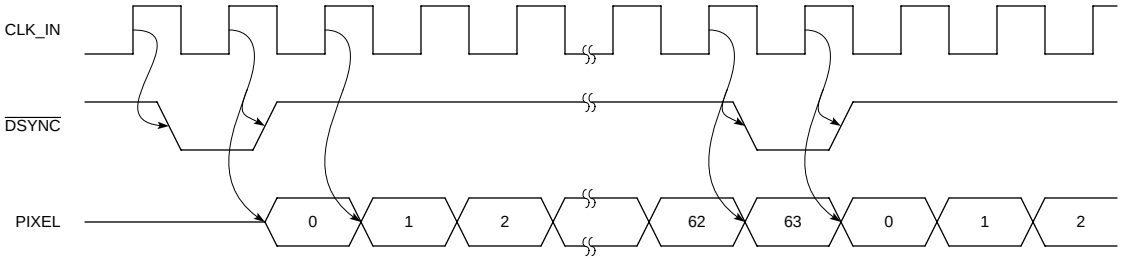


Figure 20. Beginning of a Scan in JPEG Baseline Decoding

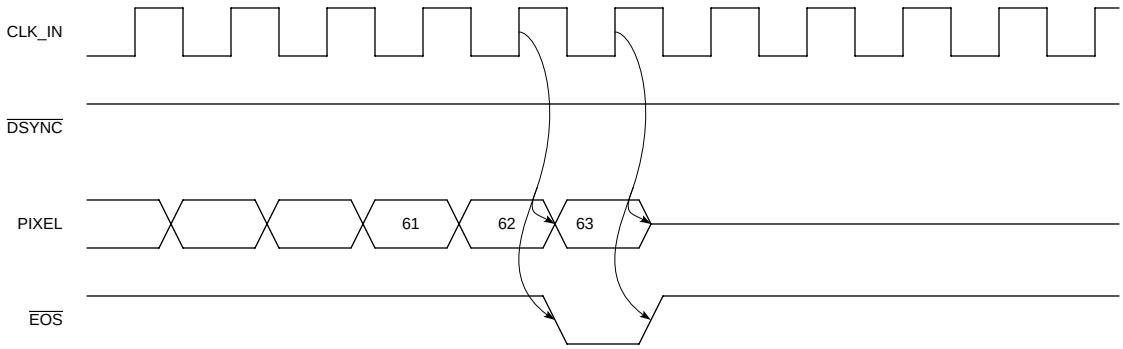


Figure 21. Activation of $\overline{EOS}$ at end of Scan in JPEG Baseline Decoding

In the example shown in Figure 20, the second block follows the first with no break. This, however is not necessarily so; the first sample of the second block could be separated from the last sample of the first block by any number of clocks. For further discussion of this, see the DATA FLOW CONTROL section.

Note that if the system logic activates $\overline{STOP}$ at least 24 clock cycles before the last data sample of the current block, the Pixel Interface halts the flow of data at the end of the current block. Otherwise, it halts the flow at the end of the next block.

Fast Preview Decoding

Figure 22 shows operation of the Pixel Interface in Fast Preview decoding. It includes the $\overline{EOS}$ output indicating the end of a scan. Note that in this mode, the Pixel Interface activates $\overline{EOS}$ within 64 CLK_IN cycles after the last sample of the scan appears on the PIXEL bus.

For each decoded block, the Pixel Interface outputs one unsigned sample representing the DC coefficient of the block, on the most significant eleven bits of the PIXEL bus, and forces the least significant bit to zero. A sample can follow the previous sample immediately, on the next clock, or the consecutive samples can be separated by any number of clock cycles. In the example shown, the last sample is separated from the previous one by one clock cycle.

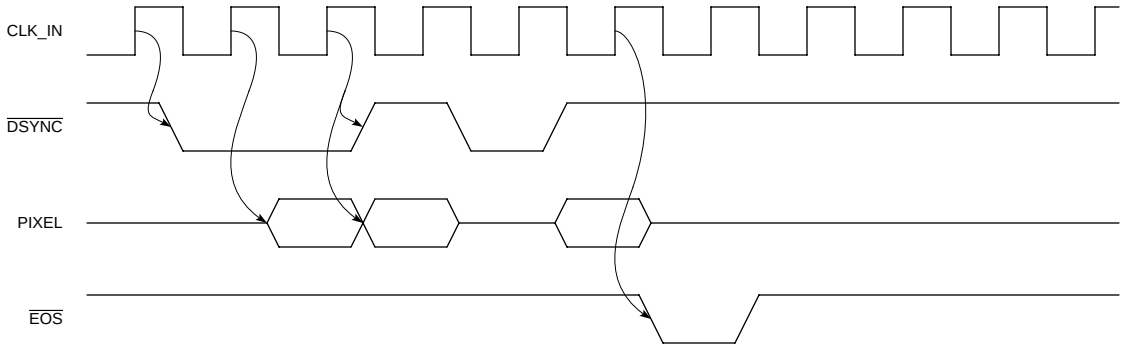


Figure 22. Fast Preview Decoding, Showing $\overline{EOS}$ at end of Scan

JPEG Lossless Encoding and Decoding

Figure 23 and Figure 24 show operation of the Pixel Interface in JPEG Lossless encoding and decoding. The number of PIXEL bus bits used depends on P, the sample precision parameter in the JPEG frame header. If P specifies fewer than 12 bits, the most significant P bits of the PIXEL bus are used. In decoding, the Pixel Interface forces the unused bits to zero.

In encoding or decoding, a sample can follow the previous sample immediately on the next clock, or the consecutive samples can be separated by any number of clock cycles.

In encoding, the system logic must activate $\overline{EOS}$ together with the last sample of the scan. In decoding, the Pixel Interface activates $\overline{EOS}$ within 64 CLK_IN cycles after the last sample of the scan appears on the PIXEL bus.

If the Pixel Interface activates $\overline{STOP}$ during JPEG Lossless encoding, the system logic must deactivate $\overline{DSYNC}$ to halt the flow of data samples, within at most three CLK_IN cycles.

If the system logic activates $\overline{STOP}$ during JPEG Lossless decoding, the Pixel Interface deactivates $\overline{DSYNC}$ one CLK_IN cycle after it samples the active $\overline{STOP}$.

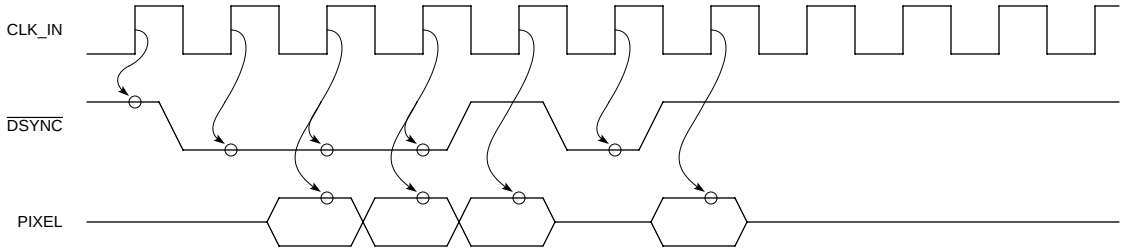


Figure 23. JPEG Lossless Encoding

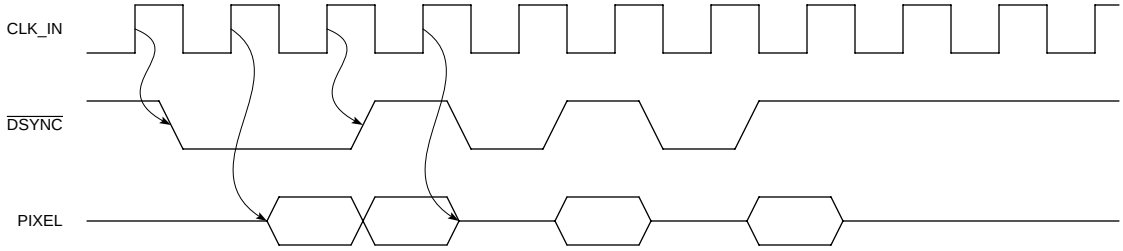


Figure 24. JPEG Lossless Decoding

DCT Coefficients Output

In JPEG Baseline encoding and decoding, the DCT coefficients are output on the auxiliary COEF bus. The $\overline{CSYNC}$ synchronizing signal indicates the beginning of a block of coefficients. The coefficients of a block are output in column order, starting with the DC coefficient.

In encoding, the COEF bus outputs the DCT coefficients (before quantization) of the block that was previously input on the PIXEL bus, with a total delay of 81 CLK_IN cycles, as shown in Figure 25. If, however, there is a break in the flow of data on the PIXEL bus causing a gap between the last data sample of block N and the first sample of block N+1, the same gap is reflected into the COEF bus with a delay of 16 clock cycles. That is to say, the gap appears between the last coefficient of the coefficient block N-1 and the first coefficient of block N.

In decoding, the COEF bus outputs the DCT coefficients (after dequantization) 80 CLK_IN cycles in advance of the next block that will be output on the PIXEL bus, as shown in Figure 26. If, however, there is a gap between the last coefficient of block N and the first coefficient of block N+1, the same gap appears 16 clocks later on the PIXEL bus, between the last data sample of block N-1 and the first sample of block N.

