SGS-THOMSON
MICROELECTRONICS

STi3520

MPEG AUDIO / MPEG-2 VIDEO INTEGRATED DECODER

PRELIMINARY DATA

- SINGLE CHIP COMBINING THE DECODING FUNCTIONS OF THE STi3500A VIDEO DECODER AND THE STi4500 AUDIO DECODER
- VIDEO DECODER FULLY SUPPORTS MPEG-2 MAIN PROFILE/MAIN LEVEL (MP@ML)
- SUPPORT FOR SYNCHRONOUS AND HYPER-PAGE-MODE (EDO) DRAMS
- AUDIO DECODER SUPPORTS LAYERS I & II OF MPEG
- ALL POPULAR PCM AUDIO OUTPUT FORMATS SUPPORTED
- AUTOMATIC VIDEO ERROR CONCEALMENT
- ENHANCED ON-SCREEN DISPLAY GENERATOR : 16 COLOURS/REGION, LINKED LIST MEMORY MANAGEMENT
- STANDARD 8-BIT INTERFACE FOR MICROCONTROLLER AND COMPRESSED DATA INPUT
- 3.3V POWER SUPPLY, I/Os 5V COMPATIBLE
- 0.5µm CMOS TECHNOLOGY
- ON CHIP PLL ALLOWING FULL CHIP OPERATION WITH TWO EXTERNAL CLOCKS

APPLICATIONS

- DBS RECEIVER
- DIGITAL TV RECEIVER
- DIGITAL CABLE TV RECEIVER

DESCRIPTION

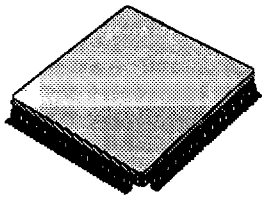
The video decoder is a real-time video decompression processor supporting the MPEG-1 and MPEG-2 standards at video rates up to 720 x 480 x 60Hz or 720 x 576 x 50Hz. Picture format conversion for display is performed by a vertical and a horizontal filter (sample rate converter). External DRAM, typically of size 16 Mbits is required.

The audio decoder is compliant with layers I and II of the MPEG standard. Sampling rates of 32, 44.1 and 48kHz can be used.

The STi3520 requires minimal support from an external microcontroller, which is mainly required to initialise the video decoder at the start of every picture. Separate audio and video data streams are input through the 8-bit data port. Provision is made for detection of time stamps for the management of audio/video synchronization.

User-defined bitmaps may be superimposed on the displayed picture through use of the on-screen display function. These bitmaps are written directly into the DRAM memory by the microcontroller.

Undetected bitstream errors which would cause decoder errors activate the error concealment functions.



PQFP160
(Plastic Quad Flat Pack)

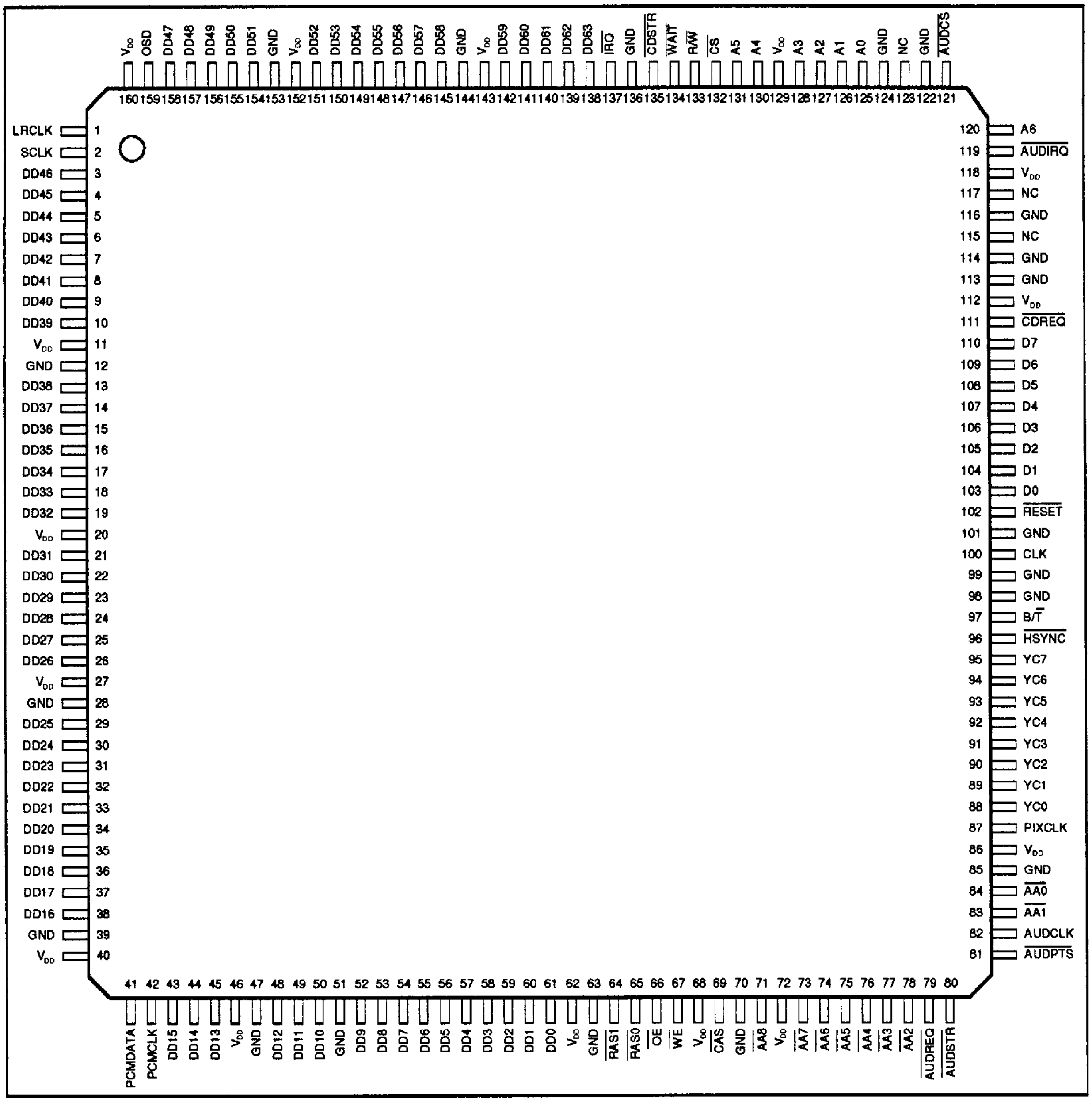
ORDER CODE : STi3520

This is advance information on a new product now in development or undergoing evaluation. Details are subject to change without notice.

CONTENTS

I - PIN DESCRIPTION

I.1 - Pin Connections



3520-01.EPS

I.2 - Pin List

Pin Number	Name	Type	Function
SYSTEM SERVICES AND NON-FUNCTIONAL PINS			
2, 11, 20, 27, 46, 62, 68, 72, 86, 112, 129, 143, 152, 160	V_{DD}		Power Supply
1, 12, 28, 47, 51, 63, 70, 85, 101, 113, 136, 144, 153	GND		Ground
100	CLK	I	Video Decoder Clock
82	AUDCLK	I	Audio Decoder Clock
102	$\overline{\text{RESET}}$	I	Master Reset
115, 117, 123		N/C	Reserved Pin, Leave Unconnected
118		V_{DD}	Reserved Pin, Connect to V_{DD}
98, 99, 114, 116, 122, 124		GND	Reserved Pins, Connect to Ground
MICROCONTROLLER INTERFACE			
110-103	D7 - D0	I/O	Bidirectional Data Bus
120, 131, 130, 128-125	A6 - A0	I	Address
132	$\overline{\text{CS}}$	I	Chip Select (video)
121	$\overline{\text{AUDCS}}$	I	Chip Select (audio)
133	R/$\overline{\text{W}}$	I	Read/Write Selection
134	$\overline{\text{WAIT}}$	O (3-state)	Data Acknowledge
137	$\overline{\text{IRQ}}$	O (open-drain)	Interrupt Request (video)
119	$\overline{\text{AUDIRQ}}$	O (open-drain)	Interrupt Request (audio)
AUDIO TIME-STAMP FLAG			
81	$\overline{\text{AUDPTS}}$	O	Indicates Latching of Audio Time-Stamp
STANDARD DRAM INTERFACE			
138-142, 145-151, 154-158, 3-10, 13-19, 21-26, 29-38, 43-45, 48-50, 52-61	DD63 - DD0	I/O	Bidirectional Data Port
71, 73-78, 83, 84	$\overline{\text{AA8}}$ - $\overline{\text{AA0}}$	O	Address
64, 65	$\overline{\text{RAS1}}$, $\overline{\text{RAS0}}$	O	Row Address Strobes for Banks 1 and 0
69	$\overline{\text{CAS}}$	O	Column Address Strobe
66	$\overline{\text{OE}}$	O	Output Enable
67	$\overline{\text{WE}}$	O	Write Enable
EDO DRAM INTERFACE (only in STi3520 cut 2.0)			
158, 3-10, 13-19	DD31 - DD16	I/O	Bidirectional Data Port
21-26, 29-38	DD15 - DD0	I/O	Bidirectional Data Port
52	$\overline{\text{AA11}}$	O	Address
53	$\overline{\text{AA10}}$	O	Address
54	$\overline{\text{AA9}}$	O	Address
56	$\overline{\text{WE}}$	O	Write Enable
57	$\overline{\text{CAS}}$	O	Column Address Strobe
58	$\overline{\text{RAS}}$	O	Row Address Strobe (single bank)
60	$\overline{\text{OE}}$	O	Output Enable
71, 73-78, 83, 84	$\overline{\text{AA8}}$ - $\overline{\text{AA0}}$	O	Address

3520-01.TBL

Pin Number	Name	Type	Function
SYNCHRONOUS DRAM INTERFACE (only in STi3520 cut 2.0)			
21-26, 29-38	DD15 - DD0	I/O	Bidirectional Data Port
52	$\overline{\text{AA11}}$	O	Address
53	$\overline{\text{AA10}}$	O	Address
54	$\overline{\text{AA9}}$	O	Address
55	$\overline{\text{CS}}$	O	Chip Select
56	$\overline{\text{WE}}$	O	Write Enable
57	$\overline{\text{CAS}}$	O	Column Address Strobe
58	$\overline{\text{RAS}}$	O	Row Address Strobe (single bank)
59	CKE	O	Clock Enable
60	$\overline{\text{OE}}$	O	Output Enable
61	CLK	O	Synchronous DRAM Clock
71, 73-78, 83, 84	$\overline{\text{AA8}}$ - $\overline{\text{AA0}}$	O	Address
VIDEO OUTPUT INTERFACE			
95-88	YC7 - YC0	O	Multiplexed YC_BC_R Video Port
87	PIXCLK	I	Pel Clock
97	B/$\overline{\text{T}}$	I	Bottom/Top Field Selection (vertical sync)
96	$\overline{\text{HSYNC}}$	I	Horizontal Sync
159	OSD	I/O	OSD Enable/OSD Active
AUDIO OUTPUT INTERFACE			
39	LRCLK	O	Left/Right Channel Select Output
40	SCLK	O	PCM Clock Output
41	PCMDATA	O	PCM Serial Data Output
42	PCMCLK	I	PCM Clock Input
COMPRESSED DATA INPUT CONTROL			
111	$\overline{\text{CDREQ}}$	O	Video Compressed Data Request
135	$\overline{\text{CDSTR}}$	I	Video Compressed Data Strobe
79	$\overline{\text{AUDREQ}}$	O	Audio Compressed Data Request
80	$\overline{\text{AUDSTR}}$	I	Audio Compressed Data Strobe

3520-02.TBL

II - BLOCK DIAGRAMS

Figure 1 : General Block Diagram

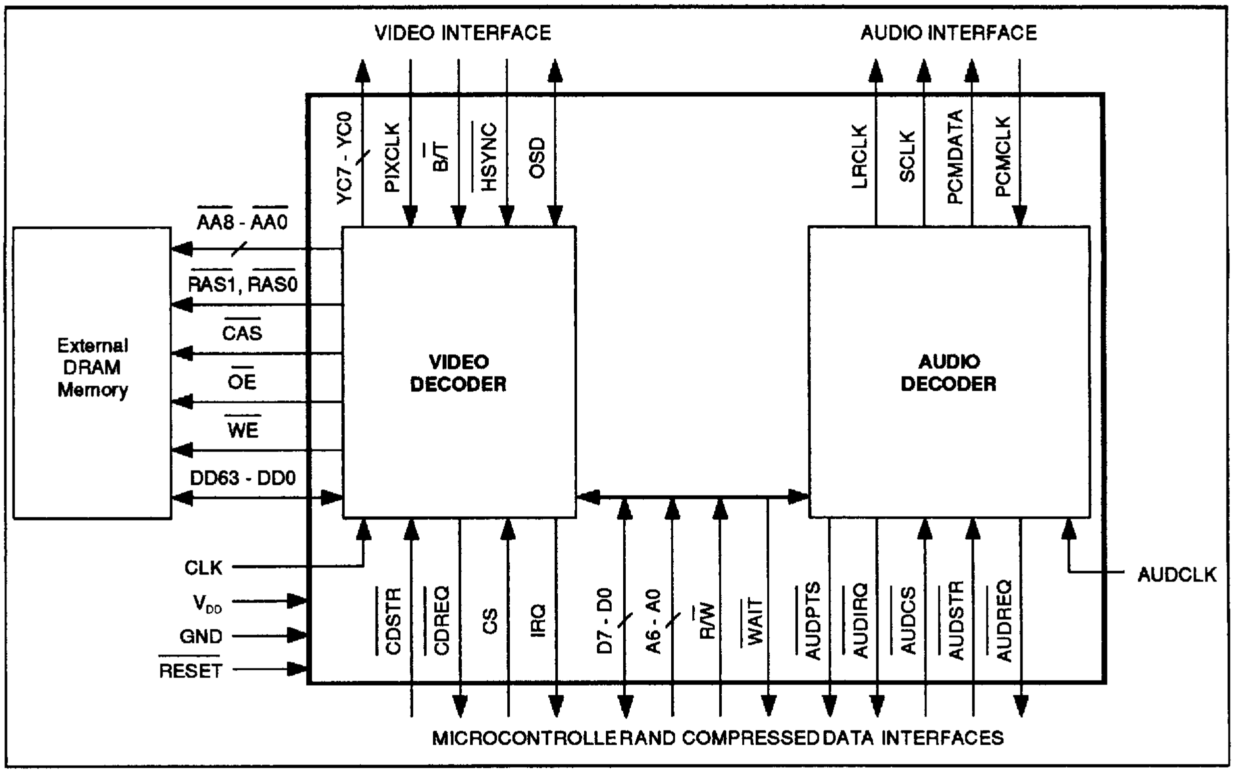


Figure 2 : Audio Decoder Block Diagram

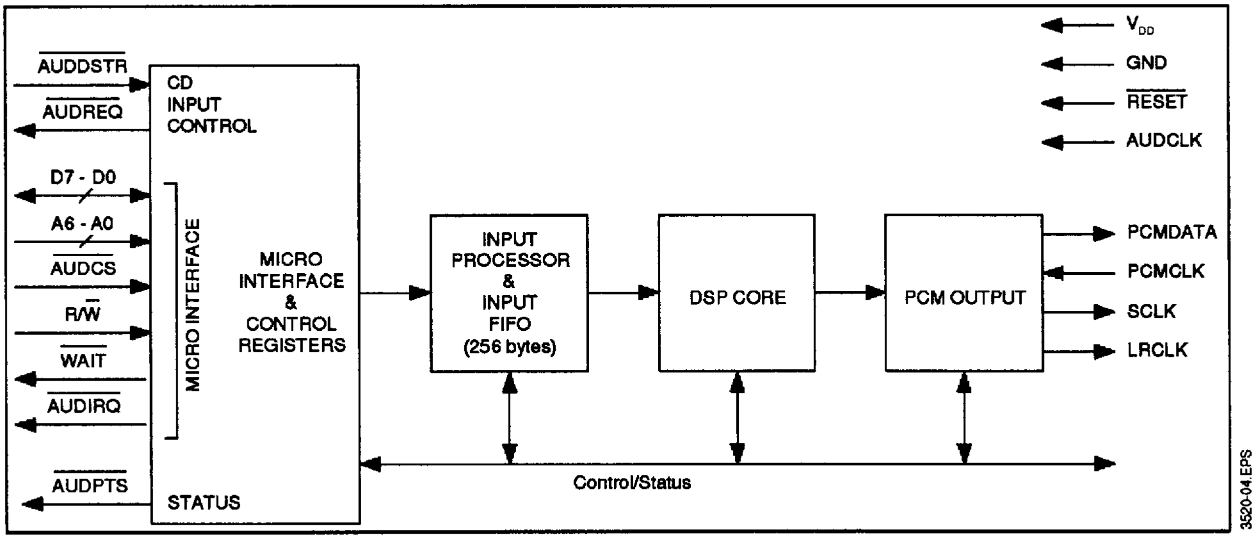
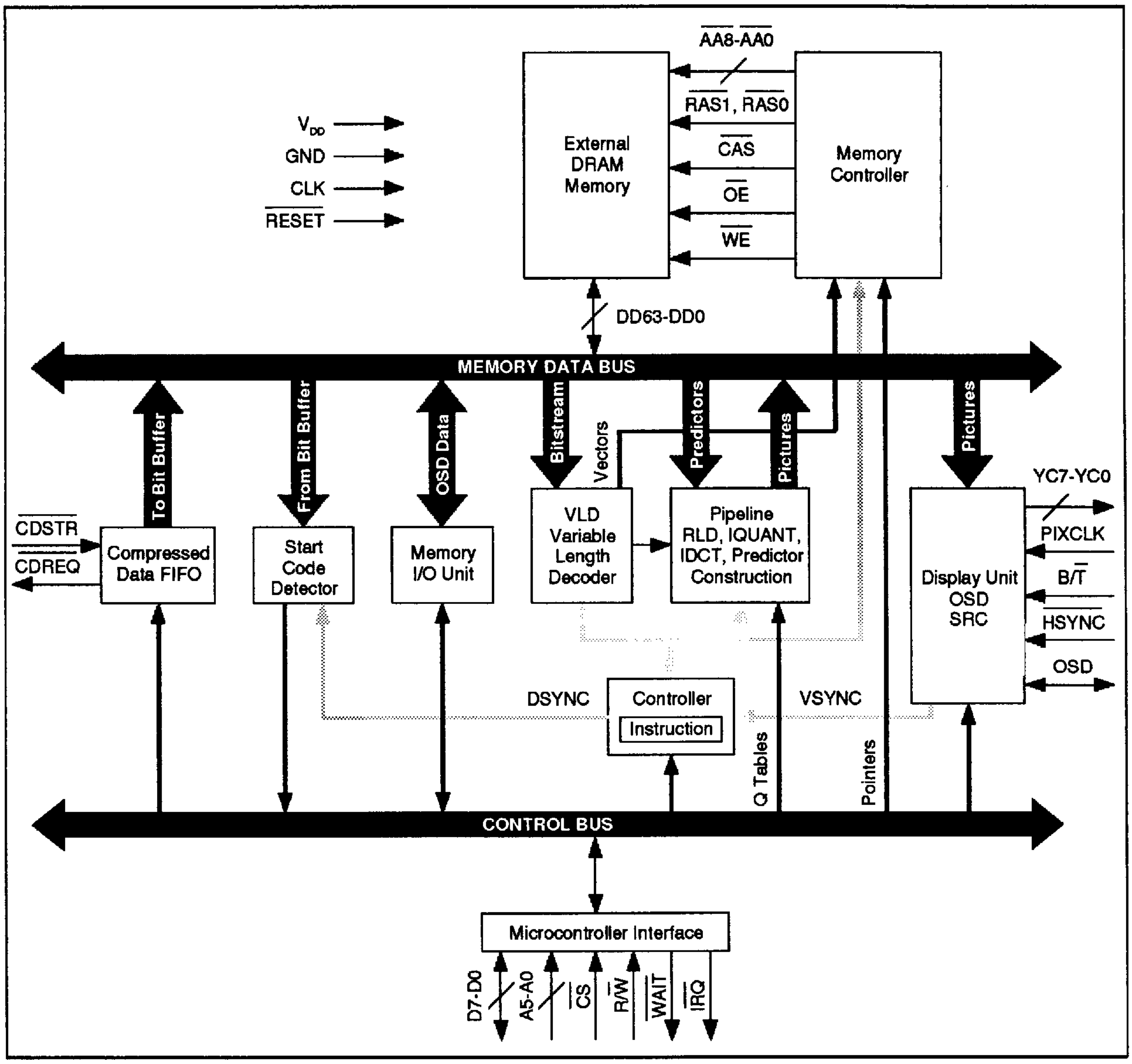


Figure 3 : Video Decoder Block Diagram



3520-03.EPS

III - FUNCTIONAL DESCRIPTION

III.1 - STi3520 Architecture

A functional block diagram of the STi3520 is given in Figure 1. The four external interfaces to the microcontroller, DRAM memory, display and audio DACs are also shown. Together with a minimum of 8 or 16 Mbitsof DRAM, a microcontroller, and some video and audio post processing, a complete audio/video decoder system can be constructed.

The video decoding section is functionally equivalent to the STi3500A video decoder, but with an enhanced on-screen display generator and a DRAM interface capable of supporting synchronous and hyper-page-mode DRAMs.

The audio decoding section is functionally equivalent to the STi4500/4510 audio decoder.

The microcontroller interface has an 8-bit data bus and a 7-bit address bus. This access port has two functions :

- to pass the compressed data to the audio and video decoders,
- to enable control of the STi3520 by providing interrupts and a path for accessing internal registers.

III.2 - Video Decoder (see Figure 3)

The DRAM interface includes all of the signals necessary for control of the memory. Refresh is handled automatically by the video decoder. The memory is used to hold the bit buffer, store decoded pictures, and provide the display buffer. It also holds the user-defined on-screen display (OSD) bitmap and can be used by the microcontroller for private storage of data. For thedecoding of MP@ML sequences, 16 Mbits of memory are required. An 8-Mbit mode is also available for the decoding of less demanding sequences.

The video interface outputs digital video in 8-bit serial C_BYC_R formatunder the control of an external clock and synchronization signals.

During the process of decoding, there are four concurrent activities :

- buffering of the incoming bitstream,
- searching for start codes in the bitstream,
- decoding of a picture,
- display of a picture.

For each of these processes, the microcontroller must set up parameters and monitor events communicated by interrupts. The main features of each of these processes are summarized below.

III.2.1 - Bitstream Buffering

The STi3520 performs the video bitstream buffering needed by the decoder. The size of this buffer, which is located in the DRAM memory, is set up by the user. The bitstream is input through the 8-bit microcontroller data bus. The writing process is asynchronous to all other processes in the video decoder. The bitstream data passes through a 1kbit internal FIFO (the compressed data FIFO) before being transferredin packetsto the bitstream buffer through the memory data bus. An output signal (and associated interrupt) indicate when this FIFO is full, enabling the use of DMA for bitstream input.

The maximum continuous bitstream input rate is application-dependent. A rate of 15Mbit/s (where 1Mbit/s = 10^6bit/s) is possible when decoding MPEG-2 MP@ML bitstreams. The maximum burst rate, for up to 1kbit bursts, is 228Mbit/s.

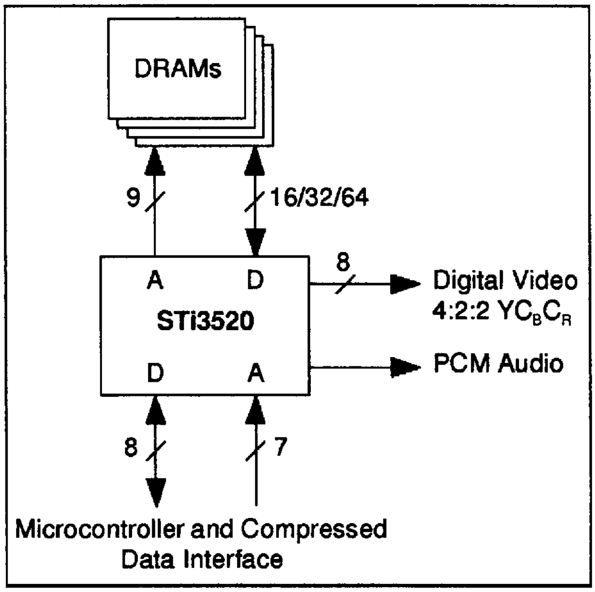
The video decoder only accepts MPEG video bitstreams ; audio and systems data must not be input.

III.2.2 - Start Code Search

The video decoder is able to decode in its entirety a video bitstream from the slice layer downwards. The user mustdecode the higher layers (i.e. picture and upwards) in order to extract the information needed for decoding and appropriately set up the video decoder registers and quantization tables. Since the header information is byte-aligned and requires minimal interpretation, this task represents only a small load on the microcontroller.

The start code detector parses the bitstreamstored in the bit buffer and locatesstart codes corresponding to picture layer and above. When one of these

Figure 4 : Decoder System



3520-05.EPS

start codes has been found, the start code detector stops and raises an interrupt. The microcontroller is then able to read the header data following the start code. The start code detector starts automatically whenever the decoding of a new picture starts and on user command. In normal operation, start code parsing is performed one picture in advance of decoding.

III.2.3 - Decoding

The video decoder is a picture decoder ; it decodes a whole picture and then stops until instructed to decode the next picture present in the video bitstream.

Normally, the decoding of a new picture commences in response to the start of the displaying of a new picture. The registers whose contents can change from picture to picture are double-banked and are updated automatically when decoding starts. The bitstream is read from the bit buffer into the variable-length code decoder (VLD), and picture reconstruction can commence. Any predictors required are fetched from the appropriate area of the external memory, and the reconstructed picture is written back into the area of this memory assigned to the decoded picture.

While a picture is being decoded the start code detector is used to locate the start of next picture header, which the microcontroller then reads in order to set up the double-banked registers for the decoding of the next picture.

III.2.4 - Display

The video decoder is optimized for use with an interlaced display. However, it can also be programmed to produce a non-interlaced (line-sequential) output. The standard video clock rate is 27MHz, which corresponds to a pel rate of 13.5MHz. The active video data output format is compatible with ITU-R 656 ; 00 and FF codes are never generated.

In order to match the horizontal size of the decoded picture to the display line length, an 8-tap upsampling filter, or sample-rate converter (SRC), is provided for both luminance and chrominance. A 2-tap vertical filter is provided for reconstruction of chrominance samples for 4:2:2 output, and for vertical luminance interpolation when displaying half-resolution pictures. The vertical filter includes a data delay line of length 720.

III.2.5 - Enhanced OSD

An on-screen display (OSD) function allows the user to define a bitmap for each field which can be superimposed on the decoded picture output. An OSD bitmap is defined as a set of rectangular regions of programmable position and size, each of which has a unique palette.

OSD data is written into memory areas assigned for this purpose by the user. Reading and writing to and from the memory through the microcontroller interface can take place at any time. A block move feature allows OSD data to be moved from one part of memory to another without microcontroller intervention.

There are two OSD modes, STi3500A compatibility mode, and enhanced mode. In the first mode, each OSD region can use in its bitmap four colours selected from a palette of 4095 colours. The 4096th "colour" is transparency. In the enhanced mode, each region can use 16 colours chosen from a 16384 colour palette. In addition, each region has a "blending factor" defined to enable the mixing of OSD with picture data.

In enhanced mode, memory management by the user is more flexible since OSD regions can be organized in a linked list structure; they do not have to be contiguous in memory.

III.2.6 - Enhanced Memory Interface (only in STi3520 cut 2.0)

The memory interface has been upgraded to allow support of a 1M x 16 synchronous DRAM or a 512k x 32 hyper-page-mode DRAM.

III.3 - Audio Decoder (see Figure 2)

The audio decoder has 4 principal blocks :

- host interface and control registers : the block implements the 8-bit interface to the host processor. All control registers are accessible through this block.
- input processor: the block is responsible for the parsing of the bitstream at the packet level, implementation of the synchronizing algorithms, decoding of time stamps, and the tagging of the audio bitstream with the appropriate PTS before storage in the DRAM. There is an internal 256-byte FIFO buffer.
- DSP core : this block performs bitstream decoding and synthesis subband filter execution, according to layers I and II of the MPEG algorithm.
- PCM output : this block organises the PCM audio output into the required serial format, and generates all of the D-A converter control signals.

The interconnection of these blocks, and all external interfaces, are shown in Figure 1.

Input Formats Accepted

The STi3520 is able to accept the following 3 types of audio bitstream :

- MPEG audio elementary stream, as defined by ISO/IEC 11172-3, Layers I and II,
- ISO/IEC 11172-1 packets with syntax as shown below :

```
                                                   n° of bits
packet() {
        packet_start_code_prefix                       24
        stream_id                                       8
        packet_length                                  16
        if (stream_id != private_stream_2) {
          while (nextbits() == '1111 1111')
            stuffing_byte                               8
          if (nextbits() == ' 01') {
            '01'                                        2
            STD_buffer_scale                            1
            STD_buffer_size                            13
          }
          if (nextbits() == '0010') {
            '0010'                                      4
            presentation_time_stamp[32..30]             3
            marker_bit                                  1
            presentation_time_stamp[29..15]            15
            marker_bit                                  1
            presentation_time_stamp[14..0]             15
            marker_bit                                  1
          }
          else if (nextbits() == '0011') {
            '0011'                                      4
            presentation_time_stamp[32..30]             3
            marker_bit                                  1
            presentation_time_stamp[29..15]            15
            marker_bit                                  1
            presentation_time_stamp[14..0]             15
            marker_bit                                  1
            '0001'                                      4
            decoding_time_stamp[32..30]                 3
            marker_bit                                  1
            decoding_time_stamp[29..15]                15
            marker_bit                                  1
            decoding_time_stamp[14..0]                 15
            marker_bit                                  1
          }
          else
            '0000 1111'                                 8
        }
        for (i=0; i N; i++) {
          packet_data_byte                              8
        }
}
```

The outlined parts indicate the information which is extracted by the STi3520. The other items of packet body information are ignored.

- uncompressed PCM data (for bypass mode).

IV - DOCUMENTATION CONVENTIONS

In this data sheet, the following conventions are used when documenting the functions of signals:

- I/O signals can either be active high or active low. The former have names without an overbar (i.e. SIGNAL), the latter have an overbar (i.e. $\overline{\text{SIGNAL}}$). Where a signal has two different and mutually exclusive actions, a dual name is used (e.g. $\overline{\text{COME}}$/GO).
- Internal signals and variable names (e.g. bits in registers) are always documented as active high.
- When the condition indicated by the name of the signal or variable is**true**, the signal or variable is said to be true, **asserted** or to have the value 1.
- When the condition indicated by the name of the signal or variable is not true, the signal or variable is said to be **false**, **de-asserted** or to have the value **0**.
- When an active high signal is true or asserted, the logic voltage level is high.
- When an active low signal is true or asserted, the logic voltage level is low.
- When an internal signal or variable is **set**, it has the value 1. A bit is never "set to 0", but "reset to 0".
- When an internal signal or variable is**reset**, it has the value 0.
- Hexadecimal numbers are indicated by appending an "h", e.g. A70h.

V - SUMMARY SPECIFICATION

V.1 - Video Decoder

Bitstreams Accepted

MPEG-1 video (ISO/IEC 11172-2).

MPEG-2 video (ISO/IEC 13818-2).

MPEG-2 Profiles/Levels Supported

Main Profile @ Main Level (MP@ML).

Main Profile @ Low Level (MP @ LL).

Simple Profile @ Main Level (SP @ ML).

Maximum Picture Size

Width : 4080.

Number of macroblocks : 16,383.

Motion Vector Range

MPEG-1 : -1024 to 1023 (full pel), -512 to 511.5 (half pel) horizontal and vertical.

MPEG-2 : -1024 to 1023.5 horizontal and vertical.

(vertical range must be reduced in 8-Mbit memory mode with certain picture sizes)

Compressed Data Input

8-bit asynchronous data port (shared with microcontroller interface).

Peak input rate : 28.5Mbyte/s (228Mbit/s).

Maximum sustained input rate (with 55MHz primary clock) : 100Mbit/s in 16-Mbit memory mode, 60Mbit/s in 8-Mbit memory mode (in practice the sustained rate will be constrained by the memory bandwidth required for real-time decoding).

Microcontroller Interface

8-bit data port with fixed length "WAIT" pulse acknowledge.

Single interrupt request pin.

DRAM Interface

External DRAM used for storage of picture buffers, bit buffer and on-screen display definitions.

16, 32 or 64 bit data bus.

Refresh handled by decoder.

DRAM directly accessible through microcontroller interface.

Configurations supported :
- DRAM: 4 Mbits (1 bank), 8 Mbits (1 bank), 16 Mbits (1 bank), 20 Mbits (2 banks), 32 Mbits (2 banks),
- SDRAM : 1M x 16 (only in STi3520 cut 2.0).

Hyper-page-mode DRAM : 512k x 32 (only in STi3520 cut 2.0).

Start Code Detection

Automatic detection of start codes of picture layer and above to enable microcontroller to access header data.

Counters provided for time-stamp tracking.

Decoding Pipeline

Instruction register set up each picture defines pipeline operation.

Double-buffered quantization matrices enable loading of new tables concurrently with decoding.

Error Concealment

Automatic concealment of errors detected by VLD and decoding pipeline by macroblock copy.

Video Output

8-bit 27MHz multiplexed C_BYC_R port, compatible with ITU-R 601 and 656 (00 and FF values never output).

External pel clock and horizontal/vertical synchronization required.

Interlaced or line-sequential output.

3:2 pulldown operation supported.

Programmable horizontal up-sampling by 8-tap filter.

Vertical chroma reconstruction or luma filtering by 2-tap filter including 720-sample delay line.

Pan & Scan Vectors

Horizontal : Maximum vector size : 512 pels, resolution : 1/8 pel.

Vertical : Maximum vector size : 508 lines, resolution : 4 lines.

On-Screen Display (OSD)

Bitmap separately definable for each field can be superimposed on final picture output.

OSD defined as rectangular regions, each with unique palette defining 4 or 16 colours (including transparency). Each region has a blending factor, which can be selectively applied to each colour in the palette.

Number of regions limited by memory space allocated to OSD. Regions definitions can be organized as a linked list (only in STi3520 cut 2.0).

Block move facility available for reduction of microcontroller loading.

Primary Clock

60MHz maximum.

V.2 - Audio Decoder

Bitstreams Accepted

ISO/IEC 11172-3 audio elementary stream.

ISO/IEC 11172-1 packets.

Audio PCM data (for decoder bypass).

Performance

ISO/IEC 11172-3 Layers I & II.

All MPEG input bitrates supported with sampling rates of 32, 44.1 and 48kHz, free format at 32 & 48kHz sampling rates.

Decodes in single channel, dual channel, stereo, or joint stereo modes.

System Clock

24MHz nominal.

Microcontroller Interface

8-bit interface with "WAIT" signal handshake. Interrupt request signal. Dedicated control inputs for "play" and "mute".

Compressed Data Input

Bit or byte-mode input.

Burst rate up to 20Mbit/s.

PCM Output

16 or 18-pit PCM output.

I^2S and other popular formats supported.

Support for A/V Sync

PTS extraction from MPEG packet layer.

Error Concealment

Automatic error concealment on CRC or synchronization error detection.

