Motion Estimation Coprocessor (KS0144)

The KS0144 (MEC: Motion Estimation Coprocessor) integrated circuit is a 7.2 GOPS parallel processor specifically optimized for the real-time execution of motion estimation block matching algorithms. The KS0144 is designed to be used in conjunction with the KS0143 (ICC: Image Compression Coprocessor) integrated circuit in motion video image compression applications implementing the MPEG-1, and H.261 compression standards.

RELATED PRODUCTS

- KS0143 Image Compression Coprocessor (ICC)
- VFEB® Hardware Evaluation Kit
- Aspen / Keystone Reference Board
- KS0143ST® Software Tool Kit

APPLICATIONS

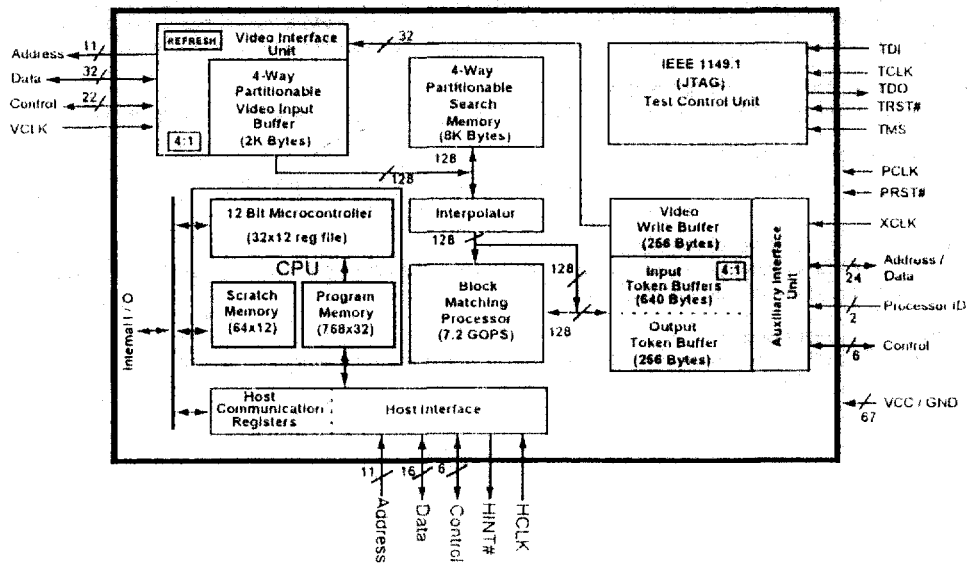
- Video Conferencing and Telephony
- Video Editing
- Video CD-ROM Mastering
- Image Storage and Retrieval
- Multimedia-based Systems

FEATURES

- A single KS0144 performs real-time motion estimation in compliance with MPEG-1 (ISO / IEC International Standard 11172-2) and ITU-T Recommendation H.261 requirements. Up to four KS0144s may be used with a single KS0143 for even greater performance.
- Internal RISC microcontroller and 7.2 GOPS block matching processor execute user-defined search algorithms based on MAD (mean absolute difference) criterion. Supports multi-scale hierarchical searches down to half-pel image resolutions.
- On-chip 8 Kbyte search memory holds search areas from one or more reference images and can be partitioned into 1 to 4 independent search windows.
- Memory-buffered DRAM and KS0143 interfaces transfer reference image pixels, prediction inputs, and prediction outputs in parallel with block matching operations.
- 50 MHz clock, 5W maximum power dissipation.
- 208-pin Metal Quad Flat Pack (MQUAD®), cavity down package.

3-11

FUNCTIONAL BLOCK DIAGRAM



SAMSUNG ELECTRONICS

Samsung Electronics Co., LTD.
Sa #14,Nongseo-Ri, Kihung-Eup, Youngin-Kun,
Kyungki-D,Korea 449-900
Suwon P.O.Box #105

Tel: +82 (331) 280-9454, 280-9484
FAX: +84(331) 280-9459
Email: sspdsp@jupiter. info. samsung. co. kr

REVISION HISTORY	
Revision Number	Release Date
SMP9602DS v1.0	7/93
SMP9602DS v1.1	7/93
SMP9602DS v1.2	1/95
KS0144DS v1.3	4/96

Product Ordering Information:

Part Number	Speed	Processing	Temperature Range (case)	Voltage Range	Pin Coun	Package Typ
KS0144-50	50 MHz	Commercial	0× to 100× C	4.75 to 5.25V	208	MQUAD[a]
KS0144-40	40 MHz	Commercial	0× to 100× C	4.75 to 5.25V	208	MQUAD

Trademark acknowledgments:

KS0143 / KS0144 is a trademark of Samsung Electronics Co.,LTD.
VIDEOFLOW is a trademark of Array Microsystems Inc.
Intel and i960 are trademarks of intel Corporation
29KTM is trademark of Advanced Micro Devices, Inc.

WARNING - LIFE SUPPORT APPLICATIONS POLICY

Samsung Electronics products should notbge used within Life Support Systems without specific written consent of Samsung Electronics Co., LTD. A Life Support System is a product or system intended to support or sustain life which, if it fails, can be reasonably expected to result in significant personal injury or death.

NOTICE - DOCUMENTATION SUPPORT POLICY

The information in this publication is subject to change without notice. Samsung assumes no responsibility for any errors or omissions, and disclaims responsibility for any consequences rsulting from the use of the information included herein. Additionally, Samsung assumes no responsibility for the functioning of undescribed features or parameters.

NOTICE - APPLICATIONS SUPPORT POLOCY

Samsung Electronics Co., LTD . assumes no liability for applications assistance or customer product design.

NOTICE - PRODUCT AVAILABILITY

Samsung Electronics Co., LTD. reserves the right to make changes to product specifications or to discontinue products without notice.

NOTICE - STATED OR IMPLIED WARRANTY

This publication neither states nor implies any warranty of any kind, including byt not limited toimplied warranties of merchantability or fitness for a particular application.

NOTICE - COPY PERMISSION

Permission is hereby expressly granted to copy this publication for informatinal purposes only.
Copying this material for any other use is strictly prohibited.

INTRODUCTION TO THE KS0143 / KS0144 CHIP SET

The KS0143 (ICC: Image Compression Coprocessor), together with its companion chip, the KS1044 (MEC: Motion Estimation Coprocessor), form the heart of a programmable video codec for implementing the JPEG, MPEG-1, and H.261 compression standards. The codec architecture shown in Figure 1.1 may be easily programmed to implement any of the following applications at 30 frames per second:

- H.261 codec processing CIF (352h x 288v) images
- MPEG-1 SIF (352h x 240v) encoder with up to two B frames per P frame and half-pel motion estimation
- MPEG-1 SIF decoder
- JPEG CCIR-601 encoder or decoder

The KS0143 implements all portions of these standards other than motion estimation, variable length coding (VLC), and bit stream syntax handling. The KS0144 executes user-defined motion estimation search algorithms and is only required in systems implementing MPEG-1 or H.261 motion-predictive encoders.

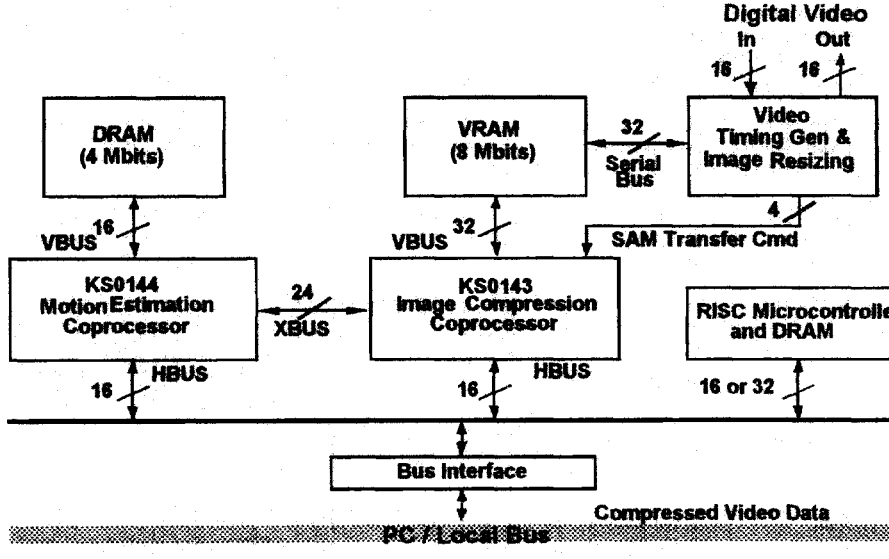
VLC and bit stream syntax handling are typically performed using an off-the-shelf RISC microcontroller which also controls both the KS0143 and KS0144 via their host processor interfaces. The KS0143 / KS0144 chip set does not perform color space conversion or image resizing.

VIDEOFLOW ® Architecture

The KS0143 / KS0144 chip set is based on a very high performance parallel processing architecture. Its parallel processing KS0144hanism has been implemented using a unique vector dataflow architecture which delivers exceptionally high performance and decouples the user from the complexities of a parallel processor. This vector dataflow (VIDEOFLOW) architecture combines dataflow control with vector instruction execution, allowing a user to control a very complex parallel processor via simple program flowgraphs. In addition, the VIDEOFLOW architecture is highly scalable and modular, allowing Array Microsystems to readily produce architecturally compatible derivative products.

3-11

Figure 1.1 - Typical Application
Programmable video codec for JPEG, MPEG-1, and H.261-based multimedia applications



SAMSUNG ELECTRONICS

KS0143 / KS0144 Chip Set Performance

The parallel processing architectures embodied in the KS0143 / KS0144 chip set permit it to deliver exceptional performance for the JPEG, MPEG-1, and H.261 standards. Table 1.1 shows representative real-time (i.e. 30 frames per second) applications of the standards using the 40 and 50 MHz versions of the chip set. The KS0144 is not required for H.261 decoder, MPEG-1 decoder, or JPEG applications.

A unique feature of the chip set, owing to the programming style and very high performance of the KS0143, is its ability to simultaneously process multiple image channels. For example, for multipoint video conferencing, a single KS0143 / KS0144 chip set may be used to implement an H.261 video codec capable of simultaneously encoding a single image sequence (e.g. the one being transmitted) and decoding multiple received bit streams. Also, for MPEG-1 multimedia applications running in a windowed environment, a single KS0143 / KS0144 can decode two different SIF image bit streams in real time.

As shown in Table 1.1, another unique feature of the chip set is its ability to decode MPEG-1 images as they are encoded. This "real-time preview" feature is the result of the high performance and programmability of the chip set. Real-time previewing allows the user of MPEG-1 multimedia applications to create bit streams and visually monitor encoding results without the need for additional hardware for MPEG-1 decoding.

Furthermore, the ability of the KS0143 chip to real-time JPEG encode or decode the large image sizes shown in Table 1.1 is extremely useful in video editing applications. These applications typically operate with very high resolution images and compress them prior to storing them on disk.

High Speed, Compatible Bus Interfaces

As shown in Figure 1.1, the KS0143 and KS0144 have three bus interfaces called HBUS, XBUS, and VBUS which are compatible between the two chips. Each bus interface operates synchronously with its own dedicated clock (HCLK, XCLK, and VCLK). A separate clock (PCLK) is used for internal data processing and has a maximum frequency of 40 or 50 MHz depending upon the speed grade.

The HBUS is a 16-bit, 33 MHz synchronous host processor interface optimized for connection to the buses of RISC microcontrollers from families such as the Intel¨ i960® and Advanced Micro Devices 29K®. The KS0143 and KS0144 operate exclusively as slaves on the HBUS. The host processor (RISC microcontroller) accesses registers and memories within these chips via memory-mapped accesses.
Both the KS0143 and KS0144 contain on-chip program memories which are typically loaded by the RISC microcontroller over the HBUS. The KS0143 instruction RAM is 128 words by 40 bits and the KS0144 instruction RAM is 768 words by 32 bits.

The highest bandwidth utilization of the HBUS occurs during the transfer of zero-run length coded symbols between the KS0143 and host processor during real-time compression and decompression operations. The HBUS also provides a level-sensitive interrupt from the KS0143 or KS0144 to the host processor.

The XBUS is a 24-bit, 40 MHz synchronous auxiliary processor bus used primarily for vector data transfers between the KS0143 and KS0144. The user may also define KS0143 ÒexternalÓ instructions to direct transfers over the XBUS which may be connected to processors other than the KS0144.

Table 1.1 - KSKS0143/KS0144 Performance Benchmarks at 30 Frames Per Second (unless otherwise noted)

	KS0143CQ-50 KS0144CQ-50	KS0143CQ-40 KS0144CQ-40
H.261 Codec	CIF (352h x 288v) encode + 1 CIF decode, or QCIF (176h x 144v) encode + 6 QCIF decodes	QCIF encode + 1 CIF decode @ 15 fps, or QCIF encode + 6 QCIF decodes @ 30 fps
MPEG-1 Encode	SIF (352h x 240v) IBP encode with real-time preview	SIF (352h x 240v) IBP encode, or SIF (320h x 240v) IBP encode with real-time preview
MPEG-1 Decode	2 SIF (352h x 240v) IBP decodes at 30 fps, or 1 SIF IBP decode @ 60 fps	2 SIF (352h x 240v) IBP decodes @ 30 fps, or 1 SIF IBP decode @ 60 fps
JPEG Encode or Decode	720h x 480v, 4:2:2 YCrCb	720h x 480v, 4:2:2 YCrCb

The KS0143 is always the master of the XBUS, and up to four slave processors may be connected to it. In particular, up to four KS0144 chips may be connected to a single KS0143 for the purpose of allowing more searches of a given search area or allowing a search area to be expanded beyond the capacity of a single KS0144. During motion estimation operations, the XBUS is used by the KS0143 to send 16 x 16 blocks of luminance pels to the KS0144 for prediction. The KS0144 in turn, uses the XBUS to return motion vectors and their corresponding 16 x 16 pel blocks found during motion estimation searches.

The VBUS is a multi-master, 32-bit memory bus designed to be directly connected to the data, address, and control pins of DRAMs and VRAMs. The timing characteristics of signals on the VBUS are determined by user-programmable registers within the KS0143 and KS0144. VBUS data transfers may be dynamically resized to 16 or 32-bits.

The VBUS data interface is divided into byte-wide lanes, each lane transferring one 8-bit pel. In 16-bit mode, the VBUS transfers two pels at a time and in 32-bit mode transfers four pels at a time. All transfers are performed as fast page mode accesses at a maximum rate of one column access every 50 nsec, allowing a 16 x 16 pel block to be read or written in about 4.5 usec. The VBUS has an 11-bit address bus and three bits for memory selection which can access up to eight different images of up to 4096 by 4096 pels each. A programmable CAS-before-RAS refresh controller is also included in the VBUS interface logic of both the KS0143 and KS0144.

The VBUS on both the KS0143 and KS0144 may be awarded to other bus masters via serial daisy chained bus arbitration. For example, an external bus master may be used to transfer images into or out of DRAMs connected to the KS0143. If an application permits, the KS0143 and KS0144 VBUS interfaces may be connected together in order to share the same memories. In this case, either the KS0143 or KS0144 is programmed to be at the head of the arbitration daisy chain, and one chip requests the VBUS from the other.

3-11

A unique feature of the KS0143 VBUS interface logic allows the KS0143 to transfer data between the serial and dynamic memory portions of VRAMs. This allows the serial port on VRAMs rather than the random access port on DRAMs to be used for image input/output, making the interface with external logic simpler and faster. Internal VRAM data transfers may be requested by either a video timing generator which asserts the KS0143 Òvideo transfer commandÓ pins (VCMD[3:0]) or the host processor through a memory mapped register.

Input, output, and scratch images are stored in DRAM- or VRAM-based memories connected to the VBUS on the KS0143 and KS0144. The 4 Mbit memory shown in Figure 1.1 stores the reference images used during H.261 and MPEG-1 motion estimation searches and may be implemented using a single 256K by 16-bit DRAM. The 8 Mbit memory stores input and output images as well as H.261 and MPEG-1 decoder reference images. It may be implemented using two 256K by 16-bit or four 256K by 8-bit VRAMs depending on how image input/output is performed. It is also possible to use DRAMs in place of the VRAMs.

VIDEOFLOW Support Tools

Array Microsystems, Inc. supplies a Hardware Evaluation Kit and Software Tool Kit to help customers design their KS0143/KS0144-based systems.

The VFEB Hardware Evaluation Kit provides a hardware vehicle for designers and application developers wishing to explore the real-time performance and multi-standard versatility of the KS0143 and KS0144 chips. It consists of a PC plug-in board for the ISA or VL bus, embedded software, a graphical user interface (GUI) that is compatible with the Windows™ operating system, and User Guide.

The KS0143/KS0144-based PC plug-in board and GUI allow the user to perform real-time (30 fps) demonstrations of the MPEG-1, JPEG, and H.261 standards for both video encoding and decoding. For encoding, users can connect their own analog video to the board and store compressed video to disk in real-time. Decoding demonstrations can read compressed video from either disk or CD-ROM and display the results on a video monitor.

The ICCST Software Tool Kit provides a UNIX workstation-based software development environment for designers wishing to architect, program, simulate, and verify video compression solutions based on the KS0143 and KS0144 chips. The tool kit contains:

- Programming tools consisting of a graphical flowgraph editor, assembler and linker for the KS0143, and a C-compiler and assembler for the KS0144
- Simulation tools consisting of an KS0143/KS0144 system simulator, configuration tools, program debugging aids, and various graphical displays for monitoring program performance
- Image analysis tools supporting the display of image sequences at or near real-time on a workstation monitor with optional graphical overlays of color-coded compression algorithm decisions and motion vectors produced by the simulator

- Comprehensive user and reference guides
- Demonstration package containing programs and simulation examples for the MPEG-1, JPEG, and H.261 video compression standards

Section 2 - KS0144 FUNCTIONAL DESCRIPTION

INTRODUCTION TO KS0144 MOTION ESTIMATION

Successive pictures in a motion video sequence normally exhibit a high degree of correlation. Video compression algorithms take advantage of correlation between pictures by predicting a picture from past and/or future pictures. Prediction improves both the compression ratios and the picture quality.

Block-coding algorithms (such as H.261 and MPEG-1) segment each picture into small regions. The H.261 and MPEG-1 standards use a YCrCb image color space and segment picture data into 4:2:0 macroblocks. Each macroblock consists of a 16x16 block of luminance pels and two spatially coincident 8x8 blocks of chrominance pels. Since different regions in a picture may contain objects moving in different directions and might require different prediction modes, segmentation into macroblocks allows for more efficient coding of predicted pictures.

Each predicted macroblock is represented by a motion vector (or vectors) and blocks of differential pels. A motion vector for a macroblock specifies the translation of the luminance pels. The vector for the chrominance pels is derived by halving the motion vector. The H.261 standard requires motion vectors to fall on full pel coordinates, while MPEG-1 permits vectors to fall on either full or half pel coordinates. Using a half pel coordinate grid allows for better predictions since the best prediction may occur with a fractional pel motion vector.

When the prediction is not exact, blocks of differential pels represent the difference between the prediction and the actual macroblock pels. The small differential pels in the predicted (inter-coded) blocks can be coded more efficiently than pels in non-differenced (intra-coded) blocks improving the compression ratio.

Prediction Modes

Compression algorithms allow macroblocks to be predicted using different prediction modes. The H.261 standard only allows macroblocks to be predicted from the previously coded picture (forward prediction), where MPEG-1 allows macroblocks to be predicted from pictures in the past and/or future (forward, backward, or bi-directional prediction). Bi-directional predictions are formed by averaging the pels from forward and backward predictions and have one motion vector pointing to each.

The KS0143 and KS0144 are specifically architected to support any or all of these prediction modes in a highly efficient manner. For example, an KS0144 program to encode an MPEG "B" (bi-directionally predicted) picture returns one forward and one backward prediction for each macroblock. The KS0143 program examines each of these predictions and the bi-directional prediction formed by averaging these predictions to select the best prediction mode to use.

Block Matching and Mean Absolute Difference

The process used to determine the best prediction for each macroblock is called a motion estimation algorithm. A good motion estimation algorithm is a combination of several techniques and strategies. The most common technique used by motion estimation algorithms is block matching. Block matching is normally performed using only the luminance component of the YCrCb picture, since the luminance component exhibits the highest resolution and the best contrast.

Block matching forms the core of an KS0144 motion estimation algorithm. The KS0144 contains hardware optimized for performing block matches using the well-known "Mean Absolute Difference" or (MAD) criterion to rate the correlation of each block match. The absolute values of the pel-by-pel difference of the two blocks being correlated are summed to form the MAD. The best match is the block match having the smallest MAD.

In most cases, the best block match is used as the prediction. However, always using the best match for prediction is not the best strategy for a motion estimation algorithm. For instance, since predictions are made in blocks and not objects, a block may contain part moving object and part stationary background. Many motion estimation algorithms make a "MC vs. non-MC" decision to choose the zero motion prediction if it is only slightly worse than the best match prediction. This decision prevents the appearance of the background being dragged along with moving objects (a common low bit-rate video compression artifact). Also, macroblock motion vectors that are the same as motion vectors for adjacent blocks can reduce block boundary discontinuities.

In addition to the visual artifact concerns, reducing the number of bits required for coding an individual macroblock allows those bits to be used to improve the overall picture quality.

Typical bit reduction strategies take advantage of the fact that H.261 and MPEG "P" (forward predicted) pictures do not transmit zero motion vectors. Also, a macroblock may not have to be coded if the motion vectors for the macroblock are the same as the previously coded motion vectors.

The KS0144 CPU can be programmed to implement these and other enhancements to the basic block matching and mean absolute difference criterion as part of a comprehensive motion estimation algorithm.

Search Strategies

The motion vector ranges permitted under H.261 and MPEG-1 differ greatly. H.261 motion vectors are limited to the range ±15 pels, since the "talking heads" video scenes common in H.261 only contain limited motion. MPEG-1, designed to encode a wide range of video material, permits very large motion vectors using either full or half pel coordinates. In the normal case, MPEG-1 allows motion vector components in half pel coordinates falling in the range [-512.0, 511.5]. When using full pel coordinates, MPEG-1 limits motion vectors components to the range [-1024, 1023].

For some applications, the KS0144 has sufficient performance to exhaustively test every possible motion vector in the permitted motion vector range for the best block match. However, for very large search ranges, this is not possible and the KS0144 CPU must be programmed with a search strategy that finds a good block match without testing every possibility.

A typical search strategy breaks the task of searching for a good block match into several stages. Each stage reduces the number of candidate predictions that needs consideration. Common simple search strategies include the three-step search, 2D logarithmic search, checkerboard search, conjugate direction search, and half-pel refinement search.

To help search large areas quickly, the KS0144 supports multi-resolution or hierarchical searches. In a multi-resolution search, the first stage searches a low resolution picture for the best match using either an exhaustive search or a simple search strategy. The low resolution search eliminates a large portion of the search area from consideration. The next stage searches the full resolution picture in a neighborhood around the best low resolution match for the best match. Figure 2.1 shows an example of a two-level hierarchical search supported by the KS0144.

Searching the low resolution picture allows a motion estimation algorithm to consider a very large motion vector range. A search area in a low resolution picture is equivalent to a much larger search area in the full resolution picture. Also, each block match at low resolution uses less computational power than a full resolution block match.

The KS0144 CPU can be programmed to execute both simple and multi-resolution search strategies. The KS0144 block match processor supports multi-resolution searches by allowing MAD computations to be performed on 8x8 or 16x16 blocks using either half or full pel motion vector coordinates.

Search Windows

The search window is a region of pels in a past or future picture containing all the candidate predictions considered by the search strategy. To allow the block match processor to proceed at full speed and to reduce the amount of video bus traffic, the KS0144 contains 8 Kbytes of fast on-chip Search Memory to store search windows.

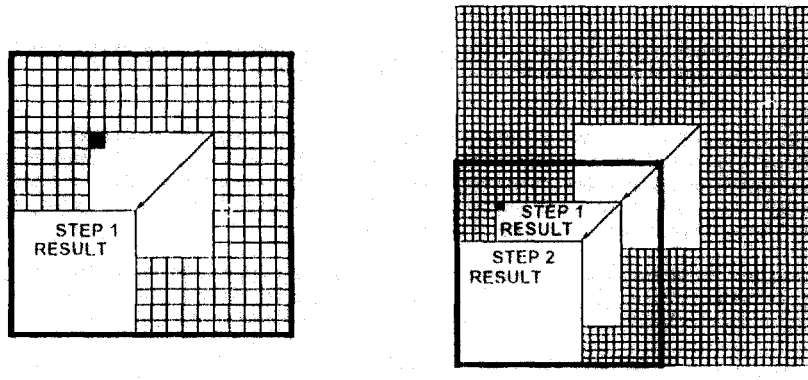
The amount of scene motion that a motion estimation algorithm is designed to detect determines the dimensions of the search window. A typical H.261 motion estimation algorithm designed to detect horizontal and vertical motion up to ±15 pels around a 16x16 block needs a 46x46 search window (2*15+16 = 46).

MPEG-1 search windows are typically much larger. In the common "B B I B B P B B P B B P B B P" Group of Pictures (GOP) encoding sequence, encoders must predict "P" pictures from the previous "I" or "P" picture. For a motion estimation algorithm designed to track motion equivalent to ± 8 pels per picture, the three picture spacing of "P" pictures requires a motion vector search range of ± 24 pels resulting in a 64x64 search window (2*24+16 = 64).

