DOCUMENT SOURCE REFERENCE	
HARDWARE	Pentium PC
SOFTWARE	FrameMaker 5.5.6
The electronic database for this document is maintained by the ASIC group. Please contact the ASIC group regarding any additions or corrections.	

Genii

Preliminary

CAUTION

PLEASE USE AN APPROPRIATE MEANS OF DISPOSAL (A DOCUMENT SHREDDER OR EQUIVALENT) WHEN DISCARDING THIS PROPRIETARY DOCUMENT.

REV	DESCRIPTION	DATE	ECO	SIGN
Preliminary (b)	IOC1-3,Parsers	6/1/99		
Preliminary (a)	Initial draft, based on Genesis	4/1/99		

Quantum

By: **Date** 6/1/99

ASIC Group Leader Trung Nguyen

Buffer Control, Bus Translator, UPI	Hang. Nguyen
PAT, PMT Parsers	Gaetano Bofiglio
Stream Control	T. Nguyen
Vide/Packeto Parser, Table Generation, Timestamp Control	Geoffrey Yung
Trick Play Control	Kaixin Song
3221Q Interface	M. Uppuluri
Documentation	R. C. Ayeras

DOCUMENT TITLE

Specification, Design **ASIC GENII**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GENII ASIC
VOLUME 1 - FIRMWARE

Volume 1 contains the GENII Firmware Specification

CAUTION

PLEASE USE AN APPROPRIATE MEANS OF DISPOSAL (A DOCUMENT SHREDDER OR EQUIVALENT) WHEN DISCARDING THIS PROPRIETARY DOCUMENT.

DOCUMENT NUMBER	REVISION	
Genii 70-113287-01	Preliminary (b) 6/15/00	

Volume 1 Section 1
GENERAL DESCRIPTION

1.1 GENII CAPABILITIES AND FEATURES

- Supports Analog in and out system.
- Two MPEG-2 TS streams in, one MPEG-2 TS stream out.
- 8-bit interface for MPEG-2 Rs, three different styles:
  - DVB SPI-LIKE 8-bit interface.
  - PHILIPS Demux 8-bit interface.
  - STI 5510 8-bit interface.
- Host translator from NEC850E to Motorola and to Intel microprocessor to control Analog input/output system peripheral components.
- Generates Time stamp to control bit rate, and the accuracy of PCR.
- Interfaces with Indigo/Ivory through new By Pass mode.
- MPEG-2 TS stream trick play support: fast forward; fast reverse.
- Provides interface to DRAM for SONY-3221Q through Genii.

1.2 SYSTEM ENVIRONMENT

Figure 1-1: shows the top level block diagram of Genii architecture. On one side, Genii interfaces with the hard disk controller chip as an interface host (similar to the SCSI or ATA interface block). This transfers data from the host to the buffer RAM, and vice versa. On the other side, Genii interfaces with either 1394 LINK for digital I/O, or with analog I/O — to be specified later. Genii can also interface to the SONY-3321Q chip.

The overall control of the Genii chip is done through the NEC850e microprocessor, which is also common to the disk controller chip.

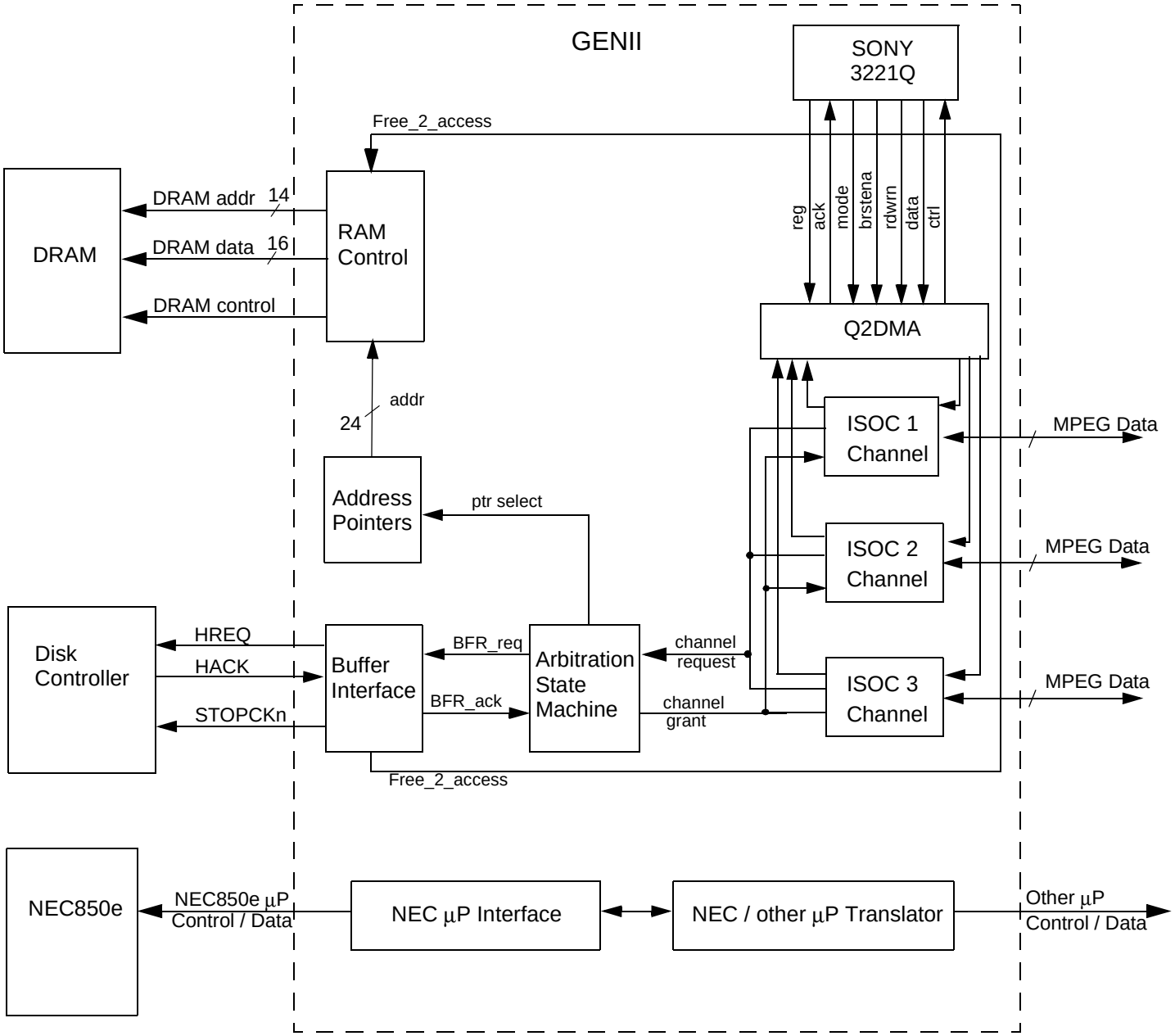


Figure 1-1: Genii Block Diagram

1.2.1 NEC Microprocessor Interface

This block is the interface between firmware and Genii. Prior to using the chip, firmware needs to initialize Genii's registers and the host interface (1394 link/phy, or other interface) through this block. When Genii needs to report an error or status, this block generates the interrupt to the microprocessor.

1.2.2 NEC/Other Microprocessor Translator

As the host interface chip is external, not controlled by Genii, this block translates NEC850e timing to the other microprocessor timing. Example: In case of TI 1394 link, Motorola 68000 timing is provided.

This block has a simple state machine to map the register read/write timings from one microprocessor to another.

1.2.3 ISOC Channels: 1, 2, 3

ISOC channels 1, 2, and 3 are identical. Each channel can be programmed as either input or output. The input/output stream is in AV format: MPEG-2/DSS/DV.

Each channel has a 512 x16 FIFO that buffers data between the SDRAM and the AV interfaces. Each channel also keeps track of “sector available” and “sector threshold”, which are associated with the architecture of the disk controller integrated circuit. When a channel needs to transfer data to/from its FIFO, that channel generates a request to the arbitration state machine, then waits for its turn to transfer.

There are three main blocks in the Iso. channel: Time Stamp Control, Stream Control, Trick Play Control. See Figure 1-2:

The Time Stamp Control tracks the internal time stamp for AV stream (MPEG-2/DSS/DV), and store the information in the drive. The Time Stamp Control reconstructs AV stream timing while transmitting to the AV application.

The Stream Control handles AV data delivered to the drive, and received from the drive.

The Trick Play Control performs the trick play mode for AV stream with MPLEG-2/DV format. To support trick play function, Iso. Channel de-multiplexes the PSI from MPEG-2 TS (Transport Stream), and decodes TS header, PES header, Sequence header, GOP header, and Picture header. This information is packed with the AV data stream, in AV data format, for each sector before writing to the drive. While transmitting data out, Trick Play Control manipulates headers and drop frames to perform the trick play function. To track where GOP headers for different trick play modes, Table Generation block asserts interrupt to the microprocessor, saves LSA value, and stores the information via firmware in EFS. This stored data can be used later: e.g., for random access.

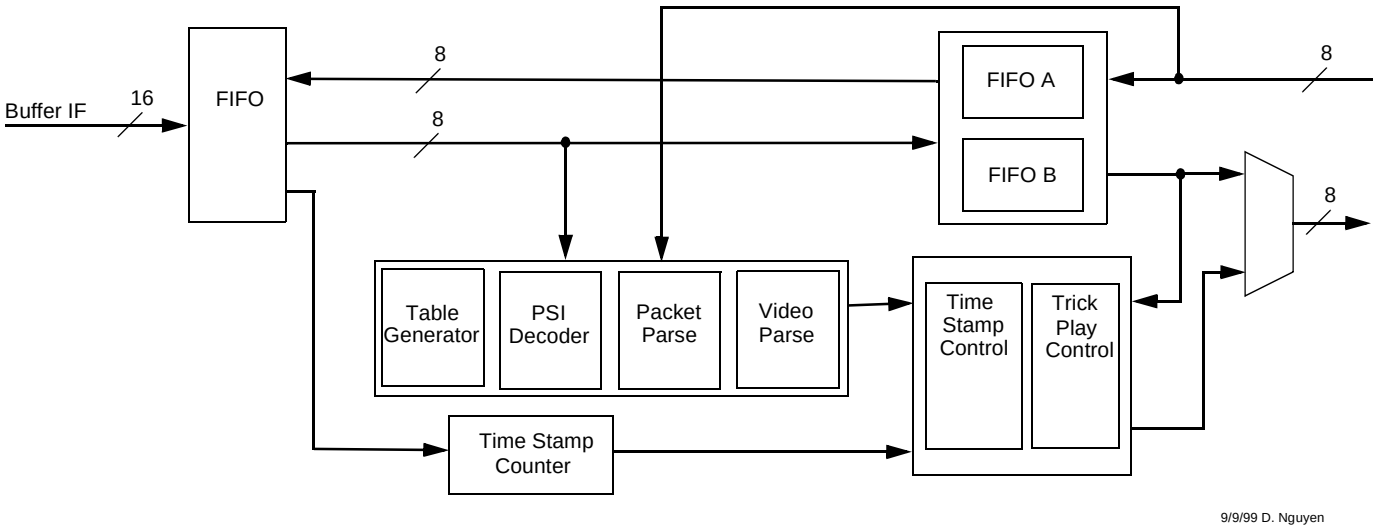


Figure 1-2: ISO Channel Block Diagram

1.2.4 Arbitration State Machine

The Arbitration State Machine determines which ISOC channel gets the right to access DRAM. This arbitration scheme is round robin. The state machine goes through the round to check for request from each ISOC channel. When a request is detected, that channel is served until it is done requesting. Then, the next channel is checked for request.

When serving an ISOC channel, the state machine selects the appropriate address from the address pointers block, and talks to the buffer interface block to acquire the right to access DRAM.

1.2.5 Buffer Interface

The Buffer Interface handshakes with the disk controller chip to ask for permission to access DRAM. Once the disk controller acknowledges Genii the right to use DRAM, it releases all DRAM control pins, and the buffer interface block informs the rest of the chip that it is Genii's turn to read/write to DRAM.

Genii always acts as a slave; the Controller chip can always preempt Genii when it requires access to DRAM. In this case, the Buffer Interface concludes its current DRAM access as soon as possible, then releases all DRAM control pins.

1.2.6 Address Pointers

The Address Pointer tracks four sets of DRAM address pointers for the four ISOC channels. Each ISOC pointer has two sets of re-load and roll-over addresses. This block also tracks the DRAM page boundary for the RAM Control block.

1.2.7 RAM Control

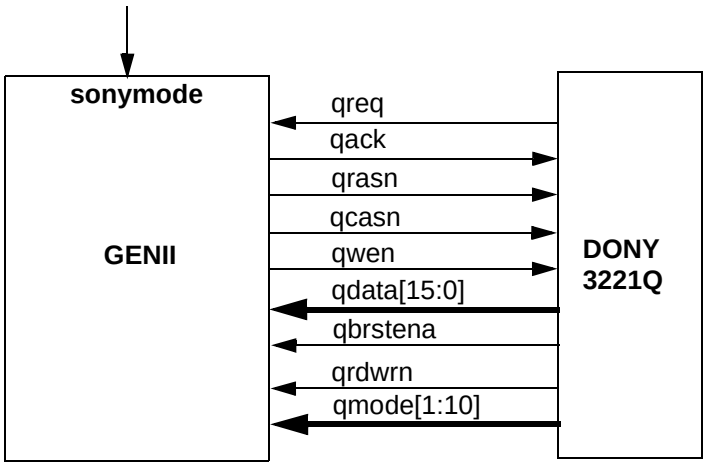
The RAM Control generates all control signals to DRAM devices. After it receives the OK signal from the Buffer Interface, RAM Control uses its internal state machine to generate appropriate RAS and CAS timing for DRAM.

1.2.8 3221Q To DMA Interface (Q2DMA Interface)

This interface enables the 3221Q chip to talk to the Ivory chip, and transfer data to the DRAM. 3221Q was designed to interface to the DRAM with a REQ-ACK mechanism, through the Disk Controller ASIC. However, this interface on that ASIC has been changed since the IVORY generation. The Q2DMA block in Genii now handles this data transfer between 3221Q and Genii DRAM through the DMA channels in Genii.

Based on the mode selected by 3221Q, data is transferred into the appropriate DMA FIFO, through a small, dedicated 32 x 16 FIFO in the Q2DMA block.

See Fig. 1-3and Fig. 1-4.



Design notes:

1: Provide Sony's 3211Q chip a path to the SDRAM through the Ivory/Indigo ASICs
2: 3321Q was designed to work with RAINIER_S

M. Uppuluri 11/19/99

Figure 1-3: Q2 DMA Interface, 3211Q To AVDMA

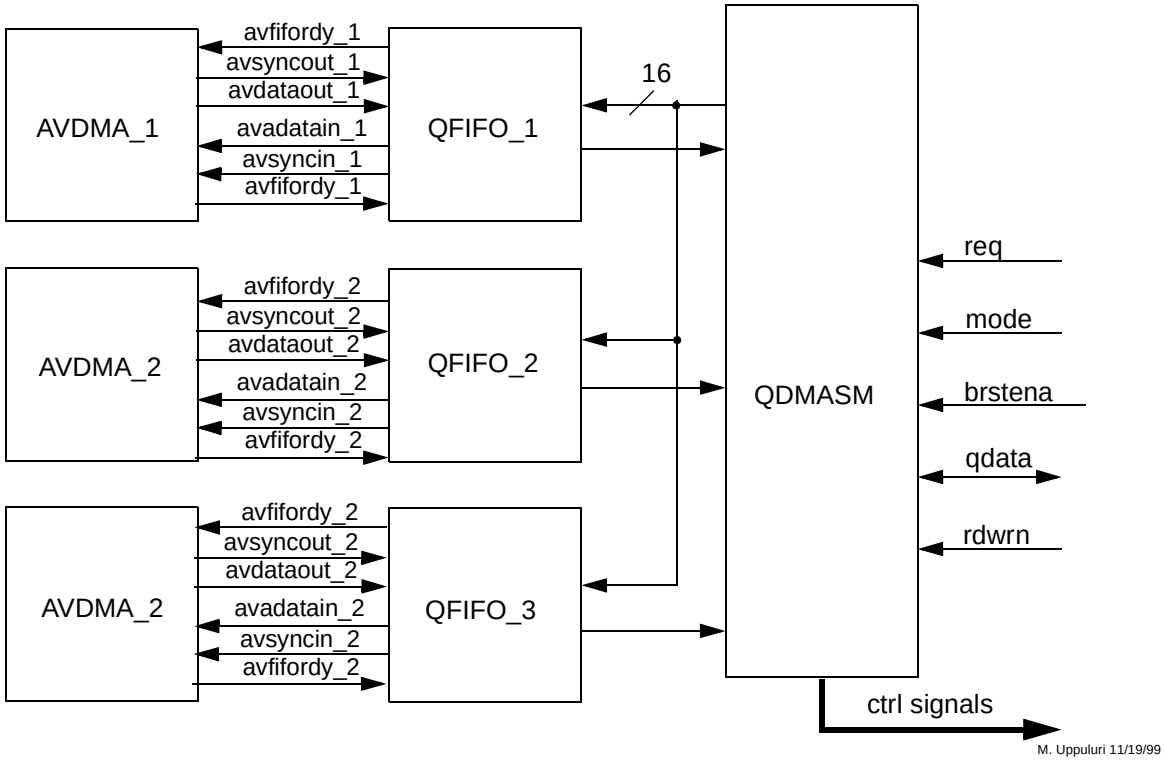


Figure 1-4: Q2 DMA Interface, Block Diagram

Volume 1 Section 2
TRICK PLAY CONTROL

2.1 IMPLEMENTATION OF TRICK PLAY IN MPEG-2 TRANSPORT STREAMS

Types of trick play include fast forward, fast reverse, slow motion, and freeze frame. The details of these trick plays implementation will be explained in the later sections. The major approaches to achieve trick play are skipping frames, and modifying time stamps i.e. PTS and DTS. Other fields may also need to be manipulated, which is explained in the following sections.

During trick play, any sounds being played back are simply annoying; all the audio data packets are removed during trick play.

Mpeg-2 transport stream is created by combining one or more elementary streams of video, audio and other data into a single stream. Each transport stream packet consists of 188 bytes in which the first 4 or more bytes is the header, followed by the payload. The payload for transport stream packets may be in the form of Elementary Streams (ES) or Packetized Elementary Streams (PES).

Transport Stream rate can vary. Program Clock Reference (PCR) is used as a time base. The timing information of the data is available in the Presentation Time Stamp (PTS) and Decoding Time Stamp (DTS). PCR timing information is contained in the adaptation field of the transport packet header.

The length of the adaptation field can vary from 1 to 183 or 184 bytes. The length depends on the adaptation_field_control value. The adaptation field may contain fill bytes only (no adaptation information). The PTS and DTS may be present in the PES packet header.

2.1.1 Fast Forward

Fast forward functions can be implemented by two methods.

- Display I-frames only;
- Skip group of pictures (GOP).

2.1.1.1 I-frames Only

All B-frames can be dropped to achieve faster rate of play. If all B-frames are dropped, then all P-frames can also be dropped for a very fast I-frame only play.The speed multiplier is a function of the frame type sequence within the stream.

The following fields in the transport stream need to be modified for fast forward (I only):

PES length

PES length is set to unbound (zero). Since all the B and P frames are dropped, the original value of PES length is no longer valid.

Continuity Counter

Adjustments are needed to compensate the discontinuities in the continuity counter field of the transport packet header. This is done by substituting the field with a locally generated value.

Before restoring to normal operation, some dummy packets may be inserted to compensate for the difference between the internal continuity counter and that of the first normal play video packet.

See Figure 2-1 on page 17.

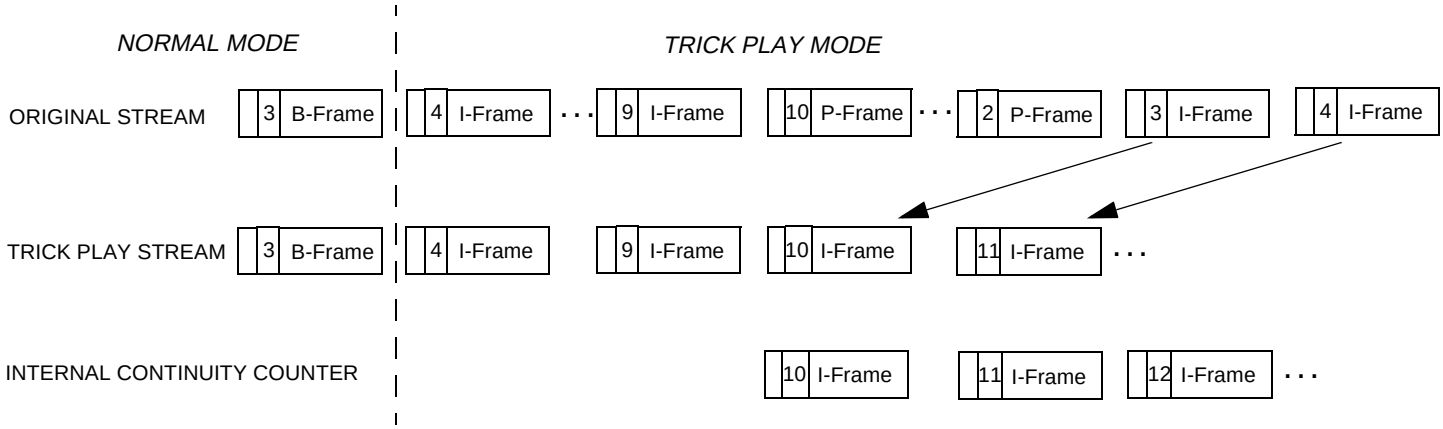


Figure 2-1: Modes: Normal, Trick Play

Drop B and P frames

Drop all the packets that only have B and P frames. However, frames can start in the middle of a transport packet. If a packet has partial I frame in it, the non-I frame part must be filled with 0xFFh.

Drop audio packets

All the audio packets will be dropped in trick play.

PAT & PMT

PAT and PMT packets are sent per the maximum spacing requirements between occurrence of PMT (PAT) table. Example: The PAT table is sent every 100ms.

Low delay bit

Set this bit in trick play to indicate that there is no B frame.

PCR

Adjustments have to be made to the PCR field in the Transport Packet Header adaptation field. The new PCR field value needs to reflect the actual timing information as it relates to the accelerated/ time-shifted data transmission.

This can be done by substituting the PCR field with a local free running counter clocked by an on-board, high accuracy oscillator (27Mhz clock).

PTS, DTS and Bit Rate Control

Adjustments have to be made to the PTS and DTS fields in the PES header to track the new time-base during trick play. Also, bit rate from drive must be controlled to not over-run or under-run the decoder's buffer. These can be done as follows:

- There is a frame delay register programmed by firmware. The value of this register indicates the minimum frame spacing between two consecutive I frames to be displayed. Example: The value 3 indicates that if the first I frame is displayed at time T0, the second I frame should be displayed at time T3. In other words, ten I frames should be displayed in one second (for 525-60 system, 30 frames/second).
- Once trick play starts, hardware will start to send one I frame of data to the decoder as soon as possible, and set the PTS and DTS value to be the time after frame delay period. Example: If trick play starts at time T0, and the frame delay register programmed to be 3, then the PTS and the DTS should be set to T3.
- Hardware always starts to send the next I frame to decoder when the PTS and DTS time of the previous I frame is reached.

Discontinuity Bit

Before switching back to use the original stream's PCR, the discontinuity bit (in adaptation field) must be set to indicate the discontinuity of the PCR.

Broken Link Bit

This bit resides in the GOP sequence header. It indicates to the decoder to ignore all B frames in the current GOP that depended on P or I frame from previous GOP.

The Broken Link bit must be set before restoring to normal operation mode.

2.1.1.1 Skipping GOPs

Fast forward play can also be implemented by dropping (N-1)/N GOPs to achieve N speed multiplier. For instance, N = 2, dropping every other GOP, results in 2x fast forward.

The following fields in the transport stream need to be modified:

PES length

Refer to PES modification in section I-frames Only, page 16.

Continuity Counter

Refer to Continuity Counter modification in section I-frames Only, page 16.

Drop GOP

The number of GOP that needs to be dropped depends on the speed to achieve. Example: 2x play can be accomplished by skipping one out of two GOPs.

Hardware searches for the GOP header and determines which GOP should be dropped. This is based on the desired speed to achieve. To drop certain GOP, all the packets within that GOP should be dropped. However, if a packet has partial data that belongs to another GOP, that packet must also be sent to the decoder, with the data portion that needs to be dropped filled with 0xFFh.

Drop audio packets

All the audio packets are dropped in trick play mode.

PAT & PMT

Refer to PAT & PMT modification in section I-frames Only.

PCR and Discontinuity Bit

The PCR field does not need to be modified. The internal counter re-syncs to the PCR value within the stream.

The discontinuity bit needs to be set at each GOP where the PCR change occurs.

PTS and DTS

It is not necessary to adjust the PTS and DTS field in remaining GOP packets.

Broken Link Bit

Broken Link bit must be set in every GOP header.

2.1.1.2 Comparisons

Displaying I-frame only results in a very high rate fast forward.

For an example, consider a video stream with one I-frame present in every 15 frames. In this case, 14 frames will be skipped in the trick play. The result will be a 15x fast forward, and it may be hard to follow the pictures in the video stream. Some video streams may have one I-frame present in every 30 frames. At 30x fast forward, is a lot of information is lost, and some scans may never be seen.

Using the method of skipping GOP, a lower rate of fast forward such as 2x can be achieved. However, the pictures will be jerky for those video streams that have large GOPs, such as 15 or 30 frames.

2.1.2 Fast Reverse

This is done by dropping all the audio and B, P frames, and reordering I frames. The difference between fast forward and fast reverse is the re-ordering of I frames.

Since the re-ordering of I frames is handled by firmware, the same hardware to implement fast forward can be used to implement fast reverse.

2.1.3 Slow Forward

There are two methods for implementing slow forward:

- One is modifying time stamps
- one is inserting zero-difference P-pictures.

2.1.3.1 Modifications of Time Stamps

Slow forward can be implemented by modifying PTS and DTS values. PTS controls the time when the frame is being displayed, while DTS only represents the time when the frame is decoded. PTS is very important in manipulating trick play. For some video clips, not every frame has a PTS attached to it.

If only the frames with PTS are modified, the result of trick play will be very jerky. In order to solve this problem, additional packets containing PTS information can be inserted into the frames that do not have time stamps.

The formula used to calculate the new values of the time stamps (TS):

$$New\ TS = DTS_1 + (old\ TS - DTS_1) \times R$$

where DTS_1 is the DTS of the first I-frame happened in the trick play, and R is the speed factor to indicate the speed of the slow motion (e.g. R = 3 will give a 1/3x speed)

An example follows:

Picture pattern: $I_1\ B_1\ B_2\ P_1\ B_3\ B_4\ P_2\ B_5\ B_6\ I_2\ B_7\ B_8$ … …

Table 2-1: and Figure 2-2: show an example of transmitted for normal playback:

Table 2-1: Normal Playback

	I_1	P_1	B_1	B_2	P_2	B_3	B_4	I_2	B_5	B_6
PTS	t+1	t+4	t+2	t+3	t+7	t+5	t+6	t+10	t+8	t+9
DTS	T	t+1	-	-	t+4	-	-	t+7	-	-

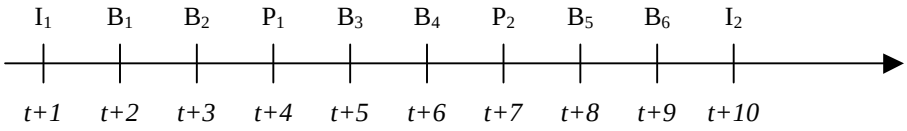


Figure 2-2: Normal Playback

Table 2-2: shows an example of transmitted for half speed playback (R = 2)/

Table 2-2: Half Speed Playback

	I_1	P_1	B_1	B_2	P_2	B_3	B_4	I_2	B_5	B_6
PTS	t+2	t+8	t+4	t+6	t+14	t+10	t+12	t+20	t+16	t+18
DTS	T	t+2	-	-	t+8	-	-	t+14	-	-

Figure 2-3: shows an example of slow motion playback.

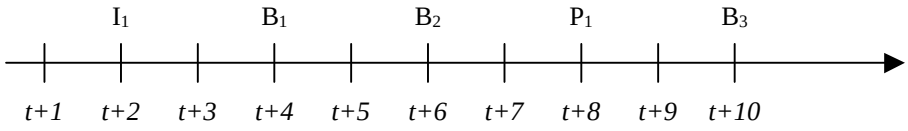


Figure 2-3: Slow Motion Playback

When the frame does not have a PTS attached to it, it's necessary to insert an additional PES packet prior to that frame to indicate its PTS value. This will be a 188-byte transport packet with a PES packet as its payload.

2.1.3.1 Insertion of Zero-difference P-pictures

Another approach to implement slow forward is by inserting zero-difference P-pictures between original frames. Since the zero-difference P-pictures simply give no motion vector to the previous frame, the previous frame stays for a longer time if the frame rate is unchanged. However, to use this approach, all the B-frames must be replaced by zero-difference P-pictures during trick play: When B-frames are present in the video clips, proper frames re-ordering is needed.

2.1.4 Freeze Frame

There are two methods for implementing freeze frame. One is repeating I-frames and the other one is inserting zero-difference P-pictures.

2.1.4.1 Repeating I-frames

The "freeze frame" function can be implemented by repeating the same I-frame over and over again.

For instance, an original clip as shown in Figure 2-4: :

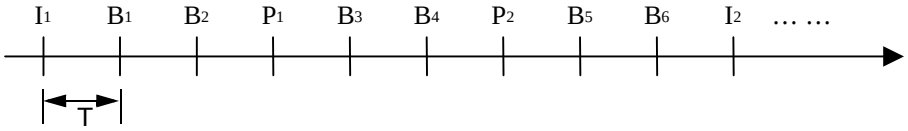
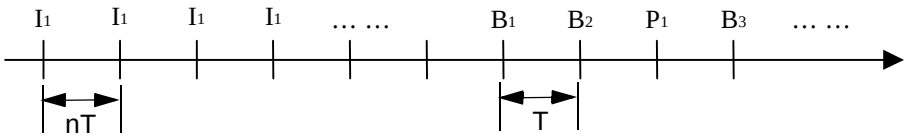


Figure 2-4: Original Clip

To freeze at I_1, I_1 has to be transmitted many times, as shown in Figure 2-5: .



nT: Where n is programmed. (See the descriptions on the following page.)