VI - MICROCONTROLLER INTERFACE AND COMPRESSED DATA INPUT

VI.1 - Interface Signals

The combined microcontroller/compressed data interface consists of the following signals :

Name	Type	Function
D7-D0	I/O	Data Port
A6-A0	I	Address
R/$\overline{\text{W}}$	I	Read/Write Selection
$\overline{\text{CS}}$	I	Read/Write Strobe (video)
$\overline{\text{AUDCS}}$	I	Read/Write Strobe (audio)
$\overline{\text{WAIT}}$	O	Acknowledge
$\overline{\text{CDSTR}}$	I	CD Write Strobe (video)
$\overline{\text{CDREQ}}$	O	CD Request (video)
$\overline{\text{AUDSTR}}$	I	CD Write Strobe (audio)
$\overline{\text{AUDREQ}}$	O	CD Request (audio)
$\overline{\text{IRQ}}$	O	Interrupt Request (video)
$\overline{\text{AUDIRQ}}$	O	Interrupt Request (audio)

Input of compressed data and read/write access to STi3520 control registers share the same 8-bit data port, D7-D0. The signal $\overline{\text{CS}}$ requests a video register access cycle, while $\overline{\text{AUDCS}}$ requests an audio register access cycle. R/$\overline{\text{W}}$ selects the direction of transfer for a register access. Register addresses are selected by the 7 address bits A6-A0. The signal $\overline{\text{CDSTR}}$ requests a video compressed data write cycle, while $\overline{\text{AUDSTR}}$ requests an audio compressed data write cycle.

The multiplexing of compressed data and control bytes onto D7-D0, as well as arbitration of compressed data and register read/write cycles, must be performed externally.

$\overline{\text{WAIT}}$ (a 3-state output) is the acknowledge for register access cycles triggered by $\overline{\text{CS}}$ or $\overline{\text{AUDCS}}$. $\overline{\text{CDREQ}}$ (or $\overline{\text{AUDREQ}}$) indicate that the STi3520 is available for the input of compressed data, and together with $\overline{\text{CDSTR}}$ (or $\overline{\text{AUDSTR}}$), make a DMA handshake.

Interrupt requests from the video decoder are signalled by $\overline{\text{IRQ}}$, and interrupt requests from the audio decoder are signalled by $\overline{\text{AUDIRQ}}$.

VI.2 - Register Access (video)

The 6 address bits, A5-A0, select one of the 64 one-byte video decoder register locations. The function of each register is detailed in section XIV, "Video Registers". Some are read only, some write only, and some read/write. The signal R/$\overline{\text{W}}$ defines whether the register access cycle is read or a write.

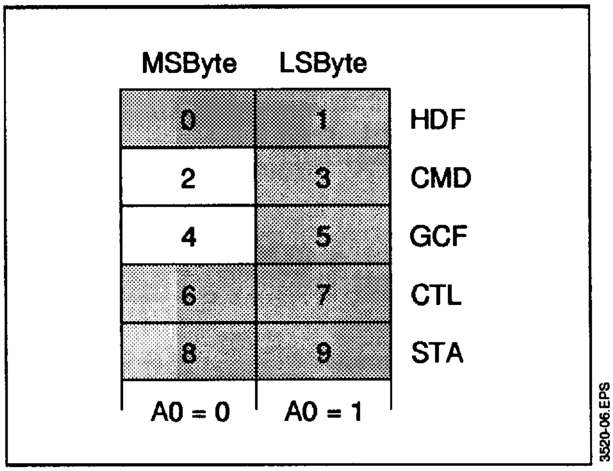
Address bit A6 is not taken into account when accessing one of the video registers.

A cycle is defined by the assertion of signal $\overline{\text{CS}}$. In response to this the signal $\overline{\text{WAIT}}$ is always asserted. When $\overline{\text{WAIT}}$ is de-asserted, indicating the completion of the read or write operation, $\overline{\text{CS}}$ can return to its high state and the cycle ends. The time from the assertion of $\overline{\text{CS}}$ and the de-assertion of $\overline{\text{WAIT}}$ has a fixed maximum value. During a register read/write cycle (i.e. while $\overline{\text{CS}}$ is low), the signal $\overline{\text{CDSTR}}$ must be high, since compressed data write and register read/write cycles are mutually exclusive. The timing diagrams of section XIII.3, "Microcontroller Interface (video)" define the timing constraints. The minimum time between two successive single-byte read or write cycles is 70ns (equal to the sum of the values of t_{SLWH} and t_{SHSL} defined in section XIII.3, "Microcontroller Interface (video)"). The signal $\overline{\text{WAIT}}$ is in its high impedance state when a read/write cycle is not in progress (i.e. while $\overline{\text{CS}}$ is high).

The internal registers are organised in 16-bit units, as shown in Figure 5. Some require both byte addresses, some only one. A 16-bit register is mapped into two consecutive addresses.

In general register accesses require two cycles, with the most significant byte being accessed first. The exceptions are noted next page.

Figure 5 : Register Addressing



MSByte	LSByte	
0	1	HDF
2	3	CMD
4	5	GCF
6	7	CTL
8	9	STA
A0 = 0	A0 = 1	

VI.2.1 - Writing to a Register

To write to a 16-bit register, the most significant byte (A0 = 0) is written first. This byte is stored in an internal temporary register during the first cycle and is not written to the destination register. In the second cycle, the least significant byte (A0 = 1) is written. In this operation, both bytes, the new byte and the one stored in the first cycle, are written to the destination register. So for example to write to the CTL register, a write must first be made to address 6, followed by a write to address 7.

It is not possible to write to the most significant byte alone, but it is possible in certain cases to write to the least significant byte only. These are where the most significant byte is not defined, or where it is a read-only register. For example a single cycle write to the one-byte MWF register (address 11) is possible because the register at the preceding even address, MRF (address 10), is a read-only register.

The two parts of a register write operation must never be interrupted by another register write operation, since the contents of the internal temporary register could be overwritten before it is used.

Write cycle timing is given in Figure 71.

VI.2.2 - Reading a Register

To read a 16-bit register, the most significant byte (A0 = 0) is read first. In this cycle the most significant byte is available at the data port, and the least significant byte of the register is stored in an internal temporary register. (This is not the same physical register as that used to store the most significant byte when writing). In the second cycle (A0 = 1), the contents of the temporary register are sent to the data port ; the register addressed is not read again.

It is thus possible to read only the most significant byte of a register. In this case the byte loaded into the temporary register is not read. It is not possible to read the least significant byte alone ; the preceding most significant byte must be read first in order to load the least significant byte into the temporary register.

The two parts of a register read operation must never be interrupted by another register read operation, since the contents of the temporary register will be overwritten before it is read. It is possible to mix the cycles of the read from one register and the write to another. Also, any register read or write cycles can be mixed freely with compressed data write cycles.

Read cycle timing is given in Figure 81.

When it is required to read two consecutive bytes, it is possible to combine these into a single extended cycle. The cycle starts normally with the reading of the most significant (even) byte. When the de-assertion of WAIT indicates that the first byte is available, the least significant address bit can be changed. This causes the contents of the temporary register to be switched to the output. WAIT will remain high during the read of the second byte. The timing of this operation is given in Figure 70.

VI.3 - Register Access (audio)

The address bits, A6-A0, select one of the 128 one-byte control register locations. The function of each register is detailed in section XIV.2, "Register Descriptions". Some are read/write, some read-only, and some write-only. The signal R/W defines whether the register access is a read or a write (high for read, low for write).

A cycle is defined by the assertion of signal AUDCS. In response to this the signal WAIT is always asserted. The address, read/write and data inputs must be set up before AUDCS line is activated. If a read cycle is requested, the data lines D7-D0 will be driven by the STi3520. For a write cycle the STi3520 will latch the data placed on the data lines on the rising edge of AUDCS.

When WAIT is de-asserted, indicating the completion of the read or write operation, AUDCS can return to its high state and the cycle ends. The timing of register read/write cycles is given in Figure 78. The minimum time between two successive read or write cycles is 100ns. The signal WAIT is in its high state when a read/write cycle is not in progress (i.e. while AUDCS is high).

During a register read/write cycle (i.e. while AUDCS is low), the compressed data input strobe AUDSTR must be high.

The relative timing constraints of these two signals are given in Figure 80.

VI.4 - Interrupts (video)

The conditions which can cause an interrupt are represented by the bits of the STA register. Any change of state from 0 to 1 of one of these bits (an "event") will cause an interrupt unless it is masked by the corresponding bit of the ITM register being reset. Each event causes a bit of the ITS register to be set, regardless of the state of ITM. Any unmasked bit becoming set in the ITS register causes an interrupt request, indicated by the driving of the open-drain signal IRQ into its asserted state.

The interrupt is acknowledged and its source(s) identified by reading the ITS register. The reading of the most significant byte of this register has the effect of clearing the whole register and thereby removing the interrupt request (i.e. IRQ returns to its undriven state). The least significant byte can be read from the temporary register in a second cycle. The timing of the removal of the interrupt request is given in Figure 81.

Before reading ITS, all interruptsshould be masked by clearing the ITM register. This prevents an event occurring during the read cycle from generating a new interrupt. When ITM is restored, events which occurred while the mask was zeroed will now generate an second interrupt request.

VI.5 - Interrupts (audio)

The conditions that can cause an interrupt are listed in the table below :

N°	Condition Signalled
14	First bit of new frame at PCM output
13	Input FIFO full
12	Input FIFO level = FIFO_THRES
11	Not used
10	De-emphasis changed
9	Sampling frequency changed
8	PCM output buffer underflow
7	Ancillary data register full
6	Not used
5	CRC error detected
4	Not used
3	Not used
2	Valid PTS registered
1	Valid header registered
0	Change in synchronization status

An interrupt is enabled by setting the corresponding bit in the interrupt mask register, INTR_EN. An interrupt is signalled externally by a 100ns pulse appearing on AUDIRQ. At the same time one of the bits of the interrupt register, INTR, becomes set. The interrupt source may then be identified by reading INTR.

The most significant byte (bits 8-14), and bits 3-5 of the least significant byte of INTR can be independently cleared by reading. Bits 0-2 and 7 are cleared by a different method, as explained in the description of the INTR register in section XIV.2, "Register Descriptions". If the condition giving rise to the interrupt remains, a new interrupt will be generated.

The INTR and INTR_EN registers are cleared on reset (assertion of RESET pin or writing to RESET register), or restart (writing to RESTART register).

VI.6 - Compressed Data Input (video)

The video compressed data input must be either an MPEG-1 or an MPEG-2 video elementary stream.

Compressed data input is placed on the shared data bus, D7 - D0, and is strobed in on the rising edge of the signal CDSTR. During a compressed write cycle the signal CS must be high, since compressed data write and register read/write cycles are mutually exclusive. Compressed data input bytes are written into the 1024-bit Compressed Data (CD) FIFO (see **Figure 1**). Data from this FIFO are transferred in 512-bit bursts to the bit buffer area of the external memory, whose base address is specified by the BBG register. The rate at which these transfers can occur is governed by the number of higher priority requests waiting for service from the memory controller. The de-assertion of the signal CDREQ indicates that the CD FIFO is full, but that 3 more bytes can be written. No further compressed data bytes must be written while CDREQ remains de-asserted, since data will be lost.

The timing diagram in Figure 80, defines the timing constraints. The minimum time between two successive write cycles is 35ns (giving an upper limit on port input rate of roughly 228Mbit/s).

Compressed data write cycles can be combined in any sequence with register read/write cycles.

Start codes in the compressed data stream must be byte-aligned.

The average rate of compressed data input over a picture period is one of the decoder performance constraints. It is related to picture size, prediction modes and primary clock frequency. This rate can be exceeded momentarily, provided that the number of bits written during a picture period does not exceed the number used in the performance calculation.

There are three ways of controlling the input of compressed data bytes :

- handshake every byte,
- byte input without handshake,
- burst input without handshake.

In all cases, the average compressed data input rate cannot exceed that used in the STi3520 video decoder performance calculation.

VI.6.1 - Handshake every Byte

In this mode, the signal is CDREQ is checked after the writing of every byte. The maximum rate of data transfer is one byte every 55ns, assuming that CDSTR can be asserted again immediately CDREQ becomes high. This is equivalent to a maximum instantaneous rate of roughly 145Mbit/s. If this is greater than the relevant sustained rate given in the next section, some cycles will be delayed due to the CD FIFO being full.

VI.6.2 - Byte Input without Handshake

Provided that the input rate is kept below given limits, it can be guaranteed that the CD FIFO will never become full, thus making unnecessary the checking of CDREQ. The limits are:

Maximum sustained rate (normal memory mode) :

primary clock frequency x 1.8

With a primary clock of 55MHz, this rate is 100Mbit/s.

Maximum sustained rate (8-Mbit memory mode) :

primary clock frequency x 1.1

With a primary clock of 55MHz, this rate is 60Mbit/s.

Maximum sustained rate (normal memory mode, display only) :

primary clock frequency x 2.3

With a primary clock of 55MHz, this rate is 128Mbit/s.

Maximum sustained rate (8-Mbit memory mode, display only) :

primary clock frequency x 1.4

With a primary clock of 55MHz, this rate is 77Mbit/s.

Maximum sustained rate (normal memory mode, no decoding or display) :

primary clock frequency x 13

With a primary clock of 55MHz, this rate limited by the maximum CD port input rate of 228Mbit/s.

Maximum sustained rate (8-Mbit memory mode, no decoding or display) :

primary clock frequency x 8

With a primary clock of 55MHz, this rate limited by the maximum CD port input rate of 228Mbit/s.

VI.6.3 - Burst Input without Handshake

Data may be entered in bursts of up to 1024 bits at the maximum rate (228Mbit/s). The minimum time between bursts is 350 primary clock cycles (6.4µs at 55MHz) in 16-Mbit memory mode, and 590 primary clock cycles (10.7µs) in 8-Mbit memory mode. These are the maximum times which must be allowed for the CD FIFO to be emptied into the bit buffer.

VI.7 - Bit Buffer Control (video)

The amount of data in the bit buffer is available (in units of 2 kbits) by reading the BBL register. When this level is greater than or equal to the value loaded into the BBT register (also defined in units of 2kbits), the status bit STA.BBF becomes true. This can be used to generate a "bit buffer nearly full" interrupt. When the bit buffer contains no data, the status bit STA.BBE becomes true. This can be used to generate a "bit buffer empty" interrupt.

When the CD FIFO is full, the status bit STA.BFF is true. This bit is thus equivalent to the signal CDREQ.

If the bit CTL.PBO is set, then transfer of data from the CD FIFO to the bit buffer is prevented if the bit buffer level is at or above the level defined in the BBT register. If BBT is set to a value equal to the size of the bit buffer, then this automatic mechanism will ensure than overflow never occurs.

If it is required to complete the decoding of a sequence before the arrival of another, for example if the sequence contains a single picture, 63 zero bytes must be written at the end of the first sequence, to ensure that the last packet is always sent from the CD FIFO to the bit buffer.

If in the application it is not possible to do this, a flush can be initiated from the microcontroller by entering a test mode. An Application Note is available on this topic.

VI.8 - Compressed Data Input (audio)

The following 3 types of bitstream may be input :

- MPEG audio elementary stream, as defined by ISO/IEC 11172-3, Layers I and II,
- ISO/IEC 11172-1 packets,
- uncompressed PCM data.

The input mode is determined by the programming of register STR_SEL (see section XIV.2, "Register Descriptions").

There are two ways of inputting compressed data :
- strobed in by AUDSTR. (DMA mode),
- writing to register DATAIN, strobed by AUDCS.

If the signal REQ is asserted (low), then another data byte can be input. REQ goes high when the input buffer is full. ONE further byte can safely be entered after this. Input timing is given in Figure 78 and Figure 83

The signal REQ is high for the duration of the reset and restart processes (see section XI.2.2, "Initialization of the Audio Decoder" and Section XI.2.3, "Play and Mute").

The relative timing constraints between AUDCS and AUDSTR specified in section VI.3, "Register Access (audio)", must be respected.

Data may be input in bursts at rates 2.5Mbyte/s.

VI.9 - Input Buffer Level Control (audio)

Input data is entered into a 256-byte FIFO store before interpretation by the parser.

Two interrupts are available for the monitoring of FIFO level. Interrupt 13 indicates that the input FIFO is full, while interrupt 12 indicates that the input FIFO level is equal to the value loaded into register FIFO_THRES. Interrupt 12 is generated whenever the level crosses the threshold. It is recommended to use REQ to control input data flow whenever possible.

All buffers are flushed after a reset or restart.

VI.10 - Audio Decoder Latency

Before being able to output audio samples, the STi3520 must first be synchronized to the audio bitstream. After this the decoder requires a certain amount of data, equal to roughly 1/12 of a frame, before the output samples can be computed. The time required to compute these samples is about 2ms.

The total delay is equal to the sum of the synchronization time, the time required to input the data, and the decoding time. The synchronization time is time required to skip zero, one, two or three frames (see section XI.2.5, "Bitstream Synchronization"). This depends on the Layer and sampling frequency, and has a maximum value of 3 x 36 = 108ms (with 36 = frame duration for layer II at 32kHz). Similarly, the longest data input time is 36/12 = 3ms. Once the STi3520 is synchronized, the longest latency in low latency mode is 3 + 2 = 5ms.

VII - EXTERNAL MEMORY

VII.1 - Standard DRAM

VII.1.1 - Memory Interface

The memory interface consists of the following signals :

Name	Function
DD63-DD0	Bidirectional Data Port
AA8-AA0	Address
RAS0	Row Address Strobe for Bank 0
RAS1	Row Address Strobe for Bank 1
CAS	Column Address Strobe
OE	Output Enable
WE	Write Enable

These signals correspond directly the DRAM signals of the same names. Note that the address output signals are inverted.

Memories with multiplexed 9-bit row and 9-bit column addresses must be used, except in the second bank when in 20-Mbit mode, in which memories with 8-bit row and column addresses are required.

The memory interface is disabled when bit CTL.EDI is reset (the default state). In this state all of the memory interface signals are in their high impedance state. If it is required to keep the RAS and CAS signals high when the interface is disabled in order to reduce DRAM power consumption, they should be pulled up to V_{DD} with 20kΩ resistors.

VII.1.1.1 - Normal (16-Mbit) Mode

In the normal mode all 64 bits of the data bus are connected to the memory.

The three diagrams below illustrate the memory configurations possible in normal mode. Figure 6 shows the configuration in which there is a single bank of 16 Mbits. The second row address strobe (RAS1) signal is not used.

Figure 7 shows the configuration in which an extra 4 Mbits is added in the second bank to give a total memory capacity of 20 Mbits. The memory space of the second bank is contiguous with that of the first. When this mode is used, bit GCF.M20 must always be set. Address signals AA7 to AA0 are used for the second bank. (It is also possible to use the second 4-Mbit bank alone, to give a 4-Mbit configuration).

Figure 8 shows the configuration in which both banks contain 16 Mbits. Bit GCF.M20 must be reset.

Figure 6 : 16-Mbit Memory Configuration

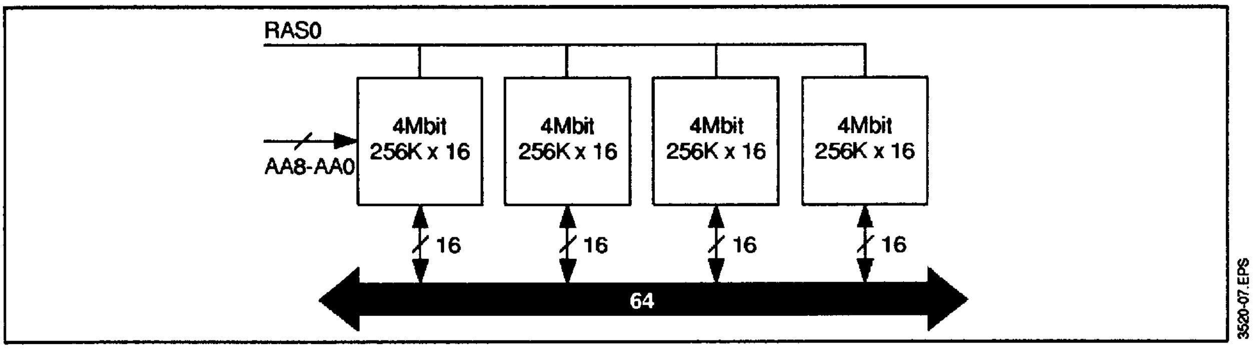


Figure 7 : 20-Mbit Memory Configuration

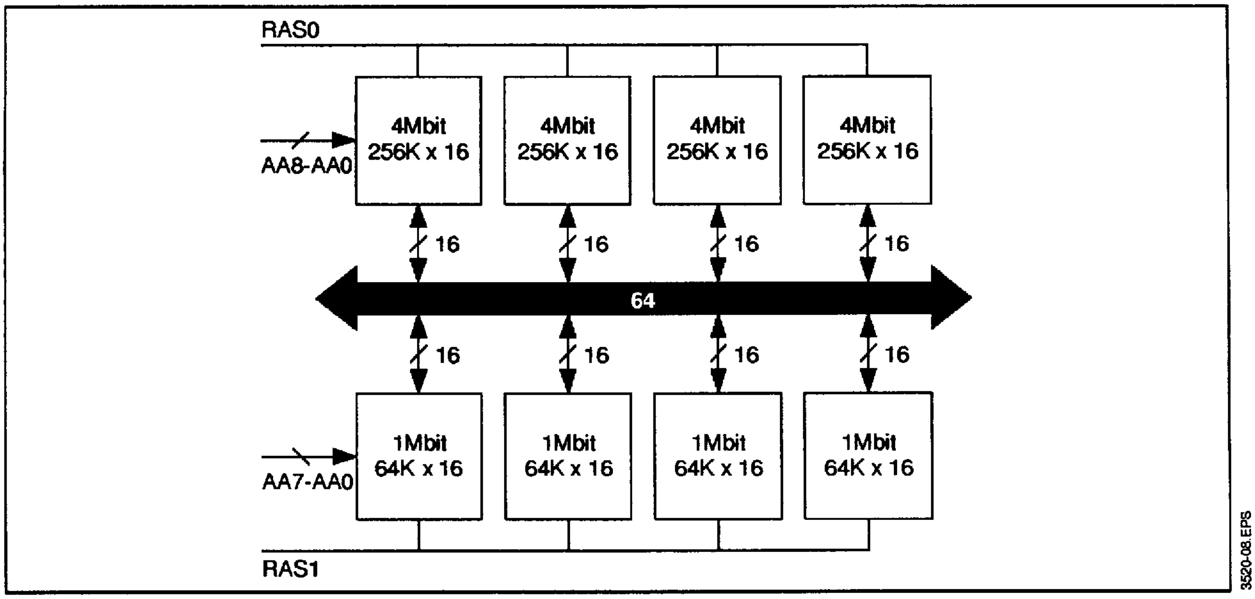
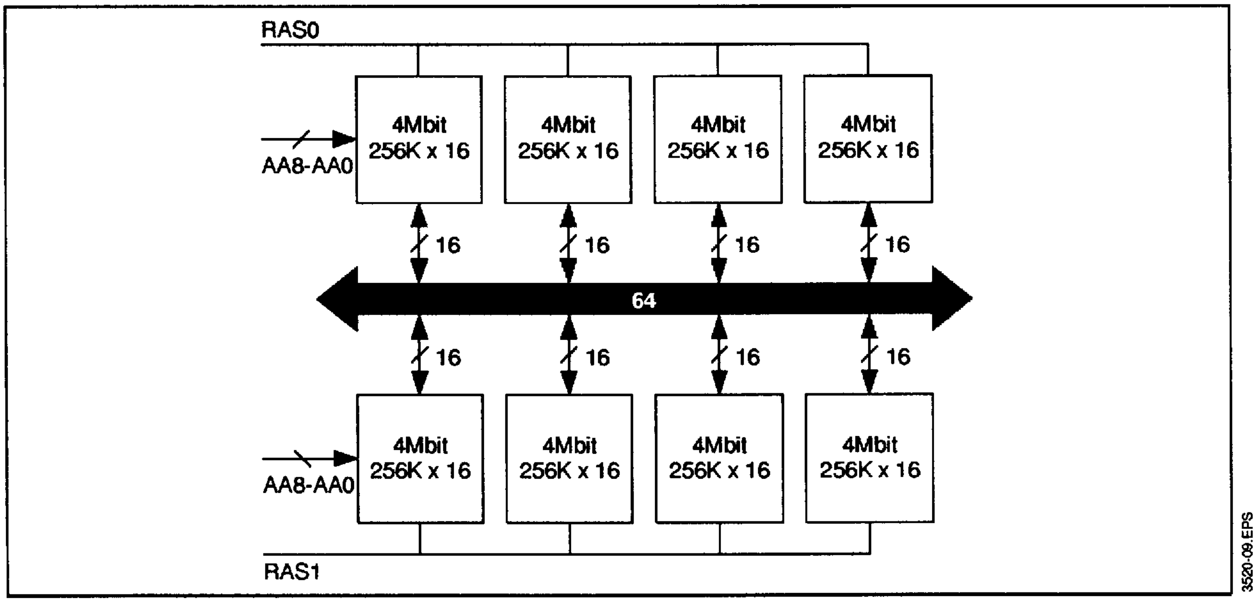


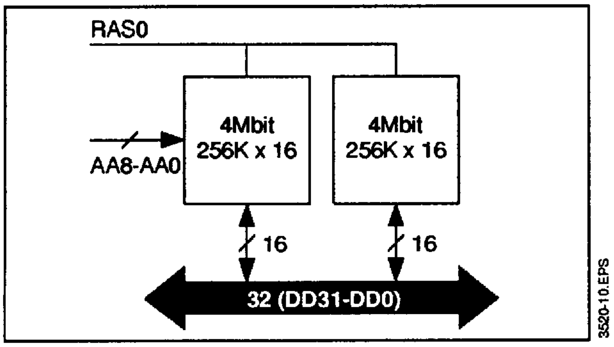
Figure 8 : 32-Mbit Memory Configuration



VII.1.1.2 - 8-Mbit Mode

In 8-Mbit mode only bits DD47 to DD16 are used. Only one memory configuration is possible - a single bank connected as shown in Figure 9. When this mode is used, bit CTL.S8M must always be set.

Figure 9 : 8-Mbit Memory Configuration



VII.1.2 - Timing Requirements

The timing parameters for the memory interface are given in section XIII.2, "DRAM Interface". In most instances, these are dependant on the primary clock frequency, which could enable slower memories to be used when the primary clock frequency is lower than the maximum. With a primary clock of 55MHz, "-80" or faster memories are required.

The maximum permissible memory access times are related to the memory interface parameters as follows :

$$t_{AA} = t_{CAL} - t_{DS}\ (\text{read}) = 3T - \text{TBD ns}$$

$$t_{ACP}{}^{*} = 3T - \text{TBD ns}$$

$$t_{CAC} = t_{CAS} - t_{DS}\ (\text{read}) = 2T - \text{TBD ns}$$

$$t_{OEA} = t_{OCH} - t_{DS}\ (\text{read}) = 3T - \text{TBD ns}$$

$$t_{RAC} = t_{CSH} - t_{DS}\ (\text{read}) = 5T - \text{TBD ns}$$

Where T is the primary clock period.

* Also sometimes referred to as t_{CPA}.

VII.1.3 - Refresh

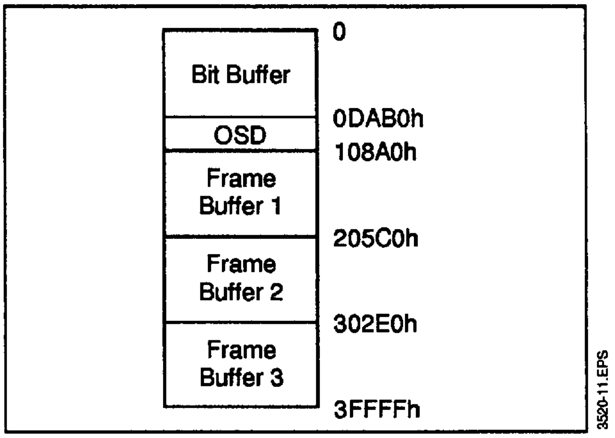
Memory refresh is handled automatically by the STi3520 memory controller by inserting "$\overline{\text{CAS}}$ before $\overline{\text{RAS}}$" refresh cycles. The duration of a refresh cycle is 9 primary clock periods. During these cycles $\overline{\text{RAS0}}$ and $\overline{\text{RAS1}}$ are driven together, thus refreshing a row from each bank. The refresh period is defined, in units of 24 primary clock periods, by the programming of GCF.RFI[6:0]. For example if 512 memory rows must be refreshed every 8ms and the primary clock is 55MHz, GCF.RFI[6:0] must be loaded with :

8ms/512 x 55MHz/24 = 35 (after rounding down)

VII.1.4 - Memory Mapping Examples

The first example (Figure 10) shows the memory map in an application in which the picture size is 720 x 480. With 3 frame buffers each occupying just less than 4 Mbits and an on-screen display (OSD) buffer sufficient to define a bitmap covering the full screen occupying 0.72 Mbits, there is sufficient space for a 3.4-Mbit bit buffer. It is assumed in the OSD space calculation that there is one OSD region per line.

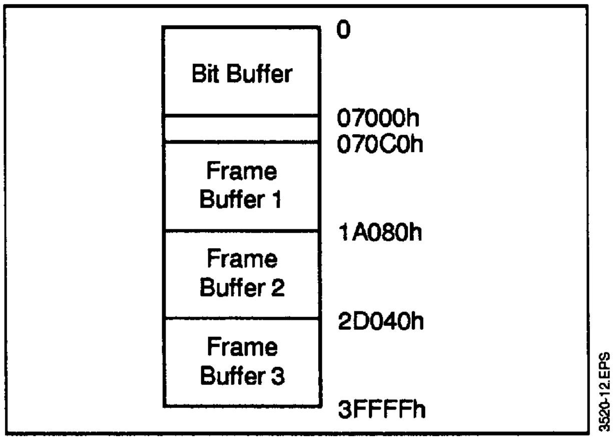
Figure 10 : 720 x 480 Frames in 16 Mbits



The second example (Figure 11) shows the memory map in an application in which the picture size is 720 x 576. With 3 frame buffers each occupying 4.75 Mbits there is sufficient space for a 1.75-Mbit bit buffer. Between the bit buffer and the frame buffers, there are 192 unused words (12288 bits) which can be used for OSD buffer or to enlarge bit buffer.

If a larger bit buffer is required, or if space needs to be made for OSD (0.86 Mbits are needed to fill a 720 x 576 screen), then a 20-Mbit configuration

Figure 11 : 720 x 576 Frames in 16 Mbits



will be necessary (it is possible to extend the bit buffer off-chip if the burst mode of compressed data input is used. An application note is available on this topic).

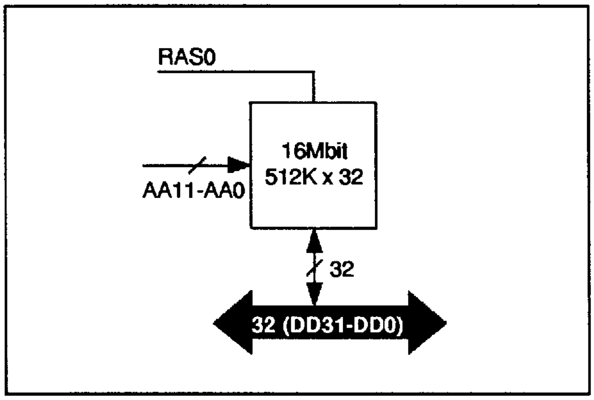
If it is required to store more than 3 frame buffers (for example if overwrite mode is not used), then at least 20 Mbits (for 720 x 480) or 32 Mbits (for 720 x 576) are necessary.

VII.2 - Extended Data Out (EDO) DRAM

Extended data out DRAMs (also known as Hyper page DRAMs) effectively place a latch on the data output. This allows the CAS signal to go high without having to wait for the data out to become valid as in normal DRAM. As soon as the CAS signal goes high precharge for the next column address can begin. This has the net effect of decreasing the page cycle time. The memory interface of the STi3520 takes advantage of this by using a special high frequency CAS signal thus allowing the data bus to operate as a 32-bit bus.

The interface can supports a 16-Mbit configuration as in Figure 12. In this mode only 32 of the 64-bit memory data bus are used. The exact pinout is given in the "Pin List". The EDO memory interface does not use the same control signals (RAS, CAS, WE , OE) as standard DRAM. The control signals use the unused data pins of the memory interface.

Figure 12 : 16-Mbit EDO DRAM Configuration



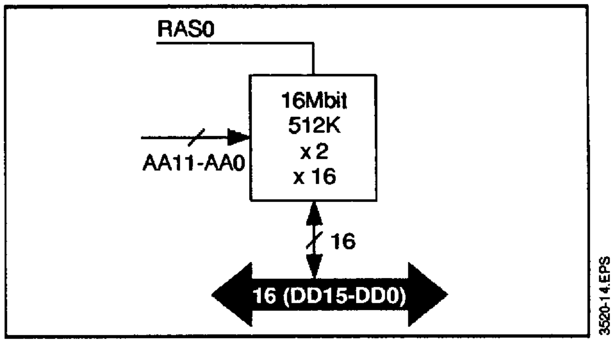
The operation of this mode is controlled using the general configuration register GCF. After power on the GCF register must be programmed to enable the EDO DRAM mode, GCF.EDO and the high frequency memory subsytem clock, GCF.MSC must be enabled.

Due to the variation in page size across different memories control is provided to enable the display process to work with 2 or 4 segments per row GCF.SEG.

VII.3 - Synchronous DRAM (only in STi3520 cut 2.0)

The interface can support a 16-Mbit configuration as shown in Figure 13. The exact connection of the memory is given in the "Pin List". The control signals for synchronous DRAM (SDRAM) are all synchronous with a clock which is generated internally in the decoder and is output with the control signals via the unused data pins. The control of the SDRAM uses the same signal types as standard DRAM but the signals are all sampled using a clock and a chip select signal. The precise timing is given in Figure 71.

Figure 13 : 16-Mbit Synchronous DRAM Configuration



The operation of SDRAMs is standardized by the JEDEC standard but unfortunately the standard does not cover all the aspects of interoperability. The memory interface, therefore, makes use of a constrained set of the JEDEC functionality. This ensures operation with SDRAMs from different vendors.

The Mode Register is programmed in thefollowing manner :

- burst length 4,
- sequential wrap,
- 3 cycle CAS latency.

The memory interface does not make use of the hidden precharge feature because this is not available from all manufacturers.

VII.3.1 - Initialisation of Synchronous DRAMs

Synchronous DRAM must be initialised in the power-on sequence like conventional DRAMs. Once power has been applied, a 100µs delay must precede any signals being toggled. During this delay all the memory interface signals are held in high impedance. In some cases all the control signals to the SDRAM must be held high during this time in some cases just DQM and CKE must be held high. Pullups should be applied to the control signals which are required to be high.

After the initial 100µs delay the both banks in the SDRAM are precharged using the "All Banks Precharge" command. Once the precharge is complete and the minimum t_{RP} is satisfied the Mode register is programmed. At least two SDRAM clock cycles after the Mode register set, refresh is enabled. A certain number of refresh cycles (manufacturer specified) must then be performed before the memory is ready for use. Care must be taken to allow enough refreshes to occur before the memory is used. A typical initialisation sequence is as follows :

- enable all clocks including the memory subsystem clock,
- enable the DRAM interface,
- set the GCF2.MRS & GCF2.SDR bits (see Initialisation Sequence Variation below),
- allow sufficient time for at least 8 refreshes before beginning to use the memory,
- enable GCF2.MRS bit.

VII.3.2 - Initialisation Sequence Variation

Certain manufacturers specify that the mode register should be programmed after a series of refreshes. To carry out a reset of this type the above sequence is used but the GCF2.MRS bit is set in step 5 and not in step 3.

VII.4 - Picture Storage Data Structure

Pictures are stored in memory as frames. They can be read and written as either frames or fields. In the latter case, every second line is accessed. Figure 14 shows how the data for a frame is mapped into memory when the memory interface is operating in normal (16-Mbit) mode.