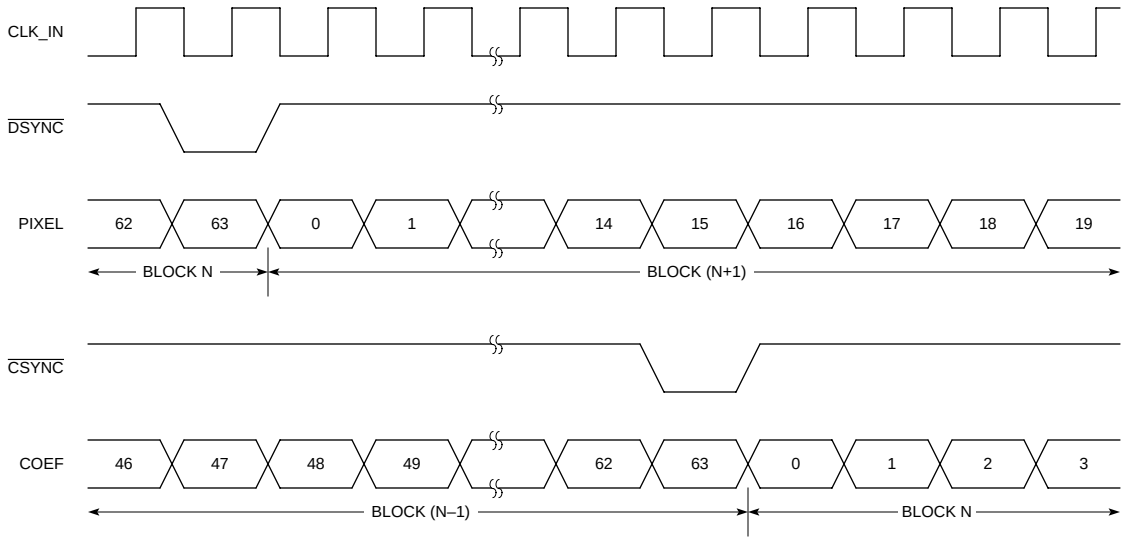


Figure 25. Coefficient Output in JPEG Baseline Encoding

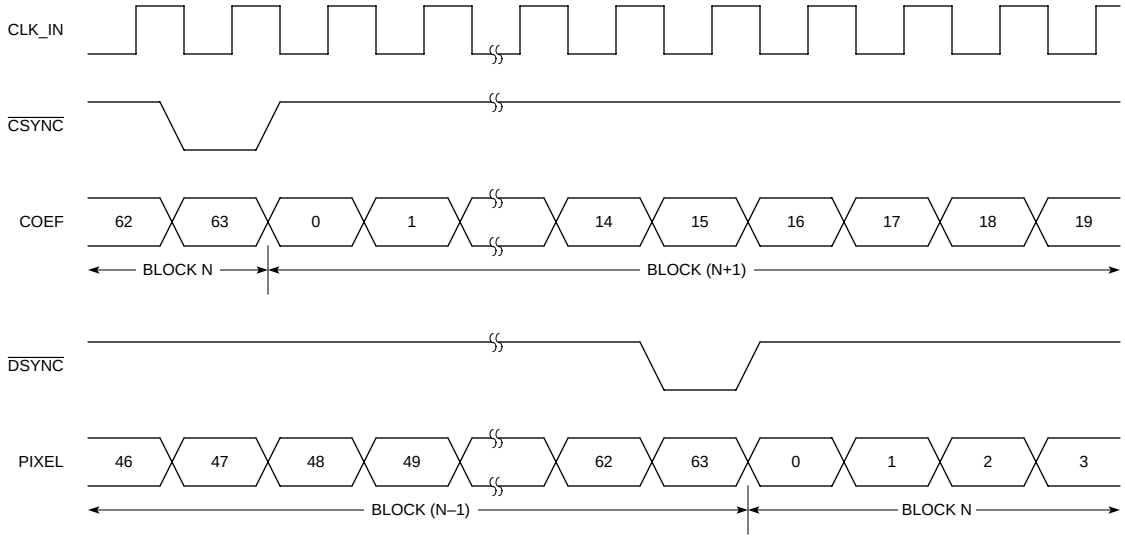


Figure 26. Coefficient Output in JPEG Baseline Decoding

DATA FLOW CONTROL

Ideally, image data and compressed data flow into or out of the ZR36050 at the maximum possible rate, without breaks. In practice, however, there are sometimes internal or external conditions that necessitate a break in the data flow.

Encoding

- **Missing $\overline{\text{DSYNC}}$**. $\overline{\text{DSYNC}}$ allows the external system, for example a strip or frame buffer controller, to affect the flow of image data into the Pixel Interface in encoding.

  If $\overline{\text{DSYNC}}$ is not activated together with the last image data sample of a block, the Pixel Interface stops sampling input data of the next block, until it samples $\overline{\text{DSYNC}}$ active again. The ZR36050 completes processing of all data that was input prior to the missing $\overline{\text{DSYNC}}$, outputs the DCT coefficients of the current block, and then enters a Waiting state. The ZR36050 resumes sampling and processing of image data upon recieving the next $\overline{\text{DSYNC}}$ activation.

  There is no lower or upper limit on the duration of the break between the missing $\overline{\text{DSYNC}}$ and the next activation of $\overline{\text{DSYNC}}$.

- **Activation of $\overline{\text{CBUSY}}$**. In Master mode Compressed Data Transfer, the external compressed data memory control logic can activate $\overline{\text{CBUSY}}$ at any time, to signal the Compressed Data Interface that the memory bus is busy.

  If the Compressed Data Interface detects $\overline{\text{CBUSY}}$ active, it completes the current write cycle (if one has started), and ceases writing compressed data. It resumes writing when $\overline{\text{CBUSY}}$ becomes inactive.

  While $\overline{\text{CBUSY}}$ is active, the Encoding Unit continues to encode the coefficients in the Coefficient Buffers until the Code Buffer is full. The Pixel Interface continues to transfer data samples to the DCT Unit until three of the four Coefficient Buffers are full, as described in the next section.

  There is no upper limit on the duration of active $\overline{\text{CBUSY}}$. The lower limit is one CLK_IN.

- **Coefficient Buffers full**. If there is a break in the transfer of compressed data during encoding, because $\overline{\text{CBUSY}}$ was activated or the Compressed Data Interface bus cycle is long in Master mode, or because of slow host response in Slave or DMA modes, the Pixel Interface continues to input image data until three of the four DCT Coefficient Buffers are full. When $\overline{\text{DSYNC}}$ of a fourth block arrives, the Pixel Interface activates $\overline{\text{STOP}}$, but continues to input the image data of the fourth block.

  If $\overline{\text{STOP}}$ becomes active, the system control logic must halt the flow of image data to the Pixel Interface at the end of the current block. If it does not, and $\overline{\text{STOP}}$ is still active, the Coefficient Buffers overflow and data is lost, the DATOVF status bit is set, and the ZR36050 aborts the encoding process and goes into the Idle state.

Decoding

- **Active $\overline{\text{STOP}}$**. In decoding, the system control logic can break the flow of data samples from the Pixel Interface by activating $\overline{\text{STOP}}$. In order to prevent the output of the next block, $\overline{\text{STOP}}$ has to be activated at least 24 CLK_IN cycles before the last image sample of the current block that is being output and must remain active at least until the end of the current block. While $\overline{\text{STOP}}$ remains active, the ZR36050 continues to decode compressed data, until the Coefficient Buffers are full.

  Upon deactivation of $\overline{\text{STOP}}$, the ZR36050 outputs the next $\overline{\text{DSYNC}}$ followed by its corresponding image data block at least 17 CLK_IN cycles after deactivation of $\overline{\text{STOP}}$.

- **Activation of $\overline{\text{CBUSY}}$**. In Master mode Compressed Data Transfer, the external compressed data memory control logic can activate $\overline{\text{CBUSY}}$ at any time, to signal the Compressed Data Interface that the memory bus is busy. The Compressed Data Interface strobes $\overline{\text{CBUSY}}$ on the rising edge of CLK_IN.

  If $\overline{\text{CBUSY}}$ is activated at least one CLK_IN prior to the beginning of a read or write cycle, then the next read or write cycle will not be performed. It resumes reading when $\overline{\text{CBUSY}}$ becomes inactive.

  There is no upper limit on the duration of active $\overline{\text{CBUSY}}$. The lower limit is one CLK_IN.

- **Slow compressed data transfer**. If Compressed Data Transfer is slowed, because $\overline{\text{CBUSY}}$ was activated or the Compressed Data Interface bus cycle is long in Master mode, or because of slow host response in Slave or DMA modes, the next Coefficient Buffer may not be fully assembled by the time it is needed for continuous image data flow.

  In this case, the Pixel Interface does not activate $\overline{\text{DSYNC}}$ together with the last output sample of the current block, and therefore does not immediately start to output the data samples of the next block. As soon as the next complete block has been decoded and is ready for output, the Pixel Interface activates $\overline{\text{DSYNC}}$ followed by the image data.

INTERNAL MEMORY FORMAT

The ZR36050 Internal Memory serves as a communication interface between the ZR36050 and the host. The host determines the operating modes and options of the ZR36050, the encoding/decoding parameter values, and the JPEG marker segments by writing the appropriate data into the ZR36050 Internal Memory. It is also used to exchange auxiliary data between the ZR36050 and the host.

The Internal Memory is divided into two parts; Control Register section, and JPEG Marker Segment section.

The Control Register section contains a virtual register for the GO command; configuration registers; the Markers Enable register; code volume registers (target and actual accumulated); scale and allocation factor registers; a Total Activity register; Status registers; and Interrupt Request enable registers.

The JPEG Marker Segment section contains the JPEG marker segments to be transferred to the compressed data bit stream during the encoding process. It is also used to save the JPEG marker segments extracted by the ZR36050 from the compressed data during the decoding process.

The Compressed Data Input/Output register, for Compressed Data Transfer in Slave mode, is also mapped in the Internal Memory space.

The Internal Memory is organized in bytes and the description of each byte is given in this section. Multiple-byte quantities are always stored together, with the most-significant byte at the lowest address, and the least-significant byte at the highest address. Reserved registers in the Internal Memory are for the ZR36050 use only and must not be accessed by the host. The values/markers that appear in brackets [ ] must be written with the specified values and are represented in hexadecimal.

All internal memory registers can be accessed by the host except the reserved registers.

The Internal Memory address and compulsory data values are specified as hexadecimal numbers. But note that quantization and Huffman table indices are in decimal representation.

CONTROL REGISTERS DESCRIPTION

GO Register

GO Command. The GO command is used to start a new pass in the encoding/decoding modes, and continuation of the compression/expansion operation after an interrupt signal is issued, except for a DATRDY interrupt.

The GO command is executed by asserting a $\overline{WR}$, $\overline{CS}$, and ADDR=0 in conjunction with random data or float of the DATA bus.

HARDWARE Register

The fields of this register set the parameters of the host and compressed data interfaces: Slave, DMA or Master mode transfer of compressed data, Compressed Data Transfer width (8 bits or 16 bits) in Slave and DMA modes, and the Compressed Data Transfer cycle time in Master mode.

In Master mode, the compressed data is transferred on the CODE bus. In 8 bit Slave and DMA modes, it is transferred on the DATA bus. In 16 bit Slave and DMA modes, the DATA bus is extended to 16 bits by the CODE bus. In the 16 bit DMA/SLAVE mode, if BELE is set to “0”, the first byte and all other even numbered bytes are output on the DATA bus, and the second byte and all other odd numbered bytes are output on the CODE bus. If BELE is set to “1”, the above byte order is reversed. Refer to section “INTERFACES” and Tables 3 and 2.