Search windows do not need to be square. For example, horizontal panning is one type of scene motion frequently encountered in film and television. With horizontal panning motion, vertical scene motion is much smaller than horizontal motion. A motion estimation algorithm optimized for horizontal panning motion uses a search window much wider than it is high. An MPEG-1 motion estimation algorithm for the common GOP encoding sequence that is designed to detect ± 8 pel per picture vertical motion and ±16 pel per picture horizontal motion will require a 112h by 64v search window for each "P" picture macroblock.

Forward and backward searches require separate search windows. For multi-resolution searches, each resolution search requires its own search window. The KS0144 Search Memory can be partitioned into one to four different search windows to support simultaneous forward, backward, and multi-resolution motion estimation algorithms.

Figure 2.1 - Two - Level Hierarchical Serach for ±16 Pel Motion



STEP 1:
SEARCH ON QUARTER RESOLUTION IMAGE
■ = CURRENT SEARCH ORIGIN

STEP 2:
SEARCH ON FULL RESOLUTION IMAGE
□ = OUTLINES CURRENT SEARCH AREA

KS0144 Performance Benchmarks

The key performance requirements for a single KS0144 are to perform real-time motion estimation for H.261 CIF (352h x 288v) video codecs or MPEG-1 SIF (352h x 240v @ 30 fps or 352h x 288v @ 25 fps) video encoders. This means that for an H.261 codec running at 30 fps, an KS0144 must be able to perform motion estimation on a single macroblock in an average time of about 84 msec. For an MPEG-1 video encoder, the average time per macroblock is about 101 msec.

Table 2.1 shows how the KS0144 fulfills these requirements for various search ranges and algorithms. Exhaustive search examines every possible block translation in a search area. Hierarchical search uses a two level quarter/full resolution search strategy. Checkerboard searches examine only those translations which fall on a checkerboard-like pattern. The execution times in the table assume no half pel searches and increase by about 2.4 msec if a final half pel refinement step is included.

Table 2.1 - Motion Estimation Performance

Search Range	Search Algorithm	Execution Time @ 50 MHz
± 8h x ± 8v	exhaustive	37 usec
± 15h x ±15v	hierarchical	21 usec
± 15h x ±15v	checkerboard	69 usec
± 15h x ±15v	exhaustive	123 usec
± 24h x ±24v	hierarchical	47 usec
± 30h x ±30v	hierarchical	68 usec
± 48h x ±24v	hierarchical	94 usec

The table entries for the ± 8h x ± 8v and ± 15h x ± 15v search ranges show that a single KS0144 easily meets 30 fps performance goals for H.261 using almost any algorithm. The entries for the other search ranges can be applied to MPEG-1 motion estimation. For example, ± 48h x ± 24v is the search range requirement for MPEG-1 encoding of P pictures assuming two B pictures per P picture and ± 16h x ± 8v pel motion be- tween pictures.

Multiple KS0144 chips may be connected to a single KS0143 chip to either increase a search range beyond that which can be handled by a single KS0144 or decrease the time required to search a smaller search range.

KS0144 FUNCTIONAL UNIT DESCRIPTIONS

As shown in the KS0144 Functional Block Diagram on Page 1, the KS0144 contains several computational, memory, and interface units enabling it to perform motion estimation functions in MPEG-1 and H.261 video encoders. These units are described in the following sections.

Central Processor Unit (CPU)

All KS0144 activities are coordinated by the CPU. CPU programs are stored in a 768 word on-chip instruction RAM which the user loads from the host processor interface. The CPU instruction set allows a wide variety of block-oriented motion estimation algorithms to be programmed.

During motion estimation algorithm execution, the CPU program typically performs the following series of operations:

- Commands the KS0144 Video Interface Unit to fetch search window pels from DRAM into the Video Input Buffer (VIB) and then copy the VIB to the Search Memory.
- Computes search window coordinates of candidate predictions in the Search Memory as determined by the search algorithm.
- Commands the Block Matching Processor (BMP) to compare the candidate predictions at these coordinates with the input macroblock from the KS0143 in the Input Token Buffer.
- Decides which candidate is the best match, computes the motion vector describing its location relative to the input macroblock, and sends both the match and its motion vector back to the KS0143 via the Output Token Buffer.

Many of these operations can be executed concurrently. For example, while the VIB is being loaded from DRAM for the next macroblock, the BMP can be performing block matches for the current macroblock. Likewise, while the latter two operations are proceeding, the CPU can be computing new search coordinates.

As shown in Figure 2.2, the CPU consists of a

- 12-bit microcontroller core,
- 768 word by 32-bit program memory,
- 64 word by 12-bit scratchpad memory,
- 16-bit timer, and
- debug controller.

The microcontroller core primarily consists of a

- 4 port by 32 word by 12-bit register file,
- 12-bit ALU, and
- 12-bit bidirectional barrel shifter.

The 25 MIPS core requires two PCLK clock cycles to execute each of its 22 native operation codes except those dealing with memory loads and program jumps. Memory load and program jump opcodes require an extra two clock cycles to complete. The instruction set of the core is described in Section 5.

The CPU communicates with the scratchpad memory, timer, and other memories and registers throughout the KS0144 using memory-mapped load and store instructions. The CPU however, does not have access to any image pels. The CPU load/store memory map is described in Section 6 and allows the CPU to control KS0144 functions in a highly concurrent fashion. The CPU scratchpad memory and timer are described in Sections 6.1 and 6.2, respectively.

Host Interface Unit (HIU)

The HIU allows the host processor connected to the HBUS to:

- Load instructions into the CPU program memory,
- Send and receive messages with the CPU,
- Control and respond to CPU-generated interrupts,
- Debug CPU programs.

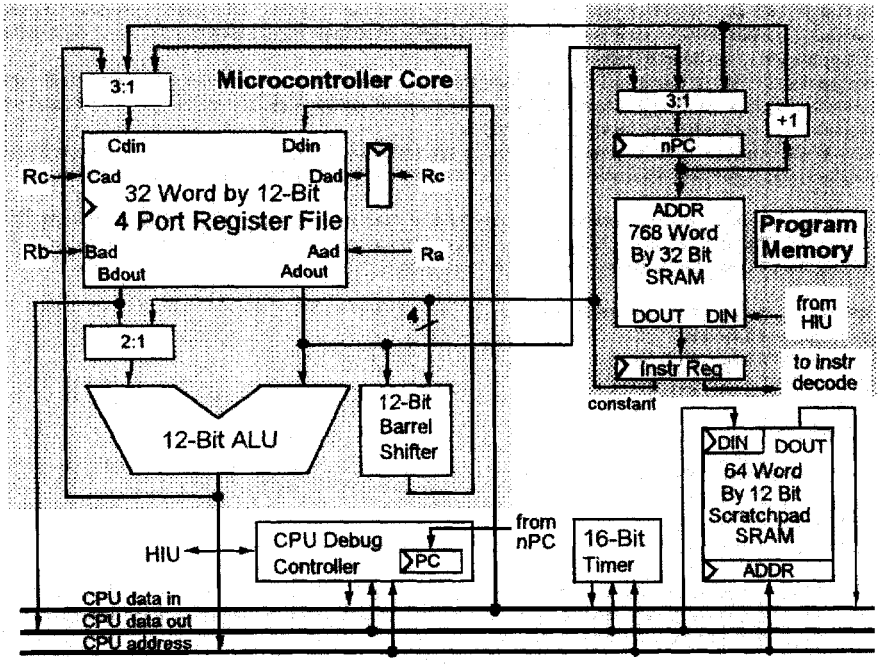
The host processor performs all of these activities by reading and writing locations in the external memory map described in Section 4.

The debug control registers described in Section 4 work in conjunction with the CPU debug controller shown in Figure 2.2. The debug control registers allow the host processor to reset, run, halt, or single step the CPU core and inspect and/or modify the contents of any register or memory which can be accessed by the CPU.

All host interface bus cycles supported by the HIU transfer 16-bit words synchronous to HCLK. Both burst and non-burst cycles are supported. Each non-burst read cycle by the host processor requires four HCLK cycles (HMODE pin = 0). Each non-burst write cycle requires two or three HCLK cycles (HMODE = 1 or 0, respectively).

Burst cycles transfer multiple words at the rate of one every one or two HCLK cycles (HMODE = 1 or 0, respectively). The host processor need only supply the address for the first word in a burst since the other addresses are generated internally by the HIU.

Figure 2.2 - CPU Block Diagram



All burst cycles other than burst reads with HMODE = 1 are terminated by the deassertion of the HBRST pin during the final word transfer and may be up to 2048 words long. A burst read cycle with HMODE = 1 can only be up to 257 words long and requires that its length be written beforehand into the BLENGTH register described in Section 4.

Video Interface Unit (VIU)

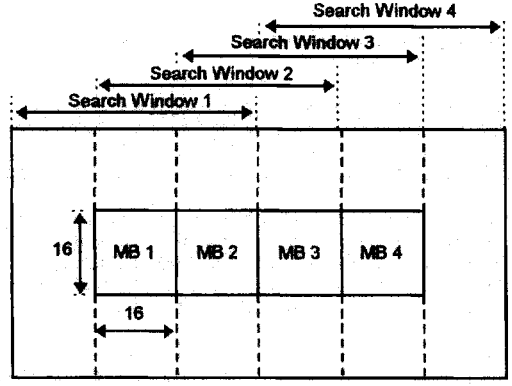
The VIU contains several elements. The 2 Kbyte Video Input Buffer (VIB) is loaded by the VIU from DRAMs connected to the KS0144 VBUS and supplies the Search Memory with reference image pels. The VIB consists of 128 pages, each page having 16 bytes. Each page logically corresponds to a two-row by eight-column array of reference image pels. The VIB pages can be partitioned into a maximum of four two-dimensional segments, each having up to 128 pels in either dimension. The CPU configures the VIB into segments using the registers described in Section 6.6.
using the registers described in Section 6.6.

The CPU writes to the LDVIB command register described in Section 6.3 to load each VIB segment with either full or quarter *resolution pels from all or part of a reference image search window.* Quarter resolution pels are computed by averaging 2x2 blocks in a simple 4:1 decimation filter and are used in the low resolution portion of hierarchical search algorithms. The VIU can simultaneously store both full and quarter-resolution pels from the same region of a reference image into two different VIB segments. *This allows the VIU to fetch both full and quarter-resolution versions of the same physical search window* without reading the same pels twice.

In a reference image, search windows surrounding adjacent 16x16 macroblocks overlap one another except for a 16 pel-wide strip as shown in Figure 2.3. Therefore, since the KS0143 normally sends input macroblocks to the KS0144 in a left-to-right raster-scan order, the CPU typically loads the VIB with only these non-redundant strips when moving the search window from one input macroblock to the next.

Once the CPU determines from the VIBRDY status bit (Section 6.4) that the VIB is full, it writes to the LDSM command register (Section 6.3) to copy VIB segments into Search Memory pages containing those search window pels which are no longer needed (i.e. those in the strip on the left side of the old search window). This copying occurs at a rate of 16 pels (one page) every PCLK clock cycle.

Figure 2.3 - Exampled of Overlapping Search Windows



A strip needed to update a search window for the next input macroblock can normally be fetched entirely while the block matching operation for the current macroblock is underway. Therefore, the contents of a search window can be updated in real-time without using high performance DRAMs.

Prior to using the VIU, the CPU must configure the video interface registers described in Section 6.5. The CPU must initialize VBUS timing parameters by writing to the ten VCNT registers, configure the VBUS data bus as either 16 or 32 bits by writing to the VBUSMODE register, and position the KS0144 on the VBUS arbitration daisy chain by writing to the VARB register.

By writing to the VREN, VRFPMS, and VRFPLS registers, the CPU can also configure the VIU to periodically perform CAS-before-RAS refresh on DRAMs connected to the KS0144 VBUS.

Search Memory

The 8 Kbyte Search Memory consists of 512 pages which can be partitioned by the CPU into a maximum of four segments. As with the VIB, each page corresponds to a two row by eight column array of pels. Each segment contains a single two dimensional search window having up to 128 pels in either dimension. The CPU configures the Search Memory segments using the registers described in Table 6.7.

Each Search Memory segment holds either full or quarter-resolution pels which were transferred from a corresponding VIB segment. The CPU loads the Search Memory from the VIB by writing to the LDSM command register as described in Section 6.3. Simple forward or backward predictive search can use up to two segments when performed hierarchically (i.e. one segment for the quarter resolution window, and another for the corresponding full resolution window). Therefore, MPEG-1 bidirectional prediction requires four segments when performed hierarchically: two for forward prediction and two for backward prediction.

The Search Memory outputs 16 pels (one page) every PCLK clock cycle which pass through the Interpolator unit on their way to the Block Matching Processor. Before starting block matching operations, the CPU can determine that VIB contents have been transferred to the Search Memory by testing the SMRDY status bit described in Section 6.4.

Interpolator

The Interpolator performs horizontal and/or vertical bilinear interpolation as defined in the MPEG-1 standard for half-pel resolution motion vectors. The Interpolator does not operate on 8x8 pel blocks and is only active when the CPU specifies fractional search coordinates to the Block Matching Processor. The Interpolator also cannot switch between performing horizontal or vertical interpolation and no interpolation at all in real-time. This places some minor restrictions on how the CPU specifies searches to the Block Matching Processor, as described in the following section.

Block Matching Processor (BMP)

The BMP is the principal number-cruncher inside the KS0144. The BMP contains multiple adder trees each of which computes and accumulates the sum of 16 absolute differences every PCLK clock cycle, attaining a total throughput of 7.2 GOPS at 50 MHz.

This computational power permits block matching to be performed on 16x16 blocks at a maximum rate of one every 120 nsec and on 8x8 blocks at a maximum rate of one every 33.3 nsec.

The BMP computes all block matches using the Mean Absolute Difference (MAD) metric in which two blocks are pel-wise subtracted and the magnitudes of all the differences added together. The BMP computes the MAD to a full 16 bits of precision on 16x16 blocks and automatically tracks both the value and location of the minimum MAD over multiple block matches. The CPU accesses the MAD by reading the MADX, MADY, MADMS, and MADLS registers described in Section 6.8.

Prior to starting a motion estimation search, the CPU initializes the MAD to "infinity" by writing to the INITMAD register described in Section 6.3. It also specifies the BMP operational mode by writing to the BMPMODE register described in Section 6.8. As shown in Figure 2.4, the BMP performs either 8x8 or 16x16 block matches in vertically aligned triplets. The block matches within a triplet can start on either successive or alternate lines. The search window location of a block match is referenced to the X and Y coordinates of the block's upper left pel. These latter coordinates are called search coordinates.

The CPU specifies the Search Memory segment (i.e. search window) to be used in block matching by writing to the BMSEG register described in Section 6.8. As shown in Figure 2.5, the CPU specifies search coordinates within the selected search window using a two dimensional coordinate system relative to a movable origin. The CPU specifies this origin using the BMXORG and BMYORG registers.

Search coordinates can be specified down to half-pel resolution and can horizontally wrap around a search window. That is, Search Memory addressing logic adds the horizontal (X) search coordinate to BMXORG and then interprets the sum modulo the horizontal dimension of the search window. Modulo addressing allows a search window to be easily addressed by the CPU once it has been updated by the 16 pel-wide strip logically to the right of the previous search window.

The CPU executes a motion estimation algorithm by specifying one or more search patterns to the BMP. A search pattern consists of a horizontal vector of triplets like the ten triplet vector shown in Figure 2.6. As described in Section 6.8, the CPU specifies the number of triplets in the SPCNT register, the horizontal spacing (possibly fractional) between adjacent triplets in the SPINC register, and the top-most search coordinates of the left-most triplet in the SPX and SPY registers.

Figure 2.4 - Block Matching Processor Operational Modes

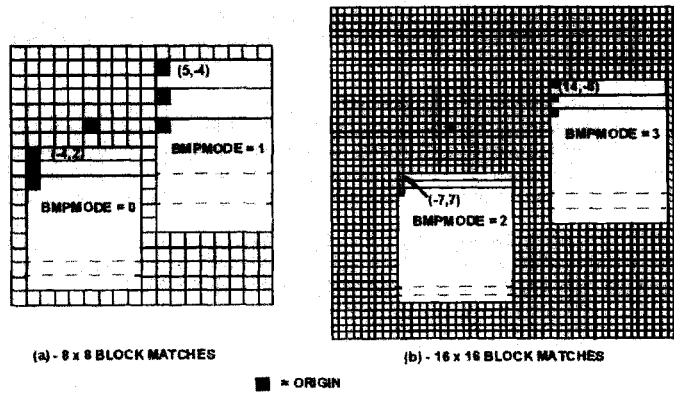


Figure 2.5 - Search Coordinate Example

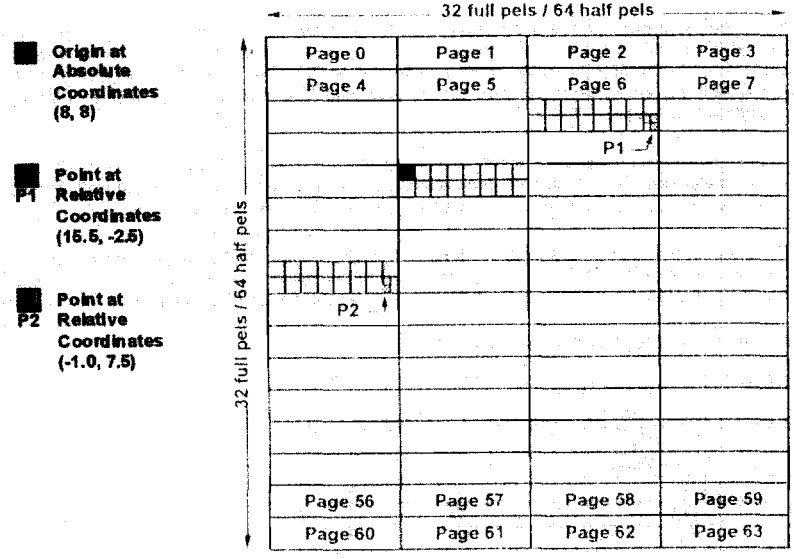
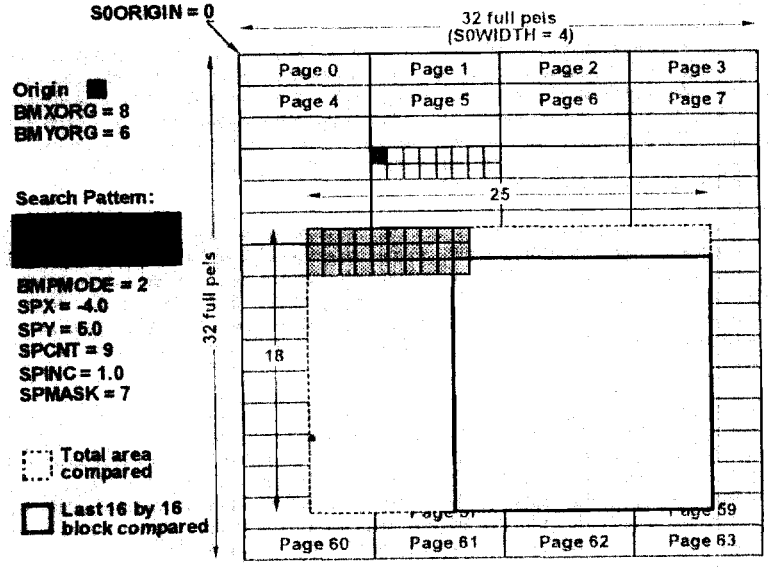


Figure 2.6 - Search Pattern Example



In addition, the CPU can mask out individual search coordinates within each triplet using the SPMASK register.

The SPCNT, SPINC, SPMASK, SPX, and SPY registers are all double buffered inside the BMP, meaning that the CPU can write to them again while the BMP is currently executing a search pattern. However, the CPU should not write to any of these registers until allowed to do so by the SPRDY status bit as described in Section 6.4. The CPU should always write to SPY last, since this signals the BMP that all the other registers contain valid data. The operational status of the BMP may also be checked by reading the BMPIDLE status bit.

The CPU is also not allowed to specify a search pattern in which SPY is an integer (i.e. non-fractional), SPINC is fractional, and SPCNT is greater than zero. Such a search pattern puts the interpolator into an illegal state by requiring it to dynamically switch between performing horizontal and/or vertical interpolation and no interpolation at all.

The time N in PCLK clock cycles required by the BMP to execute a single search pattern is given by N = T * (SPCNT + 1) + 5 + L, where T and L are defined in Table 2.2 as the BMP pipeline processing rate and pipeline latency, respectively. If two or more search patterns are executed successively, the latency factor L is only felt on the last pattern.

Table 2.2 - Block Matching Performance Parameters

BMPMODE	SPX or SPY fractional	T	L
00	not allowed	5	6
01	not allowed	6	7
10	no	18	6
10	yes	30	7
11	no	20	7
11	yes	30	7

Auxiliary Interface Unit (AIU)

The KS0143 dispatches instructions and operands to the KS0144 AIU over the XBUS. The AIU responds to any of the six instructions listed in Section 3 and sends execution results back to the KS0143 over the XBUS as required. These instructions allow the KS0143 to send prediction inputs to the KS0144, receive prediction results, and write original or reconstructed input images to the KS0144's DRAM.

The KS0143 uses an auxiliary processor write cycle to transmit an instruction and its operand (if any) to the KS0144 over the XBUS. The cycle begins with the transfer of two processor packet words (PP0 and PP1) which describe the instruction. The processor packet words are followed by zero, four, or 132 words of operand token data depending on the instruction and operand token type. One 24-bit word is transmitted on the XBUS every XCLK clock cycle. As discussed in Section 3, all instructions except OPREDICT require an operand. Operand control tokens require four words, while operand data tokens require 132 words. The processor packet and token transfer formats appear in Figure 2.8. The definitions of the token descriptor bytes in Figure 2.8 are shown in Figure 2.7.

The KS0144 AIU stores incoming processor packet and token descriptor data from IPREDICT and IPREDICT.S instructions in the IPREDICT registers as described in Section 6.9. Processor packet data for OPREDICT and S.OPREDICT instructions is stored in the OPREDICT registers as described in Section 6.10. Data blocks for IPREDICT and IPREDICT.S operand data tokens are stored in one of the Input Token Buffers, while operand data blocks and descriptors for OPREDICT and S.OPREDICT instructions are discarded. Processor packets and operand tokens for WRMEM and WRMEM.S instructions are not written to registers or memories accessible to the KS0144 CPU. Instead, the AIU stores them in the Video Write Buffer for access by the VIU.

After the KS0144 finishes executing an instruction, the KS0144 AIU requests an XBUS auxiliary processor read cycle from the KS0143 by asserting the XRQST# pin. The KS0143 then acknowledges the KS0144 by asserting the XADS# pin and broadcasting the KS0144's Processor ID (PID) on the XBUS. If the broadcasted PID matches the PID encoded on the KS0144's XID[1:0] pins, the KS0144 AIU continues the read cycle by transmitting a processor packet describing the instruction completing execution. The processor packet is then followed by zero to 132 words of result token data. As discussed in Section 3, all instructions except IPREDICT and WRMEM produce a result token. The processor packet and token transfer formats for auxiliary processor read cycles are the same as those previously described for write cycles.

The KS0144 AIU creates a processor packet for an auxiliary processor read cycle from the processor packet received when the instruction was sent from the KS0143. Result token descriptors for IPREDICT.S and WRMEM.S instructions are created from the operand token's token descriptors, while result token descriptors for OPREDICT and S.OPREDICT instructions are created from the OPREDICT registers described in Section 6.10.