The luminance and chrominance data for a frame are stored separately ; first all of the luminance macroblocks, and then all of the chrominance macroblocks. The base address of the luminance is defined by the frame base address as set up in one of the registers DFP (Display Frame Pointer), RFP (Reconstructed Frame Pointer), FFP (Forward Frame Pointer) or BFP (Backward Frame Pointer). The chrominance base addresses are calculated internally using the contents of the DFS (decoded frame size) register and the bits GCF.DFA[7:0]. For example for the display buffer the luminance start address in normal mode is :

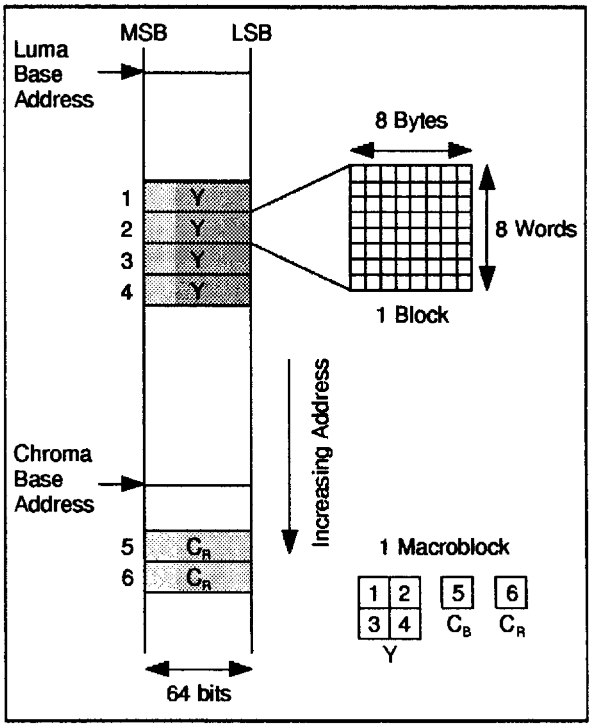
$$32 \times DFP$$

while the chrominance start address is :

$$32 \times (DFP + DFS + 4 \times GCF.DFA[7:0])$$

The size of the chrominance area is half that of the luminance area.

Figure 14 : Storage of a Macroblock

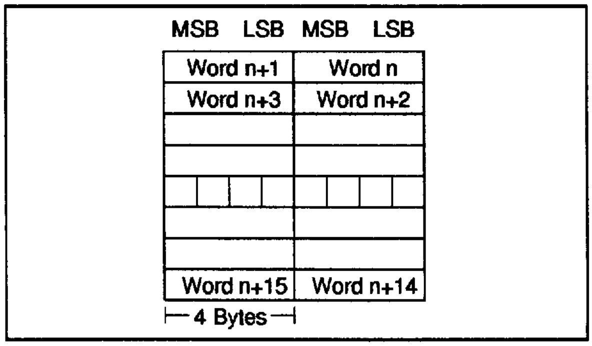


GCF.DFA[7:0] is generally set to zero when operating in normal mode. The use of this offset in 8-Mbit mode is described in section X.5, "8-Mbit Mode".

One memory page contains 512 64-bit words. One block requires 8 words. Thus one page can hold the luminance blocks of 16 macroblocks or the chrominance blocks of 32 macroblocks. In order to maximise the efficiency of memory access, macroblocks never cross page boundaries.

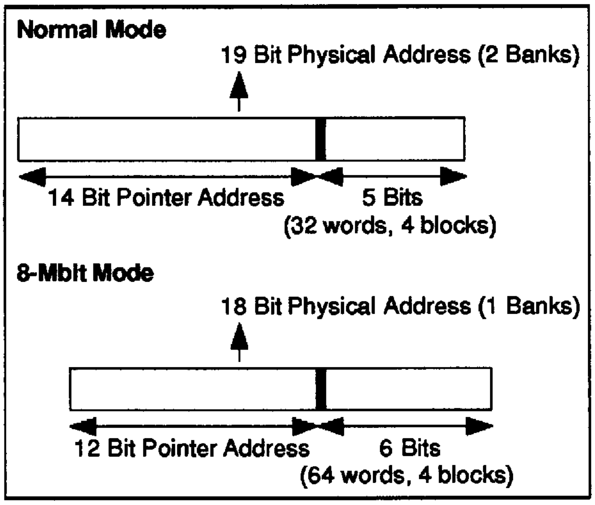
In 8-Mbit mode (in which the external memory data bus width is 32 bits), a block is stored as shown in Figure 15. Each word holds 4 8-bit luminance or chrominance values. Macroblock storage order is the same as that shown in Figure 14.

Figure 15 : Physical Storage of a Block in 8-Mbit Mode



From the user point of view, the frame pointers are programmed identically in the normal mode and 8-Mbit mode. In both cases, the pointer address units correspond to 4 blocks ; in normal mode, they correspond to 32 memory words, whereas in 8-Mbit mode, they correspond to 64 memory words. Figure 16 illustrates the mapping in the two cases. In 8-Mbit mode the top two bits of the 14-bit pointer are not used.

Figure 16 : Pointer to Physical Address Mapping



3520-17.EPS

VII.5 - Memory Read and Write through the Microcontroller Interface

Words can be read and written directly into the external memory into areas defined by the memory read pointer register, MRP, and the memory write pointer register, MWP, respectively.

VII.5.1 - Read

Data is transferred from the memory to the microcontroller interface through a 2-word FIFO. When the starting address of the area to be read from is loaded into the MRP register, this FIFO is cleared and 2 memory data words, starting at the address specified, are transferred into it. The status bit STA.RFF indicates when these two words are available in the FIFO. At this point the words can be read byte-by-byte (most significant byte first) from address MRF. When one word has been read, a new word is automatically loaded into the FIFO from the next memory address. After each transfer of a word from the memory to the FIFO, MRP is incremented. The maximum latency between the completion of the reading of a word and theloading of another into the FIFO from the memory is 99 primary clock cycles (1.8µs with a 55MHz clock) in normal mode and 159 clock cycles in 8-Mbit mode (2.9µs). Therefore, if the reading of a word takes longer than this, it is not necessary to check STA.RFF ; there will always be a word available in the FIFO for reading.

The timings of the two modes of reading, with and without polling, are given in Figure 17.

VII.5.2 - Write

Data in transferred from the microcontroller interface to the memory through a 2-word FIFO. When the starting address of the area to be written to is loaded into the MWP register, this FIFO is cleared. The status bit STA.WFE indicates when this FIFO is empty. At this point two words can be written byte-by-byte (most significant byte first) to address MWF. When one word has been written, a word is transferred into the memory at the address specified by MWP. After each transfer into the memory MWP is incremented. The maximum latency between the completion of writing of a word and the transfer of a word from the FIFO to the memory is 90 primary clock cycles (1.64µs with a 55MHz clock) in normal mode and 147 cycles in 8-Mbit mode (2.67µs). Therefore, if the writing of a word takes longer than this, it is not necessary to check STA.WFE; there will always be space available in the FIFO when writing.

The timings of the two modes of writing, with and without polling, are given in Figure 18.

VII.6 - Block Move

The block move facility allows blocks of words to be written from one part of the memory to another without the microcontroller having to perform any data read and write operations. The procedure for executing a block move is the following :

- set bit CMD.SBM to enable block move and disable user read/write to memory,
- write the base address of the block to be moved to register MRP. Write the base address of the destination to register MWP,
- set up the BMS register with the number of words to be moved. At the end of the write cycle, the block move will start. STA.BMI will be reset,
- when bit STA.BMI is set, the block move is complete,
- reset bit CMD.SBM. User read/write is now available again.

Memory read/write and block move operations can not be performed simultaneously.

Figure 17 : Memory Read Timing

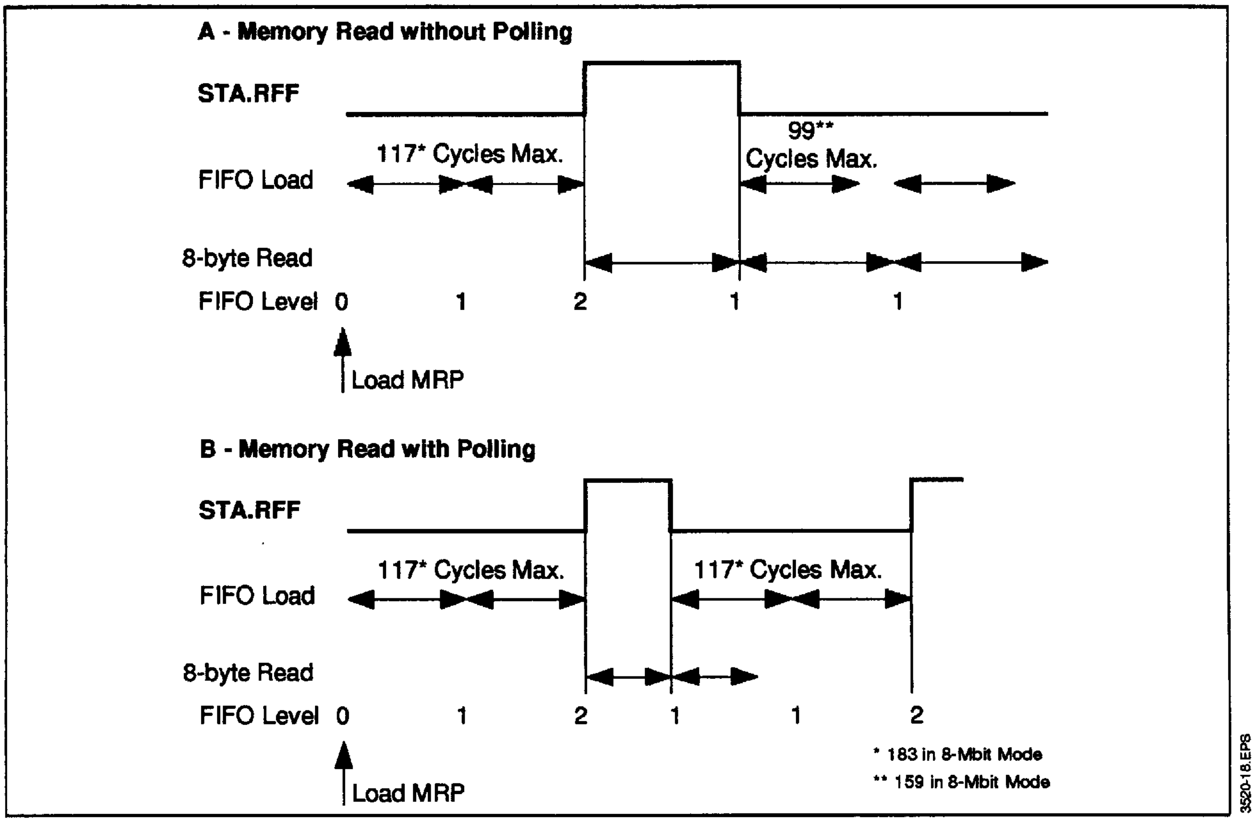
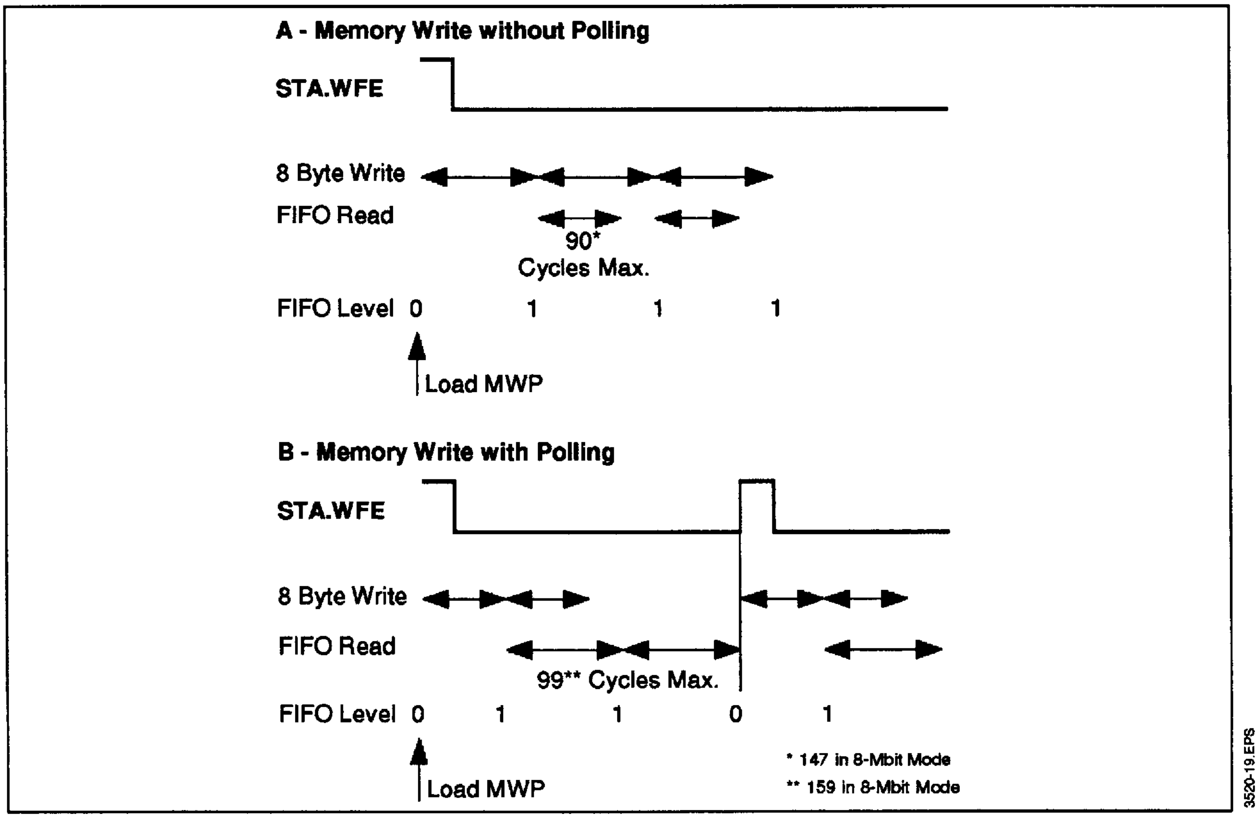


Figure 18 : Memory Write Timing



VIII - CLOCKS

VIII.1 - Video Decoder Primary Clock (CLK)

The STi3520 video decoder primary clock has a maximum value of 60MHz.

VIII.2 - Pel Clock (PIXCLK)

The pel output rate is defined by this clock. For ITU-R 601 output, the nominal value is 27MHz.

VIII.3 - Audio Primary Clock (AUDCLK)

The STi3520 audio decoder primary clock has a nominal value of 24MHz.

VIII.4 - PCM Clock (PCMCLK)

The input PCMCLK is used to generate the PCM output bit clock. Programming of the internal clock divider is described in section XI.1.2, "PCM Clock Generation".

IX - CLOCK GENERATION (only in STi3520 cut 2.0)

The on board clock generation consists of a patented frequency synthesizer circuit and associated dividers which derive all of the required system clocks from a single selectable input, thus eliminating the need for external dividers and PLL circuitry.

The reference input frequency may be obtained from two possible sources :
- the input clock, VXi,
- the pixel clock, PIXCLK.

The selected reference clock frequency is multiplied by a programmable integrated PLL and the output of the PLL is steered to a bank of dividers to generate each of the following clocks :
- the internal MPEG audio decoder clock,
- the internal MPEG video decoder clock,
- the memory sub-system clock.

IX.1 - Operational Overview

The frequency synthesizer is a classical PLL design, consisting of a phase detector, charge pump, filter, VCO and post-divider, M. The overall block diagram is show in Figure 19.

The input reference frequency, f_{in}, is selected using the mux prior to entering phase detector. The VCO feedback into the phase detector, f_{pll}, is programmably divided by M + 6, where M is in the range of 0 to 15. The phase detector detects the phase error and signals to the charge pump to increase or decrease the VCO frequency. The VCO typically operates at 200Mhz based on f_{in} and M.

The various output clocks are then generated using the three dividers. A programmable divider is provided to keep the audio decoder clock in the range 24 - 30MHz irrespective of the frequency f_{VCO}.

IX.2 - Control of the Clock Generator

The clock generation is controlled using a single register, PLL. See Register Map for more details.

IX.3 - Programming the PLL

The PLL is programmed based on a simple understanding of how the frequency synthesizer works. When the PLL is locked, the two frequencies at the input to the phase detector must be equal.

$$f_{ref} = f_{pll} \qquad \text{E.1}$$

where

$$f_{pll} = f\text{vco} / (\mathbf{M + 6}) \qquad \text{E.1.2}$$

and

$$\mathbf{0 \le M \le 15} \qquad \text{E.1.3}$$

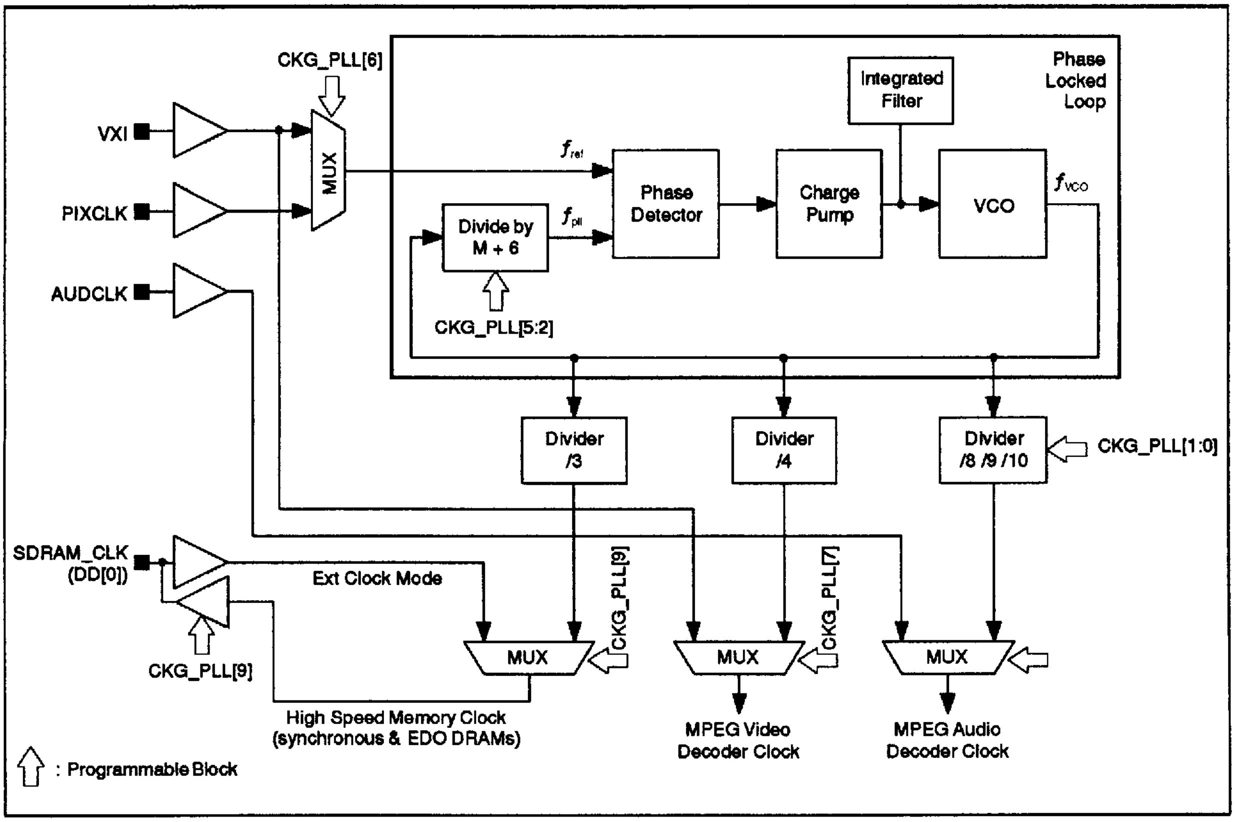
Therefore

$$f\text{vco} = f_{in} \times (\mathbf{M + 6}) \qquad \text{E.2}$$

Examples of PLL program values are shown in the following table based on common values for f_{in} :

f_{in} (MHz)	M	fVCO (MHz)
13.5	9	202.5
14.31818	8	200.45452
17.734475	5	195.079225
11.2896	12	203.2128
16.9344	6	203.2128
22.5792	3	203.2128

Figure 19 : Clock Generator Block Diagram



3520-20.EPS

X - VIDEO DECODER CIRCUIT DESCRIPTION

X.1 - Resets and Power-down Modes (video)

X.1.1 - Resets

There are three types of reset :

- a hard reset is generated by asserting Pin $\overline{\text{RESET}}$ for a duration of at least 15 primary clock cycles (270ns with a 55MHz clock),
- a soft reset is generated by setting and resetting bit CTL.SRS. It must be set for a duration of at least 40 primary clock cycles (730ns with a 55MHz clock),
- a pipeline reset is generated by setting and resetting bit CTL.PRS. It must be set for a duration of at least 3 primary clock cycles.

After a hard reset, all circuit activity stops and the registers are forced into the reset states defined in section XIV, "VIDEO REGISTERS". The circuit is put into low power mode (defined in section X.1.2, "Power-Down Modes"), and the video and memory interfaces are put into a high impedance state. All data remaining in the external memory is lost. A hard reset would normally be used after power-up and when it is required to place the circuit in low power mode.

After a soft reset, all processes concerning decoding and bit buffer control are reset. Any data remaining in the bit buffer, the compresseddata FIFO and the start code detector FIFO are lost. The interrupt unit is reset. All registers maintain their contents and the display process is not disturbed. A soft reset would normally be used when the decoding of the current bitstream must be terminated and it is required to restart on a new sequence.

After a hard or a soft reset, the first task performed by the pipeline when it has been enabled will always be a search for the beginning of a new sequence. The bit buffer data is flushed until the first picture start code following a sequence start code is detected by the pipeline, at which time it stops. At this point normal picture decoding behaviour is resumed. After a hard or a soft reset, the first search performed by the start code detector in response to the first DSYNC will always be a search for a sequence start code, after which it stops. After this, the start code detector operates normally.

A pipeline reset terminates the decoding of the current picture. The remaining bits of the picture are flushed from the bit buffer until the next picture start code is detected by the pipeline. At this point normal behaviouris resumed, i.e. the pipeline waits for the next picture decoding instruction. No other part of the circuit is affected by a pipeline reset. A pipeline reset would normally be used as part of a manual error recovery procedure. A pipeline reset has no effect if the decoding pipeline is in its idle state.

X.1.2 - Power-Down Modes

Low power mode is entered after a hard reset, or by resetting bits CTL.EVI, CTL.EDI and CTL.ECK. In low power mode, the whole circuit is shut down, and the video and memory and compressed data interfaces are disabled. In low power mode, the registers do not lose their contents (if there was no hard reset) and can be accessed normally. Power dissipation in this mode is the minimum possible.

Reduced power mode is entered by resetting bit CTL.EC2 and keeping bits CTL.EC3, CTL.ECK and CTL.EDI set. In this mode decoding is idle, but access to the external memory through the read and write FIFOs is possible. If the video interface is enabled by setting bit CTL.EVI, then the display interface may be used.

X.2 - Bit Buffer and Start Code Detection (video)

X.2.1 - Bit Buffer

The mechanism of writing compressed data into the bit buffer through the CD FIFO is detailed in section VI.6, "Compressed Data Input (video)" and section VI.7, "Bit Buffer Control (video)".

As part of the initialization sequence of the decoder, the registers BBG (bit buffer starting address), BBS (bit buffer stop address) and BBT (threshold for generation of BBF interrupt) must be set up.

The level of the bit buffer at any instant can be determined by reading the BBL register.

If it is required to change BBG before starting to decode a new sequence, this should be done before the soft reset is activated.

If the bit CTL.PBO is set (see section VI.7, "Bit Buffer Control (video)"), then BBT defines a "full" rather than "nearly full" level and should be set equal to BBS + 1 - BBG to optimize memory usage.

X.2.2 Start Code Detection

The startcode detectoroperates in parallel with the decoding pipeline. The purpose of this unit is to allow external access to the header data which follows start codes in the input bitstream. Compressed data is read twice from the bit buffer - once into the pipeline, and once into the start code detector through the 128-byte header FIFO. The transfer of data into the header FIFO does not affect the bit buffer level ; only the data transfer into the pipeline can reduce the bit buffer level.

Start code detection is initiated in two ways :

- automatically whenever the internal event DSYNC occurs. DSYNC is derived from VSYNC as described in section X.3.4, "Decoding Task Control". A DSYNC is generated every time the pipeline starts a new picture decoding task,
- manually by writing to the CMD register with bit CMD.HDS set.

When start code detection has been started, data is read continuously from the bit buffer into the header FIFO and parsed by the start code detector, which receives the FIFO output data. When a start code is detected, the data scanning stops and the status bit STA.SCH becomes 1. When a start code has been detected, it can be identified by reading the HDF register.

The start code detector detects all start codes other than the slice start codes, which are "00000101h" through "000001AFh".

After detection of a start code the HDF register will be one of the states shown in Figure 20.

Figure 20 : States of HDF After Start Code Detection

Last Byte of Start Code	First Header Byte	**HDF**
01	Last Byte of Start Code	**HDF**
Address 00	**Address 01**	

3520-21.EPS

The first step is to examine the byte at address 00. If this contains "01", then the start code can be identified by reading the byte at address 01. If the first byte is not "01" then it must be the last byte of the start code and the second byte is the first byte of the header data. In both cases subsequent reads from HDF will give access to the header data which follows the start code.

Scanning for start codes will recommence on the next DSYNC or write to CMD.HDS. Whenever a start code has been detected, the HDF register must be read in order for the start code detector to restart correctly.

The first start code search after a hard or soft reset will be a search for a sequence header start code ; all other start codes will be ignored. When this start code has been read, all subsequent searches will look for any start codes other than slice start codes.

The two status bits STA.HFE (header FIFO empty) and STA.HFF (header FIFO full) indicate the state of the header FIFO. Reading from HDF must never be performed if STA.HFE is 1. STA.HFF is set whenever the header FIFO contains at least 66 bytes.

X.2.3 - Handling of Time-Stamps

The video decoder is designed to accept only MPEG-1 or MPEG-2 video elementary streams. Time-stamps are not transmitted in this layer, but at the packet layer.

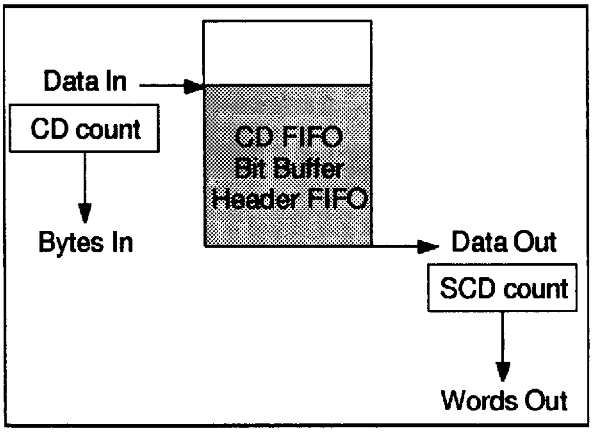
A mechanism is thus provided to enable the association of the time-stamps which are included in video packet headers with the times at which particular pictures are decoded. This is needed because the number of pictures which may be stored in the bit buffer at any instant is unknown, and therefore there is a variable delay between the input of a picture into the bit buffer and its entry into the decoding pipeline.

There is a 24-bit counter at the input and at the output of the CD FIFO - bit buffer - header FIFO chain, as shown in Figure 21. Each time a byte is written into the CD FIFO the counter "CDcount" is incremented. Each time a 16-bit word is read from the header FIFO the counter "SCDcount" is incremented. Both of the counters are reset by a hard or soft reset. Both are modulo 2^{24}, i.e. the state following FFFFFF is 000000.

When the first byte of video data from a new packet containing a time-stamp is written into the CD FIFO, CDcount is read. This value is recorded by the microcontroller in a list along with the time-stamp. When a picture start code is detected by the start code detector, SCDcount is read. If this value multiplied by two is greater (modulo 2^{24}) than the last CDcount in the list, then the next picture to be decoded is associated with the time-stamp stored at this position of the list. This time-stamp and CDcount pair is now removed from the list.

Bits CDcount[23:16] are read by first writing "01" to CMD.AVS[1:0] and then reading the least significant byte of CMD. Bits CDcount[15:0] are read by first writing "00" to CMD.AVS[1:0] and then reading CMD.

Figure 21 : Bit Buffer Read and Write Counters



Bits SCDcount[23:16] are read by first writing "11" to CMD.AVS[1:0]and then reading the least significant byte of CMD. Bits SCDcount[15:0] are read by first writing "10" to CMD.AVS[1:0] and then reading CMD.

X.3 - Video Decoding Pipeline Control

X.3.1 - General

The pipeline is that part of the circuit which converts the compressed bitstream data for each picture into a decoded (or reconstructed) picture. These pictures can be frame or field pictures. The operation of the pipeline is controlled picture-by-picture. The decoding of a new picture can potentially start on every VSYNC, but usually the rate of decoding is less than the VSYNC rate.

The pipeline is controlled by the pipeline controller (see General Block Diagram). When the pipeline controller starts the decoding pipeline a DSYNC signal is issued. This signal is also sent to the start code detector. When the pipeline has completed its decoding operation, a completion signal is sent to the pipeline controller, which is then able to launch another decoding operation, either immediately or when the next VSYNC occurs. The pipeline controller interprets certain bits of the decoding instruction, which must be set up by the user before the start of each new task. The remaining bits of the instruction define the decoding task itself.

The pipeline receives its compressed data from the bit buffer. This data is first processed by the variable length decoder (VLD) which regenerates the run/level coded DCT coefficients and the motion vectors (if present) for each macroblock. The picture data is reconstructed by passing the run/level data through the inverse quantizer and inverse DCT blocks. This is then added to the predictors which have been fetched from the memory taking into account the macroblock prediction modes and motion vectors. Finally, the decoded picture is written back into the memory, from where it can be accessed by the display unit for output.

The pipeline is also able to skip through picture data for various reasons. The different possibilities are :

- Skip to Next Sequence. This occurs unconditionally on the first instruction execution after a hard or soft reset (see section X.1.1, "Resets"). Compressed data is skipped until the first picture start code following a sequence start code is found. The pipeline then indicates task completion and waits for a new instruction.
- Skip to Next Picture. This occurs either after a pipeline reset (see section X.1.1, "Resets"), or when the decoding instruction specifies that one or two pictures should be skipped (see section X.3.4, "Decoding Task Control"). In the first case compressed data is skipped until the next picture start code is found, after which the pipeline indicates task completion and waits for a new instruction. In the second case, after the skipping operation the decoding of the following picture is started immediately.
- Skip to Next Slice. This occurs after automatic error concealment (see section X.3.6, "Error Recovery and Missing Macroblock Concealment"). Compressed data is skipped until the next slice start code in the picture is found, after which normal decoding resumes.

Before starting to decode a sequence, certain static parameters must be set up. These are :

- MPEG-1 or MPEG-2 mode selection. Bit CTL.MP2 must be set for an MPEG-2 sequence, reset for an MPEG-1 sequence.
- decoded picture size. Register DFW must be set up with the picture width in macroblocks, and register DFS must be set up with the number of macroblocks in the picture.

Decoding is enabled by setting bit CTL.EDC.

X.3.2 - Quantization Table Loading

The two quantization matrices (intra and non-intra) used by the inverse quantizer must be initialized by the user. There are no built-in quantization matrices. Therefore, they must be loaded either with default matrices or with those extracted from the bitstream by the microcontroller. The quantization tables are double-buffered. This enables one or both tables to be updated without disturbing the decoding task in progress. The STi3520 maintains two bits which record whether one or both of the tables have been modified. A modified table is automatically brought into operation at the start of the next decoding operation, i.e. when the next DSYNC occurs.

After a hard reset, the same pair of tables is always selected. The data previously loaded into the tables is not affected. Other types of reset have no effect on the quantization tables.

The quantization tables are written at address QMW. Bits CMD.QMN and CMD.QMI are used to control access to the tables. The writing procedure is described in the CMD and QMW register descriptions in section XIV, "VIDEO REGISTERS".

X.3.3 - Utilization of Picture Pointers

Before the decoding of each picture the following frame buffer pointers must be set up :

- RFP : Reconstructed Frame Pointer,
- FFP : Forward prediction Frame Pointer,
- BFP : Backward prediction Frame Pointer.

(A fourth pointer, DFP, the displayed frame pointer is described in section X.4.3, "Setting Up the Display").

RFP defines the memory buffer to which the decoded picture is written. FFP and BFP defined the areas in memory from which the predictors are fetched. How these two latter pointers are used depends on the prediction mode. The rules are given below.

Note that pictures are always stored as frames, and that to access a field (top or bottom), the starting address of the frame must be defined.

P-Frame-Picture (Frame, Field or Dual-Prime Prediction)

FFP is set to the address of the predictor frame (in which the two predictor fields lie). BFP is not used.

B-Frame-Picture (Frame or Field Prediction)

FFP is set to the address of the forward predictor frame (in which the two predictor fields lie). BFP set to the address of the backward predictor frame (in which the two predictor fields lie).

P-Field-Picture (Field, 16 x 8 or Dual-Prime Prediction)

When decoding either field, FFP is set to the address of the previous decoded I or P frame. BFP is not used.

B-Field-Picture (Field or 16 x 8 Prediction)

FFP is set to the address of frame in which the two forward predictor fields lie. BFP set to the address of the frame in which the two backward predictor fields lie.

I-Pictures

For I-picture decoding, no predictors are necessary, but FFP must be set to the address of the last decoded I or P-picture for use by the automatic error concealment function.

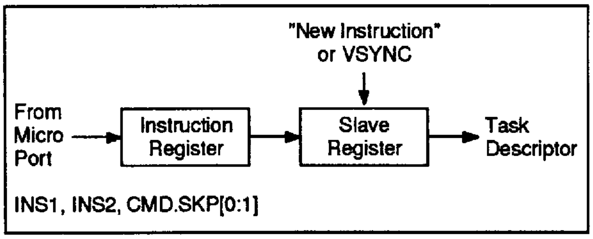
There are additional rules for frame pointer use in 8-Mbit mode. These are given in section X.5, "8-MBIT MODE".

X.3.4 - Decoding Task Control

A task is a single picture decoding operation. A task is specified by the task description or instruction, which is set up before the decoding of each picture. A task commences when the internal signal DSYNC is generated. A task completes (the pipeline becomes idle) when the picture header of the following picture is detected by the pipeline and the picture is entirely reconstructed in the memory. The instruction is double buffered, so that during execution of a decoding task, the instruction for the next task can be set up by the microcontroller. When the next instruction is activated, a DSYNC can be generated, and the next decoding task started. The buffering mechanism is illustrated in Figure 22. Note that some instruction bits are latched by VSYNC, others by a signal from the pipeline controller "new instruction" (this only has consequences when a task overruns. See Section X.3.5, "Task Overrun").

The instruction is written into registers INS1 and INS2, and bits CMD.SKP[1:0]. If a new instruction is not written, the task descriptor will be the same as the previous one.

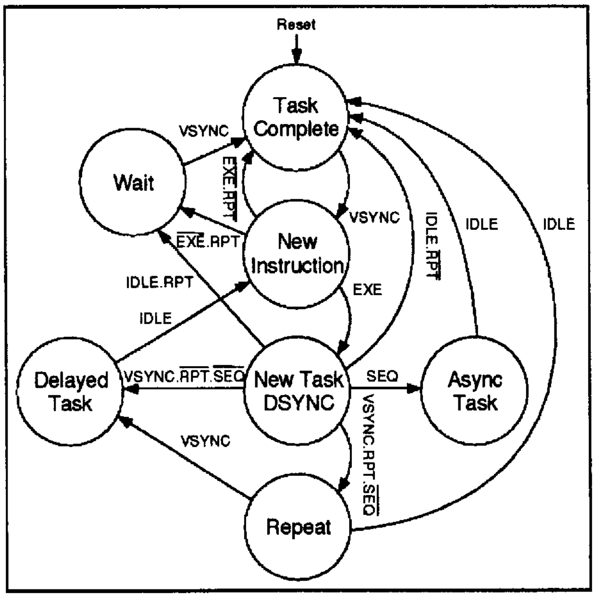
Figure 22 : Instruction Buffering



Normally, it is a VSYNC that starts the execution of a new instruction, and thus the generation of DSYNC. If however, a VSYNC occurs before task completion (i.e. before the pipeline becomes idle), the start of the next task will be delayed until the present one is completed. In this way the decoding of a picture can be allowed to extend beyond the nominal period allotted to it, usually one or two VSYNC periods.