Figure 2-5: Transmitted Clip

The following modifications are needed to implement this trick play:

- **PES Length:** Refer to section I-frames Only
- **Continuity Counter**:Refer to section I-frames Only
- **Partial I-frames:** Frames can start and end in the middle of a transport packet. To align with the 188-byte transport packet boundary while repeating I-frame, it is necessary to fill any non-I-frame part in the transport packet with 0xFFh.
- **Drop Audio:** Refer to section I-frames Only, page 16.
- **PAT & PMT:** Refer to section I-frames Only, page 16.
- **Low Delay But:** Refer to section I-frames Only, page 16.
- **PCR:** When transmitting all the additional I1 frames, the PCR values has to be updated using an internal PCR counter. It is not recommended to repeatedly send the same PCR (refer to PCR modifications in I-frames Only, page 16). When trick play ends, the original PCR in the video clip is used.
- **PTS, DTS and Bit Rate Control:** Refer to section I-frames Only. Since I-frames are much larger in size compared to other frames, transmission exceed the bit rate limit if all these I-frames are delivered at the original frame rate. Firmware will have to calculate a new frame rate according to size of the I-frame, and then adjust the value of PTS and DTS.
- **Discontinuity Bit:** Refer to section I-frames Only, page 16.
- **Broken Link Bit:** Refer to section I-frames Only, page 16.

2.1.4.1 Insertion of Zero-Difference P-pictures

Another approach of freezing a frame is to insert zero-difference P-pictures after an I-frame. Since the zero-difference P-pictures simply give no motion vector to the previous I-frame, the previous I-frame will stay on the screen as long as zero-difference P-Pictures (P_{zero}) are transmitted. Therefore, the more P_{zero} added, the longer the I-frame will stay on the screen. This results in a long pause.

For instance, the original clip is like this:

To freeze at I_1, P_{zero} are added after I_1.

If the frame rate is 30 frames/sec., 30 P_{zero} can only freeze the I-frame for 1 second. Details on how to generate the zero-difference P-pictures are shown in Appendix A.

When inserting zero-difference P-pictures, the following modifications need to be made:

PES Length

Refer to section I-frames Only, page 16.

Continuity Counter

Refer to section I-frames Only, page 16

Partial I-frames

For the last transport packet containing the I-frame, it is necessary to fill any non-I-frame part with 0xFFh so as to align with the 188-byte transport packet boundary.

Drop Audio

Refer to section I-frames Only, page 16.

PAT & PMT

Refer to section I-frames Only, page 16.

Low Delay Bit

Refer to section I-frames Only, page 16.

- **PCR:** When transmitting all these additional P_{zero} frames, the PCR values has to be updated using an internal PCR (refer to PCR modifications in I-frames Only) When trick play ends, the original PCR in the video clip is used.

PTS, DTS and Bit Rate Control

Refer to section I-frames Only

Discontinuity Bit

Refer to section I-frames Only

2.1.5 Step Forward

2.1.6 Step Reverse

2.2 TRICK PLAY CONTROL STATE MACHINE

Figure 2-6: shows the block diagram of the Trick Play Control State Machine

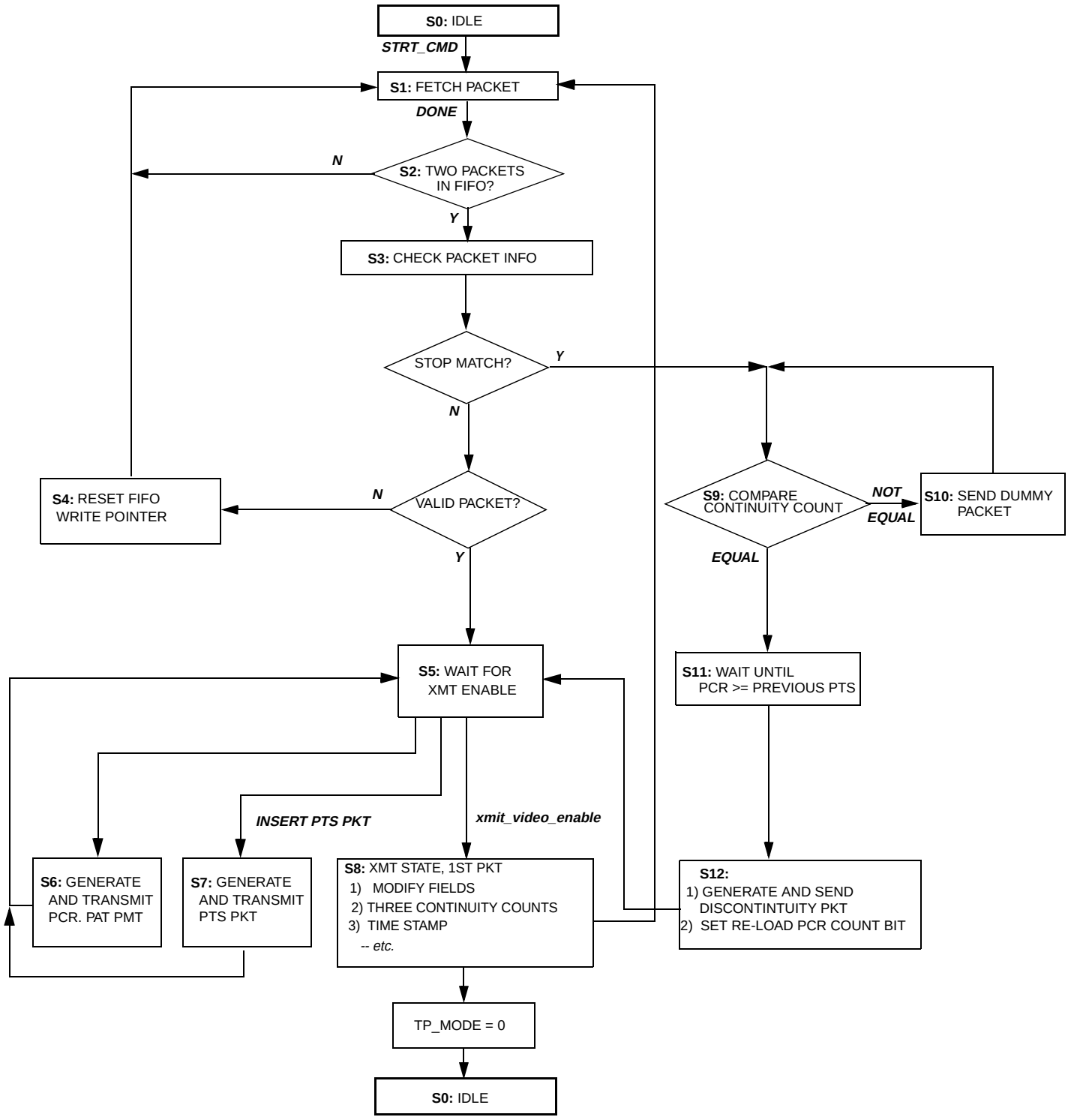


Figure 2-6: Trick Play 1 Control State Machine

2.3 TRICK PLAY CONTROL CONFIGURATIONS

Types of trick play that are supported in Genii ES1 include fast forward and fast reverse (I frame only) for MPEG2 transport stream. The details of these trick plays implementation are explained in previous chapters of this section.

2.3.1 Register set up Start Trick Play

The major approach to achieve trick play is by skipping frames. Since all the B and P frames are dropped in trick play, the original value of PES length become invalid. Firmware must continue to provide the original stream (must not start to drop sectors), even after issuing Trick Play Start Command until Trick Play detects and unbounds the PES length in the MPEG2 transport stream. Rollover address register, pair A or pair B, can be used to accomplish this.

Firmware can program Rollover address register to let Trick Play Control have enough space to search for PES header, and also use this to let Trick Play Control know when Firmware starts to drop sectors.

Example:

Firmware issues Trick Play Start Command at Location A, and continues to provide normal stream until reaching location B. There are two I frames within location A and B. This should be long enough fro Trick Play Control to detect PES header. Firmware can program the RollAValid (or RollBValid) bit as '1' to indicate which pair of rollover address registers to use for this rollover. Also, program the associated channel3 rollover register pair A (or pair B) to be the address of location B.

Trick Play Control starts fast forward when the rollover pair A (or pair B) address hit by knowing the firmware will start dropping sectors after that.

See Figure 2-7 on page 24.

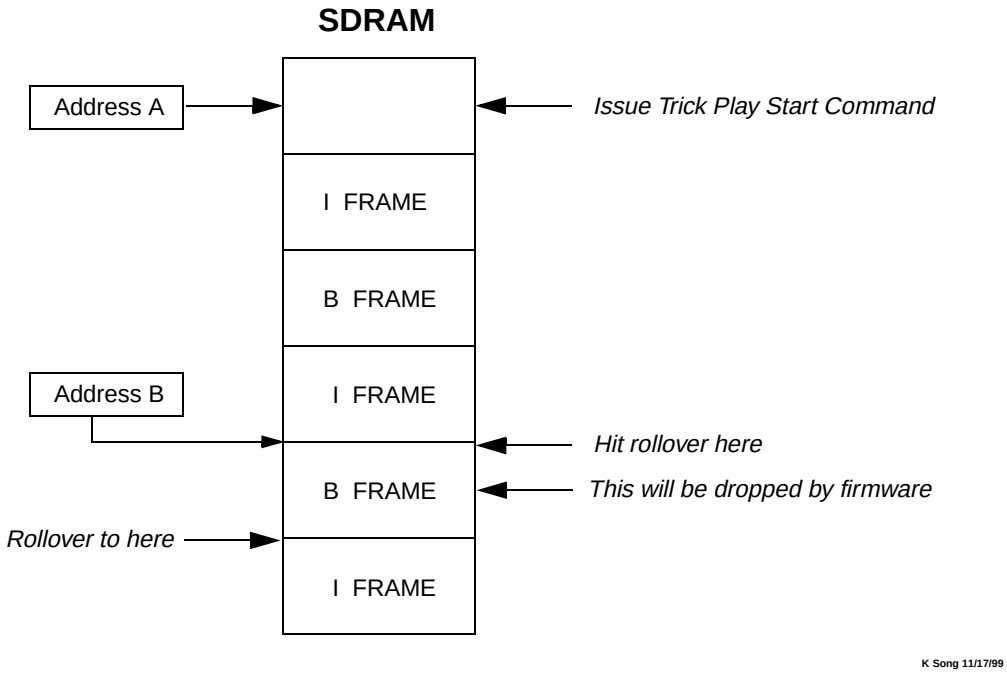


Figure 2-7: Start Trick Play

In summary, the following registers must be programmed to start trick play:

- DMA channel configuration registers: Start normal play as described in DMA channel section. “disable_ts” bit must be ‘0.’
- I3_TPCONFIG: Trick play configuration register, including trick play type, rollAValid or rollBvalid, start command, etc.
- Channl3 rollover registers (pair A or pair B)
- I3_TSTAMPCFG: Control the frame rate during trick play: Refer to time stamp control registers.

2.3.2 Register Setup for Stop Trick Play

After detecting the Trick lay Stop Command, Trick Play Control continues the trick play until detecting that firmware started providing the normal stream (stops dropping sectors). Similar to the start of Trick Play, this can be accomplished by programming the Rollover address register: pair A or pair B.

Example:

Firmware issues Trick Play STOP Command at location C, and continues to provide I frame sectors (dropping sectors containing only B or P frames). Location D has the START of an I frame, and after that, Firmware stops dropping sectors (i.e., provides normal streams). In this case, Firmware can program the RollAValid (or RollBValid) bit as ‘1’. Setting this bit indicates which pair of rollover address registers are being used for this rollover. Also, program the channel3 rollover register pair A (or pair B) to be the address of location D. Trick Play Control stops fast forward when the rollover pair A (or pair B) address hit, from knowing that firmware will provide normal stream after that.

Rollover pair A (or pair B) address register should be programmed to be location D (start of random access unit, either I frame or GOP), not location E.

See Figure 2-8 on page 25.

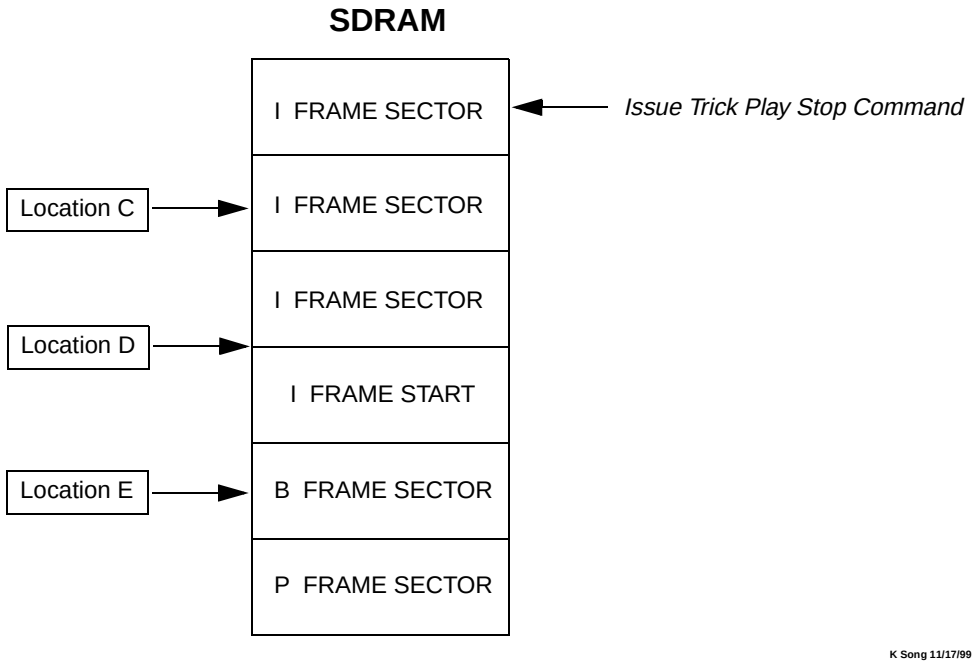


Figure 2-8: Stop Trick Play

The following registers must be programmed to stop trick play:

- I3_TPCONFIG: Including trick play type, rollAValid or rollBValid, stop command, etc.

Note: “disable_ts” bit in DMA configuration register must still be ‘0:’ Trick play **must not** be stopped by setting “disable_ts” as ‘1’.

Volume 1 Section 3
REGISTER DESCRIPTIONS

3.1 DMA CHANNELS REGISTERS

There are three ISOC channels in the Genii chip: All channels are identical in terms of programmable register, and mode of operation.

The following table specifies the register of the ISOC1 channel with its base address. The other ISOC channels, ISOC2 and ISOC3, have the same registers with appropriate base addresses.

Table 3-1: ISOC Channel Base Address

ISOC	Base Address
I1base	FF8380
I2base	FF83C0
I3base	FF8400
TPbase	FF8440
Gen_base	FF8480

3.1.1 ISOC1 Register Addresses

Table 3-2: I1 Register Addresses

REGISTER	ADDRESS	DESCRIPTION
I1_ADDR	8380h	swrstbh[0]
I1_ADPG	8382h	swrstbh[1]
I1_ROLA	8384h	swrstbh[2]
I1_RLDA	8386h	swrstbh[3]
I1_ROLB	8388h	swrstbh[4]
I1_RLDB	838Ah	swrstbh[5]
I1_CONFIG	838Ch	swrstbhl[6]
I1_INTENA	838Eh	swrstbl[7]
I1_INT	8390h	swrstbl[8]
I1_AVDATASIZE	8392h	swrstbh[9]

Table 3-2: I1 Register Addresses (Continued)

REGISTER	ADDRESS	DESCRIPTION
I1_STARTLSA_LO	8394h	swrstbh[10]
I1_STARTLSA_HI	8396h	swrstbh[11]
I1_TIMESTAMPL	8398h	swrstbh[12]
I1_TIMESTAMPM	839Ah	swrstbh[13]
I1_TIMESTAMPH	839Ch	swrstbh[14]
I1_SECAVLB	839Eh	swrstbh[15]
I1_ADDSECAVLB	83A0h	swrstbh[16]
I1_SECTHRESHLD	83A2h	swrstbh[17]
I1_FFTHRESHLD	83A4h	swrstbh[18]
I1_TSEARLYWIN	83A6h	swrstbh[19]
I1_TSLATEWIN	83A8h	swrstbh[20]
I1_ERRSTATUS	83AAh	swrstbh[21]
I1_CONFIG2	83ACh	swrstbh[22]
I1_FIFODAT	83AEh	swrstbh[23]
I1_FIFOADD	83B0h	swrstbh[24]
I1_UPPCRPID	83B2h	swrstbh[25]
I1_UPVIDEOPID	83B4h	swrstbh[26]
I1_UPPARSSM	83B6h	swrstbh[27]
I1_UPPARSSTAT	83B8h	swrstbh[28]
I1_PATTABLEDAT	83BAh	swrstbh[29]
I1_PMTTABLEDAT	83BCh	swrstbh[31]

3.1.2 ISOC2 Register Addresses

Table 3-3: I2 Register Addresses

Register	ADDRESS	DESCRIPTION
I2_ADDR	83C0h	swrstbh[32]
I2_ADPG	83C2h	swrstbh[33]
I2_ROLA	83C4h	swrstbh[34]
I2_RLDA	83C6h	swrstbh[35]
I2_ROLB	83C8h	swrstbh[36]
I2_RLDB	83CAh	swrstbh[37]
I2_CONFIG	83CCh	swrstbl[38]
I2_INTENA	83CEh	swrstbl[39]
I2_INT	83D0h	swrstbl[40]
I2_AVDATASIZE	83D2h	swrstbh[41]
I2_STARTLSA_LO	83D4h	swrstbh[42]
I2_STARTLSA_HI	83D6h	swrstbh[43]
I2_TIMESTAMPL	83D8h	swrstbh[44]
I2_TIMESTAMPM	83DAh	swrstbh[45]
I2_TIMESTAMPH	83DCh	swrstbh[46]
I2_SECAVLB	83DEh	swrstbh[47]
I2_ADDSECAVLB	83E0h	swrstbh[48]
I2_SECTHRESHLD	83E2h	swrstbh[49]
I2_FFTHRESHLD	83E4h	swrstbh[50]
I2_TSEARLYWIN	83E6h	swrstbh[51]
I2_TSLATEWIN	83E8h	swrstbh[52]
I2_ERRSTATUS	83EAh	swrstbh[53]
I2_CONFIG2	83ECh	swrstbh[54]
I2_FIFODAT	83EEh	swrstbh[55]
I2_FIFOADD	83F0h	swrstbh[56]
I2_UPPCRPID	83F2h	swrstbh[57]
I2_UPVIDEOPID	83F4h	swrstbh[58]
I2_UPPARSSM	83F6h	swrstbh[59]
I2_UPPARSSTAT	83F8h	swrstbh[60]
I2_PATTABLEDAT	83FAh	swrstbh[61]
I2_PMTTABLEDAT	83FCh	swrstbh[62]

3.1.3 ISOC3 Register Addresses

Table 3-4: I3 Register Address

REGISTER	ADDRESS	DESCRIPTION
I3_ADDR	8400h	swrstbh[64]
I3_ADPG	8402h	swrstbh[65]
I3_ROLA	8404h	swrstbh[66]
I3_RLDA	8406h	swrstbh[67]
I3_ROLB	8408h	swrstbh[68]
I3_RLDB	840Ah	swrstbh[69]
I3_CONFIG	840Ch	swrstbh[70]
I3_INTENA	840Eh	swrstbh[71]
I3_INT	8410h	swrstbh[72]
I3_AVDATASIZE	8412h	swrstbh[73]
I3_STARTLSA_LO	8414h	swrstbh[74]
I3_STARTLSA_HI	8416h	swrstbh[75]
I3_TIMESTAMPL	8418h	swrstbh[76]
I3_TIMESTAMPM	841Ah	swrstbh[77]
I3_TIMESTAMPH	841Ch	swrstbh[78]
I3_SECAVLB	841Eh	swrstbh[79]
I3_ADDSECAVLB	8420h	swrstbh[80]
I3_SECTHRESHLD	8422h	swrstbh[81]
I3_FFTHRESHLD	8424h	swrstbh[82]
I3_TSEARLYWIN	8426h	swrstbh[83]
I3_TSLATEWIN	8428h	swrstbh[84]
I3_ERRSTATUS	842Ah	swrstbh[85]
I3_CONFIG2	842Ch	swrstbh[86]
I3_FIFODAT	842Eh	swrstbh[87]
I3_FIFOADD	8430h	swrstbh[88]
I3_UPPCRPID	8432h	swrstbh[89]
I3_UPVIDEOPID	8434h	swrstbh[90]
I3_UPPARSSM	8436h	swrstbh[91]
I3_UPPARSSTAT	8438h	swrstbh[92]
I3_PATTABLEDAT	843Ah	swrstbh[93]
I3_PMTTABLEDAT	843Ch	swrstbh[94]

3.1.4 Trick Play Register, Ch. 3

Table 3-5: I3 Register Address

REGISTER	ADDRESS	DESCRIPTION
I3_TPCONFIG	8440h	swrstbh[96]
I3_TPCONFIG_HI	8441h	swrstbh[96]
I3_TPCURRENT	8442h	swrstbh[97]
I3_TSTAMPCFG	8444h	swrstbh[98]
I3_PRSSTATUS	8446h	swrstbh[99]

3.1.5 General Registers

Table 3-6: General Registers

REGISTER	ADDRESS	DESCRIPTION
DRAMTIME	8480h	swrstbh[128]
GEN_CONFIG	8482h	swrstbh[129]
GEN_INTENA	8484h	swrstbh[130]
GEN_INT	8486h	swrstbh[131]
D1_ADDRANGE	8488h	swrstbh[132]
D1_CONFIG	848Ah	swrstbh[133]
D2_ADDRANGE	848Ch	swrstbh[134]
D2_CONFIG	848Eh	swrstbh[135]
D3_ADDRANGE	8490h	swrstbh[136]
D3_CONFIG	8492h	swrstbh[137]
D4_ADDRANGE	8494h	swrstbh[138]
D4_CONFIG	8496h	swrstbh[139]
D5_ADDRANGE	8498h	swrstbh[140]
D5_CONFIG	849Ah	swrstbh[141]
CLOCKSELLO	849Ch	swrstbh[142]
CLOCKSELHI	849Dh	swrstbh[142]
APLL_REG1LO	848Eh	swrstbh[143]
APLL_REG1HI	848Fh	swrstbh[143]
APLL_REG2LO	8490h	swrstbh[144]
APLL_REG2HI	8491h	swrstbh[144]
TEST_MUXSEL1	8492h	swrstbh[145]

Table 3-6: General Registers (Continued)

REGISTER	ADDRESS	DESCRIPTION
TEST_MUXSEL2	8494h	swrstbh[146]
TEST_MUXSEL3	8496h	swrstbh[147]
Q_CONFIG	8498h	swrstbh[148]
Q_STATUS	849Ah	swrstbh[149]

3.1.6 ISOC1 Channel

There are 2 pairs of roll over and re-load addresses for each ISOC channel, which gives firmware flexibility and time to respond. When {I1_ADPG[23:16], I1_ADDR[15:8]} is equal to {I1_ROLPG_A[23:16], I1ROLHI_A[15:0], it is re-loaded to the value of {I1_RLPG_A[23:16], I1_RLHI_A[15:8]}. If it is equal to the value of the roll over B, then it is re-loaded with the value B.

REG - 8380 – 8381 h I1_ADDR: I1 Address Hi/Lo

R/W	15	14	13	12	11	10	9	8
	I1_ADDR[15:8]							
	0	0	0	0	0	0	0	0
8380h	7	6	5	4	3	2	1	0
	I1_ADDR[7:0]							
	0	0	0	0	0	0	0	0

(I1base + 00)

Bits 15 — 0: I1_ADDR

ISOC channel 1 address

This register, with 8382h I1, forms the 24 bit address to DRAM. Firmware programs this address initially, then Genii increments this address at the end of each access.

REG - 8382h – 8383h I1_ADPG: I1 Page Address Register

R/W	31	30	29	28	27	26	25	24
	Reserved							
	0	0	0	0	0	0	0	0
8382h I1	23	22	21	20	19	18	17	16
	I1_ADPG[23:16]							
	0	0	0	0	0	0	0	0

(I1 base + 02)

Bits 31 — 24: Reserved

Bits 23 — 16: I1_ADPG

ISOC channel 1 address

This register, with register I1_ADDR, forms the 24 bit address to DRAM. Firmware programs this address initially, then Genii increments this address at the end of each access.

REG - 8384h – 8385h I1_ROLA: I1 Rollover Address Register (pair A)

R/W	23	22	21	20	19	18	17	16
	I1_ROLPG[23:16]							
	0	0	0	0	0	0	0	0
8384h	15	14	13	12	11	10	9	8
	I1_ROLPG[15:0]							
	0	0	0	0	0	0	0	0

Bits 23 — 16: ROLPG_A

ISOC channel 1 roll over address, pair A

Bits 15 — 0: ROLHI_A

ISOC channel 1 roll over address, pair A

REG -8386h – 8387h I1_RLDA: I1 Re-Load Address Register (pair A)

R/W	23	22	21	20	19	18	17	16
	I1_RLPG[23:16]							
	0	0	0	0	0	0	0	0
8386h	15	14	13	12	11	10	9	8
	I1_RLHI[15:8]							
	0	0	0	0	0	0	0	0

Bits 23 — 16: I1_RLPG[23:16]

ISOC channel 1 re-load address, pair A

BITS 15 — 8: I1_RLHI[15:8]

ISOC channel 1 re-load address, pair A

REG - 8388h – 8389h I1_ROLB: I1 Rollover Address Register (pair B)

R/W	23	22	21	20	19	18	17	16
	I1_ROLPG_B[23:16]							
	0	0	0	0	0	0	0	0
8388h	15	14	13	12	11	10	9	8
	I1_ROLHI_B[15:8]							
	0	0	0	0	0	0	0	0

Bits 23 — 16: I1_ROLPG_B

ISOC channel 1 roll over address, pair B

Bits 15 — 8: I1_ROLHI_B

ISOC channel 1 roll over address, pair B

REG - 838Ah – 838Bh I1_RLDB: I1 Re-Load Address Register (pair B)

R/W	23	22	21	20	19	18	17	16
	I1_RLPG_B[23:16]							
	0	0	0	0	0	0	0	0
832Ah	15	14	13	12	11	10	9	8
	I1_RLHI_B[15:8]							
	0	0	0	0	0	0	0	0

Bits 23 — 16: I1_RLPG_B

ISOC channel 1 re-load address, pair B

Bits 15 — 8: I1_RLHI_B

ISOC channel 1 re-load address, pair B

REG - 838Ch – 838Dh I1_CONFIG: I1_Configuration Register

R/W	15	14	13	12	11	10	9	8
	EOSTREAM	DELNULL_ ENA	DVB_MODE	DISABLE_ TP	AVCLKOUTSEL[1:0]		AVCLK- OUTENA	SYNCPCR ENA
	0	0	0	0	0	0	0	0
838Ch	7	6	5	4	3	2	1	0
	BFR_SIZE			MPEG_ FORMAT	PAUSE	DISABLE_ TS	XFERENA	INDIR
	0	0	0	0	0	0	0	0

Bit 15: EOSTREAM

End of stream.

This bit is set by firmware when recording is being stopped.

Hardware sets the eostream bit in the sector header. While transmitting, hardware detects this bit and asserts interrupt to firmware.

Bit 14: DELNULL_ENA

Delete null packet enable.

When this bit is set, the null packets are detected and dropped while receiving.

Bit 13: DVB_MODE

Enable DVB mode.

Bits 12: DISABLE_TP

To disable Trick Play, set this bit.

Bit 11–10: AVCLKOUTSEL[1:0]

Bits AVCLKOUTSEL[0] and AVCLKOUTSEL[1] are used to select the frequency of the AVCLK output:

00: AV1CLK (system dependent)
01: AV1CLK (system dependent)
10: 20MHz clock
11:TSCLK (27MHz)

Bit 9: AVCLKOUTENA

Enables driving AVCLK externally.

Bit 8: SYNCPCRENA

Enables time stamp counter to synchronize with incoming PCR.

Bits 7–5: BFR_SIZE

These 3 bits set the limit of each DMA channel when it accesses DRAM. This helps set up a fair arbitration scheme, which depends on the bit rate of each DMA channel.

000: each request to DRAM access will be 8 words long
001: each request to DRAM access will be 16 words long
010: each request to DRAM access will be 32 words long
011: each request to DRAM access will be 64 words long
100: each request to DRAM access will be 128 words long
101: each request to DRAM access will be 256 words long
110: each request to DRAM access will be 512 words long
111: Reserved

Bit 4: MPEG_FORMAT

MPEG Format

When this bit is programmed to ‘1’, this channel will be receiving/transmitting packets in MPEG 2 Transport Stream format.

Bit 3: PAUSE

To pause the stream, set this bit.

To receive or to transmit, clear this bit.

Bit 2: DISABLE_TS

Disable Time Stamp.
When this bit is programmed to ‘1’, Genii will not generate sector index and data index for each sector it stores in DRAM. Data will be stored as is without any index.

Bit 1: XFERENA

Transfer Enable.

After finishing all necessary registers in this DMA channel, firmware set XFERENA bit to ‘1’ to enable the channel to receive or transmit data. To stop the channel from receiving/transmitting data, firmware reset this bit to ‘0’.

Bit 0: INDIR

In direction.

This bit is set to ‘1’ to let this DMA channel know that it should be RECEIVING data from the link and store that data in Dram.

If the bit is reset to ‘0’, this DMA channel should get data from Dram & Transmit to the link.

REG - 838Eh – 838Fh I1_INTENA: I1 Interrupt Enable Register

R/W	15	14	13	12	11	10	9	8
	Reserved		QINT_ENA	PAT_INT_ENA	PMT_INT_ENA	AVFIFO-ERR_INT_ENA	ROLLB_INT_ENA	ROLLA_INT_ENA
	0	0	0	0	0	0	0	0

838Eh I1_RLHI_B	7	6	5	4	3	2	1	0
	BUFERR_INT_ENA	MISS_SYNC_INT_ENA	Data_size_mis-matched_int_ena	EOSTREAM_INT_ENA	IFULL_INT_ENA	ISTART_INT_ENA	NOP-SPACE_INT_ENA	SPHRESHOLD_INT_ENA
	0	0	0	0	0	0	0	0

Each bit in this reg matches one on one w/ ea. bit in the following Interrupt reg.It's used to let firmware choose which interrupt is worth setting the HINT output pin.

Examples follow.

Example 1

- SPTHRESHOLD_INT_ENA bit is programmed to '1'.
- Genii detects the condition to set SPTHRESHOLD_INT bit.
- It sets SPTHRESHOLD_INT bit to '1', and asserts HINT output pin.

Example 2

- SPTHRESHOLD_INT_ENA bit is programmed to '0' Genii detects the condition to set SPTHRESHOLD_INT bit.
- It set SPTHRESHOLD_INT bit to '1', but does not assert HINT output pin.

Bits 15 – 14: Reserved

Not used.

Bit 13: QINT_ENA

Enables interrupt from Q2DMA block.

Bit 12: PAT_INT_ENA

PAT Interrupt Enable.

Bit 11: PMT_INT_ENA

PMT Interrupt Enable.