Figure 2.7 - Token Descriptor Programmer's Mode

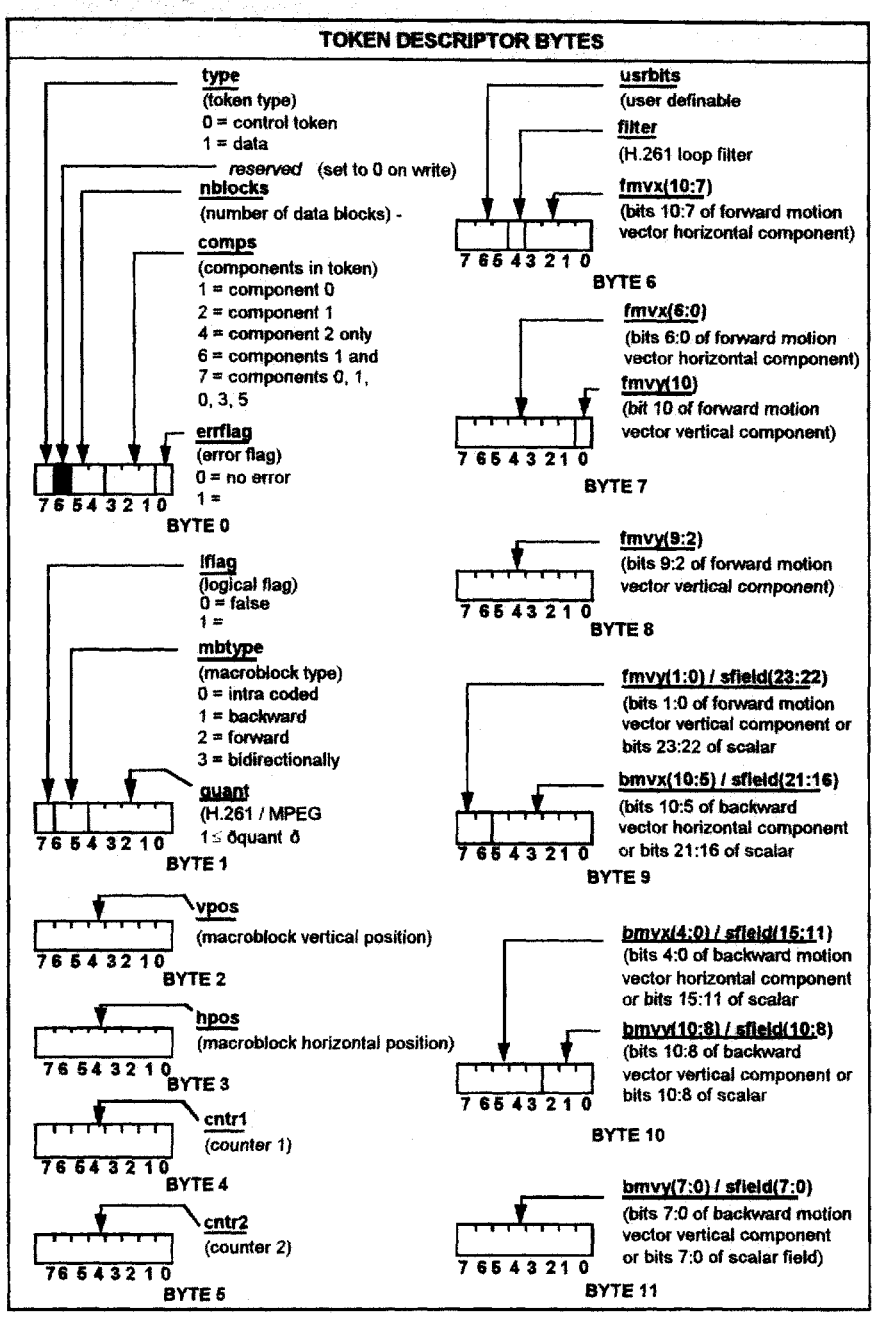
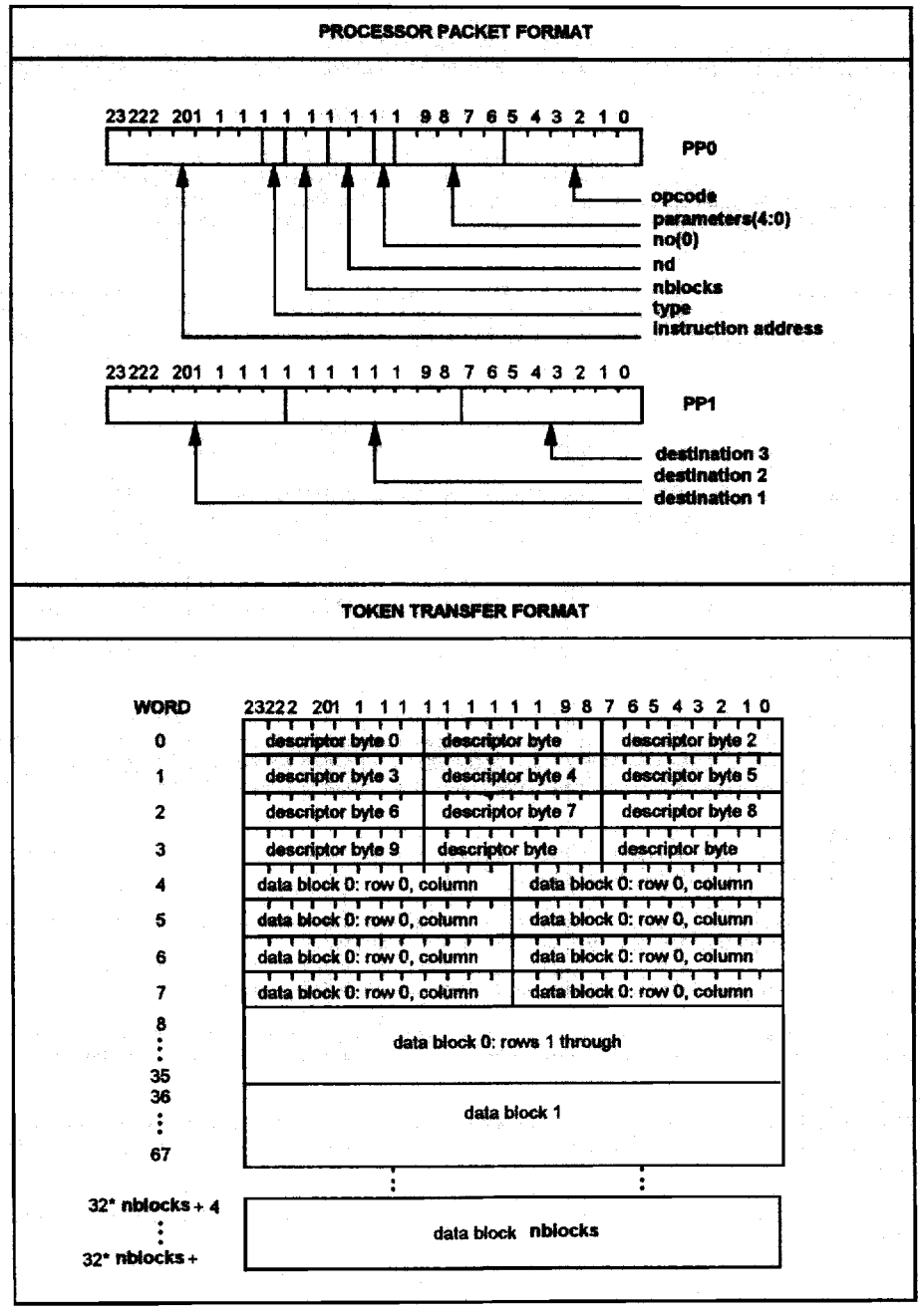


Figure 2.8 - AuxiallyProcissor Bus Data Formats



The KS0144 AIU reads data blocks for OPREDICT and S.OPREDICT result data tokens from the Output Token Buffer.

Input Token Buffers
The KS0143 sends a motion estimation input token to the KS0144 over the XBUS when the KS0143 executes an IPREDICT or IPREDICT.S instruction as discussed in Section 3. The KS0144 AIU receives this token into one of two Input Token Buffers (ITBs) which form a double buffered input interface with the KS0143. One ITB can receive an input token from the KS0144 AIU while the other supplies a previous input token to the BMP for use in block matching.

Each ITB holds 320 pels: 256 luminance pels from the IPREDICT or IPREDICT.S operand data token, plus another 64 pels which are computed by 4:1 decimation filtering the 256 pels as they are received from the KS0143. As with the VIB, the decimation filter used to compute quarter resolution pels is a simple 2 x 2 averager. The quarter resolution pels in an ITB are automatically output to the BMP during 8 x 8 block matches as indicated by BMPMODE, while the full resolution pels are output during 16 x 16 block matches.

The processor packet and token descriptor data in the IPREDICT registers of Section 6.9, always corresponds to the token in the ITB currently connected to the BMP. The CPU controls ITB usage via the SWAP command register described in Section 6.3. The CPU determines when the ITB connected to the KS0144 AIU has received a new input token by reading the ITBRDY status bit as described in Section 6.4.

Output Token Buffer (OTB)
Once a motion estimation search is completed, the CPU transfers the resultant 16x16 prediction from the Search Memory into the OTB. Once the prediction has been written to the OTB, the CPU is free to start another motion estimation search.

The CPU controls the loading of the OTB using the OTBSEG, OTBXORG, OTBYORG, OTBX, and OTBY registers described in Section 6.8. The CPU should write to OTBY last since this also commands the OTB to start loading. The CPU checks OTB loading status using the LDOTBRDY status bit described in Section 6.4. The output from the Search Memory passes through the interpolator on its way to the OTB so that half-pel block interpolation can be performed if required.
In addition to loading the OTB from the Search Memory, the CPU also must load the OPREDICT registers described in Section 6.10 with motion vector and other data which will be transmitted to the KS0143 as the output token's token descriptor.

The CPU typically copies much of this data from the IPREDICT registers (Section 6.9) which contain information from the input token's token descriptor. token's token descriptor.

The CPU then enables the KS0144 AIU to transfer the contents of the OTB and OPREDICT registers to the KS0143 as a four block data token. The CPU does this by writing to the SEND register described in Section 6.3. The AIU does not attempt the transfer until the KS0144 has received an OPREDICT or S.OPREDICT instruction from the KS0143 as described in Section 3. The CPU can check OTB transfer status using the OTBRDY and ORQST status bits described in Section 6.4.

Video Write Buffer
The KS0144 allows the KS0143 to write to reference images stored in DRAM connected to the KS0144 VBUS using the WRMEM and WRMEM.S instructions described in Section 3.

The operand of a WRMEM or WRMEM.S instruction is a four block data token typically computed by the luminance reconstruction portion of an KS0143 H.261 or MPEG-1 video encoder program.

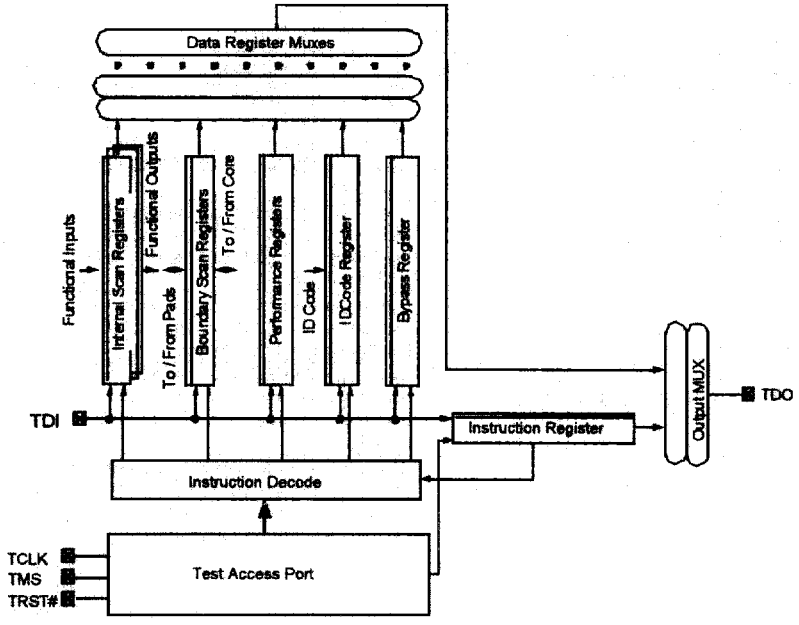
The KS0144 AIU loads this token from the XBUS into the 256 byte Video Write Buffer (VWB).

The KS0144 VIU then internally arbitrates VBUS usage between the VIB, VWB, and refresh function and then writes the contents of the VWB to DRAM. The KS0144 CPU plays no role in KS0144 execution of WRMEM and WRMEM.S instructions.

Test Control Unit (TCU)
A simplified block diagram of the KS0144 TCU, associated registers, and instruction set appear in Figure 2.9. The TCU supports boundary scan in compliance with the IEEE 1149.1 (JTAG) standard and also allows external logic to read and write the KS0144's internal scan paths.

For further details regarding the KS0144 boundary and internal scan paths, please contact the Array Microsystems technical applications support group.

Figure 2.9 - Test Control Unit (TCU)



Instruction Set

SAMPLE/PRELOAD	Load and sample boundary registers through I/O
EXTEST	Stimulate and observe I/O pins through serial test
INTEST	Stimulate and observe IC core through serial test port.
BYPASS	Allow quick bypass of serial data input to serial data
IDCODE	Observe internal hardwired ID code through serial test
PERFORMANCE	Enable and observe internal performance ring
INSCAN	Stimulate and observe internal data registers through serial test port (with
INSAMPLE	Stimulate and observe internal data registers through serial test port (without

SECTION 3 - KS0143 TO KS0144 COMMUNICATION INSTRUCTIONS

Table 3.1 lists the KS0143 instructions that pass data between the KS0143 and the KS0144 over the XBUS. Figures 3.1 through 3.6 are the bit maps for the instructions listed in the table.

Table 3.1 - KS0143 External Instructions For Communicating with the KS0144

Instruction Name	Operand Token	Result Token	Description	Parameters
IPREDICT	Data or Control	None	Sends control token with motion vector or data token with four 8x8 blocks of luminance pels to KS0144 Input Token Buffer for use in motion estimation.	None predefined.
IPREDICT.S	Data or Control	Control	Same as IPREDICT. In addition, it returns a control token upon completion. Output descriptor is copied from operand.	None predefined.
OPREDICT	None	Data or Control	Returns prediction result of motion estimation to KS0143 from the KS0144 Output Token Buffer.	None predefined.
S.OPREDICT	Data or Control	Data or Control	Same as OPREDICT. In addition, it takes an operand token to initiate the execution. The operand token data is ignored.	None predefined.
WRMEM	Data	None	Sends a token containing four 8x8 blocks of luminance pels to KS0144 Video Write Buffer for writing to external reconstruction memory.	memsel (memory select) - Contents are output on VMSEL pins. horgsel (horizontal origin select) - Multiplied by 128 and added to all DRAM column addresses computed by this instruction on the KS0144. vorgsel (vertical origin select) - Multiplied by 128 and added to all DRAM row addresses computed by this instruction on the KS0144.
WRMEM.S	Data	Control	Same as WRMEM. In addition, it returns a control token upon completion. Output descriptor is copied from operand descriptor.	

Figure 3.1 - IPREDICT Instructin Bit Map

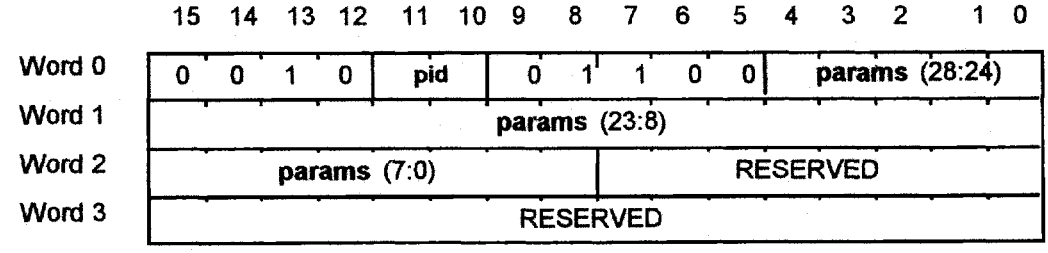


Figure 3.2 - IPREDICT .S Instruction Bit Map

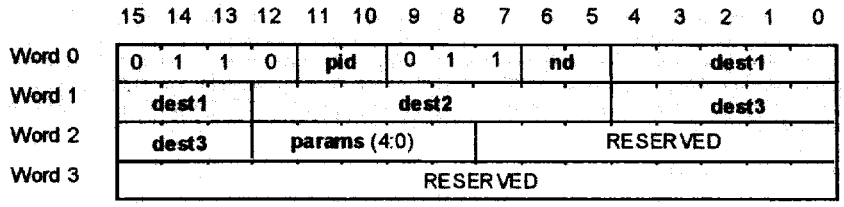


Figure 3.3 - OPREDICT Instruction Bir Map

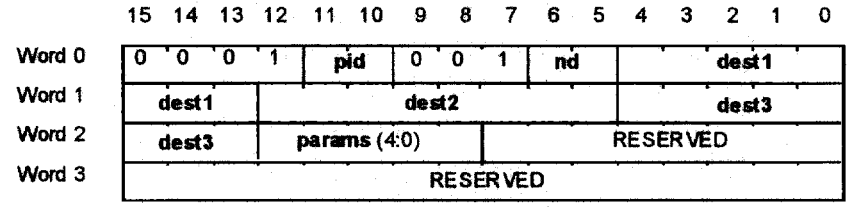


Figure 3.4 - S.OPREDICT Instruction Bit Map

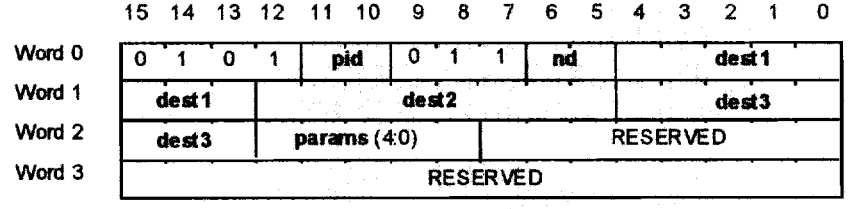


Figure 3.5 - WRMEM Instruction Bit Map

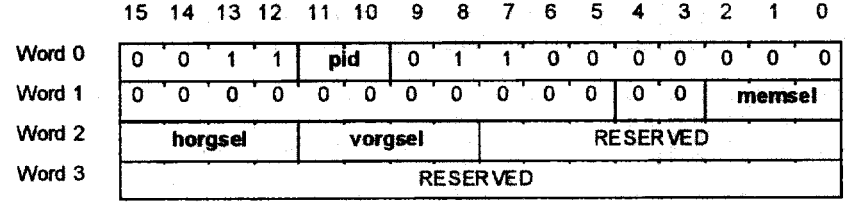
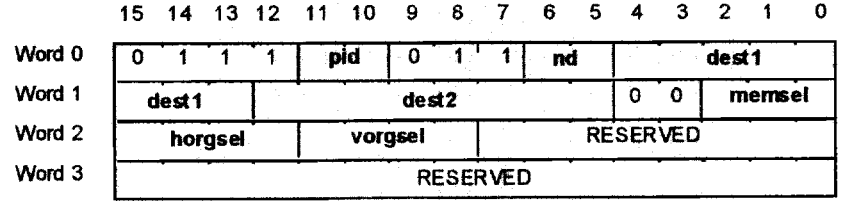


Figure 3.6 - WRMEM .S Instruction Bit Map



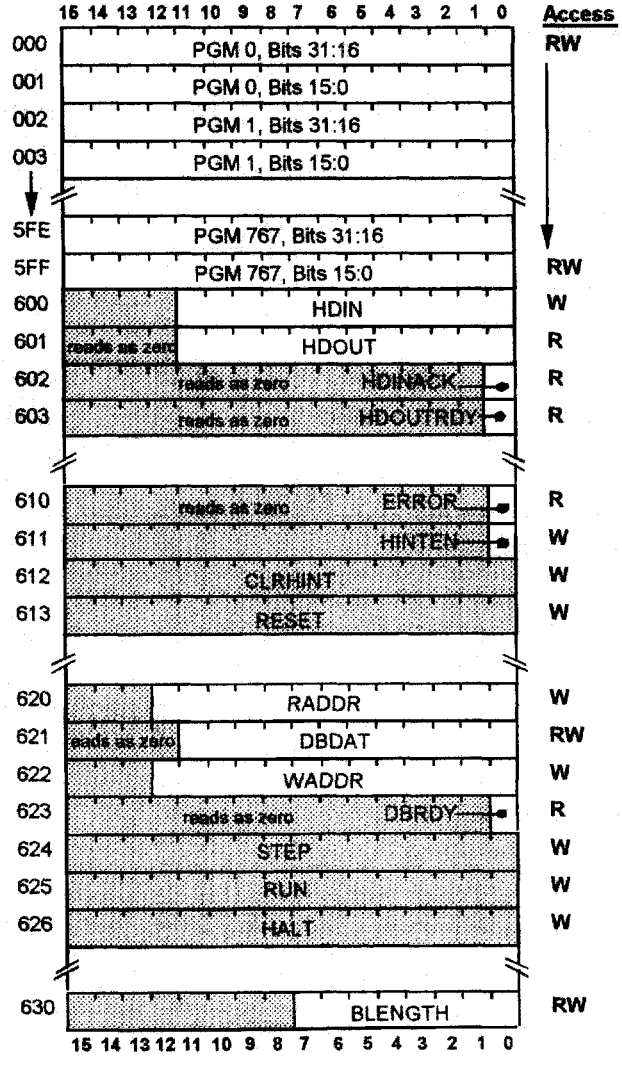
3-11

SAMSUNG ELECTRONICS

SECTION 4 - EXTERNAL HOST PROCESSOR MEMORY MAP

This section contains bit map diagrams and descriptions of KS0144 registers and memories which can be accessed by the external host processor. Unless indicated otherwise, shaded bits in the bit map diagrams have undefined contents. If these bits are written, they should be written with zeroes. Unless indicated otherwise, the Reset States shown in the description tables are assumed in response to either the assertion of the PRST# pin or the host processor writing to the RESET register. Any Reset State annotated with asterisks (**) is assumed only in response to the assertion of the PRST# pin. All addresses shown in this section are in hexadecimal.

External Host Processor Memory Map



External Host Processor Memory Map (Continued)

Register Name	Address	Reset State	Description
			CPU PROGRAM MEMORY
PGM (0) to PGM (767) (32 bits)	000 to 5FF (15:0)	none	PROGRAM MEMORY - contains the user KS0144 CPU program. Each 32-bit instruction word of the program is mapped onto two consecutive memory locations. The first program instruction must be loaded at the first program memory location (000). The host processor may read or write program memory during debug mode only when the DBRDY status bit is set to "1".
			HOST INTERFACES REGISTERS
HDIN (12 bits)	600 (11:0)	none	HOST DATA IN register - used by the host processor to pass data to the KS0144 CPU program. The host processor may place data in the HDIN register only when the HDINACK status bit is set to "1". The host processor may not read from, nor the KS0144 CPU write to, this register.
HDOUT (12 bits)	601 (11:0)	none	HOST DATA OUT register - Used by the KS0144 CPU program to pass data to the host processor. The host processor may read the contents of the HDOUT register only when the HDOUTRDY status bit is set to "1". The host processor may not write to, nor the KS0144 CPU read from, this register.
HDINACK (1 bit)	602 (0)	1	HOST DATA IN ACKNOWLEDGE status - indicates whether the KS0144 CPU program has read the data placed in the HDIN register by the host processor. •Set to "0" when the host processor writes data to the HDIN register •Set to "1" whenever the KS0144 CPU reads the contents of the HDIN register
HDOUTRDY (1 bit)	603 (0)	1	HOST DATA OUT READY status - indicates if HDOUT register is holding unread data written by the KS0144 CPU program. •Set to "1" when the KS0144 CPU writes to the HDOUT register •Set to "0" when the host processor reads the HDOUT register
ERROR (1 bit)	610 (0)	0	Error Status - indicates that one of the following error conditions exists in the KS0144 when set to a "1": ·Undefined external KS0143 instruction opcode (opcode other than those described in Section 3) ·IPREDICT, IPREDICT.S, WRMEM, or WRMEM.S instruction has a data token operand with less than four blocks ·IPREDICT, IPREDICT.S, WRMEM, or WRMEM.S instruction has no data token operand ·WRMEM or WRMEM.S instruction operand is a control token · OPREDICT or S.OPREDICT instruction has no destination
HINTEN (1 bit)	611 (0)	0	HOST INTERRUPT ENABLE bit - enables/disables host processor interrupts generated by the KS0144 through the HINT# output pin. The HINT# pin is asserted (low) in response to the KS0144 CPU writing to the HINT command register or in response to an interrupt caused by one of the hardware errors described in ERROR (above). The host processor enables interrupts by setting HINTEN to "1" and disables interrupts by setting HINTEN to "0"
CLRHINT (NA)	612 (NA)	NA	CLEAR HOST INTERRUPT register - clears interrupts generated by the KS0144. The KS0144 HINT# output pin remains asserted following a host interrupt generated by the KS0144 until it is cleared by the host writing any value to the CLRHINT register.

External Host Processor Memory Map (Continued)

Register Name	Address	Reset State	Description
			DEBUG CONTROL REGISTERS
RESET (NA)	613 (NA)	NA	KS0144 RESET register - resets the entire KS0144 when the host processor writes any value to this register. When reset, KS0144 CPU program execution is halted and all KS0144 registers assume initialized states (if any), and program execution jumps to location PGM[0].The debug controller enters the debug mode and the DBRDY status bit is set to "1".
RADDR (13 bits)	620 (12:0)	none	READ ADDRESS register - contains the address of the KS0144 register to be inspected during debug mode. The host processor may read the contents of the KS0144 CPU register file or a CPU address-mapped register by writing its address to RADDR. The data resulting from the register access is placed in the DBDAT register. The host processor may write to RADDR only when DBRDY bit ="1". The debug address mappings for the CPU register file and CPU memory-mapped registers are listed in the following table: **RADDR or WADDR** — **Selected CPU Register or Memory** 0 XXXX XXXX XXXX — Register or memory (see CPU Memory Maps in Section 6) 1 000X XXX0 0000 — R0 in register file 1 000X XXX0 0001 — R1 in register file 1 000X XXX0 0010 — R2 in register file • • • — • 1 000X XXX1 1111 — R31 in register file 1 001X XXXX XXXX — nPC (read only access) 1 010X XXXX XXXX — PC (read only access) 1 011X XXXX XXXX — Reserved • • • — • 1 111X XXXX XXXX — Reserved
DBDAT (12 bits)	621 (11:0)	none	DEBUG DATA register - contains data associated with the host processor write/read access using RADDR or WADDR in debug mode: · Read - KS0144 places the contents of the address found in RADDR into DBDAT. The host processor may read DBDAT only when the DBRDY status bit is set to "1". · Write - host processor places data in DBDAT that is to be written to the address found in WADDR. The host processor may write to DBDAT only when DBRDY status is "1" and prior to writing to WADDR.
WADDR (13 bits)	622 (12:0)	none	WRITE ADDRESS register - contains the address of an KS0144 register to be modified with data from DBDAT. The host processor may write to the CPU register file or a CPU memory-mapped register in debug mode by writing its address to the WADDR register. The host processor may write to WADDR only when DBRDY is set to a "1". The debug address mappings for the CPU register file and CPU memory-mapped registers are shown in the description of RADDR

External Host Processor Memory Map (Continued)

Register Name	Address	Reset State	Description
			DEBUG CONTROL REGISTERS
DBRDY (1 bit)	623 (0)	1**	DEBUG READY status - determines the state of the KS0144 CPU and debug controller. Used by the host processor to enable reading and writing of various registers during debug mode. Set to a "1" when the debug controller is in the debug mode and the KS0144 CPU is halted. Set to a "0" when the debug controller is in any other state and during CPU memory or register accesses initiated by the host writing to the RADDR or WADDR registers.
STEP (NA)	624 (NA)	NA	STEP register - the act of writing to this register by the host processor causes the KS0144 CPU to execute the next program instruction when the debug controller is in debug mode. The host processor may write to STEP only when DBRDY is set to a "1". DBRDY is set to a "0" during the execution of the instruction.
RUN (NA)	625 (NA)	NA	RUN register - the act of writing to this register by the host processor causes the debug controller to transition from debug to run mode (DBRDY is set to a "0"). The KS0144 CPU "free runs" executing a new instruction every two PCLK clock cycles.
HALT (NA)	626 (NA)	NA	HALT register - the act of writing to this register by the host processor causes the KS0144 CPU to halt program execution. The debug controller enters debug mode and DBRDY is set to "1". Program execution is now the result of subsequent writes to the RUN or STEP registers, continuing from the instruction following the last one executed during run mode.
			BURST MODE CONFIGURATION REGISTER
BLENGTH (8 bits)	630 (7:0)	0	BURST LENGTH register - contains a value that specifies the length of a mode 1 burst read cycle. The value loaded into this register by the host processor must be two less than the total number of words to be read. Hence, the minimum length of the mode 1 burst read is two words and the maximum length is 257. This register has no effect on mode 0 burst read and mode 0 /mode1 burst write cycles.