Two status bits (and thus interrupts) are associated with pipeline control. STA.PSD indicates the occurrence of a DSYNC. STA.PID indicates that the pipeline is idle.

Figure 23 : Task Control State Diagram



The operation of the pipeline controller is shown in the state diagram of Figure 23.

The instruction bits which affect state transitions are INS1.EXE, INS1.RPT and INS1.SEQ. The events to which the controller responds are VSYNC, which could be a "VSYNC top" or a "VSYNC bottom" and "IDLE" representing the idle state of the pipeline.

The resting state of the controller is "task complete" which is entered after hard or soft resets or when a decoding task is completed. If a VSYNC occurs while the controller is in this state, the controller moves to state "new instruction". Here new instruction bits are loaded using the mechanism shown in Figure 22.

If the action required is "wait for one VSYNC period", i.e. do not generate a DSYNC and thus do not start the pipeline and start code detector, bits INS1.EXE and INS1.RPT must both be 0. The controller returns to state "task complete" and waits for the next VSYNC. If the action required is "wait for two VSYNC periods", the bit INS1.EXE must be 0 and bit INS1.RPT must be 1. The controller now passes through the states "new instruction" and "wait".

If the pipeline and start code detector are to be started, then bit INS1.EXE must be set. This will cause the controller to enter the state "new task" from which a DSYNC is generated. The pipeline will now execute the operation defined by the task description bits of the instruction (see later), provided that bit CTL.EDC is set. If CTL.EDC is not set, controller operation is not affected, but the task will not be executed by the pipeline and idle will remain reset.

If the time allocated to the task is one VSYNC period, INS1.RPT must be 0. The controller will remain in state "new task" until either the task completes or a VSYNC occurs. In the former case, the controller returns to state "task complete" and waits for the next VSYNC. In the latter case, the task has overrun and VSYNC arrives before IDLE. The state "delayed task" is entered. The controller remains here until IDLE occurs, when the state "new instruction" is immediately entered. Thus the next task is chained immediately to the one just completed.

If the time allocated to the task is two VSYNC periods, INS1.RPT must be 1. The controller will remain in state "new task" until the VSYNC at the end of the first period occurs. It now moves into state "repeat". The controller will remain in this state until either the task completes or the second VSYNC occurs. In the former case, the controller returns to state "task complete" and waits for the next VSYNC. In the latter case, the task has overrun and VSYNC arrives before IDLE. The state "delayed task" is entered. The controller remains here until IDLE occurs, when the state "new instruction" is immediately entered. Thus the next task is chained immediately to the one just completed.

The bit INS1.SEQ has a special function and is only used after a hard or soft reset. Before decoding is enabled after such a reset (bit CTL.EDC is set), bit INS1.SEQ and INS1.EXE must be set. On the occurrence of the reset, the pipeline starts scanning the bitstream until the first picture start code following a sequence start code is found. The controller will enter the state "async task" in which it is not responsive to VSYNCs. When the pipeline has found the startcode for which it was searching, IDLE becomes true and the state "task complete" is entered. To prevent the pipeline starting a task immediately on the following VSYNC, an instruction with INS1.EXE set to 0 can be loaded while the controller is in state "async task". (Entry into this state can be detected by monitoring bit STA.PSD).

It is possible to skip one or two pictures by using the bits CMD.SKP[1:0] as defined in the CMD register description (see section XIV, "VIDEO REGISTERS"). A skipping task consists of a skip followed by the decoding of the next picture.

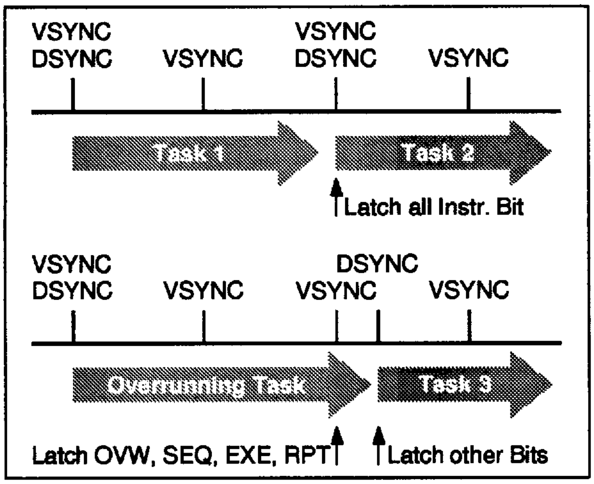
The bit INS1.OVW must be set when the picture is being reconstructed into the same buffer as that from which the displayed picture is being read, i.e. RFP = DFP. Overwrite mode is explained in more detail in section X.6.2, "Buffer Sequencing and Overwrite Mode".

The remaining bits of INS1 and INS2 are all set up with picture parameters derived directly from the picture header of the bitstream. These are detailed in the INS1 and INS2 descriptions (see Section XIV, "VIDEO REGISTERS").

X.3.5 - Task Overrun

Figure 24 illustrates the timing, with respect to VSYNC, of both normal and overrunning tasks. In the former case, the decoding task is complete before the next (or second when INS1.RPT = 1) VSYNC arrives. At this point all bits of the new instruction are updated. In the case of an overrunning task, the latching of the new instruction bits occurs in two stages ; the first when the first (INS1.RPT = 0) or second (INS1.RPT = 1) VSYNC arrives, the second when the task is completed, at which point a DSYNC is also generated (if INS1.EXE = 1). The instruction must thus never be changed between the last VSYNC and the following DSYNC.

Figure 24 : Normal and Overrunning Task Timing with INS1.RPT = 1



X.3.6 - Error Recovery and Missing Macroblock Concealment

There are four levels of error detection and recovery available in the video decoder :

- bitstream syntax error detection with the option of automatic missing macroblock concealment,
- bitstream semantic error detection with the option of automatic concealment or skip to the next picture,
- pipeline overflow or underflow error detection,
- user-initiated skip to next sequence using soft reset.

Syntax Error Detection and Concealment

In normal operation of the STi3520, error concealment must always be enabled, i.e. CTL.DEC should be reset.

If the VLD detects a syntax error in the bitstream, the pipeline will copy macroblocks from the previous picture using the motion vectors reconstructed for the previous row of macroblocks in the current picture, while scanning the bitstream until a slice start code is detected. At this point normal decoding resumes. If the slice in which the error occurred was the last one in the picture, concealment will continue until the end of the picture, at which time the pipeline stops normally (assuming that the following picture start code is intact).

Concealment of macroblocks is carried out by using the vectors of the macroblock immediately above the lost macroblock. The pipeline is able to store one row of such information, for a decoded picture size of up to a maximum of 46 macroblocks (for decoded picture widths of more than this, error concealment will be degraded). Two vectors are stored for each macroblock in the row.

The concealment macroblocks are accessed using the pointers FFP and BFP. Lost Macroblocks in the first row are copied directly from the previous pictures (i.e. as P-macroblocks with zero motion vectors). If an intra picture is coded with concealment motion vectors, these will be used. If not, then the concealment will be a simple copy from the previous picture using zero vectors. Even in intra pictures, the pointer FFP must be set up.

The following rules are used for the fetching of concealment macroblocks :

- I-pictures :
  - I-macroblocks without vectors → copy with zero motion,
  - I-Macroblocks with vectors → copy as forward predicted macroblock.
- P-pictures :
  - I-macroblocks → as above,
  - P- macroblocks → copy using stored vector,
  - P-field-macroblocks → copy in field mode using both vectors,
  - Skipped macroblocks → copy with zero vector,
  - Dual-prime macroblocks → same as for normal P-macroblock, since both full vectors are saved.
- B-pictures :
  - I-macroblocks → as above,
  - Forward macroblocks → as above for P-macroblocks,
  - Backward macroblocks → as above for P-macroblocks, but using backward vectors,
  - Bidirectional macroblocks → only the forward vectors are stored, concealed as forward macroblock,
  - Skipped macroblocks → copy in frame mode using the same mode and vectors as the previous macroblock.

If an error is detected in the bitstream before it enters the STi3520, then an error start code can be inserted into the bitstream in order to initiate concealment. However, when doing this there are certain restrictions on the placement of the error start code in order to avoid emulation of other start codes. An Application Note is available on this topic.

Pipeline Error

A pipeline error occurs whenever the pipeline reconstructs more than 64 coefficientsin a block. This condition is signalled by bit STA.PER. If bit CTL.EPR is set, a pipeline reset is automatically generated, and STA.PER is reset. If a pipelinereset is generated, the remainder of the picture will not be reconstructed ; the data displayed will be that which was already in the buffer from a previously decoded picture.

Overflow or Underflow Error

An overflow error occurs whenever the pipeline reconstructs more macroblocks than are defined by the decoded picture size, DFS. This condition is signalled by bit STA.SER. Decoding is halted when this error is detected. In order to restart decoding a pipeline reset must be performed.

An underflow error occurs whenever the pipeline reconstructs less macroblocks than are defined by the decoded picture size, DFS. This condition is signalled by bit STA.PDE. Decoding is halted when this error occurs. In order to restart decoding a pileline reset must be performed.

Soft Reset

The effect of this last resort action is described in section X.1.1, "Resets".

X.4 - Display Functions

X.4.1 - Operation of the Display Unit

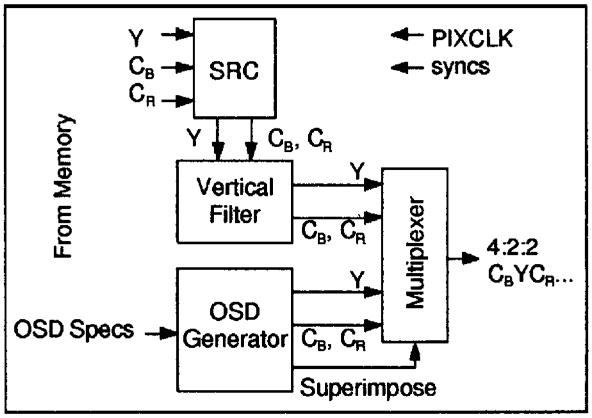
The display unit performs the following functions :

- requests and receives from the picture buffer in external memory the decoded picture data for display,
- optionally performs horizontal resampling of both luminance and chrominance data,
- reconstructs vertical data to create 4:2:2 sample format,
- generates on-screen display bitmap for superposition onto picture output.

Figure 25 is a simplified block diagram of the display unit, showing the sequence in which the horizontal and vertical filtering operations occur and where the on-screen display (OSD) data is added.

The picture data is received from the display frame buffer area of the external memory through three FIFOs (one for luminance and two for chrominance) into the luminance and chrominance horizontal sample-rate converter (SRC). For every line of luminance (Y) samples, an equivalent number of chrominance (C_B, C_R) samples are read from the memory. This chrominance line duplication is done in order to simplify the vertical filtering operation. The lines of a frame can be read out from the memory either in interlaced or line-sequential order. The integer part of the horizontal pan-scan offset set up by the user defines which pel will be the first one of a line to be read from the FIFOs.

Figure 25 : Display Unit Block Diagram



The sample rate converter (SRC), an 8-tap filter, has two functions : upsampling of pel data when the displayed line length is greater than the decoded picture width, and implementation of the fractional part of the pan-scan horizontal offset. The outputs of the SRC are upsampled lines each having equal numbers of luminance and chrominance samples. The SRC can be bypassed if desired.

The vertical filter, used for vertical interpolation, is 2-tap filter which includes a 720-sample delay line. The filter can be applied either to chrominance or luminance data, but not both at the same time. There are four different modes for vertical reconstruction of chrominance data for interlaced pictures, and 3 modes for the display of half resolution pictures.

The coded specification for the top or bottom field OSD bitmap is received through a FIFO from the external memory into the OSD generator, which creates a bitmap. When there is bitmap data which is not defined as "transparent", the output multiplexer replaces the decoded picture with the OSD data. Four bits are available to define each of the Y, C_B and C_R values of an OSD region. A special OSD output value is available to define transparency. The OSD bitmap is defined with respect to the display area and is independent of decoded picture size and any pan-scan offset.

X.4.2 - Video Interface

The video interface consists of the following signals :

Name	Function
YC7-YC0	Video Port
PIXCLK	Pel Clock
$\overline{\text{HSYNC}}$	Horizontal Sync
B/$\overline{\text{T}}$	Bottom/Top Field Selection

In response to each rising edge of PIXCLK a new 8-bit data word is available at the video port YC7-YC0. These data words are output in the 4:2:2 format sequence :

C_B Y C_R Y C_B ...

The first three words are the co-sited samples of the first pel, while the fourth is the luminance only sample of the second pel. The next three refer to the third pel, and so on. The first data word output in each active line period is always C_B.

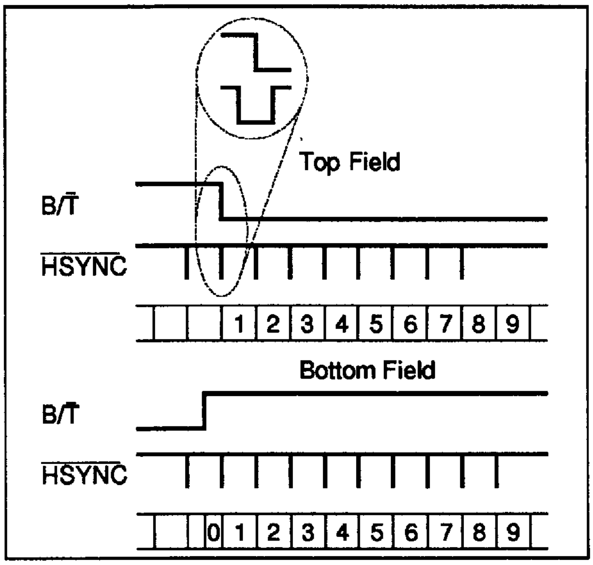
The start of each line is signalled by a falling edge of the $\overline{\text{HSYNC}}$ input. Internal pel (i.e. horizontal) counting is started by this event. The internal line counter is incremented by the rising edge of this signal. $\overline{\text{HSYNC}}$ must be low for at least 4 PIXCLK cycles, and must return to high before the end of horizontal blanking.

Vertical synchronization is derived from the B/$\overline{\text{T}}$ input. This signal is high during the bottom field and low during the top field. The internal vertical synchronization signals "VSYNC top" and "VSYNC bottom" are derived from the transitions of B/$\overline{\text{T}}$ (when the term "VSYNC" is used alone, it refers to "VSYNC top" or "VSYNC bottom"). These signals are mapped onto the register bits STA.VST and STA.VSB, and can be used to generate interrupt requests.

The internal line counter is reset on every transition of B/$\overline{\text{T}}$, and incremented by the rising edge of $\overline{\text{HSYNC}}$. The state of the line counter at the start of the top and bottom fields is shown in Figure 26. Note that the internal line count restarts every field. In the 525/60 standard, internal line count 1 in the top field corresponds to line 4, and internal line count 1 in the bottom field corresponds to line 267. In the 625/50 standard, internal line count 1 in the top field corresponds to line 1, and internal line count 1 in the bottom field corresponds to line 314.

At the beginning of the top field, the falling transition of the signal B/$\overline{\text{T}}$ must be sampled by the STi3520 before the rising transition of the first $\overline{\text{HSYNC}}$ of this field. The timing constraints of the video interface are given in section XIII.10, "Video Interface Timing". In most applications the value of PIXCLK will be 27MHz, in order for the output to be compatible with the ITU-R 601 standard.

Figure 26 : Internal Line Numbering



The video interface can be disabled in order to reduce power consumption by resetting register bit CTL.EVI. When the video interface is disabled, the outputs are put into their high impedance state and the inputs are disabled. PIXCLK is thus removed from the internal circuitry. After a hard reset, the video interface is in the disabled state.

X.4.3 - Setting Up the Display

The DFP register must be set up with the base address of the buffer containing the picture to be displayed. This register is double-buffered ; when a new value is written it is taken into account on the occurrence of a VSYNC. Thus it is possible to write a new value of DFP every field, although it would normally be updated only once per frame.

The picture stored in the buffer is always treated as a frame by the STi3520.

The STi3520 has two built-in ways of displaying this data :

- every second line of this data is read from the buffer. The reading of one of these fields is started on every VSYNC. If B/$\overline{\text{T}}$ indicates the top field, the 1st, 3rd, 5th, etc. lines will be read. If B/$\overline{\text{T}}$ indicates the bottom field, then the 2nd, 4th, 6th, etc. lines will be read. This mode is selected by resetting bit DCF.VFC[2] (together with bit CTL.HRD if overwrite mode is enabled).
- every line of the picture buffer is read starting on every VSYNC, i.e. scan is line-sequential. This mode would normally be used when displaying a picture which was decoded at half of the resolution of the display, and where it is necessary to

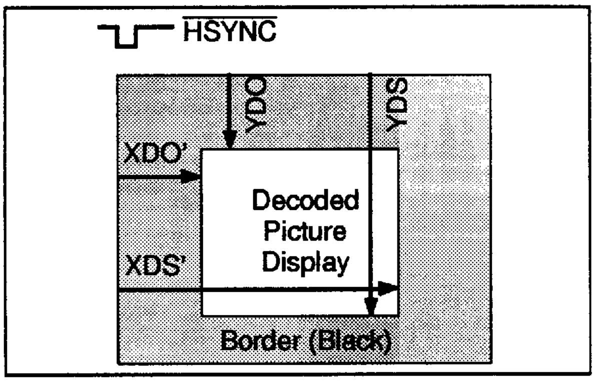
use the same picture data for each field. It is selected by setting bit DCF.VFC[2] (together with bit CTL.HRD if overwrite mode is enabled).

To use these standard modes, bit DCF.USR must be reset. Other, non-standard, modes of controlling the display sequence are described in section X.4.9, "Displayed Field Sequence Control".

If at any time no display is required, bit DCF.EVD may be reset, in which case a constant black value ($Y = 16$, $C_B = C_R = 128$) is output.

The size and location of the display window is defined by the registers XDO, XDS, YDO and YDS. The values loaded into these registers define the horizontal and vertical boundaries of the displayed picture, as shown in Figure 27. Outside of the picture area a black ($Y = 16$, $C_B = C_R = 128$) border is generated.

Figure 27 : Display Window Positioning



3520-28.EPS

Register YDO is loaded with the number of the last line of the upper border, where lines are numbered in fields as shown in Figure 26. The first active line is therefore defined by :

First active line = YDO + 1

The same YDO value serves for both fields ; the uppermost line of the picture display will be in the top field.

Register YDS is loaded with a number defining the last line of the picture display in a field, according to the relation :

Last active line = YDO + (vertical size/2) = YDS + 129

For example, with a 525/60 display, in which the vertical size of the decoded picture is 480 lines, typical values of YDO and YDS could be :

YDO = 21, YDS = YDO + 240 - 129 = 132,

and with a 625/50 display, in which the vertical size of the decoded picture is 576 lines, typical values of YDO and YDS could be :

YDO = 22, YDS = YDO + 288 - 129 = 181.

Register XDO defines the number of PIXCLK cycles between the falling edge of the signal HSYNC and the beginning of the picture display, according to the relation :

Cycles from HSYNC to start of picture = 2XDO + 40

The ITU-R601 standard defines this number to be 264 27 MHz clock cycles for a 625/50 display, and 244 for a 525/60 display. The respective values of XDO are thus 112 and 102.

XDO must never be less than ($177 \times f_{pel}$)/($2 \times f_{primary}$), where $f_{primary}$ is the video decoder primary clock frequency and f_{pel} is the pel clock frequency. If $f_{primary}$ = 55MHz and f_{pel} = 27MHz, the minimum value of XDO is 44.

The first picture data in a line, output in the 2XDO + 41st PIXCLK cycle after the falling edge of HSYNC, is always a C_B component. Since the external video generation circuitry will usually relate its Y/C phasing to the horizontal synchronization signal, and has no knowledge of the value of XDO, not all values of horizontal offset will be useable ; some will cause incorrect interpretation of the colour difference components. In any given system, XDO values will have to be either always odd or always even.

XDS is loaded with a number defining the last active sample in each line, counted in units of PIXCLK cycles from the falling edge of the signal HSYNC, according to the relation :

Last sample of active video = 2XDS + 28.

Thus, if L is the number of pels per line of the displayed picture, then :

2XDO + 40 + 2L = 2XDS + 28, and thus

XDS = XDO + L + 6

If L = 720, then XDS = XDO + 726.

The resolution to which the horizontal offset and end values can be defined is equal to two cycles of PIXCLK. In order to position the display window horizontally to a finer precision the DCF.PXD bit is used. When this bit is set, the active video is delayed by one PIXCLK cycle. Since the first active video sample is C_B, the Y/C phasing with respect to the horizontal synchronization signal will change.

X.4.4 - Sample Rate Converter and Pan Vector

X.4.4.1 - Sample Rate Converter

The purpose of the sample rate converter (SRC) is allow upsampling of picture data in order to increase the number of horizontal samples in a line. This is necessary if the horizontal size of the display is greater than the decoded picture width. For example if it is required to display a 720-pel wide

16:9 source image on a 4:3 display also of 720-pel width, then 540 pels selected each source line must be upsampled to 720.

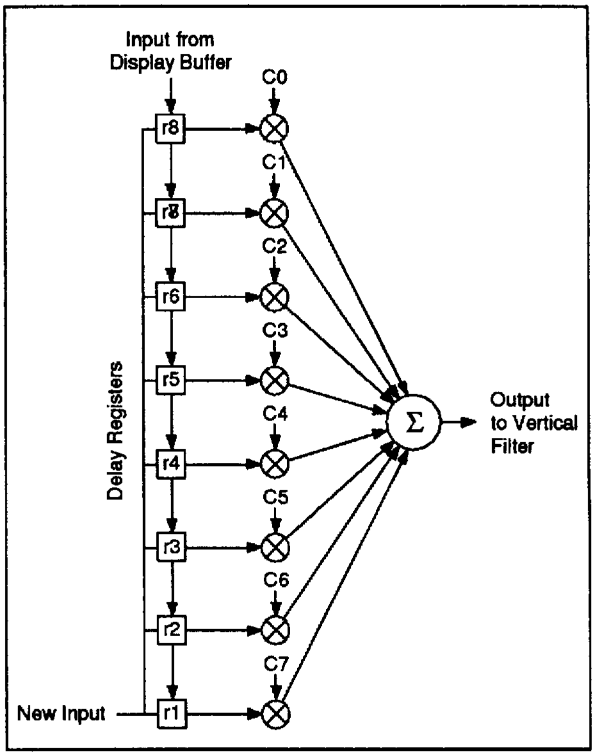
To enable the SRC, bit DCF.DSR must be reset. If this bit is set, the SRC is bypassed and the horizontal resolution of the decoded picture is not changed.

The SRC operates by interpolating samples between those of the decoded picture data read from the display buffer. This is performed by an 8-tap interpolation filter with the structure shown in Figure 28. The filter has three sets of delay registers multiplexed between the Y, C_B and C_R samples. It has 8 sets of coefficients, each set defining one of 8 sub-pel interpolation positions. For example, for sub-pel position 0, the output is aligned with stored sample "r4", for sub-pel position 1, the output corresponds to an interpolated pel position one eighth of the distance from sample "r4" to sample "r5", and so on.

The number of inputs clocked into the SRC is equal to the number of samples used in each line of the source image, and the number of outputs generated is equal to the number of samples displayed. Thus the rate of generation of outputs will be greater than the input data rate.

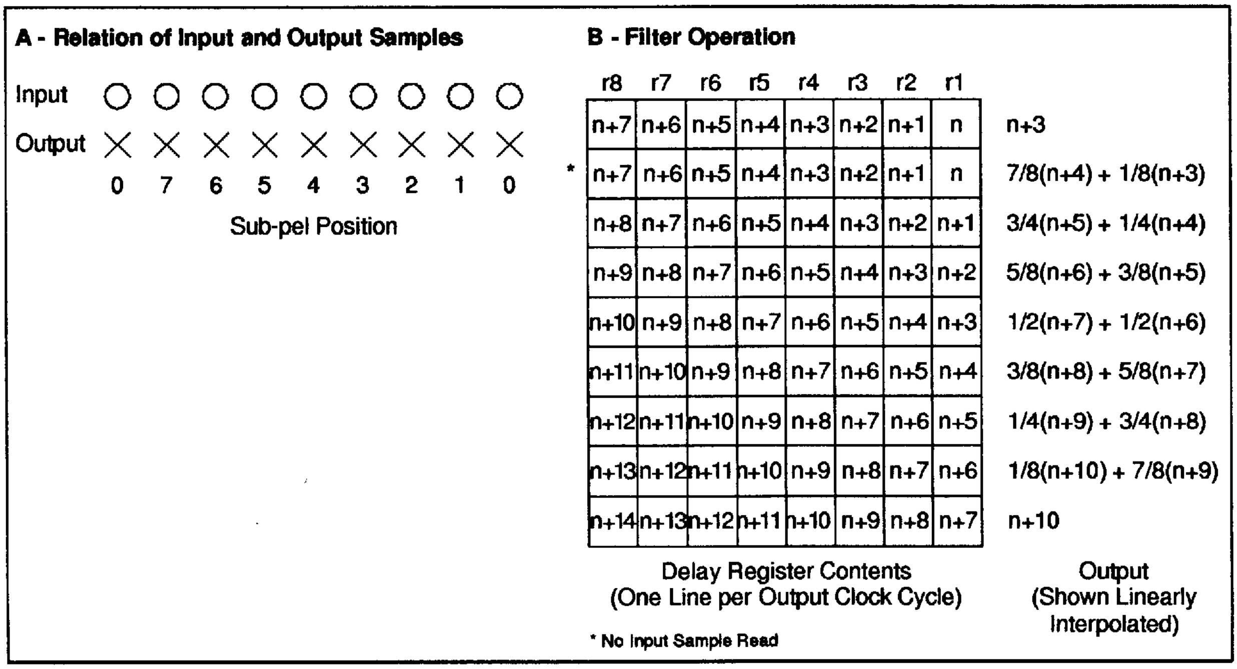
The example in Figure 29 illustrates the operation of the sample rate converter when the upsampling ratio is 8:7. For every 8 samples clocked out of the filters, 7 samples are clocked in. To illustrate the interpolation positions, at the right of the figure are shown the outputs which would occur with a simple linear interpolation (i.e.a 2-tap filter). The actual SRC output values are the 8-tap filter outputs with coefficients appropriate to sub-pel positions 0, 7, 6, 5, 4, 3, 2, 1, 0 etc.

Figure 28 : 8-Tap Interpolation Filter



3520-29.EPS

Figure 29 : SRC Example for 8:7 Upsampling



	r8	r7	r6	r5	r4	r3	r2	r1	
	n+7	n+6	n+5	n+4	n+3	n+2	n+1	n	n+3
*	n+7	n+6	n+5	n+4	n+3	n+2	n+1	n	7/8(n+4) + 1/8(n+3)
	n+8	n+7	n+6	n+5	n+4	n+3	n+2	n+1	3/4(n+5) + 1/4(n+4)
	n+9	n+8	n+7	n+6	n+5	n+4	n+3	n+2	5/8(n+6) + 3/8(n+5)
	n+10	n+9	n+8	n+7	n+6	n+5	n+4	n+3	1/2(n+7) + 1/2(n+6)
	n+11	n+10	n+9	n+8	n+7	n+6	n+5	n+4	3/8(n+8) + 5/8(n+7)
	n+12	n+11	n+10	n+9	n+8	n+7	n+6	n+5	1/4(n+9) + 3/4(n+8)
	n+13	n+12	n+11	n+10	n+9	n+8	n+7	n+6	1/8(n+10) + 7/8(n+9)
	n+14	n+13	n+12	n+11	n+10	n+9	n+8	n+7	n+10

Delay Register Contents
(One Line per Output Clock Cycle)

Output
(Shown Linearly
Interpolated)

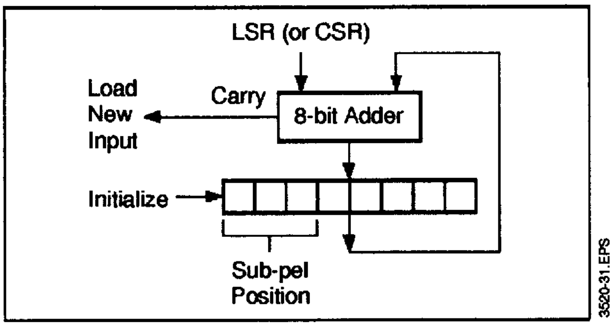
* No Input Sample Read

3520-30.EPS

The SRC output is limited to lie within the range [1,254], i.e. the codes 00h and FFh are never output, giving compatibility with ITU-R 656.

The SRC upsampling factor is set up in the LSR and CSR registers, which hold the factors for the luminance and chrominance components respectively. The same value must be loaded into both of these registers. The upsampling factor is equal to 256/LSR (or 256/CSR). This value is used to determine both the rate of input of data into the filters and the sequence of sub-pel interpolation positions. The mechanism by which this is achieved is shown in Figure 30.

Figure 30 : Upsampling Filter Control



The LSR value is added into an accumulator register at a rate equal to the filter output rate. A new input is loaded into the filter at beginning of a line and whenever a carry is generated by the adder. The top three bits of the accumulator register are used to select the sub-pel position. For example, with an upsampling factor of 8:7, the LSR value is (256/8) x 7 = 224. The sequence of values in the accumulator register will be (assuming that it is initialized to zero) :

Acc. Register	New Input	Sub-pel Position
0	Yes	0
224	No	7
192	Yes	6
160	Yes	5
128	Yes	4
96	Yes	3
64	Yes	2
32	Yes	1
0	Yes	0

The LSR/CSR value thus defines a cycle of sub-pel positions as well as the rate of data input. If a value of less than 32 is loaded into LSR/CSR, i.e. an upsampling ration of greater than 8 is defined, there could be repeated values in the filter output. This may cause unacceptable display artifacts.

At the start of a line, the 3 sets of delay registers r1, r2 and r3 are loaded with the black value (Y = 16, $C_B = C_R = 128$). The first output is thus derived from the inputs stored in registers r4 to r8. At the end of a line, the last eight input samples are stored in registers r1 to r8. The last valid interpolation is between the samples stored in r4 and r5. Correct Interpolation is not possible beyond this except in the case where the next output is in sub-pel position 0. This output is valid since coefficient C0 is zero for this position and the invalid sample beyond the end of the line is ignored. There is thus no valid interpolation possible between the last four input samples. This is illustrated in Figure 31 in which 544 pels are upsampled to 721, in which the upsampling ratio is 4:3. The LSR and CSR registers would be loaded with the value 192.

The number of valid outputs generated can be calculated as follows.

The ratio of the distances between input and output samples is 256:LSR. Given that the last output sample cannot occupy a position beyond the fourth-last input sample, the following inequality is always true :

$$LSR(N-1) \leq 256(M-4)$$

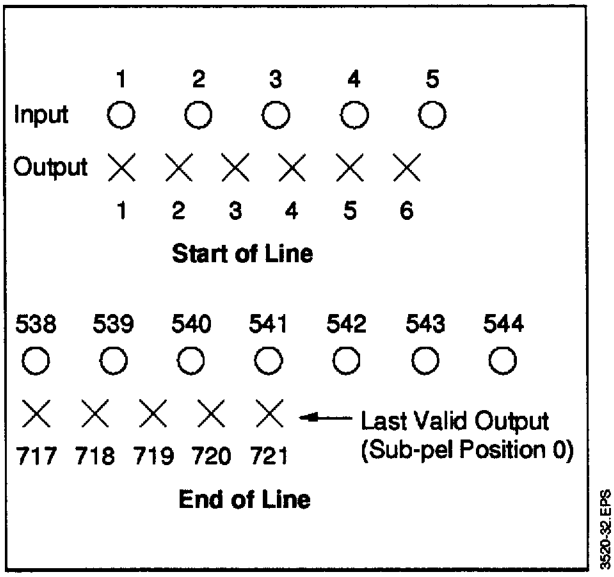
where N is the number of output samples and M is the number of inputs samples. The value of N is thus given by :

$$N = \lfloor 256(M-4)/LSR + 1 \rfloor$$

where "$\lfloor x \rfloor$" indicates the integer part of x.

The value programmed into the XDS register must be such that all samples beyond the last valid one are masked.

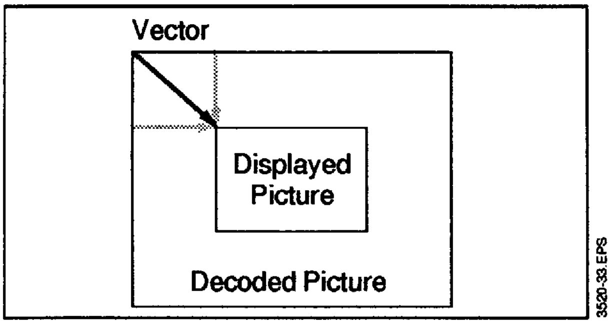
Figure 31 : Upsampling from 544 to 720



X.4.4.2 - Pan/Scan Vectors

When the display window has a smaller horizontal dimension than the decoded picture, a vector can be programmed in order to define the starting point of the displayed picture, as shown in Figure 32. This vector defines the point in the decoded picture which corresponds to the top-left-hand corner of the displayed picture. The displayed picture size and location is defined by the numbers programmed in registers XDO, XDS, YDO and YDS.

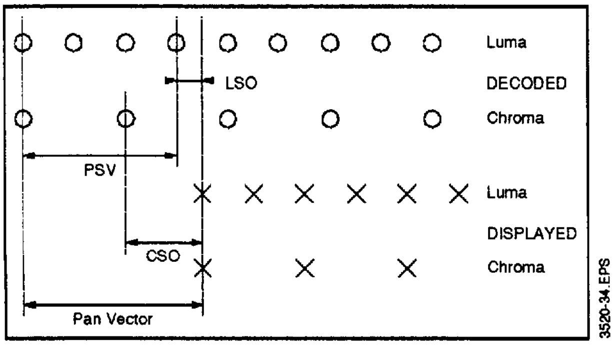
Figure 32 : Pan/Scan Vector



The pan/scan vector is programmed into the PSV register. This register is double-buffered ; when a new value is written it is taken into account on the occurrence of a VSYNC. Thus it is possible to write a new value of PSV every field.

The integer part of the horizontal component of the pan/scan vector is loaded into PSV.H[8:0], and the fractional part defines the contents of the LSO and CSO registers. The relationship between these quantities is illustrated in Figure 33. The numbers loaded into the LSO and CSO registers are used to initialize the luminance and chrominance up-sampling control registers (see Figure 30) at the start of every line. LSO is set up directly with the value of the fractional part of the pan/scan vector horizontal component. CSO is set up with half of this number, plus 128 if the integer part is an odd number. The resolution to which the horizontal component can be defined is 1/8 pel.

Figure 33 : Components of the Pan Vector



The vertical component of the pan/scan vector is programmed into PSV.V[7:1], in units of 2 field lines, or 4 frame lines. For example, if PSV.V[7:1] is set to the value 16, then the 32 top lines of every field would not be output to the display.

X.4.5 - Vertical Filter

The vertical filter performs the reconstruction of the 4:2:2 data output from the decoded (and possibly upsampled) picture data which is stored in 4:2:0 format.