BSWD	MSTR	DMA	CFIS	CFIS	CFIS	0	BELE

Figure 27. HARDWARE Register

Table 2. CFIS Setting In Master Mode

CFIS	Compressed Data Interface Bus Speed
000	1 CLK_IN per cycle
001	2 CLK_IN per cycle
010	3 CLK_IN per cycle
011	4 CLK_IN per cycle
100	5 CLK_IN per cycle
101	6 CLK_IN per cycle
110	7 CLK_IN per cycle
111	8 CLK_IN per cycle

MODE Register

This register selects the operating Mode of the device. Refer to Tables 4, 5, 7, 8.

COMP	ATP	PASS2	TLM	DC Only	BRC	0	0

Figure 28. MODE Register

OPTIONS Register

Used only in Encoding modes. The fields of this register select the Overflow detection option, and set the number of scans to be encoded in compression by setting the OVF and NSCN bits respectively. When OVF bit is set, the ZR36050 selects the overflow option. Refer to Tables 4, 5, 6, 7, 8.

NSCN	NSCN	NSCN	OVF	0	0	0	0

Figure 29. OPTIONS Register

MBCV Register

This parameter specifies the Maximum Block Code Volume. When Bit Rate Control is used, MBCV limits the maximum number of bits that will be used to encode each block.

The number of bits is twice the value coded in the 8 bit MBCV register. MBCV=01 represents two bits, and MBCV=FF represents 510 bits.

MARKERS_EN Register

In encoding, this register specifies which of the marker segments to include in the compressed data. Marker segments are APP, COM, DRI, DNL, DQT and DHT. The DQTI and DHTI fields of this register are also used in Table Preload for encoding and decoding, to specify which tables to preload.

APP	COM	DRI	DQT	DHT	DNL	DQTI	DHTI

Figure 30. MARKERS_EN Register

Table 3. Programming the ZR36050 Bus Interfaces For Compressed Data Transfer

Data Transfer Mode	BSWD	MSTR	DMA	CFIS	BELE
Master mode	0	1	0	0-7	x
Slave mode with 8-bit compressed data bus	0	0	0	0	x
Slave mode with 16-bit compressed data bus	1	0	0	0	0/1
DMA mode with 8-bit compressed data bus	0	0	1	0	x
DMA mode with 16-bit compressed data bus	1	0	1	0	0/1

Table 4. Programming the ZR36050 for Encoding Modes

Encoding Mode	COMP	ATP	PASS2	TLM	DC Only	BRC	NSCN	OVF
JPEG Baseline Compression Pass	1	0	1	0	0	0	0-7	0/1
Auto Bit Rate Control	1	1	0	0	0	1	0-7	0/1
Statistical Pass	1	0	0	0	0	1	0-7	0
Compression Pass with Bit Rate Control	1	0	1	0	0	1	0-7	0/1
Tables-only Pass	1	0	1	1	0	0	0	0
Tables Preload for Encoding	1	0	0	1	0	0	0	0

Table 5. Programming the ZR36050 for JPEG Lossless Encoding Mode

JPEG Lossless Encoding Mode	COMP	ATP	PASS2	TLM	DC Only	BRC	NSCN	OVF
JPEG Lossless Compression Pass	1	0	1	0	0	0	0-7	0

Bits of MARKERS_EN are as follows:

APP. Reads the Application segment from the Internal Memory and writes it to the compressed data during the Compression Pass. Can also be used in a Tables-only Pass.

COM. Reads the Comment segment from the Internal Memory and writes it to the compressed data during the Compression Pass. Can also be used in a Tables-only Pass.

DRI. Enables the restart mechanism and writes the DRI marker segment to the compressed data during the Compression Pass.

DQT. Reads the base Quantization Tables defined in the DQT segment in the Internal Memory, multiplies the quantization values by Scale Factor, rounds them to eight bits and writes the results together with the DQT marker and parameters in the compressed data during the Compression Pass or the Tables-only Pass, and without the header in the internal quantization tables during all passes. The number of Quantization Tables to be processed is inferred from the LEN (segment length) parameter of the DQT segment.

DHT. Reads the DHT segment from the Internal Memory, and writes it to the compressed data during the Compression Pass or the Tables-only Pass.

DNL. Reads the DNL segment from the Internal memory and writes it to the compressed file at the end of the first scan of Compression Pass. In scans other than the first, the DNL bit is ignored.

DQTI. Same function as the DQT bit, except the ZR36040 does not transfer the DQT segment to the compressed data during the Compression Pass. If DQT is set, then DQTI must be clear.

DHTI. Reads the Huffman Tables, defined in the DHT segment of Internal Memory, and writes the decoded tables in the Huffman Tables Store. The number of tables to decode and store is inferred from the length parameter of the DHT segment.

Table 6. NSCN Setting of the OPTIONS Register

NSCN	Number of Scans
000	1 scan
001	2 scans
010	3 scans
011	4 scans
100	5 scans
101	6 scans
110	7 scans
111	8 scans

INT_REQ_0, INT_REQ_1 and STATUS_0, STATUS_1 Registers

Two Interrupt Request registers are used in conjunction with two corresponding STATUS registers, to enable generation of interrupts to the host. If a bit in the Interrupt Request register is set, the $\overline{INT}$ pin is activated when the corresponding bit in the STATUS register becomes active, and the ZR36050 stops processing and waits until the host restarts it with a GO command. $\overline{INT}$ is deactivated when the host reads the STATUS register.

The $\overline{RESET}$ pulse, GO command, or reading the STATUS registers reset the bits in the STATUS_(0,1), (except for the END and DATRDY bits of the STATUS_1 register), and disable the $\overline{INT}$ signal. The END bit is set by the $\overline{RESET}$ pulse and is reset by the GO command. The DATRDY bit gets reset by a $\overline{RESET}$ pulse but is unaffected by the GO command (see DATRDY description). Table 9 summarizes the effects of reading the STATUS_1 bits, $\overline{RESET}$ pulse, and the GO command on the STATUS_1 register.

Table 7. Programming the ZR36050 for Decoding Modes

Decoding Mode	COMP	ATP	PASS2	TLM	DC Only	BRC	NSCN	OVF
JPEG Baseline Expansion Pass	0	0	0	0	0	0	0	0
Fast Preview	0	0	0	0	1	0	0	0
Tables Preload for Decoding	0	0	0	1	0	0	0	0

Table 8. Programming the ZR36050 for JPEG Lossless Decoding Mode

JPEG Lossless Decoding Mode	COMP	ATP	PASS2	TLM	DC Only	BRC	NSCN	OVF
JPEG Lossless Expansion Pass	0	0	0	0	0	0	0	0

Table 9. Effect of Reading the STATUS_1 Bits, $\overline{\text{RESET}}$ Pulse, and the GO Command on the STATUS_1 Control Register.

	DATRDY	MRKDET	RFM	RFD	END	TCVOVF	DATOVF
$\overline{\text{RESET}}$	0	0	0	0	1	0	0
GO	-	0	0	0	0	0	0
Reading STATUS_1 register	0	* -	0	0	0	0	0

* = MRKDET is reset to “0” when STATUS_0 is read.

STATUS_0 and INT_REQ_0 bits

In decoding, INT_REQ_0 and STATUS_0 are used to notify the host when specified markers or marker segments have been extracted from the compressed data and written in the internal memory. The markers and marker segments that can be specified are APP, COM, DRI(RST), DQT, DHT, DNL, SOF and SOS. The APP and COM bits allow the extraction of APP or COM segments longer than 64 bytes, by activating $\overline{\text{INT}}$ after each 64 byte section until the whole segment has been extracted. In encoding, only the APP and COM bits are used to control the transfer of APP or COM segments longer than 64 bytes.

APP	COM	DRI (RST)	DQT	DHT	DNL	SOF	SOS

Figure 31. Status_0 and INT_REQ_0 Bits

DATRDY	MRKDET	0	RFM	RFD	END	TCVOVF	DATOVF

Figure 32. Status_1 Bits

DATRDY	0	0	RFM	RFD	END	1	1

Figure 33. INT_REQ_1 Bits

Bits of INT_REQ_1 and STATUS_1 are as follows:

DATRDY. This bit is used only in Slave mode Compressed Data Transfer. It is a special case in that it activates $\overline{\text{DINT}}$ instead of $\overline{\text{INT}}$, and the processing does not stop. In the encoding process, the status bit is set when new compressed data is ready in the Compressed Data Input/Output register, and cleared when the host reads the compressed data. In the decoding process, it is set when the ZR36050 is ready to accept new compressed data, and is cleared when the host reads the STATUS_1 register, or reads from or writes into the Compressed Data Input/Output register.

MRKDET. Marker Detected. This bit is used to notify the host that one of the STATUS_0 bits has been set. MRKDET bit is set whenever any of the STATUS_0 bits is updated, i.e. it is the “OR” function of the STATUS_0 register bits. The host can determine the status of the STATUS_0 register by merely reading this bit. MRKDET does not have a corresponding bit in the INT_REQ_1 register and therefore does not generate an $\overline{\text{INT}}$ signal. Note that the MRKDET is not cleared by reading the STATUS_1 register. It is cleared when the STATUS_0 register is read.

RFM. Request For Markers. RFM is used to notify the host that the ZR36050 is ready to process new marker segments. In encoding and decoding modes, when the ZR36050 stops due to interrupt activation of RFM, it reads the MARKERS_EN, INT_REQ_1, and INT_REQ_2 registers again. In the encoding process, RFM is set before the start of each scan,m except for the first scan. This allows the host to add optional marker segments to each scan, update the SOS marker segments in the internal memory (if more than four scans are used – since the internal memory has provisions for four scans), and change the interrupt request registers for the following scan.

It is also set in Auto Bit Rate Control mode, at the end of the Statistical Pass. In the decoding process, it is set after the last sample of the scan has appeared on the PIXEL bus. RFM is cleared when the host reads the STATUS_1 register. A GO command will resume all internal operations.

RFD. Request For Data is set when the ZR36050 has completed processing of markers and is ready to input or output data on the PIXEL bus.

END. Indicates the end of processing in encoding and decoding, when the ZR36050 enters the Idle state. The $\overline{\text{END}}$ pin reflects the state of the END status bit. The END status bit is cleared when the host issues a GO command.

TCVOVF. Used only with OVF option. Indicates Target Code Volume overflow. This status bit is set by the ZR36050 only if the overflow detection option was selected. When the TCVOVF status bit is set, the ZR36050 aborts the encoding process and goes into the Idle state. This bit must be set to “1” by the host in the INT_REQ_1 register after every $\overline{\text{RESET}}$ in encoding operation with the overflow option chosen. Refer toTable 4.