SECTION 5 - KS0144 CPU INSTRUCTION SET

The instruction set table in this section shows:
·assembler syntax for each KS0144 CPU instruction,
·bit layout of the two 16-bit words stored in instruction memory, and
·the operation that the instruction performs.
Note that R0 in the CPU register file always contains the value zero, regardless of what is written to it.
The alternating shaded and non-shaded areas denote hexadecimal number boundaries.

KS0144 CPU Instructions

ARITHMETIC INSTRUCTIONS		
Instruction	Even External Address Contents (15 … 0)	Odd External Address Contents (15 … 0)
add *Rc,Ra,Rb*	0 0 0 0 0 | *Ra* | *Rb* | *Rc* | 0 0 0 0 0 0 0 0 0 0 0	
	Rc = Ra + Rb	
addi *Rc,Ra,const*	0 0 0 0 1 | *Ra* | 0 0 0 0 0 | *Rc* | *const*	
	Rc = Ra + const	
sub *Rc,Ra,Rb*	0 0 0 1 0 | *Ra* | *Rb* | *Rc* | 0 0 0 0 0 0 0 0 0 0 0	
	Rc = Ra - Rb	
subi *Rc,Ra,const*	0 0 0 1 1 | *Ra* | 0 0 0 0 0 | *Rc* | *const*	
	Rc = Ra - const	
and *Rc,Ra,Rb*	0 0 1 0 0 | *Ra* | *Rb* | *Rc* | 0 0 0 0 0 0 0 0 0 0 0	
	Rc = Ra & Rb	
andi *Rc,Ra,const*	0 0 1 0 1 | *Ra* | 0 0 0 0 0 | *Rc* | *const*	
	Rc = Ra & const	
xor *Rc,Ra,Rb*	0 0 1 1 0 | *Ra* | *Rb* | *Rc* | 0 0 0 0 0 0 0 0 0 0 0	
	Rc = Ra ^ Rb	
xori *Rc,Ra,const*	0 0 1 1 1 | *Ra* | 0 0 0 0 0 | *Rc* | *const*	
	Rc = Ra ^ const	
or *Rc,Ra,Rb*	0 1 0 0 0 | *Ra* | *Rb* | *Rc* | 0 0 0 0 0 0 0 0 0 0 0	
	Rc = Ra | Rb	
ori *Rc,Ra,const*	0 1 0 0 1 | *Ra* | 0 0 0 0 0 | *Rc* | *const*	
	Rc = Ra | Rb	

KS0144 CPU Instructions (continued)

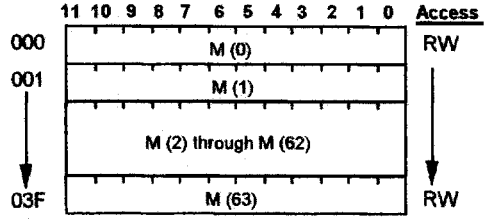
Instruction	Even External Address Contents (15 … 0)	Odd External Address Contents (15 … 0)
`nor` *Rc,Ra,Rb*	0 1 0 1 0 | *Ra* | *Rb* | *Rc* | 0 0 0 0 0 0 0 0 0 0 0	
	Rc = ~ (*Ra* | *Rb*)	
`nori` *Rc,Ra,const*	0 1 0 1 1 | *Ra* | 0 0 0 0 0 | *Rc* | *const*	
	Rc = ~ (*Ra* | *const*)	
`sra` *Rc Ra shift*	0 1 1 0 0 | *Ra* | 0 0 0 0 0 | *Rc* | 0 0 0 0 0 0 0 0 | *shift*	
	Rc = *Ra* >> *shift* (arithmetic right shift)	
`srl` *Rc,Ra,shift*	0 1 1 0 1 | *Ra* | 0 0 0 0 0 | *Rc* | 0 0 0 0 0 0 0 0 | *shift*	
	Rc = *Ra* >> *shift* (logical right shift)	
`sll` *Rc,Ra,shift*	0 1 1 1 0 | *Ra* | 0 0 0 0 0 | *Rc* | 0 0 0 0 0 0 0 0 | *shift*	
	Rc = *Ra* << *shift* (logical left shift)	
BRANCH AND JUMP INSTRUCTIONS		
`beqal` *Ra,Rb,addr Rl*	1 0 0 0 0 | *Ra* | *Rb* | *Rl* | *addr*	
	Rl = *nPC* + *1*; `if` (*Ra*==*Rb*) `goto` *addr*	
`bneal` *Ra,Rb,addr,Rl*	1 0 0 0 1 | *Ra* | *Rb* | *Rl* | *addr*	
	Rl = *nPC* + *1*; `if` (*Ra*! = *Rb*) `goto` *addr*	
`bgtal` *Ra,Rb,addr,Rl*	1 0 0 1 1 | *Ra* | *Rb* | *Rl* | *addr*	
	Rl = *nPC*+*1*; `if` (*Ra*>*Rb*) `goto` *addr*	
`bgeal` *Ra,Rb,addr Rl*	1 0 1 0 1 | *Ra* | *Rb* | *Rl* | *addr*	
	Rl = *nPC*+*1*; `if` (*Ra*>=*Rb*) `goto` *addr*	
`jalr` (*Ra*) *Rl*	1 0 1 1 0 | *Ra* | 0 0 0 0 0 | *Rl* | 0 0 0 0 0 0 0 0 0 0 0	
	Rl = *nPC*+*1*; `goto` (*Ra*)	
LOAD AND STORE INSTRUCTIONS		
`st` (*Ra*+*offset*),*Rb*	1 1 0 0 1 | *Ra* | *Rb* | 0 0 0 0 0 | *offset*	
	*(*Ra*+*offset*)= *Rb*	
`ld` *Rd,* (*Ra*+*offset*)	1 1 0 1 1 | *Ra* | 0 0 0 0 0 | *Rd* | *offset*	
	Rd = *(*Ra*+*offset*) (result not available for next instruction)	

NOTE: All branch and jump instructions execute the next instruction in the pipeline (the "delay" slot instruction) regardless of whether the branch or jump is taken.

SECTION 6 - KS0144 CPU MEMORY MAP

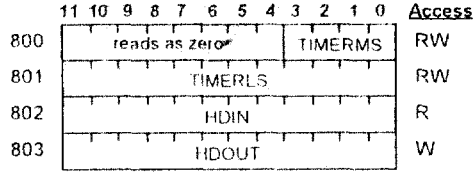
This section contains bit map diagrams and descriptions of KS0144 registers and memories which can be accessed by the KS0144 CPU using load and store instructions. Unless indicated otherwise, shaded bits in the bit map diagrams have undefined contents. If these bits are written, they should be written with zeroes. Unless indicated otherwise, the Reset States shown in the description tables are assumed in response to either the assertion of the PRST# pin or the host processor writing to the RESET register. Any Reset State annotated with asterisks (**) is assumed only in response to the assertion of the PRST# pin. All addresses noted in this section are in hexadecimal.

6.1 - CPU Scratchpad Memory



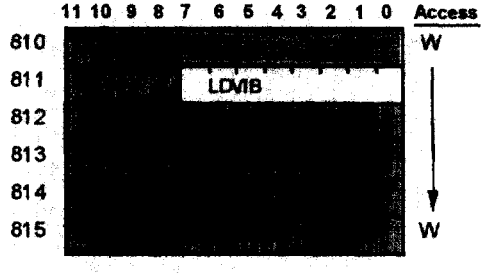
Register Name	Address	Reset State	Description
M (0) to M (63) (12 bits)	000 to 03F	none	This 64 word by 12-bit memory may be used as needed by the MEC CPU program

6.2 - CPU Timer and Host Interface Data Registers



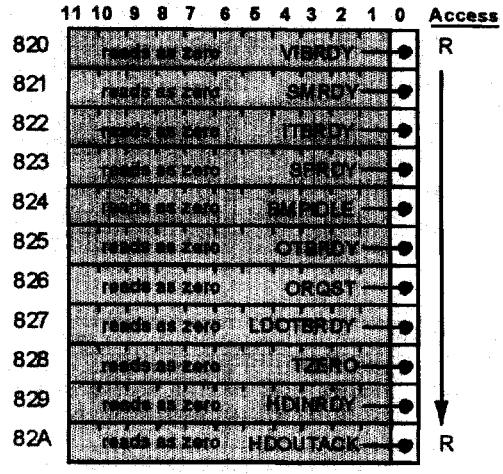
Register Name	Address	Reset State	Description
TIMERMS (4 bits)	800	0	Most significant four bits of the general purpose16-bit CPU timer. The CPU must initialize the timer count before use. After that, the timer count may be read at any time by the CPU without affecting its operation.
TIMERLS (12 bits)	801	0	Least significant 12 bits of the general purpose 16-bit CPU timer. The CPU must initialize the timer count before use. After that, the timer count may be read at any time by the CPU without affecting its operation. Writing to this register causes the timer to decrement once on every other PCLK cycle(stopping when it reaches zero)
HDIN (12 bits)	802	none	Used to pass data from the host processor to the KS0144 CPU. Data written to this register by the host processor may be read by the KS0144 CPU. The KS0144 CPU cannot write to this register, nor can the host processor read from it.
HDOUT (12 bits)	803	none	Used to pass data from the KS0144 CPU to the host processor. Data written to this register by the KS0144 CPU may be read by the host processor. The host processor cannot write to this register, nor can the KS0144 CPU read from it.

6-3 CPU Command Registers



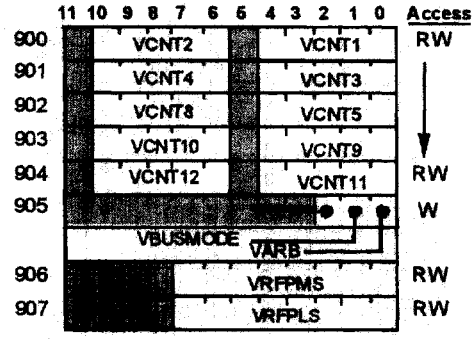
Register Name	Address	Reset State	Description
INITMAD (NA)	810	none	Initialize MAD. Erases any previously stored minimum MAD from the MAD tracking registers inside the block matching processor by setting the minimum MAD to infinity. The BMPIDLE status bit must be a "1" before this register is written.
LDVIB (8 bits)	811	00	Load video input buffer. Loads the video input buffer memory with pels from DRAM connected to the KS0144 video bus. The VIBRDY status bit must be "1" before this register is written. Each pair of bits in the LDVIB register controls loading of a different VIB segment; bits 1:0, 3:2, 5:4, and 7:6 control loading of segments 0, 1, 2, and 3 respectively. Each pair of bits contains one of four possible LDVIB commands defined as follows: COMMAND DEFINITION 0 No operation (i.e. do not load segment) 1 Load selected segment with full resolution pels 2 Load selected segment with quarter resolution pels 3 Load selected segment with full resolution pels and segment following with quarter resolution pels (applicable only to segments 0 and 2)
LDSM (NA)	812	none	Load search memory. Copies the contents of the video input buffer into the search memory after waiting for the VIBRDY and LDOTBRDY status bits to be "1". The SMRDY status bit must be a "1" before this register is written.
SWAP (NA)	813	none	Swap input token buffers. Toggles the state of the double-buffered memory used to receive prediction input tokens from the KS0143 IPREDICT and IPREDICT.S instructions. The ITBRDY status bit must be a "1" before this register is written.
SEND (NA)	814	none	Send output token to KS0143. Transmits the contents of the output token buffer to the KS0143 after waiting for the ORQST and LDOTBRDY status bits to be "1". The OTBRDY status bit must be a "1" before this register is written. Used by the KS0144 CPU to send prediction results back to the KS0143 in response to KS0143 OPREDICT and S.OPREDICT instructions.
HINT (NA)	815	none	Host processor interrupt. Sets the HINT# pin to the active (low) state. HINT# remains active until the host processor writes to the CLRHINT register.

6.4 - CPU Status Bits



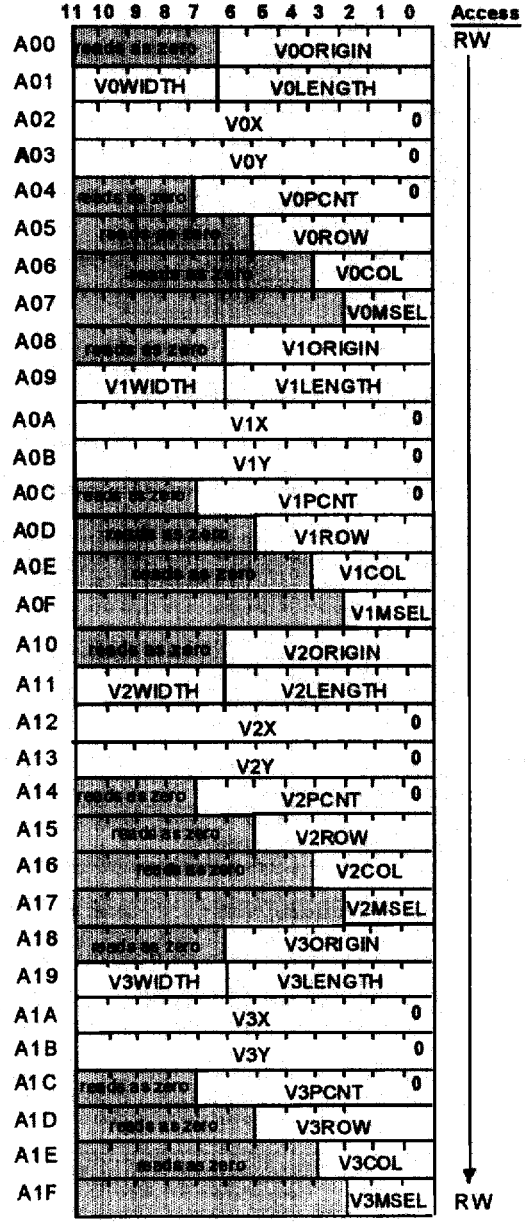
Register Name	Address	Reset State	Description
VIBRDY (1 bit)	820	1	Video input buffer ready. Set to a "0" when LDVIB command register is written. Set to a "1" after the video input buffer has been loaded from DRAM.
SMRDY (1 bit)	821	1	Search memory ready. Set to a "0" when LDSM command register is written. Set to a "1" after the search memory has been loaded from the video input buffer.
ITBRDY (1 bit)	822	0	Input token buffer ready. Set to a "0" when the SWAP command register is written. Set to a "1" after the input token buffer has received a new token from the KS0143.
SPRDY (1 bit)	823	1	Search pattern ready. Set to a "0" when the SPY register is written. Set to a "1" when the search pattern configuration registers can be written again.
BMPIDLE (1 bit)	824	1	Block matching processor idle. Set to a "1" when both the SPRDY status is a "1" and the block matching processor is not busy. At all other times BMPIDLE is set to a "0".
OTBRDY (1 bit)	825	1	Output token buffer ready. Set to a "0" when the SEND command register is written. Set to a "1" after the KS0144 has sent the contents of the output token buffer to the KS0143.
ORQST (1 bit)	826	0	Output token request. Set to a "1" after the KS0144 has received an OPREDICT or S.OPREDICT instruction from the KS0143. Set to a "0" after the prediction result has been sent to the KS0143.
LDOTBRDY (1 bit)	827	1	Output token buffer load ready. Set to a 000 when the OTBY register is written. Set to a "1" after the output token buffer has been loaded from the search memory.
TZERO (1 bit)	828	1	Timer zero. Set to a "0" when the TIMERLS register is written (i.e. when timer is started). Set to a "1" after the timer has decremented to zero.
HDINRDY (1 bit)	829	0	Host data input ready. Set to a "1' when the host processor writes to the HDIN register. Set to a "0" after the KS0144 CPU reads the HDIN register.
HDOUTACK (1 bit)	82A	1	Host output data acknowledge. Set to a "0" when the KS0144 CPU writes to the HDOUT register. Set to a "1" after the host processor reads the HDOUT register.

6.5 - CPU Video Interface and Timing Control Registers

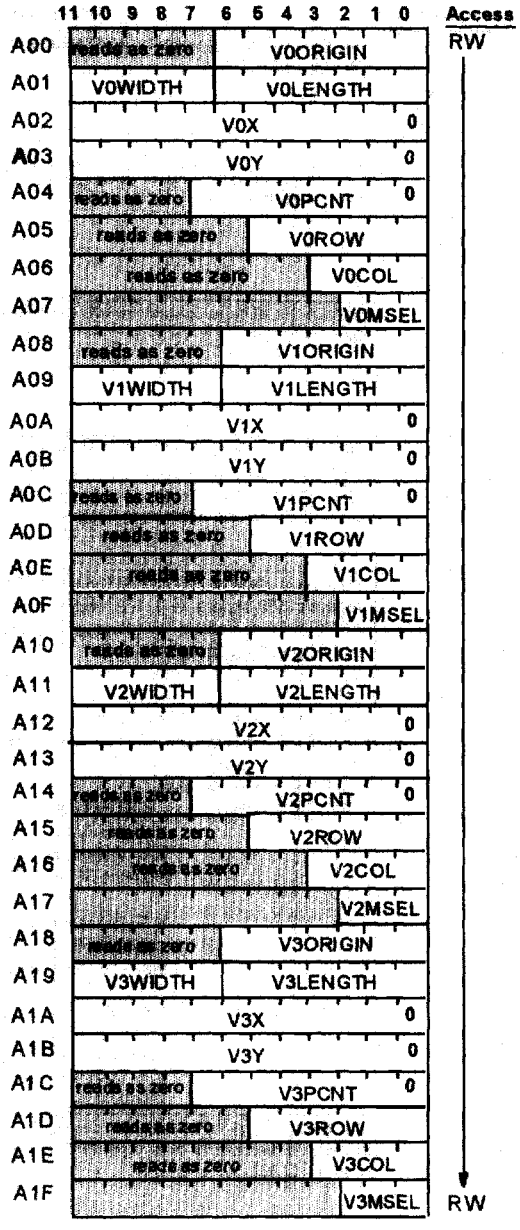


Register Name	Address	Reset State	Description
VCNT1 (5 bits)	900	none	Number of VCLK periods between row address assertion on VA pins and assertion of VRAS# pins.
VCNT2 (5 bits)	900	none	Number of VCLK periods between row address assertion and column address assertion on VA pins.
VCNT3 (5 bits)	901	none	Number of VCLK periods between VRAS# assertion and VCASO# assertion.
VCNT4 (5 bits)	901	none	Number of VCLK periods between successive assertions of VCASO# pins during page mode memory cycles (i.e. the CAS# page mode cycle time).
VCNT5 (5 bits)	902	none	Minimum number of VCLK periods VRAS# is precharged at the end of a memory cycle (i.e. the RAS# precharge is held two additional VCLK cycles between row accesses).
VCNT8 (5 bits)	902	none	Number of VCLK periods VCASO# is asserted during each page mode data transfer (i.e. the CAS# pulse width).
VCNT9 (5 bits)	903	none	Number of VCLK periods between the assertion of VCASO# pins and the deassertion of the VRAS# pins during the final transfer of a page mode memory cycle.
VCNT10 (5 bits)	903	none	Number of VCLK periods between VCASO# assertion and VRAS# assertion during CAS-before-RAS refresh.
VCNT11 (5 bits)	904	none	Number of VCLK periods between VRAS# assertion and deassertion of VCASO# during CAS-before-RAS refresh.
VCNT12 (5 bits)	904	none	Number of VCLK periods VRAS# is asserted during CAS-before-RAS refresh.
VREN (1 bit)	905	O**	Video refresh enable. Writing a "1" causes the KS0144 to perform a CAS-before-RAS refresh cycle on the KS0144 video bus and perform further refresh cycles periodically as determined by the refresh counter. Writing a "0" stops the counter and inhibits further refreshes.
VBUSMODE (1 bit)	905	O**	Video bus mode. Writing a "1" causes the KS0144 video bus to transfer 16 bits with each DRAM access. Writing a "0" causes the KS0144 video bus to transfer 32 bits with each access.
VARB (1 bit)	906	O**	Video bus master arbiter select. Writing a "1" configures the KS0144 as the highest priority bus master on the video bus. Writing a "0" configures the KS0144 as a lower priority master as determined by its position on the video bus arbitration daisy chain.
VRFPMS (8 bits)	906	none	The most significant 8 bits of the 16-bit DRAM refresh period. The refresh period is the number of VCLK periods the KS0144 waits between performing successive CAS-before-RAS refresh cycles and needs to be loaded prior to setting VREN to a "1".
VRFPLS (8 bits)	907	none	The least significant 8 bits of the 16-bit DRAM refresh period.