The vertical filter has six modes for the handling of different methods of chrominance reconstruction and for the display of half-resolution pictures. The vertical filter mode is selected by the programming of bits DCF.VFC[2:0].

In order to explain the modes, the following line numbering system will be used :

- for an interlaced frame, the fields stored in the memory have the sampling patterns and line numbers shown in Figure 34,
- for a line-sequential frame, the sampling pattern and line numbering is as shown in Figure 35.

These patterns are preserved by the SRC and are thus the patterns entering the vertical filter. For every luminance line output from the memory, a chrominance line is also output. The sequence in which chrominance lines are output is determined by the vertical filter mode.

Figure 34 : Field Sampling Patterns

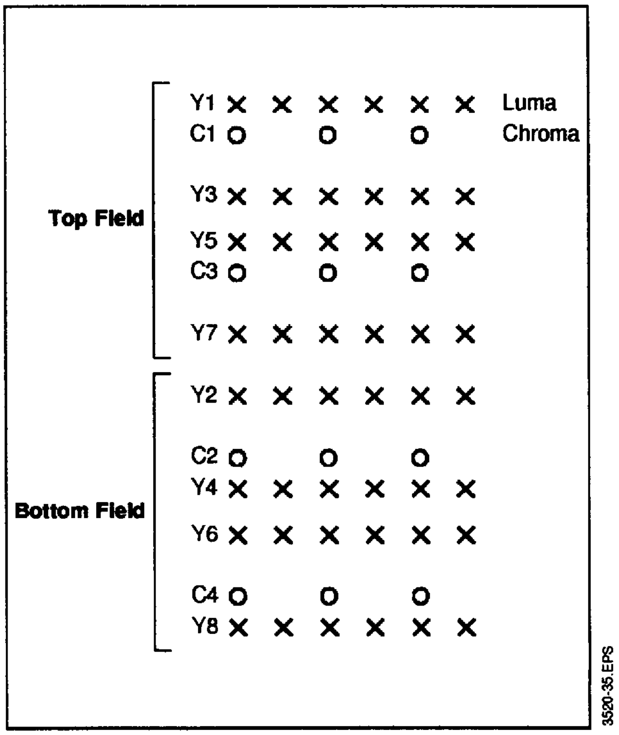
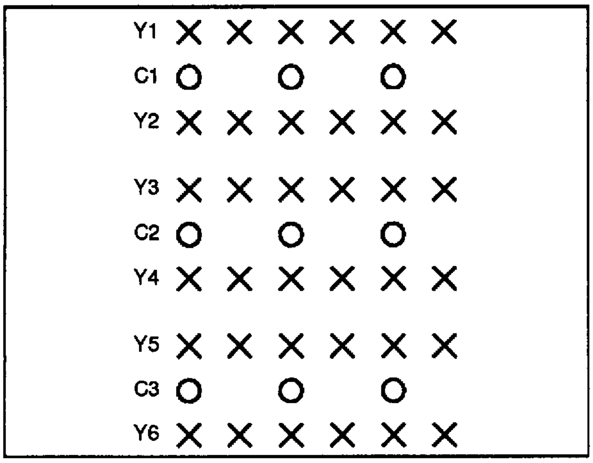


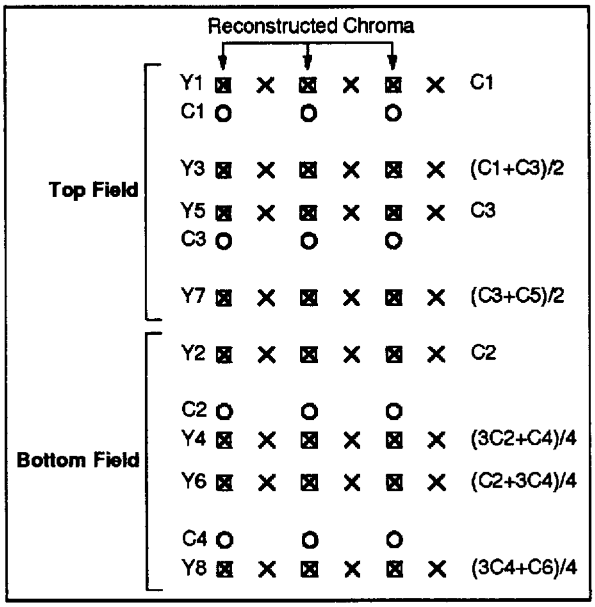
Figure 35 : Frame Sampling Pattern



Mode 0 : Full Resolution, Chrominance Line Repeat with Interpolation

This mode is used when the MPEG-2 picture-layer bitstream variable "progressive_frame" is 0, indicating that the chrominance samples were derived for each field independently, and when interpolation is required. Figure 36 shows how the chrominance is reconstructed in the two fields for 4:2:2 output. The luminance is unaffected by the vertical filter.

Figure 36 : Derivation of Chrominance in Mode 0

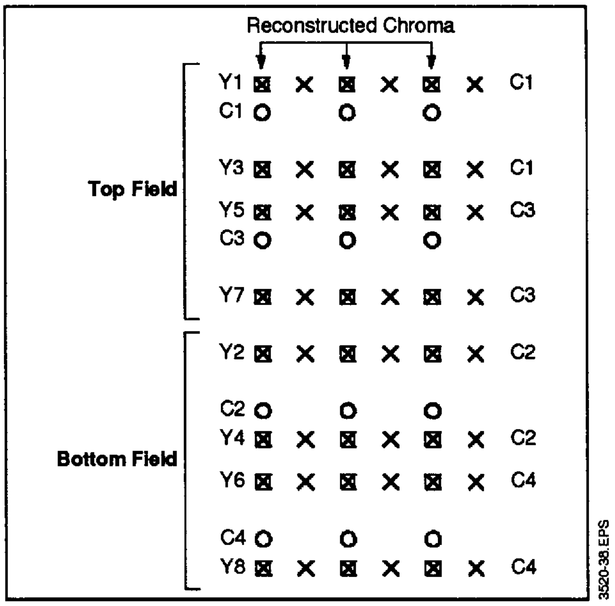


The chrominance data output with the last line of the top field (line 479 in a 480-line frame, or line 575 in a 576-line frame) is the same as that output with the previous line (line 477 or line 573). The chrominance data output with the last line of the bottom field (line 480 or 576) is the same as the last line of chrominance input.

Mode 1 : Full Resolution, Chrominance Line Repeat

This mode is used when the MPEG-2 picture-layer bitstream variable "progressive_frame" is 0, indicating that the chrominance samples were derived for each field independently, and when interpolation is not required. Figure 37 shows how the chrominance is reconstructed in the two fields for 4:2:2 output. The luminance is unaffected by the vertical filter.

Figure 37 : Derivation of Chrominance in Mode 1

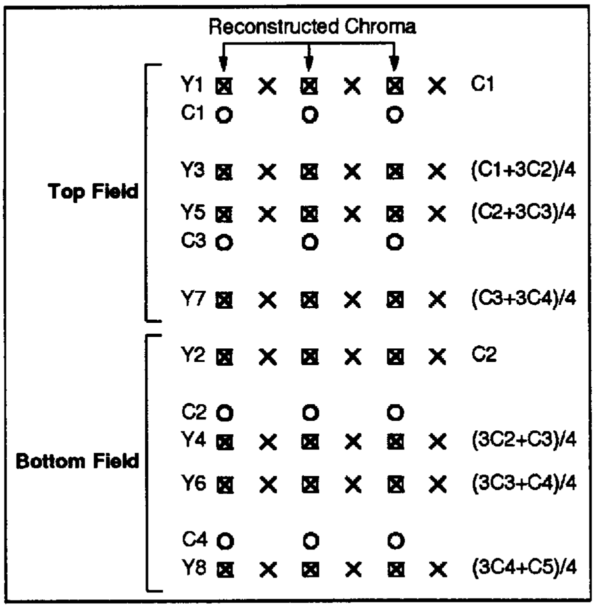


Mode 2 : Full Resolution, Chrominance Field Repeat with Interpolation

This mode is used when the MPEG-2 picture-layer bitstream variable "progressive_frame" is 1, indicating that the chrominance samples in each field were derived from both fields, and when interpolation is required. Figure 38 shows how the chrominance is reconstructed in the two fields for 4:2:2 output. The luminance is unaffected by the vertical filter.

The chrominancefor the last line of the bottomfield (line 480 in a 480-line frame, or line 576 in a 576-line frame) is equal to the last line of chrominance input.

Figure 38 : Derivation of Chrominance in Mode 2


Reconstructed Chroma
Top Field
Y1 ☒ X ☒ X ☒ X C1
C1 O O O
Y3 ☒ X ☒ X ☒ X (C1+3C2)/4
Y5 ☒ X ☒ X ☒ X (C2+3C3)/4
C3 O O O
Y7 ☒ X ☒ X ☒ X (C3+3C4)/4
Bottom Field
Y2 ☒ X ☒ X ☒ X C2
C2 O O O
Y4 ☒ X ☒ X ☒ X (3C2+C3)/4
Y6 ☒ X ☒ X ☒ X (3C3+C4)/4
C4 O O O
Y8 ☒ X ☒ X ☒ X (3C4+C5)/4

3520-39.EPS

Mode 3 : Full Resolution, Chrominance Field Repeat

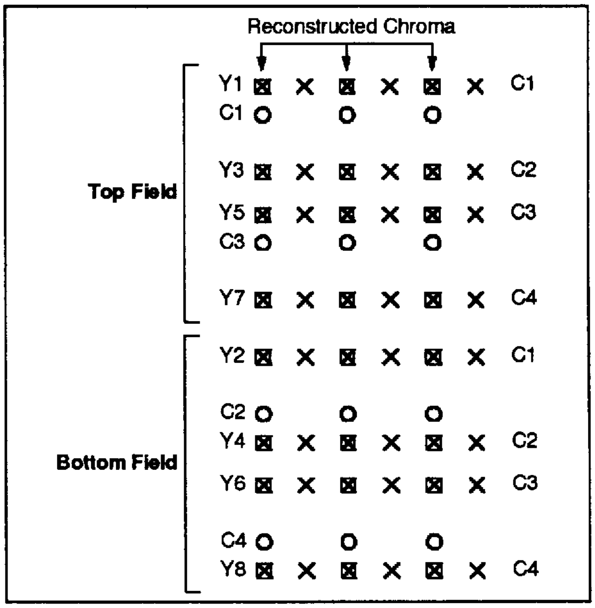
This mode is used when the MPEG-2 picture-layer bitstream variable "progressive_frame" is 1, indicating that the chrominance samples in each field were derived from both fields, and when interpolation is not required. Figure 39 shows how the chrominance is reconstructed in the two fields for 4:2:2 output. The luminance is unaffected by the vertical filter.

Mode 4 : Half Resolution with Chrominance Interpolation

This mode is for the display of pictures having only half the resolution of the display, when the same picture is displayed in both fields, and when interpolation of chrominance is required. The most common application is the display of MPEG-1 SIF pictures on a ITU-R 601 resolution interlaced display. Figure 40 shows how the chrominance is constructed for each identical field. The luminance is unaffected by the vertical filter.

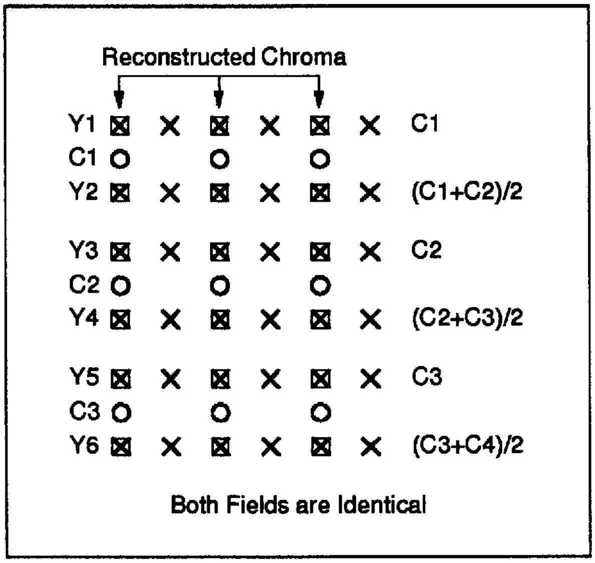
The chrominance output in last line of each field is equal to that output in the second-last line.

Figure 39 : Derivation of Chrominance in Mode 3



3520-40.EPS

Figure 40 : Derivation of Chrominance in Mode 4

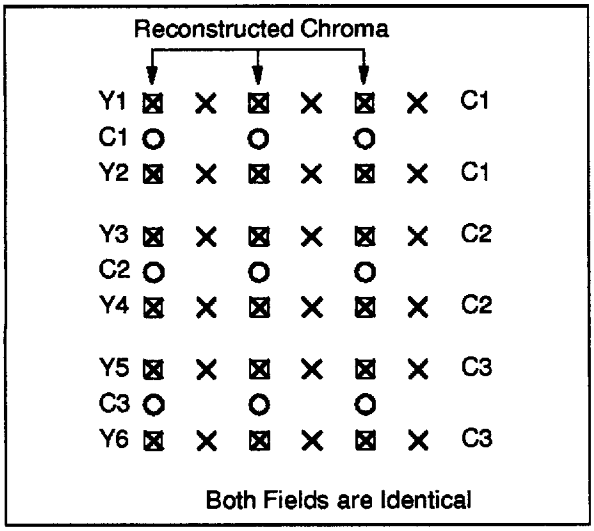


3520-41.EPS

Mode 5 : Half Resolution, Chrominance Repeat

This mode is for the display of pictures having only half the resolution of the display, when the same picture is displayed in both fields, and when interpolation of chrominance is not required. The most common application is the display of MPEG-1 SIF pictures on a ITU-R 601 resolution interlaced display. Figure 41 shows how the chrominance is constructed for each identical field. The luminance is unaffected by the vertical filter.

Figure 41 : Derivation of Chrominance in Mode 5



Mode 6 : Half Resolution with Luminance Interpolation

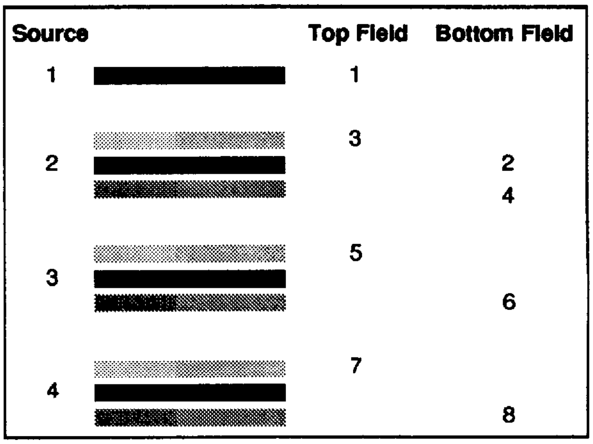
This mode is for the display of pictures having only half the resolution of the display, and when it is required to interpolate two different fields from the luminance lines of the source picture. The most common application is the display of MPEG-1 SIF pictures on a ITU-R 601 resolution interlaced display. Figure 42 shows how the two luminance fields are interpolated.

The interpolation of the luminance lines is performed as follows, where the subscript "s" indicates the source picture :

Top Field	Bottom Field
$Y1 = Y1_s$	$Y2 = Y2_s$
$Y3 = (Y1_s + 3Y2_s)/4$	$Y4 = (3Y2_s + Y3_s)/4$
$Y5 = (Y2_s + 3Y3_s)/4$	$Y6 = (3Y3_s + Y4_s)/4$
...	...
$Y2N\text{-}1 = (YN\text{-}1_s + 3YN_s)/4$	$Y2N = YN_s$

The output chrominance is constructed by duplication. The first line of chrominance is used for lines 1, 2, 3 and 4, the second for lines 5, 6, 7, 8, and so on as in mode 5.

Figure 42 : Luminance Interpolation in mode 6



Vertical Filter Precision

The vertical filter calculation is performed in unsigned arithmetic with full precision and then the 10-bit results are rounded to 8 bits for output. Different rounding rules are used for luminance and chrominance. The rules are as follows :

- for luminance, the results are rounded towards zero, i.e. if the bottom two bits are 00, 01 or 10, the 10-bit number is truncated to 8, while if the bottom two bits are 11, one is added to the truncated 8-bit number.
- for chrominance, the results are rounded towards 128, i.e. if the 10-bit number is larger than 1000000000_2, the rule above is applied, while if the number is less than this, 10_2 is added to the 10-bit number before truncation.

X.4.6 - Decoding and Display across Sequence Boundaries

In certain instances it will be neccessary to maintain seamless transistions accross sequence boundaries. This functionality requires that the decoder can display and decode simultainiously two sequen ces of different size.

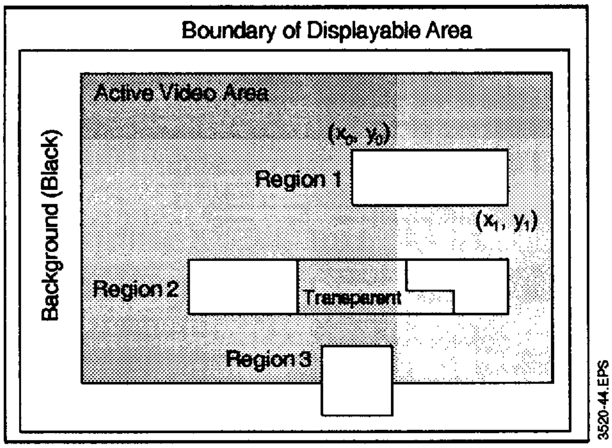
To facilitate implementation of this mode the registers DFW, DFS and DFA have been quadruple buffered. The function is controlled using the GCF2.SQF bit. When the bit is set XFW, XFS and XFA can be programmed. These registers are synchronized on VSync where as the registers DFW, DFS and DFA are synchronized to Dsync. This enables different parameters to be attached to the display and decode processes.

X.4.7 - On-Screen Display (OSD)

X.4.7.1 - OSD Regions

The OSD function can be used to display a user-defined bitmap over any part of the displayable (i.e. non-blanked) screen, independent of the size and location of the active video area (defined by XDO, XDS, YDO, YDS). This bitmap can be defined independently for each field, and is specified as collections of "OSD regions". A region is a rectangular area specified by its boundaries and by a bitmap defining its contents. Each region has associated with it a palette defining four or sixteeen colours which can be used within that region. If required, one of these colours can be "transparent", allowing the background to show through. Figure 43 shows examples of OSD regions.

Figure 43 : OSD Regions

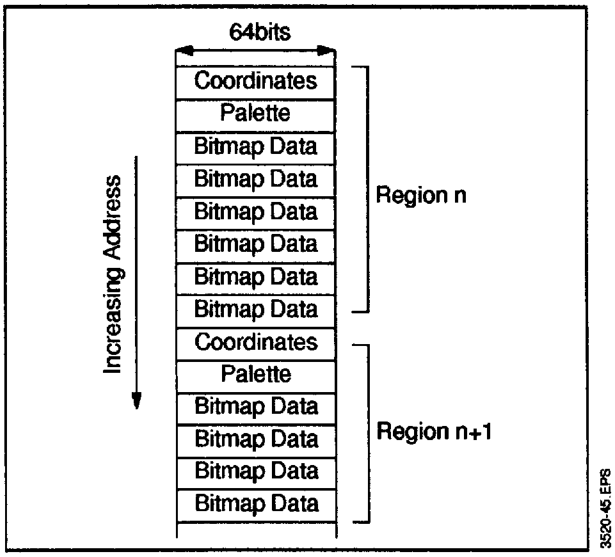


OSD is enabled if bit DCF.EOS is set.

The starting address in memory of the OSD specification for the top field is defined by register OTP, and that for the bottom field is defined by register OBP. OSD specifications are written into the memory using the procedure described in section VII.5, "Memory Read and Write through the Microcontroller Interface". OSD specifications can be rapidly moved in memory using the procedure described in section VII.6, "Block Move".

Figure 44 shows how OSD specifications are stored in memory as a sequence of words. The first word defines the boundaries of the region, the second word defines the palette for the region, and the subsequent words define the colour of every pel in the region. There must not be unused words between the specifications of any two regions within an OSD specification. A display line cannot be included in more than one OSD region ; in other words, only one OSD region can be active on a line (this limitation can be partially overcome by the use of transparency within an OSD region). Within an OSD specification the region specifications must be stored in order of increasing starting line number. The last word in an OSD specification must define a starting line which is beyond the displayable area. Line numbers are the internal (field) line numbers defined in Figure 26. It is thus possible to share the same OSD specification for both fields of a frame. In this case the OTP and OBP registers would be loaded with the same address. The number of OSD specifications which may be resident in memory at any time is limited only by the amount of memory available.

Figure 44 : OSD Buffer Format



Two methods of OSD specification are possible, a mode which is backwards-compatible with the STi3500A, and mode new to the STi3520.

X.4.7.2 - OSD Specification : STi3500 Compatible Mode

The format of the OSD region specification words is given in Figure 45. In this mode, all non used bits must be reset.

In the first word, the positions of the left and right edge samples of an OSD region, counted in numbers of PIXCLK cycles from the falling edge of HSYNC, are defined as follows :

$$\text{left edge position} = 2X_{left} + 9$$

$$\text{right edge position} = 2X_{right} + 10$$

where X_{left} and X_{right} are the values defined in the first word of the OSD region specification. This is illustrated in Figure 46. X_{left} must always have the same parity as the offset loaded into the XDO register (i.e. both must be even or both must be odd). These constraints ensure that the OSD region data samples are always correctly phased with respect to the active video. The first sample output in an OSD region is always a C_B value.

The top line specified in the first word of an OSD region specification must be greater than or equal to 3.

Figure 45 : OSD Region Specification Word Formats

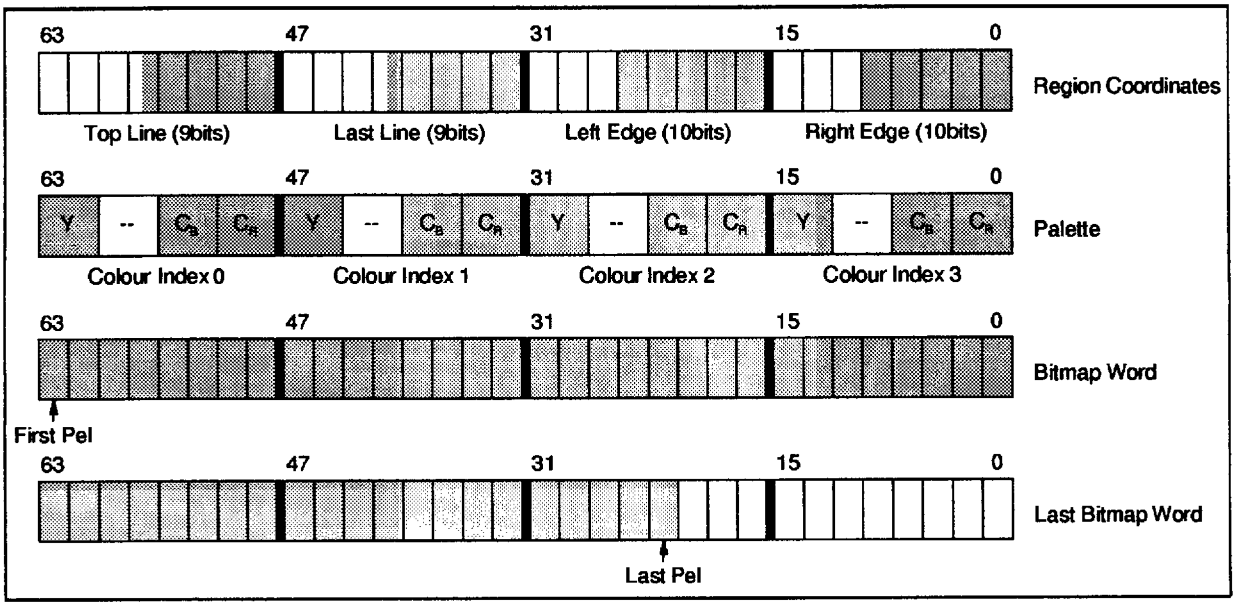
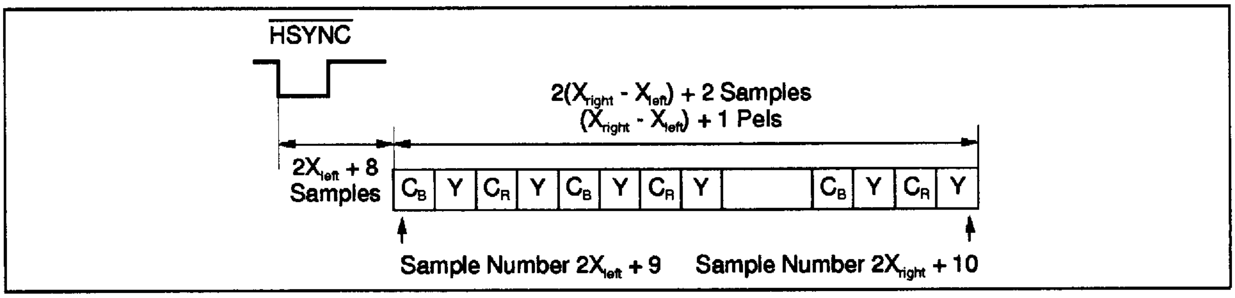


Figure 46 : OSD Region Horizontal Positioning


HSYNC
2(Xright - Xleft) + 2 Samples
(Xright - Xleft) + 1 Pels
2Xleft + 8 Samples
CB
Y
CR
Sample Number 2Xleft + 9
Sample Number 2Xright + 10
3520-47.EPS

The second word defines the palette which applies to the OSD region. Four colours can be defined, each chosen from a set of 4096. The 4 bits of Y, C_R and C_B output from the palette define the top 4 bits of an output sample. The bottom four are set to zero. The "colour" Y = C_R = C_B = 0, in palette index 0 only, defines transparency.

The following words define the bitmap in left to right and top to bottom order. Two bits are used to define the colour index (i.e. palette address) of each pel. The first, third, fifth, etc. bitmap bit-pairs are used to reference all three components (Y, C_B and C_R) of the respective pels ; the second, fourth, sixth, etc. reference only the luminance components of the palette (in the 4:2:2 format chrominance is only defined for every second pel). For this reason the bitmap for a region must define horizontal segments containing a whole number of pel-pairs. It is possible, however, to define a value of X_{right} such that an odd number of pels will be output in a segment. In this case the two bits of the bitmap defining the end of each line segment are redundant. Also, at the transition between OSD and the picture, the C_R chrominance value associated with the first pel of the decoded picture display will be defined by the OSD bitmap, not by the picture.

For the same reason, the transition to and from transparency must only occur at points which are an even number of pels from the start of the left-hand edge of an OSD region.

The following table shows the 4-bit OSD palette Y, C_B, C_R values nearest to the standard "colour bar" colours.

	Y/16	C_B/16	C_R/16
White	1111	1000	1000
Black	0001	1000	1000
Red	0101	0110	1111
Green	1001	0011	0010
Blue	0011	1111	0111
Yellow	1101	0001	1001
Cyan	1011	1010	0001
Magenta	0111	1101	1110

X.4.7.3 - OSD Specification : STi3520 Mode

The OSD of the STi3520 has been upgraded with the addition of three new modes.

- linked list memory management,
- 16 palette mode with 6-bit luma resolution and 4-bit chroma resolution,
- programmable mixing factor (7-bits) for each osd block to blend video and OSD data.

The bit map for the OSD is backwards comptible with the 2 bits/pixel (STi3500) mode. The compatibility is controlled with the M flag, see Table 1 & Table 2 OSD Header Definitions.

If M = 0 (2 bits/pel) the header contains only 8, 16-bit words. If M = 1 (6,4,4 bits/pixel) the header contains 20 16-bit words.

In the compatible mode all the fields which control the new features (Mix Weight,T, OSDp) are supposed to be '0', however, the new features can be used in this mode if required.

OSD Zone Pointer

Each OSD block contains the pointer OSDp[18:0] in the header. The pointer contains the address of the next block to load from memory. The blocks can be anywhere in memory (up to 32 Mbits are addressable), however, the pointer OSDp[18:0] is constrained to be a multiple of 8 (OSD[2:0] = 0). If OSDp[18:0] = 0 then the next OSD block is read contiguously from the memory after the current one (backward compatibility).

The first OSD block of a field is constrained to be at an address multiple of 32 64-bit words as it is defined by the OBT or OTP (previously OEP or OOP) registers.

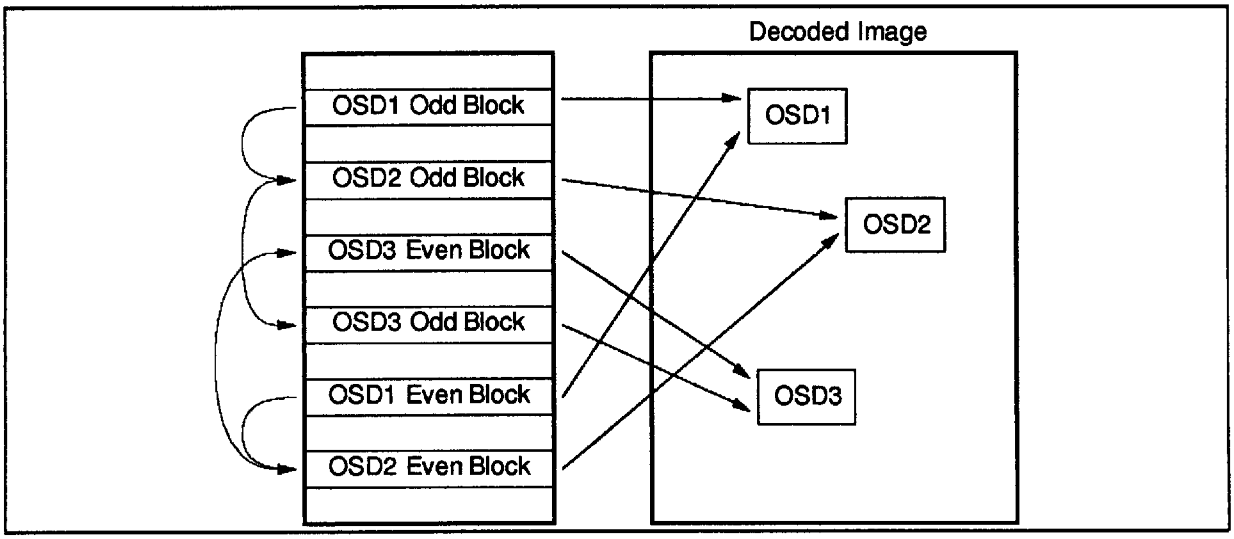
Table 1 : 2 Bits/Pixel OSD Header Definition (M = 0)

OSD Header Definition																
16 Bit Words																Description
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bit Position
M=0	0	0	OSD p[3]	0	0	0	Row Start Position									Header Word 0
MixWeight				OSDp[6:4]			Row Stop Position									Header Word 1
OSDp[12:7]						Column Start Position										Header Word 2
OSDp[18:13]						Column Stop Position										Header Word 3
Palette0 Y						0	T0	Palette0 U				Palette0 V				Header Word 4
Palette1 Y						0	T1	Palette1 U				Palette1 V				Header Word 5
Palette2 Y						0	T2	Palette2 U				Palette2 V				Header Word6
Palette3 Y						0	T3	Palette3 U				Palette3 V				Header Word 7
Bit Map for 8 OSD Pixels																Body Word 0

M	T	Operation
0		2 bit / Pixel
1		4 bit / Pixel
	0	Do NOT blend video with OSD for this colour
	1	Blend video with OSD for this colour with mix weight

3520-03.TBL

Figure 47 : Linked List Structure for OSD Data



3520-48.EPS

Table 2 : 6,4,4 Bits/Pixel OSD Header Definition (M = 1)

OSD Header Definition																
16 Bit Words																Description
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bit Position
M=1	0	0	OSD p[3]	0	0	0	Row Start Position									Header Word 0
MixWeight				OSDp[6:4]			Row Stop Position									Header Word 1
OSDp[12:7]						Column Start Position										Header Word 2
OSDp[18:13]						Column Stop Position										Header Word 3
Palette0 Y						0	T0	Palette0 U				Palette0 V				Header Word 4
Palette1 Y						0	T1	Palette1 U				Palette1 V				Header Word 5
Palette2 Y						0	T2	Palette2 U				Palette2 V				Header Word6
Palette3 Y						0	T3	Palette3 U				Palette3 V				Header Word 7
Palette4 Y						0	T4	Palette4 U				Palette4 V				Header Word 8
Palette5 Y						0	T5	Palette5 U				Palette5 V				Header Word 9
Palette6 Y						0	T6	Palette6 U				Palette6 V				Header Word10
Palette7 Y						0	T7	Palette7 U				Palette7 V				Header Word 11
Palette8 Y						0	T8	Palette8 U				Palette8 V				Header Word 12
Palette9 Y						0	T9	Palette9 U				Palette9 V				Header Word 13
Palette10 Y						0	T10	Palette10 U				Palette10 V				Header Word 14
Palette11 Y						0	T11	Palette11 U				Palette11 V				Header Word 15
Palette12 Y						0	T12	Palette12 U				Palette12 V				Header Word 16
Palette13 Y						0	T13	Palette13 U				Palette13 V				Header Word 17
Palette14 Y						0	T14	Palette14 U				Palette14 V				Header Word 18
Palette15 Y						0	T15	Palette15 U				Palette15 V				Header Word 19
Bit Map for 4 OSD Pixels																Body Word 0

3520-04.TBL

Mixing OSD with Video

The mixing function (schematic shown in Figure 48) allows each OSD value to be blended with the spatially corresponding video pixel. The mix weight is a programmable parameter and can be set for each OSD block. Furthermore, each individual colour in the palette can be designated to be used with or without mixing by setting the 'T' bit of the palette. A 'T' bit equal to 0 designates no mixing for the particular colour and a 'T' of 1 designates that mixing should be used for the particular colour. The mix weight is a 4-bit number allowing mixing ratios from 0 to 1 with a resolution of 1/16 therefore, the resulting pixel can be completely transparent (weighting of 1/16) or can completly cover the video (15/16). Palette zero can also be transparent by setting the YUV values to zero in palette 0. Only palette zero can be used in this way.

X.4.8 - OSD Active Signal

The OSD active signal can be used in two modes. The mode is controlled using DCF.OAM. In the first mode the OSD active signal is configured as an output. In this mode the OSD active signal denotes when an active OSD pixel (non transparent) is on the YC output bus, Figure 50. The signal, in this mode, has a programmable delay controlled by DCF.OAD. This delay can be set such that the OAD active signal is set as much as two clocks before or one clock after the actual pixel.

In the second mode of operation, the OSD active signal is configured as an input and is used to disable the OSD. When the signal goes high OSD will be placed on the YC bus if OSD is enabled. When this signal is low then no OSD will be placed on the bus even if OSD is enabled. The programmable delay is used in the same way as for the input signal.

Table 3 : OSD Active Signal Operation

OSD Active Mode	OSD Active Signal	Meaning
0	0	Signal is an output. Video Pixels only on display bus.
0	1	Signal is an output. OSD Pixels on the display bus.
1	0	Signal is an input. Disable the OSD output.
1	1	Signal is an input. Enable OSD output if available.

X.4.9 - Displayed Field Sequence Control

This function is performed automatically, unless specifically disabled by the user. By default the display field output sequence is governed by the input signal B/$\overline{T}$; the bottom field is output when it is high, and the top field is output when it is low. When bit DCF.USR is set, the user has direct control over which field is to displayed after the occurrence of VSYNC ; the built-in modes are disabled. For example in an interlaced sequence it is possible to program the same field to be displayed twice if desired.

Control over the order in which picture data is read from the picture buffer is given by bits DCF.FLD and DCF.DAM[2:0]. These bits must not be programmed independently of bits DCF.VFC[2:0], which define the vertical filter mode. The allowed combinations are detailed below. The DCF register is double buffered, and new values written are taken into account on the occurrence of a VSYNC. The values of bits DCF.FLD and DCF.DAM[2:0] are not taken in account when bit DCF.USR is reset.