Bit 10: AVFIFOERR_INT_ENA

Bit 9: ROLLB_INT_ENA

Enables Rollover B interrupt.

Bit 8: ROLLA_INT_ENA

Enables Rollover A interrupt.

Bit 7: BUFERR_INT_ENA

Enables Buffer Error (underflow/overflow) interrupt.

Bit 6: MISS_SYNC_INT_ENA

Enables missed synchronize interrupt.

Bit 5: DATA_SIZE_MISMATCHED_INT_ENA

Enables Data_size_mismatched_int.

Bit 4: EOSTREAM_INT_ENA

End of Stream Interrupt Enable

Bit 3: IFULL_INT_ENA

I-frame table Full Interrupt Enable

Bit 2: ISTART_INT_ENA

I-Frame Start Interrupt Enable

Bit 1: NOPSPACE_INT_ENA

No Space Interrupt Enable

Bit 0: SPHRESHOLD_INT_ENA:

Space Threshold Interrupt Enable

REG - 8390h – 8391h I1 INT: I1 Interrupt Register

R/W	15	14	13	12	11	10	9	8
	Reserved		QINT	PAT_INT	PMT_INT	AVFIFO ERR_INT	ROLLB_ INT	ROLLA_ INT
	0	0	0	0	0	0	0	0
8390h		7	6	5s	4	3	2	1	0
		BUF_ERR_ INT	MISS_SYNC _BIT	Data_size_ mis-matched_ int	EO-STREAM _INT	IFULL_INT	ISTART_ INT_ENA	NO SPACE _INT	SPHRESHOLD _INT
		0	0	0	0	0	0	0	0

Internal condition of Genii will set the corresponding bit in this register to let firmware know the condition. To clear the bit, firmware writes a '1' to the desired bit.

Bits 15 — 14: Reserved

Read as zeroes.

Bit 13: QINT

Indicates FIFO over-run/under-run error in the Q2DMA. Check REG QSTATUS for more information.

Bit 12: PAT_INT

PAT Interrupt.

This bit is set when a PAT version change is detected in the video stream.

Bit 11: PMT_INT

PMT Interrupt.

This bit is set when a PMT version change is detected in the video stream.

Bit 10: AVFIFOERR_INT

Bit 9: ROLLB_INT

This interrupt informs firmware that rollover B has occurred.

Bit 8: ROLLA_INT

This interrupt informs firmware that rollover A has occurred.

Bit 7: BUF_ERR_INT

Buffer Error (underflow/overflow) interrupt.

Bit 6: MISS_SYNC_INT

This bit is valid only when it is in the receive direction, and Genii is programmed in MPEG 2 mode.

Set when the first byte of an MPEG 2 packer is not a SYNC byte (SYNC byte has the value of 47h).

Bit 5: DATA_SIZE_MISMATCHED_INT

This bit is only valid in the receive direction.

Set when the receiving packet length does not match the value programmed in the AVXFERSIZE register.

Bit 4: EOSTREAM_INT

End Of Stream Interrupt

Set when Genii detects the end of frame from the transmit packet. This bit is valid only in transmit to the link direction.

Bit 3: IFULL_INT

I-frame table Full Interrupt.

This bit is set when the I-frame FIFO is full.

Bit 2: ISTART_INT

I-frame Start Interrupt.

This bit is set when a start of an I-frame is detected in the video stream. Firmware can read this information from FIFODAT register.

1.When Genii detects an incoming I-Frame, it saves the beginning LSA number of that I-Frame in the SAVE_LSA register.

2.Genii resets the sector counter at the beginning of the I-Frame.

3.At the end of that I-Frame, Genii saves the sector counter value in the SAVE_SECTOR register.

4.Genii only sets HINT pin — if the interrupt enable bit is set — at the end of the I-Frame before the LSA number and the sector counter are saved in the FIFO.

Bit 1: NOSPACE_INT

No Space Interrupt

Set when the space available register reaches to '0'. In normal working condition, this bit should never get set because firmware should add to space available when it gets interrupted by SPTHRESHOLD_INT bit already.

When this condition happens, Genii will not receive or transmit until firmware fixes it by adding more space to space available register.

Bit 0: SPHRESHOLD_INT

Space Threshold Interrupt

This bit is set when the space available counter decrements to or below the value programmed in space threshold register.

REG - 8392h – 8393h I1_AVDATASIZE: I1 AV Transfer Size Register

R/W	15	14	13	12	11	10	9	8
	I1_DATASIZE[15:8]							
	0	0	0	0	0	0	0	0

8392h	7	6	5	4	3	2	1	0
	I1_DATASIZE[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 0: I1_DATASIZE

ISOC channel 1 Data size

Firmware programs this register to let Genii know the size of each packet in word unit (16 bits). Genii uses this information to pack data to the sector size required by the disk controller chip.

REG - 8394h–8395 I1 STARTLSA_LO: I1 Start LSA High Register

R/W	31	30	29	28	27	26	25	24
8397h	I1_STARTLSA[31:24]							
	0	0	0	0	0	0	0	0
8394h	23	22	21	20	19	18	17	16
	I1_STARTLSA[23:16]							
	0	0	0	0	0	0	0	0

This register is associated with the following registers (8396h – 8397h)

Bits 31 — 16: I1_STARTLSA

Before starting the channel, firmware programs a number to this register. At each receiving packet, Genii packs them together into a sector, and increases the LSA counter at each sector. This LSA counter is used to keep track of the I-Frame, and is saved in the SAVE_LSA register. This makes it available for firmware to use later for search algorithm.

REG - 8396h – 8397 I1 STARTLSA_HI: I1 Start LSA Low Register

R/W	15	14	13	12	11	10	9	8
8397h	I1_STARTLSA[15:8]							
	0	0	0	0	0	0	0	0
8396h	7	6	5	4	3	2	1	0
	I1_STARTLSA[7:0]							
	0	0	0	0	0	0	0	0

This register is associated with the previous registers (8394h – 8395h)

Bits 15 — 0: I1_STARTLSA

Before starting the channel, firmware programs a number to this register. At each receiving packet, Genii packs them together into a sector, and increases the LSA counter at each sector. This LSA counter is used to keep track of the I-Frame, and is saved in the SAVE_LSA register. This makes it available for firmware to use later for search algorithm.

REG - 8398h–8399h I1_TIMESTAMPL: I1 Time Stamp L Register

R/W	15	14	13	12	11	10	9	8
8399h	I1_TIMESTAMP[15:8]							
	0	0	0	0	0	0	0	0
8398h	7	6	5	4	3	2	1	0
	I1_TIMESTAMP[7:0]							
	0	0	0	0	0	0	0	0

Bit 15–0: I1_TIMESTAMP[15:0]

The least significant timestamp[15:0] bits from Time Stamp counter.

REG - 839Ah–839Bh I1_TIMESTAMPM: I1 Time Stamp M Register

R/W	31	30	29	28	27	26	25	24
839Bh	I1_TIMESTAMP[31:24]							
	0	0	0	0	0	0	0	0
839Ah	23	22	21	20	19	18	17	16
	I1_TIMESTAMP[23:16]							
	0	0	0	0	0	0	0	0

Bits 31–16: I1_TIMESTAMP[31:26]

Bits from the Time Stamp counter.

REG - 839Dh I1_TIMESTAMPH: I1 Time Stamp H Register

R/W	15	14	13	12	11	10	9	8
	Reserved						I1_TIMESTAMP[41:40]	
	0	0	0	0	0	0	0	0
839Ch	7	6	5	4	3	2	1	0
	I1_TIMESTAMP[39:32]							
	0	0	0	0	0	0	0	0

Bits 15–10: Reserved

Not used.

Bits 9–0: I1_TIMESTAMP[41:32]

Bits from the Time Stamp register.

REG - 839Eh–839Fh I1_SECTAVLB: I1 Sector Available Register

R/W	15	14	13	12	11	10	9	8
	I1_SECTAVLB[15:8]							
	0	0	0	0	0	0	0	0
839Eh	7	6	5	4	3	2	1	0
	I1_SECTAVLB[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 0: I1_SECTAVLB

Sector Available (in sector unit)

Before starting the channel, firmware programs this register with a sector available number.

If the channel is in Receive mode, this register indicates the available space for receiving data. If the channel is in Transmit mode, this register shows the number of sectors in buffer memory that are ready to be transmitted.

At the end of each sector, Genii decrements this register by 1. Genii stops transferring when this register reaches 0. Interrupt is set when this register is equal to the I1_SECTORTHRESHOLD register.

Firmware can modify this register by writing to the I1_ADDSECAVLB register. In this case, Genii adds the value of I1_SECTAVLB with the value of I1_ADDSECAVLB, then stores that sum in I1_SECTAVLB

REG 83A0h–83A1:I1_ADDSECAVLB Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0

83A0h	7	6	5	4	3	2	1	0
	ADDSECAVLB[7:0]							
	0	0	0	0	0	0	0	0

Bits 7 — 0: ADDSECAVLB[7:0]

Each time firmware writes to this register, Genii performs the addition of this register to the I1_SECAVLB register.

For more information, see REG - 839Eh–839Fh I1_SECTAVLB: I1 Sector Available Register

REG - 83A2h – 83A3h I1_SECTHRESHLD Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0

83A2h	7	6	5	4	3	2	1	0
	I1_SECTHRESHLD[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 8: Reserved

Not used.

Bits 7–0: I1_SECTHRESHLD[7:0]

The threshold level of space (sector) that is available to assert space threshold interrupt to firmware.

Genii compares this register to the I1_SECTAVLB register. The these two registers are equal, an interrupt is generated.

REG - 83A4h – 83A5h I1_FFTHRESHLD Register

R/W	15	14	13	12	11	10	9	8
	Reserved							I1_FFTHRESHLD[8]
	0	0	0	0	0	0	0	0

83A4h	7	6	5	4	3	2	1	0
	I1_FFTHRESHLD[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 9: Reserved

Not used.

Bits 8–0: I1_FFTHRESHLD[8:0]

The threshold level of the Buffer FIFO to assert space dma_reg to buffer control.

REG - 83A6h – 83A7h I1_TSEARLYWIN Register

R/W	15	14	13	12	11	10	9	8
	I1_SEARLYWIN[15:8]							
	0	0	0	0	0	0	0	0

83A6h	7	6	5	4	3	2	1	0
	I1_TSEARLYWIN[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 0: I1_TSEARLYWIN[15:0]

When receiving, TSEARLYWIN register is subtracted from the Time Stamp counter (early window). This informs how early to expect the incoming PCR value. If PRC value is less than the early window, it is treated as a discontinuous PCR value.

When transmitting, TSEARLYWIN register is substitute from Time Stamp from the data index (early window). This informs how early the packet can be transmitted.

REG - 83A8h – 83A9h I1_TSLATEWIN Register

R/W	15	14	13	12	11	10	9	8
	I1_TSLATEWIN[15:8]							
	0	0	0	0	0	0	0	0

83A8h	7	6	5	4	3	2	1	0
	I1_TSLATEWIN[7:0]							
	0	0	0	0	0	0	0	0

Bits 15–0: I1_TSLATEWIN[15:0]

While receiving, the TSLATEWIN register is added to Time Stamp counter (late window). This informs how late to expect the incoming PCR value. If PCR value is greater than late window, it will be treated as a discontinuous PCR value.

While transmitting, the TSLATEWIN register is added to Time Stamp from the data index (late window). If the current Time Stamp counter value is greater than that of late window, the packet will be discarded.

REG - 83AAh – 83ABh I1_ERRSTATUS Register

R/W	15	14	13	12	11	10	9	8
	Reserved		QFIFO_ STATUS	PAT_INT_ STATUS	PMT_INT_ STATUS	AVFIFO_ ERR	ROLLB_ INT_ STATUS	ROLLA_ INT_ STATUS
	0	0	0	0	0	0	0	0

83AAh	7	6	5	4	3	2	1	0
	BFIFO_ ERROR	MISS_ SYNC_ ERROR	DATA_SIZE _MIS- MATCHED	EO- STREAM	I_FRAME_ FULL	I_FRAME_ DETECTED	NO_SPACE _STATUS	SPACE_ THRESHOLD_ STATUS
	0	0	0	0	0	0	0	0

This read-only register shows the errors and status of the internal Genii condition.

Bits 15–14: Reserved

Read as zeroes.

BIT 13: QFIFO_STATUS

When this bit is set, Firmware must go to the Q_STATUS2 register

BIT 12: PAT_INT_STATUS

When set, this bit indicates that PAT table is detected from the MPEG2 stream.

BIT 11: PMT_INT_STATUS

When set, this bit indicates that PMT table is detected from the MPEG2 stream.

Bit 10: AVFIFO_ERR

When this bit is set, it indicates that AVFIFO ever-run is detected.

Bit 9: ROLLB_INT_STATUS

When set, this bit indicates that roll-over Pointer B has occurred.

Bit 8: ROLLA_INT_STATUS

When set, this bit indicates that roll-over Pointer A has occurred.

Bit 7: BFIFO_ERROR

When this bit is set, it indicates that either buffer FIFO overflow or buffer FIFO underflow has occurred.

Bit 6: MISS_SYNC_ERROR

When this bit is set, it indicates that MISS_SYNC error is detected.

Bit 5: DATA_SIZE_MISMATCHED

When set, this bit indicates that the receiving AV data packet is either too long or too short.

Bit 4: EOSTREAM

When set, this bit indicates that the End of Stream condition is detected while transmitting to AV system.

Bit 3: I_FRAME_FULL

When this bit is set, it indicates that the I FRAME TABLE FIFO is full.

Bit 2: I_FRAME_DETECTED

When this bit is set, it indicates that the I-FRAME from MPEG2 stream has been detected.

Bit 1: NO_SPACE STATUS

When this bit is set, either no space is available in SDRAM while receiving, or no sector is available in SDRAM while transmitting.

Bit 0: SPACE_THRESHOLD_STATUS

When this bit is set, the value in the I1_SECTAVLB register is less than the value in the I1_SETCTHRESHOLD register.

REG - 83ACh – 83ADh I1_CONFIG2 Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0

83ACh	7	6	5	4	3	2	1	0
	Reserved		EXT_EARLY_WIN[2:0]			FIFOSEL[1:0]		TEST_ MODE
	0	0	0	0	0	0	0	0

Bits 15–6: Reserved

Read as zeroes.

Bits 5–3: EXT_EARLY_WIN[2:0]

These bits extend the early WIN to determine late packet. Each unit in this register is equal to about 11 µs.

The early window is subtracted by EXT_EARLY_WIN to form the extended_early_window. If the time stamp of the packet is not between_early_window and late_window, then the packet will be dropped.

BitS 2–1: FIFOSEL[1:0]

FIFO select:

00: I FRAME FIFO
01: AV FIFO
10: BUFFER FIFO
11: Q FIFO

Bit 0: TEST_MODE

To test FIFOs, enable this bit.

REG - 83AEh – 83AFh I1_FIFODAT Register

R/W	15	14	13	12	11	10	9	8
	I1_FIFODAT[15:8]							
	0	0	0	0	0	0	0	0

83AEh	7	6	5	4	3	2	1	0
	I1_FIFODAT[7:0]							
	0	0	0	0	0	0	0	0

Bits 15–0: I1_FIFODAT[15:0]

16-bit FIFO data register.

This is the FIFO data of the FIFO selected by FIFOSEL in the I1_CONFIG2 register.
In test mode, Firmware can read/write data to/from FIFOs.

REG - 83B0h – 83B1h I1_FIFOADD Register

R/W	15	14	13	12	11	10	9	8
	Reserved							I1_ FIFOADD[8]
	0	0	0	0	0	0	0	0

83B0h	7	6	5	4	3	2	1	0
	I1_FIFOADD[7:0]							
	0	0	0	0	0	0	0	0

In the test mode,firmware needs to initialize this register only once. Hardware detects the WR/RD strobe of I1_FIFODAT, then increments I1_FIFOADD.

Bits 15–9: Reserved

Not used. Read as zeroes.

Bits 8–0: I1_FIFOADD[8:0]

9-bit FIFO address register.

This is the FIFO address of the FIFO selected by FIFOSEL in the I1_CONFIG2 register:

- When firmware is writing data to the FIFO, this is the write address.
- When firmware is reading data from the FIFO, this is the read address.

REG – 83B2h—83B3h I1_UPPCRPID Register

R/W	15	14	13	12	11	10	9	8
	disable_table_p arsers	PAT_crc_ err_ena	PMT_crc_ err_ena	PCR_PID[12:8]				
	0	0	0	0	0	0	0	0

84B2h R/W	7	6	5	4	3	2	1	0
	PCR_PID[7:0]							
	0	0	0	0	0	0	0	0

Bit 15: disable_table_parsers (R/W)

When set to ‘1’, this bit disables table parsers and allows uP to override PCR_PID and Video_pid values

Bit 14: PAT_crc_err_ena (R/W)

When set to ‘1’, this bit enables crc checker in PAT parser.

Bit 13: PMT_crc_err_ena (R/W)

When set to ‘1’, this bit enables crc checker in PMT parser.

Bits [12:0]: PCR_PID[12:0] (Write)

Firmware controls the PID value when bit 15 = ‘1’.

Bits [12:0]: PCR_PID[12:0] (Read)

When bit 15 = ‘1’, read back firmware written value, else read back parser-detected PID value.

REG – 83B4h – 83B5h I1_UPVIDEOPID Register

R/W	15	14	13	12	11	10	9	8
	Reserved			Video_PID[12:8]				
	0	0	0	0	0	0	0	0
84B4h	7	6	5	4	3	2	1	0
R/W	Video_PID[7:0]							
	0	0	0	0	0	0	0	0

Bits [15:13]: Reserved

Read as zeroes.

Bits [12:0:] Video_PID[12:0] (Write)

Firmware controls PID value.

Bits [12:0] Video_PID[12:0] (Read)

When bit 15 (register I1_UPPCRPID) = '1', read back firmware written value, else read back parser-detected PID value.

REG – 83B6h–83B7h I1_UPPARSSM Register

R/W	15	14	13	12	11	10	9	8
	up_pat_parsed (R/W)	Reserved		uppmtsm[4:0]				
	0	0	0	0	0	0	0	0
84B6h	7	6	5	4	3	2	1	0
R/W	Reserved			uppatsm[4:0]				
	0	0	0	0	0	0	0	0

Bit 15: up_pat_parsed (R/W)

Forces 'pat_parsed' state when written.

Monitors the state when read.

Bits 14 — 13: Reserved

Not used. Read as zeroes.

Bits 12 — 8: uppmtsm[4:0]

Forces PMT FSM state when written.

Monitors the state when read.

Bits 7— 5: Reserved

Read as zeroes.

Bits 4 — 0: uppatsm[4:0]

Forces PAT FSM state when written.

Monitors the state when read.

REG – 83B8h–83B9h I1_UPPARSSTAT (Read Only) Register

R/O	15	14	13	12	11	10	9	8	
	Reserved		PMT_ adapt_err	PMT _ cont_err	PMT_ crc_err	PMT_ pointer_err	PMT_ section_ err	PMT_ table_ id_err	
	0	0	0	0	0	0	0	0	
84B8h		7	6	5	4	3	2	1	0
R/O		Reserved			PAT_ adapt_err	PAT_ cont_err	PAT_ crc_err	PAT_ pointer_err	PAT_table _id_err
		0	0	0	0	0	0	0	0

Bits [15:14] Reserved

Read as zeroes

Bit 13: PMT_adapt_err

This bit indicates adaptation length field error

Bit 12: PMT_cont_err

This bit indicates continuity counter error for tables spanning multiple packets.

Bit 11: PMT_crc_err

This bit indicates crc error detected

Bit 10: PMT_pointer_err

This bit indicates pointer value error

Bit 9: PMT_section_err

This bit indicates invalid values for section_number and/or last_section_number fields.

Bit 8: PMT_table_id_err

This bit indicates table ID not equal to 02h

Bit [7:5] Reserved

Read as zeroes

Bit 4: PAT_adapt_err

This bit indicates adaptation length field error

Bit 3: PAT_cont_err

This bit indicates continuity counter error for tables spanning multiple packets.

Bit 2: PAT_crc_err

This bit indicates crc error detected

Bit 1: PAT_pointer_err

This bit indicates pointer value error

Bit 0: PAT_table_id_err

This bit indicates table ID not equal to 00

REG - 83BAh – 83BBh I1_PATTABLEDAT: I1 PAT Table Data Register

	15	14	13	12	11	10	9	8
R/O	PATTABLEDAT[15:8]							
	0	0	0	0	0	0	0	0
84BAh	7	6	5	4	3	2	1	0
R/O	PATTABLEDAT[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 – 0: PATTABLEDAT[15:0]

16-bit PAT Table Data.

This is the data that is stored in the PAT table. It contains information of the PAT version change.

REG - 83BCh – 83BDh I1_PMTTABLEDAT: I1 PMT Table Data Register

	15	14	13	12	11	10	9	8
R/O	PMTTABLEDAT[15:8]							
	0	0	0	0	0	0	0	0
84BCh	7	6	5	4	3	2	1	0
R/O	PMTTABLEDAT[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 – 0: PMTTABLEDAT[15:0]

16-bit PMT Table Data.

This is the data that is stored in the PMT table. It contains information of the PMT version change.

3.1.7 ISOC2 Channel

REG - 83C0h – 83C1h I2_ADDR: I2 Address Hi/Lo

	15	14	13	12	11	10	9	8
R/W	I2_ADDR[15:8]							
	0	0	0	0	0	0	0	0
83C0h	7	6	5	4	3	2	1	0
	I2_ADDR[7:0]							
	0	0	0	0	0	0	0	0

(I1base + 00)

Bits 15 — 0: I2_ADDR[15:0]

ISOC channel 1 address

This register, with 8382h I1, forms the 24 bit address to DRAM. Firmware programs this address initially, then Genii increments this address at the end of each access.

REG - 83C2h – 83C3h I2_ADPG: I2 Page Address Register

R/W	31	30	29	28	27	26	25	24
	Reserved							
	0	0	0	0	0	0	0	0
83C2h I1	23	22	21	20	19	18	17	16
	I2_ADPG[23:16]							
	0	0	0	0	0	0	0	0

(I2 base + 02)

Bits 31 — 24: Reserved

Not used. Read as zeroes.

Bits 23 — 16: I2_ADPG[23:16]

ISOC channel 1 address

This register, with register I1_ADDR, forms the 24 bit address to DRAM. Firmware programs this address initially, then Genii increments this address at the end of each access.

REG - 83C4h – 83C5h I2_ROLA: I2 Rollover Address Register (pair A)

R/W	23	22	21	20	19	18	17	16
	I2_ROLPG[23:16]							
	0	0	0	0	0	0	0	0
83C4	15	14	13	12	11	10	9	8
	I2_ROLPG[15:0]							
	0	0	0	0	0	0	0	0

Bits 23 — 16: ROLPG_A

ISOC channel 1 roll over address, pair A

Bits 15 — 0: ROLHI_A

ISOC channel 1 roll over address, pair A

REG -83C6h – 83C7h I2_RLDA: I2 Re-Load Address Register (pair A)

R/W	23	22	21	20	19	18	17	16
	I2_RLPG[23:16]							
	0	0	0	0	0	0	0	0
83C6h	15	14	13	12	11	10	9	8
	I2_RLHI[15:8]							
	0	0	0	0	0	0	0	0

Bits 23 — 16: I2_RLPG

ISOC channel 1 re-load address, pair A

BITS 15 — 8: I2_RLHI

ISOC channel 1 re-load address, pair A

REG - 83C8h – 83C9h I2_ROLB: I2 Rollover Address Register (pair B)

R/W	23	22	21	20	19	18	17	16
	I2_ROLPG_B[23:16]							
	0	0	0	0	0	0	0	0
83C8h	15	14	13	12	11	10	9	8
	I2_ROLHI_B[15:8]							
	0	0	0	0	0	0	0	0

Bits 23 — 16: I2_ROLPG_B

ISOC channel 1 roll over address, pair B

Bits 15 — 8: I2_ROLHI_B

ISOC channel 1 roll over address, pair B

REG - 83CAh – 83CBh I2_RLDB: I2 Re-Load Address Register (pair B)

R/W	23	22	21	20	19	18	17	16
	I2_RLPG_B[23:16]							
	0	0	0	0	0	0	0	0
83CAh	15	14	13	12	11	10	9	8
	I2_RLHI_B[15:8]							
	0	0	0	0	0	0	0	0

Bits 23 — 16: I2_RLPG_B

ISOC channel 1 re-load address, pair B

Bits 15 — 8: I2_RLHI_B

ISOC channel 1 re-load address, pair B

REG - 83CCh – 83CDh I2_CONFIG: I2_Configuration Register

R/W	15	14	13	12	11	10	9	8
	EOSTREAM	DELNULL_ ENA	DVB_MODE	DISABLE_ TP	AVCLKOOUSEL[1:0]		AVCLK- OUTENA	SYNCPCR ENA
	0	0	0	0	0	0	0	0
83CCh	7	6	5	4	3	2	1	0
	BFR_SIZE			MPEG_ FORMAT	PAUSE	DISABLE_ TS	XFERENA	INDIR
	0	0	0	0	0	0	0	0

Bit 15: EOSTREAM

End of stream.

This bit is set by firmware when recording is being stopped.

Hardware sets the eostream bit in the sector header. While transmitting, hardware detects this bit and asserts interrupt to firmware.

Bit 14: DELNULL_ENA

Delete null packet enable.

When this bit is set, the null packets are detected and dropped while receiving.

Bit 13: DVB_MODE

Enable DVB mode.

Bits 12: DISABLE_TP

To disable Trick Play, set this bit.

Bit 11–10: AVCLKOUTSEL[1:0]

Bits AVCLKOUTSEL[0] and AVCLKOUTSEL[1] are used to select the frequency of the AVCLK output:

00: AV1CLK (system dependent)
01: AV1CLK (system dependent)
10: 20MHz clock
11:TSCLK (27MHz)

Bit 9: AVCLKOUTENA

Enables driving AVCLK externally.

Bit 8: SYNCPCRENA

Enables time stamp counter to synchronize with incoming PCR.

Bits 7–5: BFR_SIZE

These 3 bits set the limit of each DMA channel when it accesses DRAM. This helps set up a fair arbitration scheme, which depends on the bit rate of each DMA channel.

000: each request to DRAM access will be 8 words long
001: each request to DRAM access will be 16 words long
010: each request to DRAM access will be 32 words long
011: each request to DRAM access will be 64 words long
100: each request to DRAM access will be 128 words long
101: each request to DRAM access will be 256 words long
110: each request to DRAM access will be 512 words long
111: Reserved

Bit 4: MPEG_FORMAT

MPEG Format

When this bit is programmed to ‘1’, this channel will be receiving/transmitting packets in MPEG 2 Transport Stream format.

Bit 3: PAUSE

To pause the stream, set this bit.

To receive or to transmit, clear this bit.

Bit 2: DISABLE_TS

Disable Time Stamp.
When this bit is programmed to ‘1’, Genii will not generate sector index and data index for each sector it stores in DRAM. Data will be stored as is without any index.

Bit 1: XFERENA

Transfer Enable.

After finishing all necessary registers in this DMA channel, firmware set XFERENA bit to ‘1’ to enable the channel to receive or transmit data. To stop the channel from receiving/transmitting data, firmware reset this bit to ‘0’.

Bit 0: INDIR

In direction.

This bit is set to '1' to let this DMA channel know that it should be RECEIVING data from the link and store that data in Dram.

REG - 83CEh – 83CFh I2_INTENA: I2 Interrupt Enable Register

R/W	15	14	13	12	11	10	9	8
	Reserved			PAT_INT_ ENA	PMT_INT_ ENA	Reserved	ROLLB_ INT_ENA	ROLLA_ INT_ENA
	0	0	0	0	0	0	0	0
83CEh	7	6	5	4	3	2	1	0
	BUFERR_ INT_ENA	MISS_SYNC _INT_ENA	Data_size_ mis-matched _int_ena	EOSTREAM _INT_ENA	IFULL_ INT_ENA	ISTART_ INT_ENA	NOSPACE _INT_ENA	SPHRESH OLD_INT_ ENA
	0	0	0	0	0	0	0	0

Each bit in this reg matches one on one w/ ea. bit in the following Interrupt reg.It's used to let firmware choose which interrupt is worth setting the HINT output pin.

Examples follow:

Example 1

- SPTHRESHOLD_INT_ENA bit is programmed to '1'.
- Genii detects the condition to set SPTHRESHOLD_INT bit.
- It sets SPTHRESHOLD_INT bit to '1', and asserts HINT output pin.

Example 2

- SPTHRESHOLD_INT_ENA bit is programmed to '0' Genii detects the condition to set SPTHRESHOLD_INT bit.
- It sets SPTHRESHOLD_INT bit to '1', but does not assert HINT output pin.

Bits 15—13: Reserved

Not used.

Bit 12: PAT_INT_ENA

PAT Interrupt Enable.

Bit 11: PMT_INT_ENA

PMT Interrupt Enable.

Bit 9: ROLLB_INT_ENA

Enables Rollover B interrupt.

Bit 8: ROLLA_INT_ENA

Enables Rollover A interrupt.

Bit 7: BUFERR_INT_ENA

Enables Buffer Error (underflow/overflow) interrupt.

Bit 6: MISS_SYNC_INT_ENA

Enables missed synchronize interrupt.

Bit 5: DATA_SIZE_MISMATCHED_INT_ENA

Enables Data_size_mismatched_int.

Bit 4: EOSTREAM_INT_ENA

End of Stream Interrupt Enable

Bit 3: IFULL_INT_ENA

I-frame table Full Interrupt Enable

Bit 2: ISTART_INT_ENA

I-Frame Start Interrupt Enable

Bit 1: NOSPACE_INT_ENA

No Space Interrupt Enable

Bit 0: SPHRESHOLD_INT_ENA:

Space Threshold Interrupt Enable

REG - 83D0h – 83D1h I2 INT: I2 Interrupt Register

R/W	15	14	13	12	11	10	9	8
	Reserved			PAT_INT	PMT_INT	Reserved	ROLLB_ INT	ROLLA_ INT
	0	0	0	0	0	0	0	0

83D0h	7	6	5s	4	3	2	1	0
	BUF_ERR_ INT	MISS_SYNC _BIT	Data_size_ mis-matched_ int	EO-STREAM _INT	IFULL_INT	ISTART_ INT	NOSPACE _INT	SPHRESHOLD _INT
	0	0	0	0	0	0	0	0

Internal condition of Genii will set the corresponding bit in this register to let firmware know the condition. To clear the bit, firmware writes a ‘1’ to the desired bit.

Bits 15 — 13: Reserved

Read as zeroes.

Bit 12: PAT_INT

PAT Interrupt.

This bit is set when a PAT version change is detected in the video stream.

Bit 11: PMT_INT

PMT Interrupt.

This bit is set when a PMT version change is detected in the video stream.

Bit 9: ROLLB_INT

This interrupt informs firmware that rollover B has occurred.