6.6 - Video Input Buffer (VIB) Configuration Registers



6.6 - Video Input Buffer (VIB) Configuration Registers

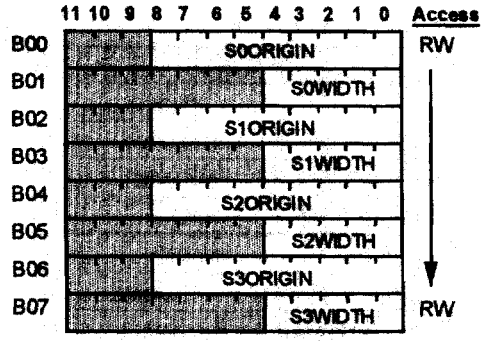


6.6 - Video Input Buffer (VIB) Configuration Registers (Continued)Search Memory Configura

Register Name	Address	Reset State	Description
V0ORIGIN V1ORIGIN V2ORIGIN V3ORIGIN (7 bits)	A00 A08 A10 A18	none	The starting VIB page addresses of segments 0, 1, 2, and 3 respectively. Range = [0,127]
V0LENGTH V1LENGTH V2LENGTH V3LENGTH (7 bits)	A01 A09 A11 A19	none	The number of page rows in VIB segments 0, 1, 2, and 3 respectively. Range = [1, 64]
V0WIDTH V1WIDTH V2WIDTH V3WIDTH (5 bits)	A01 A09 A11 A19	none	The number of page columns in VIB segments 0, 1, 2, and 3 respectively. Range = [1, 16]
V0X V1X V2X V3X (12 bits)	A02 A0A A12 A1A	none	The logical starting column (X) addresses in image memory of the upper left corners of VIB segments 0,1,2, and 3 respectively. Bit 0 of V0X, V1X, V2X, or V3X is hardwired to ÒOÓ forcing all fetches to start on even numbered columns of the image memory.
V0Y V1Y V2Y V3Y (12 bits)	A03 A0B A13 A1B	none	The logical starting row (Y) addresses in image memory of the upper left corners of VIB segments 0, 1, 2, and 3 respectively. Bit 0 of V0Y, V1Y, V2Y, or V3Y is hardwired to ÒOÓ forcing all fetches to start on even numbered rows of the image memory.
V0PCNT V1PCNT V2PCNT V3PCNT (8 bits)	A04 A0C A14 A1C	none	Determines the number of pels to be read from each row of image memory when filling VIB segments 0, 1, 2, and 3 respectively. Bit 0 of VkPCNT (for k = 0, 1, 2, and 3) is hardwired to "0" forcing an even number of pels to be fetched from each row. **LDVIB COMMAND** — **DEFINITION** 1 — VkPCNT is the number of pels stored in each row of segment k and must be less than or equal to 8* VkWIDTH 2 — VkPCNT is twice the number of pels stored in each row of segment k and must be less than or equal to 16* VkWIDTH 3 — VkPCNT is the number of pels stored in each row of segment k and is twice the number of pels stored in each row of segment n, where n = k+1. VkPCNT must be less than or equal to both 8*VkWIDTH and 16*VnWIDTH.
V0ROW V1ROW V2ROW V3ROW (6 bits)	A05 A0D A15 A1D	none	The starting page row address in segments 0, 1, 2, and 3 (respectively) of the search memory for pels copied from corresponding VIB segments. Range = [0,63]

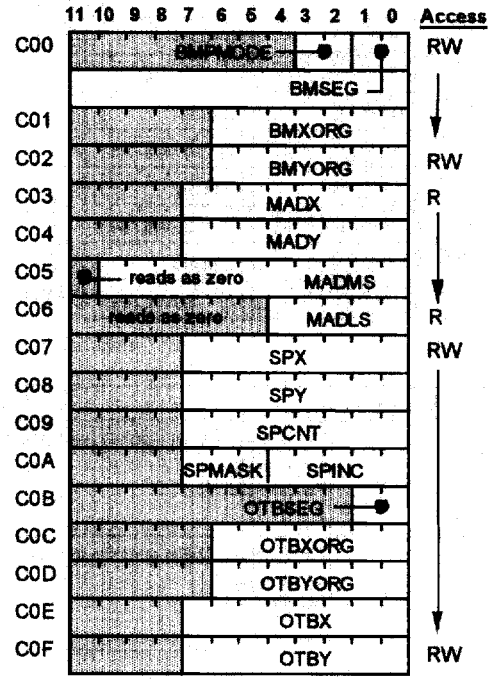
SAMSUNG ELECTRONICS

6.7 - Search Memory Configuration Register



Register Name	Address	Reset State	Description
S0ORIGIN S1ORIGIN S2ORIGIN S3ORIGIN (9 bits)	B00 B02 B04 B06	none	The starting search memory page addresses of segments 0, 1, 2, and 3 respectively. Range = [0,511]
S0WIDTH S1WIDTH S2WIDTH S3WIDTH (5 bits)	B01 B03 B05 B07	none	The number of page columns in search memory segments 0,1,2, and 3 respectively. Range = [1,16]

6.8 - Block Matching Processor Control and MAD Registers



NOTE:
In the following register descriptions,
MAD = Mean Absolute Difference
BMP = Block Matching Processor

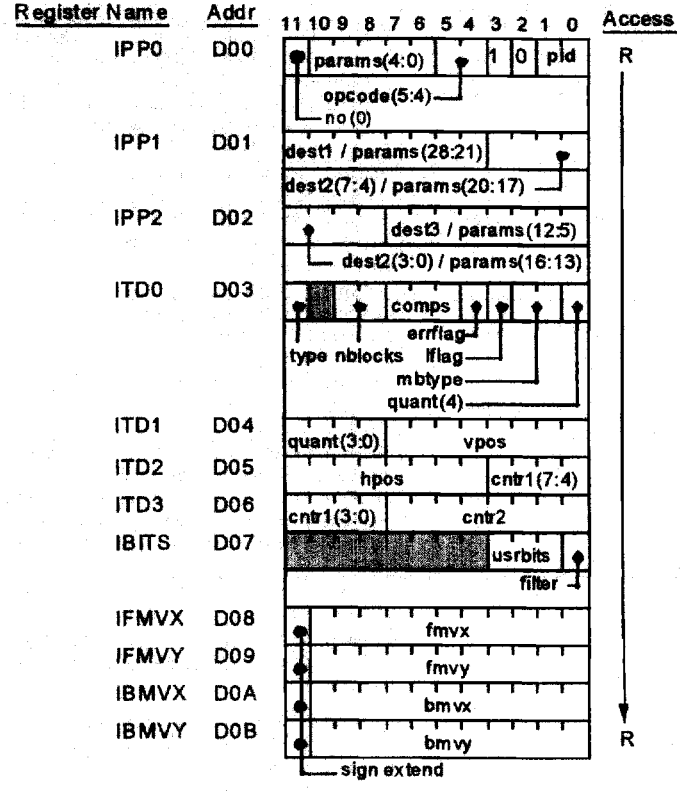
Register Name	Address	Reset State	Description
BMSEG(2 bits)	C00	none	Search memory segment (0, 1, 2, or 3) to be searched.
BMPMODE (2 bits)	C00	none	Block Matching Processor operational mode: MODE DEFINITION 00 Match 8x8 blocks on adjacent rows 01 Match 8x8 blocks on alternate rows 10 Match 16x16 blocks on adjacent rows 11 Match 16x16 blocks on alternate rows
BMXORG BMYORG (7 bits)	C01 C02	none	Absolute X and Y coordinates (respectively) of the search origin in the search memory segment selected by register BMSEG. Range = [0, 127]
MADX MADY (8 bits)	C03 C04	none	Relative X and Y coordinates (respectively) of the current minimum MAD found by the BMP. May be read at any time by the KS0144 CPU, even when the BMP is running. Range = [-64.0, 63.5]
MADMS (11 bits)	C05	7FF	Bits 15:5 of the current minimum MAD value found by the BMP. May be read at any time by the KS0144 CPU, even when the BMP is running. 16-bit MAD range = [0, 65535]

SAMSUNG ELECTRONICS

6.8 - Block Matching Processor Control and MAD Registers (Continued)

Register Name	Address	Reset State	Description
MADLS (5 bits)	CO6	1F	Bits 4:0 of the current minimum MAD value found by the BMP. May be read at any time by the KS0144 CPU, even when the BMP is running. 16-bit MAD range = [0, 65535]
SPX (8 bits)	CO7	none	Relative X coordinate of the first search coordinate triplet in the vector of triplets to be processed by the BMP. Range = [-64.0, 63.5]
SPY (8 bits)	CO8	none	Relative Y coordinate of the first search coordinate triplet in the vector of triplets to be processed by the BMP and indicates to the BMP that the SPX, SPY, SPCNT, SPINC, and SPMASK registers are valid. This sets the BMPIDLE register to ÒOOÓ, and queues block match processor request. Range = [-64.0, 63.5]
SPCNT (8 bits)	CO9	none	One less than the total number of triplets in the vector of triplets to be processed by the BMP. Range = [0, 255]
SPINC (5 bits)	COA	none	Horizontal (X) spacing between triplets in the vector of triplets to be processed by the BMP. Range = [0.5, 15.5]
SPMASK (3 bits)	COA	none	Binary mask which selects from one to three MADs from each triplet for use in calculation of the minimum MAD by the BMP. Whichever SPMASK bit equals a 1, the MAD from that corresponding search coordinate of each triplet is factored into the minimum MAD calculation. Note that bit 0 corresponds to the topmost coordinate in a triplet.
OTBSEG (2 bits)	COB	none	Search memory segment (0,1,2, or 3) whose contents are to be copied to the Output Token Buffer.
OTBXORG OTBYORG (7 bits)	COC COF	none	Absolute X and Y coordinates (respectively) of the origin of the search memory segment selected by the OTBSEG register. Range = [0,127]
OTBX (8 bits)	COE	none	Relative X coordinate of the16x16 block to be copied into the Output Token Buffer from the search memory segment selected by the OTBSEG register. Range = [-64.0, 63.5]
OTBY (8 bits)	COF	none	Relative Y coordinates of the 16x16 block to be copied into the Output Token Buffer from the search memory segment selected by the OTBSEG register and indicates to the BMP that the OTBSEG, OTBXORG, OTBYORG, OTBX, and OTBY registers are valid. This sets the LDOTBRDY register to ÒOOÓ and transfers the specified block to the output token buffer. Range = [-64.0, 63.5]

6.9 - IPREDICT Registers

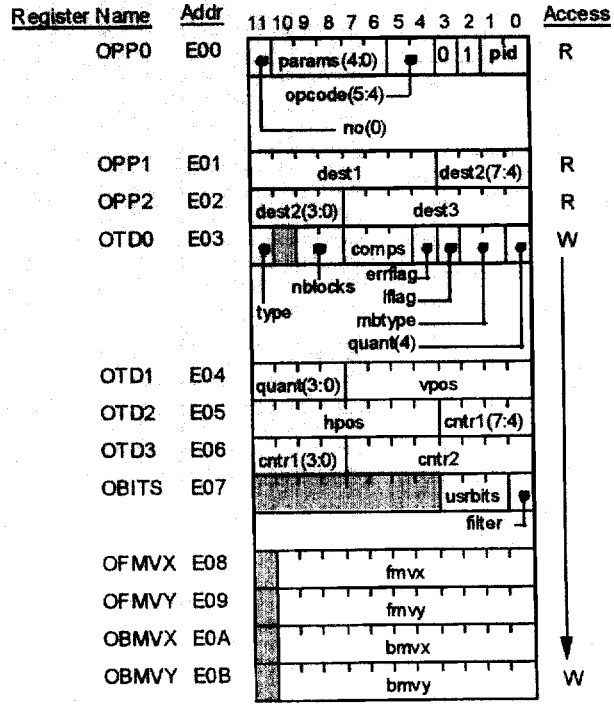


Register Name	Address	Reset State	Description
IPP0 (12 bits)	DO0	none	Bits 11:0 from word 0 of the IPREDICT or IPREDICT.S instruction processor packet; contains instruction params(4:0), no(0), opcode, and pid fields.
IPP1 (12 bits)	DO1	none	Bits 23:12 from word 1 of the IPREDICT or IPREDICT.S instruction processor packet; contains instruction dest1 field and the most significant nibble of dest2 field.
IPP2 (12 bits)	DO2	none	Bits 11:0 from word 1 of the IPREDICT or IPREDICT.S instruction processor packet; contains instruction dest3 field and the least significant nibble of dest2 field.
ITD0 (12 bits)	DO3	none	Contains byte 0 and the most significant nibble of byte 1 from the token descriptor of the IPREDICT or IPREDICT.S instruction operand token.
ITD1 (12 bits)	DO4	none	Contains byte 2 and the least significant nibble of byte 1 from the token descriptor of the IPREDICT or IPREDICT.S instruction operand token.
ITD2 (12 bits)	DO5	none	Contains byte 3 and the most significant nibble of byte 4 from the token descriptor of the IPREDICT or IPREDICT.S instruction operand token.
ITD3 (12 bits)	DO6	none	Contains byte 5 and the least significant nibble of byte 4 from the token descriptor of the IPREDICT or IPREDICT.S instruction operand token.
IBITS (4 bits)	DO7	none	Contains the most significant nibble of byte 6 from the token descriptor of IPREDICT or IPREDICT.S instruction operand token.

6.9 - IPREDICT Registers (Continued)

Register Name	Address	Reset State	Description
IFMVX (11 bits)	DO8	none	The forward motion vector horizontal component field from the token descriptor of the IPREDICT or IPREDICT.S instruction operand.
IFMVY (11 bits)	DO9	none	The forward motion vector vertical component field from the token descriptor of the IPREDICT or IPREDICT.S instruction operand token.
IBMVX (11 bits)	DOA	none	The backward motion vector horizontal component field from the token descriptor of the IPREDICT or IPREDICT.S instruction operand token.
IBMVY (11 bits)	DOB	none	The backward motion vector vertical component from the token descriptor of the IPREDICT or IPREDICT.S instruction operand token.

6.10 - OPREDICT Registers



3-11

6.10 - OPREDICT Registers (Continued)

Register Name	Address	Reset State	Description
OPP0 (12 bits)	EO0	none	Bits 11:0 from word 0 of the OPREDICT or S.OPREDICT instruction processor packet; contains instruction params(4:0), opcode, no and pid fields
OPP1 (12 bits)	EO1	none	Bits 23:12 from word 1 of the OPREDICT or S.OPREDICT instruction processor packet; contains instruction dest1 field and most significant nibble of dest2 field.
OPP2 (12 bits)	EO2	none	Bits 11:0 from word 2 of the OPREDICT or S.OPREDICT instruction processor packet; contains instruction dest3 field and the least significant nibble of dest2 field.
OTD0 (12 bits)	EO3	none	Written by the KS0144 CPU with byte 0 and the most significant nibble of byte 1 of the OPREDICT or S.OPREDICT output token descriptor.
OTD1 (12 bits)	EO4	none	Written by the KS0144 CPU with byte 2 and the least significant nibble of byte 1 of the OPREDICT or S.OPREDICT output token descriptor.
OTD2 (12 bits)	EO5	none	Written by the KS0144 CPU with byte 3 and the most significant nibble of byte 4 of the OPREDICT or S.OPREDICT output token descriptor.
OTD3 (12 bits)	EO6	none	Written by the KS0144 CPU with byte 5 and the least significant nibble of byte 4 of the OPREDICT or S.OPREDICT output token descriptor.
OBITS (4 bits)	EO7	none	Written by the KS0144 CPU with the most significant nibble of byte 6 of the OPREDICT or S.OPREDICT output token descriptor.
OFMVX (11 bits)	EO8	none	Written by the KS0144 CPU with the horizontal component of the forward motion vector resulting from motion estimation. OFMVX is copied into the fmvx field of the OPREDICT or S.OPREDICT output token descriptor.
OFMVY (11 bits)	EO9	none	Written by the KS0144 CPU with the vertical component of the forward motion vector resulting from motion estimation. OFMVY is copied into the fmvy field of the OPREDICT or S.OPREDICT output token descriptor.
OBMVX (11 bits)	EOA	none	Written by the KS0144 CPU with the horizontal component of the backward motion vector resulting from motion estimation. OBMVX is copied into the bmvx field of the OPREDICT or S.OPREDICT output token descriptor.
OBMVY (11 bits)	EOB	none	Written by the KS0144 CPU with the vertical component of the backward motion vector resulting from motion estimation. OBMVY is copied into the bmvy field of the OPREDICT or S.OPREDICT output token descriptor.

SECTION 7 - KS0144 PIN ASSIGNMENTS AND DESCRIPTIONS

Figure 7.1 - KS0144 Pin Layout

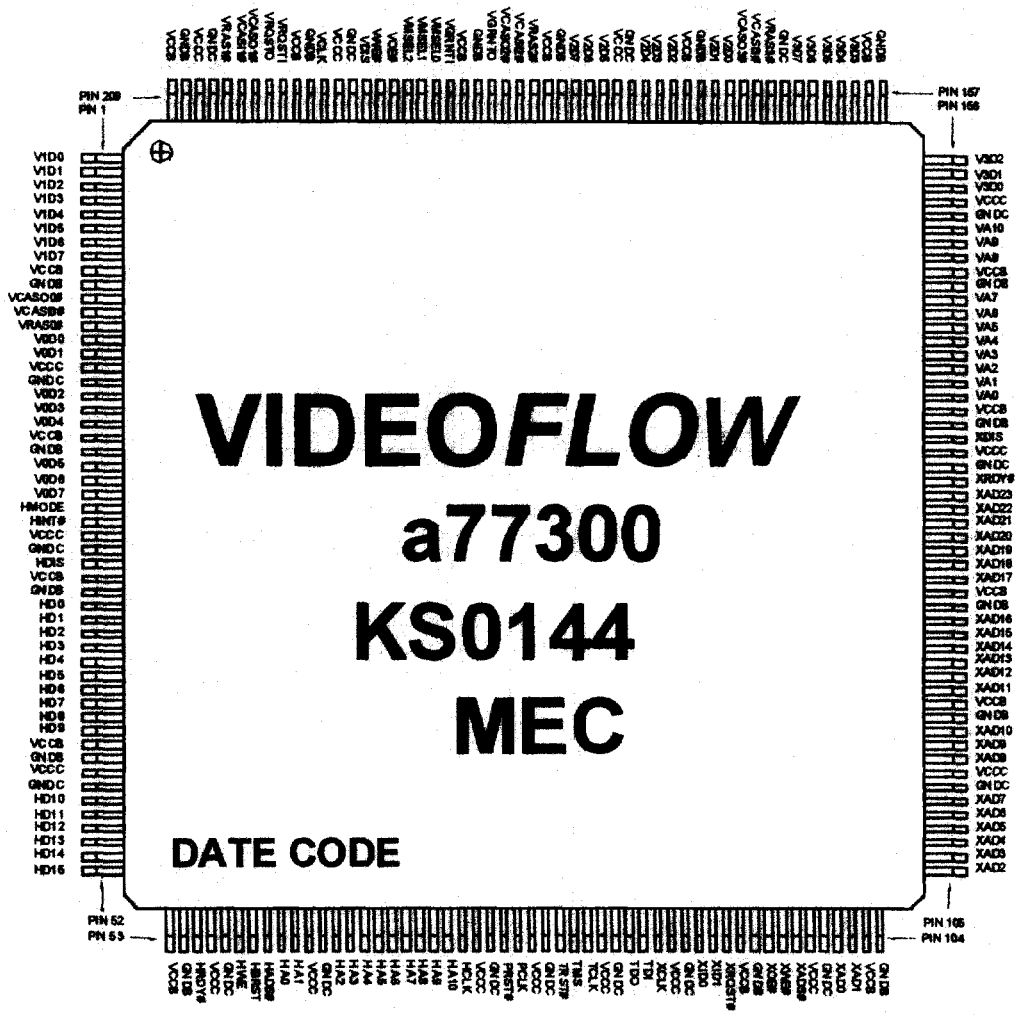


Table 7.1 - KS0144 Signal Name to Pin Number Correspondence

Name	Pin Number	Name	Pin Number	Name	Pin Number	Name	Pin Number
HA0	61	V0D3	19	VCASI1#	203	XID0	91
HA1	62	V0D4	20	VCASI2#	183	XID1	92
HA2	65	V0D5	23	VCASI3#	166	XOE#	96
HA3	66	V0D6	24	VCLK	197	XRDY#	133
HA4	67	V0D7	25	VDIS	194	XRQST#	93
HA5	68	V1D0	1	VGRNTI	188	XWE#	97
HA6	69	V1D1	2	VGRNTO	185		
HA7	70	V1D2	3	VMSEL0	189		
HA8	71	V1D3	4	VMSEL1	190		
HA9	72	V1D4	5	VMSEL2	191		
HA10	73	V1D5	6	VOE#	192		
HADS#	60	V1D6	7	VRAS0#	13		
HBRST	59	V1D7	8	VRAS1#	204		
HCLK	74	V2D0	168	VRAS2#	182		
HD0	33	V2D1	169	VRAS3#	165		
HD1	34	V2D2	172	VRQSTI	200	VCCB	9,21,31,43,
HD2	35	V2D3	173	VRQSTO	201	(+5V)	53,94,103,
HD3	36	V2D4	174	VWE#	193		117,125,138,
HD4	37	V2D5	177	XAD0	101		148,158,171,
HD5	38	V2D6	178	XAD1	102		181,187,199,
HD6	39	V2D7	179	XAD2	105		208
HD7	40	V3D0	154	XAD3	106		
HD8	41	V3D1	155	XAD4	107		
HD9	42	V3D2	156	XAD5	108	VCCC	16,28,45,56,
HD10	47	V2D3	159	XAD6	109	(+5V)	63,75,79,84,
HD11	48	V3D4	160	XAD7	110		89,99,112,
HD12	49	V3D5	161	XAD8	113		135,153,176,
HD13	50	V3D6	162	XAD9	114		196,206
HD14	51	V3D7	162	XAD10	115		
HD15	52	VA0	139	XAD11	118		
HDIS	30	VA1	140	XAD12	119	GNDB	10,22,32,44,
HINT#	27	VA2	141	XAD13	120	(ground)	54,95,104,
HMODE	26	VA3	142	XAD14	121		116,124,137,
HRDY#	55	VA4	143	XAD15	122		147,157,170,
HWE	58	VA5	144	XAD16	123		180,186,198,
PCLK	78	VA6	145	XAD17	126		207
PRST#	77	VA7	146	XAD118	127		
TCLK	83	VA8	147	XAD19	128		
TDI	87	VA9	149	XAD20	129	GNDC	17,29,46,57,
TDO	86	VA10	150	XAD21	130	(ground)	64,76,80,85,
TMS	82	VCASO0#	11	XAD22	131		90,100,111,
TRST#	81	VCASO1#	202	XAD23	132		134,152,164,
V0D0	14	VCASO2#	184	XADS#	98		175,195,205
V0D1	15	VCASO3#	167	XCLK	88		
V0D2	18	VCASI0#	12	XDIS	136		

Table 7.1 - KS0144 Signal Name to Pin Number Correspondence

Name	Pin Number	Name	Pin Number	Name	Pin Number	Name	Pin Number
V1D0	1	VCCB	53	XAD2	105	GNDB	157
V1D1	2	GNDB	54	XAD3	106	VCCB	158
V1D2	3	HRDY#	55	XAD4	107	V3D3	159
V1D3	4	VCCC	56	XAD5	108	V3D4	160
V1D4	5	GNDC	57	XAD6	109	V3D5	161
V1D	6	HWE	58	XA7	110	V3D6	162
V1D6	7	HBRST	59	GNDC	111	V3D7	163
V1D7	8	HADS#	60	VCCC	112	GNDC	164
VCCB	9	HA0	61	XAD8	113	VRAS3#	165
GNDB	10	HA1	62	XAD9	114	VCASI3#	166
VCASO0#	11	VCCC	63	XAD10	115	VCASO3#	167
VCASI0#	12	GNDC	64	GNDB	116	V2D0	168
VRAS0#	13	HA2	65	VCCB	117	V2D1	169
V0D0	14	HA3	66	XAD11	118	GNDB	170
V0D1	15	HA4	67	XAD12	119	VCCB	171
VCCC	16	HA5	68	XAD13	120	V2D2	172
GNDC	17	HA6	69	XAD14	121	V2D3	173
V0D2	18	HA7	70	XAD15	122	V2D4	174
V0D3	19	HA8	71	XAD16	123	GNDC	175
V0D4	20	HA9	72	GNDB	124	VCCC	176
VCCB	21	HA10	73	VCCB	125	V2D5	177
GNDB	22	HCLK	74	XAD17	126	V2D6	178
V0D5	23	VCCC	75	XAD18	127	V2D7	179
V0D6	24	GNDC	76	XAD19	128	GNDB	180
V0D7	25	TRST#	77	XAD20	129	VCCB	181
HMODE	26	TMS	78	XAD21	130	VRAS2#	182
HINT#	27	TCLK	79	XAD22	131	VCASI2#	183
VCCC	28	VCCC	80	XAD23	132	VCASO2#	184
GNDC	29	GNDC	81	XRDY#	133	VGRNTO	185
HDIS	30	TDO	82	GNDC	134	GNDB	186
VCCB	31	TDI	83	VCCC	135	VCCB	187
GNDB	32	XCLK	84	XDIS	136	VGRNTI	188
HD0	33	VCCC	85	GNDB	137	VMSEL0	189
HD1	34	GNDC	86	VCCB	138	VMSEL1	190
HD2	35	XID0	87	VA0	139	VMSEL2	191
HD3	36	XUD1	88	VA1	140	VOE#	192
HD4	37	XRQST#	89	VA2	141	VWE#	193
HD5	38	GNDC	90	VA3	142	VDIS	194
HD6	39	XID0	91	VA4	143	GNDC	195
HD7	40	XID1	92	VA5	144	VCCC	196
HD8	41	XRQST#	93	VA6	145	VCLK	197
HD9	42	VCCB	94	VA7	146	GNDB	198
VCCB	43	GNDB	95	GNDB	147	VCCB	199
GNDB	44	XOE#	96	VCCB	148	VRQSTI	200
VCCC	45	XWE#	97	VA8	149	VRQSTO	210
GNDC	46	XADS#	98	VA9	150	VCASO1#	212
HD10	47	VCCC	99	VA10	151	VCASI1#	203
HD11	48	GNDC	100	GNDC	152	VRAS1#	204
HD12	49	XAD0	101	VCCC	153	GNDC	205
HD13	50	XAD1	102	V3D0	154	VCCC	206
HD14	51	VCCB	103	V3D1	155	GNDB	207
HD15	52	GNDB	104	V3D2	156	VCCB	208

Table 7.2 - KS0144 Signal Name to Pin Number Correspondence

Name	Pin Number	Name	Pin Number	Name	Pin Number	Name	Pin Number
V1D0	1	VCCB	53	XAD2	105	GNDB	157
V1D1	2	GNDB	54	XAD3	106	VCCB	158
V1D2	3	HRDY#	55	XAD4	107	V3D3	159
V1D3	4	VCCC	56	XAD5	108	V3D4	160
V1D4	5	GNDC	57	XAD6	109	V3D5	161
V1D	6	HWE	58	XA7	110	V3D6	162
V1D6	7	HBRST	59	GNDC	111	V3D7	163
V1D7	8	HADS#	60	VCCC	112	GNDC	164
VCCB	9	HA0	61	XAD8	113	VRAS3#	165
GNDB	10	HA1	62	XAD9	114	VCASI3#	166
VCASO0#	11	VCCC	63	XAD10	115	VCASO3#	167
VCASI0#	12	GNDC	64	GNDB	116	V2D0	168
VRAS0#	13	HA2	65	VCCB	117	V2D1	169
V0D0	14	HA3	66	XAD11	118	GNDB	170
V0D1	15	HA4	67	XAD12	119	VCCB	171
VCCC	16	HA5	68	XAD13	120	V2D2	172
GNDC	17	HA6	69	XAD14	121	V2D3	173
V0D2	18	HA7	70	XAD15	122	V2D4	174
V0D3	19	HA8	71	XAD16	123	GNDC	175
V0D4	20	HA9	72	GNDB	124	VCCC	176
VCCB	21	HA10	73	VCCB	125	V2D5	177
GNDB	22	HCLK	74	XAD17	126	V2D6	178
V0D5	23	VCCC	75	XAD18	127	V2D7	179
V0D6	24	GNDC	76	XAD19	128	GNDB	180
V0D7	25	TRST#	77	XAD20	129	VCCB	181
HMODE	26	TMS	78	XAD21	130	VRAS2#	182
HINT#	27	TCLK	79	XAD22	131	VCASI2#	183
VCCC	28	VCCC	80	XAD23	132	VCASO2#	184
GNDC	29	GNDC	81	XRDY#	133	VGRNTO	185
HDIS	30	TDO	82	GNDC	134	GNDB	186
VCCB	31	TDI	83	VCCC	135	VCCB	187
GNDB	32	XCLK	84	XDIS	136	VGRNTI	188
HD0	33	VCCC	85	GNDB	137	VMSEL0	189
HD1	34	GNDC	86	VCCB	138	VMSEL1	190
HD2	35	XID0	87	VA0	139	VMSEL2	191
HD3	36	XUD1	88	VA1	140	VOE#	192
HD4	37	XRQST#	89	VA2	141	VWE#	193
HD5	38	GNDC	90	VA3	142	VDIS	194
HD6	39	XID0	91	VA4	143	GNDC	195
HD7	40	XID1	92	VA5	144	VCCC	196
HD8	41	XRQST#	93	VA6	145	VCLK	197
HD9	42	VCCB	94	VA7	146	GNDB	198
VCCB	43	GNDB	95	GNDB	147	VCCB	199
GNDB	44	XOE#	96	VCCB	148	VRQSTI	200
VCCC	45	XWE#	97	VA8	149	VRQSTO	210
GNDC	46	XADS#	98	VA9	150	VCASO1#	212
HD10	47	VCCC	99	VA10	151	VCASI1#	203
HD11	48	GNDC	100	GNDC	152	VRAS1#	204
HD12	49	XAD0	101	VCCC	153	GNDC	205
HD13	50	XAD1	102	V3D0	154	VCCC	206
HD14	51	VCCB	103	V3D1	155	GNDB	207
HD15	52	GNDB	104	V3D2	156	VCCB	208