Figure 48 : OSD/Video Mixing Detail

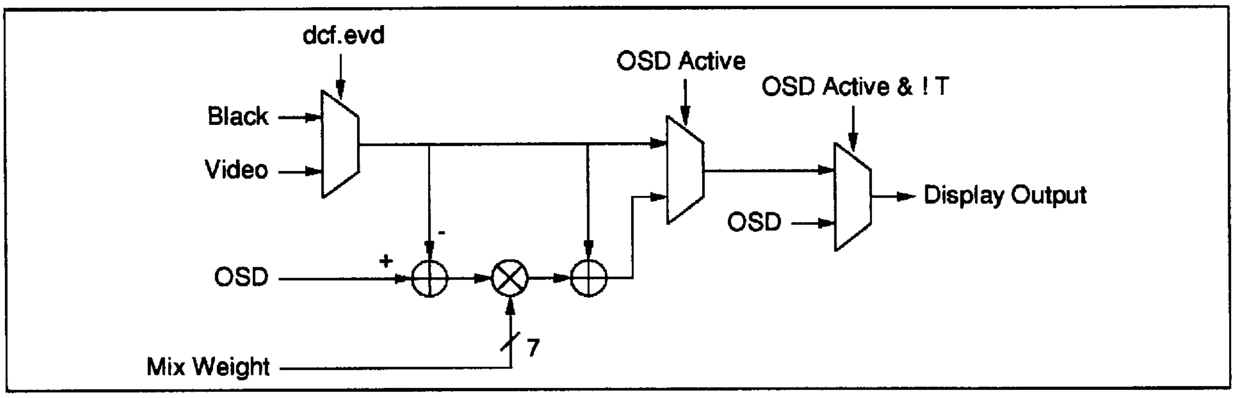


Figure 49 : OSD Active Timing when DCF.OEM = 1

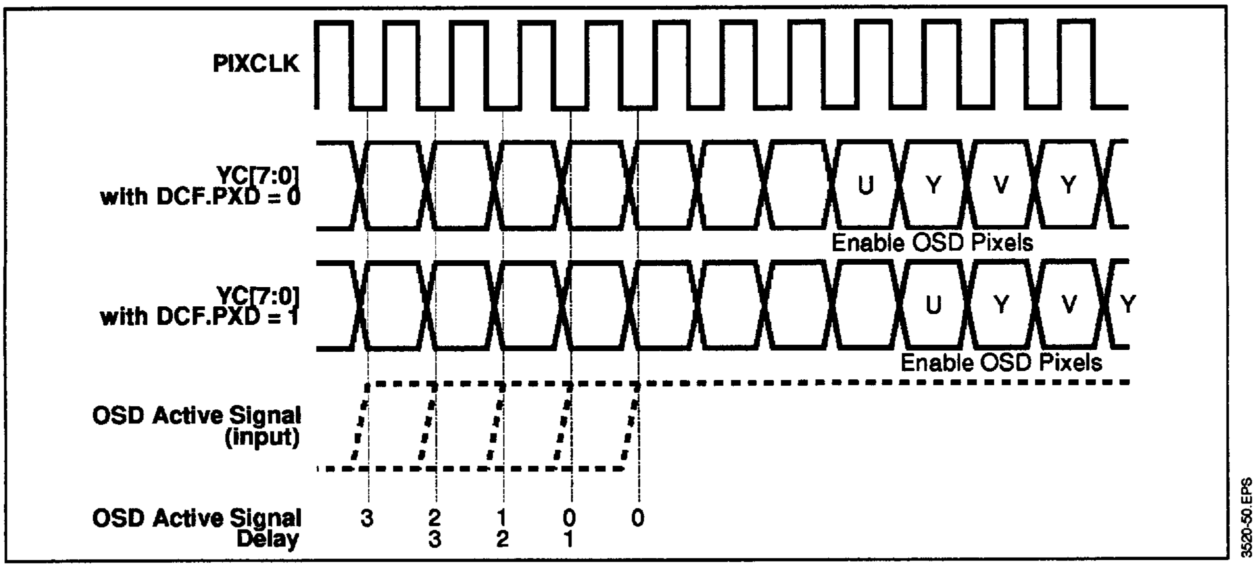
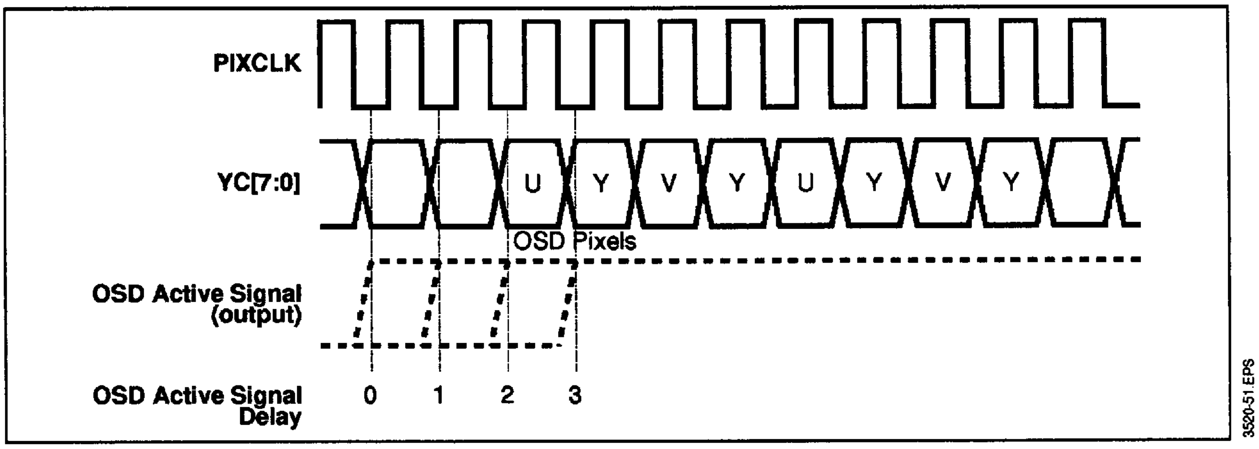


Figure 50 : OSD Active Timing when DCF.OEM = 0



DCF.VFC[2:0]	Field	DCF.FLD	DCF.DAM[2:0]
0	Top	0	6
	Bottom	1	6
1	Top	0	2
	Bottom	1	2
2	Top	0	3
	Bottom	1	7
3	Top	0	3
	Bottom	1	3
4	Both	0	4
5	Both	0	0
6	Top	0	0
	Bottom	1	0

The value of bit DCF.FLD controls which field is to be displayed : 1 for the bottom field, 0 for the top field. The signal B/$\overline{T}$ is thus overridden.

The allowed combinations are given in this table.

These values must be updated every field.

These bits do not affect the display of OSD. OSD field selection is under the exclusive control of the B/$\overline{T}$ Pin.

An application of user display sequence control is when the displayed picture must be frozen. In this case the same field can be output continuously in order to reduce flicker.

X.5 - 8-MBit Mode

X.5.1 - General

8-Mbit mode can be used for decoding smaller (up to approximately half-ITU-R 601) pictures without any restrictions on prediction modes, or for decoding larger pictures with certain restrictions on prediction modes and motion vector range.

The setting up of 8-Mbit mode is explained in section VII.1, "Standard DRAM", and the data storage structure in memory is described in section VII.4, "Picture Storage Data Structure".

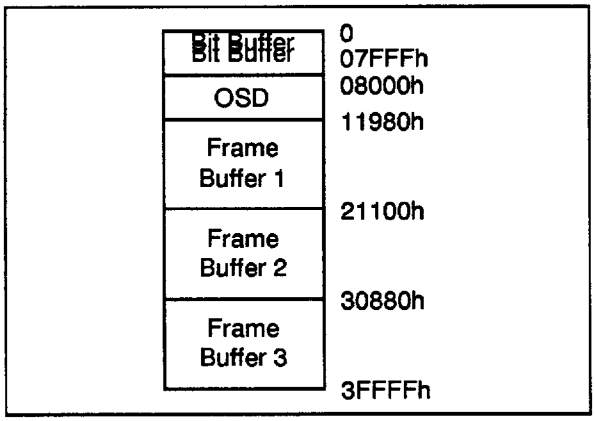
The physical memory address range is the same as that in normal (16-Mbit) mode, i.e. 00000h to 3FFFFh, but each memory word contains 32 bits instead of 64 bits.

X.5.2 - Unrestricted Decoding

Decoding can be performed in 8 Mbits without any restriction on decoding modes, provided that all of the required picture buffers, the bit buffer and any OSD area required can be fitted into the 8 Mbits of memory available. Figure 51 show how 3 frames of size 352 x 480, together with a 1-Mbit bit buffer and an OSD buffer area can be placed in 8 Mbits of memory. The pointer values for the frame buffers are 1126, 2116 and 3106.

The BFP/FBP register is loaded with the value of the backward prediction frame pointer. The foldback function is never activated since memory addressing never passes the end of memory, 3FFFh.

Figure 51 : 352 x 480 Frames in 8 Mbits



3520-52.EPS

X.5.3 - Memory Folding Principle

A memory folding option is available in 8-Mbit mode. This allows the decoding of forward-predicted pictures of size too large to allow the storage of 2 frames and the bit buffer in 8 Mbits of memory. This folding is invoked automatically in 8-Mbit mode whenever memory addressing passes the end of memory.

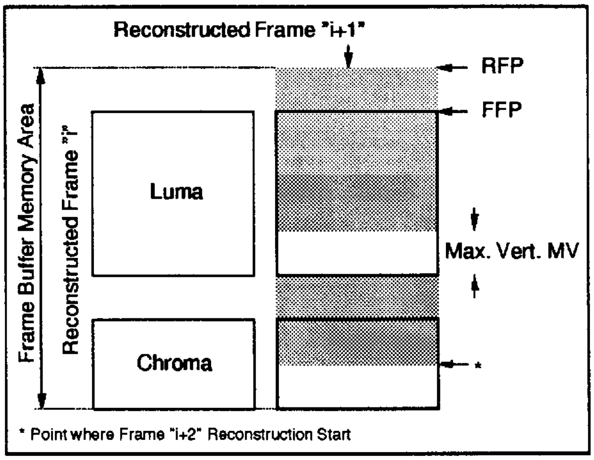
Memory is conserved by overwriting the last reconstructed picture with the one currently being decoded, and rotating memory addresses around a predefined frame buffer area. This is illustrated in Figure 52.

Reconstructed frame "i+1" is written into an area of memory starting at a lower address than the start of the reconstructed picture "i". Note that there must be a gap between the luminance and chrominance storage areas to prevent overwriting of the end of the luminance area of picture "n" with chrominance of picture "n+1". This gap is equal to the difference in the starting addresses of the two frame buffers. The size of this gap also defines the maximum negative vertical motion vector possible.

The size of the gap, expressed in macroblocks, allows the value of GCF.DFA[7:0] to be calculated. An example calculation is given in section XIV, "VIDEO REGISTERS".

In addition, the register BFP/FPB needs to be set up with the start address of the frame buffer area.

Figure 52 : Memory Reuse in 8-Mbit Mode



* Point where Frame "i+2" Reconstruction Start

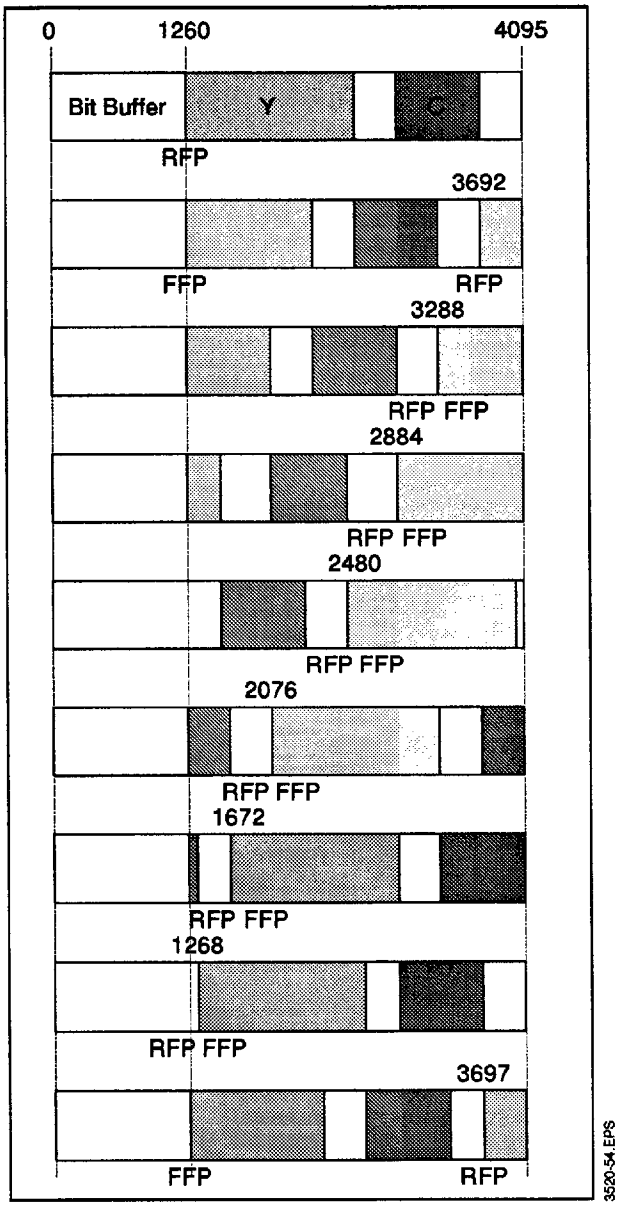
3520-53.EPS

This is the address to which fold-back occurs when memory addresses exceed the 8-Mbit limit. When folding is invoked, BFP must not be used ; backward prediction is thus not allowed.

The amount of memory required for the frame buffer area is equal to the size of one frame buffer plus twice the gap defined by DFA.

The example (Figure 53) illustrates the evolution of memory usage during the decoding of a sequence of pictures of size 720 x 480. The motion vector range is -128 to 127, and the gap required between luma and chroma is thus 404 macroblocks (giving a value of GCF.DFA of 101).

Figure 53 : Buffer Sequencing Example in 8-Mbit Mode



The value of 1260 for the start of the picture buffer area is calculated as follows :

Total amount of memory = 4096 x 64 32-bit words (pointer addresses are in units of 64 32-bit words in 8-Mbit mode).

Memory for one frame buffer = 720 x 480 x 12 bits = 2025 x 64 32-bit words.

2 DFA gaps = 2 x 404 x 4 x 64 x 8 bits = 808 x 64 32-bit words.

4096 - 2025 - 808 = 1263, the space left for bit buffer and OSD.

In this example the buffer states are shown after reconstruction of the picture into the buffer pointed to by RFP. The forward predictors are fetched from the buffer pointed to by FFP. The pointer address values given are the values loaded into the RFP register. The algorithm used to calculate pointer value is the following :

- First frame :
  RFP = first allowed starting address beyond bit buffer
  FBP = first allowed starting address beyond bit buffer
- Subsequent frames :
  FFP = RFP
  RFP = RFP - 4 x GCF.DFA
  If RFP < FBP then RFP = RFP + 4096 - FBP

X.5.4 - Performance in 8-Mbit Mode

STi3520 video decoding performance is lower in 8-Mbit mode since the reduced memory data bus width reduces the available data transfer bandwidth.

For picture sizes which allow 3 frame buffers to be stored in the memory, all prediction modes can be supported.

For larger picture sizes, where memory folding is required, only forward prediction is possible. In this case, for ITU-R 601 sized pictures, there is a limitation on the number of dual-prime macroblocks which can be decoded in any picture.

X.6 - Video Decoding Control

This section explains in outline how the STi3520 video decoder must be controlled in order for it to perform real-time decoding. Only the principles are given here ; more detail is available in the STi3500/3520 Application Notes.

X.6.1 - Initialization

After a power-on hard reset, certain registers must be set up before decoding can start. These concern :

- general configuration : enable interfaces, set memory mode and refresh interval (CTL and GCF).
- bit buffer : set up BBS.
- interrupt unit : read ITS to clear all pending interrupts. Set up ITM.
- memory pointers : set up RFP for first decoded picture.
- display : set up display window (XDO, XDS, YDO, YDS). Set up OSD if required. Disable display.

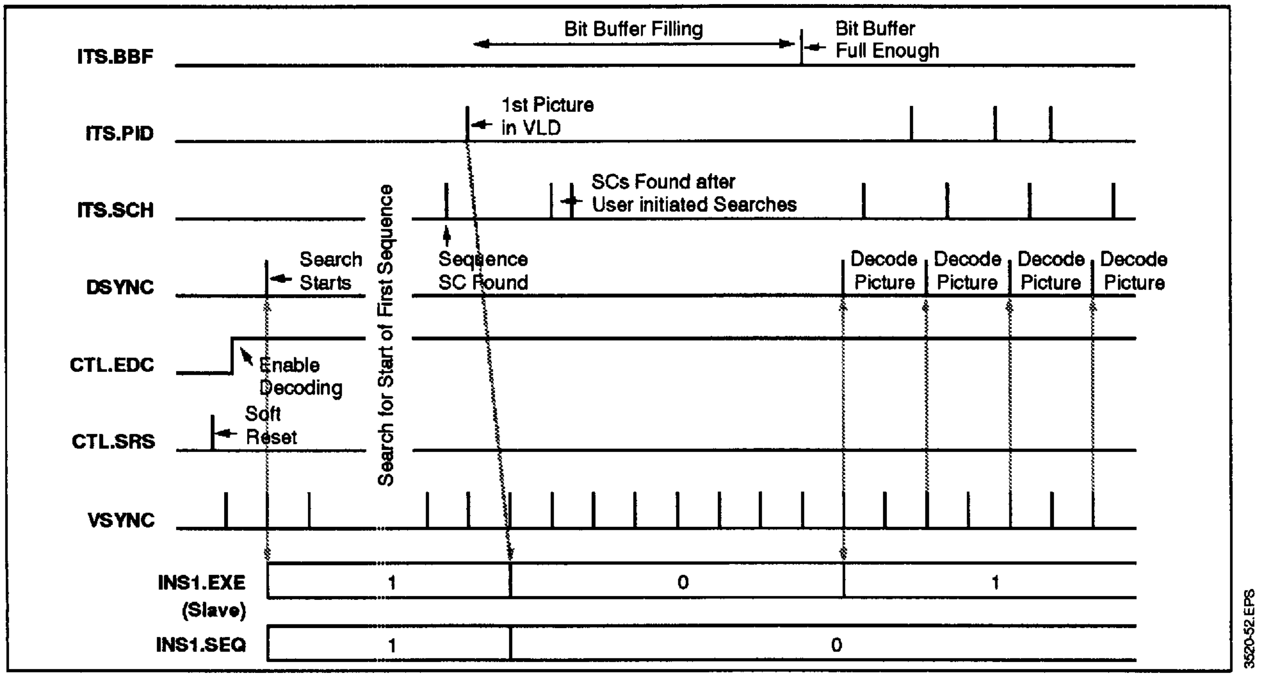
Parameters concerning the format of the decoded picture will be read from the first sequence header.

Figure 54 shows the events which occur during the start-up of decoding the new sequence. This would occur after a hard or a soft reset. (in the diagram a soft reset is shown).

After the reset, the instruction bits INS1.EXE and INS1.SEQ are set, and decoding is enabled by setting bit CTL.EDC. When the next VSYNC occurs, the pipeline starts its task and a DSYNC is generated. The latter event starts the start code detector. When the DSYNC has been detected (indicated by the PSD interrupt), the instruction bits INS1.EXE and INS1.SEQ are written as zero. This will take effect on the next "new instruction" event (see section X.3.4, "Decoding Task Control").

The search for the beginning of the new sequence now takes place - this is the only possible action after a hard or soft reset. When the start code detector has found a sequence start code it stops and an SCH interrupt is generated. When the pipeline has found the first picture header after a sequence header, it too stops and a PID interrupt is generated. On the next VSYNC the new instruction is executed, but since this has INS1.EXE = 0, the pipeline waits. During this searching process the bit buffer does not fill ; all data entering the bit buffer is transferred immediately to the pipeline. However, when the pipeline has found the start of the sequence, the bit buffer starts to fill. During this time the sequence header can be read from the header FIFO and analysed. The decoder set-up can be completed and the quantization tables loaded. The start codes following the first sequence header can be detected by launching additional start code searches (using command CMD.HDS). This will include the first picture start code, following which is the picture header containing the information needed to complete the first decoding instruction. The bit buffer threshold, BBT, can now be calculated from the "vbv_delay" parameter and set up.

Figure 54 : Initialization of a Sequence



When the BBF interrupt occurs, there is enough data in the bit buffer for decoding to commence. The decoding instruction for the first picture is now loaded, with bit INS1.EXEset. On the next VSYNC, the instruction is executed, and a DSYNC is generated. The correct phasing of the first picture decode with respect to the top and bottom fields can be ensured by monitoring the VST and VSB interrupts before loading the first instruction. A start code detection is launched in response to the DSYNC, enabling the start code for the next picture to be found and analysed. When decoding is complete, a PID (pipeline idle) interrupt is generated.

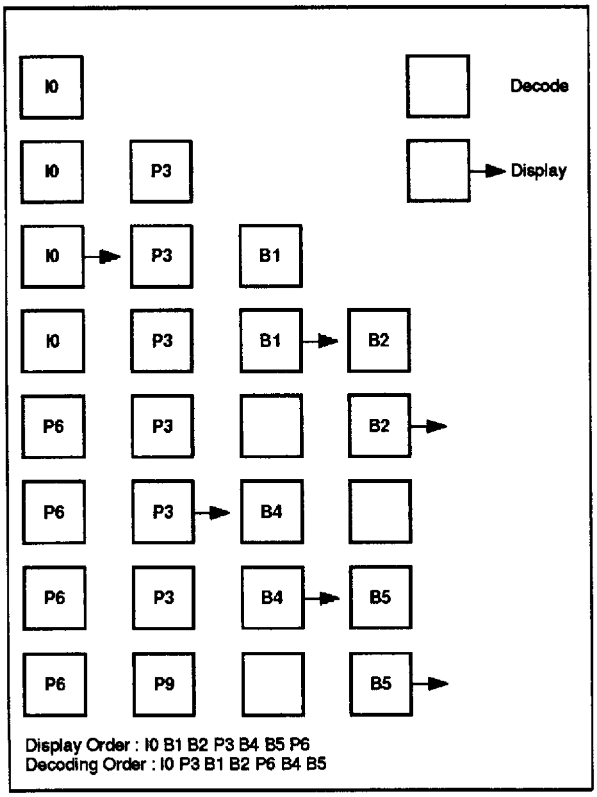
This pattern of events is now repeated until the end of the sequence.

X.6.2 - Buffer Sequencing and Overwrite Mode

Before the decoding of each picture, when the picture header has been analysed, the memory addresses pictures buffers from which the predictors are to be fetched, the buffer in which the pictures is reconstructed, and the display buffer must be set up.

The use of the different buffers is described in section X.3.3, "Utilization of PIcture Pointers".

Figure 55 : Allocation of 4 Picture Buffers for Decoding of Sequence with M = 3



Display Order : I0 B1 B2 P3 B4 B5 P6
Decoding Order : I0 P3 B1 B2 P6 B4 B5

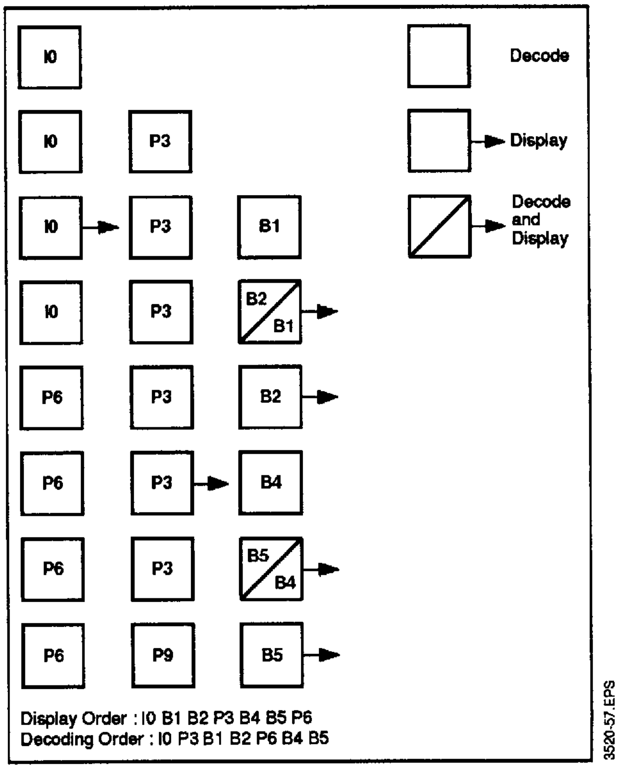
The pointer to the displayed picture, DFP, is updated on every VSYNC. The pointers to the buffers used by the pipeline BFP, FFP and RFP, are updated when a DSYNC occurs, i.e. at the start of decoding.

Figure 55 shows one way of allocating picture buffers for decoding a sequence in which there are 2 B-pictures between I- or P-pictures (M = 3). Four buffers are used, since in the worst case - the decoding of a B-picture while the previously decoded B-pictureis being displayed - two buffers are required for the reference pictures, one for the picture currently being decoded, and one for the picture being displayed.

Figure 56 shows how the same sequence can be decoded using only three picture buffers. This is made possible by using the same picture buffer for reconstruction and display (i.e. RFP = DFP) when the displayed picture does not have to be saved, which is the case for B-pictures. The displayed picture is thus overwritten by the picture being constructed.

Overwrite Mode is enabled for a picture decoding operation by setting bit INS1.OVW of the instruction, and setting RFP and DFP to the same value.

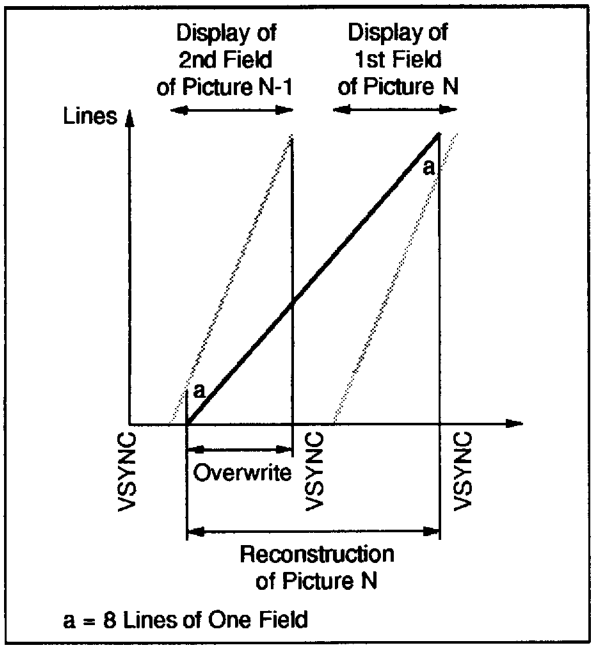
Figure 56 : Allocation of 3 Picture Buffers for Decoding of Sequence with M = 3 with Overwrite Mode Selected



Display Order : I0 B1 B2 P3 B4 B5 P6
Decoding Order : I0 P3 B1 B2 P6 B4 B5

Figure 57 shows how overwrite mode operates when using the same buffer for reconstruction and display of an interlaced picture.

Figure 57 : Picture Buffer Overwrite



3520-58.EPS

The reconstruction of a picture, which is performed macroblock-by-macroblock, is overlapped with the display of the second field of the previous picture and the display of the first field of the current picture. Overwriting cannot start until 8 lines have been displayed ; this frees one row of macroblocks in the buffer for writing. Reconstruction must be complete 8 lines before the end of the display ; this ensures that the last line of macroblocks has been written. There is an automatic mechanism which ensures that the overwriting of reconstructed data never overtakes the display process.

When decoding a picture with overwrite mode enabled, there is less than a full frame period available for decoding. If the decoding of a picture can not be completed in this time, the end of the display would not be updated, but the decoding sequence would not be disturbed, since reconstruction would still be completed before the next picture decoding starts (see section X.3.4, "Decoding Task Control").

X.6.3 - Decoding/Display Synchronization

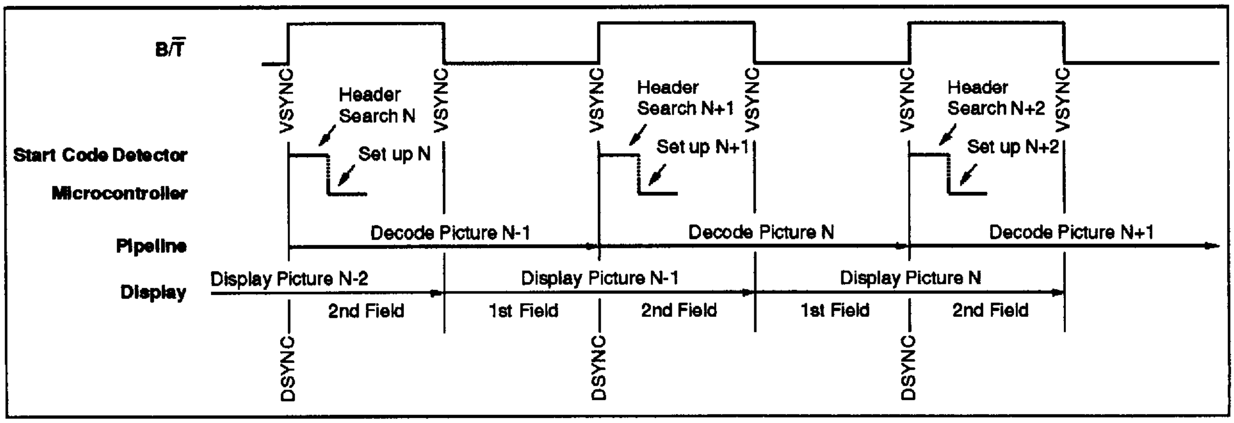
Figure 58 shows how picture decoding and display are synchronized in the most common application of the STi3520 : full resolution pictures, interlaced display with overwrite mode enabled when decoding and displaying B-pictures. It is assumed in this diagram that the decoding and display order is the same. This does not affect the principle of operation.

The bit INS1.RPT is set in every instruction ; this ensures that the decoding task duration is two VSYNC intervals. A DSYNC is generated on every second VSYNC.

While a picture is being decoded the start code detector scans through the same picture data in order to locate the start code of the following picture. When this has been detected the microcontroller can read the header data and set up in advance the parameters which will be taken into account on the next DSYNC.

When decoding half-resolution pictures, the same information must be read from the display in both fields. If overwrite mode is used, then the timing will be the same as that shown in Figure 58. Bit CTL.HRD must be set since overwriting cannot commence until 16 lines have been displayed.

Figure 58 : Decoding/Display Synchronization : Full Resolution, Interlaced Display, Overwrite Mode



3520-59.EPS

X.6.4 - 3:2 Pull-Down Operation

For display of a progressive sequence coded at 24 frames/sec on a 30 frames/sec display, pictures must be alternately displayed for periods of 2 and 3 fields. Figure 59 shows how decoding and display are synchronized in this application. (as before it is assumed in this diagram that the decoding and display order is the same).

For one VSYNC period in five, no decoding operation is required. In these periods the pipeline is put into a waiting state by defining an instruction with INS1.EXE = 0 and INS1.RPT = 0.

X.6.5 - Control in 8-Mbit Mode

X.6.5.1 - Overwrite Mechanism

In 8-Mbit mode the display buffer overwrite can work in two modes. One mode for use with a circular buffer when the 8 Mbits is not enough to allow space for the bit buffer and the frame buffers. The second mode is for use without a circular buffer when the 8 Mbits is sufficient space for the bit buffer and frame buffers. The selection of the circular buffer is controlled using CTL.CBC.

Overwrite With Circular Buffer

If memory folding is used (because 8 Mbits are not enough in which to do unrestricted decoding), overwrite.

mode must always be activated and control must be done as described in this section.

The operation of the overwrite in this mode has three phases, illustrated in Figure 60.

Phase 1

The STi3520 generates DSYNC (and a PSD interrupt) and latches the new instruction. The reconstruction of the new picture can start into the free memory area whose size is equal to GCF.DFA X 4 macroblocks. Reconstruction will stop when this area is filled. Note that if GFA.DFA = 0, no data are reconstructed during this phase.

During this phase the first field of the previous picture is displayed.

Phase 2

Reconstruction will continue with the constraint that memory locations still required for the display of the

Figure 59 : Decoding/Display Synchronization in 3:2 Pull-Down Mode with Overwrite

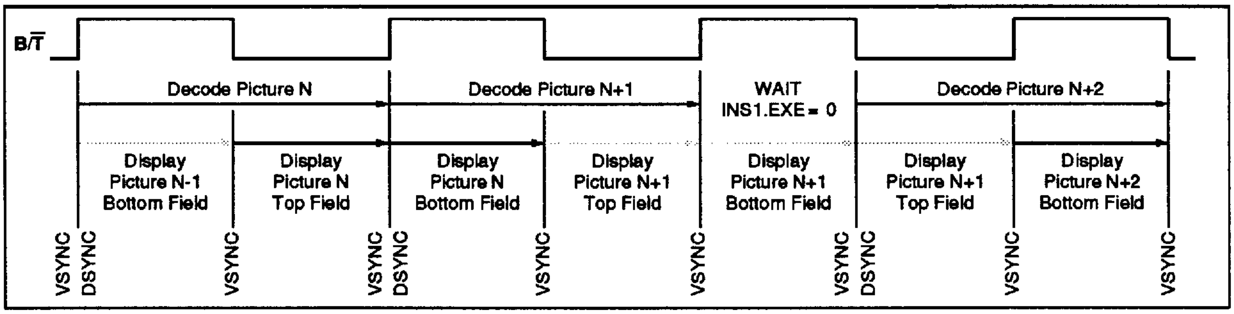
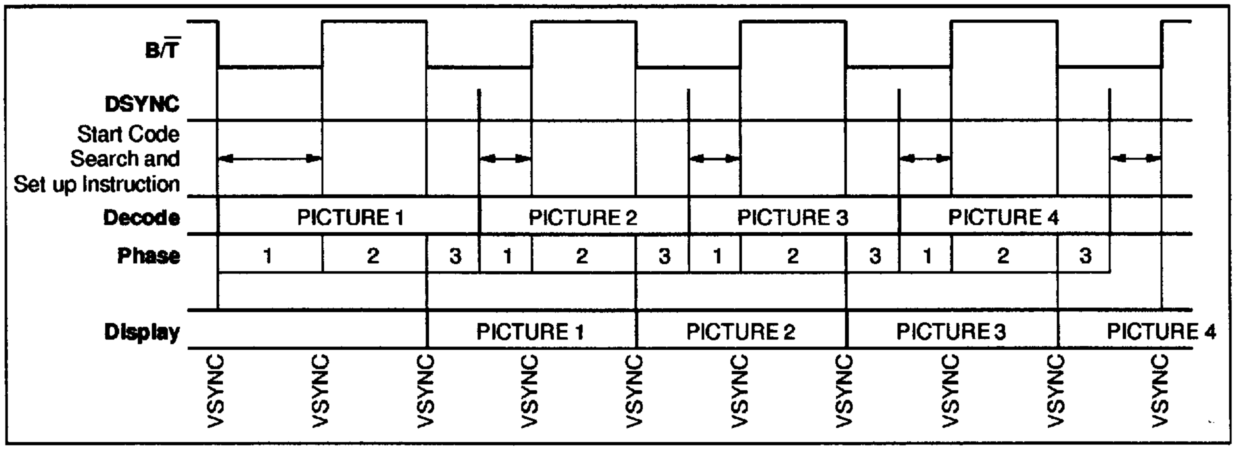

B/T
Decode Picture N
Decode Picture N+1
WAIT
INS1.EXE = 0
Decode Picture N+2
Display Picture N-1 Bottom Field
Display Picture N Top Field
Display Picture N Bottom Field
Display Picture N+1 Top Field
Display Picture N+1 Bottom Field
Display Picture N+1 Top Field
Display Picture N+2 Bottom Field
VSYNC
DSYNC
3520-60.EPS

Figure 60 : Decoding/Display Synchronization in 8-Mbit Mode



second field of the previous picture will not be overwritten ; the condition:

decode address < display address + GCF.DFA x 4

is satisfied.