Bit 8: ROLLA_INT

This interrupt informs firmware that rollover A has occurred.

Bit 7: BUF_ERR_INT

Buffer Error (underflow/overflow) interrupt.

Bit 6: MISS_SYNC_INT

This bit is valid only when is in the receive direction, and Genii is programmed in MPEG 2 mode.

Set when the first byte of an MPEG 2 packer is not a SYNC byte (SYNC byte has the value of 47h).

Bit 5: DATA_SIZE_MISMATCHED_INT

This bit is only valid in the receive direction.

Set when the receiving packet length does not match the value programmed in the AVXFERSIZE register.

Bit 4: EOSTREAM_INT

End Of Stream Interrupt

Set when Genii detects the end of frame from the transmit packet. This bit is valid only in transmit to the link direction.

Bit 3: IFULL_INT

I-frame table Full Interrupt.

This bit is set when the I-frame FIFO is full.

Bit 2: ISTART_INT_ENA

I-frame Start Interrupt.

This bit is set when a start of an I-frame is detected in the video stream. Firmware can read this information from FIFODAT register.

5.When Genii detects an incoming I-Frame, it saves the beginning LSA number of that I-Frame in the SAVE_LSA register.

6.Genii resets the sector counter at the beginning of the I-Frame.

7.At the end of that I-Frame, Genii saves the sector counter value in the SAVE_SECTOR register.

8.Genii only sets HINT pin — if the interrupt enable bit is set — at the end of the I-Frame before the LSA number and the sector counter are saved in the FIFO.

Bit 1: NOPSPACE_INT

No Space Interrupt

Set when the space available register reaches to '0'. In normal working condition, this bit should never get set because firmware should add to space available when it gets interrupted by SPTHRESHOLD_INT bit already.

When this condition happens, Genii will not receive or transmit until firmware fixes it by adding more space to space available register.

Bit 0: SPHRESHOLD_INT

Space Threshold Interrupt

This bit is set when the space available counter decrements to or below the value programmed in space threshold register.

REG - 83D2h – 83D3h I2_AVDATASIZE: I2 AV Transfer Size Register

R/W	15	14	13	12	11	10	9	8
	I2_DATASIZE[15:8]							
	0	0	0	0	0	0	0	0
83D2h	7	6	5	4	3	2	1	0
	I2_DATASIZE[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 0: I2_DATASIZE

ISOC channel 1 Data size

Firmware programs this register to let Genii know the size of each packet in word unit (16 bits). Genii uses this information to pack data to the sector size required by the disk controller chip.

REG - 82D4h–82D5 I2 STARTLSA_LO: I2 Start LSA Low Register

R/W	31	30	29	28	27	26	25	24
83D5h	I2_STARTLSA[31:24]							
	0	0	0	0	0	0	0	0
83D4h	23	22	21	20	19	18	17	16
	I2_STARTLSA[23:16]							
	0	0	0	0	0	0	0	0

Before starting the channel, firmware programs a number to this register. At each receiving packet, Genii packs them together into a sector, and increases the LSA counter at each sector. This LSA counter is used to keep track

of the I-Frame, and is saved in the SAVE_LSA register. This makes it available for firmware to use later for search algorithm.

Bits 31 — 16: I2_STARTLSA

REG - 82D6h–82D7 I2 STARTLSA_HI: I2 Start LSA High Register

R/W	15	14	13	12	11	10	9	8
83D7h	I2_STARTLSA[15:8]							
	0	0	0	0	0	0	0	0
83D6h	7	6	5	4	3	2	1	0
	I2_STARTLSA[7:0]							
	0	0	0	0	0	0	0	0

Before starting the channel, firmware programs a number to this register. At each receiving packet, Genii packs them together into a sector, and increases the LSA counter at each sector. This LSA counter is used to keep track of the I-Frame, and is saved in the SAVE_LSA register. This makes it available for firmware to use later for search algorithm.

Bits 15 — 0: I2_STARTLSA

REG - 83D8h–83D9h I2_TIMESTAMPL: I2 Time Stamp L Register

R/W	15	14	13	12	11	10	9	8
8399h	I2_TIMESTAMP[15:8]							
	0	0	0	0	0	0	0	0
83D8h	7	6	5	4	3	2	1	0
	I2_TIMESTAMP[7:0]							
	0	0	0	0	0	0	0	0

Bit 15–0: I2_TIMESTAMP[15:0]

The least significant timestamp[15:0] bits from Time Stamp counter.

REG - 83DAh–83DBh I2_TIMESTAMPM: I2 Time Stamp M Register

R/W	31	30	29	28	27	26	25	24
83DBh	I2_TIMESTAMP[31:24]							
	0	0	0	0	0	0	0	0
83DAh	23	22	21	20	19	18	17	16
	I2_TIMESTAMP[23:16]							
	0	0	0	0	0	0	0	0

Bits 31–16: I2_TIMESTAMP[31:26]

Bits from the Time Stamp counter.

REG - 83DCh – 83DDh I2_TIMESTAMPH: I2 Time Stamp H Register

R/W	15	14	13	12	11	10	9	8
	Reserved						I2_TIMESTAMP[41:40]	
	0	0	0	0	0	0	0	0
83DCh	7	6	5	4	3	2	1	0
	I2_TIMESTAMP[39:32]							
	0	0	0	0	0	0	0	0

Bits 15–10: Reserved

Not used.

Bits 9–0: I2_TIMESTAMP[41:32]

Bits from the Time Stamp register.

REG - 83DEh–83DFh I2_SECTAVLB: I2 Sector Available Register

R/W	15	14	13	12	11	10	9	8
	I2_SECTAVLB[15:8]							
	0	0	0	0	0	0	0	0
83DEh	7	6	5	4	3	2	1	0
	I2_SECTAVLB[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 0: I2_SECTAVLB

Sector Available (in sector unit)

Before starting the channel, firmware programs this register with a sector available number.

If the channel is in Receive mode, this register indicates the available space for receiving data. If the channel is in Transmit mode, this register shows the number of sectors in buffer memory that are ready to be transmitted.

At the end of each sector, Genii decrements this register by 1. Genii stops transferring when this register reaches 0. Interrupt is set when this register is equal to the I2_SECTORTHRESHOLD register.

Firmware can modify this register by writing to the I2_ADDSECAVLB register. In this case, Genii adds the value of I1_SECTAVLB with the value of I2_ADDSECAVLB, then stores that sum in I2_SECTAVLB

REG 83E0h–83E1:I2_ADDSECAVLB Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0
83E0h	7	6	5	4	3	2	1	0
	ADDSECAVLB[7:0]							
	0	0	0	0	0	0	0	0

Bits 7 – 0: ADDSECAVLB[7:0]

Each time firmware writes to this register, Genii performs the addition of this register to the I1_SECAVLB register.

For more information, see REG - 83DEh–83DFh I2_SECTAVLB: I2 Sector Available Register

REG - 83E2h – 83E3h I2_SECTHRESHLD Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0
83E2h	7	6	5	4	3	2	1	0
	I2_SECTHRESHLD[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 8: Reserved

Not used.

Bits 7–0: I2_SECTHRESHLD[7:0]

The threshold level of space (sector) that is available to assert space threshold interrupt to firmware.

REG - 83E4h – 83E5h I2_FFTHRESHLD Register

R/W	15	14	13	12	11	10	9	8
	Reserved							I2_FFTHRESHLD[8]
	0	0	0	0	0	0	0	0
83E4h	7	6	5	4	3	2	1	0
	I2_FFTHRESHLD[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 9: Reserved

Not used.

Bits 8–0: I2_FFTHRESHLD[8:0]

The threshold level of the Buffer FIFO to assert space dma_reg to buffer control.

REG - 83E6h – 83E7h I2_TSEARLYWIN Register

R/W	15	14	13	12	11	10	9	8
	I2_TSEARLYWIN[15:8]							
	0	0	0	0	0	0	0	0
83E6h	7	6	5	4	3	2	1	0
	I2_TSEARLYWIN[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 0: I2_SEARLYWIN[15:0]

When receiving, TSEARLYWIN register is subtracted from the Time Stamp counter (early window). This informs how early to expect the incoming PCR value. If PRC value is less than the early window, it is treated as a discontinuous PCR value.

When transmitting, TSEARLYWIN register is substitute from Time Stamp from the data index (early window). This informs how early the packet can be transmitted.

REG - 83E8h – 83E9h I2_TSLATEWIN Register

R/W	15	14	13	12	11	10	9	8
	I2_TSLATEWIN[15:8]							
	0	0	0	0	0	0	0	0
83E8h	7	6	5	4	3	2	1	0
	I2_TSLATEWIN[7:0]							
	0	0	0	0	0	0	0	0

Bits 15–0: I2_TSLATEWIN

While receiving, the TSLATEWIN register is added to Time Stamp counter (late window). This informs how late to expect the incoming PCR value. If PCR value is greater than late window, it will be treated as a discontinuous PCR value.

While transmitting, the TSLATEWIN register is added to Time Stamp from the data index (late window). If the current Time Stamp counter value is greater than that of late window, the packet will be discarded.

REG - 83EAh – 83EBh I2_ERRSTATUS Register

R/W	15	14	13	12	11	10	9	8
	Reserved		QFIFO_ STATUS	PAT_INT_ STATUS	PMT_INT_ STATUS	AVFIFO_ ERR	ROLLB_ INT_ STATUS	ROLLA_ INT_ STATUS
	0	0	0	0	0	0	0	0
83EAh		7	6	5	4	3	2	1	0
		BFIFO_ ERROR	MISS_ SYNC_ ERROR	DATA_SIZE _MIS- MATCHED	EO- STREAM	I_FRAME_ FULL	I_FRAME_ DETECTED	NO_SPACE _STATUS	SPACE_ THRESHOLD_ STATUS
		0	0	0	0	0	0	0	0

This read-only register shows the errors and status of the internal Genii condition.

Bits 15–14: Reserved

Read as zeroes.

BIT 13: QFIFO_STATUS

When this bit is set, Firmware must go to the Q_STATUS2 register

BIT 12: PAT_INT_STATUS

When set, this bit indicates that PAT table is detected from the MPEG2 stream.

BIT 11: PMT_INT_STATUS

When set, this bit indicates that PMT table is detected from the MPEG2 stream.

Bit 10: AVFIFO_ERR

When this bit is set, it indicates that AVFIFO ever-run is detected.

Bit 9: ROLLB_INT_STATUS

When set, this bit indicates that roll-over Pointer B has occurred.

Bit 8: ROLLA_INT_STATUS

When set, this bit indicates that roll-over Pointer A has occurred.

Bit 7: BFIFO_ERROR

When this bit is set, it indicates that either buffer FIFO overflow or buffer FIFO underflow has occurred.

Bit 6: MISS_SYNC_ERROR

When this bit is set, it indicates that MISS_SYNC error is detected.

Bit 5: DATA_SIZE_MISMATCHED

When set, this bit indicates that the receiving AV data packet is either too long or too short.

Bit 4: EOSTREAM

When set, this bit indicates that the End of Stream condition is detected while transmitting to AV system.

Bit 3: I_FRAME_FULL

When this bit is set, it indicates that the I FRAME TABLE FIFO is full.

Bit 2: I_FRAME_DETECTED

When this bit is set, it indicates that the I-FRAME from MPEG2 stream has been detected.

Bit 1: NO_SPACE STATUS

When this bit is set, either no space is available in SDRAM while receiving, or no sector is available in SDRAM while transmitting.

Bit 0: SPACE_THRESHOLD_STATUS

When this bit is set, the value in the I1_SECTAVLB register is less than the value in the I1_SETCTHRESHOLD register.

REG - 83ECh – 83EDh I2_CONFIG2 Register

	R/W	15	14	13	12	11	10	9	8
		Reserved							
		0	0	0	0	0	0	0	0
83ECh		7	6	5	4	3	2	1	0
		Reserved		EXT_EARLY_WIN[2:0]			FIFOSEL[1:0]		TEST_ MODE
		0	0	0	0	0	0	0	0

Bits 15–6: Reserved

Read as zeroes.

Bits 5–3: EXT_EARLY_WIN[2:0]

These bits extend the early WIN to determine late packet. Each unit in this register is equal to about 11 µs.

The early window is subtracted by EXT_EARLY_WIN to form the extended_early_window. If the time stamp of the packet is not between_early_window and late_window, then the packet will be dropped.

BitS 2–1: FIFOSEL[1:0]

FIFO select:

00: I FRAME FIFO
01: AV FIFO
10: BUFFER FIFO
11: Q FIFO

Bit 0: TEST_MODE

To test FIFOs, enable this bit.

REG - 83EEh – 83EFh I2_FIFODAT Register

R/W	15	14	13	12	11	10	9	8
	I2_FIFODAT[15:8]							
	0	0	0	0	0	0	0	0
83EEh	7	6	5	4	3	2	1	0
	I2_FIFODAT[7:0]							
	0	0	0	0	0	0	0	0

Bits 15–0: I2_FIFODAT[15:0]

16-bit FIFO data register.

This is the FIFO data of the FIFO selected by FIFOSEL in the I2-CONFIG2 register.
In test mode, Firmware can read/write data to/from FIFOs.

REG - 83F0h – 83F1h I2_FIFOADD Register

R/W	15	14	13	12	11	10	9	8
	Reserved							I2_ FIFOADD[8:]
	0	0	0	0	0	0	0	0
83F0h	7	6	5	4	3	2	1	0
	I2_FIFOADD[7:0]							
	0	0	0	0	0	0	0	0

Bits 15–9: Reserved

Read as zeroes

Bits 8–0: I2_FIFOADD[8:0]

9-bit FIFO address register.

This is the FIFO address of the FIFO selected by FIFOSEL in the I2_CONFIG2 register:

- When firmware is writing data to the FIFO, this is the write address.
- When firmware is reading data from the FIFO, this is the read address.

REG – 83F2h—83F3h I2_UPPCRPID Register

R/W	15	14	13	12	11	10	9	8
	disable_tables_parsers	PAT_crc_err_ena	PMT_crc_err_ena	PCR_PID[12:8]				
	0	0	0	0	0	0	0	0

84F2h	7	6	5	4	3	2	1	0
R/W	PCR_PID[7:0]							
	0	0	0	0	0	0	0	0

Bit 15: disable_table_parsers (R/W)

When set to '1', this bit disables table parsers and allows uP to override PCR_PID and Video_pid values

Bit 14: PAT_crc_err_ena (R/W)

When set to '1', this bit enables crc checker in PAT parser.

Bit 13: PMT_crc_err_ena (R/W)

When set to '1', this bit enables crc checker in PMT parser.

Bits [12:0]: PCR_PID[12:0] (Write)

Firmware controls the PID value.

Bits [12:0]: PCR_PID[12:0] (Read)

When this bit = '1', read back firmware written value, else read back parser-detected PID value.

REG – 83F4h – 83F5h I2_UPVIDEOPID Register

R/W	15	14	13	12	11	10	9	8
	Reserved			Video_PID[12:8]				
	0	0	0	0	0	0	0	0

84F4h	7	6	5	4	3	2	1	0
R/W	Video_PID[12:8]							
	0	0	0	0	0	0	0	0

Bits [15:13]: Reserved

Read as zeroes.

Bits [12:0:] Video_PID[12:0] (Write)

Firmware controls PID value.

Bits [12:0] Video_PID[12:0] (Read)

When this bit = '1', read back firmware written value, else read back parser-detected PID value.

REG – 83F6h–83F7h I2_UPPARSSM Register

R/W	15	14	13	12	11	10	9	8
	up_pat_parsed (R/W)	Reserved		uppmtsm[4:0]				
	0	0	0	0	0	0	0	0

83F6h	7	6	5	4	3	2	1	0
R/W	Reserved			uppatsm[4:0]				
	0	0	0	0	0	0	0	0

Bit 15: up_pat_parsed (R/W)

Forces 'pat_parsed' state when written.

Monitors the state when read.

Bits 14:13: Reserved

Read as zeroes.

Bits 12:8: uppmtsm[4:0]

Forces PMT FSM state when written.

Monitors the state when read.

Bits 7:5]: Reserved

Not used. Read as zeroes.

Bits 4:0: uppatsm[4:0]

Forces PAT FSM state when written.

Monitors the state when read.

REG – 83F8h–83F9h I2_UPPARSSTAT (Read Only) Register

R/O	15	14	13	12	11	10	9	8
	Reserved		PMT_ adapt_err	PMT _ cont_err	PMT_ crc_err	PMT_ pointer_err	PMT_ section_ err	PMT_ table_ id_err
	0	0	0	0	0	0	0	0

84F8h	7	6	5	4	3	2	1	0
R/O	Reserved			PAT_ adapt_err	PAT_ cont_err	PAT_ crc_err	PAT_ pointer_err	PAT_table _id_err
	0	0	0	0	0	0	0	0

Bits [15:14] Reserved

Read as zeroes

Bit 13: PMT_adapt_err

This bit indicates adaptation length field error

Bit 12: PMT_cont_err

This bit indicates continuity counter error for tables spanning multiple packets.

Bit 11: PMT_crc_err

This bit indicates crc error detected

Bit 10: PMT_pointer_err

This bit indicates pointer value error

Bit 9: PMT_section_err

This bit indicates invalid values for section_number and/or last_section_number fields.

Bit 8: PMT_table_id_err

This bit indicates table ID not equal to 02h

Bit [7:5] Reserved

Read as zeroes

Bit 4: PAT_adapt_err

This bit indicates adaptation length field error

Bit 3: PAT_cont_err

This bit indicates continuity counter error for tables spanning multiple packets.

Bit 2: PAT_crc_err

This bit indicates crc error detected

Bit 1: PAT_pointer_err

This bit indicates pointer value error

Bit 0: PAT_table_id_err

This bit indicates table ID not equal to 00

REG - 83FAh – 83FBh I2_PATTABLEDAT: I2 PAT Table Data Register

R/O

15	14	13	12	11	10	9	8
PATTABLEDAT[15:8]							
0	0	0	0	0	0	0	0

84FAh

R/O

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

Bits 15 – 0: PATTABLEDAT[15:0]

16-bit PAT Table Data.

This is the data that is stored in the PAT table. It contains information of the PAT version change.

REG - 83FCh – 83FDh I2_PMTTABLEDAT: I2 PMT Table Data Register

R/O

15	14	13	12	11	10	9	8
PMTTABLEDAT[15:8]							
0	0	0	0	0	0	0	0

84FAh

R/O

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

Bits 15 – 0: PMTTABLEDAT[15:0]

16-bit PMT Table Data.

This is the data that is stored in the PMT table. It contains information of the PMT version change.

3.1.8 ISOC3 Channel

REG - 8400h – 8401h I3_ADDR: I3 Address Hi/Lo

R/W	15	14	13	12	11	10	9	8
	I3_ADDR[15:8]							
	0	0	0	0	0	0	0	0
8400h	7	6	5	4	3	2	1	0
	I3_ADDR[7:0]							
	0	0	0	0	0	0	0	0

(I1base + 00)

Bits 15 — 0: I3_ADDR[15:0]

ISOC channel 1 address

This register, with 8382h I1, forms the 24 bit address to DRAM. Firmware programs this address initially, then Genii increments this address at the end of each access.

REG - 8402h – 8403h I3_ADPG: I3 Page Address Register

R/W	31	30	29	28	27	26	25	24
	Reserved							
	0	0	0	0	0	0	0	0
8402h I1	23	22	21	20	19	18	17	16
	I3_ADPG[23:16]							
	0	0	0	0	0	0	0	0

(I2 base + 02)

Bits 31 — 24: Reserved

Bits 23 — 16: I3_ADPG[23:16]

ISOC channel 1 address

This register, with register I1_ADDR, forms the 24 bit address to DRAM. Firmware programs this address initially, then Genii increments this address at the end of each access.

REG - 8404h – 8405h I3_ROLA: I3 Rollover Address Register (pair A)

R/W	23	22	21	20	19	18	17	16
	I3_ROLPG[23:16]							
	0	0	0	0	0	0	0	0
8404h I1_ROLPG	15	14	13	12	11	10	9	8
	I3_ROLPG[15:0]							
	0	0	0	0	0	0	0	0

Bits 23 — 16: i3_ROLPG_A

ISOC channel 1 roll over address, pair A

Bits 15 — 0: i3_ROLHI_A

ISOC channel 1 roll over address, pair A

REG -8406h – 8407h I3_RLDA: I3 Re-Load Address Register (pair A)

R/W	23	22	21	20	19	18	17	16
	I3_RLPG[23:16]							
	0	0	0	0	0	0	0	0
8406h I2_RLHI	15	14	13	12	11	10	9	8
	I3_RLHI[15:8]							
	0	0	0	0	0	0	0	0

Bits 23 — 16: I3_RLPG

ISOC channel 1 re-load address, pair A

BITS 15 — 8: I3_RLHI

ISOC channel 1 re-load address, pair A

REG - 8408h – 8409h I3_ROLB: I3 Rollover Address Register (pair B)

R/W	23	22	21	20	19	18	17	16
	I3_ROLPG_B[23:16]							
	0	0	0	0	0	0	0	0
8408h I1_ROLHI_B	15	14	13	12	11	10	9	8
	I3_ROLHI_B[15:8]							
	0	0	0	0	0	0	0	0

Bits 23 — 16: I3_ROLPG_B

ISOC channel 1 roll over address, pair B

Bits 15 — 8: I3_ROLHI_B

ISOC channel 1 roll over address, pair B

REG - 840Ah – 840Bh I3_RLDB: I3 Re-Load Address Register (pair B)

R/W	23	22	21	20	19	18	17	16
	I3_RLPG_B[23:16]							
	0	0	0	0	0	0	0	0
840Ah RLHI_B	15	14	13	12	11	10	9	8
	I3_RLHI_B[15:8]							
	0	0	0	0	0	0	0	0

Bits 23 — 16: I3_RLPG_B

ISOC channel 1 re-load address, pair B

Bits 15 — 8: I3_RLHI_B

ISOC channel 1 re-load address, pair B

REG - 840Ch – 840Dh I3_CONFIG: I3_Configuration

R/W	15	14	13	12	11	10	9	8
	EOSTREAM	DELNULL_ ENA	DVB_MODE	DISABLE_ TP	AVCLKOUTSEL[1:0]		AVCLK-OUTENA	SYNCPCR ENA
	0	0	0	0	0	0	0	0
840Ch	7	6	5	4	3	2	1	0
	BFR_SIZE			MPEG_ FORMAT	PAUSE	DISABLE_ TS	XFERENA	INDIR
	0	0	0	0	0	0	0	0

Bit 15: EOSTREAM

End of stream.

This bit is set by firmware when recording is being stopped.

Hardware sets the eostream bit in the sector header. While transmitting, hardware detects this bit and asserts interrupt to firmware.

Bit 14: DELNULL_ENA

Delete null packet enable.

When this bit is set, the null packets are detected and dropped while receiving.

Bit 13: DVB_MODE

Enable DVB mode.

Bits 12: DISABLE_TP

To disable Trick Play, set this bit.

Bit 11–10: AVCLKOUTSEL[1:0]

Bits AVCLKOUTSEL[0] and AVCLKOUTSEL[1] are used to select the frequency of the AVCLK output:

00: AV1CLK (system dependent)
01: AV1CLK (system dependent)
10: 20MHz clock
11:TSCLK (27MHz)

Bit 9: AVCLKOUTENA

Enables driving AVCLK externally.

Bit 8: SYNCPCRENA

Enables time stamp counter to synchronize with incoming PCR.

Bits 7–5: BFR_SIZE

These 3 bits set the limit of each DMA channel when it accesses DRAM. This helps set up a fair arbitration scheme, which depends on the bit rate of each DMA channel.

000: each request to DRAM access will be 8 words long
001: each request to DRAM access will be 16 words long
010: each request to DRAM access will be 32 words long
011: each request to DRAM access will be 64 words long
100: each request to DRAM access will be 128 words long
101: each request to DRAM access will be 256 words long
110: each request to DRAM access will be 512 words long
111: Reserved

Bit 4: MPEG_FORMAT

MPEG Format

When this bit is programmed to '1', this channel will be receiving/transmitting packets in MPEG 2 Transport Stream format.

Bit 3: PAUSE

To pause the stream, set this bit.

To receive or to transmit, clear this bit.

Bit 2: DISABLE_TS

Disable Time Stamp.
When this bit is programmed to '1', Genii will not generate sector index and data index for each sector it stores in DRAM. Data will be stored as is without any index.

Bit 1: XFERENA

Transfer Enable.

After finishing all necessary registers in this DMA channel, firmware set XFERENA bit to '1' to enable the channel to receive or transmit data. To stop the channel from receiving/transmitting data, firmware reset this bit to '0'.

Bit 0: INDIR

In direction.

This bit is set to '1' to let this DMA channel know that it should be RECEIVING data from the link and store that data in Dram.

REG - 840Eh – 840Fh I3_INTENA: I3 Interrupt Enable Register

R/W	15	14	13	12	11	10	9	8
	Reserved			PAT_INT_ ENA	PMT_INT_ ENA	Reserved	ROLLB_ INT_ENA	ROLLA_ INT_ENA
	0	0	0	0	0	0	0	0

840Eh	7	6	5	4	3	2	1	0
	BUFERR_ INT_ENA	MISS_SYNC _INT_ENA	Data_size_ mis-matched _int_ena	EOSTREAM _INT_ENA	IFULL_ INT_ENA	ISTART_ INT_ENA	NOSPACE _INT_ENA	SPHRESH OLD_INT_ ENA
	0	0	0	0	0	0	0	0

Each bit in this reg matches one on one w/ ea. bit in the following Interrupt reg.It's used to let firmware choose which interrupt is worth setting the HINT output pin.

Examples follow:

Example 1

- SPTHRESHOLD_INT_ENA bit is programmed to ‘1’.
- Genii detects the condition to set SPTHRESHOLD_INT bit.
- It sets SPTHRESHOLD_INT bit to ‘1’, and asserts HINT output pin.

Example 2

- SPTHRESHOLD_INT_ENA bit is programmed to ‘0’ Genii detects the condition to set SPTHRESHOLD_INT bit.
- It sets SPTHRESHOLD_INT bit to ‘1’, but does not assert HINT output pin.

Bits 15—13: Reserved

Not used.

Bit 12: PAT_INT_ENA

PAT Interrupt Enable.

Bit 11: PMT_INT_ENA

PMT Interrupt Enable.

Bit 9: ROLLB_INT_ENA

Enables Rollover B interrupt.

Bit 8: ROLLA_INT_ENA

Enables Rollover A interrupt.

Bit 7: BUFERR_INT_ENA

Enables Buffer Error (underflow/overflow) interrupt.

Bit 6: MISS_SYNC_INT_ENA

Enables missed synchronize interrupt.

Bit 5: DATA_SIZE_MISMATCHED_INT_ENA

Enables Data_size_mismatched_int.

Bit 4: EOSTREAM_INT_ENA

End of Stream Interrupt Enable

Bit 3: IFULL_INT_ENA

I-frame table Full Interrupt Enable

Bit 2: ISTART_INT_ENA

I-Frame Start Interrupt Enable

Bit 1: NOPSPACE_INT_ENA

No Space Interrupt Enable

Bit 0: SPHRESHOLD_INT_ENA:

Space Threshold Interrupt Enable

REG - 8410h – 8411h I3 INT: I3 Interrupt Register

R/W	15	14	13	12	11	10	9	8
	Reserved			PAT_INT	PMT_INT	Reserved	ROLLB_ INT	ROLLA_ INT
	0	0	0	0	0	0	0	0
8411h	7	6	5s	4	3	2	1	0
	BUF_ERR_ INT	MISS_SYNC _BIT	Data_size_ mis-matched_ int	EO-STREAM _INT	IFULL_INT	ISTART_ INT	NOSPACE _INT	SPHRESHOLD _INT
	0	0	0	0	0	0	0	0

Internal condition of Genii will set the corresponding bit in this register to let firmware know the condition. To clear the bit, firmware writes a '1' to the desired bit.

Bits 15 — 13: Reserved

Read as zeroes.

Bit 12: PAT_INT

PAT Interrupt.

This bit is set when a PAT version change is detected in the video stream.

Bit 11: PMT_INT

PMT Interrupt.

This bit is set when a PMT version change is detected in the video stream.

Bit 9: ROLLB_INT

This interrupt informs firmware that rollover B has occurred.

Bit 8: ROLLA_INT

This interrupt informs firmware that rollover A has occurred.

Bit 7: BUF_ERR_INT

Buffer Error (underflow/overflow) interrupt.

Bit 6: MISS_SYNC_INT

This bit is valid only when is in the receive direction, and Genii is programmed in MPEG 2 mode.

Set when the first byte of an MPEG 2 packer is not a SYNC byte (SYNC byte has the value of 47h).

Bit 5: DATA_SIZE_MISMATCHED_INT

This bit is only valid in the receive direction.

Set when the receiving packet length does not match the value programmed in the AVXFERSIZE register.

Bit 4: EOSTREAM_INT

End Of Stream Interrupt

Set when Genii detects the end of frame from the transmit packet. This bit is valid only in transmit to the link direction.

Bit 3: IFULL_INT

I-frame table Full Interrupt.

This bit is set when the I-frame FIFO is full.

Bit 2:ISTART_INT

I-frame Start Interrupt.

This bit is set when a start of an I-frame is detected in the video stream. Firmware can read this information from FIFODAT register.

9.When Genii detects an incoming I-Frame, it saves the beginning LSA number of that I-Frame in

the SAVE_LSA register.

10.Genii resets the sector counter at the beginning of the I-Frame.

11.At the end of that I-Frame, Genii saves the sector counter value in the SAVE_SECTOR register.

12.Genii only sets HINT pin, if the interrupt enable bit is set, at the end of the I-Frame before the LSA number and the sector counter are saved in the FIFO.

Bit 1: NOSPACE_INT

No Space Interrupt

Set when the space available register reaches to '0'. In normal working condition, this bit should never get set because firmware should add to space available when it gets interrupted by SPTHRESHOLD_INT bit already.

When this condition happens, Genii will not receive or transmit until firmware fixes it by adding more space to space available register.

Bit 0: SPHRESHOLD_INT

Space Threshold Interrupt

This bit is set when the space available counter decrements to or below the value programmed in space threshold register.

REG - 8412h – 8413h I3_AVDATASIZE: I3 AV Transfer Size Register

R/W	15	14	13	12	11	10	9	8
	I3_DATASIZE[15:8]							
	0	0	0	0	0	0	0	0
8412h	7	6	5	4	3	2	1	0
	I3_DATASIZE[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 0: I3_DATASIZE

ISOC channel 1 Data size

Firmware programs this register to let Genii know the size of each packet in word unit (16 bits). Genii uses this information to pack data to the sector size required by the disk controller chip.