Table 7.3 - KS0144 Pin Descriptions

Name (1)	Type (2)	PU/PO (3)	Description
			Video Memory Interface (VBUS)
VA[10:0]	TSO	PU	VIDEO ADDRESS BUS - multiplexed row or column address asserted in conjunction with VRAS# and VCASO#. Connected to the DRAM or VRAM address pins used by the KS0144.
VMSEL[2:0]	TSO	PU	VIDEO MEMORY SELECT - selects the desired DRAM or VRAM from those that are connected to the KS0144 video data bus. The state of the VMSEL pins is determined by a WRMEM instruction parameter or the KS0144 CPU registers V0MSEL, V1MSEL, V2MSEL, or V3MSEL. VMSEL pins are driven to the low state while the KS0144 is performing a memory refresh.
V0D[7:0] V1D[7:0] V2D[7:0] V3D[7:0]	I/O I/O I/O I/O	PU PU PU PU	VIDEO DATA BUSES - four buses that transfer video data between the off-chip DRAMs and the KS0144. Each bus transfers a single 8-bit pel during the execution of a LDVIB command from the KS0144 CPU or a WRMEM instruction from the KS0143. V0D and V1D are used to simultaneously transfer pels if VBUSMODE =1. All four buses simultaneously transfer pels if VBUSMODE = 0.
VWE#	TSO	PU	VIDEO MEMORY WRITE ENABLE - when active (low), enables writing to the selected DRAM or VRAM. Connected to the WE# pins of memories connected to the KS0144.
VOE#	TSO	PU	VIDEO MEMORY OUTPUT ENABLE - when active (low), enables reading from the selected DRAM or VRAM. Connected to the OE# pins of memories connected to the KS0144.
VRAS[3:0]#	TSO	PU	VIDEO MEMORY ROW ADDRESS STROBE - when active (low), enables the VA[10:0] pins to select the row address of the selected DRAM or VRAM. Connected to the RAS# pins of memories connected to the KS0144.
VCASO[3:0]#	TSO	PU	VIDEO MEMORY COLUMN ADDRESS STROBE OUTPUT - when active (low), enables the VA[10:0] pins to select the column address of the selected DRAM or VRAM. Connected to the CAS# pins of memories connected to the KS0144.
VCASI[3:0]#	TSO	PU	VIDEO MEMORY COLUMN ADDRESS STROBE INPUT - when active (low), latches the data from the corresponding video data bus during the execution of video memory read instructions. When common page mode DRAMs and VRAMs are used, the VCASI# pins should be connected to the corresponding VCASO# pins using as short a path as possible. If hyper page mode DRAMs and VRAMs are being used, the VCASI# pins are tied to the low state.
VRQSTI VRQSTO	I O	PD none	VIDEO BUS REQUEST DAISY CHAIN (INPUT and OUTPUT) - two signals used by the potential bus masters to request the video bus during arbitration. VRQSTI connects to VRQSTO of the next lower priority bus master. VRQSTO connects to VRQSTI of the next higher priority bus master. VRQSTI of the lowest priority bus master is tied to the low state.
VGRNTI VGRNTO	I O	PD none	VIDEO BUS GRANT DAISY CHAIN (INPUT and OUTPUT) - two signals used in the awarding of the video bus to a bus master during bus arbitration. VGRNTI connects to VGRNTO of the next higher priority bus master. VGRNTO connects to VGRNTI of the next lower priority bus master. VGRNTI of the highest priority master is tied to either the low or high state.
VDIS	I	PU	VIDEO BUS DISABLE - when active (high), places a high impedance state on the output drivers of the following video bus pins: VA, VMSEL, V0D, V1D, V2D, V3D, VRAS#, VCASO#, VWE#, and VOE#.
VCLK	I	PD	VIDEO BUS CLOCK - the master clock input used by the KS0144 for asserting and sampling all signals on the video bus. The KS0144 controls the timing of all memory read, write, and refresh operations using programmable registers which count periods of VCLK.

Table 7.3 - MEC Pin Descriptions (Continued)

Name (1)	Type (2)	PU/PD (3)	Description
			AUXILIARY PROCESSOR INTERFACE (XBUS) (Continued)
XRQST#	O	none	AUXILIARY PROCESSOR SERVICE REQUEST - when asserted (low), indicates that the MEC has completed executing the previous instruction sent to it and is requesting a read cycle from the ICC.
XID[1:0]	I	PD	AUXILIARY PROCESSOR ID SELECT - two pins used to hardwire a processor address for the MEC. The MEC determines if it is being selected by the ICC by comparing these pins with the pid output on XAD[1:0] at the start of a read write cycle.
XCLK	I	PD	AUXILIARY INTERFACE BUS CLOCK - the master clock used by the MEC for asserting and sampling all signals on the auxiliary bus.
			IEEE 1149.1 TEST INTERFACE
TDI	I	PU	TEST DATA INPUT - serial data input shifted into an internal scan path or the boundary scan path for the purpose of testing internal logic I/O buffers; synchronous to the rising edge of TCLK.
TDO	O	none	TEST DATA OUTPUT - serial data output from an internal scan path or the boundary scan path; synchronous to the falling edge of TCLK.
TCLK	I	PD	TEST CLOCK - strobes data from or to the test access port.
TMS	I	PU	TEST MODE SELECT - when asserted (high), controls the operation of the test access port.
TRST#	I	PU	TEST RESET - when asserted (low), asynchronously resets the test access port. When not using the IEEE test access port for testing, tie this signal to the low state.
			PROCESSOR CLOCK AND RESET
PCLK	I	PD	PROCESSOR CLOCK - the primary clock for all MEC internal processing and data transfers.
PRST#	I	PU	PROCESSOR RESET - when asserted (low), asynchronously sets the MEC to its initial state.
			POWER SUPPLY
GND8, GNDC			CHIP GROUND - 33 pins which must be connected to a board ground plane.
VCC8, VCCC			CHIP POWER - 34 pins which must be connected to a board VCC plane (norminally, +5.0V)

Notes: 1. Name Definitions:

> Pin names ending in "#" are active - low
> Pin names ending in "#" are active - high
> names with an "x:y" designation indicate a bus of pins numbered "y" to "x", where "x" is the most significant bit.

2. Type Definitions:

> I = Input
> O = Output
> I/O = Bidirectional
> TSO = Tri - state output
> OD = Open drain output

3. PU/;PD Definitions :

> PD = An active pull-down device (-200K ohms) is present (built into the pin's circuitry ; see Figure 9.6)
> PU = An active pull-up device (-200K ohms) is present (built into the pin's circuitry; see Figure 9.6)
> none = no pull-up or pull-down device present

SECTION 8 - PROGRAMMING THE VBUS TIMING CONTROL REGISTERS

Section 6.5 describes the KS0144 VCNTn registers which may be programmed by the KS0144 CPU to support a wide range of DRAM devices at various clock speeds. These registers set the relationship and duration of the various strobes used to access memory on the VBUS.

Video memory bus timing is established in multiples of VCLK clock periods as shown in Figures 8.1 and 8.2. In the following paragraphs, the symbol tv refers to the VCLK duration (tv = 1/VCLK). For example, if VCLK is 40 MHz, then tv = 25ns.

Note:
The VCNTn registers must not be set to the value 0.

Page Mode Memory Cycle Timing
The KS0144 accesses the selected DRAM using fast page mode memory cycles (see Figure 8.1). The first step in programming the VCNTn registers for fast page mode operation is to select the CAS cycle time with the VCNT4 and VCNT8 registers. The VCNT8 register contents represent the active (low) CAS pulse width and must be large enough to satisfy the minimum CAS pulse width (tCAS), the access time from CAS (tCAC), and the column address hold time (tCAH) specified for the memories in use:

$$VCNT8 * tv \geq \max(tCAS, tCAC, tCAH)$$

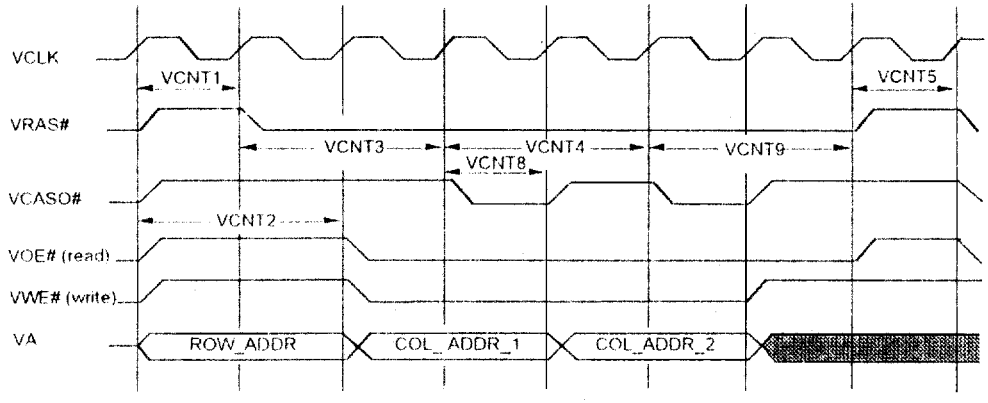
The VCNT4 register contents represent the cycle time of the CAS signal and must be selected to satisfy the minimum CAS cycle time (tPC) without causing a violation of the minimum CAS precharge time (tCP) or the column address setup time (tASC). This is described by the following relation:

$$VCNT4 * tv \geq \max(tPC, VCNT8 * tv + tCP, VCNT8 * tv + tASC)$$

The VCNT3 register contents must also satisfy the RAS to CAS delay time (tRCD), the access time from RAS (tRAC), and the CAS hold time (tCSH):

$$VCNT3 * tv \geq \max(tRCD, tRAC - VCNT8 * tv, tCSH - VCNT8 * tv)$$

Figure 8.1 - VBUS Fast Page Mode Timing Diagram



The VCNT9 register contents must be set to at least VCNT4 and possibly more to satisfy the RAS hold time (tRSH):

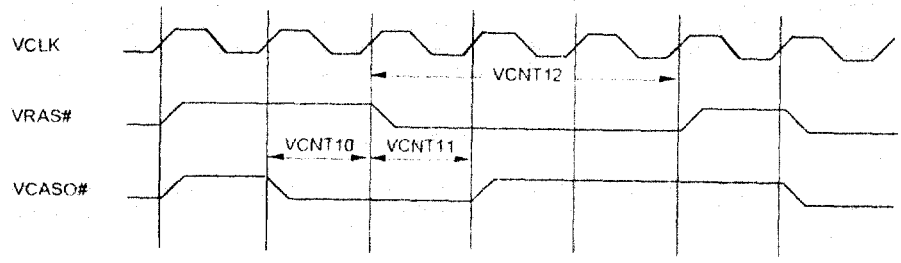
$$VCNT9 * tv \geq \max(VCNT4 * tv, tRSH)$$

Refresh Timing

If the refresh feature of the KS0144 is used, the refresh timing parameters must be initialized (see Figure 8.2). For CAS before RAS refresh cycles, the contents of VCNT10 and VCNT11 can typically be set to "1" since both the CAS setup time (tCSR) and the CAS hold time (tCHR) are less than one clock cycle (tv). VCNT12, however, must meet the minimum RAS pulse width (tRAS) which is usually 3 to 4 cycles:

$$VCNT12 * tv \geq tRAS$$

Figure 8.2 - VBUS Refresh Timing Diagram



SAMSUNG ELECTRONICS

SECTION 9 - ELECTRICAL SPECIFICATIONS

Table 9.1 - Absolute Maximum Ratings

Parameter	Max Rating
Power Supply Voltage	-0.5V to 7.0V
DC Output Voltage (applied in Hi-Z state)	-0.5V to 7.0V
Low Level Output Current	20mA
Case Temperature Under Bias	-55×C to 110×C
Storage Temperature	-65×C to 150×C

Note: Operation at any Max Rating is not implied. Ratings are provided for guidance purposes only and are not tested. Exposure to absolute maximum rating conditions over extended periods may affect device reliability.

Table 9.2 - Operating Conditions

Parameter	Symbol	Test Conditions	Limits		
			Min	Max	Unit
Supply Voltage	V_{CC}		4.75	5.25	V
Case Temperature	T_C		0	100	℃

Table 9.3 - DC Electrical Characteristics (within operating conditions)

Parameter	Symbol	Test Conditions	Limits		
			Min	Max	Unit
Standby Current	KS0143Q	V_{CC} = Max, f = 0.0 MHz (all clocks) XDIS = HDIS = VDIS = 5.0V VIH = 5.0V, V_{IL} = 0.0V, I_{OUT} = 0.0mA		200	mA
Dynamic Supply Current	KS0143A	V_{CC} = Max, f = 50.0 MHz (PCLK) f = 25 MHz (XCLK, VCLK, HCLK) V_{IH} = 5.0V, V_{IL} = 0.0V, I_{OUT} = 0.0mA		1000	mA
Input High Voltage	V_{IH}		2.0		V
Input Low Voltage	V_{IL}			0.8	V
Output High Voltage	V_{OH}	V_{CC} = Min, I_{OH} = -4.0 mA	2.4		V
Output Low Voltage	V_{OL}	V_{CC} = Min, I_{OL} = 8.0 mA		0.4	V
Output Low Voltage Open Drain	V_{OLD}	V_{CC} = Min, I_{OLD} = 30.0 mA		0.4	V
Input High Current Pull-up	I_{IHU}	V_{CC} = Max, V_{IN} = VCC		5.0	μA
Input High Current Pull-down	I_{IHD}	V_{CC} = Max, V_{IN} = VCC		30.0	μA
Input Low Current Pull-up	I_{ILU}	V_{CC} = Max, V_{IN} = 0.0V	-30.0		μA
Input Low Current Pull-down	I_{ILD}	V_{CC} = Max, V_{IN} = 0.0V	-5.0		mA
Output Hi-Z Current Pull-up	I_{OZH}	V_{CC} = Max, V_{OUT} = VCC		5.0	μA
Output Hi-Z Current Pull-up	I_{OZL}	V_{CC} = Max, V_{OUT} = 0.0V	-30.0		μA
Input Clamp Voltage	V_{IC+}	V_{CC} = 0.0 V, I_{IN} = 1 mA	0.3	1.5	V
Input Clamp Voltage	V_{IC-}	V_{CC} = 5.0 V, I_{IN} = -1 mA	-1.5	-0.3	V
Input Pin Capacitance **	C_{IN}	V_{CC} = 5.0 V, T_C = 25×C		7.0	pF
I/O Pin Capacitance **	$C_{I/O}$	V_{CC} = 5.0 V, T_C = 25×C		10.0	pF

Table 9.4 - AC Test Conditions

Parameter	Test Condition
AC Switching Waveform Input Rise Time (tr) and Fall (tf)Time (10-90%) See Figures 9.1 and 9.2	4.0 ns (Max)
AC Switching Waveform Input Timing Reference Level See Figures 9.1 and 9.2	1.5V
AC Switching Waveform Output Timing Reference Level See Figures 9.1 and 9.2	1.5V
Output Load: See AC Output Equivalent Load Configuration (Figure 9.3) See Open Drain Equivalent Load Configuration (Figure 9.4) See Hi-Z Equivalent Load Configuration (Figure 9.5)	
Tc	0× to 100×C
Vcc	5.0V ± 5%

Figure 9.1 - AC Switching Waveform

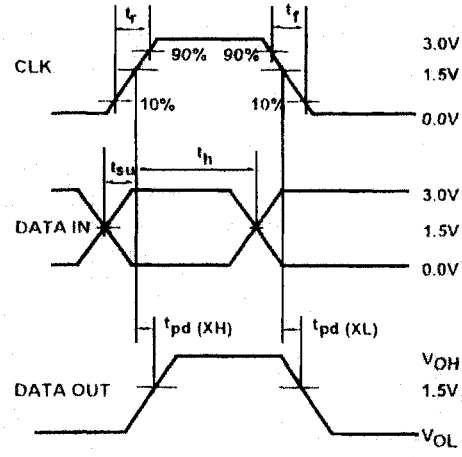
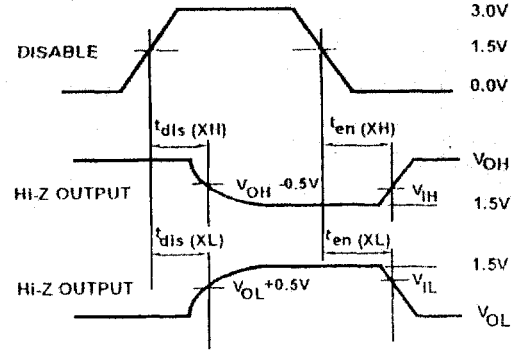


Figure 9.2 - Hi-Z Switching Waveform



SAMSUNG ELECTRONICS

Figure 9.3 - AC Output Equivalent Load Configuration

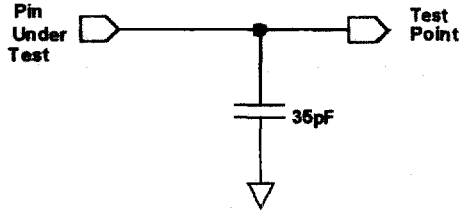


Figure 9.4 - Open Drain Equivalent Load Configuration

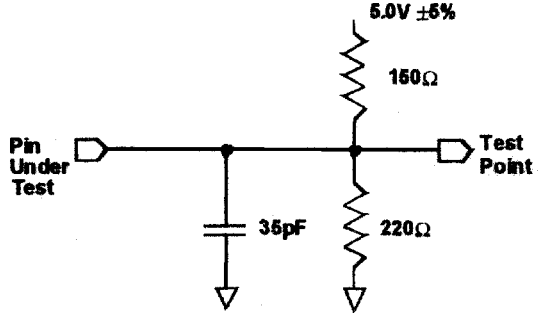


Figure 9.5 - Hi-Z Output Equivalent Load Configuration

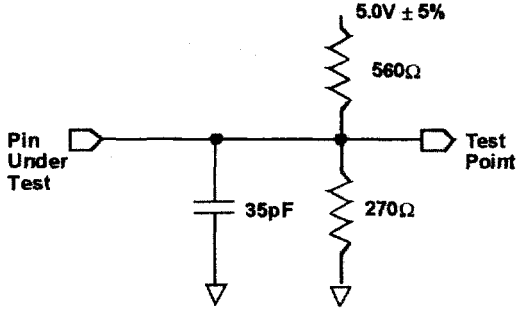


Figure 9.6 - Pull-up and Pull-down Circuits

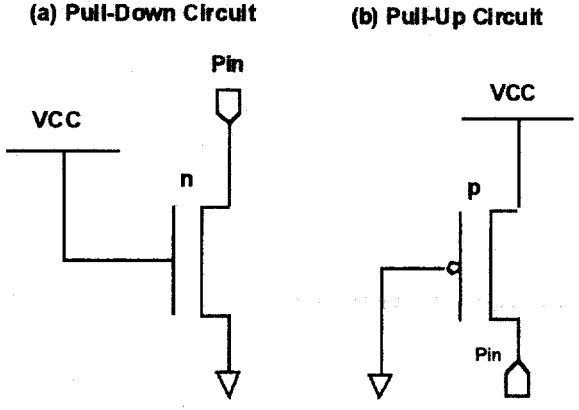


Figure 9.7 - HBUS State Transition Diagram

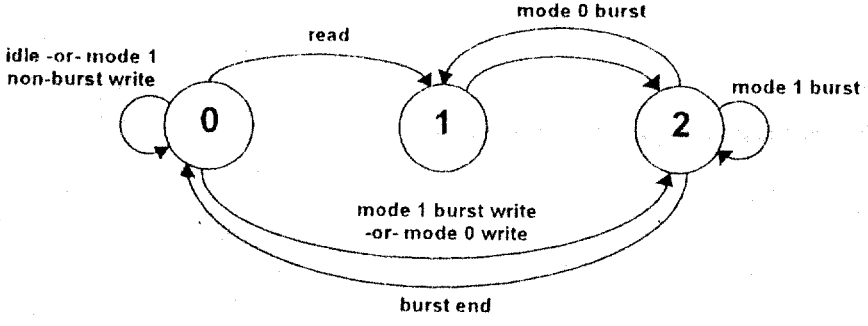


Table 9.5 - HBUS Timing Parameters

Identifier	Symbol	Description	KS0143CQ-50 Min	KS0143CQ-50 Max	KS0143CQ-40 Min	KS0143CQ-40 Max	Units	
Host Bus Clock								
H1	tc (HCLK)	Host bus clock period			30		ns	2
H2	tw (HCLKH)	Host bus clock high pulse width			12		ns	
H3	twb (HCLKL)	Host bus clock pulse width			12		ns	
	tr (HCLK)	Host bus clock rise time				4	ns	1
	tf (HCLK)	Host bus clock fall time				4	ns	1
Host Bus Control Strobe Timing								
H4	tsu (HS)	Host address strobe setup time			2		ns	
H5	th (HS)	Host address strobe hold time			6		ns	
H6	tsu (HM)	Host bus mode setup time			8		ns	
H7	th (HM)	Host bus mode hold time			2		ns	
H8	tsu (HW)	Host write enable setup time			8		ns	
H9	th (HW)	Host write enable hold time			2		ns	
H10	tsu (HBL)	Host burst low setup time			8		ns	
H11	th(HBL)	Host burst low hold time			2		ns	
H12	tsu (HBH)	Host burst high setup time			8		ns	
H13	th (HBR)	Host burst high hold time			2		ns	
H14	tpd (HRL)	Host ready low delay					ns	
H15	tv (HRL)	Host ready low valid			3		ns	
H16	tpd (HRH)	Host ready high delay					ns	
H17	tv (HRH)	Host ready high valid			3		ns	
Host Bus Address and Data Timing								
H18	tsu (HA)	Host address setup time			8		ns	
H19	th (HA)	Host address hold time			2		ns	
H20	ta (HD)	Host output data access time				19	ns	
H21	tv (HD)	Host output data vaild time			5		ns	
H22	tv (Hdt)	Host terminal output data valid time			5		ns	
H23	tsu (HD)	Host input data setup time			8		ns	
H24	th (HD)	Host input data hold time			2		ns	
H25	ten (HD)	Host clock to host data bus driven				25	ns	1
H26	tdis (HD)	Host clock to host data bus tri-state				25	ns	1
H27	ten (HE)	Host bus enable to host data bus valid				35	ns	
H28	tdis (HE)	Host bus enable to host data bus tri-state				35	ns	
Host Interrupt Timing								
H29	tdp (HI)	Host interrupt delay				24	ns	
H30	tv (HI)	Host interrupt valid			5		ns	

Notes:

1. Parameter is guaranteed by design and characterization data, but not tested.
2. tc(HCLK) must be Š tc(PCLK)

Figure 9.8 - Non-Burst Host Read Cycle Waveform
(See Figure 9.7 for a diagram of the state transitions)

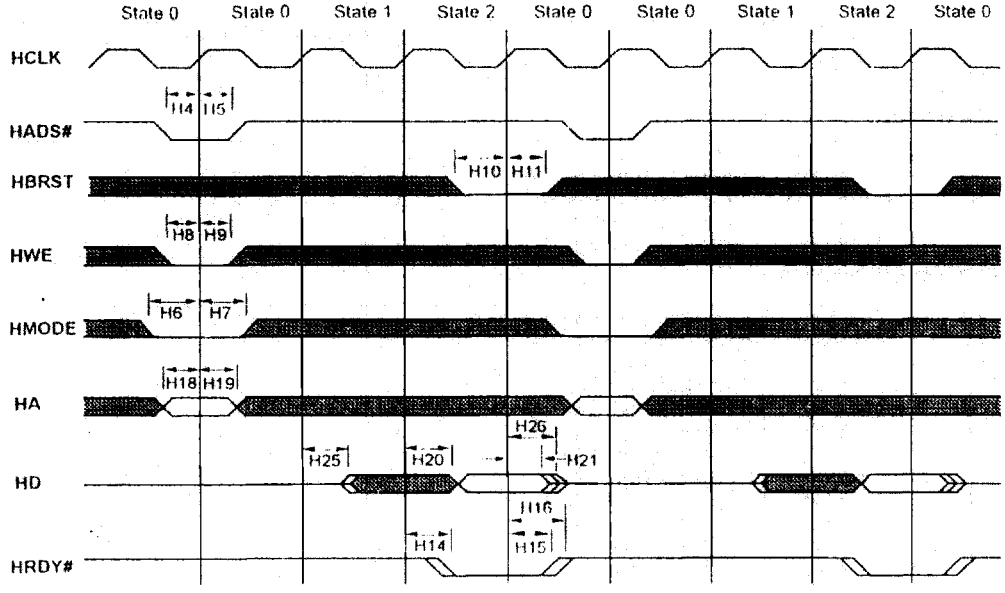
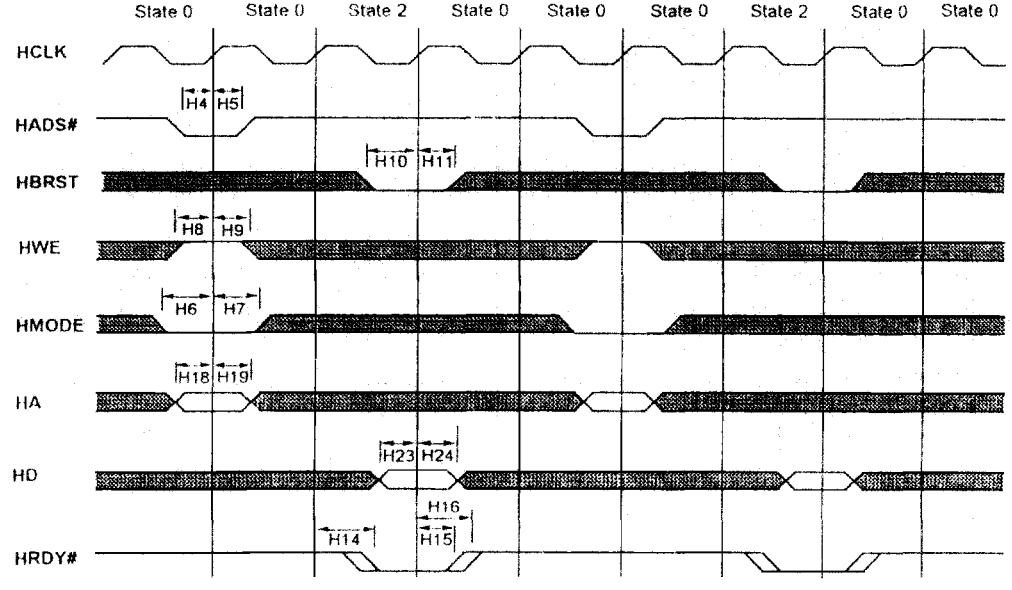