DATOVF. Indicates overflow of the internal coefficient buffers in encoding operation. Overflow in the internal coefficient buffers can only happen if the ZR36050 activated the $\overline{\text{STOP}}$ pin, and the system did not respond in time by suspending the flow of image data to the PIXEL bus at the end of the current block. When the DATOVF status bit is set, the ZR36050 aborts the encoding process and goes into the Idle state. This bit must be set to “1” by the host in the INT_REQ_1 register after every $\overline{\text{RESET}}$ in encoding operation.

TCV_NET Register

Target Net Code Volume. Used only with OVF option. When the overflow option is chosen, the ZR36050 compares the accumu-

lated code volume against TCV_NET on every cycle. If the accumulated code volume exceeds the TCV_NET value, then the ZR36050 sets the TCVOVF status bit to "1", activates the $\overline{INT}$ signal and aborts the encoding process. The Target Code Volume is in bits unit for the compressed data excluding the marker segments. TCV_NET is represented as a 32 bit fixed point binary number.

TCV_DATA Register

Target Data Code Volume. Used only in Auto Bit Rate Control and Statistical Pass. TCV_DATA is used by the ZR36050 to calculate the new Scale Factor (SF) and Allocation Factor (AF) after a Statistical Pass. It is the Target Code Volume in bits for the compressed data excluding the marker segments, the End Of Block codes, the bit and byte stuffings; (i.e., it is the target code volume for the parts of the compressed data which contain only Huffman codes and the appended data to each code). TCV_DATA is represented as a 32 bit fixed point binary number.

SF Register

Scale Factor. Scale Factor is used for scaling the quantization tables values. SF should be provided to the ZR36050 as a parameter at the beginning of every encoding operation. SF is represented as a 16 bit fixed point binary number, with 8 bits after the binary point.

AF Register

Allocation Factor. Used only in Compression Pass with Bit Rate Control. AF is used to compute the Allocated Code Volume for each block. The AF is computed by the ZR36050 and written to the AF register at the end of the Statistical Pass. This value can then be used in a Compression Pass with Bit Rate Control. Otherwise the user is responsible for inputting the AF value into the Internal Memory register prior to performing a Compression Pass with Bit Rate Control. AF is represented as a 24 bit fixed point binary number, with 19 bits after the binary point.

ACV Register

Accumulated Code Volume. ACV register is used: (1) To store the Net Code Volume in bits unit excluding marker segments, End-Of-Block codes and bit and byte stuffings at the completion of the Statistical Pass (ACV_DATA), and (2) The Net Code Volume in bits including the coded data, End-Of-Block codes, and bit and byte stuffings at the completion of every encoding operation, excluding the Statistical Pass (ACV_NET). Both ACV_DATA and ACV_NET are represented as a 32 bit fixed point binary numbers.

Compressed Data Register

This register is used in Slave mode Compressed Data Transfer. During the encoding process, ZR36050 loads the compressed data into this register for the host to read. Similarly, during the decoding process, the host loads the compressed data into this register for the ZR36050 to read.

ACT Register

This register contains the total activity of the image. ACT is updated at the end of Statistical Pass and Auto bit rate control encoding modes. It is represented as a 32 bit fixed point binary number.

ACV_TRUN

This register contains the total number of truncated bits of the frame as a result of block truncation in Compression Pass with Bit Rate Control and Auto Bit Rate Control encoding modes. ACV_TRUN is updated at the end of Compression Pass with Bit Rate Control and Auto Bit Rate Control encoding modes. It is represented as a 32 bit fixed point binary number.

CONTROL REGISTERS FORMAT

The following table describes the encoding/decoding control registers format. The MBCV, TCV_NET, TCV_DATA, SF, AF, and ACT registers are not used in the JPEG Lossless mode.

Table 10. Control Register Format

Address	Content	Description
000	GO	GO command virtual register (Write only)
001	[0 0]	
002	HARDWARE	Hardware setting register
003	MODE	Operational mode register
004	OPTIONS	Options register
005	MBCV	Maximum block code volume
006	MARKERS_EN	Marker segments enable
007	INT_REQ_0	Interrupt masks for Status register 0.
008	INT_REQ_1	Interrupt masks for Status register 1.
009	TCV_NET(H)	High byte of the Target Net Code Volume
00A	TCV_NET(MH)	Middle high byte of the Target Net Code Volume
00B	TCV_NET(ML)	Middle low byte of the Target Net Code Volume
00C	TCV_NET(L)	Lowest byte of the Target Net Code Volume
00D	TCV_DATA(H)	High byte of the Target Data Code Volume
00E	TCV_DATA(MH)	Middle high byte of the Target Data Code Volume
00F	TCV_DATA(ML)	Middle low byte of the Target Data Code Volume
010	TCV_DATA(L)	Low byte of the Target Data Code Volume
011	SF(H)	High byte of the Scale Factor
012	SF(L)	Low byte of the Scale Factor
013	AF(H)	High byte of the Allocation Factor
014	AF(M)	Middle byte of the Allocation Factor
015	AF(L)	Low byte of the Allocation Factor
016	ACV(H)	High byte of the Accumulated data/total Code Volume
017	ACV(MH)	Middle high byte of the Accum. data/total Code Volume
018	ACV(ML)	Middle low byte of the Accum. data/total Code Volume
019	ACV(L)	Low byte of the Accumulated data/total Code Volume
01A	ACT(H)	High byte of the Accumulated Total Activity
01B	ACT(MH)	Middle High byte of the Accumulated Total Activity
01C	ACT(ML)	Middle low byte of the Accumulated Total Activity
01D	ACT(L)	Low byte of the Accumulated Total Activity
01E	ACV_TRUN(H)	High byte of the accumulated truncated bits
01F	ACV_TRUN(MH)	Middle high byte of the accumulated truncated bits

Table 10. Control Register Format (Continued)

Address	Content	Description
020	ACV_TRUN(ML)	Middle low byte of the accumulated truncated bits
021	ACV_TRUN(L)	Low byte of the accumulated truncated bits
022	Reserved	
...		
02D	Reserved	
02E	STATUS_0	Status register 0 (Read only)
02F	STATUS_1	Status register 1 (Read only).
030	Compressed Data Register	8 or 16-bit Compressed Data Input/Output register in Slave mode
031	Reserved	
.....		
03F	Reserved	

JPEG MARKER SEGMENTS

The formats of the marker segments are implemented according to the JPEG standard IS 10918–1. These segments are: SOF, SOS, DRI, DNL, DQT, DHT, APP, and COM. The SOI, EOI and RSTn are supported internally.

The JPEG marker segments contained in the compressed data are saved into the Internal Memory during the decoding mode. The starting location of each JPEG marker segment in the Internal Memory is fixed in both encoding and decoding modes. However the length and the content of each marker segment may vary. Table 11 represents the JPEG markers segments for the JPEG Baseline mode. The JPEG Lossless marker segments are described in Table 12.

Table 11. JPEG Baseline Marker Segments

Address	Content	Description
040	SOF0[FF]	Start of frame. This variable length segment contains up to 34 bytes that define 1–8 components and is used both in the encoding and decoding modes. If there are fewer than 8 components, the remaining memory locations are left unused
041	SOF0[C0]	
042	LEN(H)	
043	LEN(L)	
044	P[8]	Precision
045	Y(H)	Most significant byte of number of Lines
046	Y(L)	Least significant byte of number of Lines
047	X(H)	Most significant byte of number of Columns
048	X(L)	Least significant byte of number of Columns
049	Nf	Number of Components in a frame
04A	C(1)	ID of the first Component
04B	H(1), V(1)	Sampling ratio factors of the first Component
04C	Tq(1)	Quantization table ID number of the first Component
⋮		
05F	C(8)	ID number of the eighth Component
060	H(8), V(8)	Sampling ratio factors of the eighth Component
061	Tq(8)	Quantization table ID number of the eighth Component
062	Reserved	
⋮		
079	Reserved	
07A	SOS[FF]	Start of the first/fifth scan. There are four groups, 16-bytes each, one for each scan. Only the first SOS is used in the decoding mode. In the case of multiple SOS markers in the compressed file, the old SOS is over written with the new. Note that the example shown here is for a scan consisting of four components. If there are fewer components the length of the segment will be shorter, and the fixed values [00, 3F, 00] at the end of this segment will be at lower addresses.
07B	SOS[DA]	
07C	LEN(H)	
07D	LEN(L)	

Table 11. JPEG Baseline Marker Segments (Continued)

Address	Content	Description
07E	Ns	Number of Components in a scan
07F	C(1)	ID number of the first Component
080	Td(1), Ta(1)	Code tables ID number's selection.
081	C(2)	ID number of the second Component
082	Td(2), Ta(2)	Code tables ID number's selection.
083	C(3)	ID number of the third Component
084	Td(3), Ta(3)	Code tables ID number's selection.
085	C(4)	ID number of the fourth Component
086	Td(4), Ta(4)	Code tables ID number's selection.
087	[00]	
088	[3F]	
089	[00]	
08A	SOS[FF]	Start of the second/sixth scan. Note that the addresses of the SOS markers are fixed.
08B	SOS[DA]	
⋮		
09A	SOS[FF]	Start of the third/seventh scan.
09B	SOS[DA]	
⋮		
0AA	SOS[FF]	Start of the fourth/eighth scan.
0AB	SOS[DA]	
⋮		
0BA	Reserved	6 bytes.
⋮		
0BF	Reserved	
0C0	DRI[FF]	Restart Interval definition. This six-byte segment is used in both the encoding and decoding modes. In the case of multiple DRI markers in the compressed file, the old DRI is overwritten with the new. The RSTs are also written in the first two bytes of this segment during decoding. The host can distinguish between an RST and a DRI by their marker codes.
0C1	DRI[DD]	
0C2	LEN[00]	
0C3	LEN[04]	
0C4	RI(H)	Most significant byte of length of Restart Interval in MCU's
0C5	RI(L)	Least significant byte of length of Restart Interval in MCU's
0C6	DNL[FF]	Define number of lines. DNL is a six-byte segment used in the first scan of encoding and decoding modes.
0C7	DNL[DC]	

Table 11. JPEG Baseline Marker Segments (Continued)