REG - 8414h–8415 I3 STARTLSA_LO: I3 Start LSA Low Register

R/W	31	30	29	28	27	26	25	24
8415h	I3_STARTLSA[31:24]							
	0	0	0	0	0	0	0	0
8414h	23	22	21	20	19	18	17	16
	I3_STARTLSA[23:16]							
	0	0	0	0	0	0	0	0

Before starting the channel, firmware programs a number to this register. At each receiving packet, Genii packs them together into a sector, and increases the LSA counter at each sector. This LSA counter is used to keep track of the I-Frame, and is saved in the SAVE_LSA register. This makes it available for firmware to use later for search algorithm.

Bits 31 — 16: I2_STARTLSA

REG - 8416h–8417 I3 STARTLSA_HI: I3 Start LSA High Register

R/W	15	14	13	12	11	10	9	8
8417h	I3_STARTLSA[15:8]							
	0	0	0	0	0	0	0	0
8416h	7	6	5	4	3	2	1	0
	I3_STARTLSA[7:0]							
	0	0	0	0	0	0	0	0

Before starting the channel, firmware programs a number to this register. At each receiving packet, Genii packs them together into a sector, and increases the LSA counter at each sector. This LSA counter is used to keep track of the I-Frame, and is saved in the SAVE_LSA register. This makes it available for firmware to use later for search algorithm.

Bits 15 — 0: I2_STARTLSA

REG - 8418h–8419h I3_TIMESTAMPL: I3 Time Stamp L Register

R/W	15	14	13	12	11	10	9	8
8419h	I3_TIMESTAMP[15:8]							
	0	0	0	0	0	0	0	0
8418h	7	6	5	4	3	2	1	0
	I3_TIMESTAMP[7:0]							
	0	0	0	0	0	0	0	0

Bit 15–0: I3_TIMESTAMP[15:0]

The least significant timestamp[15:0] bits from Time Stamp counter.

REG - 841Ah–841Bh I3_TIMESTAMPM: I3 Time Stamp M Register

R/W	31	30	29	28	27	26	25	24
841Bh	I3_TIMESTAMP[31:24]							
	0	0	0	0	0	0	0	0
841Ah	23	22	21	20	19	18	17	16
	I3_TIMESTAMP[23:16]							
	0	0	0	0	0	0	0	0

Bits 31–16: I3_TIMESTAMP[31:26]

Bits from the Time Stamp counter.

REG - 841Ch – 841Dh I3_TIMESTAMPH: I3 Time Stamp H Register

R/W	15	14	13	12	11	10	9	8
	Reserved						I3_TIMESTAMP[41:40]	
	0	0	0	0	0	0	0	0
841Ch	7	6	5	4	3	2	1	0
	I3_TIMESTAMP[39:32]							
	0	0	0	0	0	0	0	0

Bits 15–10: Reserved

Not used.

Bits 9–0: I3_TIMESTAMP[41:32]

Bits from the Time Stamp register.

REG - 841Eh–841Fh I3_SECTAVLB: I3 Sector Available Register

R/W	15	14	13	12	11	10	9	8
	I3_SECTAVLB[15:8]							
	0	0	0	0	0	0	0	0
841Eh	7	6	5	4	3	2	1	0
	I3_SECTAVLB[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 0: I3_SECTAVLB

Sector Available (in sector unit)

Before starting the channel, firmware programs this register with a sector available number.

If the channel is in Receive mode, this register indicates the available space for receiving data. If the channel is in Transmit mode, this register shows the number of sectors in buffer memory that are ready to be transmitted.

At the end of each sector, Genii decrements this register by 1. Genii stops transferring when this register reaches 0. Interrupt is set when this register is equal to the I3_SECTORTHRESHOLD register.

Firmware can modify this register by writing to the I3_ADDSECAVLB register. In this case, Genii adds the value of I1_SECTAVLB with the value of I3_ADDSECAVLB, then stores that sum in I3_SECTAVLB

REG 8420h–8421h I3_ADDSECAVLB Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0
8420h	7	6	5	4	3	2	1	0
	ADDSECAVLB[7:0]							
	0	0	0	0	0	0	0	0

Writing to this register causes it to automatically add to space available counter.

Bits 15 — 8: Reserved

Not used.

Bits 7–0: ADDSECAVLB[7:0]

REG - 8422h – 8423h I3_SECTHRESHLD Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0
8422h	7	6	5	4	3	2	1	0
	I3_SECTHRESHLD[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 8: Reserved

Not used.

Bits 7–0: I3_SECTHRESHLD[7:0]

The threshold level of space (sector) that is available to assert space threshold interrupt to firmware.

REG - 8424h – 8425h I3_FFTHRESHLD Register

R/W	15	14	13	12	11	10	9	8
	Reserved							I3_ FFTHRES HLD[8]
	0	0	0	0	0	0	0	0
8424h	7	6	5	4	3	2	1	0
	I3_FFTHRESHLD[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 9: Reserved

Not used.

Bits 8–0: I3_FFTHRESHLD[8:0]

The threshold level of the Buffer FIFO to assert space dma_reg to buffer control.

REG - 8426h – 8427h I3_TSEARLYWIN Register

R/W	15	14	13	12	11	10	9	8
	I3_TSEARLYWIN[15:8]							
	0	0	0	0	0	0	0	0
8426h	7	6	5	4	3	2	1	0
	.I3_TSEARLYWIN[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 — 0: I3_TSEARLYWIN[15:0]

When receiving, TSEARLYWIN register is subtracted from the Time Stamp counter (early window). This informs how early to expect the incoming PCR value. If PRC value is less than the early window, it is treated as a discontinuous PCR value.

When transmitting, TSEARLYWIN register is substitute from Time Stamp from the data index (early window). This informs how early the packet can be transmitted.

REG - 8428h – 8429h I3_TSLATEWIN Register

R/W	15	14	13	12	11	10	9	8
	I3_TSLATEWIN[15:8]							
	0	0	0	0	0	0	0	0
8428h	7	6	5	4	3	2	1	0
	I3_TSLATEWIN[7:0]							
	0	0	0	0	0	0	0	0

Bits 15–0: I3_TSLATEWIN

While receiving, the TSLATEWIN register is added to Time Stamp counter (late window). This informs how late to expect the incoming PCR value. If PCR value is greater than late window, it will be treated as a discontinuous PCR value.

While transmitting, the TSLATEWIN register is added to Time Stamp from the data index (late window). If the current Time Stamp counter value is greater than that of late window, the packet will be discarded.

REG - 842Ah – 842Bh I3_ERRSTATUS Register

R/W	15	14	13	12	11	10	9	8
	Reserved		QFIFO_ STATUS	PAT_INT_ STATUS	PMT_INT_ STATUS	AVFIFO_ ERR	ROLLB_ INT_ STATUS	ROLLA_ INT_ STATUS
	0	0	0	0	0	0	0	0
83EAh	7	6	5	4	3	2	1	0
	BFIFO_ ERROR	MISS_ SYNC_ ERROR	DATA_SIZE _MIS- MATCHED	EO- STREAM	I_FRAME_ FULL	I_FRAME_ DETECTED	NO_SPACE _STATUS	SPACE_ THRESHOLD_ STATUS
	0	0	0	0	0	0	0	0

This read-only register shows the errors and status of the internal Genii condition.

Bits 15–14: Reserved

Read as zeroes.

BIT 13: QFIFO_STATUS

When this bit is set, Firmware must go to the Q_STATUS2 register

BIT 12: PAT_INT_STATUS

When set, this bit indicates that PAT table is detected from the MPEG2 stream.

BIT 11: PMT_INT_STATUS

When set, this bit indicates that PMT table is detected from the MPEG2 stream.

Bit 10: AVFIFO_ERR

When this bit is set, it indicates that AVFIFO ever-run is detected.

Bit 9: ROLLB_INT_STATUS

When set, this bit indicates that roll-over Pointer B has occurred.

Bit 8: ROLLA_INT_STATUS

When set, this bit indicates that roll-over Pointer A has occurred.

Bit 7: BFIFO_ERROR

When this bit is set, it indicates that either buffer FIFO overflow or buffer FIFO underflow has occurred.

Bit 6: MISS_SYNC_ERROR

When this bit is set, it indicates that MISS_SYNC error is detected.

Bit 5: DATA_SIZE_MISMATCHED

When set, this bit indicates that the receiving AV data packet is either too long or too short.

Bit 4: EOSTREAM

When set, this bit indicates that the End of Stream condition is detected while transmitting to AV system.

Bit 3: I_FRAME_FULL

When this bit is set, it indicates that the I FRAME TABLE FIFO is full.

Bit 2: I_FRAME_DETECTED

When this bit is set, it indicates that the I-FRAME from MPEG2 stream has been detected.

Bit 1: NO_SPACE STATUS

When this bit is set, either no space is available in SDRAM while receiving, or no sector is available in SDRAM while transmitting.

Bit 0: SPACE_THRESHOLD_STATUS

When this bit is set, the value in the I1_SECTAVLB register is less than the value in the I1_SETCTHRESHOLD register.

REG - 8420Ch – 842Dh I3_CONFIG2 Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0

842Ch	7	6	5	4	3	2	1	0
	Reserved		EXT_EARLY_WIN[2:0]			FIFOSEL[1:0]		TEST_ MODE
	0	0	0	0	0	0	0	0

Bits 15–6: Reserved

Read as zeroes.

Bits 5–3: EXT_EARLY_WIN[2:0]

These bits extend the early WIN to determine late packet. Each unit in this register is equal to about 11 µs.

The early window is subtracted by EXT_EARLY_WIN to form the extended_early_window. If the time stamp of the packet is not between_early_window and late_window, then the packet will be dropped.

BitS 2–1: FIFOSEL[1:0]

FIFO select:

00: I FRAME FIFO
01: AV FIFO
10: BUFFER FIFO
11: Q FIFO

Bit 0: TEST_MODE

To test FIFOs, enable this bit.

REG - 842Eh – 842Fh i3_FIFODAT Register

R/W	15	14	13	12	11	10	9	8
	I3_FIFODAT[15:8]							
	0	0	0	0	0	0	0	0
842Eh	7	6	5	4	3	2	1	0
	I3_FIFODAT[7:0]							
	0	0	0	0	0	0	0	0

Bits 15–0: I3_FIFODAT[15:0]

16-bit FIFO data register.

This is the FIFO data of the FIFO selected by FIFOSEL in the DMANCNFG2 register.

REG - 8430h – 8431h I3_FIFOADD Register

R/W	15	14	13	12	11	10	9	8
	Reserved							I3_ FIFOADD[8]
	0	0	0	0	0	0	0	0
83F0h	7	6	5	4	3	2	1	0
	I3_FIFOADD[7:0]							
	0	0	0	0	0	0	0	0

Bits 15–9: Reserved

Read as zeroes

Bits 8–0: I3_FIFOADD[8:0]

9-bit FIFO address register.

This is the FIFO address of the FIFO selected by FIFOSEL in the I3_CONFIG2 register:

- When firmware is writing data to the FIFO, this is the write address.
- When firmware is reading data from the FIFO, this is the read address.

REG – 8432h—8433h I3_UPPCRPID Register

R/W	15	14	13	12	11	10	9	8
	disable_tables_parsers	PAT_crc_err_ena	PMT_crc_err_ena	PCR_PID[12:8]				
	0	0	0	0	0	0	0	0

84F2h	7	6	5	4	3	2	1	0
R/W	PCR_PID[7:0]							
	0	0	0	0	0	0	0	0

Bit 15: disable_table_parsers (R/W)

When set to '1', this bit disables table parsers and allows uP to override PCR_PID and Video_pid values

Bit 14: PAT_crc_err_ena (R/W)

When set to '1', this bit enables crc checker in PAT parser.

Bit 13: PMT_crc_err_ena (R/W)

When set to '1', this bit enables crc checker in PMT parser.

Bits [12:0]: PCR_PID[12:0] (Write)

Firmware controls the PID value.

Bits [12:0]: PCR_PID[12:0] (Read)

When this bit = '1', read back firmware written value, else read back parser-detected PID value.

REG – 8434h – 8435h I3_UPVIDEOPID Register

R/W	15	14	13	12	11	10	9	8
	Reserved			Video_PID[12:8]				
	0	0	0	0	0	0	0	0

8434h	7	6	5	4	3	2	1	0
R/W	Video_PID[12:8]							
	0	0	0	0	0	0	0	0

Bits [15:13]: Reserved

Read as zeroes.

Bits [12:0:] Video_PID[12:0] (Write)

Firmware controls PID value.

Bits [12:0] Video_PID[12:0] (Read)

When this bit = '1', read back firmware written value, else read back parser-detected PID value.

REG – 8436h–8437h I3_UPPARSSM Register

	15	14	13	12	11	10	9	8
R/W	up_pat_parsed (R/W)	Reserved		uppmtsm[4:0]				
	0	0	0	0	0	0	0	0
8436h	7	6	5	4	3	2	1	0
R/W	Reserved			uppatsm[4:0]				
	0	0	0	0	0	0	0	0

Bit 15 up_pat_parsed (R/W)

Forces 'pat_parsed' state when written.

Monitors the state when read.

Bits 14 – 13: Reserved

Read as zeroes.

Bits 12 –9 uppmtsm[4:0]

Forces PMT FSM state when written.

Monitors the state when read.

Bits 7 – 5: Reserved

Read as zeroes.

Bits 4– 0: uppatsm[4:0]

Forces PAT FSM state when written.

Monitors the state when read.

REG – 8438h–8439h I3_UPPARSSTAT (Read Only) Register

	15	14	13	12	11	10	9	8
R/O	Reserved		PMT_ adapt_err	PMT _ cont_err	PMT_ crc_err	PMT_ pointer_err	PMT_ section_ err	PMT_ table_ id_err
	0	0	0	0	0	0	0	0
8438h	7	6	5	4	3	2	1	0
R/O	Reserved			PAT_ adapt_err	PAT_ cont_err	PAT_ crc_err	PAT_ pointer_err	PAT_table _id_err
	0	0	0	0	0	0	0	0

Bits [15:14] Reserved

Read as zeroes

Bit 13: PMT_adapt_err

This bit indicates adaptation length field error

Bit 12: PMT_cont_err

This bit indicates continuity counter error for tables spanning multiple packets.

Bit 11: PMT_crc_err

This bit indicates crc error detected

Bit 10: PMT_pointer_err

This bit indicates pointer value error

Bit 9: PMT_section_err

This bit indicates invalid values for section_number and/or last_section_number fields.

Bit 8: PMT_table_id_err

This bit indicates table ID not equal to 02h

Bit [7:5] Reserved

Read as zeroes

Bit 4: PAT_adapt_err

This bit indicates adaptation length field error

Bit 3: PAT_cont_err

This bit indicates continuity counter error for tables spanning multiple packets.

Bit 2: PAT_crc_err

This bit indicates crc error detected

Bit 1: PAT_pointer_err

This bit indicates pointer value error

Bit 0: PAT_table_id_err

This bit indicates table ID not equal to 00

REG - 843Ah – 843Bh I3_PATTABLEDAT: I3 PAT Table Data Register

	15	14	13	12	11	10	9	8	
R/O	PATTABLEDAT[15:8]								
	0	0	0	0	0	0	0	0	
843Ah	7	6	5	4	3	2	1	0	
R/O	PATTABLEDAT[7:0]								
	0	0	0	0	0	0	0	0	

Bits 15 – 0: PATTABLEDAT[15:0]

16-bit PAT Table Data.

This is the data that is stored in the PAT table. It contains information of the PAT version change.

REG - 843Ch – 843Dh I3_PMTTABLEDAT: I3 PMT Table Data Register

	15	14	13	12	11	10	9	8	
R/O	PMTTABLEDAT[15:8]								
	0	0	0	0	0	0	0	0	
843Ch	7	6	5	4	3	2	1	0	
R/O	PMTTABLEDAT[7:0]								
	0	0	0	0	0	0	0	0	

Bits 15 – 0: PMTTABLEDAT[15:0]

16-bit PMT Table Data.

This is the data that is stored in the PMT table. It contains information of the PMT version change.

REG - 8440h I3_TPCONFIG: I3 Trick Play Config Register

R/W

7	6	5	4	3	2	19	0
XmtPsiDis	RolBValid	RolAValid	TpType				TpCmd
0	0	0	0	0	0	0	0

Bit 7: XmtPsiDis

When set to '1,' Trick Play Control does **not** transmit PCR packet to decoder in trick play mode.

Bit 6: RolBValid

When set to '1,' rollover address register (pair B for channel 3) is used to indicate the start point of the sector dropping (for starting trick play), or the stop point of the sector dropping (for stopping trick play).

Bit 5: RolAValid

When set to '1,' rollover address register (pair A for channel 3) is used to indicate the start point of the sector dropping (for starting trick play), or the stop point of the sector dropping (for stopping trick play).

Bits 4 – 1: TpType

This four bits indicate the type of trick play:

0000: I-frame only, fast forward or fast reverse
0001 – 1111: Reserved

Bit 0: TpCmd

Trick Play command.

0: Allows DMA channl3 to stop trick play, and to switch back to normal play
1: Allows DMA channel3 to start trick play

REG - 8441h I3_TPCONFIG_HI: I3 Trick Play Config High Register

R/W

15	14	13	12	11	10	9	8
PSI forcemode	PTS forcemode	XmtVideo-Dis	Reserved				
0	0	0	0	0	0	0	0

Bit 15: PSI forcemode

This bit is for chip debugging only.

When set to '1,' Trick Time Control is enabled to transmit PCR in less than 100ms.

Bit 14: PTS forcemode

For chip debugging only.

When set to '1', Trick Play Control forces PTS/DTS to a certain value.

Bit 13: XmtVideoDis

For chip debugging only.

When set to '1', Trick Play Control disables transmitting video packet.

Bits 12 – 8: Reserved

Not used; read as zeroes.

REG - 8442h – 8443h I3_TPCURRENT: I3 TP Current Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0

8442h	7	6	5	4	3	2	1	0
R/W	Reserved				TPcurrent			
	0	0	0	0	0	0	0	0

Bits 15 – 4: Reserved

Bits 3 – 0: TPcurrent

This is only used for debugging.

The value of this register represents the current state of the trick plat state machine.If a value is written to this register, the state machine will start from that state.

The states are as follows:

4’d00: Idle
4’d01:Fetch packet
4’d02: FIF check
4’d03: Check packet information
4’d04: Reset FIFO write pointer
4’d05: Wait for SMT enable
4’d06: Generate and XMT PSI
4’d07: Generate and XMT PTS packet
4’d08: XMT state
4’d09: Compare continuity counter
4’d10: Send dummy packet
4’d11: Wait to reload PCR
4’d21: XMT discontinuity packet/Reload PCR

REG - 8444h 13_TSTAMPCFG: I3 Timestamp Config Register

R/W	7	6	2	4	3	2	1	0
	Reserved	TSCTRLCURRENT			TPFRAMERATE			
	0	0	0	0	0	0	0	0

Bits 6 – 4: TSCTRLCURRENT (R/O)

Current state (3-bit) of the state machine in timestamp control block. This field is read only, and is not available in ES1.

Bits 3– 0: TPFRAMERATE (R/W)

This 4-bit field controls the frame rate during trick play.

In I-only fast forward mode, only I frames are sent out: This framerate is the number of I frames sent out per second. See Table 3-7 on page 82

Table 3-7: Frame Rate

TPFRAMERATE Value	Actual Framerate
4’h0	5000 frame/sec: for test mode
4’h1	1 frame/sec

Table 3-7: Frame Rate

TPFRAMERATE Value	Actual Framerate
4'h2	2 frame/sec
4'h3	3 frame/sec
4'h4	4 frame/sec
4'h5	5 frame/sec
4'h6	6 frame/sec
4'h7	7 frame/sec
4'h8	8 frame/sec
4'h9	9 frame/sec
4'hA	10 frame/sec
4'hB	11 frame/sec
4'hC	12 frame/sec
4'hD	13 frame/sec
4'hE	14 frame/sec
4'hF	15 frame/sec

REG – 8445h – 8446h I3_PRSSTATUS: I3_Parsers Status Register

R/O	15	14	13	12	11	10	9	8
	PRSSTSEL				Reserved			PRS STATUS [8]
	0	0	0	0	0	0	0	0

8445h R/O	7	6	5	4	3	2	1	0
	PRSSTATUS[7:0]							
	0	0	0	0	0	0	0	0

Bits 15 – 12: PRSSTSEL

Parsers Status Select

Firmware can write to this field to determine which internal parsers (packet parse and video parse block) information to read. Those values are shown in PRSSTATUS, and at the test I/O pins (TESTIO[5:0])

Table 3-8: Parser Information Selection

PRSSTSEL	PRSSTATUS	Test I/O Pins
4'h0	pa1_st_os_A	payl_st_os_A[3:0]
4'h1		payl_st_os_A[8:4]

Table 3-8: Parser Information Selection (Continued)

PRSSTSEL	PRSSTATUS	Test I/O Pins
4'h2	unit_st_os_A	unit_st_os_A[3:0]
4'h3		unit_st_os_A[8:4]
4'h4	broklnk_os_A	broklnk_os_A[3:0]
4'h5		broklnk_os_A[8:4]
4'h6	low_dly_os_A	low_dlly_os_A[3:0]
4'h7		low_dlly_os_A[8:4]
4'h8	pay1_st_os_B	payl_st_os_B[3:0]
4'h9		payl_st_os_B[8:4]
4'hA	unit_st_os_B	unit_st_os_B[3:0]
4'hB		unit_st_os_B[8:4]
4'hC	broklnk_os_B	broklnk_os_B[3:0]
4'hD		broklnk_os_B[8:4]
4'hE	low_dlly_os_B	low_dly_os_B[3:0]
4'hF		low_dly_os_B[8:4]

Bits 8 – 0: PRSSTATUS

Parsers Status

The values are as shown in Table 3-8 on page 83. This register is only for debugging purposes.

3.2 GENERAL REGISTERS

There are five general registers: Base. The base address of Gbase is FF8480.

REG - 8480–8481h DRAMTIMING: DRAM State Machine Timing Control

R/W	15	14	13	12	11	10	9	8	
	Reserved		CASLATENCY[1:0]		DWRTP	DRDTP	TRDC[1:0]]		
	1	1	1	1	1	0	0	1	
8480h		7	6	5	4	3	2	1	0
		TRP	TRAS[2:0]				TRC[3:0]		
		1	1	0	0	0	1	1	1

The default value of this register after power-on reset is F9C7: This is to be consistent with the default value of the Ivory chip.

Note: This register must be set up to conform with the type of SDRAM used on the board, and must match with the same register in the Ivory chip.

Bits 15 – 14: Reserved

Read as zero.

Bits 13 – 12: CASLATENCY[1:0]

CAS latency for a given speed of DRAM.

Bit 11: DWRTP

Write to pre-charge delay.

Bit 10: DRDTP

Read to pre-charge delay.

Bits 9 — 8: TRDC[1:0]

RAS to CAS delay.

Bit 7: TRP

RAS pre-charge delay.

Bits 6 — 4: TRAS[2:0]

Minimum RAS active time.

Bits 3 — 0: TRC[3:0]

Minimum RAS cycle time.

REG - 8482h–8483h CONFIG: Configuration

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0

8482h	7	6	5	4	3	2	1	0
	Reserved				PWRDN	S_WAITNUM		
	0	0	0	0	0	0	0	0

Bits 15–4: Reserved

Firmware programs the bits of this register to tailor the chip to each specific operation mode.

Bit 3: PWRDN

Power down.

This bit is used to activate/deactivate the pull-up terminator of all I/O cells of the chip to test the current leakage of the chip.

Bits 2–0: S_WAITNUM

The number of the wait state when address FE is detected.

Firmware programs this register to let Genii assert the UPWAIT signal a programmable number of clocks, prior to using the wait signal from the internal blocks, or from the Sony chip (Q3221).

As soon as Genii detects MA[23:20] to be 'F', and MA[16] to '0, it asserts UPWAIT for the specific number of clocks programmed in this register. Afterwards, the other blocks determine whether to continue the assertion of UPWAIT, or to release it.

REG - 8484h–8485h INT_ENA: Interrupt Enable

R/W	15	14	13	12	11	10	9	8
								D5_INT_ENA
	0	0	0	0	0	0	0	0

8484h	7	6	5	4	3	2	1	0
	D4_INT_ENA	D3_INT_ENA	D2_INT_ENAa	D1_INT_ENA	D0_INT_ENA	I3_INT_ENA	I2_INT_ENA	I1_ENT_ENA
	0	0	0	0	0	0	0	0

Bits 15 — 9:

Each bit in this register corresponds with the appropriate bit in the INTERRUPT register. If int_enable bit is set, and the corresponding interrupt condition occurs, then Genii sets Interrupt pin to the microprocessor.

Bit 8: D5_INT_ENA

Firmware sets this bit to '1' to allow the D5-interrupt bit in the Interrupt register to set the HINT output pin.

Bit 7: D4_INT_ENA

Firmware sets this bit to '1' to allow the D4-interrupt bit in the Interrupt register to set the HINT output pin.

Bit 6: D3_INT_ENA

Firmware sets this bit to '1' to allow the D3-interrupt bit in the Interrupt register to set the HINT output pin.

Bit 5: D52_INT_ENA

Firmware sets this bit to '1' to allow the D2-interrupt bit in the Interrupt register to set the HINT output pin.

Bit 4: D1_INT_ENA

Firmware sets this bit to '1' to allow the D1-interrupt bit in the Interrupt register to set the HINT output pin.

Bit 3: I3_INT_ENA

Firmware sets this bit to '1' to allow the I1 interrupt bit in the Interrupt register to set the HINT output pin.

Bit 2: I2_INT_ENA

Firmware sets this bit to '1' to allow the I1 interrupt bit in the Interrupt register to set the HINT output pin

Bit 0: I1_INT_ENA

Firmware sets this bit to '1' to allow the I1 interrupt bit in the Interrupt register to set the HINT output pin.

REG - 8486h–8487h INTERRUPT: Interrupt

R/W	15	14	13	12	11	10	9	8
								D5_INT
	0	0	0	0	0	0	0	0

8486h	7	6	5	4	3	2	1	0
	D4_INT	D3_INT	D2_INT	D1_INT	D0_INT	I3_INT	I2_INT	I1_INT
	0	0	0	0	0	0	0	0

Bits 15 — 9:

Each bit in this register, when set by Genii, indicates an internal condition in the chip. Depending on the corresponding bit in INT_ENA register, Genii decides whether or not to interrupt the microprocessor.

To clear the appropriate interrupt condition, firmware writes '1' to the desired bit in this register.

REG - 8488h–8489h D1_ADDRANGE: Device 1 Address Range Register

R/W	15	14	13	12	11	10	9	8
	D1_UPPER_ADDRESS_RANGE							
	0	0	0	0	0	0	0	0

8488h	7	6	5	4	3	2	1	0
	D1_LOWER_ADDRESS_RANGE							
	0	0	0	0	0	0	0	0

This is the address range of the external device that controls the analog in/analog out devices. Each device has an address range. When Genii detects an address within that range, Genii translates the microprocessor timing into the appropriate timing of that device (Motorola type or Intel type, depending on the configuration bit).

Example:

D1_ADDRANGE = 3130h

Any microprocessor address from 31FF to 3000 will be within the Device 1 address range. The timing will be translated to that device timing.

Bits 15–8: D1_UPPER_ADDRESS_RANGE

The upper address range.

Bits 7–0: D1_LOWER_ADDRESS_RANGE

The lower address range.

REG - 848Ah–848Bh D1_CONFIG: Device 1 Configuration Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0

848Ah	7	6	5	4	3	2	1	0
	Reserved				D1_WAIT		D1_MOT	D1_INTEL
	0	0	0	0	0	0	0	0

Bits 15–4: Reserved

Read as zeroes.

Bits 3–2: D1_WAIT

This is the programmable number of wait state an Intel device needs before it looks at the Ready signal.

This register is not needed if the device is in Motorola mode.

Bit 1: D1_MOT

When this bit is set to '1', Device 1 is Motorola type, and Genii will translate the microprocessor timing for Motorola-type timing.

Note: Bit 0 and Bit 1 must not be set simultaneously.

Bit 0: D1_INTEL

When this bit is set to '1', Device 1 is Intel type, and Genii will translate the microprocessor timing for Intel-type timing.

Note: Bit 0 and Bit 1 must not be set simultaneously.

REG - 848Ch–848Dh D2_ADDRANGE: Device 2 Address Range Register

R/W	15	14	13	12	11	10	9	8
	D2_UPPER_ADDRESS_RANGE							
	0	0	0	0	0	0	0	0

8488h	7	6	5	4	3	2	1	0
	D2_LOWER_ADDRESS_RANGE							
	0	0	0	0	0	0	0	0

This is the address range of the external device that controls the analog in/analog out devices. Each device has an address range. When Genii detects an address within that range, Genii translates the microprocessor timing into the appropriate timing of that device (Motorola type or Intel type, depending on the configuration bit).

Example:

D2_ADDRANGE = 3130h

Any microprocessor address from 31FF to 3000 will be within the Device 2 address range. The timing will be translated to that device timing.

Bits 15–8: D2:UPPER_ADDRESS_RANGE

The upper address range.

Bits 7–0: D2_LOWER_ADDRESS_RANGE

The lower address range.

REG - 848Eh–848Fh D2_CONFIG: Device 2 Configuration Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0

848Fh	7	6	5	4	3	2	1	0
	Reserved				D2_WAIT		D2_MOT	D2_INTEL
	0	0	0	0	0	0	0	0

Bits 15–4: Reserved

Read as zeroes.

Bits 3–2: D2_WAIT

This is the programmable number of wait state an Intel device needs before it looks at the Ready signal.

This register is not needed if the device is in Motorola mode.

Bit 1: D2_MOT

When this bit is set to ‘1’, Device 2 is Motorola type, and Genii will translate the microprocessor timing for Motorola-type timing.

Note: Bit 0 and Bit 1 must not be set simultaneously.

Bit 0: D2_INTEL

When this bit is set to ‘1’, Device 2 is Intel type, and Genii will translate the microprocessor timing for Intel-type timing.

Note: Bit 0 and Bit 1 must not be set simultaneously.

REG - 8490h–8491h D3_ADDRANGE: Device 3 Address Range Register

R/W	15	14	13	12	11	10	9	8
	D3_UPPER_ADDRESS_RANGE							
	0	0	0	0	0	0	0	0

8490h	7	6	5	4	3	2	1	0
	D3_LOWER_ADDRESS_RANGE							
	0	0	0	0	0	0	0	0

This is the address range of the external device that controls the analog in/analog out devices. Each device has an address range. When Genii detects an address within that range, Genii translates the microprocessor timing into the appropriate timing of that device (Motorola type or Intel type, depending on the configuration bit).

Example:

D3_ADDRANGE = 3130h
Any microprocessor address from 31FF to 3000 will be within the Device 3 address range. The timing will be translated to that device timing.

Bits 15–8: D3:UPPER_ADDRESS_RANGE

The upper address range.

Bits 7–0: D3_LOWER_ADDRESS_RANGE

The lower address range.