At the end of this phase the picture will not be completely decoded.

Phase 3

With the overwrite constraint removed, decoding will proceed as quickly as possible up the end of the picture. The end of decoding can occur several milliseconds after VSYNC. At this point a new DSYNC is generated., the next instruction is latched and a new decoding "Phase 1' starts for the next picture. Note that some bits of the instruction (INS1.OVW, INS1.SEQ, INS1.EXE, INS1.RPT and CMD.SKP[1:0]) are latched on VSYNC, while the others are latched on DSYNC which occurs several milliseconds later (cf. section X.3.5, "Task Overrun").

Except for the very first picture, the complete decoding task will take roughly 2 VSYNC periods, as in normal mode, except that decoding is shifted with respect to VSYNC. All tasks are delayed instead of starting immediately.

X.6.5.2 - Decoder Control

The consequence of the changes in the overwrite mechanism is that decoding control must be modified ; the decoding process must start two fields before the first display instead of one. If the display is done top-field-first (as in the example of Figure 60), then the decoding task must also start on top fields.

The following points must be noted :

- the instruction can only be computed after the next start code is detected. As start code detection is started automatically by DSYNC, the header will only be available in the middle of the VSYNC period. The time available for the microcontroller to compute the instruction is less than normal. The instruction must be set up before the next VSYNC which leaves less than one and half fields.
- the DFP register is not set up at the same time as before - it must not be changed on DSYNC or on start code detection but only after the following VSYNC (i.e. at the start of decoding Phase 2).

Unrestricted Decoding

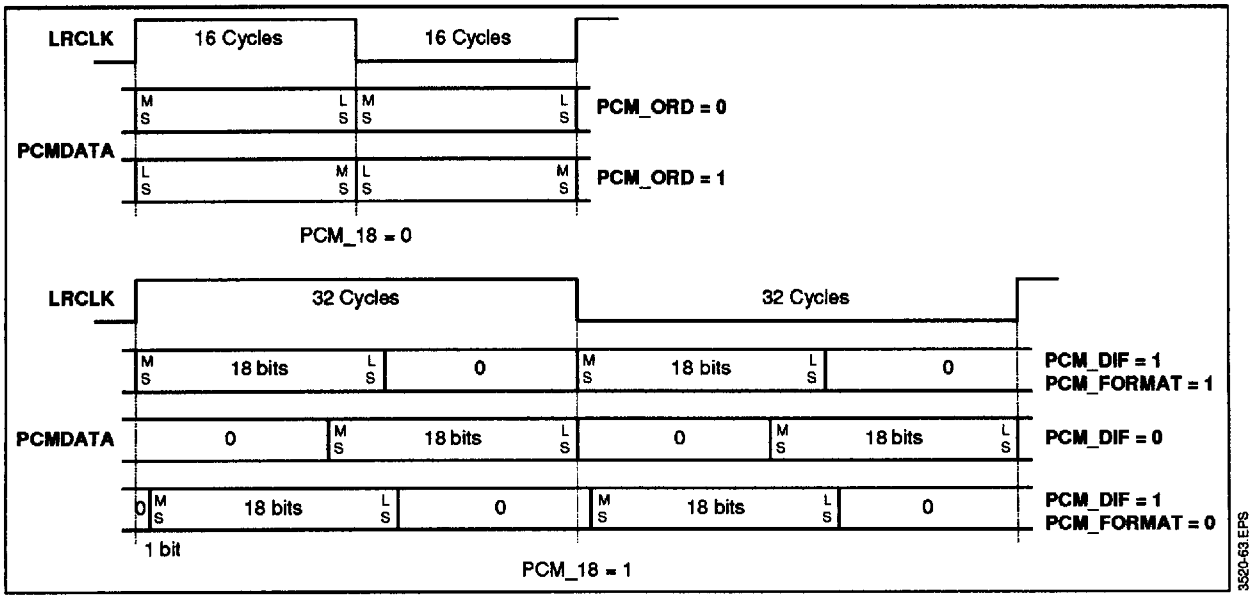
In this case the 8 Mbits of memory allow enough space for the bit buffer and the frame buffers. The control can be done in three different ways :

- overwrite mode always off. In this case the control of the STi3520 is exactly as in standard mode without overwrite.
- overwrite mode always on + Non-Circular Buffer. In this case the control is performed as in standard mode with overwrite.

It is not recommended to switch the overwrite control bits (INS1.OVW or CTL.CBC) on and off while decoding since the control phase depends on overwrite and may become difficult to manage.

Note that when DFP is changed in Phase 2, it must be changed to the value of the current RFP, not to the value of the next one which has at this time been pre-computed.

Figure 61 : PCM Output Formats



XI - AUDIO DECODER CIRCUIT DESCRIPTION

XI.1 - PCM Output

XI.1.1 - Interface and Output Formats

The decoded audio data is output in serial PCM format. The interface consists of the following signals :

Name	Function
PCMDATA	PCM Serial Data Output
SCLK	PCM Clock Output
LRCLK	Left/Right Channel Select Output
PCMCLK	PCM Clock Input

Output precision is selectable to be either 16 bits/word or 18 bits/word by setting the output precision select, PCM_18, register. In 16-bit mode, data may be output either with the most significant bit first or least significant bit first as selected by the contents of the output order select, PCM_ORD, register. When 18-bit data is selected, 32 bits are output for each channel. The data in front register, PCM_DIF, is used to position the 18 data bits either at the beginning or at the end of each 32-bit frame. The PCM_FORMAT register is used to select standard or I^2S compatible format when 18-bit precision is selected.

Figure 61 shows the five different output formats which are possible. PCM_ORD only has significance in 16-bit mode. PCM_DIF only has significance in 18-bit mode. PCM_FORMAT only has significance in 18-bit mode and when PCM_DIF = 1. The last option shown in Figure is compatible with the I^2S format.

The polarity of the PCM serial output clock, SCLK, and the left/right channel selection, LRCLK, are selected by bits INV_SCLK and INV_LRCLK, respectively.

Figure 62 shows the two polarities of SCLK. Normally, the DAC will sample LRCLK and PCMDATA on the rising edge of SCLK in the first case, and on the falling edge of SCLK in the second. The first option (INV_SCLK = 0) is the one normally used in I^2S systems.

Figure 62 : SCLK Polarity

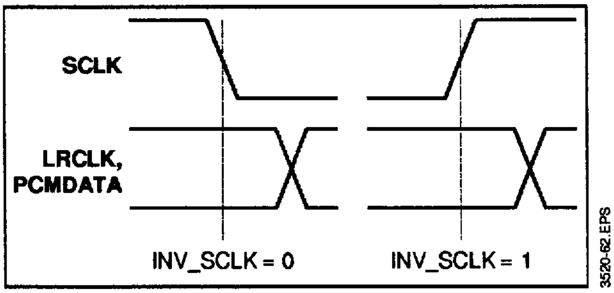
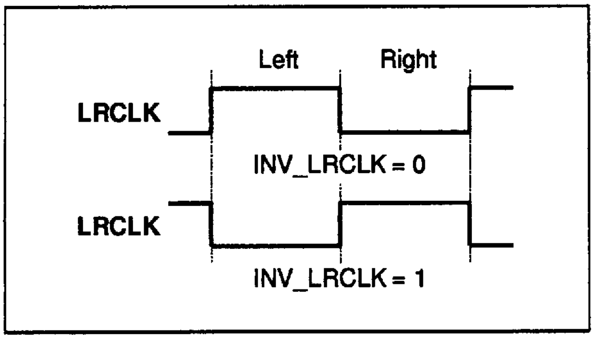


Figure 63 shows how the polarity of LRCLK is selected. The second option (INV_LRCLK = 1) is compatible with the I^2S format.

PCM interface timing is given in Figure 83.

Figure 63 : LRCLK Polarity



XI.1.2 - PCM Clock Generation

The PCM serial clock SCLK is derived from the clock input PCMCLK. The frequency of PCMCLK may be equal to the PCM output bit rate, or it may be an integer multiple of this, allowing the use of oversampling D-A converters. In many applications PCMCLK is externally synchronized to the compressed audio bit stream.

SCLK is derived from PCMCLK by dividing it by the contents of the divider register, PCM_DIV.This number, in the range 0 to 63, defines the ratio of the frequency of the PCM bit clock, SCLK, to that of PCMCLK, according to the relationship :

$$f_{SCLK} = \frac{f_{PCMCLK}}{2 \cdot (PCM\_DIV + 1)}$$

For example, PCM_DIV is loaded with 0, the frequency of SCLK is one half of the frequency of PCMCLK, while if PCM_DIV is loaded with 63, the frequency of SCLK is one 128th of the frequency of PCMCLK.

The value of PCM_DIV = 16 is reserved. If this number is loaded, the divider is bypassed and the frequency of SCLK is equal to the frequency of PCMCLK.

PCM_DIV must be set up before the output of SCLK starts. This can be done by first disabling PCM outputs by de-asserting the MUTE and PLAY commands, and then writing to the PCM_DIV register. Once the register is set up, the MUTE and/or PLAY commands can be asserted. PCM_DIV cannot be changed "on the fly".

The frequency of LRCLK is given by :

$$f_{LRCLK} = \frac{f_{SCLK}}{32} \text{ for 16-bit PCM output}$$

$$f_{LRCLK} = \frac{f_{SCLK}}{64} \text{ for 18-bit PCM output}$$

XI.1.3 - Interrupts Associated with PCM Output

There are two interrupts associated with the PCM output, interrupt 8, "PCM Buffer Underflow", and interrupt 14, "Output of New Frame".

An interrupt 8 is generated (if not masked) when a new output sample is required and the PCM buffer is empty. The PCM buffer, which contains up to 64 samples (i.e. 64 word-pairs in stereo), receives the decoded outputs from the DSP core. If the buffer is empty the output sample will have the value zero. Decoding will not stop. If the PCM buffer becomes full, decoding will stop, but PCM output will not be affected.

An interrupt 14 is generated (if not masked) whenever the first bit of a frame appears at the PCM output.

XI.2 - Audio Decoder Control

XI.2.1 - Control and Status Pins

$\overline{\text{AUDPTS}}$ becomes active when the first data word associated with a frame that contained a PTS (presentation time stamp) is at the PCM output stage. $\overline{\text{AUDPTS}}$ is active for a duration of 1ms. The PTS associated with the current frame may be read from PTS[32:0]. This register is updated only if its associated interrupt request is enabled.

The action of $\overline{\text{RESET}}$ is described in the following section.

XI.2.2 - Initialization of the Audio Decoder

There are two methods of initiating a reset of the STi3520 audio decoder :

- writing 0 or 1 (after which it is automatically restored to the 0 state) to the register RESET,
- asserting the $\overline{\text{RESET}}$ Pin for a duration of at least 200ns.

Either of these actions will reset the INTR and INTR_EN registers. In addition, asserting the $\overline{\text{RESET}}$ Pin will reset the registers INV_SCLK, MUTE, PLAY, STC_INC and STC_CTL. All other registers must be set up by the microcontroller before decoding is started.

A reset, initiated either by RESET Pin or by register write initiates the following actions :

- the RESET register is set (if not set already),
- the REQ Pin goes high,
- the INTR, INTR_EN and BUFF_LEV registers are reset,
- the registers INV_SCLK, STC_INC, STC_DIV and STC_CTL are cleared (only if reset was activated by RESET Pin),
- all data buffers are cleared. This takes multiple clock cycles,
- the MUTE and PLAY registers are reset (only if reset was activated by RESET Pin). This inhibits the output clocks, LRCLK and SCLK, and places PCMDATA in their inactive state,
- all other control registers remain at their pre-existing state,
- the STi3520 terminates the reset cycle. The RESET register is cleared, and the REQ Pin goes low. The PCM output clocks and data remain inactive.

Register accesses by the host interface are not disabled during the reset process. However, while the REQ Pin is asserted audio data cannot be input.

The IRQ and IRQOD outputs are low while the RESET Pin is asserted.

Changing of layer or sampling frequency requires a prior reset of the decoder.

XI.2.3 - Play and Mute

Once initialized and configured, decoding and output of PCM data is controlled by the commands PLAY and MUTE.

The command PLAY is asserted when the PLAY register is written to.

The command MUTE is asserted when the MUTE register is written to.

The actions of the PLAY and MUTE commands are specified in the Table 4.

XI.2.4 - Restart

The restart procedure is invoked when it is required to flush all buffers and restart decoding immediately.

Restart is initiated by writing 0 or 1 (after which it is automatically restored to the 0 state) to the RESTART register. A restart initiates the following actions :

- the REQ Pin goes high,
- the INTR and INT_EN registers are cleared,
- the BUFF_LEV register is cleared,
- all data buffers are cleared,
- the MUTE, PLAY and all others registers (except those mentioned above) remain in their existing state,
- registers access is not disabled. However, while the REQ Pin is high audio data cannot be input,
- the STi3520 terminates the restart cycle. The RESTART register is cleared, and the REQ Pin goes low.

The DRAM does not go through the power-up refresh cycle during the restart sequence.

Table 4 : Mute and Play Functions

Mute	Play	Function
De-asserted	Asserted	No output or decoding. SCLOCK, LRCLK, PCMDATA all move into their inactive state. LRCLK completes its current cycle and stops, SCLK completes its last cycle in the second LRCLK frame and stops. Decoding stops when all internal buffers become full.
De-asserted	Asserted	Normal decoding and PCM output. When PLAY is re-asserted, PCMDATA resumes where it left off without data loss.
Asserted	De-asserted	PCM clocks only, no decoding. PCMDATA becomes low after the output of the last complete sample. LRCLK and SCLK are not stopped. Decoding stops when all internal buffers become full. When PLAY is re-asserted, PCMDATA resumes where it left off without data loss.
Asserted	Asserted	Decoding and muted output (soft mute). PCMDATA gradually decays to zero. Decoding continues normally. Data consumed as if output were playing.

3250-05.TBL

XI.2.5 - Bitstream Synchronization

The compressed input bit stream must be synchronized before the decompression step may begin. This is done by looking for synchronization words inserted into the data stream at encoding. Synchronization must be done both at the audio frame and at the system packet layer if present.

At the packet level, the audio decoder will look for a valid start code, doing a bit by bit search. Once an audio packet is found, it extracts the presentation time stamp (PTS), if present, and starts the audio synchronization described below.

At the audio frame level, there is a non-unique sync word at the beginning of the header. The STi3520 attempts to find this sync word by doing a bit by bit search. When found the action taken depends on the contents of the SYNC_LOCK and LATENCY registers.

XI.2.5.1 - Packet Level Synchronization

The complete algorithm is given in Figure 64.

To help the synchronization algorithm ignore an emulated packet synchronization word, an extension of the packet start code to be matched is possible. Depending to the content of registers PACKET_SYNC, AUDIO_ID and AUDIO_ID_EN, synchronization can be made on the 24-bit packet_start_code_prefix or can be extended to the stream_id field.

Synchronization mode depends on the type of packets received by the STi3520. The decoder can receive either :

- Multiplexed audio/video bitstream (PACKET_SYNC = 0) :
  In this case the STi3520 can receive both video and audio streams multiplexed together. Packet synchronization is possible only on the 24-bit start code.
  All packets are used by the synchronization algorithm but all non-audio packets and, if AUDIO_ID_EN is set, all audio packets which have a stream_id which does not match the AUDIO_ID register value, are not decoded.
- Multiplexed audio bitstream (PACKET_SYNC = 1) :
  In this case, the STi3520 expects to receive only multiplexed audio streams. Synchronization is performed on 27 bits (24 bits packet_start_code_prefix + 3 first bits of stream_id).
  All packets are used by the synchronization algorithm but if AUDIO_ID_ENis set, all audio packets that have a stream_id which does not match the AUDIO_ID register value are not decoded.
- Single audio bitstream (PACKET_SYNC = 2, AUDIO_ID_EN=1) :
  Synchronization is performed on 32 bits.
  All packets are used by the synchronization algorithm, and all audio packets that have a stream_id which matches the AUDIO_ID register value are decoded.

The SYNCHRO_CONFIRM register is also taken into account in the global synchronization algorithm.

If SYNCHRO_CONFIRM = 1, after the first packet synchronization word is found the STi3520 is considered to be synchronized. If SYNCHRO_CONFIRM = 0, after the first packet synchronization word is found, the STi3520 must read the packet length and confirm synchronization by finding the next synchronization word in the correct position.

XI.2.5.2 - Audio Frame Synchronization

The synchronization algorithm is given in Figure 65.

Because the audio syncword can be emulated in the bitstream, it is useful to extend this audio start code to avoid the detection of a false sync word. Each time the STi3520 detects a false sync word during the synchronization process, the delay to reach the locked state increases.

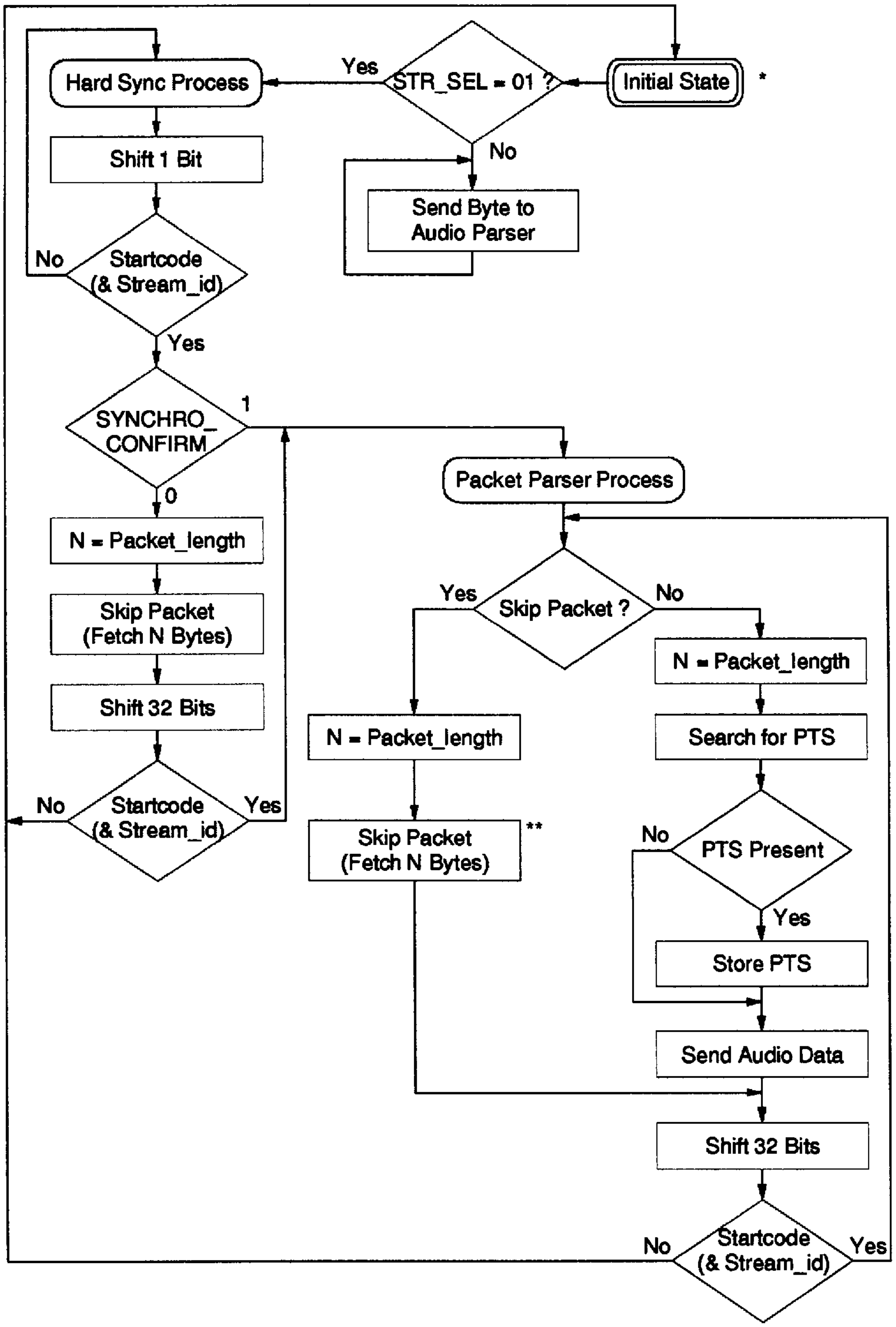
The SYNC_REG register is used for this purpose. When no field of the SYNC_REG register is enabled, the STi3520 saves the layer and sampling frequency information after synchronization has been achieved. This aids the task of resynchronization, should synchronization be lost owing to an error in the audio data or the system layer. This internal register is disabled on RESET or RESTART and will not be reinitialized until the audio parser is synchronized.

The SYNC_LOCK register specifies how many valid synchronization words after the initial one have to be found before entering the locked state. The highest value of SYNC_LOCK (i.e. 3) is assumed when the SYNC_REG register has its default value. The definition of a valid synchronization word depends on the LATENCY register value.

In high latency mode (LATENCY = 1) a valid synchronization word is a sequence of bits that matches the expected word. In addition, the calculated length of the audio frame must be equal to the distance to the start of the next sequence of bits matching the synchronization word.

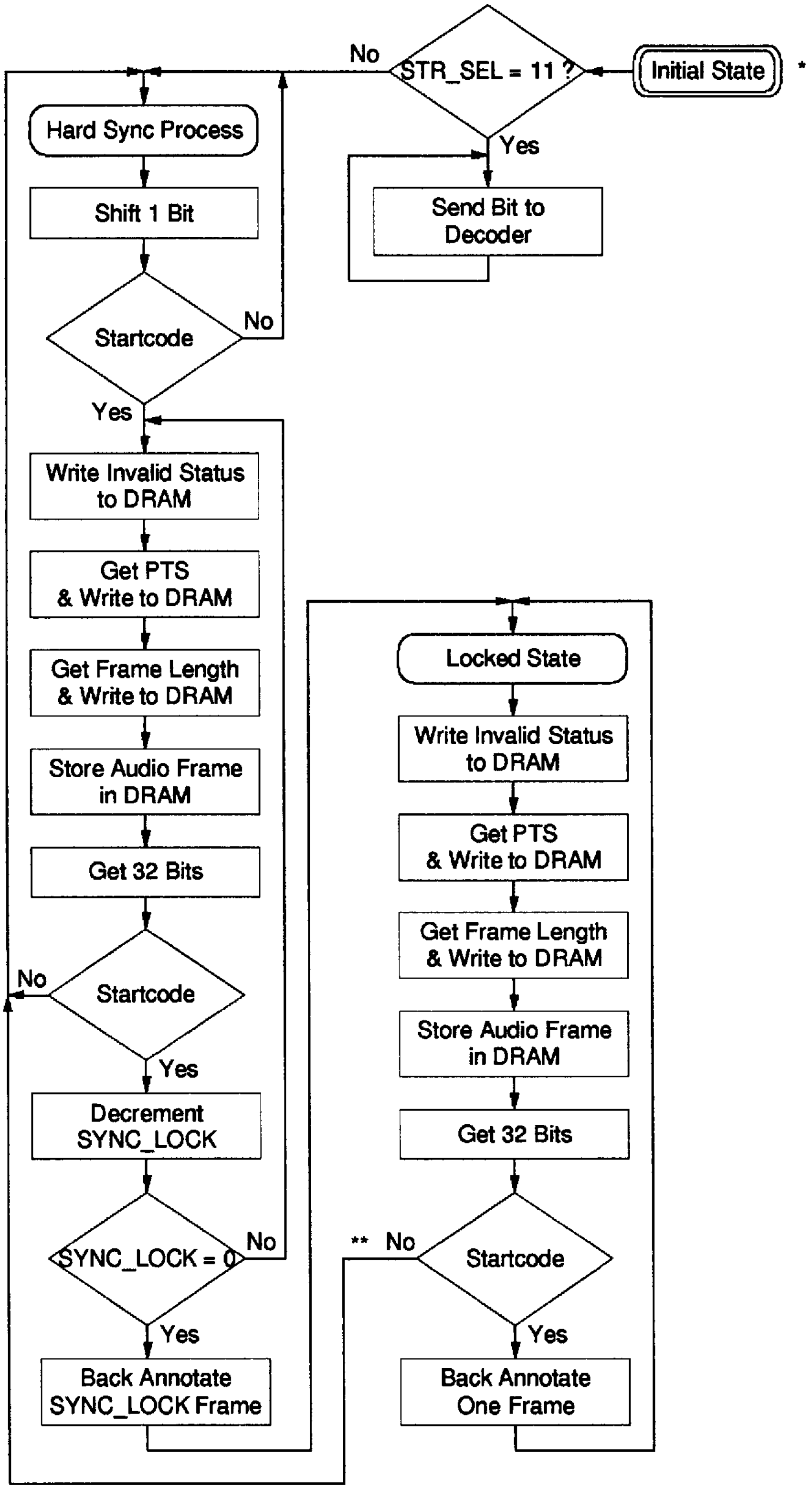
A valid synchronization word is a sequence of bits matching the expected word.

Figure 64 : Packet Synchronization Algorithm



* On reset, packet parser goes to "Initial State"
** Skip video or audio packet that does not have Stream_id

Figure 65 : Audio Frame Synchronization Algorithm



* On reset, audio parser goes to "Initial State"

** On loss of packet sync, audio parser goes to "Hard Sync" process

In free-format mode one additional register (FREE_FORM) can be used. The FREE_FORM register is a way of specifying the length of an audio frame in free-format mode. This register is 16 bits long and contains the length of the frame in bits.

By default a frame is always written always "bad" status. The back annotation process writes a "good" status word. A frame with a "bad" status word is decoded using the concealment method specified by the value of SYNC_ECM register.

The algorithm without DRAM is similar except that back annotation is not possible. In this case frames are lost before the locked state is achieved.

XI.2.6 - Error Recovery and Concealment

The STi3520 audio decoder is able to recover from certain detectable errors. For this purpose it has a number of user-selectable error concealment modes.

Detectable errors may be caused by a bad audio frame CRC or by loss of synchronization. Concealment is similar, but may be selected independently by setting the CRC_ECM and SYNC_ECM registers.

The register CRC_ECM defines the action which will be taken upon detection of a CRC error in an input frame.

00 : Disable CRC detection and error concealment.
01 : Mute on detection of CRC error.
10 : Illegal.
11 : Skip invalid frame.

The register SYNC_ECM defines the action which will be taken upon detection of a synchronization error.

00 : Ignore error.
01 : Mute on detection of synchronization error.
10 : Illegal.
11 : Skip invalid frame.

XI.2.7 - Ancillary Data Extraction

The ancillary data which may be held at the end of audio frames can be extracted and read from the ANC register. This register constitutes a 32-bit FIFO. The first bit of ancillary data received is stored in bit ANC[0].

The extraction of ancillary data in ANC is started by enabling interrupt 7. An interrupt 7 is generated when either :

- 32 bits of ancillary data have been received from the bitstream and written into ANC, i.e. when it is full, or,
- the end of a frame is reached.

Register ANC_AV holds the number of bits available in the ancillary data buffer, ANC[31:0].

When ANC[31:24] is read, interrupt 7 is cleared, ANC_AV is cleared and the ancillary data buffer is reinitialized.

Decoding stops if the STi3520 tries to write data into ANC when it is full. The normal response would be to read ANC_AV and then ANC. However, if interrupt 7 is disabled (by resetting bit INTR_EN[7]), decoding will continue and the registers ANC and ANC_AV will retain their contents until ANC[31:24] is read.

If ANC is not read at the end of the frame, and it is not full, ancillary data bits in the next frame will be appended.

XI.2.8 - Bypass Mode

The STi3520 has an audio bypass capability that allows 16-bit PCM data to be loaded directly and passed through to the PCMDATA output.

To set up audio bypass operation the STR_SEL register must be set to the bypass mode (STR_SEL = 3) and then a reset or restart sequence must be executed. While the $\overline{REQ}$ output is low, PCM data can be loaded into the device directly. The data is loaded in exactly the same way as compressed data, using one of the three loading mechanisms available. The data can be input at up to the maximum burst rate of 20 Mbit/s.

Data must be input in the sequence : 2 bytes left, 2 bytes right, 2 bytes left, and so on. In each byte-pair, the most significant byte must be loaded first. All 16-bit PCM output modes can be used in bypass mode. The volume control is operational in bypass mode.

The PCM underflow interrupt (interrupt 8) can be used to detect when all bypass data has been output.

To switch back to compressed data input, the STR_SEL register and the LATENCY register must be changed back to select the normal input configuration (STR_SEL = 0 or 1, LATENCY = 0 or 1) and then a reset or restart sequence must be executed.

XII - ELECTRICAL CHARACTERISTICS

XII.1 - Absolute Maximum Ratings

Maximum limits indicate where permanent device damage occurs. continuous operation at these limits is not intended and should be limited to those conditions specified in section XII.2, "DC Electrical Characteristics".

Symbol	Parameter	Min.	Max.	Unit
V_{DD}	Power Supply			V
V_I, V_O	Voltages on Input and Output Pins	-0.3	V_{DD} + 0.3	V
T_{stg}	Storage Temperature	-65	150	°C
T_{oper}	Ambient Operating Temperature	0	70	°C

3250-06.TBL

XII.2 - DC Electrical Characteristics

Operating conditions : V_{DD} = 3.3V ±0.3%, T_{amb} = 0 to 70°C unless otherwise specified.

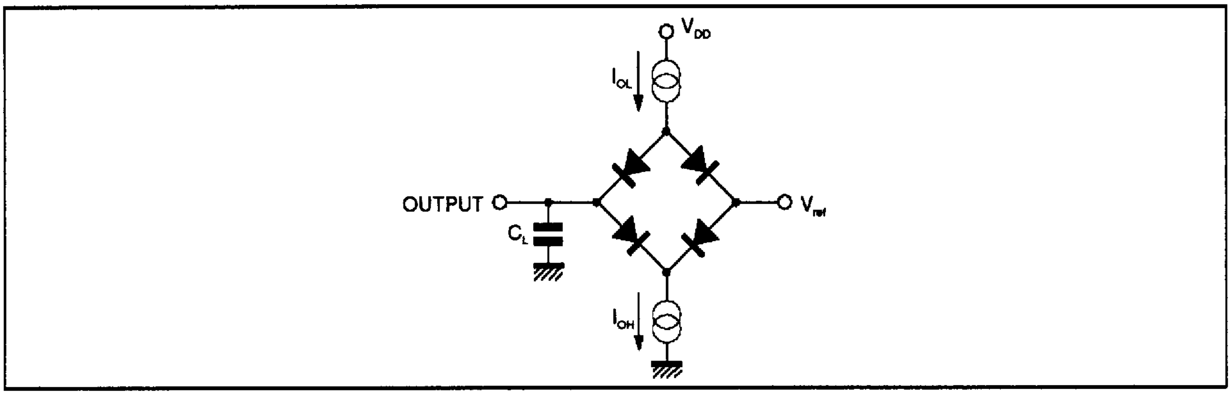
Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_{DD}	Operating Voltage		3.0		3.6	V
I_{DD}	Average Power Supply Current	C_{LOAD} = 50pF on all outputs $f_{primary}$ = 24MHz, all inputs at V_{DD} or 0V $f_{primary}$ = 0Hz			TBD TBD	mA
V_{IL}	Input Logic Low Voltage				0.8	V
V_{IH}	Input Logic High Voltage		2			V
	Input Leakage Current Inputs I/Os	V_{DD} = TBD V, $0 \leq V_{IN} \leq V_{DD}$	TBD TBD		TBD TBD	µA µA
V_{OL}	Output Logic Low Voltage	I_{LOAD} = TBD			0.4	V
V_{OH}	Output Logic High Voltage	I_{LOAD} = TBD	2.4			V
C_{IN}	Input Capacitance	V_{offset} = TBD V, f = 1MHz			TBD	pF

3250-07.TBL

XII.3 - AC Electrical Characteristics

Test Conditions : V_{DD} = 3.3V ±0.3%, T_{amb} = 0 to 70°C, unless otherwise specified.

Figure 66 : Test Load Circuit



Test Loads

Output	I_{OL}	I_{OH}	C_L	V_{ref}
YC7-YC0, D7-D0, $\overline{CDREQ}$, $\overline{WAIT}$				
$\overline{IRQ}$				
$\overline{OE}$				
$\overline{WE}$, $\overline{CAS}$				
$\overline{AA8}$ - $\overline{AA0}$				
$\overline{RAS1}$, $\overline{RAS0}$, DD15-DD0				

Transition Times

Symbol	Parameter	Min.	Max.	Unit
t_T	Transition Time between 0.8V and 2.4V for all Outputs			ns

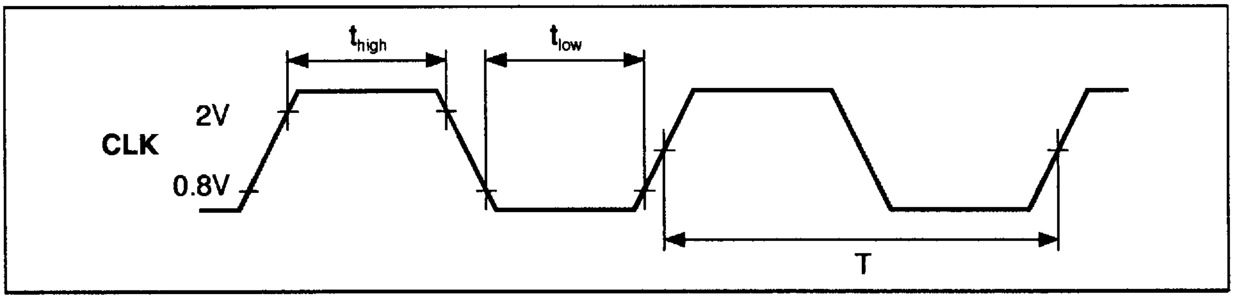
3250-08.TBL

XIII - TIMING DIAGRAMS

Timings other than rise and fall times are specified with respect to a threshold of 1.5V.

XIII.1 - Clock

Figure 67 : Clock Signals



Symbol	Parameter	Min.	Max.	Units
T	Primary Clock Period (1)	20		ns
t_{high}	Clock High Time	8		ns
t_{low}	Clock Low Time	8		ns

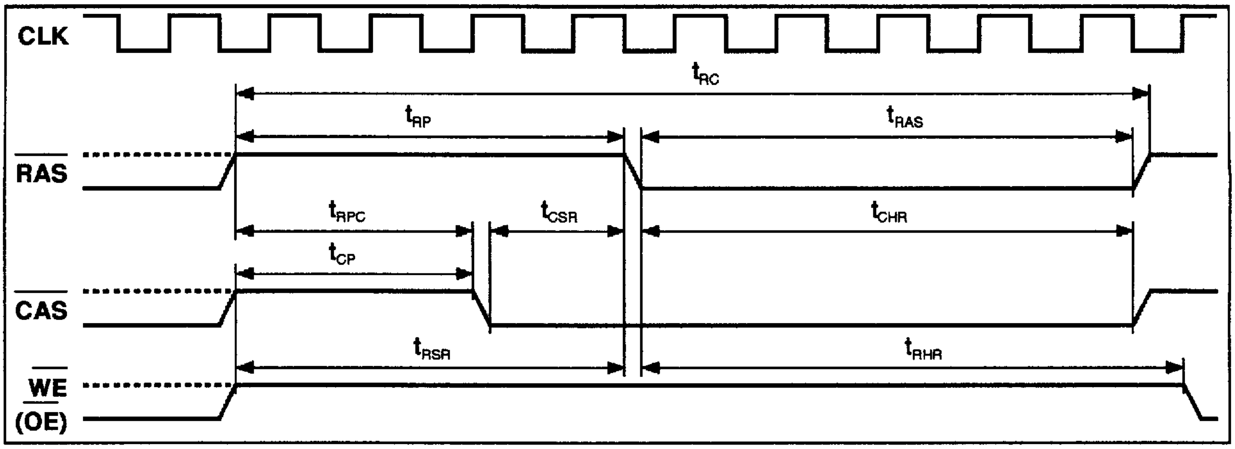
3250-09.TBL

Note : 1. This corresponds to a maximum primary clock frequency of 50MHz.