Address	Content	Description
0C8	LEN[00]	
0C9	LEN[04]	
0CA	Y(H)	Most significant byte of number of Lines
0CB	Y(L)	Least significant byte of number of Lines
Address	Content	
0CC	DQT[FF]	Quantization Tables definition. This variable-length segment contains up to 264 bytes used in both the encoding and decoding modes. Up to four tables can be saved in DQT. Each table contains 65 bytes. In the case of multiple DQT markers in the compressed file, the old DQT is overwritten with the new. If there are fewer than four quantization tables, the remaining memory locations are left unused.
0CD	DQT[DB]	
0CE	LEN(H)	Possible values are: 0043, 0084, 00C5, 0106.
0CF	LEN(L)	
0D0	Pq0, Tq0	Start the first Base Quantization Table
0D1	BQ0(0)	
⋮		
110	BQ0(63)	
111	Pq1, Tq1	Start the second Base Quantization Table
112	BQ1(0)	
⋮		
151	BQ1(63)	
152	Pq2, Tq2	Start the third Base Quantization Table
153	BQ2(0)	
⋮		
192	BQ2(63)	
193	Pq3, Tq3	Start the fourth Base Quantization Table
194	BQ3(0)	
⋮		
1D3	BQ3(63)	
Address	Content	
1D4	DHT[FF]	Huffman tables definition. This variable-length segment contains up to 422 bytes used in both the encoding and decoding modes. It can contain up to two DC tables and two AC tables in any order. In the case of multiple DHT marker segments in the compressed file, the old DHT is overwritten with the new. If there are fewer than four Huffman tables, the remaining memory locations are left unused. The following example depicts two DC tables followed by two AC tables.
1D5	DHT[C4]	
1D6	LEN(H)	
1D7	LEN(L)	

Table 11. JPEG Baseline Marker Segments (Continued)

Address	Content	Description
1D8	Nt[00]	Start definition of the first DC Huffman table
1D9	L1	
⋮		
1E8	L16	
1E9	V1	
⋮		
1F4	V12	
1F5	Nt[01]	Start definition of the second DC Huffman table
1F6	L1	
⋮		
205	L16	
206	V1	
⋮		
211	V12	
212	Nt[10]	Start definition of the first AC Huffman table
213	L1	
⋮		
222	L16	
223	V1	
⋮		
2C4	V162	
2C5	Nt[11]	Start definition of the second AC Huffman table
2C6	L1	
⋮		
2D5	L16	
2D6	V1	
⋮		
377	V162	
378	Reserved	6 bytes
⋮		
37F	Reserved	
380	APPn[FF]	Start of APP. This 64-byte segment is used in both the encoding and decoding modes. If more than one APPn is encountered in the compressed file, or is needed in the encoding process, the new APPn always starts from beginning of the segment.

Table 11. JPEG Baseline Marker Segments (Continued)

Address	Content	Description
381	APPn[En]	
382	LEN(H)	
383	LEN(L)	
⋮		
3BF		APP end
3C0	COM[FF]	Start of COM. This 64-byte segment is used in both the encoding and decoding modes. If more than one COM is encountered in the compressed file, or is needed in the encoding process, the new COM always starts from beginning of the segment.
3C1	COM[FE]	
3C2	LEN(H)	
3C3	LEN(L)	
⋮		
3FF		COM end

Table 12. JPEG Lossless Marker Segments

Address	Content	Description
040	SOF3[FF]	Start of frame. This variable length segment contains up to 34 bytes that defines 1–8 components and is used both in the encoding and decoding modes. If there are fewer than 8 components, the remaining memory locations are left unused
041	SOF3[C3]	
042	LEN(H)	
043	LEN(L)	
044	P[2-12]	Precision
045	Y(H)	Most significant byte of number of Lines
046	Y(L)	Least significant byte of number of Lines
047	X(H)	Most significant byte of number of Columns
048	X(L)	Least significant byte of number of Columns
049	Nf	Number of Components in a frame
04A	C(1)	ID of the first Component
04B	H(1), V(1)	Sampling ratio factors of first Component
04C	[00]	
⋮		
05F	C(8)	ID number of last Component
060	H(8), V(8)	Sampling ratio factors of last Component
061	[00]	

Table 12. JPEG Lossless Marker Segments (Continued)

Address	Content	Description
062	Reserved	
⋮		
079	Reserved	
07A	SOS[FF]	Start of the first/fifth scan. There are four groups, 16-bytes each, one for each scan. Only the first SOS is used in the decoding mode. In the case of multiple SOS markers in the compressed file, the old SOS is over written with the new.
07B	SOS[DA]	
07C	LEN(H)	
07D	LEN(L)	
07E	Ns	Number of Components in a scan
07F	C(1)	ID number of the first Component
080	[Td(1), 0]	Code tables ID number's selection.
081	C(2)	ID number of the second Component
082	[Td(2),0]	Code tables ID number's selection.
083	C(3)	ID number of the third Component
084	[Td(3),0]	Code tables ID number's selection.
085	C(4)	ID number of the forth Component
086	[Td(4),0]	Code tables ID number's selection.
087	[01]	Predictor selection
088	[00]	
089	Pt	Point transform
08A	SOS[FF]	Start of the second/sixth scan
08B	SOS[DA]	
⋮		
09A	SOS[FF]	Start of the third/seventh scan
09B	SOS[DA]	
.....		
0AA	SOS[FF]	Start of the forth/eighth scan
0AB	SOS[DA]	
⋮		
0BA	Reserved	6 bytes
⋮		
0BF	Reserved	

Table 12. JPEG Lossless Marker Segments (Continued)

Address	Content	Description
0C0	DRI[FF]	Restart Interval definition. This six-byte segment is used in both the encoding and decoding modes. In the case of multiple DRI markers in the compressed file, the old DRI is overwritten with the new. The RSTs are also written in the first two bytes of this segment during decoding. The host can distinguish between an RST and a DRI by their marker codes.
0C1	DRI[DD]	
0C2	LEN[00]	
0C3	LEN[04]	
0C4	RI(H)	Most significant byte of length of Restart Interval in MCU's
0C5	RI(L)	Least significant byte of length of Restart Interval in MCU's
0C6	DNL[FF]	Define number of lines. DNL is a six-byte segment used in the first scan of encoding and decoding modes.
0C7	DNL[DC]	
0C8	LEN[00]	
0C9	LEN[04]	
0CA	Y(H)	Most significant byte of number of Lines
0CB	Y(L)	Least significant byte of number of Lines
0CC	[00]	
⋮		
1D3	[00]	
1D4	DHT[FF]	Huffman tables definition. This variable-length segment contains up to 422 bytes used in both the encoding and decoding modes. It can save up to two DC tables for JPEG Lossless mode. In the case of multiple DHT marker segments in the compressed file, the old DHT is overwritten with the new. If there are fewer than four Huffman tables, the remaining memory locations are left unused. The following example depicts two DC tables for 12-bit precision. Typically only 2 DC Huffman code tables are used in JPEG Lossless.
1D5	DHT[C4]	
1D6	LEN(H)	
1D7	LEN(L)	
1D8	Nt[00]	Start definition of the first DC Huffman table
1D9	L1	
⋮		
1E8	L16	
1E9	V1	
⋮		
1F5	V13	
1F6	Nt[01]	Start definition of the second DC Huffman table
1F7	L1	
⋮		
206	L16	

Table 12. JPEG Lossless Marker Segments (Continued)

Address	Content	Description
207	V1	
⋮		
213	V13	
214	[00]	
⋮		
379	[00]	
37A	Reserved	
⋮		
37F	Reserved	
380	APPn[FF]	Start of APP. This 64-byte segment is used in both the encoding and decoding modes. If more than one APPn is encountered in the compressed file, or is needed in the encoding process, the new APPn always starts from beginning of the segment.
381	APPn[En]	
382	LEN(H)	
383	LEN(L)	
⋮		
3BF		APP end
3C0	COM[FF]	Start of COM. This 64-byte segment is used in both the encoding and decoding modes. If more than one COM is encountered in the compressed file, or is needed in the encoding process, the new COM always starts from beginning of the segment.
3C1	COM[FE]	
3C2	LEN(H)	
3C3	LEN(L)	
⋮		
3FF		COM end

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... -65°C to +150°C
Supply Voltage to Ground Potential Continuous ..... -0.5V to +7.0V
DC Voltage Applied to Outputs for High Impedance Output State ..... -0.5V to V_{CC} (Max)
DC Input Voltage ..... -0.5V to V_{CC}+0.5V
DC Output Current, into Outputs (not to exceed 200mA total) 20mA/output
DC Input Current ..... -30mA to +5.0mA

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

OPERATING RANGE

Temperature ..... $0°C \le T_A \le +70°C$
Supply Voltage ..... $4.75V \le V_{CC} \le 5.25V$

DC CHARACTERISTICS

Symbol	Parameter	21 MHz		27 MHz		Units	Test Conditions
		Min	Max	Min	Max		
V_{IL}	Input Low Voltage	-0.5	0.8	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	2.0	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage		0.4		0.4	V	I_{OL} = 2mA
V_{OH}	Output High Voltage	2.4		2.4		V	I_{OH} = -400µA
I_{CC}	Power Supply Current		320		400	mA	V_{CC} = 5.0 V V_{IL} = .45V, V_{IH} = 4.0V
I_{SC0}	Standby Current		5		5	mA	
I_{SC1}	Clock Standby Current[1]		15		15	mA	
I_{LI}	Input Leakage Current		±10		±10	µA	$0 < V_{IN} < V_{CC}$
I_{LP}	Pull-Down Leakage Current		50		50	µA	$0 < V_{IN} < V_{CC}$
I_{LO}	Output Leakage Current[2]		±10		±10	µA	$0 < V_{OUT} < V_{CC}$
C_{IN}	Input Capacitance		10		10	pF	
C_{IO}	I/O and Output Capacitance		10		10	pF	

1. This is the standby current when CLK_EN is high.
2. Data buses will have a +50µA leakage current when $\overline{\text{RESET}}$ is high due to the pull-down device.

During AC testing, inputs are driven at 0.4V and 2.4V levels. Unless othrewise specified, switching times are measured from the 1.5V level of DCLK to the 0.8V or 2.0V levels at the input/output.