REG - 8492h–8493h D3_CONFIG: Device 3 Configuration Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0

8492h	7	6	5	4	3	2	1	0
	Reserved				D3_WAIT		D3_MOT	D3_INTEL
	0	0	0	0	0	0	0	0

Bits 15–4: Reserved

Read as zeroes.

Bits 3–2: D3_WAIT

This is the programmable number of wait state an Intel device needs before it looks at the Ready signal.

This register is not needed if the device is in Motorola mode.

Bit 1: D3_MOT

When this bit is set to ‘1’, Device 3 is Motorola type, and Genii will translate the microprocessor timing for Motorola-type timing.

Note: Bit 0 and Bit 1 must not be set simultaneously.

Bit 0: D3_INTEL

When this bit is set to ‘1’, Device 3 is Intel type, and Genii will translate the microprocessor timing for Intel-type timing.

Note: Bit 0 and Bit 1 must not be set simultaneously.

REG - 8494h–8495h D4_ADDRANGE: Device D4 Address Range Register

R/W	15	14	13	12	11	10	9	8
	D4_UPPER_ADDRESS_RANGE							
	0	0	0	0	0	0	0	0

8494h	7	6	5	4	3	2	1	0
	D4_LOWER_ADDRESS_RANGE							
	0	0	0	0	0	0	0	0

This is the address range of the external device that controls the analog in/analog out devices. Each device has an address range. When Genii detects an address within that range, Genii translates the microprocessor timing into the appropriate timing of that device (Motorola type or Intel type, depending on the configuration bit).

Example:

D4_ADDRANGE = 3130h

Any microprocessor address from 31FF to 3000 will be within the Device 4 address range. The timing will be translated to that device timing.

Bits 15–8: D4:UPPER_ADDRESS_RANGE

The upper address range.

Bits 7–0: D4_LOWER_ADDRESS_RANGE

The lower address range.

REG - 8496h–8497h D4_CONFIG: Device 4 Configuration Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0

8496h	7	6	5	4	3	2	1	0
	Reserved				D4_WAIT		D4_MOT	D4_INTEL
	0	0	0	0	0	0	0	0

Bits 15–4: Reserved

Read as zeroes.

Bits 3–2: D4_WAIT

This is the programmable number of wait state an Intel device needs before it looks at the Ready signal.

This register is not needed if the device is in Motorola mode.

Bit 1: D4_MOT

When this bit is set to '1', Device 4 is Motorola type, and Genii will translate the microprocessor timing for Motorola-type timing.

Note: Bit 0 and Bit 1 must not be set simultaneously.

Bit 0: D4_INTEL

When this bit is set to '1', Device 4 is Intel type, and Genii will translate the microprocessor timing for Intel-type timing.

Note: Bit 0 and Bit 1 must not be set simultaneously.

REG - 8498h–8499h D5_ADDRANGE: Device 5 Address Range Register

R/W	15	14	13	12	11	10	9	8
	D5_UPPER_ADDRESS_RANGE							
	0	0	0	0	0	0	0	0

8498h	7	6	5	4	3	2	1	0
	D5_LOWER_ADDRESS_RANGE							
	0	0	0	0	0	0	0	0

This is the address range of the external device that controls the analog in/analog out devices. Each device has an address range. When Genii detects an address within that range, Genii translates the microprocessor timing into the appropriate timing of that device (Motorola type or Intel type, depending on the configuration bit).

Example:

D5_ADDRANGE = 3130h
Any microprocessor address from 31FF to 3000 will be within the Device 5 address range. The timing will be translated to that device timing.

Bits 15–8: D5:UPPER_ADDRESS_RANGE

The upper address range.

Bits 7–0: D5_LOWER_ADDRESS_RANGE

The lower address range.

REG - 849Ah–849Bh D5_CONFIG: Device 5 Configuration Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0

849Ah	7	6	5	4	3	2	1	0
	Reserved				D5_WAIT		D5_MOT	D5_INTEL
	0	0	0	0	0	0	0	0

Bits 15–4: Reserved

Read as zeroes.

Bits 3–2: D5_WAIT

This is the programmable number of wait state an Intel device needs before it looks at the Ready signal.

This register is not needed if the device is in Motorola mode.

Bit 1: D5_MOT

When this bit is set to ‘1’, Device 5 is Motorola type, and Genii will translate the microprocessor timing for Motorola-type timing.

Note: Bit 0 and Bit 1 must not be set simultaneously.

Bit 0: D5_INTEL

When this bit is set to ‘1’, Device 5 is Intel type, and Genii will translate the microprocessor timing for Intel-type timing.

Note: Bit 0 and Bit 1 must not be set simultaneously.

REG - 849Eh–849Fh APLL_REG1: APLL 1 Register

R/W	15	14	13	12	11	10	9	8
	CLKMUX	UNLOCK (read only)	MSEL	PLL_TEST	MODE[1:0]		STBY	RSTB_REG

849Eh	7	6	5	4	3	2	1	0
	NSEL[4:0					PSEL[2:0]		
	0	0	0	0	0	0	0	0

Bit 15: CLKMUX

0: Crystal (40MHz)
1: PLL

Bit 14: UNLOCK

This bit is read-only.

0: APLL Frequency is stable.

1: APLL Frequency is not stable.

Bit 13: MSEL

This bit is used to select the M divider value.

0: M = 1
1: M = 3 (default, MSEL = 1)

Note: If an invalid P value or N value is written, the MSEL is 1.

Bit 12:PLL_TEST

The bit enables the PLL_freq test mode: Reserved for testing at TI.

0: APLL in normal mode: 40MHz crystal drives lock control.
1: APLL in test mode: PLL drives lock control

Bits 10–11: MODE[1:0]

Mode selection bits.

00: Normal operation mode.

For more information, see THEORY OF OPERATIONS on page 100.

Bit 9: STBY

This bit selects the STandby mode: '1' places the APL1 in Standby mode. The Standby mode outputs a zero COUT (APLL clock output) to save power.

CAUTION: Do not use the Standby mode unless CPU is running off 40MHz crystal (divide by 2) with CLKMUX (bit 15), as it will shut down the CPU core and will require toggling power to restart.

Bit 8: RSTB_REG

0: Operating mode
1: Rest the APLL

Bits 7–3: NSEL[4:0]

These bits select the value for the "N" divider. The valid ranges are N=1h to N=1Fh. N=1h will give an N divide value of 1. N=2h will give an N divide value of 2, and so on.

Note: If an invalid N value is written, the default value of N=10 will be used.
For Genii:
The N value is not rated to exceed 20. The PLL will not function properly if internal frequencies exceed 250MHz. To prevent this from happening, use the following restriction for M/N/P combinations. CIN x N/M, 250MHz.

CAUTION: Only adjust this register if CPU is running on a 40 MHz crystal. PLL dividers can cause glitches in the clock (confuses circuitry).

Bits 2–0: PSEL[2:0]

These bits select the value for the "P" divider. The valid ranges are P=0h to P=7h. P=0h will give a P divide value of 1. P=1h will give an P divide value of 2, and so on.

Note: If an invalid P value is written, the default value of P=2 will be used.

REG - 84A0–84A1h APLL_REG2: APLL 2 Register

R/W	15	14	13	12	11	10	9	8
	PASSWORD[7:0]							
8A0Fh	7	6	5	4	3	2	1	0
	Reserved		TST1_SEL[2:0]			TST2_SEL[2:0]		
	0	0	0	0	0	0	0	0

Bits 15–8: PASSWORD[7:0

PASSWORD must be written with the correct password before changing any of the APLL registers. The next write operation to any of the APLL registers disables the register write enable.

Note: The password must be re-written before each new APLL register write.
The password is 8'hAB.

Bits 7-6: Reserved

Read as zeros.

Bits 5-3: TST1_SEL[2:0]

These bits determine the output of the APLLTST1 pin.

000: msel
001: mode0
010: stby
011: test
100: test
101: unlock
110: reset_aplin
111: mode1

Bits 2-0: TST_SEL[2:0]

These bits determine the output of the APLLTST2 pin.

000: msel
001: mode0
010: stby
011: test
100: test
101: unlock
110: reset_aplin
111: mode1

REG 84A2h–84A3h TEST_MUXSEL1: Test Mux Select 1 Register

R/W	15	14	13	12	11	10	9	8
	Reserved	TRICK_PLAY2_TEST_SELECT			TRICK_PLAY1_TEST_SELECT			DMA_CH3_TEST_SELECT
	0	0	0	0	0	0	0	0

84A2h	7	6	5	4	3	2	1	0
	DMA_CH3_TEST_SELECT		DMA_CH2_TEST_SELECT			DMA_CH1_TEST_SELECT		
	0	0	0	0	0	0	0	0

Bit 15: Reserved

Read as zero.

Bits 14–12: TRICK_PLAY2_TEST_SELECT

Bits 11–9: TRICK_PLAY1_TEST_SELECT

Bits 8–6: DMA_CH3_TEST_SELECT

Bits 5–3: DMA_CH2_TEST_SELECT

Bits 0–2: DMA_CH1_TEST_SELECT

REG 84A4h–84A5h TEST_MUXSEL2: Test Mux Select 12 Register

R/W	15	14	13	12	11	10	9	8
	GENERAL_TEST_SELECT							EXT_HOST_TRANS_TEST_SEL
	0	0	0	0	0	0	0	0

84A4h	7	6	5	4	3	2	1	0
	EXT_HOST_TRANS_TEST_SEL		Q3221_TRANS_TEST_SEL			BUFFER_CONTROL_TEST_SEL		
	0	0	0	0	0	0	0	0

Bits 9–15: GENERAL_TEST_SELECT

Bits 8–6: EXT_HOST_TRANS_TEST_SEL

BITS 3–5: Q3221_TRANS_TEST_SEL

Bits 2–0: BUFFER_CONTROL_TEST_SEL

REG 84A6h–84A7h TEST_MUXSEL3: Test Mux Select 3 Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0
84A6h	7	6	5	4	3	2	1	0
	Reserved		TEST_IO_OUTPUT_ENA					
	0	0	0	0	0	0	0	0

Bits 6–15: Reserved

Read as zeroes.

Bits 0–5: TEST_IO_OUTPUT_ENA

REG 84A8h – 84A9h QCONFIG: Q2DMA Config Register

R/W	15	14	13	12	11	10	9	8
	Reserved							
	0	0	0	0	0	0	0	0
8498h	7	6	5	4	3	2	1	0
	Reserved						CAS_LATENCY[1:0]	
	0	0	0	0	0	0	0	0

Bits 15 – 2: Reserved

Read as zeroes.

Bits 1 – 0: CAS_LATENCY[1:0]

These bits must be programmed the same as for a SONY 3221Q chip.

REG 84AAh – 84ABh QSTATUS: Q2DMA Status Register

R/W	15	14	13	12	11	10	9	8
	Reserved			QERR_ CLR	QSTATUS[3:0]			
	0	0	0	0	0	0	0	0
84ABh	7	6	5	4	3	2	1	0
	QSTATUS2[3:0]				QSTATUS1[3:0]			
	0	0	0	0	0	0	0	0

Bits 15 – 13: Reserved

Read as zeroes

Bit 14: QERR_CLR

When set to ‘1’, this bit clears all errors in the Q2DMA FIFOs. This bit does not clear itself: Reset to ‘0’ after clearing errors.

Bits 11 – 8: QSTATUS3[3:0]

FIFO status for channel3 Q2 DMA FIFO:

Bit 11: AVside over-run
Bit 10: AVside under-run
Bit 9: Qside over-run
Bit 8: Q-side under-run

Bits 7 – 4: QSTATUS2[3:0]

FIFO status for channel2 Q2 DMA FIFO:

Bit 7: AVside over-run
Bit 6: AVside under-run
Bit 5: Qside over-run
Bit 3: Q-side under-run

Bits 3 – 0: QSTATUS1[3:0]

FIFO status for channel1 Q2DMA FIFO:

Bit 3: AVside over-run
Bit 2: AVside under-run
Bit 1: Qside over-run
Bit 0: Q-side under-run

REG - 849Ch CLOCKSELLO Register

R/W

7	6	5	4	3	2	1	0
APLLSEL[1:0]		AV3CLKSEL[1:0]		AV2CLKSEL[1:0]		AV1CKSEL[1:0]	
0	0	0	0	0	0	0	0

Bits 7–6: APLLSEL[1:0]

These bits are used to select the frequency of buffer clock:

00: TSCLK (27MHz)
01: APLL (100 MHz)
10: REFCLK (40MHz)
11: REFCLK (40MHz)

Bits 5–4: AV3CLKSEL[1:0]

These bits are used to select the frequency of AV channel 3 clock input.

00: AV3CLK (system dependent)
01: AV3CLK (system dependent)
10: 20MHz
11: TSCLK (27MHz)

Bits 3–2: AV2CLKSEL[1:0]

These bits are used to select the frequency of AV channel 2 clock input.

00: AV2CLK (system dependent)
01: AV2CLK (system dependent)
10: 20MHz
11: TSCLK (27MHz)

Bits 1–0: AV1CKSEL[1:0]

These bits, AV1CKSEL[1] and AV1CKSEL[0], are used to select the frequency of AV channel 1 clock input.

00: AV1CLK (system dependent)
01: AV1CLK (system dependent)
10: 20MHz
11: TSCLK (27 MHz)

REG - 849Dh SOFTRESET Register

R/W

7	6	5	4	3	2	1	0
Reserved				GENII_ SFTRSTN	DMA3_ SFTRSTN	DMA2_ SFTRSTN	DMA1_ SFTRSTN
0	0	0	0	0	0	0	0

Bits 7 – 4: Reserved

Read as zeroes.

Bit 3: GENII_SFTRSTN

Writing '1' to this bit resets the GENII chip. This bit clears itself.

Bit 2: DMA3_SFTRSTN

Writing '1' to this bit resets dma_channel 3. This bit clears itself.

Bit 1: DMA2_SFTRSTN

Writing '1' to this bit resets dma_channel 2. This bit clears itself.

Bit 0: DMA1_SFTRSTN

Writing '1' to this bit resets dma_channel 1. This bit clears itself.

REG - 849Eh – 84AF: Reserved

REG - 84B0h VERSION: Version Register

R/O

7	6	5	4	3	2	1	0
Reserved			VERSION[4:0]				
0	0	0	0	0	0	0	0

Bits 7– 0: Reserved

Read as zeroes.

Bits 4 – 0: VERSION[4:0]

These bits are used to show the version number of the Genii chip. This register is read-only.

4.1 OPERATIONS

4.1.1 Packet Parse Block

The major function of the packet parse block is to analyze the information of the MPEG-2 transport stream packets that are received by the AVFIFO. This packet information is also used by video parse block, trick play control block and table generation block to implement trick play. Additionally, PCR information is delivered to the timestamp counter. This information enables the counter to update the PCR value, when necessary.

After the packet parse block analyzes a packet, information is passed to other blocks. See Table 4-1:

Table 4-1: Packet Parse Block

Packet Information	Description
PAT_pkt	'1' = current packet is a PAT packet
PMT_pkt	'1' = current packet is a PMT packet
video_pkt	'1' = current packet is a video packet
PCR_pkt	If the current packet is a PCR packet. Note that if a video packet contains PCR information, this flag will not be set. This flag is only set to 'high' when the current packet contains purely PCR information.
null_pkt	If the current packet is a null packet
contcnt[3:0]	Continuity counter
adpt_f_ctrl[1:0]	Adaptation field control
discontbit	Discontinuity bit
PCR_flag	If the current packet contains PCR information
PES_hdr_flag	'1' = PES header is present: PES header flag
PES_hdr_os[8:0]	PES header offset. This indicates the location of the PES header in the AVFIFO
stream_id[7:0]	8-bit stream ID
PTS_DTS_flag[1:0]	Bit 0 indicates the presence of DTS in the PES header Bit 1 indicates the presence of PTS in the PES header

4.1.2 Video Parse Block

The major function of the video parse block is to analyze the video information of the MPEG-2 transport stream: It only analyzes the packet containing video data. The video information is further used by trick play control block, and table generation block to implement trick play.

Table 4-2: shows the information generated by video parse block:

Table 4-2: Video Parse Block

Information	Description
payl_st_os[8:0]	This is the location of the start of the payload in the AVFIFO
unit_st_os[8:0]	This is the location of the start of a video unit in the AVFIFO. Note that the video unit start can be a frame start, a GOP start or a sequence start. Depends on the situation, the picture start code of a B-frame or P-frame is recognized as an unit start definitely. For I-frame, if sequence header or GOP header is present before the I-frame picture start code, then that header is counted as the start of the I-frame unit. In other words, if sequence header is present, unit_st_os = sequence header offset. If there is no sequence header, but GOP header is present, then unit_st_os = GOP header offset. If there is no sequence header and no GOP header is present, then unit_st_os = picture state offset.
broklnk_os[8:0]	This is the location of the broken link bit in the AVFIFO.
low_dly_os[8:0]	This is the location of the low delay bit in the AVFIFO
os_valid[3:0]	This is the offset valid which indicates if the above 4 offsets are valid.
frame_type[2:0]	This is the frame type. 3'b1 = I-frame 3'b2 = P-frame 3'b3 = B-frame If a video unit starts in the middle of the packet, the frame type will correspond to the frame type of that video unit. For instance, if within a packet, a B-frame ends and an I-frame starts after that, frame type will be I-frame.

4.1.3 Table Generation Block

The functions of the table generation block are generating the I-frame, PAT and PMT information. These information may be needed during playback, especially during trick play.

4.1.3.1 I-frame Table

In I-only fast forward, only I frames are displayed. Since the other frames are skipped, the buffer does not have enough time to read all the frames from the hard disk. Only the sectors containing I-frame should be read by the buffer. To help the firmware to locate where the I frames are, the following information (Table 4-3:) is generated during recording, and is saved on the hard disk with the video data.

Table 4-3: I-frame Table

(bit 31)			Start LBA (32-bit) (bit 0)
I-int	PAT-int	PMT-int	Sector Count (29-bit)

Start LBA is the location of where the I-frame starts. For I-frame table, I-int is be set to '1,' while PAT-int and PMT-int are set to '0'. Sector count represents the number of sectors that the I-frame occupies. This lets the firmware know the size of the I-frame, so that the location of where the I-frame ends can be calculated.

Firmware can read the data in the table by reading FIFODATA register. Note that FIFOSEL in CONFIG2 register must be set to 2'b00 to read I-frame table. When the firmware reads the table, the lower word of the quadlet is be read first, then the upper word. Every time the microprocessor reads a word, FIFOADD register is automatically incremented by 1.

When one data quadlet is written into the FIFO, the Itable_wrptr (in FIFOADD register) is incremented by 1.

4.1.3.1 PAT and PMT Tables

The PAT and PMT tables are used to store the location of the video stream whenever version change occurs in the PAT or PMT packets. During video playback, some of the frames may be skipped, and PAT or PMT version change may occur anywhere in the video stream. If those packets contain information about version change are also skipped, problems will occur because the video decoder will not be able to figure out the value of the video PID, and other information. This makes it necessary to keep track of the location of the version change, and to send out those PAT and PMT packets when needed.

Following is the format of the data in the PAT and PMT tables.

Table 4-4: PAT and PMT Tables

(bit 31)			Start LBA (32-bit)		(bit 0)
I-int	PAT-int	PMT-int	Packet Count (17-bit)	Sector Count (12-bit)	

Start LBA is the location of where the PAT or PMT starts. For PAT table, PAT-int is set to ‘1,’ while I-int and PMT-int are set to ‘0’. For PMT table, PMT-int will be set to ‘1’ while I-int and PAT-int are set to ‘0’. Sector count represents the number of sectors that the PAT or PMT occupies. Packet count represents the number of packets used by that PAT or PMT. This lets the firmware know the size of the PAT or PMT.

Again, the data in the table is in quadlet format. When the firmware reads it, the lower word will be read first and then followed by the upper word. Whenever a word is read from the FIFO, the PAT_rdptr or the PMT_rdptr is incremented by 1.

These read pointers are in the FIFOADD register when the FIFOSEL is set to 2’b00.

4.1.4 Timestamp Control Block

The function of the timestamp control block is to control the timing pattern of the video stream during trick play. Also, it provides the new PTS, DTS and PCR values to the trick play control block, which allows the proper values to be placed on the video stream.

In I-only fast forward mode, only I frames are displayed. The frame rate of these I frames is determined by TPFRAMERATE in TSTAMPCFG register. See the TSTAMPCFG register description, page 82, for details for programming this register.

Figure 4-1: Timestamp TIming, Start of Trick Play

Figure 4-1: shows how the timestamp control block controls the timing when trick play starts. Before the trick play starts, nothing is modified. The timestamp control block catches the timestamp of the packet that contains the start of I-

frame. From TPFRAMERATE, the interval between each I-frame during trick play can be calculated. In this example, the interval is 300.

When trick play starts:

- new timestamp = previous timestamp + interval
- new PTS = new timestamp + interval

The first I-frame has a timestamp equal to 1000; the new timestamp is equal to 1000+300 = 1300. New PTS is equal to 1300 + 300 = 1600. Note that the new PTS applies to the packet containing the start of an I-frame unit, while the new timestamp applies to the first packet after the I-frame start. The timing pattern shows that all the packets are sent out in burst mode until the I-frame start packet is also sent out. The system stops and waits until the PCR counter increments to the value of the new timestamp. The 2nd I-frame packet is then sent out, then packets are again transmitted in burst mode.

To switch from trick play back to normal play mode, the PCR base has to be updated. In the figure, the time base is switched from 2200 to 4000. The time of switching (i.e. 2200) is determined by the PTS of the last I-frame before trick play ends. It is necessary to make sure that the previous I-frame has been displayed before restoring to the normal mode. Otherwise, the next I-frame may overflow the decoder buffer. The new time base (i.e. 4000) is equal to the timestamp (4100) of the first packet after trick play ends, subtracted by a certain offset (100 in this example). In Genii chip, this offset is hardwired to 33'b4, which gives 44.4 us.

4.1.5 PAT and PMT Parsers

The theory of operation of PAT and PMT parsers follows.

4.1.5.1 PAT Parser

The PAT parser block looks for MPEG-2 transport packets with PID value of 0x0000, then extracts the program map table PIDs and their associated program numbers. For the parsing to start, the '*payload_unit_start_indicator'* bit contained in the header must be set. Once the first three PMT PIDs are found, or the end of the table is encountered, parsing will stop. Parsing errors are detected and flagged to the other Genii blocks that use this information.

These errors include:

- Continuity counter error: indicates that a long table (spanning more than one TS packet) was described by non-adjacent packets.
- Adaptation length error: indicates that the adaptation field length field exceeded the maximum allowed value.
- Crc error: indicates that a crc error was detected.
- Pointer error: indicates that the pointer field value exceeded the maximum allowed value.
- Table ID error: indicates that the table ID field value was not 0x00.

When a new table version becomes effective, the condition is detected and flagged to the appropriate Genii blocks. An interrupt is then generated along with an index table entry indicating the first LSA containing the new version of the table.

The internal state machine may be forced into any state by firmware control. The current state of the PAT parsers' FSM is readable through the same register.

4.1.5.2 PMT Parser

The PMT parser block waits for a PAT to have been parsed error-free, and for the information contained in it to be current, before looking for and parsing PMTs. In acquisition mode, each PMT is parsed in the order received. Upon successfully detecting both PCR_PID and Video_PID, the program number is locked and, subsequently, only the PMT associated with that program is parsed.

In the case that the program is no longer available, the PMT parser will automatically re-acquire. This will be revised in ES2.

As in the PAT parser, PMT parsing can only start on a packet where the '*payload_unit_start_indicator'* bit, contained in the header, is set. Parsing errors are detected and flagged to the other Genii blocks that use the

parsed information. In addition to the errors mentioned above, '*section error'* is also flagged for the PMT parser. This indicates either '*section_number'* or '*next_section_number'* are non-zero, a condition that is not allowed as there can only be one PMT section per PMT table.

When a new table version becomes effective, the condition is detected and flagged to the appropriate Genii blocks. An interrupt is then generated along with an index table entry indicating the first LSA containing the new version of the table.

CRC error detection block for each parser is under firmware control as the function can be enabled or disabled by programmable register bits.

The results of the parsers may also be overridden, and the PID values may be programmed in directly by firmware.

4.1.6 Interface with Disk Controller chip

HREQ: Host Request (output – active hi)
Request to Disk Controller chip to access DRAM. Stays active as long as Genii has data to transfer to/from DRAM.

HACK: Host Acknowledge (input – active hi)
Acknowledge from Disk Controller chip that Genii has the right to access DRAM. This input should be used together with STOPCKN to determine when Genii has the right to drive DRAM bus.

STOPCKN: Stopped driving DRAM Clock (bidirectional – active lo)
This pin is used as input when Genii is requesting the bus, i.e. HREQ is getting active, waiting for HACK to be active. In this mode, Genii should wait for both HACK and STOPCKN active before it can drive DRAM bus.

This pin is output when Disk Controller chip wants to regain control of DRAM bus. In this mode, Disk Controller chip will need to de-assert HACK, then wait for Genii to de-assert STOPCKN before it can drive DRAM bus.

When STOPCKN is active, all DRAM control and data bus should be tristated.

The following table describes each case of HACK and STOPCKN combination:

Table 4-5: Truth Table: HACK, STOPCKN

HACK	STOPCKN	Description
0	X	Disk Controller is using DRAM Genii should wait for its turn. Disk Controller wants to regain control of DRAM. If Genii is accessing DRAM, Genii should release DRAM bus as soon as possible.
1	0	Genii is free to drive DRAM bus.
1	1	Disk Controller gives Genii permission to access DRAM, but Genii should wait for STOPCKN to be active before it can actually drive the DRAM bus.

4.1.7 Interface With Dram

Genii can only drive I/O pins to DRAM when it's given permission to from Disk Controller chip according to HACK and STOPCKN truth table above.

Note: The timing of these signals must be compliant to Sync DRAM spec.

SDRAMCLK: Sync DRAM clock (output)
Maximum frequency for this clock is 100Mhz.

CSN: Chip Select (output – active lo)

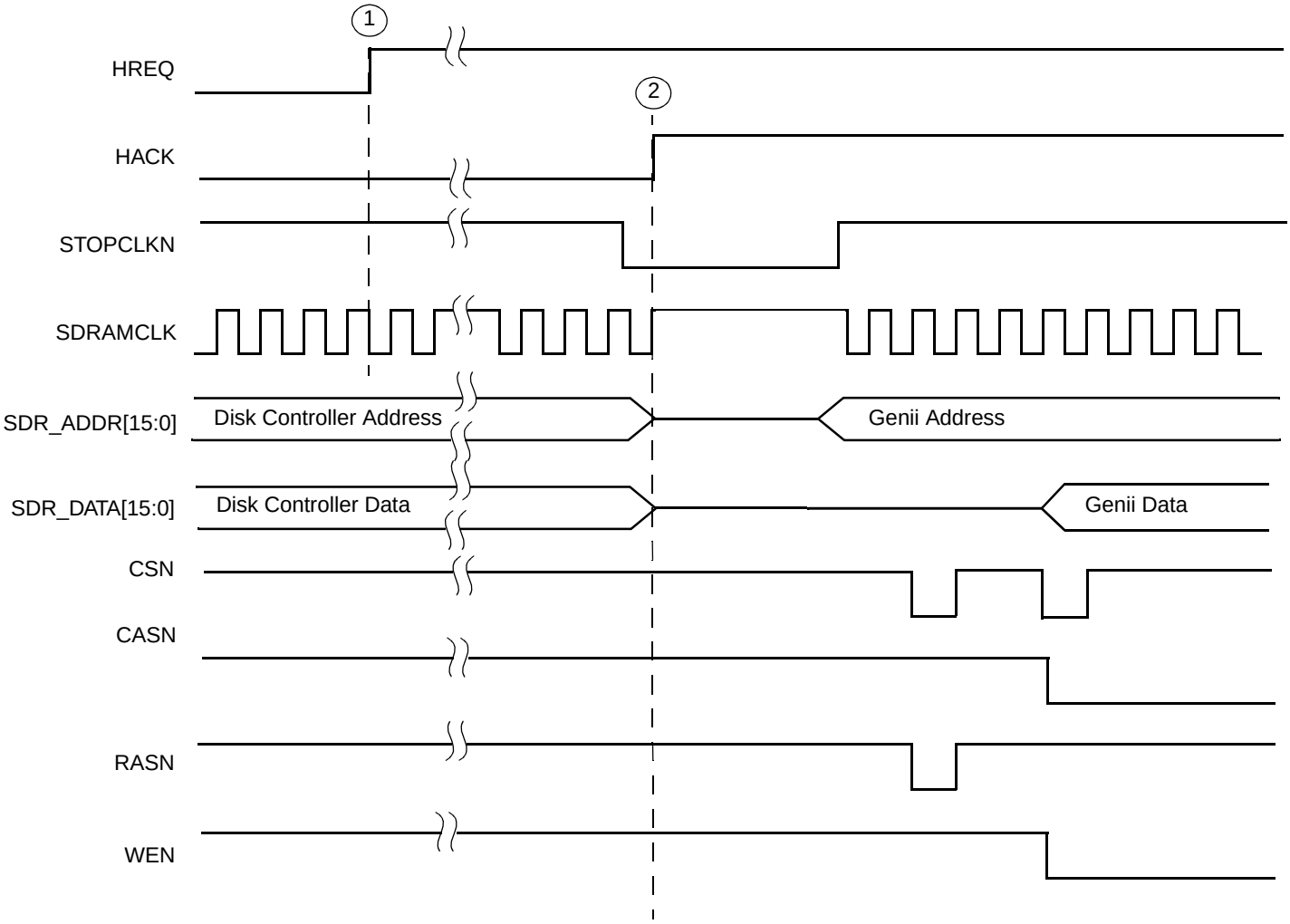
RASN: Row Address Select (output – active lo)

CASN: Column Address Select (output – active lo)

WEN: Write Enable (output – active lo)
When this pin is asserted, Genii outputs data on SDR_DATA bus.

SDR_ADDR[15:0]: Sync DRAM address bus (output)

SDR_DATA[15:0]: Sync DRAM data but (bi-directional)
This bus is input to Genii when WEN is not active, output from Genii when WEN is active.