Figure 9.9 - Mode 0 Non-Burst Host Write Cycle Waveform
(See Figure 9.7 for a diagram of the state transitions)



SAMSUNG ELECTRONICS

Figure 9.10 - Mode 1 Non-Burst Host Write Cycle Waveform
(See Figure 9.7 for a diagram of the state transitions)

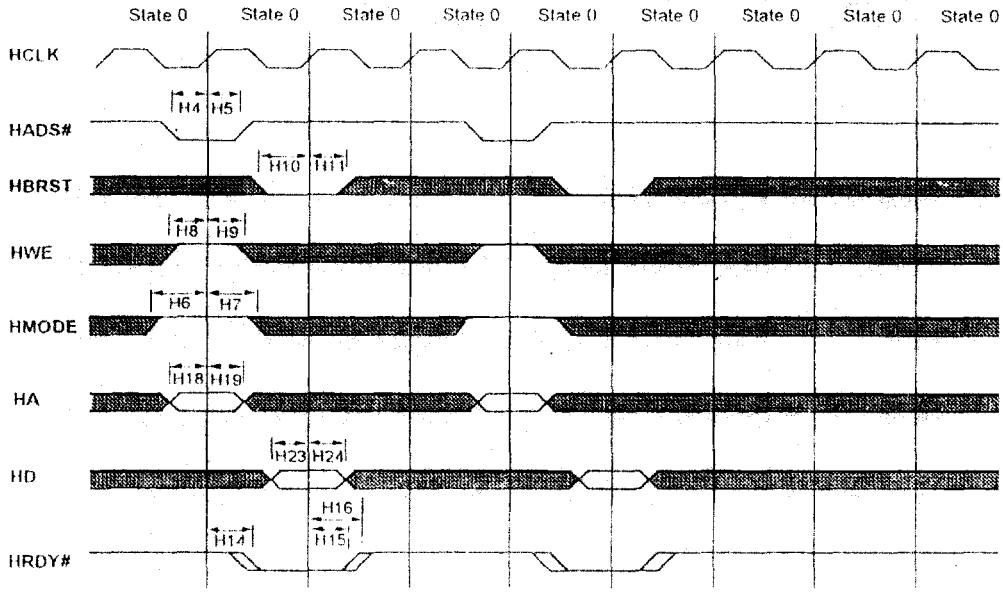
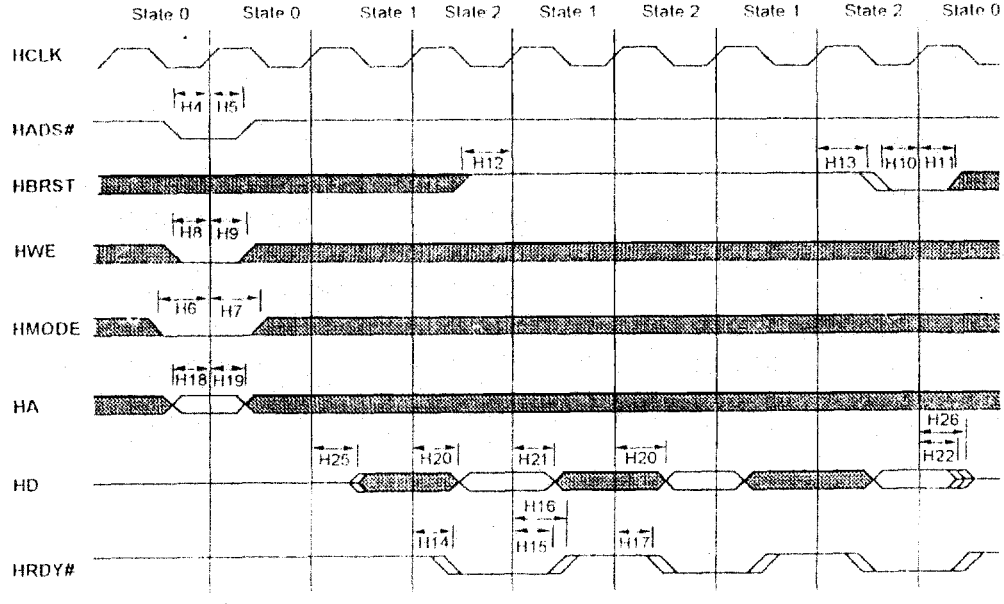


Figure 9.11 - Mode 0 Burst Host Read Cycle Waveform
(See Figure 9.7 for a diagram of the state transitions)



SAMSUNG ELECTRONICS

Figure 9.12 - Mode 1 Burst Host Read Cycle (BLENGTH = 2) Waveform
(See Figure 9.7 for a diagram of the state transitions)

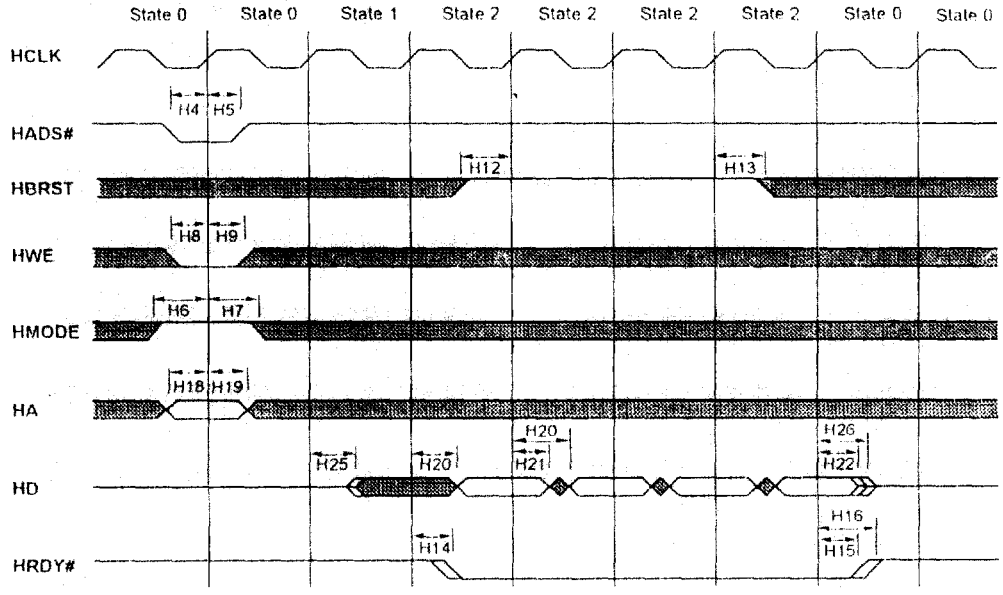


Figure 9. 13 - Mode 0 Burst Host Write Cycle Waveform
(See Figure 9.7 for a diagram of the state transitions)

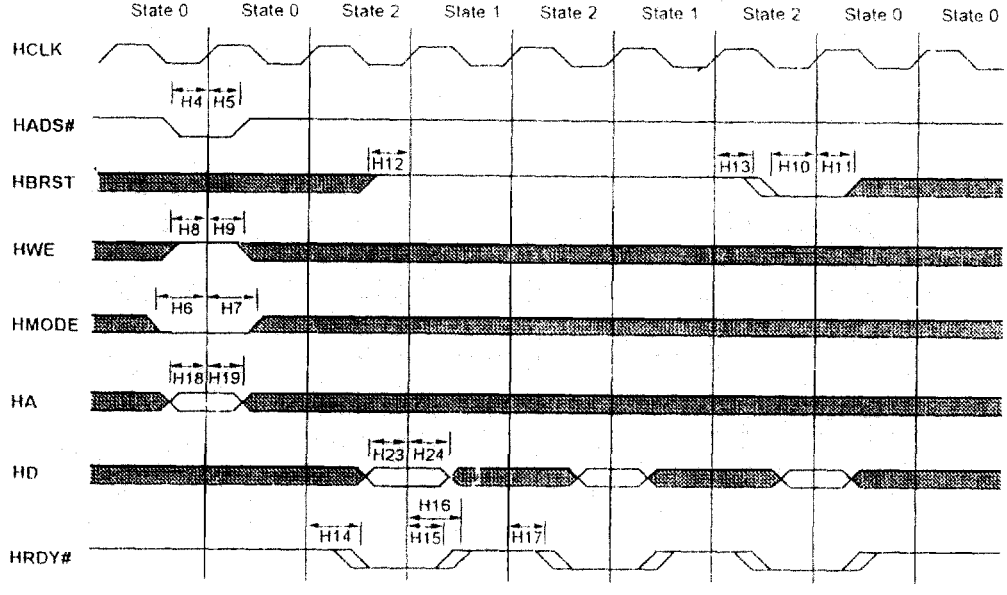


Figure 9.14 - Mode 1 Burst Host Write Cycle Waveform
(See Figure 9.7 for a diagram of the state transitions)

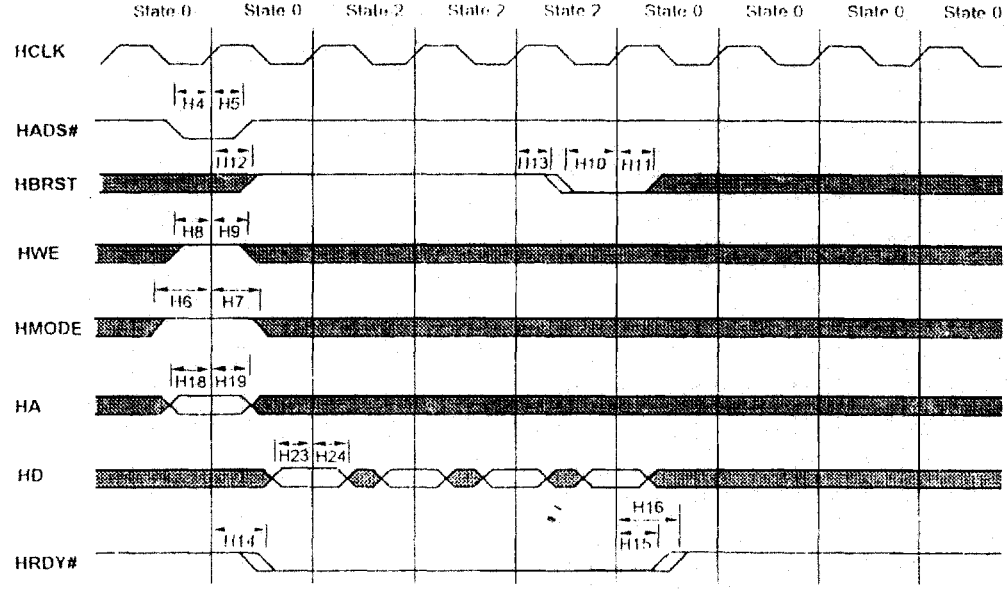
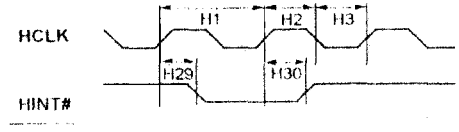


Figure 9.16 - Host Interrupt Waveform



NOTE: HINT# remains low until host processor clears interrupts by writing to CLRHINT register.

Figure 9.16 - Host Bus Disable Waveform

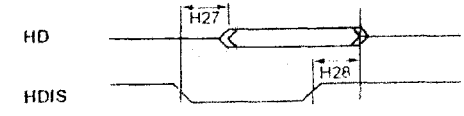


Table 9.6 - VBUS Timing Parameters

Identifier	Symbol	Description	KS0143CQ-50 Min	KS0143CQ-50 Max	KS0143CQ-40 Min	KS0143CQ-40 Max	Units	Notes
Host Bus Clock								
V1	ten(VM)	VCLK to video memory select valid				24		
V2	tdis(VM)	VCLK to video memory select tri-state				24		1
V3	ten(VA)	VCLK to video memory address valid				24		
V4	ten(VA)	VCLK to video memory address tri-state				24		1
V5	ten(VD)	VCLK to video memory data driven				24		
V6	tdis(VD)	VCLK to video memory data tri-state				24		1
V9	ten(VB)	VDIS to output valid				28		
V10	tdis(VB)	VDIS to output tri-state				28		1
Video Memory Bus								
V11	tv(VM)	Video memory select valid after VCLK			5		ns	
V12	tpd(VRL)	VCLK to video memory row addr strobe low				20	ns	
V13	tv(VRL)	Video memory row addr strobe low after VCLK			5		ns	
V14	tpd(VCL)	VCLK to video memory column addr strobe low			5	20	ns	
V15	tpd(VCH)	VCLK to video memory column addr strobe high			5	20	ns	
V16	tpd(VA)	VCLK to video memory addr prop delay			5	20	ns	
V17	tpd(VWL)	VCLK to video memory write enable				20	ns	
V18	tv(VWL)	Video memory write enable low after VCLK			5		ns	
V19	tpd(VEL)	VCLK to video memory output enable				20	ns	
V20	tv(VEL)	Video memory output enable low after VCLK			5		ns	
V21	tpd(VD)	VCLK to video memory data output prop delay			5	20	ns	
V22	tsu(VSH)	VCAS# to VCLK setup time			5		ns	
V23	th(VSH)	VCAS# to VCLK hold time			4		ns	
V24	tw(VSL)	VCAS# pulse width low			16		ns	
V25	tsu(VDs)	VnD to VCAS# high setup time			6		ns	
V26	th(VDs)	VnD to VCAS# high hold time			0		ns	
V27	tsu(VSL)	VCAS# to VCLK setup time			5		ns	1
V28	th(VSL)	VCAS# to VCLK hold time			4		ns	1
V29	tsu(VDc)	VnD to VCLK setup time (VCAS# low)			5		ns	
V30	th(VDc)	VnD to VCLK hold time (VCAS# low)			5		ns	
Video Memory Bus								
V31	tw(VCL)	VCASO# low width			T8-3		ns	3
V32	tw(VCH)	VCASO# high width			T4-T8-3		ns	3
V33	tc(VC)	VCASO# cycle time			T4-1		ns	3
V34	td(VRLCL)	VRAS# low to VCASO# low			T3-3		ns	3
V35	td(VCLCH)	VCASO# low to VCASO# high			T8-3		ns	3
V36	td(VRLCH)	VRAS# low to VCASO# high			T3+T8-3		ns	3
V37	td(VCLRL)	VCASO# low to VRAS# low (refresh cycle)			T10-3		ns	3
V38	td(VRLCLr)	VCASO# low after VRAS# low (refresh cycle)			T11-3		ns	3
Video Memory Bus Clock								
V40	tc(VCLK)	Video memory bus clock period			25		ns	2
V41	tw(VCLKH)	Video memory bus clock high pulse width			10		ns	
V42	tw(VCLKL)	Video memory bus clock low pulse width			10		ns	
	tr(VCLK)	Video memory bus clock rise time					ns	1
	tf(VCLK)	Video memory bus clock fall time					ns	1

Table 9.6 - VBUS Timing Parameters (continued)

Identifier	Symbol	Description	KS0143CQ-50 Min	KS0143CQ-50 Max	KS0143CQ-40 Min	KS0143CQ-40 Max	Units	Notes
Video Memory Bus Artitration								
V43	tsu(VQ)	Video mem bus request input setup time			3		ns	
V44	th(VQ)	Video mem bus request input hold time			4		ns	
V45	tsu(VG)	Video mem bus grant input setup time			3		ns	
V46	th(VG)	Video mem bus grant input hold time			4		ns	
V47	tpd(VQ)	Video mem bus grant request output prop delay			5	20	ns	
V48	tpd(VG)	Video mem bus grant output prop delay			5	20	ns	

Notes:

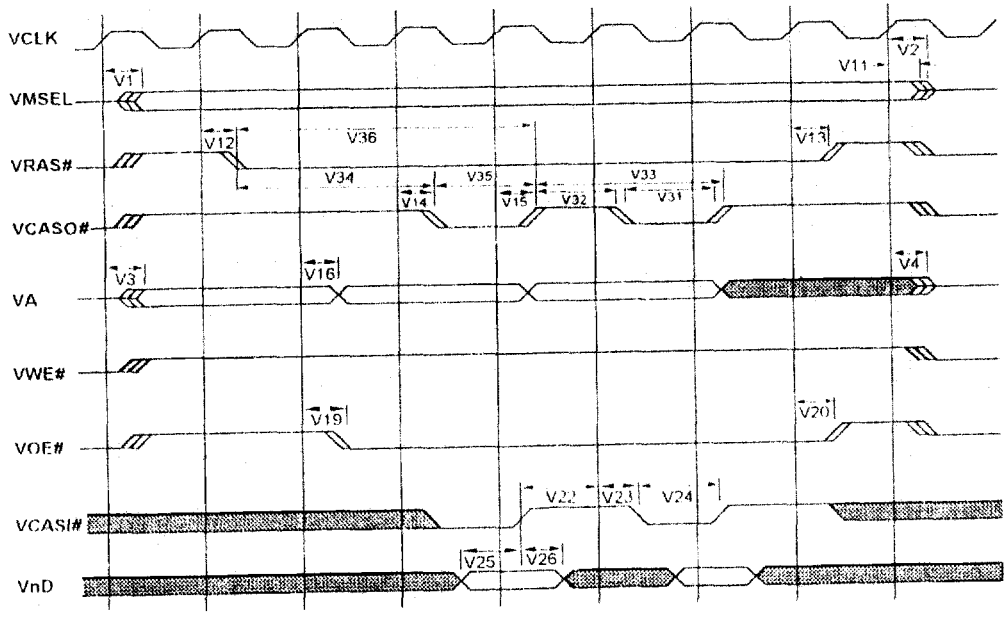
1.*Parameter is guaranteed by design and characterization data, but not tested*

2tc(VCLK) must be ≥ tc(PCLK)

3.Tn =tc(VCLK) x VCNTn

3-11

Figure 9.17 - Page Mode Read Cycle Waveforms



SAMSUNG ELECTRONICS

Figure 9.18 - Hyper Page Mode Read Cycle Waveforms

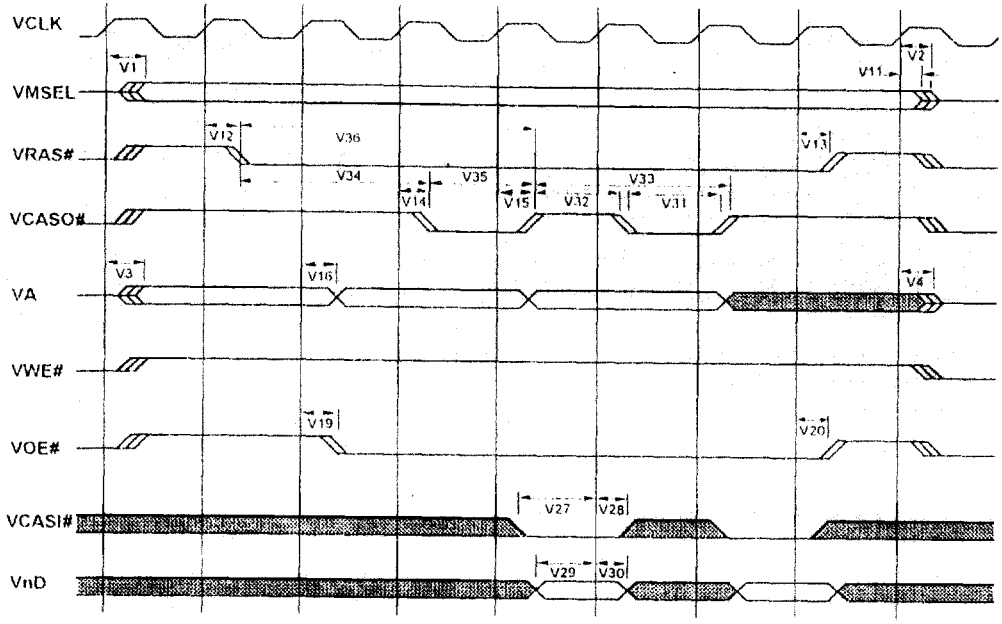


Figure 9.19 - Page Mode Write Cycle Waveforms

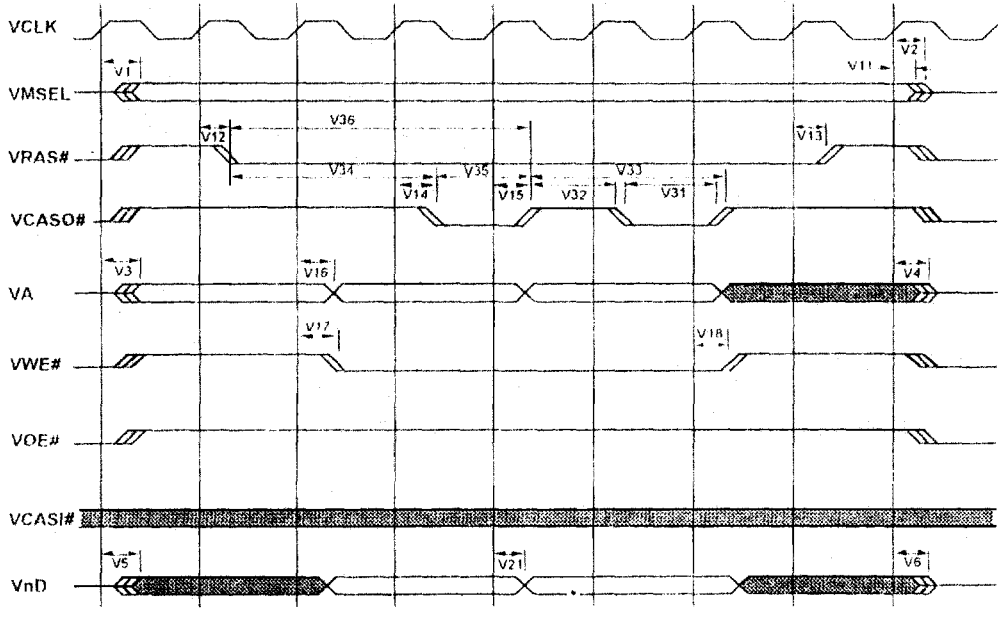


Figure 9.20 - CAS Before RAS Refresh Waveform

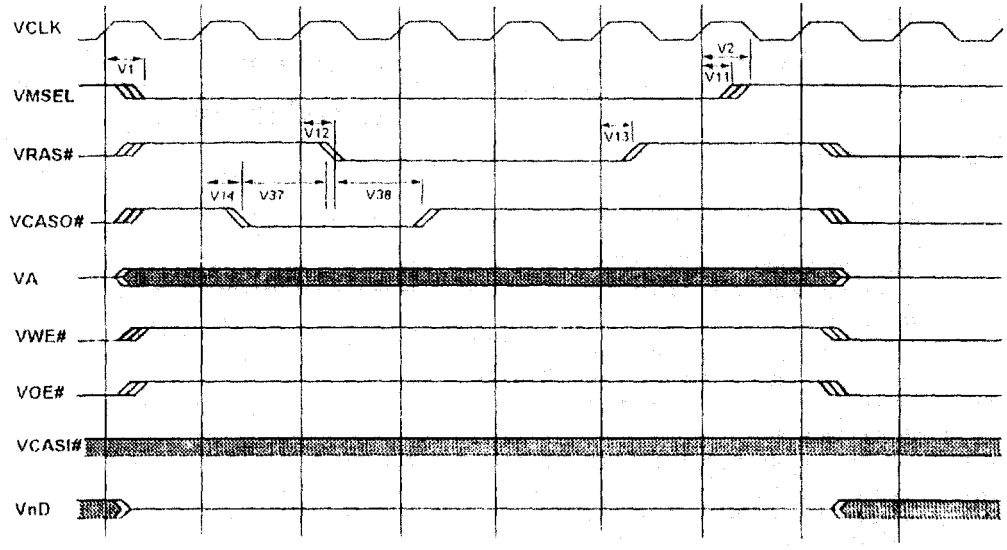


Figure 9.21 - Video Bus Arbitration Waveform

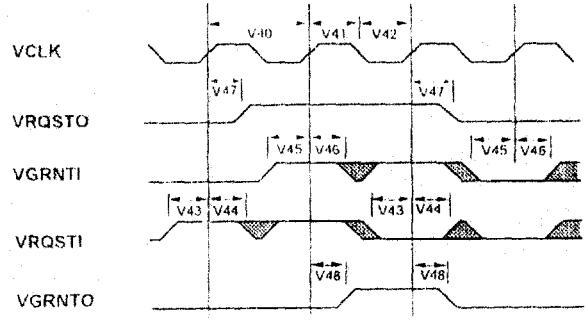


Figure 9.22 - Video Bus Disable Waveform

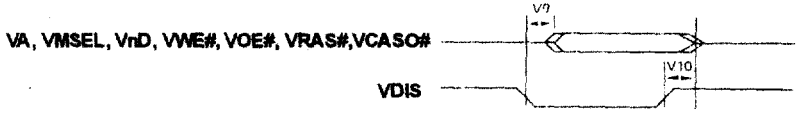


Table 9.7 - XBUS Timing Parameters

Identifier	Symbol	Description	KS0143CQ-50 Min	KS0143CQ-50 Max	KS0143CQ-40 Min	KS0143CQ-40 Max	Units	Notes
Video Memory Bus Artitration								
X1	tc(XCLK)	Auxiliary bus clock period			25		ns	2
X2	tw(XCLKH)	Auxiliary bus clock high pulse width			10		ns	
X3	tw(XCLKL)	Auxiliary bus clock low pulse width			10		ns	
	tr(XCLK)	Auxiliary bus clock rise time				4	ns	1
	tf(XCLK)	Auxiliary bus clock fall time				4	ns	1
Auxiliary Bus Arbitration								
X4	ta(XQ)	Auxiliary service request access time			7	20	ns	
X5	tpd(XQL)	Auxiliary service request strobe low delay				20	ns	
X6	tv(XQL)	Auxiliary service request low valid			7			
Auxiliary Bus Strobe Timing								
X7	tsu(XS)	Auxiliary address strobe setup time			0		ns	
X8	th(XS)	Auxiliary address strobe hold time			8		ns	
X9	tsu(XW)	Auxiliary write enable setup time			0		ns	
X10	th(XW)	Auxiliary write enable hold time			8		ns	
X11	ten(XEA)	Auxiliary output enable to address tri-state				25	ns	
X12	tdis(XEA)	Auxiliary output enable to address valid				25	ns	1
X13	ten(XED)	Auxiliary output enable to data tri-state				25	ns	
X14	tdis(XED)	Auxiliary output enable to data valid				25	ns	1
X15	tpd(XRL)	Auxiliary processor ready strobe low delay				20	ns	
X16	tv(XRL)	Auxiliary processor ready low valid			7		ns	
Auxiliary Bus Data Transfer Timing								
X17	tsu(XA)	Auxiliary address setup time			0		ns	
X18	th(XA)	Auxiliary address hold time			8		ns	
X19	tsu(XD)	Auxiliary bus input data setup time			0		ns	
X20	th(XD)	Auxiliary bus input data hold time			8		ns	
X21	ta(XD)	Auxiliary bus output data access time				24	ns	
X22	tv(XD)	Auxiliary bus output data valid time			9		ns	
X23	ten(XD	Auxiliary bus enable to data valid				25	ns	
X24	tdis(XD)	Auxiliary bus disable to data tri-state				25	ns	1