Figure 34. AC Testing Input, Output

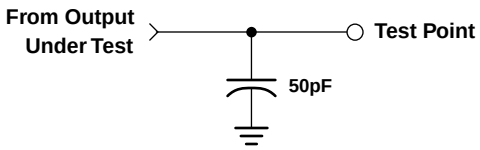


Figure 35. Normal AC Test Load

AC CHARACTERISTICS

Signal Number	Parameter	21 MHz		27 MHz		Units	Test Conditions
		Min	Max	Min	Max		
1	Clock Period, T_{CP}	47.5	400	37	400	ns	
2	Clock High Width	17		10		ns	@2.0V
3	Clock Low Width	17		10		ns	@0.8V
4	Clock Rise Time		3		3	ns	0.8V to 2.0V
5	Clock Fall Time		3		3	ns	2.0V to 0.8V
6	$\overline{RESET}$ Width	4 * T_{CP}		4 * T_{CP}		ns	
7	Synchronous Control Input Setup time	12		12		ns	Note 1
8	Synchronous Control Input Hold Time	0		0		ns	Note 1
9	Pixel Data Input Setup Time	12		12		ns	
10	Pixel Data Input Hold Time	0		0		ns	
11	Pixel and Coefficient Data Output Propagation Delay	1	25	1	22	ns	
12	Miscellaneous Control Output Propagation Delay	1	25	1	22	ns	Note 2
13	Pixel and Coefficient Data Disable Time	0	25	0	22	ns	
14	Bidirectional Control Disable Time	0	25	0	22	ns	Note 3
15	Address and Chip Select Setup Time	5		5		ns	
16	Address and Chip Select Hold Time	5		5		ns	
17	$\overline{RD}$ or $\overline{WR}$ Pulse Width	3 * T_{CP}		3 * T_{CP}		ns	
18	Host Interface Read or Write Recovery Time	1 * T_{CP}		1 * T_{CP}		ns	Note 4
19	Read Data Enable Delay	0		0		ns	
20	Read Data valid Delay		3 * T_{CP}-10		3 * T_{CP}-10	ns	C_L = 50 pF
21	Read Data Hold Time	10	T_{CP}	10	T_{CP}	ns	
22	Write Data Setup Time	40	2 * T_{CP}	40	2 * T_{CP}	ns	
23	Write Data Hold Time	0		0		ns	
24	Interrupt Hold Time After Acknowledge		2 * T_{CP}		2 * T_{CP}	ns	
25	$\overline{DREQ}$ Hold Time After Acknowledge		2 * T_{CP}		2 * T_{CP}	ns	
26	$\overline{DACK}$ Setup Time	5		5		ns	
27	$\overline{DACK}$ Hold Time	5		5		ns	
28	Compressed Data Interface Control Output Propagation Delay	1	15	1	15	ns	
29	$\overline{COE}$, $\overline{CWE}$ Falling Edge Propagation Delay	0.5 * T_{CP} +1	0.5 * T_{CP} +15	0.5 * T_{CP} +1	0.5 * T_{CP} +15	ns	
30	$\overline{COE}$, $\overline{CWE}$ Pulse Width	Note 5		Note 5		ns	
31	CODE Output Propagation Delay (Encoding)	1	25	1	22	ns	
32	CODE Output Disable Time (Encoding)	1	25	0	22	ns	
33	CODE Output Disable Time (Encoding)	0		0		ns	

Signal Number	Parameter	21 MHz		27 MHz		Units	Test Conditions
		Min	Max	Min	Max		
34	CODE Input Setup Time (Decoding)	12		12		ns	
35	CODE Input Hold Time (Decoding)	0		0		ns	

1. $\overline{\text{DSYNC}}$ (encoding), $\overline{\text{EOS}}$ (encoding), $\overline{\text{STOP}}$ (decoding), $\overline{\text{FREEZE}}$, $\overline{\text{CBUSY}}$.
2. $\overline{\text{DSYNC}}$ (decoding), $\overline{\text{EOS}}$ (decoding), $\overline{\text{STOP}}$ (encoding), $\overline{\text{CSYNC}}$, COMP, CL, $\overline{\text{END}}$.
3. Note shown on timing diagrams. The disable time is measured from the rising edge of CLK_IN. Bidirectional control signals are: $\overline{\text{DSYNC}}$, $\overline{\text{EOS}}$, $\overline{\text{STOP}}$.
4. The recovery time applies to consecutive read cycles, consecutive write cycles, read after write, write after read.
5. Minimum $\overline{\text{COE}}$, $\overline{\text{CWE}}$ pulse width is $(k-0.5)T_{CP}$-5ns, where k=CFIS+1 is the number of CLK_IN cycles per bus cycle.

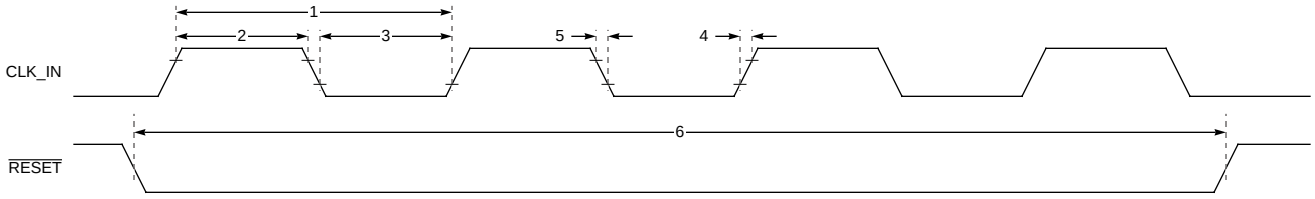


Figure 36. CLK_IN and $\overline{\text{RESET}}$ Timing

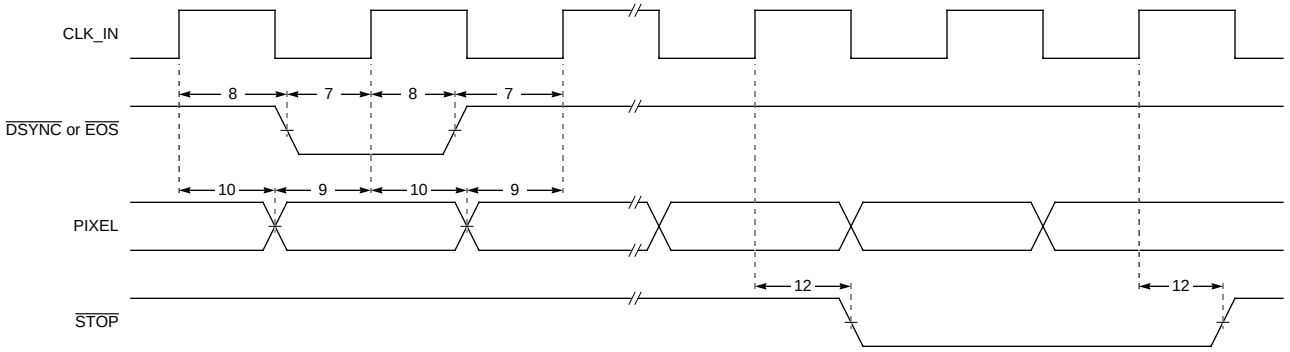


Figure 37. PIXEL Interface Timing (Encoding)

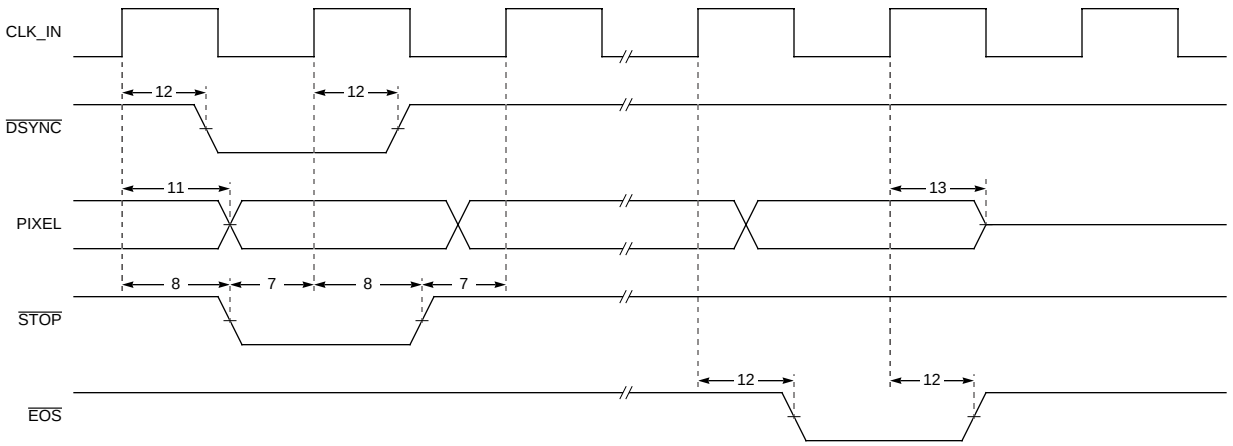


Figure 38. PIXEL Interface Timing (Decoding)