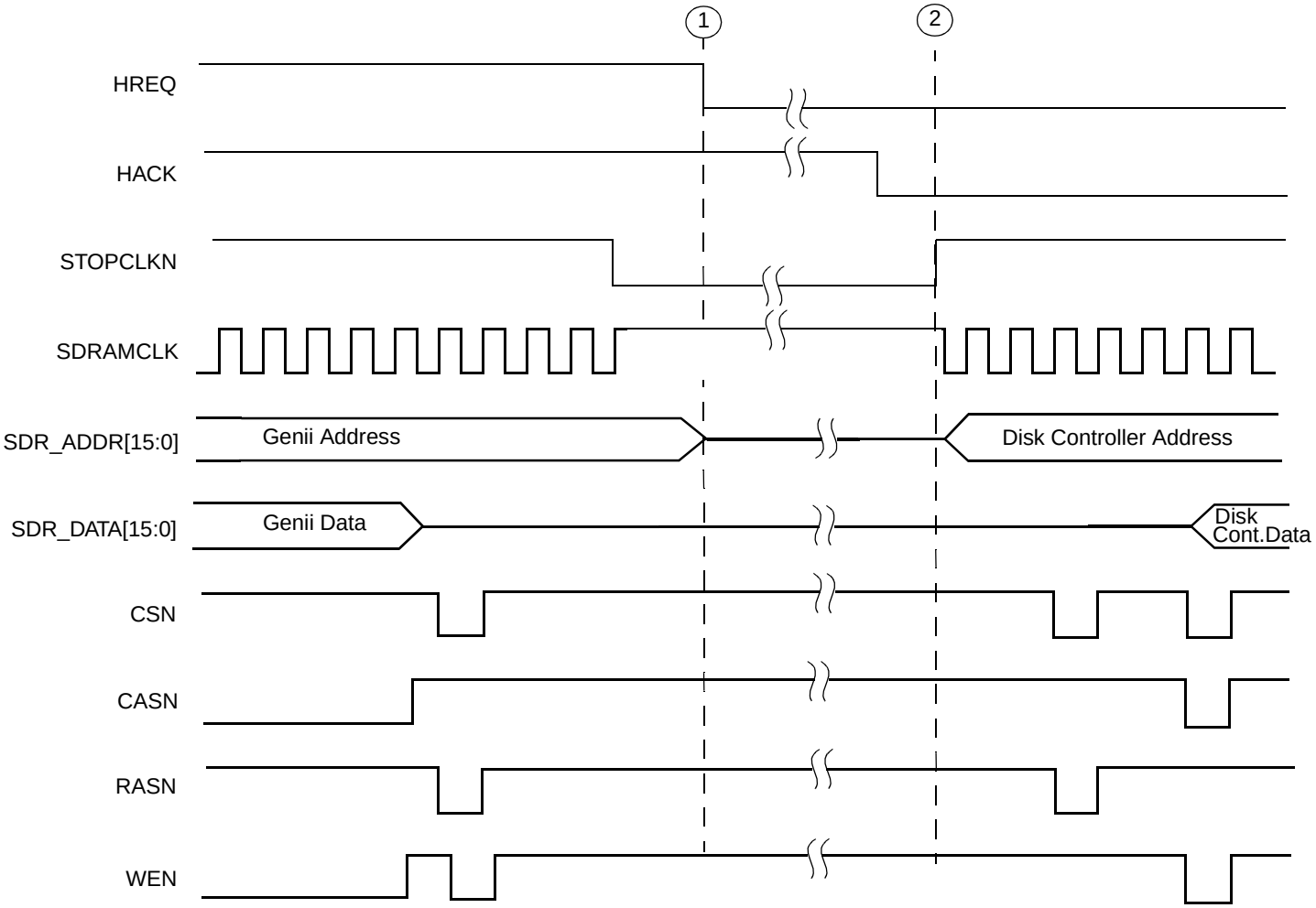
(1) Genii asserts HREQ to request the right to access DRAM from the Disc Controller chip.

(2) Disc Controller chip translates all control pins to SDRAM, then asserts HACK and STOPCKN pins
Genii is granted control of SDRAM only after Genii detects the assertion of both HACK and STOPCKN pins.
After HACK and STOPCKN are asserted, Genii can: a) Assert STOPCKN to signal that it is now driving the SDRAM bus; b) Access SDRAM.

(3) Disc Controller chip must tristate all DRAM control pins before asserting HACK and STOPCLKN pins.

H. Nguyen 3/19/99

Figure 4-2: Genii Access Request to DRAM Timing

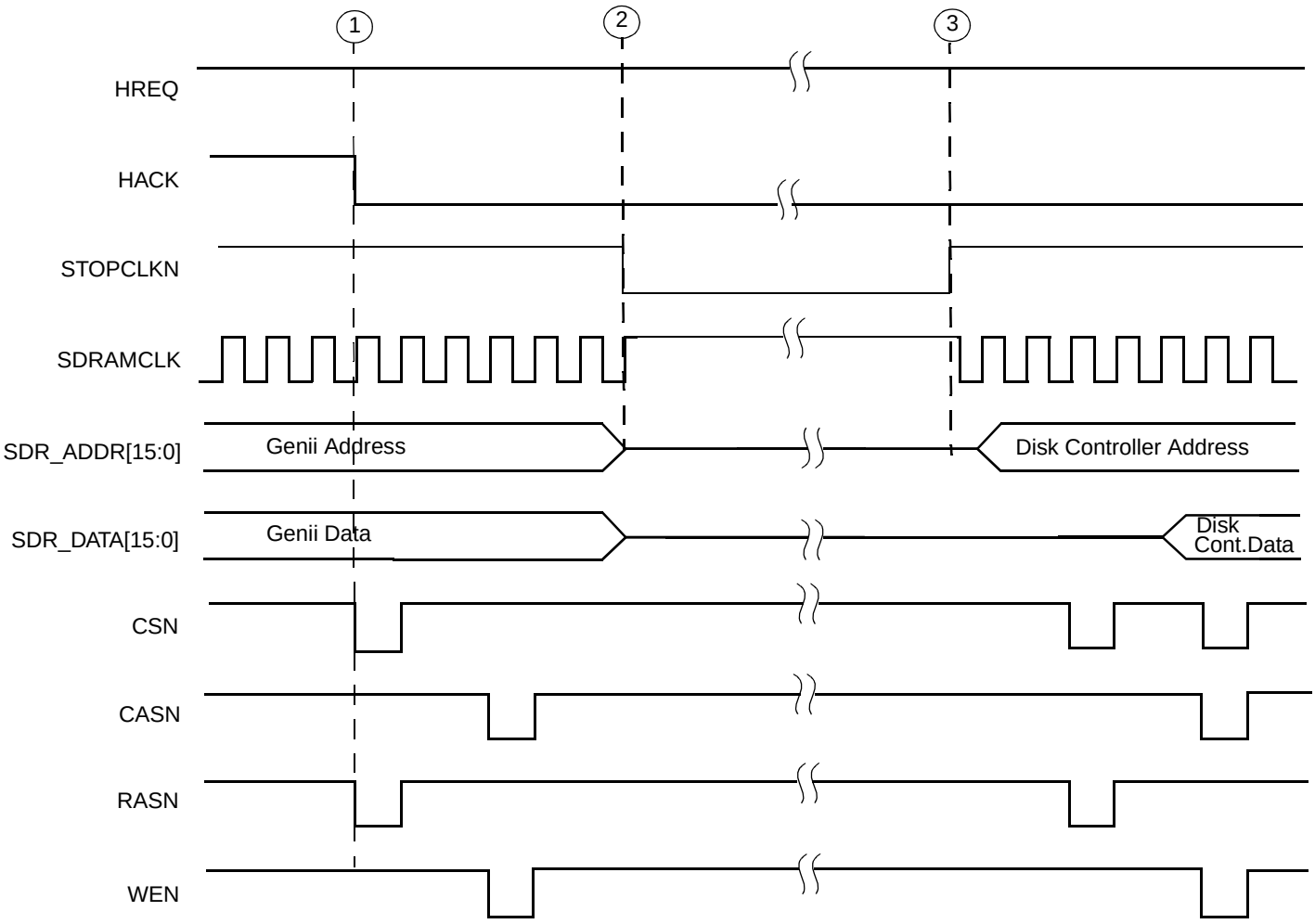


(1) After finishing the DRAM access, Genii: a) puts SDRAMCK in high level, then tristate; b) puts STOPCKN in low level then tristate; c) puts all other DRAM control pins in inactive state, then tristates them; and d) drives HREQ low.

(2) Upon detecting HREQ low, the Disk Controller drives HACK low. The Disk Controller can start DRAM access as soon as it detects HREQ low and STOPCKN low.

H. Nguyen 3/19/99

Figure 4-3: Genii Gives Up SDRAM Bus



(1) While Genii is controlling the SDRAM, the Disk Controller chip pre-empts it by driving HACK low.

(2) Upon detection of HACK low, Genii finishes the current transfer (if any), then: a) puts SDRAMCK in high level, then tristates it; b) puts STOPCKN in low level, then tristates it; and c) puts all other DRAM control pins in the inactive state, then tristates them.

(3) The Disk Controller chip can only drive the DRAM bus when it detects a STOPCKN low. HREQ can stay high as long as Genii has data to transfer to the SDRAM. The Disk Controller will come back to acknowledge it when it is done accessing SDRAM.

H. Nguyen 3/19/99

Figure 4-4: Disk Controller Pre-empts Genii to Regain Control of SDRAM

4.2 UP INTERFACE I/O DESCRIPTION

There are 2 different modes of interfacing with the microprocessor in Genii.

1) µP was integrated in the Disk Controller chip, and Genii interfaces with µP via Disk Controller chip. This is also called internal µP mode.

2) µP is a separate chip on the board, and it controls both Genii and Disk Controller chip through its address/data/control lines. This is alos called external µP mode, or emulation mode.

In both internal and external µP mode, the I/O pins are the same, except for pins MA[23:16]. These pins exist only in the External µP mode. In the Internal µP mode MA[23:16] are replaced by the AVSEL pin (the Disk Controller chip does the decoding of the upper address bits internally and sends the results (AVSEL On/Off) over the AVSEL select line.

4.2.1 I/O Description

MA[23:16]: Microprocessor Address bus. (input). These pins are only valid in External mP mode.

MAD[15:0]: Microprocessor Address/Data bus. (Bi-directional)

UPASTB: Microprocessor Address STrobe (input — active high). When UPASTB is active, the address of the register mP wants to access is present of MA/MAD bus. This address should be valid with appropriate setup and hold time on the falling edge of UPASTB, and Genii uses the falling edge of UPASTB to latch in the address.

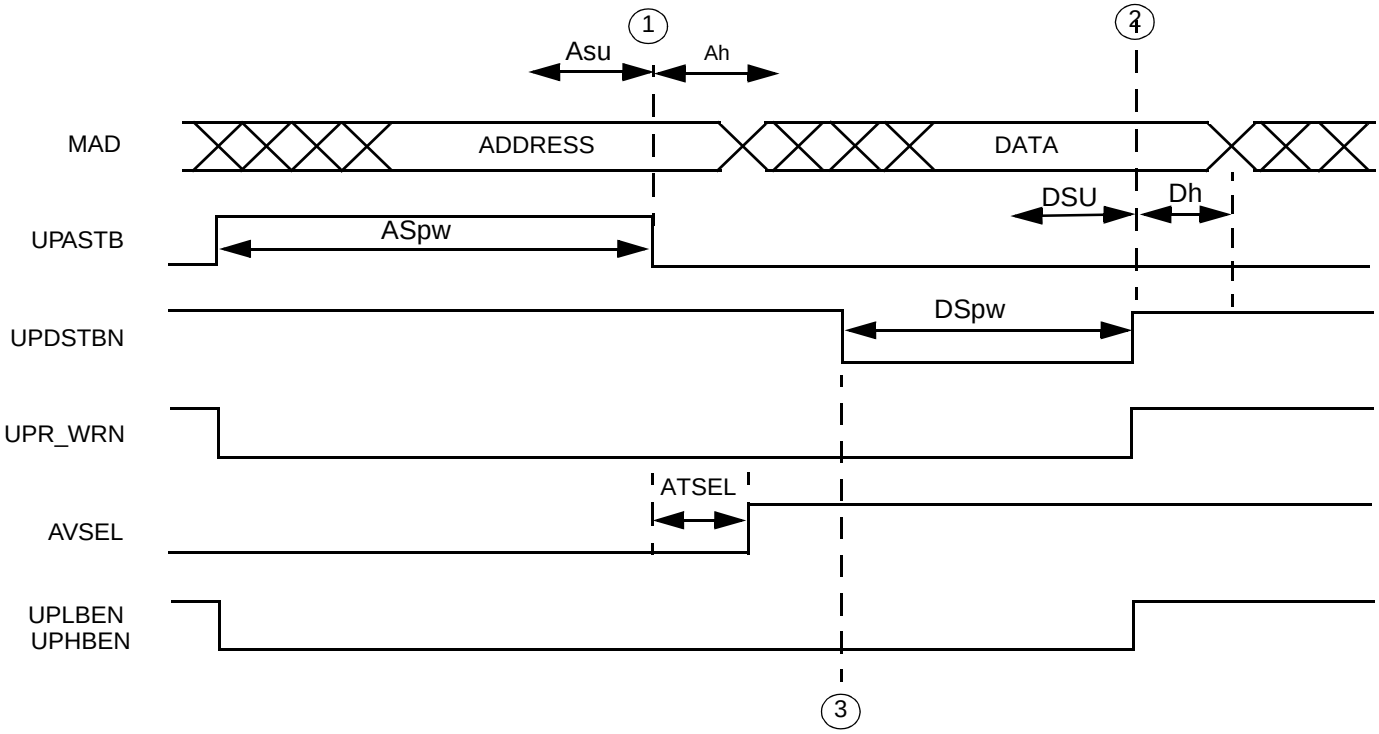
UPDSTBN: Microprocessor Data Strobe (input — active low). When mP tries to write to Genii registers (UPR4_WN = 0), data it puts on MAD bus should be valid with appropriate setup and hold on rising edge of UPDSTBN. Genii uses rising edge of UPDSTBN to latch in data on MAD bus. When mP tries to read from Genii registers (UPR_WRN = 1), Genii should have data valid on MAD bus as soon as ti detects UPDSTBN low, mP will sample data on MAD bus at rising edge of UPDSTBN. To delay the assertion of UPDSTBN, Genii may assert IFWAITN pin to but more time for getting data ready on MAD bus. UPDSTBN will stay low as long as IFWAITN stays low.

IFWAIGN: Interface Wait (output – active low). Genii asserts this pin to get more time on µP register read. As long as this pin stays low, µP will keep UPDSTBN low, so Genii has time to get data setup on MAD BUS.

UPR_WRN: Microprocessor Low Byte write Enable (input — active low).

UPLBEN: Microprocessor Hi Byte write Enable (input — active high).

AVSEL: Select line for Genii chip (input — active high). This pin is only valid in internal µP mode. Used for internal µP mode only. Disk controller chip will decode the upper bits of the address bus, and assert AVSEL pin if the address is in Genii address range.



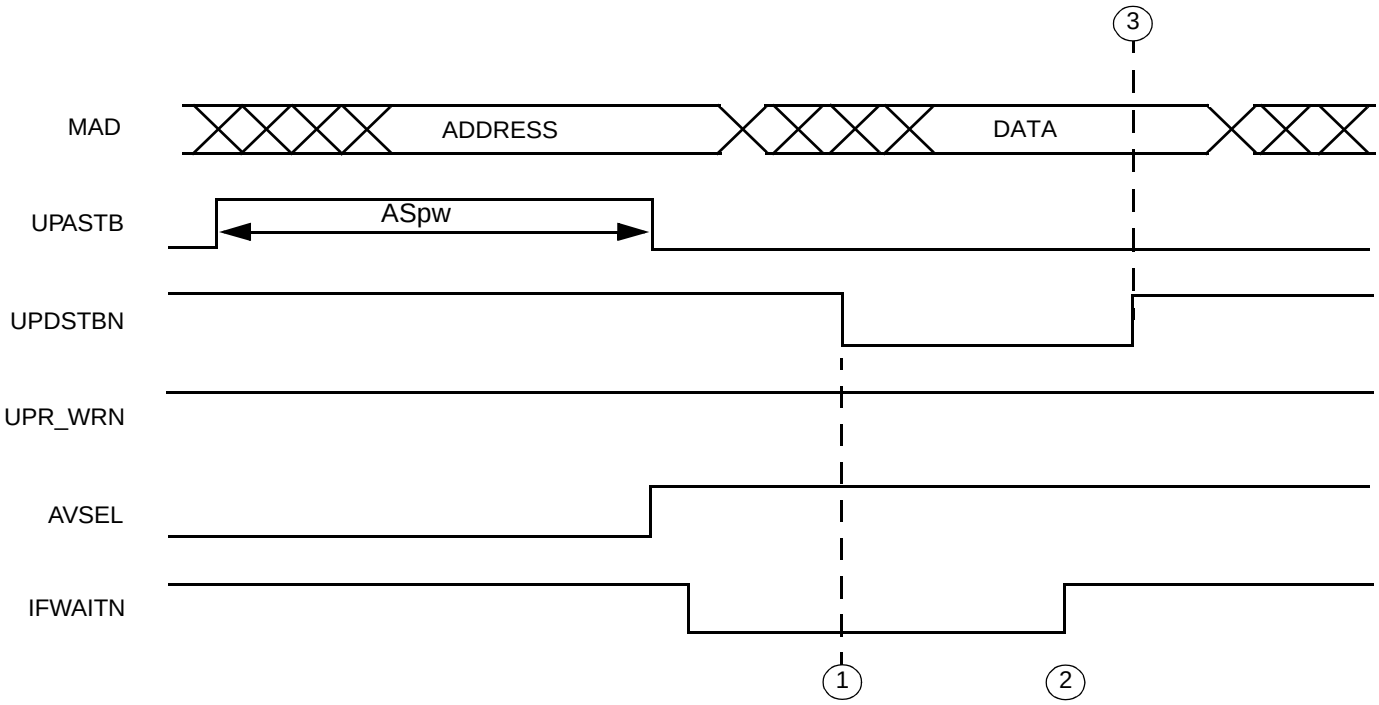
Asu — Address setup time at falling edge of UPASTB (2.5ns)
Ah — Address hold time at falling edge of UPASTB (10ns)
ASpw — Address strobe pulse width minimum (15 ns)
Dsu — Data setup time at rising edge of UPDSTBN (10ns)
DSpw — Data strobe pulse width minimum (15ns)
ASTSEL — From UPASTB falling edge to AVSEL rising edge (10ns)

(1) Genii should latch in the Address on MAD bus at falling edge of UPASTB.

(2) Genii should latch in the data on MAD bus at rising edge of UPDSTBN

(3) Genii should latch in all control signal (UPRD_WQRN, UPLBEN, UPHBEN) at falling edge of UPDSTBN

Figure 4-5: Microprocessor Write to Genii Register



(1) Genii asserts IFWAITN signal as soon as it detects its own address range. Microprocessor (µP) will keep UPDSTBN asserted as long as IFWAITN is low.

(2) After Genii has valid data on MAD bus, it will de-assert IFWAITN.

(3) Detecting IFWAITN rising, µP will de-assert UPDSTEN, µP samples data on rising edge of UPDSTBN.

Figure 4-6: Microprocessor Read Genii Register

4.3 MPEG2 INTERFACE

4.3.1 STI 5510 8-Bit Interface

Figure 4-7, on page 111 and Figure 4-7, on page 111 give examples of a single packet transfer in transport stream modes A and B, respectively. The number of null bytes depends on the time taken to start a second DMA transfer.

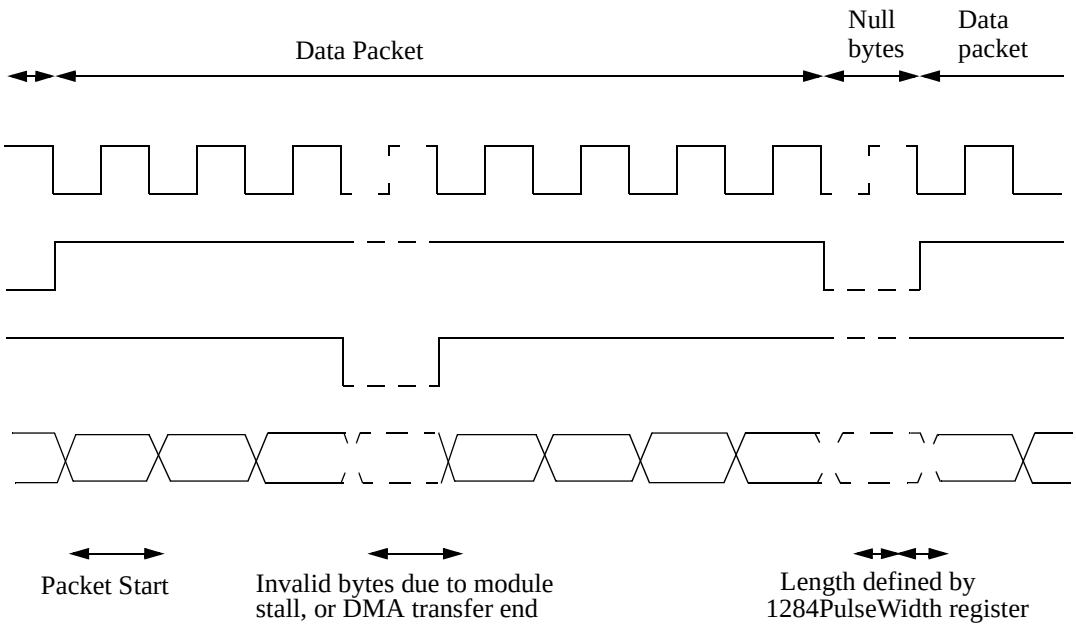


Figure 4-7: STI 5510 8-Bit Interface, Packet Transfer in Transport Stream, Mode A

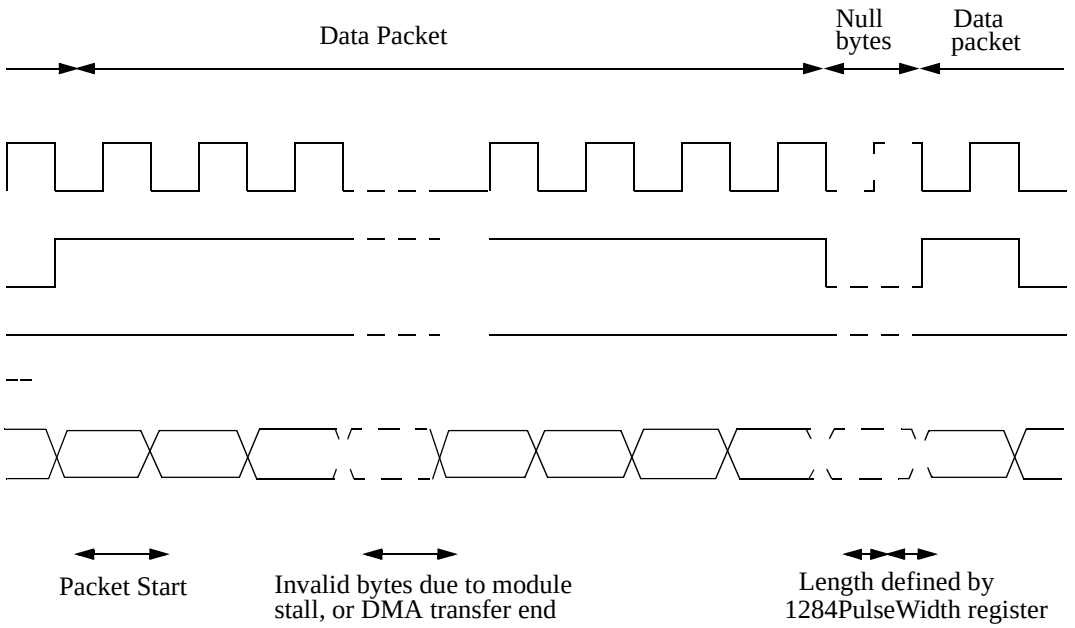


Figure 4-8: STI 5510 8-Bit Interface, Packet Transfer in Transport Stream, Mode B

4.3.2 DVB INTERFACE

The following figures show transmission formats for the DVB 8-bit interface.

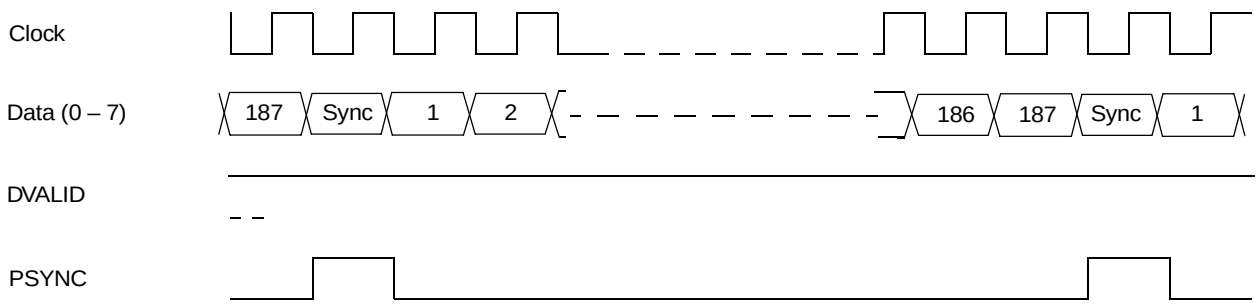


Figure 4-9: Transmission Format, 188 Byte Packets

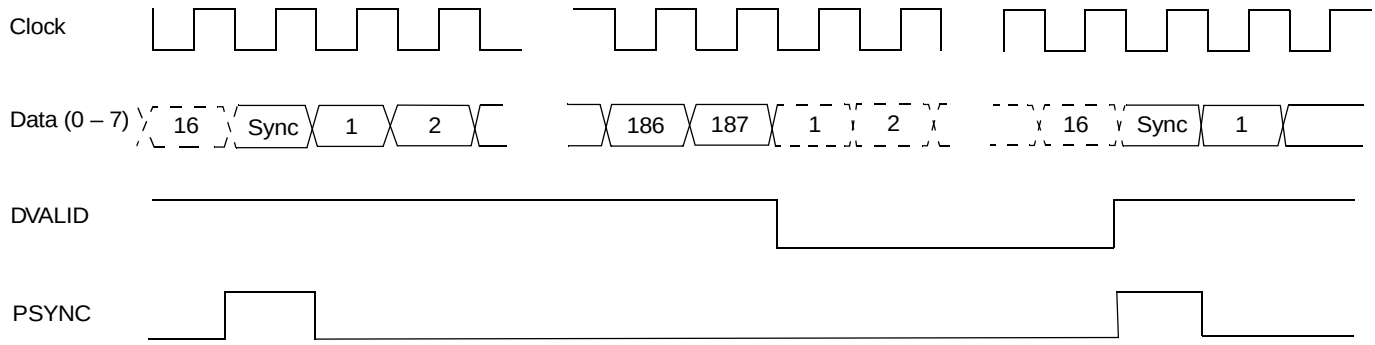


Figure 4-10: Transmission Format, 204 Byte Packets (188 Data Bytes, 16 Dummy Bytes)

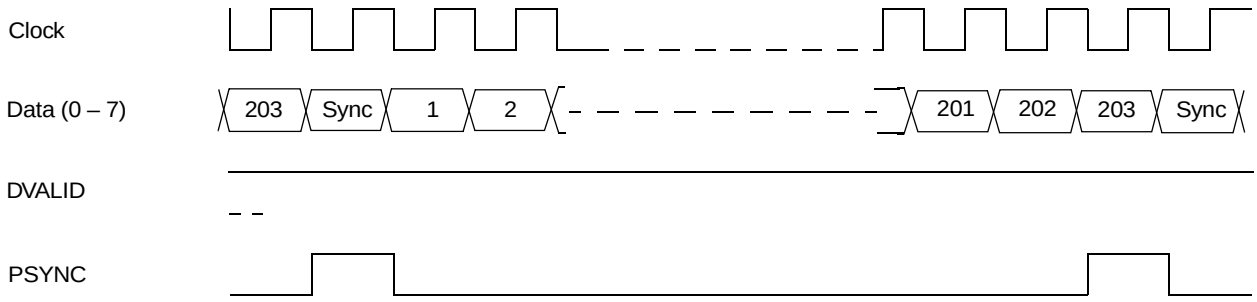


Figure 4-11: Transmission Format, 204 Bytes Packet (188 Data Bytes, 16 Valid Extra Bytes)