Notes:
1. Parameter is guaranteed by design and characterization data, but not tested
2. tc(XCLK) must be Š tc(PCLK)

Figure 9.23 - XBUS Read Cycle Waveforms (start)

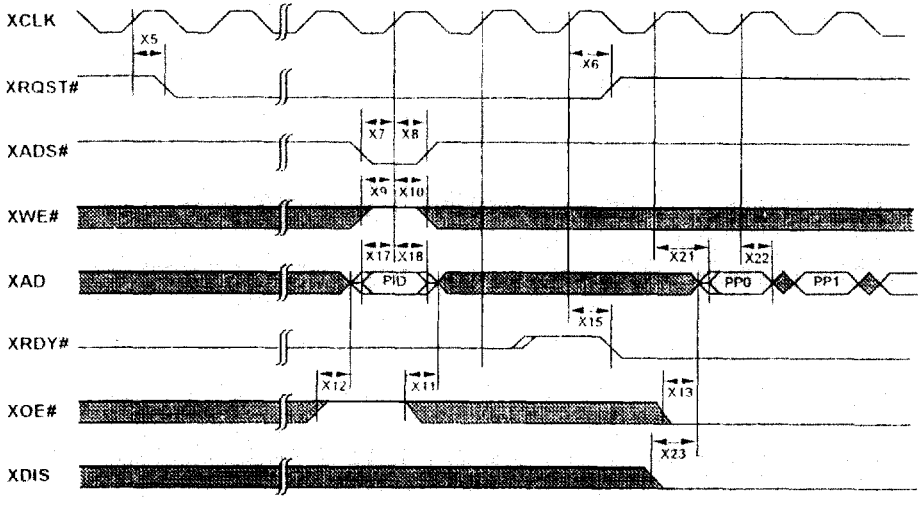


Figure 9.24 - XBUS Read Cycle Waveforms (end)

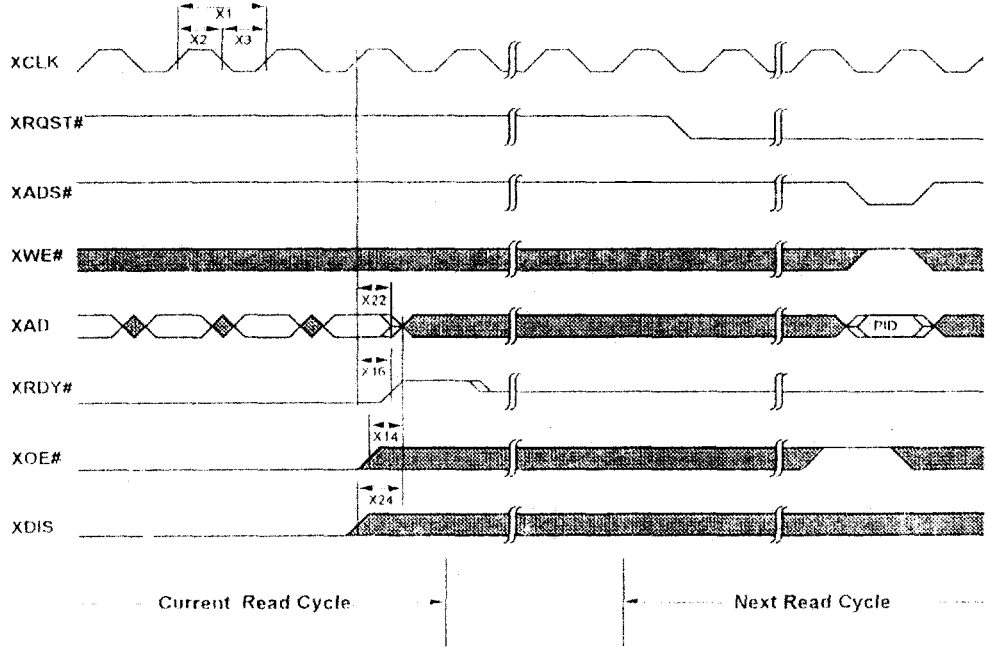


Figure 9.25 - XBUS Write Cycle Waveform

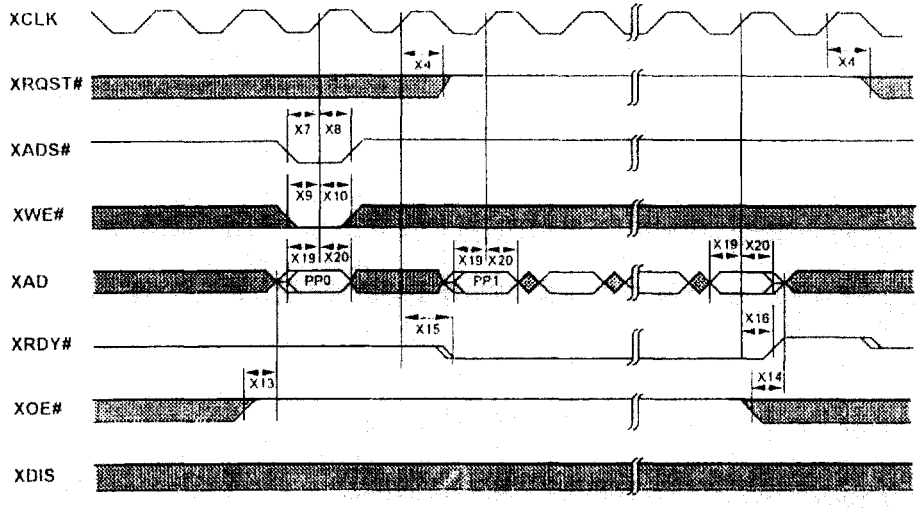


Table 9.8 - Test Bus Timing Parameters

Identifier	Symbol	Description	KS0143CQ-50 Min	KS0143CQ-50 Max	KS0143CQ-40 Min	KS0143CQ-40 Max	Units	Notes
Test Bus Clck								
T1	tc(TCLK)	Test bus clock period			100		ns	
T2	tw(TCLKH)	Test bus clock high pulse width			40		ns	
T3	tw(TCLKL)	Test bus clock low pulse width			40		ns	
	tr(TCLK)	Test bus clock rise time				4	ns	1
	tf(TCLK)	Test bus clock fall time				4	ns	1
Test Bus Timing								
T4	tsu(TI)	TMS, TDI setup time			10		ns	2
T5	th(TI)	TMS, TDI hold time			10		ns	2
T6	ten(TDO)	Test clock low to TDO driven			4	30	ns	3
T7	tpd(TDO)	Test clock low to TDO delay			4	30	ns	3
T8	tdis(TDO)	Test clock low or reset pulse low to TDO tri-state			4	30	ns	1,3
T9	tw(TRSTL)	Test reset low pulse width			100		ns	

Notes:
1. Parameter is guaranteed by design and characterization data, but not tested.
2. Input setup/hold times are with respect to the positive edge of TCLK.
3. Output delay times are with respect to the negative edge of TCLK.

Figure 9.26 - Test Bus Timing Waveforms

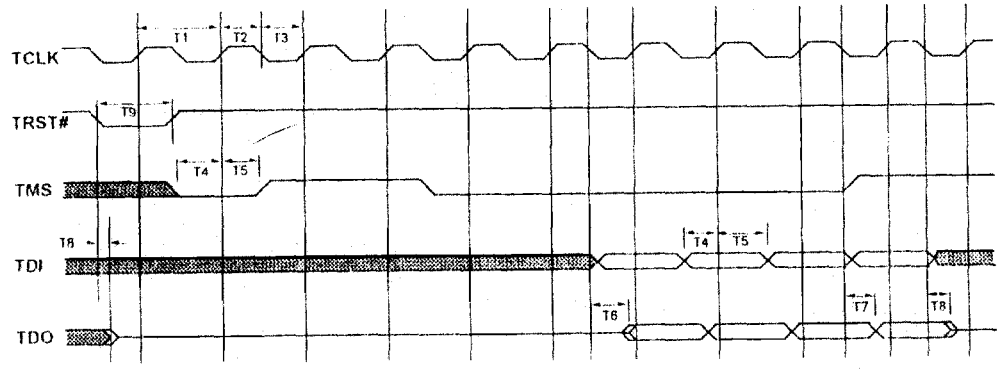


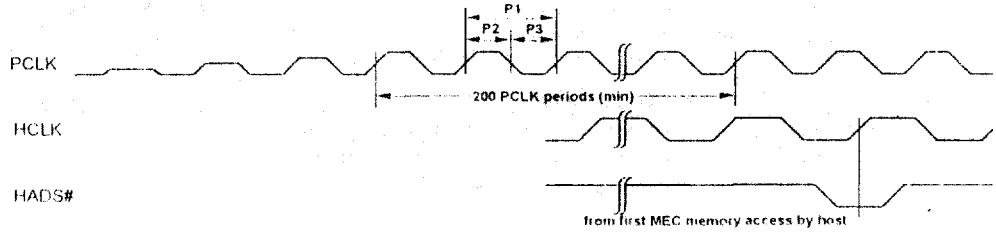
Table 9.9 - Processor Clock (PCLK) Timing Parameters

Identifier	Symbol	Description	KS0143CQ-50 Min	KS0143CQ-50 Max	KS0143CQ-40 Min	KS0143CQ-40 Max	Units	Notes
Processor Clock								
P1	tc(PCLK)	Processor clock period			25		ns	
P2	tw(PCLKH)	Processor clock high pulse width			10		ns	
P3	tw(PCLKL)	Processor clock low pulse width			10		ns	
	tr(PCLK)	Processor clock rise time				4	ns	1
	tf(PCLK)	Processor clock fall time				4	ns	1

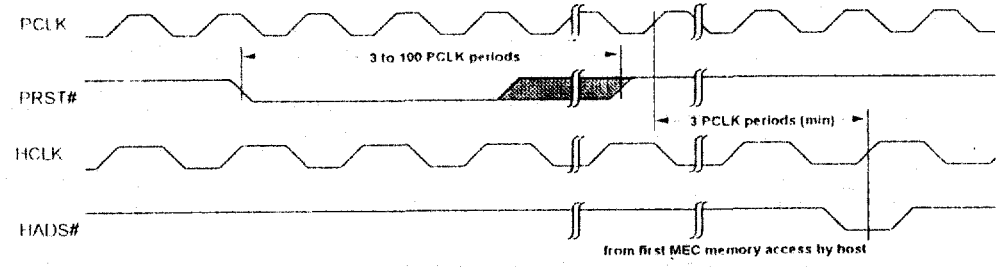
Notes: 1.Parameter is guaranteed by design and characterization data, but not tested.

Figure 9.27 - Reset Waveforms

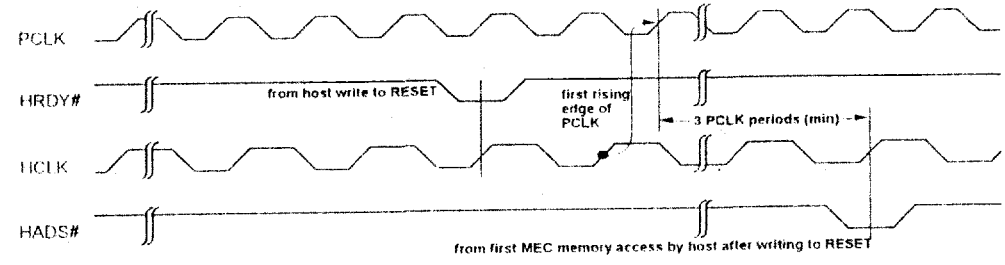
(a) Power-up Reset (Internally Generated)



(b) Hardware Reset (i.e.PRST# Pin Asserted)

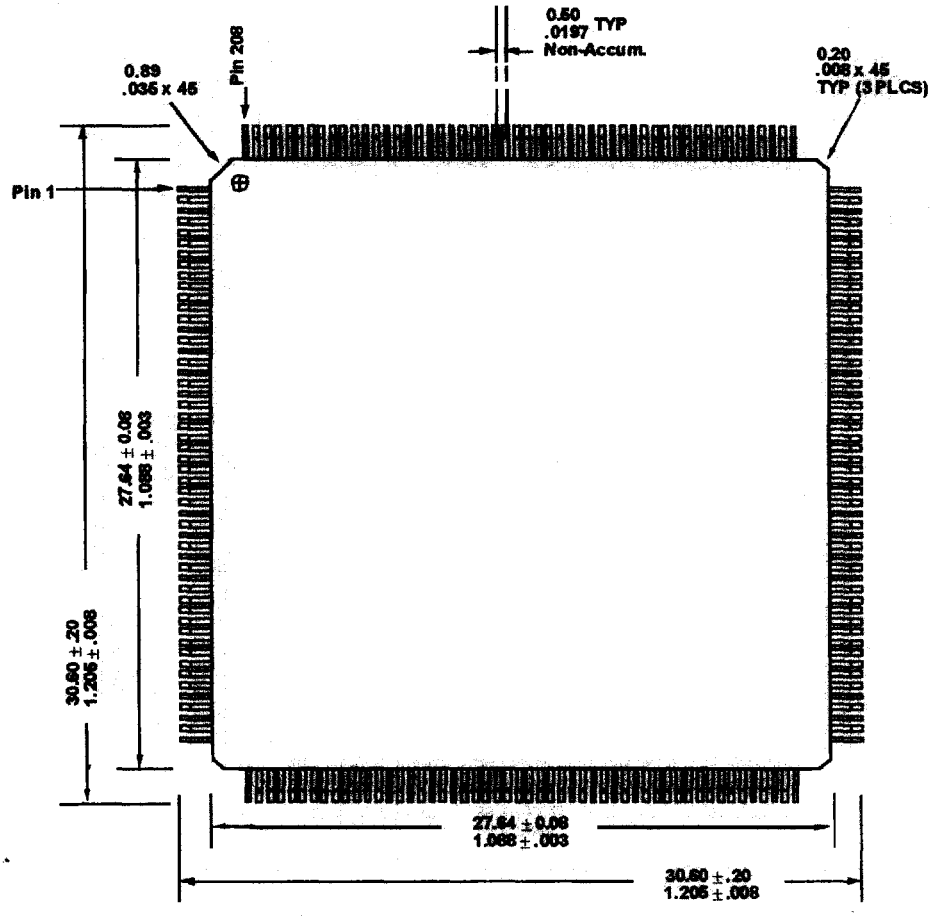


(c) Software Reset (i.e. Host Write to RESET)



SECTION 10 - MECANICAL SPECIFICATIONS

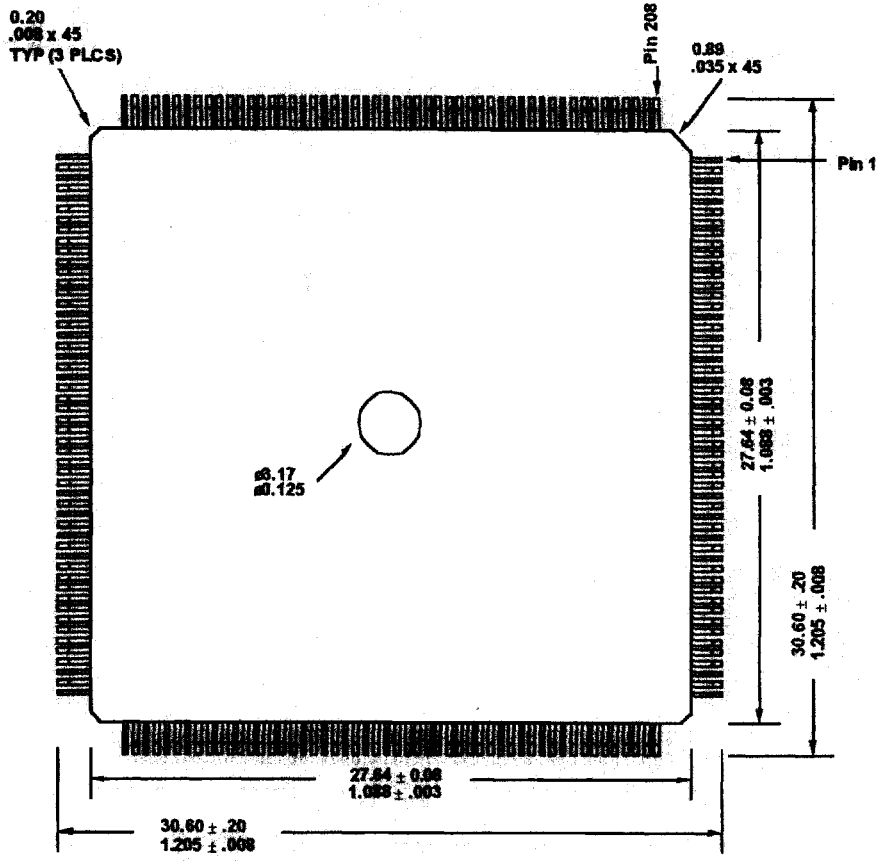
Figure 10 .1- Package Dimensions (top view, cavity down)



Notes:

1. Dimensions are in mm inches
2. Controlling dimensions are in mm
3. Package tolerance is ± 0.08 unless otherwise stated ± .003
4. Tolerance of leadframe to package offset is ± 0.10 max ± .004.

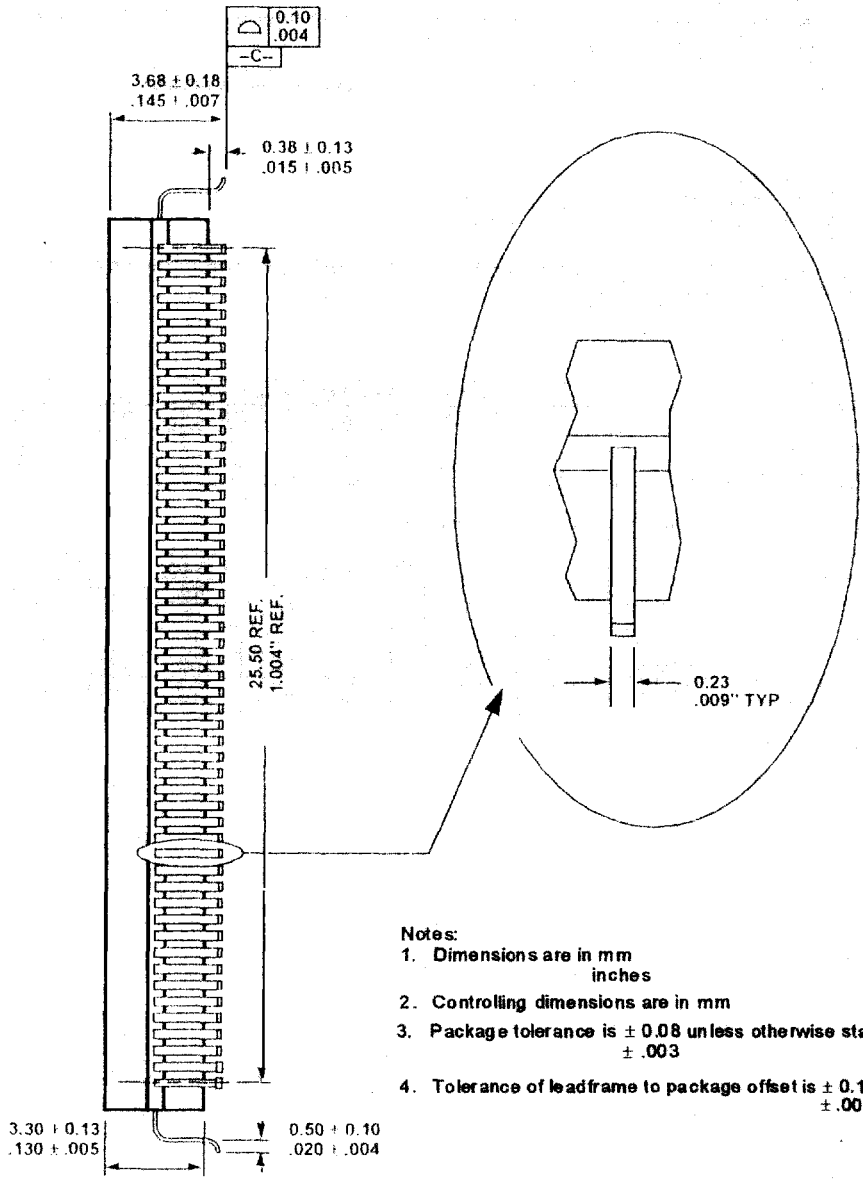
Figure 10.2 - Package Dimensions (bottom view, cavity down)



Notes:

1. Dimensions are in mm inches
2. Controlling dimensions are in mm
3. Package tolerance is ± 0.08 unless otherwise stated ± .003
4. Tolerance of leadframe to package offset is ± 0.10 max ± .004.

Figure 10.3 - Package Dimensions (side view, cavity down)



Notes:
1. Dimensions are in mm
   inches
2. Controlling dimensions are in mm
3. Package tolerance is ± 0.08 unless otherwise stated
   ± .003
4. Tolerance of leadframe to package offset is ± 0.10 max
   ± .004

Package Thermal Specifications

The MSP96300 is specified for operation when the case temperature (Tc) is within the range of 0× C to 100× C. The case temperature is measured at the center of the top surface of the package.

The maximum allowable ambient temperature (TA) can then be calculated using the following equation:

TA [max] = TC [max] - (P * (q JA - q JC))

where: P = Maximum Power Dissipation
q JC = Thermal Resistance from junction to case
q JA = Thermal Resistance from junction to ambient

The following tables give the thermal resistance parameters and the maximum allowable ambient temperatures for the 208 MQUAD™ package under various airflow conditions.

Table 10.1 - 208 MQUAD Thermal Resistance (℃/Watt)

Parameter	Airflow (Linear Feet Per Minute)						
	0	100	200	400	600	800	1000
θ JC (junction -to-case)	3	3	3	3	3	3	3
θ JA (junction -to-ambient)	21	18	16	12	9	7	6

Table 10.2 - Maximum Ambient Temperature at Various Operating Conditions (℃)

Parameter	f_{POLK}	Airflow (Linear Feet Per Minute)						
		0	100	200	400	600	800	1000
TA	50 MHz	10	25	35	55	70	80	85
(without heat sink)	40 MHz	28	40	48	64	76	84	88
TA	50 MHz	TBD	TBD	TBD	TBD	TBD	TBD	TBD
(with heat sink) **	40 MHz	TBD	TBD	TBD	TBD	TBD	TBD	TBD

** = Heat Sink to be determined