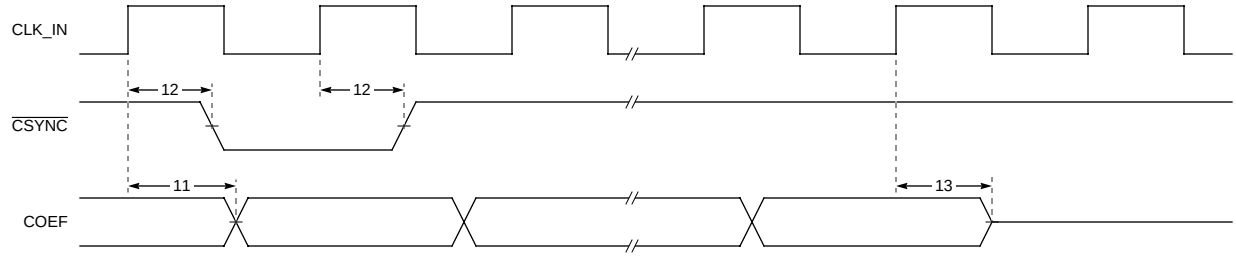


Figure 39. Coefficient Bus Timing

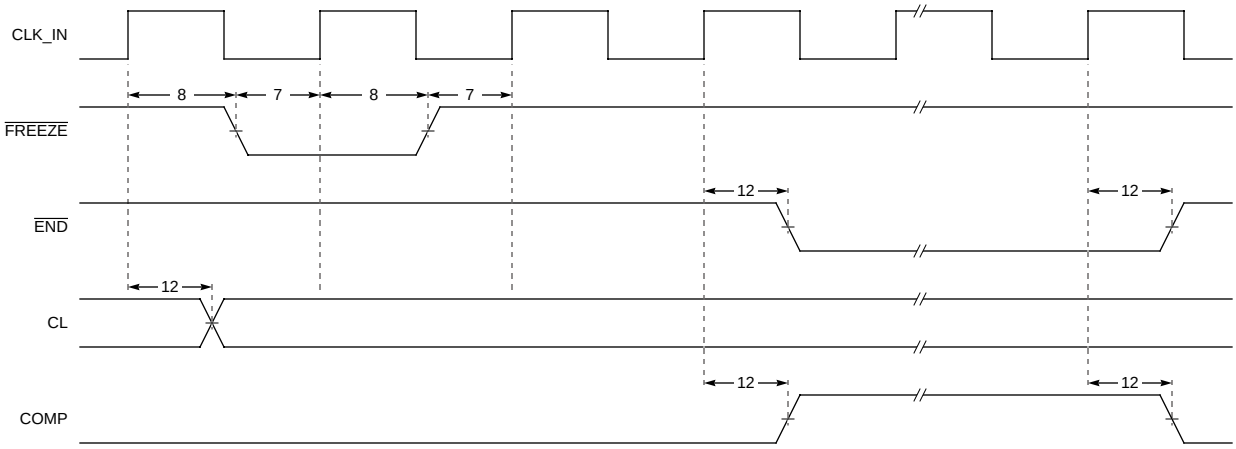
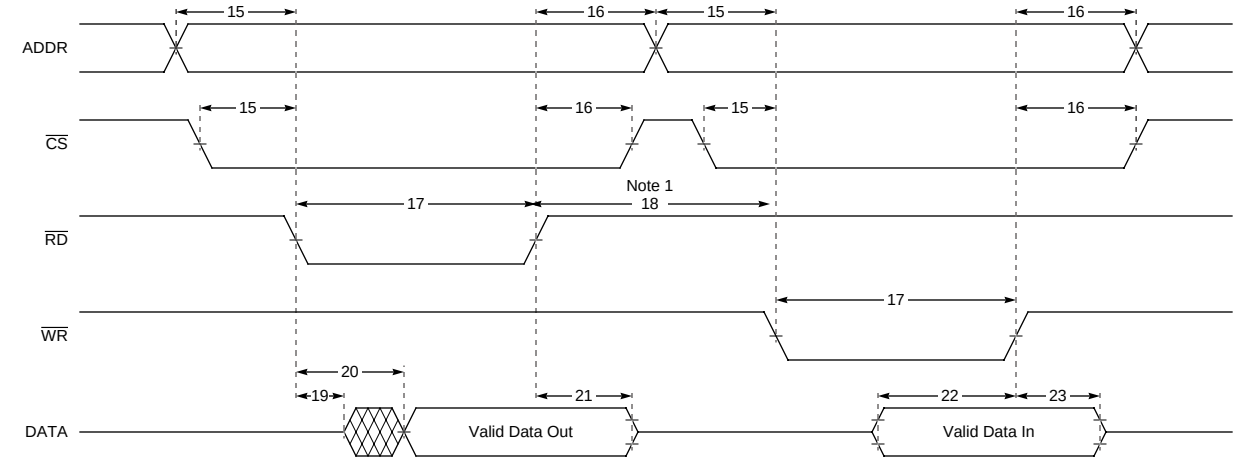
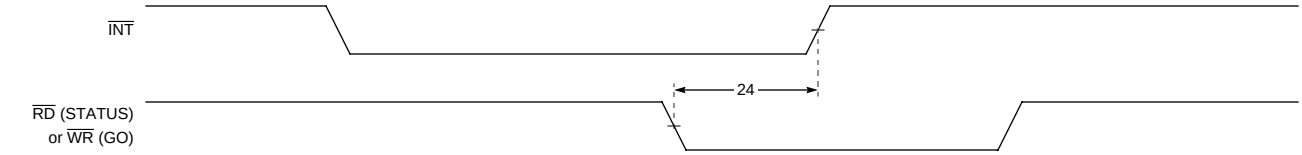


Figure 40. Miscellaneous Control Input and Output Timing



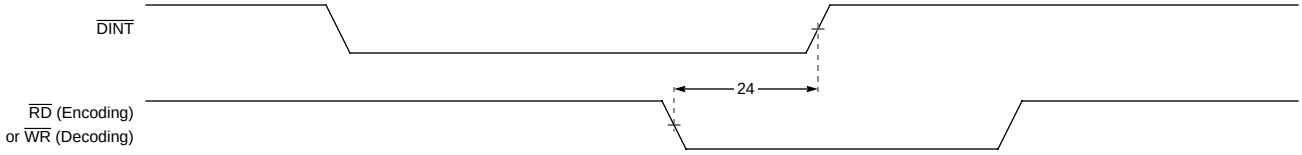
1. This parameter, the recovery time, applies to consecutive read cycles, consecutive write cycles, read-after-write, and write-after-read (shown).

Figure 41. Internal Memory Access Timing



1. With the exception of the INT signal, an interrupt acknowledge cycle is identical to an internal memory access cycle.

Figure 42. Interrupt Acknowledge Timing



1. With the exception of the DINT signal, a slave mode data transfer cycle is identical to an internal memory access cycle.

Figure 43. Slave Mode Compressed Data Transfer Timing

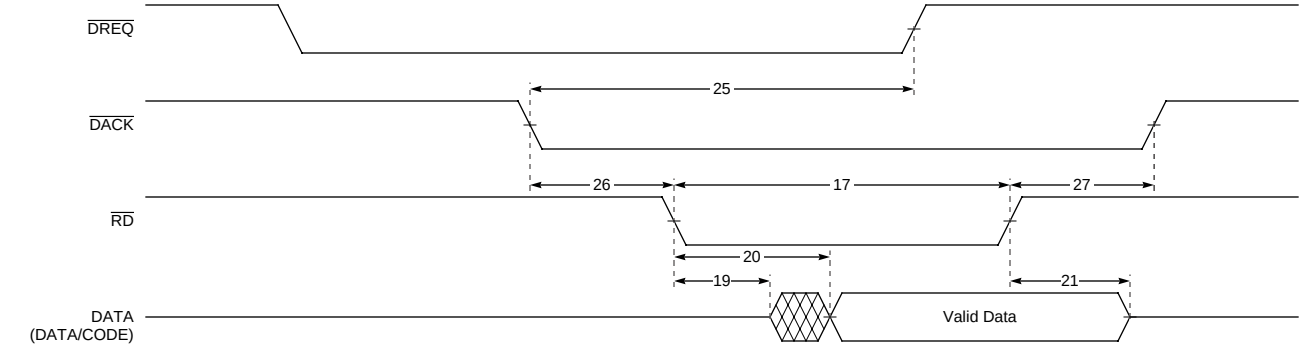


Figure 44. DMA Mode Compressed Data Transfer Timing (Encoding)

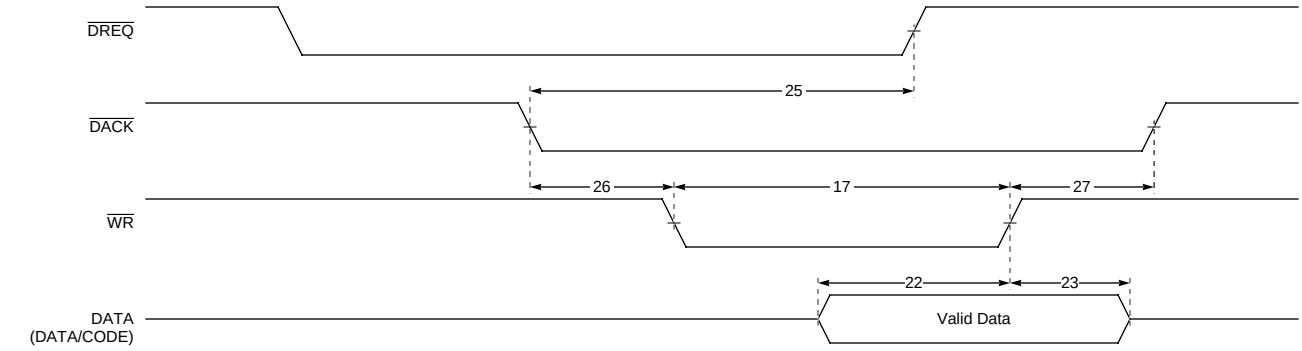
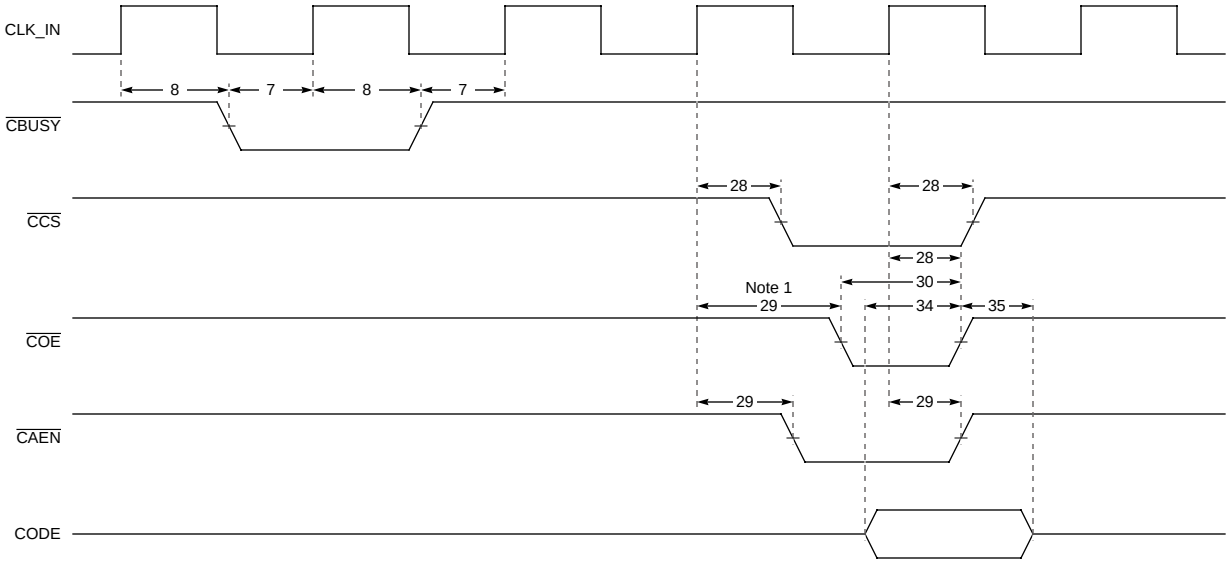
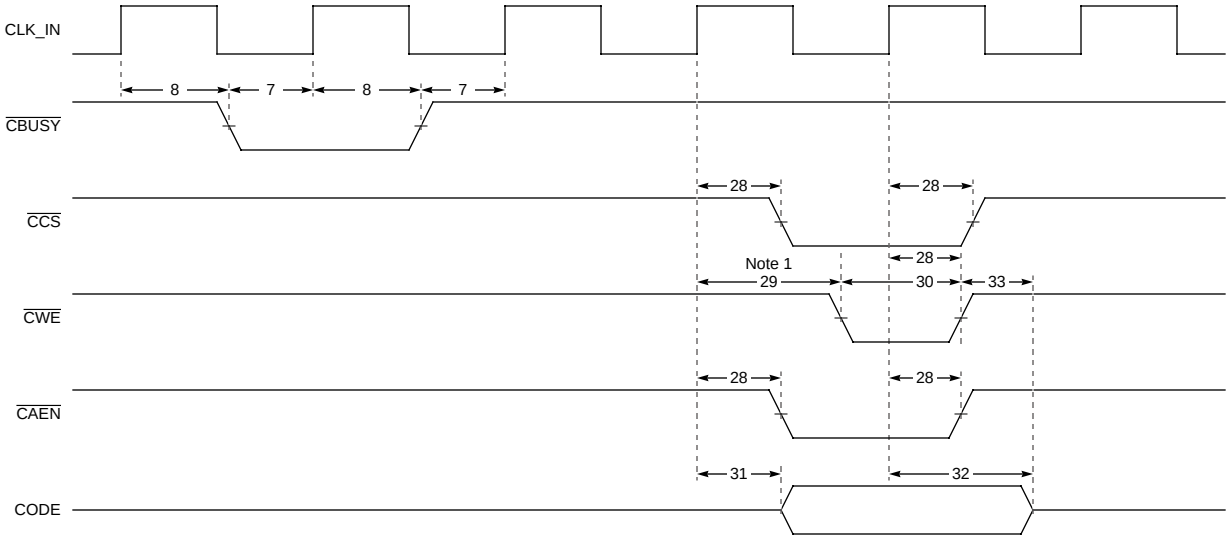