Volume 1 Section 5.
PHYSICAL DESCRIPTION

5.1 MECHANICAL SPECIFICATIONS

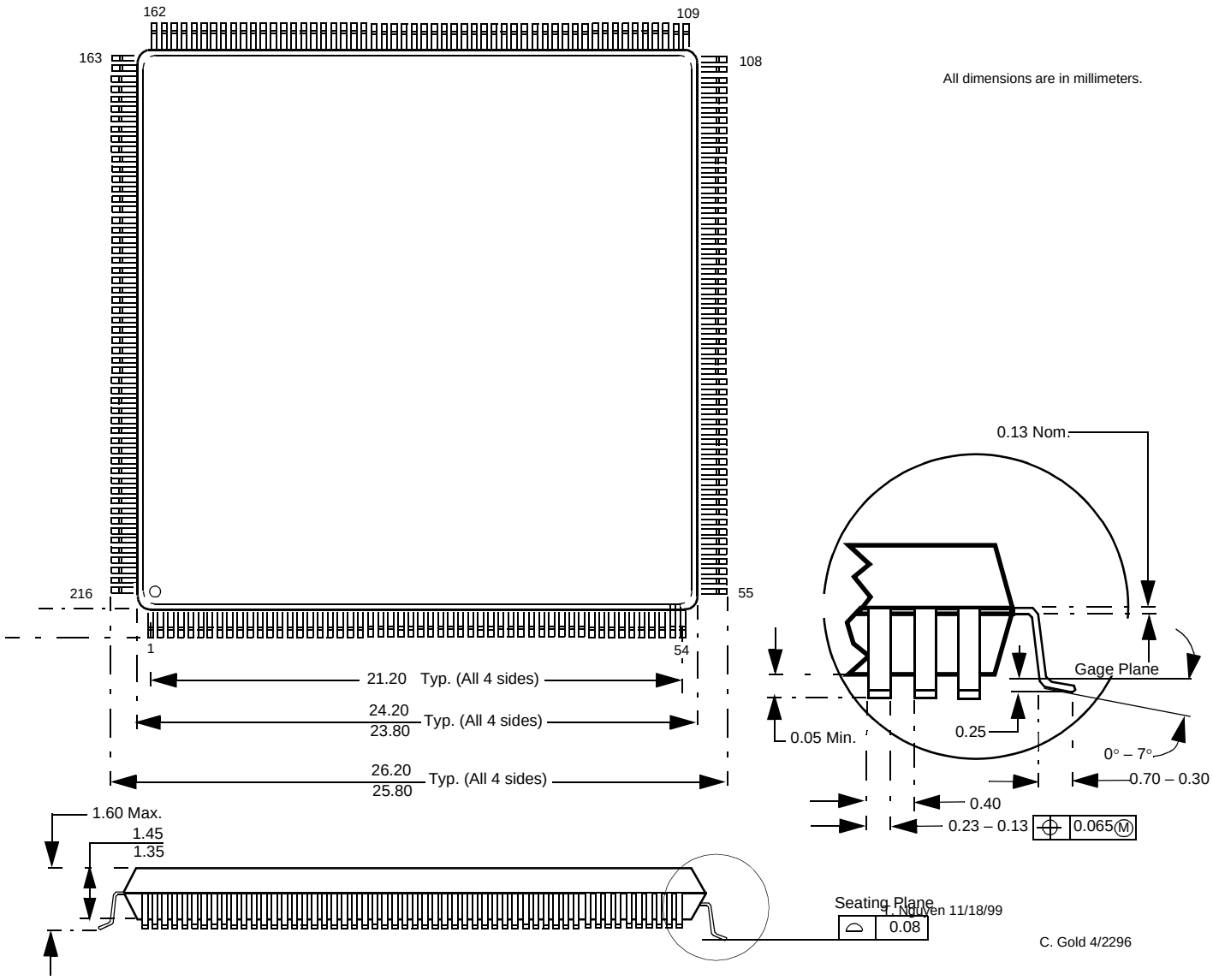
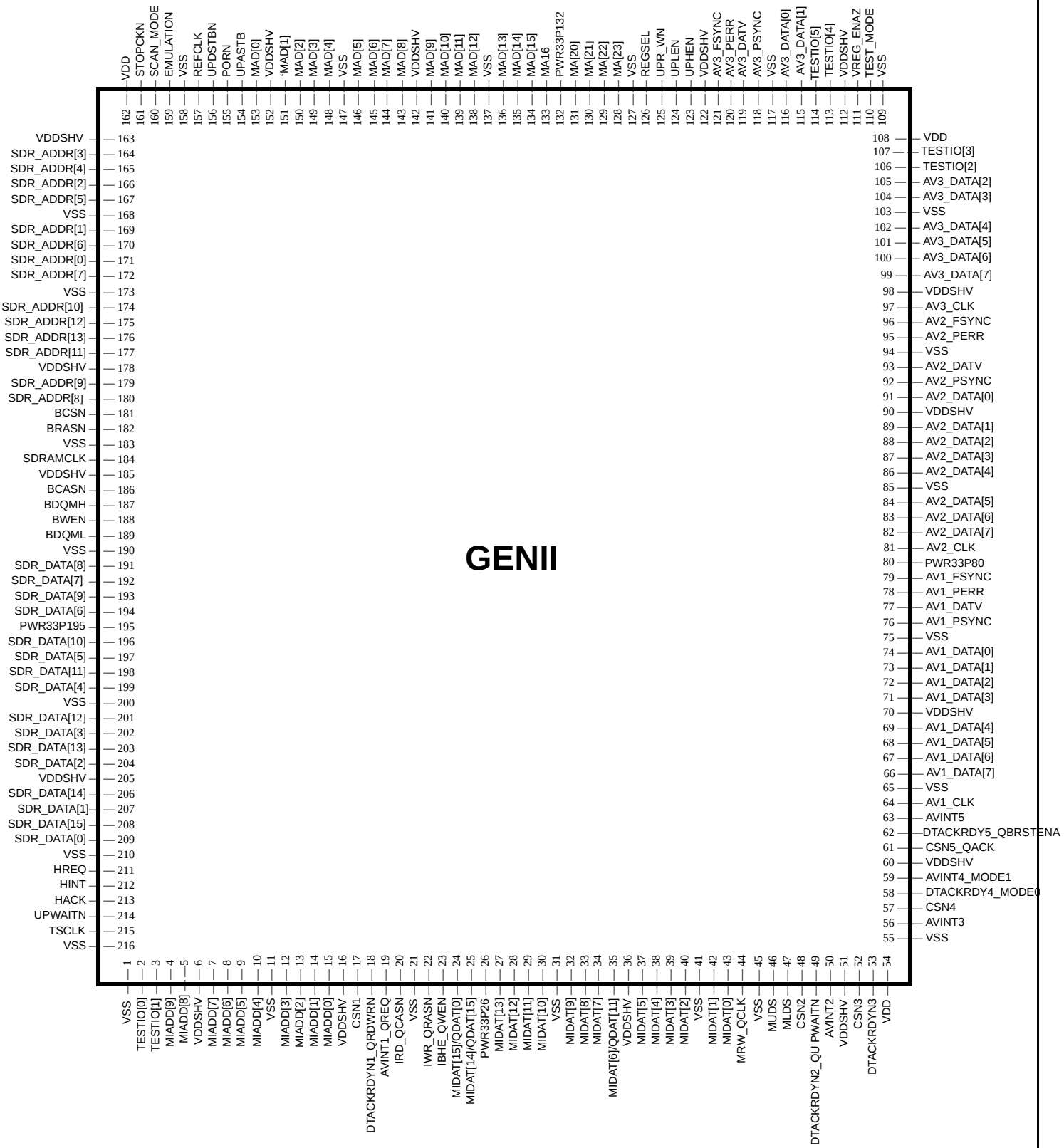


Figure 5-1: Genii 216-Pin Plastic QFP

5.2 PIN LAYOUT



5.3 PIN DESCRIPTION

Table 5-1, lists the pinout of the I/O signals.

Table 5-1: Genii Pin List (216 pins)

SIGNAL	Qty	I/O Type	IOL mA	IOH mA	RP uA	Load pF	Fmx Mhz	Pin #	DESCRIPTION
System									**Total 60pins**
PORN	1	IDG09,PS020			20U		0.01	155	System reset
REFCLK	1	IDG09,PS020			20U		40	157	Reference Clock (40 MHz)
TSCLK	1	IDG09,PE020			20U		27	215	Crystal oscillator input (27 MHz)
SCAN_MODE	1	IDG09,PE020			20D			160	Scan enable
TEST_MODE	1	IDG09,PE100			100 D			110	Test enable
TESTIO[5:0]	6	IDG09,OUK43,PE100	4	-4	100 D	20	1	Test IO bus; pin# starting from MSB: 114(bit 5), 113, 107, 106, 3, 2 (bit 0).	
VREG_ENAZ	1	IDG09,PE100			100 D			111	Voltage Regulator Enable
PWR33P26	1							26	
PWR33P80	1							3.3V Power; pin# 80	
PWR33P132	1							3.3V Power; pin# 132	
PWR33P195	1							3.3V Power; pin# 195	
VDDSHV	16							3.3V Power VDD; pin#: 6, 16, 36, 51, 60, 70, 90, 98, 112, 122, 142, 152, 163, 178, 185, 205	
VDD	3							1.8V Power VDD; VREG 1.8v output, for external filter caps; pin#: 54, 108, 162	
VSS	25							Ground pin VSS; pin#: 1, 11, 21, 31, 41, 45, 55, 65, 75, 85, 94, 103, 109, 117, 127, 137, 147, 158, 168, 173, 183, 190, 200, 210, 216	
Atlas interface									**Total 30 pins**
MA[23:20]	4	IDG09,PE020			20D		8.25	NEC 850E address bus; pin# starting from MSB: 128(bit 23), 129, 130, 131(bit 20).	
MA16	1	IDG09,PE020			20D		8.25	133	NEC 850E address bit 16
MAD[15:0]	16	IDG09,OUK43,PE100	4	-4	100 D	20	8.25	NEC 850E Address/Data multiplexer; pin# starting from MSB: 134(bit 15), 135, 136, 138, 139, 140, 141, 143, 144, 145, 146, 148, 149, 150, 151, 153(bit 0)	
UPR_WN	1	IDG09,PS020			20U		16.5	125	NEC 850E read/write enable
UPDSTBN	1	IDG09,PS020			20U		16.5	156	NEC 850E data strobe
UPHEN	1	IDG09,PS020			20U		16.5	123	NEC 850E high byte enable
UPLEN	1	IDG09,PS020			20U		16.5	124	NEC 850E low byte enable
UPASTB	1	IDG09,PE020			20D		16.5	154	NEC 850E address strobe
UPWAITN	1	OUK43	4	-4		20	16.5	214	NEC 850E wait
REGSEL	1	IDG09,PE020			20D		16.5	126	Register range is selected (MA[23:20] = F)
EMULATION	1	IDG09,PE020			20D		16.5	159	Emulation mode enable
HINT	1	OUK43	4	-4		20	0.01	212	Host interrupt
New Bypass mode interface with Ivory									**Total 40**
HREQ	1	OUK43	4	-4		20	0.02	211	Host request
HACK	1	IDG09,PE020			20D	20	0.02	213	Host acknowledge
STOPCKN	1	IDG09,OUK43,PS100			100 U	20	0.02	161	SDRAM clock enable
SDR_DATA[15:0]	16	IDG09,OUK43,PE020	4	-4	20D	25	50	SDRAM data bus; pin# starting from MSB: 208(bit 15), 206, 203, 201, 198, 196, 193, 191, 192, 194, 197, 199, 202, 204, 207, 209(bit 0)	
SDR_ADDR[13:0]	14	OUK43	4	-4		25	2	SDRAM address bus; pin# starting from MSB: 176(bit 13), 175, 177, 174, 179, 180, 172, 170, 167, 165, 164, 166, 169, 171(bit 0)	
BRASN	1	OUK43	4	-4		25	1	182	SDRAM Row address select
BCASN	1	OUK43	4	-4		25	1	186	SDRAM Column address Select
BWEN	1	OUK43	4	-4		25	1	188	SDRAM write enable
BDQMH	1	OUK43	4	-4		25	1	187	SDRAM high byte enable
BDQML	1	OUK43	4	-4		25	1	189	SDRAM low byte enable
BCSN	1	OUK43	4	-4		25	1	181	SDRAM chip select
SDRAMCLK	1	OUK43	4	-4		25	100	184	SDRAMCLK
AV chips interface									**Total 39 pins**
AV1_CLK	1	IDG09,OUK43,PS100	4	-4	100 U	20	25	64	Audio/Video CLK (25 MHz). This pin can be programmed as an input or output.

Table 5-1: Genii Pin List (216 pins) (Continued)

SIGNAL	Qty	I/O Type	IOL mA	IOH mA	RP uA	Load pF	Fmx Mhz	Pin #	DESCRIPTION
AV1_DATA[7:0]	8	IDG09,OUK43,PE100	4	-4	100 D	20	12.5	Audio/Video data bus; pin # starting from MSB: 66(bit 7), 67, 68, 69, 71, 72, 73, 74(bit 0)	
AV1_PSYNC	1	IDG09,OUK43,PE100	4	-4	100 D	20	0.07	76	Audio/Video start package indicator
AV1_DATV	1	IDG09,OUK43,PE100	4	-4	100 D	20	0.07	77	Indicates data is valid
AV1_PERR	1	IDG09,OUK43,PE100	4	-4	100 D	20	0.07	78	Indicates current package is unreliable (CRC error, sequence error, data length error)
AV1_FSYNC	1	IDG09,OUK43,PE100	4	-4	100 D	20	0.07	79	Indicates frame is sync; Used for digital video
AV2_CLK	1	IDG09,OUK43,PS100	4	-4	100 U	20	25	81	Audio/Video CLK (25 MHz). This pin can be programmed as an input or output.
AV2_DATA[7:0]	8	IDG09,OUK43,PE100	4	-4	100 D	20	12.5	Audio/Video data bus; pin# starting from MSB: 82(bit 7), 83, 84, 86, 87, 88, 89, 91(bit 0)	
AV2_PSYNC	1	IDG09,OUK43,PE100	4	-4	100 D	20	0.07	92	Audio/Video start package indicator
AV2_DATV	1	IDG09,OUK43,PE100	4	-4	100 D	20	0.07	93	Indicates data is valid
AV2_PERR	1	IDG09,OUK43,PE100	4	-4	100 D	20	0.07	95	Indicates current package is unreliable (CRC error, sequence error, data length error)
AV2_FSYNC	1	IDG09,OUK43,PE100	4	-4	100 D	20	0.07	96	Indicates frame is sync; Used for digital video
AV3_CLK	1	IDG09,OUK43,PS100	4	-4	100 U	20	25	97	Audio/Video CLK (25 MHz). This pin can be programmed as an input or output.
AV3_DATA[7:0]	8	IDG09,OUK43,PE100	4	-4	100 D	20	12.5	Audio/Video data bus; pin# starting from MSB: 99(bit 7), 100, 101, 102, 104, 105, 115, 116(bit 0)	
AV3_PSYNC	1	IDG09,OUK43,PE100	4	-4	100 D	20	0.07	118	Audio/Video start package indicator
AV3_DATV	1	IDG09,OUK43,PE100	4	-4	100 D	20	0.07	119	Indicates data is valid
AV3_PERR	1	IDG09,OUK43,PE100	4	-4	100 D	20	0.07	120	Indicates current package is unreliable (CRC error, sequence error, data length error)
AV3_FSYNC	1	IDG09,OUK43,PE100	4	-4	100 D	20	0.07	121	Indicates frame is sync; Used for digital video
Motorola and Intel Interface									**Total 47 pins**
MIADD[9:0]	10	OUK83	8	-8		30	1	Motorola/Intel host address; pin# starting from MSB: 4(bit 9), 5, 7, 8, 9, 10, 12, 13, 14, 15(bit 0)	
MIDAT[15]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	24	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[14]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	25	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[13]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	27	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[12]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	28	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[11]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	29	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[10]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	30	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[9]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	32	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[8]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	33	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[7]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	34	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[6]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	35	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[5]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	37	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[4]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	38	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[3]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	39	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[2]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	40	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MIDAT[1]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	42	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)

Table 5-1: Genii Pin List (216 pins) (Continued)

SIGNAL	Qty	I/O Type	IOL mA	IOH mA	RP uA	Load pF	Fmx Mhz	Pin #	DESCRIPTION
MIDAT[0]	1	IDG09,OUK83,PE100	8	-8	100 D	30	25	43	Motorola/Intel host data (3221Q_MODE = 0) DMA data (3221Q_MODE = 1)
MRW_QCLK	1	IDG09,OUK83,PS100	8	-8	100 U	30	50	44	Motorola Read/Write Enable (3221Q_MODE = 0) 50MHz DMA data clock (3221Q_MODE = 1)
MIADD[7]	1							7	
MIADD[6]	1							8	
MIADD[5]	1							9	
MIADD[4]	1							10	
MIADD[3]	1							12	
MIADD[2]	1							13	
MIADD[1]	1							14	
MIADD[0]	1							15	
MUDS	1	OUK83	8	-8		30	2	46	Motorola High Byte Enable
MLDS	1	OUK83	8	-8		30	2	47	Motorola Low Byte Enable
IRD_QCASN	1	OUK83	8	-8		30	2	20	Intel read strobe (3221Q_MODE = 0) DMA column address select(3221Q_MODE = 1)
IWR_QRASN	1	OUK83	8	-8		30	2	22	Intel write strobe (3221Q_MODE = 0) DMA row address select(3221Q_MODE = 1)
IBHE_QWEN	1	OUK83	8	-8		30	2	23	Intel high byte enable (3221Q_MODE = 0) DMA write enable (3221Q_MODE = 1)
CSN1	1	OUK43	4	-4		25	0.01	17	Chip Select for AV device 1
DTACKRDYN1_QRDWRN	1	IDG09,PS100			100 U		2	18	Data acknowledge/ready for AV device 1 (3221Q_MODE = 0) DMA direction (3221Q_MODE = 1)
AVINT1_QREQ	1	IDG09,PE100			100 D		2	19	Interrupt from AV device 1 (3221Q_MODE = 0) DMA request (3221Q_MODE = 1)
CSN2	1	OUK43	4	-4		25	0.01	48	Chip Select for AV device 2
DTACKRDYN2_QUPWAITN	1	IDG09,PS100			100 U		2	49	Data acknowledge/ready for AV device 2 (3221Q_MODE = 0) DMA uP wait (3221Q_MODE = 1)
AVINT2	1	IDG09,PE100			100 D		0.01	50	Interrupt from AV device 2
CSN3	1	OUK43	4	-4		25	0.01	52	Chip Select for AV device 3
DTACKRDYN3	1	IDG09,PS100			100 U		2	53	Data acknowledge/ready for AV device 3
AVINT3	1	IDG09,PE100			100 D		0.01	56	Interrupt from AV device 3
CSN4	1	OUK43	4	-4		25	0.01	57	Chip Select for AV device 4
DTACKRDY4_MODE0	1	IDG09,PS100			100 U		2	58	Data acknowledge/ready for AV device 4 (3221Q_MODE = 0) DMA channel select (3221Q_MODE = 1)
AVINT4_MODE1	1	IDG09,PE100			100 D		2	59	Interrupt from AV device 4 (3221Q_MODE = 0) DMA channel select (3221Q_MODE = 1)
CSN5_QACK	1	OUK43	4	-4		25	2	61	Chip Select for AV device 5 3221Q_MODE = 0) DMA acknowledge (3221Q_MODE = 1)
DTACKRDY5_QBRSTENA	1	IDG09,PS100			100 U		2	62	Data acknowledge/ready for AV device 5 (3221Q_MODE = 0) DMA burst enable (3221Q_MODE = 1)
AVINT5	1	IDG09,PE100			100 D		0.01	63	Interrupt from AV device 5

Appendix A
Generate Zero Difference P-Picture

A.1 TRANSPORT PACKET HEADER

Transport packet header can be copied from the video stream's transport packet header. The continuity counter needs to assume the value of the on board continuity counter. The continuity counter must be locked to the original stream's continuity counter before the occurrence of the trick play command where the null or the zero difference P-picture will be used.

A.1.1 PES header

The first transport stream packet of the null p-picture will contain the PES header along with the picture_start_code (6.2.3 IEC13818-2) and any extension_start_codes (6.2.2.2.1 IEC 13818-2) fields which may follow (and can also be copied from any P-picture within the stream being played).

A.1.1.1 Fields of interest:

Vertical_size and horizontal_size fields in the sequence_header (6.2.2.1 IEC 13818-2) determine the number of slices per picture ((vertical_size+8)/16), and the number of macroblocks (horizontal_size/16) per slice.

Vertical_size value larger than 2800 and sequence_scalable_extension (6.2.2.5 IEC 13818-2) present in the stream (extension start code followed by 0x5X) affect the slice structure.

The values of the fields picture_structure and frame_pred_frame_dct within the picture_coding_extension (6.2.3.1 IEC13818-2) affect the decoding of the macroblock_modes (6.2.5.1 IEC13818-2) This also implies selection of either frame_motion_type or field_motion_type (tables 6-17 and 6-18 IEC 13818-2) dictating the number and meaning of motion_vectors.

A.1.1.2 Slices

Vertical_size is not due to exceed 2800 anytime in the foreseeable future. In the presence of sequence_scalable_extension (see 6.2.4 in specification IEC 13818-2). The streams that we have in our possession do not contain any scalable extensions; the first 5 bits after slice_start_code are the quantiser_scale_code.

These are followed by one zero-value bit (extra_bit_slice).

A.1.1.3 Slice Construction

The following table shows slice construction for zero difference p-picture

Table E-2: Zero Difference Picture Slice Construction

		picture_structure = 11 Frame_pred_frame_dct = 0 sequence_scalabe_extension: Not present scalable_mode: N/A vertical_size < 2800	picture_structure = 11 Frame_pred_frame_dct = 0 sequence_scalabe_extension: Present scalable_mode: Data partitioning vertical_size < 2800	picture_structure = 11 Frame_pred_frame_dct = 0 sequence_scalabe_extension: Present scalable_mode: SNR vertical_size < 2800	picture_structure = XX Frame_pred_frame_dct = X sequence_scalabe_extension: Present scalable_mode: Spatial vertical_size < 2800	picture_structure = 11 Frame_pred_frame_dct = 0 sequence_scalabe_extension: Present scalable_mode: Temporal vertical_size < 2800	picture_structure = 11 Frame_pred_frame_dct = 1 sequence_scalabe_extension: Not present scalable_mode: N/A vertical_size < 2800	picture_structure != 11 Frame_pred_frame_dct = X sequence_scalabe_extension: Not present scalable_mode: N/A vertical_size < 2800	EXAMPLE: 1920H X 480V, frame, fpfd = '0' "skiing" clip slice: 0X 00 00 01 01 1A 72 6C 04 05 9B Repeated 30 times, 4th bypte values 1 – 0x1E	EXAMPLE: 920H X 1080V, frame, fpfd = '0' "soap1" clip slide: 0X 00 00 01 01 1A 6C 04 00 80 10 09 9B Repeated 68 times, 4th byte values, 1 – 0x44
A	**Start_code (hex, 24 bits)**	000001'	000001'	000001'	000001'	000001'	000001'	000001	000001'	000001'
B	**Slice number (hex, 8 bits)[a]**	01 — AF	01 — AF	01 — AF	01 — AF	01 — AF	01 — AF	01 — AF	01	01
C	**Slice_vertical position_ extension (3 bits)[b]**	N/A	N/A	N/A	N/A	N/A	N/A	N/A	01110	00011
D	**priority_ breakpoint (7 bits, from original P-picture slice)**	N/A	PB[6:0]	N/A	N/A	N/A	N/A	N/A		
E	**quantizer_ scale_code (5 bits, from original P-picture slice)**	Q[4:0]	Q[4:0]	Q[4:0]	Q[4:0]	Q[4:0]	Q[4:0]	Q[4:0]		
F	**extra_bit_ slice (1 bit, 0 value)**	0	0	0	0	0	0	0	0	0
G	**macro_block_ address_ increment (1 – 11 bits VLBC)**	1	1	1	1	1	1	1	1	1
H	**macroblock_ type (1 – 9 bits, VLBC)**	001'	001'	001'	0000 001'	001'	001'	001'	001'	001'
I	**frame_ motion_type (2 bits)**	10	10	N/A	N/A	10	N/A	N/A	10	10

Table E-2: Zero Difference Picture Slice Construction (Continued)

		picture_structure = 11 Frame_pred_frame_dct = 0 sequence_scalabe_extension: Not present scalable_mode: N/A vertical_size < 2800	picture_structure = 11 Frame_pred_frame_dct = 0 sequence_scalabe_extension: Present scalable_mode: Data partitioning vertical_size < 2800	picture_structure = 11 Frame_pred_frame_dct = 0 sequence_scalabe_extension: Present scalable_mode: SNR vertical_size < 2800	picture_structure = XX Frame_pred_frame_dct = X sequence_scalabe_extension: Present scalable_mode: Spatial vertical_size < 2800	picture_structure = 11 Frame_pred_frame_dct = 0 sequence_scalabe_extension: Present scalable_mode: Temporal vertical_size < 2800	picture_structure = 11 Frame_pred_frame_dct = 1 sequence_scalabe_extension: Not present scalable_mode: N/A vertical_size < 2800	picture_structure != 11 Frame_pred_frame_dct = X sequence_scalabe_extension: Not present scalable_mode: N/A vertical_size < 2800	EXAMPLE: 1920H X 480V, frame, fpfd = '0' "skiing" clip slice: 0X 00 00 01 01 1A 72 6C 04 05 9B Repeated 30 times, 4th bypte values 1 – 0x1E	EXAMPLE: 920H X 1080V, frame, fpfd = '0' "soap1" clip slide: 0X 00 00 01 01 1A 6C 04 00 80 10 09 9B Repeated 68 times, 4th byte values, 1 – 0x44
J	**field_motion_ type (2 bits)**	N/A	N/A	N/A	N/A	N/A	N/A	01’	1	1
K	**motion_ vertical_field_ select (1 bit)**	N/A	N/A	N/A	N/A	N/A	N/A	1		
L	**motion_code [0][s][0] (1 – 11 bits, VLBC)**	1	1	N/A	N/A	1	1	1		
M	**motion code [0][s][1] (1 – 11 bits, VLBC)**	1	1	N/A	N/A	1	1	1	1	1
N	**macroblock_ escape (repeated N times)[c]**	0000 0001 000’	0000 0001 000’	0000 0001 000’	0000 0001 000’	0000 0001 000’	0000 0001 000’	0000 0001 000’	00 0000 0100 0’	00 0000 0100 0000 0000 10000 0000 0001 000
O	**macro_block_ address_ increment (1 – 11 bits, VLBC)[d]**	A[1:0]	A[1:0]	A[1:0]	A[1:0]	A[1:0]	A[1:0]	A[1:0]	000 0101 1 (10 decimal)	0 0000 1001 1 (20 decimal)
P	**duplicate bits in rows I – N**	00110011’	00110011’	001’	0011 011’	00111’	0010 1111’	0010 1111’	001 1011’	001 1011’
Q	**zero value fill bits: type-align the slice**	zeroes as required	zeroes as required	zeroes as required	zeroes as required	zeroes as required	zeroes as required	zeroes as required	no zeroes required (40 bits)	no zeroes required (64 bits)

a. slice_number starts at ’01 hex’, and increments to M, with each slice. M = INT ((vertical_size/16) + 0.5)jk

b. This field is present in a stream only for pictures of vertical_size >= 2800. The limit by today’s standards (DVB, ATSC) is of 1152 lines.

c. This code is repeated N times. N = INT (((horizontal_size/16) - 2) / 33)

d. This field is determined as follows: macroblock_address_increment = ((horizontal_size / 16) - 1) % 33)

A.1.1.4 Slice Construction Fields

The following table shows the location of and the legend for fields used in slice construction.

Table E-3: Slice Construction Fields

FIELD	LOCATION
sequence_header_code	00 00 01 B3 HEX
horizontal_size_value	12 bits following sequence header code
vertical_size_value	12 bits following horizontal_size_value
sequence_scalable_extension start code	00 00 01 B5 BX Hex: **X** **scalable_mode** 00XX data partitioning 01XX spatial 10XX SNR 11XX temporal
picture_coding_extension start code: 00 0001 B5 8X	f_code[0][0] (forward, horizontal) X (4 bits) f_code[0][1] (forward, vertical) next 4 bits f_code[1][0] (backward, horizontal) next 4 bits f_code[1][1] (backward, vertical) next 4 bits intra_dc_precision next 2 bits picture_structure next 2 bits *The definition of picture_structure follows:* *picture_structure code* *definition* *00* *Reserved* *01* *Top field* *10* *Bottom field* *11* *Frame picture* top_field_first 1 bit frame_pred_frame_dct 1 bit *NOTE: frame_pred_frame_dct is '0' for field pictures. It must be set to '1' if the progressive_frame field is '1'.* concealment_motion_vectors 1 bit q_scale_type 1 bit intra_vic_format 1 bit alternate_scan 1 bit repeat_first_field 1 bit chroma_420_type 1 bit progressive_frame 1 bit composite_display_flag 1 bit

A.1.2 Macroblock

The first field in a macroblock is either the macroblock_address_increment or the macroblock_escape code. It is a variable length code (Huffman code) and, when added to the previous macroblock address, it indicates the address of the current macroblock. Skipping macroblocks in a p-picture indicate that there were no changes to them. The first and last macroblocks of a slice can not be skipped. The macroblock_escape code can be used to skip 33 macroblocks and it must be followed by more macroblock_escape codes or by a macroblock_address_increment code.

Theoretically, we only need to describe the first and last macroblocks in a slice, and skip all the ones in between.

A.1.2.1 Macroblock modes

This section extracts the description of the macroblock_type from tableB.3 (or B.6 or B.8 for spatially-scalable and SNR-scalable pictures). The parameters extracted from the macroblock_type tables, along with the information described above under “fields of interest,” are used to define the data in the macroblock (frame_motion_type or field_motion_type: Table E-4 on page 124 TTable E-5 on page 124 — based on tables 6-17 and 6-18 in IEC13818-2).

A.1.2.2 Motion Vectors

The parameters extracted from the _motion_type tables above, determine how many vectors will follow.

A.1.2.3 Motion Vector

In this module, the value of the motion vectors are extracted. Based on their values and the values of the f_code[s][r] fields (picture coding extension), it is then determined if there is a residual error for this macroblock. The ‘dmv’ flag, extracted from Table E-4 on page 124 or Table E-5 on page 124, determine the presence of a differential_motion_vector that is also extracted in this module.

Table E-4: Definition of frame_motion_type[a]

Code	spatial_temporal_weight_class (from table B.3, B.6, or B.6)	Prediction Type	motion_vector _count	mv_format	dmv
00		Reserved			
01	0, 1	Field-Based	2	Field	0
01	2, 3	Field-Based	1	Field	0
10	0, 1, 2, 3	Field	1	Frame	0
11	0, 2, 3	Dual-Prime	1	Field	1

a. This information is copied from Table 6-17 of IEC13818-2

Table E-5: Definition of field_motion_type[a]

Code	spatial_temporal_weight_class (from table B.3, B.6, or B.6)	Prediction Type	motion_vector _count	mv_format	dmv
00		Reserved			
01	0, 1	Field-Based	1	Filed	0
10	0, 1	16 x 18 MC	2	Frame	0
11	0	Dual-Prime	1	Field	1

a. This information is copied from Table 6-18 of IEC13818-2

Appendix B
Update Log

This section lists the updates of this document. Recent changes are marked with change bars: vertical lines on the left side of the page.

Jan 6 2000

- GENII CAPABILITIES AND FEATURES on page 10
- Figure 1-1, “Genii Block Diagram,” on page 11
- 3221Q To DMA Interface (Q2DMA Interface) on page 13
- Figure 1-3, “Q2 DMA Interface, 3211Q To AVDMA,” on page 13
- Figure 1-4, “Q2 DMA Interface, Block Diagram,” on page 14
- Figure 2-1, “Modes: Normal, Trick Play,” on page 17
- PCR and Discontinuity Bit on page 18
- Comparisons on page 18
- Figure 2-4, “Original Clip,” on page 20
- Figure 2-5, “Transmitted Clip,” on page 20
- Figure 2-6, “Trick Play 1 Control State Machine,” on page 23
- Figure 2-7, “Start Trick Play,” on page 24
- Register Setup for Stop Trick Play on page 25
- Table 3-6, “General Registers,” on page 32
- Bit 12: PAT_INT on page 38
- Bit 2: ISTART_INT on page 39
- Bit 1: NOSPACE_INT on page 39
- Bits 15–0: I1_FIFODAT[15:0] on page 45
- Bits 8–0: I1_FIFOADD[8:0] on page 46
- Bits [12:0]: PCR_PID[12:0] (Write) on page 46
- Bits [12:0]: PCR_PID[12:0] (Read) on page 46
- Bits [12:0] Video_PID[12:0] (Read) on page 47
- REG - 83BCh – 83BDh I1_PMTTABLEDAT: I1 PMT Table Data Register on page 49
- Table 4-1, “ Packet Parse Block,” on page 100
- Table 4-2, “Video Parse Block,” on page 101
- Table 4-3, “I-frame Table,” on page 101
- Table 4-4, “PAT and PMT Tables,” on page 102
- Table 4.1.5.2, “PMT Parser,” on page 103
- Figure 4-10, “Transmission Format, 204 Byte Packets (188 Data Bytes, 16 Dummy Bytes),” on page 112

December 10 1999

- REG - 84B0h VERSION: Version Register on page 98

December 7 1999

- REG - 8480–8481h DRAMTIMING: DRAM State Machine Timing Control on page 85
- REG - 8482h–8483h CONFIG: Configuration on page 86
- REG - 8484h–8485h INT_ENA: Interrupt Enable on page 86
- REG - 8486h–8487h INTERRUPT: Interrupt on page 87

Numerics

A

B

C

D

I

S

T

U

V

X