Figure 45. DMA Mode Compressed Data Transfer Timing (Decoding)



1. The falling edge of this output signal is triggered by the internal clock, the output of the PLL clock frequency doubler, at a point half way through the CLK_IN cycle. For external signal compatibility, it is defined relative to the rising edge of CLK_IN.

Figure 46. Master Mode Compressed Data Transfer Timing (Decoding)

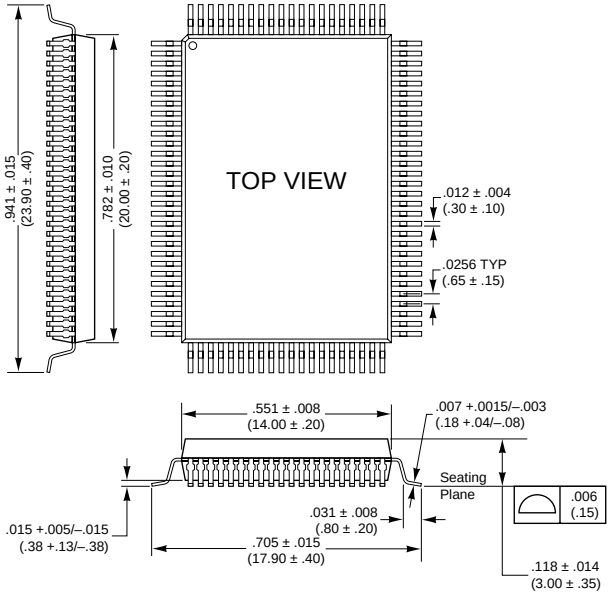


1. The falling edge of this output signal is triggered by the internal clock, the output of the PLL clock frequency doubler, at a point half way through the CLK_IN cycle. For external signal compatibility, it is defined relative to the rising edge of CLK_IN.

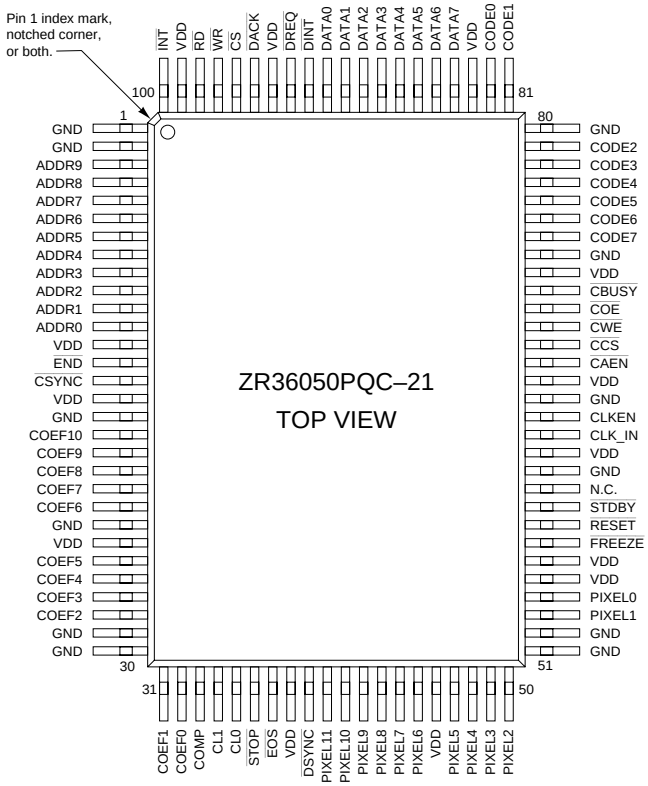
Figure 47. Master Mode Compressed Data Transfer Timing (Encoding)

100-Pin Flat Pack Pin Assignment

Pin No	Pin Name	Type	Pin No	Pin Name	Type	Pin No	Pin Name	Type	Pin No	Pin Name	Type	Pin No	Pin Name	Type
1	GND	–	21	COEF7	O	41	PIXEL10	B	61	GND	–	81	CODE1	B
2	GND	–	22	COEF6	O	42	PIXEL9	B	62	VDD	–	82	CODE0	B
3	ADDR9	I	23	GND	–	43	PIXEL8	B	63	CLK_IN	I	83	VDD	–
4	ADDR8	I	24	VDD	–	44	PIXEL7	B	64	CLKEN	I	84	DATA7	B
5	ADDR7	I	25	COEF5	O	45	PIXEL6	B	65	GND	–	85	DATA6	B
6	ADDR6	I	26	COEF4	O	46	VDD	–	66	VDD	–	86	DATA5	B
7	ADDR5	I	27	COEF3	O	47	PIXEL5	B	67	$\overline{\text{CAEN}}$	O	87	DATA4	B
8	ADDR4	I	28	COEF2	O	48	PIXEL4	B	68	$\overline{\text{CCS}}$	O	88	DATA3	B
9	ADDR3	I	29	GND	–	49	PIXEL3	B	69	$\overline{\text{CWE}}$	O	89	DATA2	B
10	ADDR2	I	30	GND	–	50	PIXEL2	B	70	$\overline{\text{COE}}$	O	90	DATA1	B
11	ADDR1	I	31	COEF1	O	51	GND	–	71	$\overline{\text{CBUSY}}$	I	91	DATA0	B
12	ADDR0	I	32	COEF0	O	52	GND	–	72	VDD	–	92	$\overline{\text{DINT}}$	O
13	VDD	–	33	COMP	O	53	PIXEL1	B	73	GND	–	93	$\overline{\text{DREQ}}$	O
14	$\overline{\text{END}}$	O	34	CL1	O	54	PIXEL0	B	74	CODE7	B	94	VDD	–
15	$\overline{\text{CSYNC}}$	O	35	CL0	O	55	VDD	–	75	CODE6	B	95	$\overline{\text{DACK}}$	I
16	VDD	–	36	$\overline{\text{STOP}}$	B	56	VDD	–	76	CODE5	B	96	$\overline{\text{CS}}$	I
17	GND	–	37	$\overline{\text{EOS}}$	B	57	$\overline{\text{FREEZE}}$	I	77	CODE4	B	97	$\overline{\text{WR}}$	I
18	COEF10	O	38	VDD	–	58	$\overline{\text{RESET}}$	I	78	CODE3	B	98	$\overline{\text{RD}}$	I
19	COEF9	O	39	$\overline{\text{DSYNC}}$	B	59	$\overline{\text{STDBY}}$	I	79	CODE2	B	99	VDD	–
20	COEF8	O	40	PIXEL11	B	60	N.C.	–	80	GND	–	100	$\overline{\text{INT}}$	O

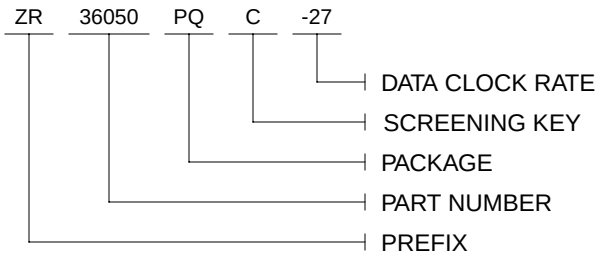


NOTE: Principal dimensions in inches, dimensions in brackets in millimeters.



NOTE: Pins identified as “N.C.” must remain completely unconnected.

ORDERING INFORMATION



PACKAGE
PQ - Plastic Quad Flat Pack (EIAJ)

DATA CLOCK RATE
21 MHz
27 MHz

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

SALES OFFICES

■ ***U.S. Headquarters***
Zoran Corporation
1705 Wyatt Drive
Santa Clara, CA 95054 USA
Telephone: 408-986-1314
FAX: 408-986-1240

■ ***Israel Design Center***
Zoran Microelectronics, Ltd.
Advanced Technology Center
P.O. Box 2495
Haifa, 31024 Israel
Telephone: 972-4-551-551
FAX: 972-4-551-550

■ ***Japan Operations***
Zoran Corporation
1-5-3 Ebisu Kogetsu Bldg.
4th Floor
Shibuya-Ku, Tokyo Japan
Telephone: 81-3-3448-1980
FAX: 81-3-3448-1690

The material in this data sheet is for information only. Zoran Corporation assumes no responsibility for errors or omissions and reserves the right to change, without notice, product specifications, operating characteristics, packaging, etc. Zoran Corporation assumes no liability for damage resulting from the use of information contained in this document.