XIII.2 - DRAM Interface

For the following, T is the primary clock period.

Figure 68 : $\overline{\text{CAS}}$ Befor $\overline{\text{RAS}}$ Refresh Cycle



Symbol	Parameter	Min.	Max.	Units
t_{RC}	Cycle Time	9T		ns
t_{RP}	$\overline{\text{RAS}}$ Precharge Time	4T-9		ns
t_{RAS}	$\overline{\text{RAS}}$ Pulse Width	5T-9		ns
t_{RPC}	$\overline{\text{RAS}}$ to $\overline{\text{CAS}}$ Precharge Time (2)	3T-8		ns
t_{CSR}	$\overline{\text{CAS}}$ to $\overline{\text{RAS}}$ Set-up Time	T-6		ns
t_{CHR}	$\overline{\text{CAS}}$ from $\overline{\text{RAS}}$ Hold Time (1)	5T-10		ns
t_{CP}	$\overline{\text{CAS}}$ Precharge Time	3T-2		ns
t_{RSR}	Read Command to $\overline{\text{RAS}}$ Set-up Time (2)	4T-10		ns
t_{RHR}	Read Command from $\overline{\text{RAS}}$ Hold Time (3, 4)	11T-5		ns

3250-10.TBL

Notes :
1. Worst case is with one bankof memory, and each package having a single $\overline{\text{CAS}}$ Pin.
2. Sometimes referred to as t_{WSR}.
3. Sometimes referred to as t_{WHR}.
4. Worst case is with one bankof memory, and each package having a single $\overline{\text{WE}}$ Pin.

Figure 69 : Page Mode Read Cycle

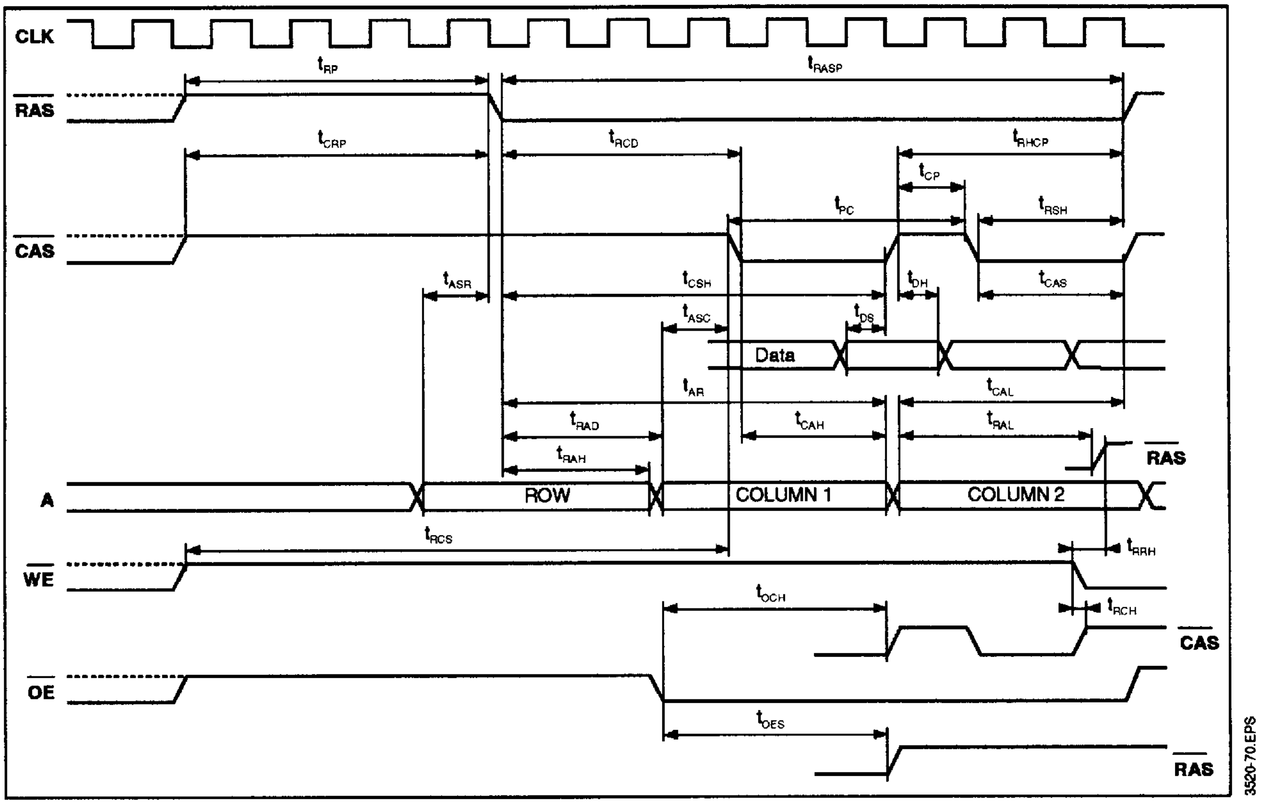
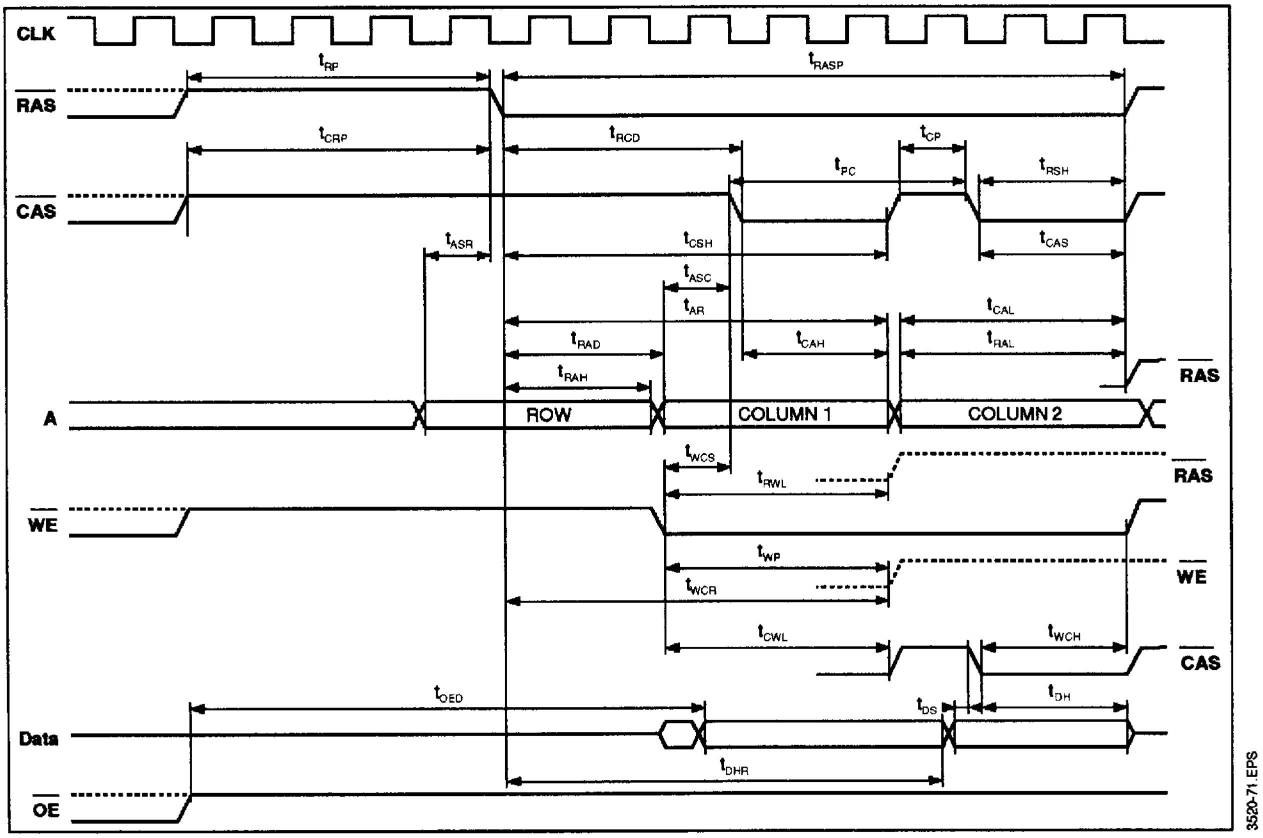


Figure 70 : Page Mode Early Write Cycle



Page Mode Read Cycle

Symbol	Parameter	Min.	Max.	Units
t_{RC}	Read Cycle Time	9T		ns
t_{RP}	$\overline{\text{RAS}}$ Precharge Time	4T-9		ns
t_{RASP}	$\overline{\text{RAS}}$ Pulse Width	5T-9		ns
t_{CRP}	$\overline{\text{CAS}}$ to $\overline{\text{RAS}}$ Precharge Time	4T-2		ns
t_{RCD}	$\overline{\text{RAS}}$ Low to $\overline{\text{CAS}}$ Low Delay Time (1)	3T-7.5	3T+6	ns
t_{PC}	Fast Page Mode Read Cycle Time	3T		ns
t_{CP}	$\overline{\text{CAS}}$ Precharge Time	T-2		ns
t_{RHCP}	$\overline{\text{RAS}}$ Hold Time after $\overline{\text{CAS}}$ Precharge (2)	3T-6		ns
t_{RSH}	$\overline{\text{RAS}}$ Hold Time after $\overline{\text{CAS}}$	2T-10		ns
t_{CSH}	$\overline{\text{CAS}}$ Hold Time after $\overline{\text{RAS}}$ (1)	5T-10		ns
t_{CAS}	$\overline{\text{CAS}}$ Pulse Width	2T-10		ns
t_{ASR}	Row Address Set-up Time to $\overline{\text{RAS}}$	T-TBD		ns
t_{RAH}	Row Address Hold Time after $\overline{\text{RAS}}$ (4)	2T-TBD		ns
t_{ASC}	Column Address Set-up Time to $\overline{\text{CAS}}$ (3)	T-TBD		ns
t_{RAD}	Column Address Delay Time from $\overline{\text{RAS}}$ (4)	2T+TBD	2T+TBD	ns
t_{CAH}	Column Address Hold Time from $\overline{\text{CAS}}$	2T-TBD		ns
t_{AR}	Column Address Hold Time from $\overline{\text{RAS}}$ (4)	5T-TBD		ns
t_{CAL}	Column Address to $\overline{\text{CAS}}$ Lead Time	3T-TBD		ns
t_{RAL}	Column Address to $\overline{\text{RAS}}$ Lead Time	3T-TBD		ns
t_{DS}	Data in Set-up Time to $\overline{\text{CAS}}$ Rising Edge	-4		ns
t_{DH}	Data in Hold Time from $\overline{\text{CAS}}$ Rising Edge	0		ns
t_{RCS}	Read Command to $\overline{\text{CAS}}$ Set-up Time	7T-10		ns
t_{RCH}	Read Command from $\overline{\text{CAS}}$ Hold Time	1		ns
t_{RRH}	Read Command from $\overline{\text{RAS}}$ Hold Time (5)	-5		ns
t_{OCH}	$\overline{\text{CAS}}$ from $\overline{\text{OE}}$ Hold Time	3T-13		ns
t_{OES}	$\overline{\text{RAS}}$ from $\overline{\text{OE}}$ Hold Time	3T-13		ns

3250-11.TBL

Notes :
1. Worst case for min. value is with one bank of memory, and each package having a single $\overline{\text{CAS}}$ Pin.
2. Sometimes referred to as t_{CPRH}.
3. Worst case is with two banks of memory, and each package having a single $\overline{\text{CAS}}$ Pin.
4. Worst case for min. value is with one bank of memory.
5. Worst case is with one bank of memory, and each package having a single $\overline{\text{WE}}$ Pin.

Page Mode Early Write Cycle

Symbol	Parameter	Min.	Max.	Units
t_{RP}	$\overline{\text{RAS}}$ Precharge Time	4T-9		ns
t_{RASP}	$\overline{\text{RAS}}$ Pulse Width	5T-9		ns
t_{CRP}	$\overline{\text{CAS}}$ to $\overline{\text{RAS}}$ Precharge Time	4T-2		ns
t_{RCD}	$\overline{\text{RAS}}$ Low to $\overline{\text{CAS}}$ Low Delay Time (1)	3T-7.5	3T+6	ns
t_{PC}	Fast Page Mode Read Cycle Time	3T		ns
t_{CP}	$\overline{\text{CAS}}$ Precharge Time	T-2		ns
t_{RSH}	$\overline{\text{RAS}}$ Hold Time after $\overline{\text{CAS}}$	2T-10		ns
t_{CSH}	$\overline{\text{CAS}}$ Hold Time after $\overline{\text{RAS}}$ (1)	5T-10		ns
t_{CAS}	$\overline{\text{CAS}}$ Pulse Width	2T-10		ns
t_{ASR}	Row Address Set-up Time to $\overline{\text{RAS}}$	T-TBD		ns
t_{RAH}	Row Address Hold Time after $\overline{\text{RAS}}$ (2)	2T-TBD		ns
t_{ASC}	Column Address Set-up Time to $\overline{\text{CAS}}$ (3)	T-TBD		ns
t_{RAD}	Column Address Delay Time from $\overline{\text{RAS}}$ (2)	2T+TBD	2T+TBD	ns
t_{CAH}	Column Address Hold Time from $\overline{\text{CAS}}$	2T-TBD		ns
t_{AR}	Column Address Hold Time from $\overline{\text{RAS}}$ (2)	5T-TBD		ns
t_{CAL}	Column Address to $\overline{\text{CAS}}$ Lead Time	3T-TBD		ns
t_{RAL}	Column Address to $\overline{\text{RAS}}$ Lead Time	3T-TBD		ns
t_{WCS}	Write Command Set-up Time to $\overline{\text{CAS}}$	T-11		ns
t_{RWL}	Write Command to $\overline{\text{RAS}}$ Lead Time	3T-15		ns
t_{CWL}	Write Command to $\overline{\text{CAS}}$ Lead Time	3T-15		ns
t_{WCR}	Write Command Hold Time after $\overline{\text{RAS}}$ (4)	5T-8		ns
t_{WCH}	Write Command Hold Time after $\overline{\text{CAS}}$	2T-6		ns
t_{WP}	Write Command Pulse Width	3T-13		ns
t_{DS}	Data in Set-up Time to $\overline{\text{CAS}}$ Rising Edge	T-18		ns
t_{DH}	Data in Hold Time from $\overline{\text{CAS}}$ Rising Edge	2T+9		ns
t_{OED}	$\overline{\text{OE}}$ High before Data Valid	6T-10		ns
t_{OEH}	$\overline{\text{OE}}$ High Hold Time after Write Command ($\overline{\text{WE}}$ falling edge) (5)	9T-14		ns
t_{DHR}	Data Hold Time after $\overline{\text{RAS}}$ (2)	5T+7		ns

3250-12.TBL

Notes : 1. Worst case for min. value is with one bank of memory, and each package having a single $\overline{\text{CAS}}$ Pin.

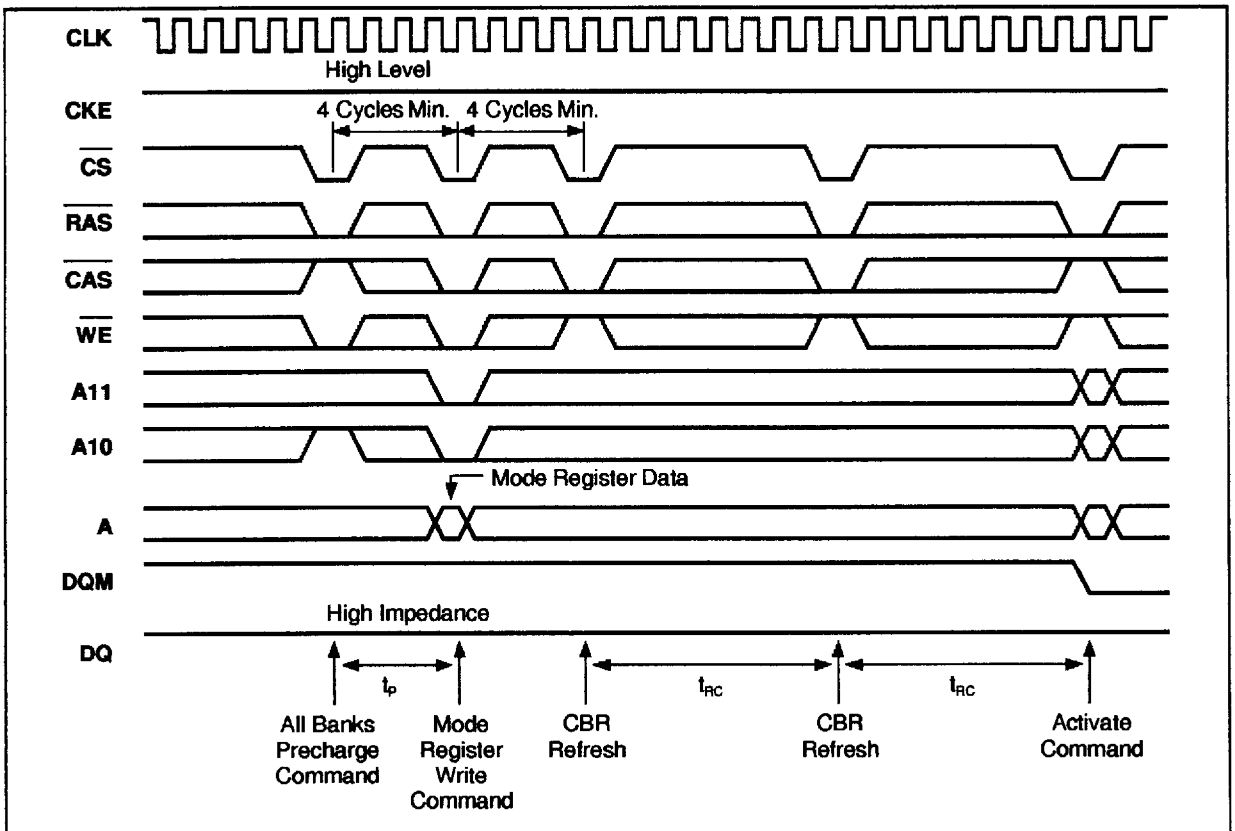
2. Worst case for min. value is with one bank of memory.

3. Worst case is with two banks of memory, and each package having a single $\overline{\text{CAS}}$ Pin.

4. Worst case is with one bank of memory, and each package having a single $\overline{\text{WE}}$ Pin.

5. Not shown on timing diagram.

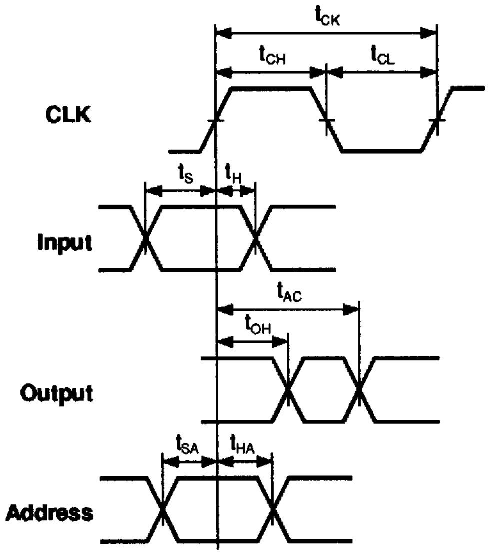
Figure 71 : Synchronous DRAM Power-on Sequence



Note : the number of refreshes required varies for different suppliers.

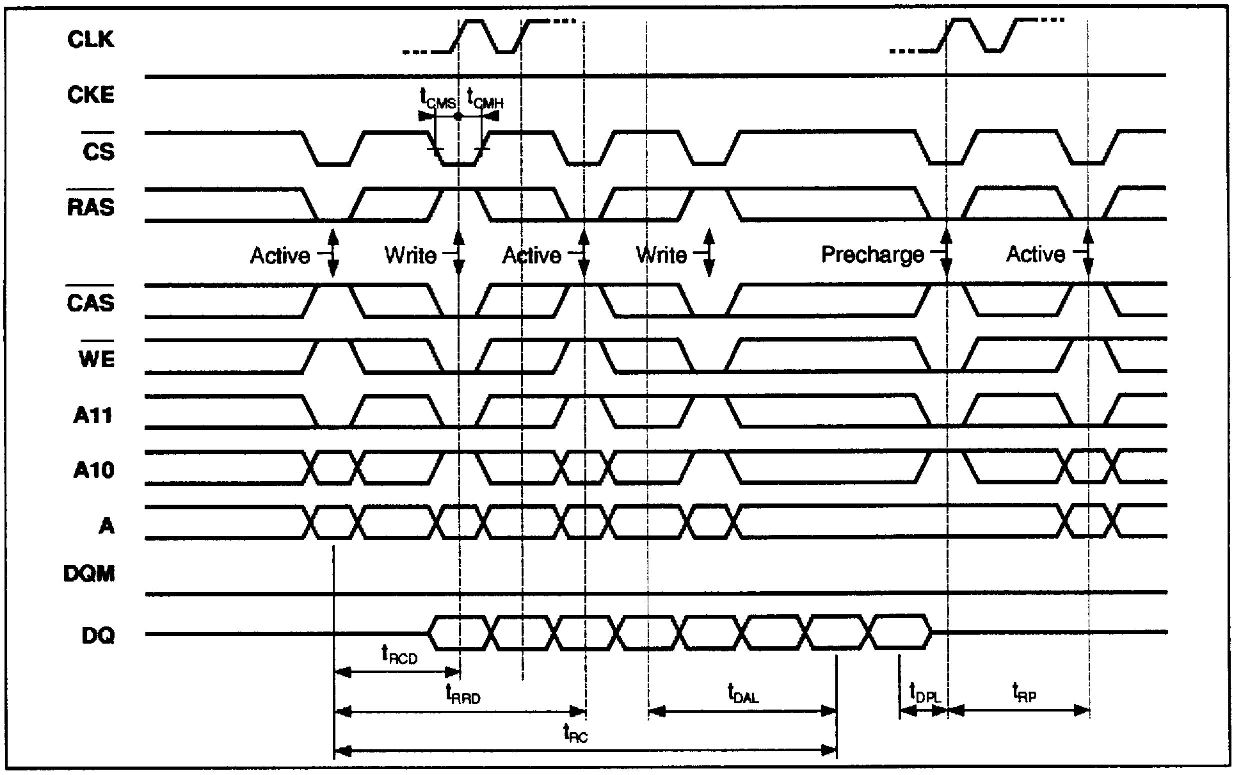
3520-72.EPS

Figure 72 : AC Parameters for Read and Write (synchronous DRAM)



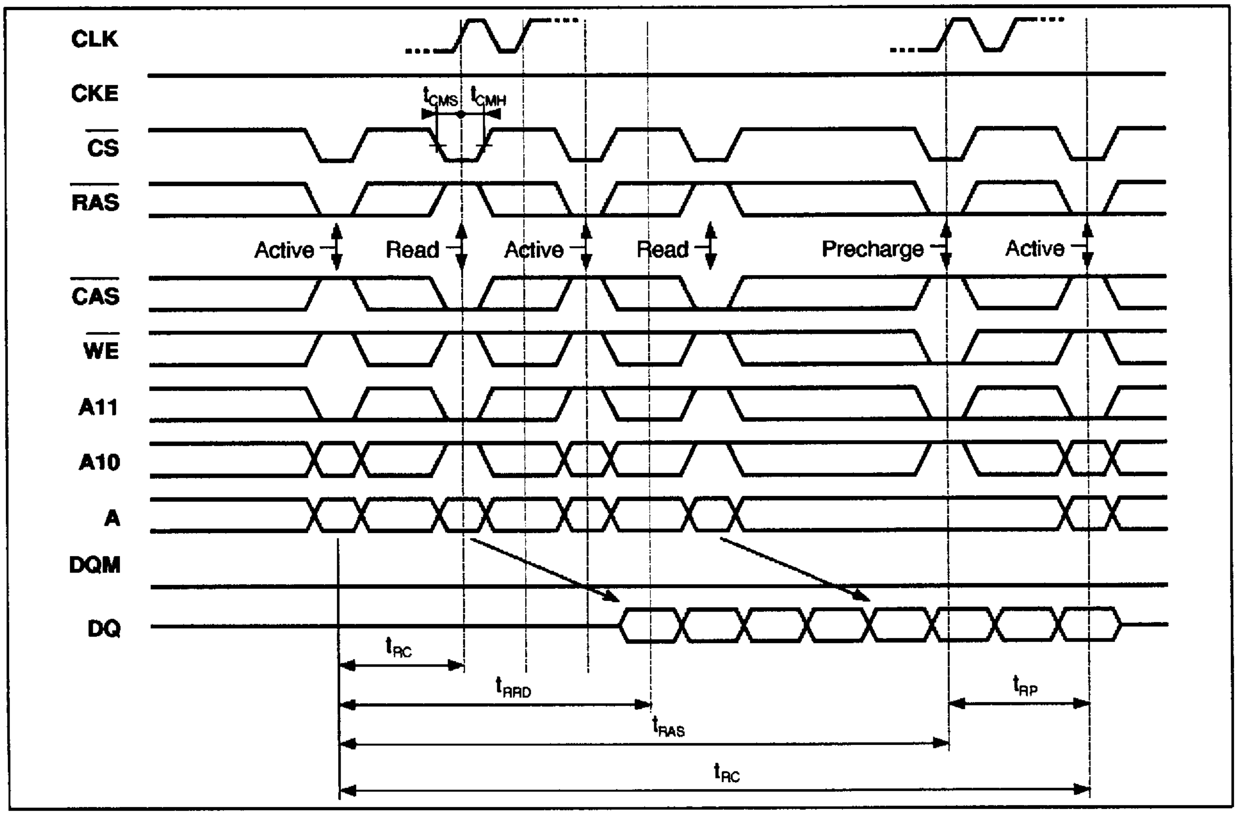
3520-73.EPS

Figure 73 : Synchronous DRAM Write (burst length = 4, CAS latency = 3)



3520-74.EPS

Figure 74 : Synchronous DRAM Read (burst length = 4, CAS latency = 3)



3520-75.EPS

Synchronous DRAM Read and Write

Symbol	Parameter	Min.	Max.	Units
t_{RC}	REF to REF / ACTIVE Command Period	84		ns
t_{RP}	ACTIVE to PRE Command Period	120		ns
t_{CK}	Clock Cycle Time	12		ns
t_{CH}	Clock High Level Width	4		ns
t_{CL}	Clock Low Level Width	4		ns
t_S	Data Input Set-up Time	3		ns
t_H	Data Input Hold Time	4		ns
t_{OH}	Output Data Hold Time	1.5		ns
t_{AC}	Output Data Access Time	8.5		ns
t_{SA}	Address Set-up	3.5		ns
t_{HA}	Address Hold Time	1.5		ns
t_{CMS}	Command ($\overline{CS}$, $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, DQM) Set-up Time	3.5		ns
t_{CMH}	Command ($\overline{CS}$, $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, DQM) Hold Time	1.5		ns
t_{RCD}	Delay Time ACTIVE to READ/WRITE Command	36		ns
t_{RRD}	ACTIVE to ACTIVE Command Period	36		ns
t_{DAL}	Data-out to ACTIVE Command Period (1)	2T'+36		ns
t_{DPL}	Data-out to PRECHARGE Command Period	T'+12		ns
t_{RAS}	ACTIVE to PRECHARGE Command Period	84		ns

3250-13.TBL

Note : 1. T' memory subsystem clock period.

Figure 75 : EDO Early Write Cycle

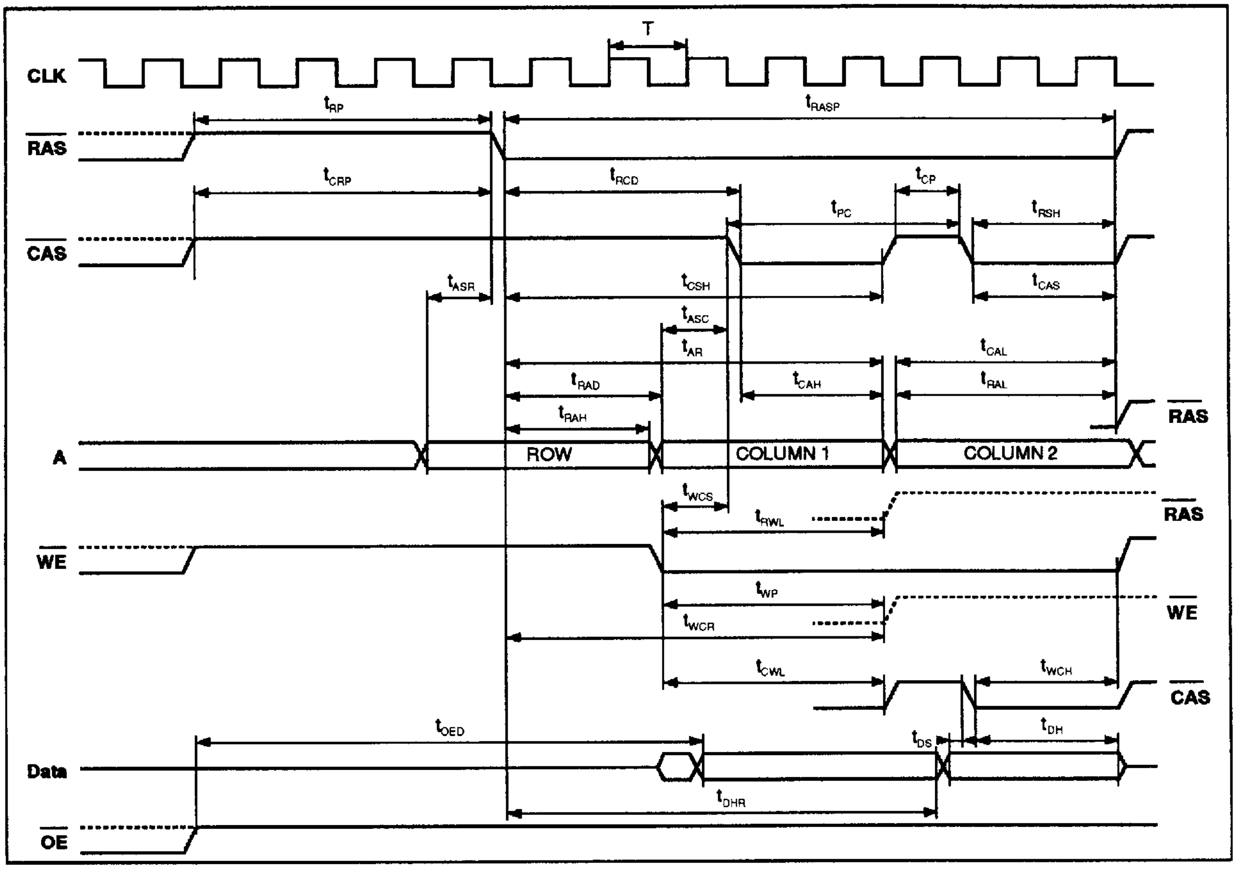
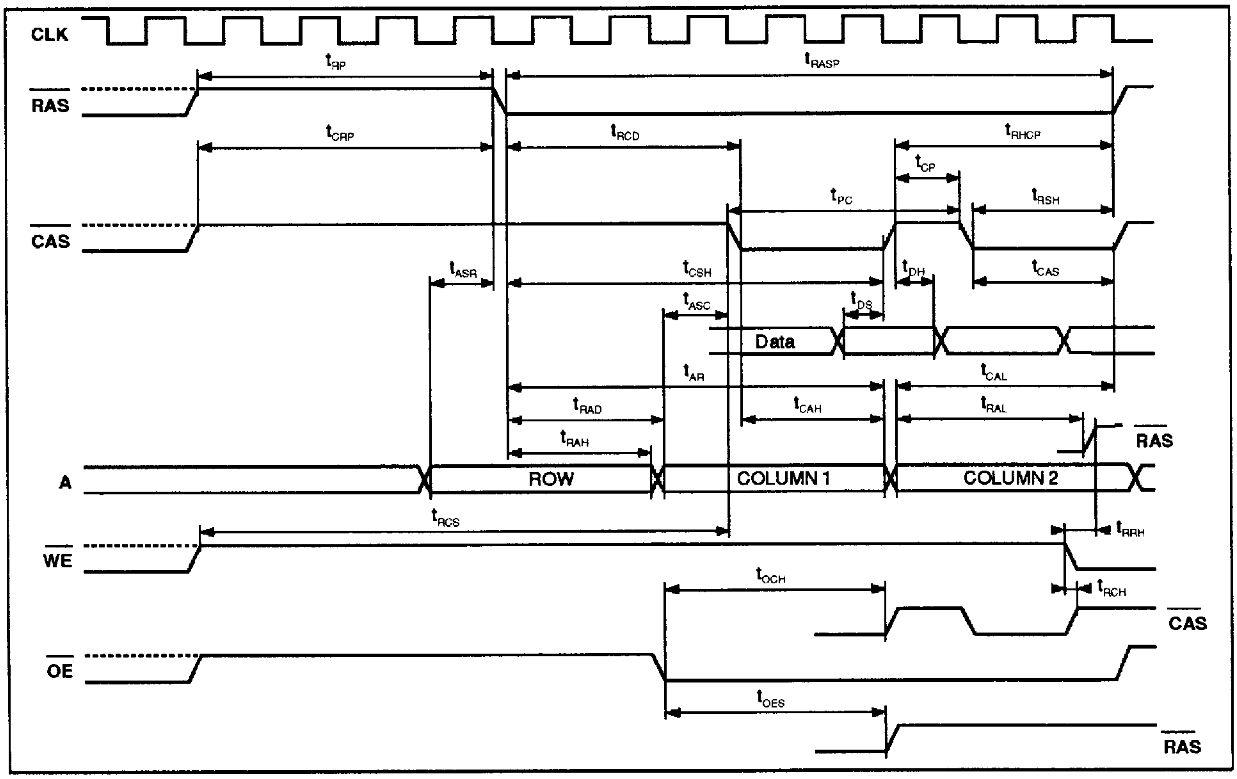


Figure 76 : EDO Read Cycle



EDO Early Write Cycle (see Note 1)

Symbol	Parameter	Min.	Max.	Units
t_{RP}	$\overline{\text{RAS}}$ Precharge Time	4T'-TBD		ns
t_{RASP}	$\overline{\text{RAS}}$ Pulse Width	8T'-TBD		ns
t_{CRP}	$\overline{\text{CAS}}$ to $\overline{\text{RAS}}$ Precharge Time	5T'-TBD		ns
t_{RCD}	$\overline{\text{RAS}}$ Low to $\overline{\text{CAS}}$ Low Delay Time	3T'-TBD	4T'+TBD	ns
t_{PC}	Fast Page Mode Read Cycle Time	2T'		ns
t_{CP}	$\overline{\text{CAS}}$ Precharge Time	T'-TBD		ns
t_{RSH}	RAS Hold Time after $\overline{\text{CAS}}$	2T'-TBD		ns
t_{CSH}	$\overline{\text{CAS}}$ Hold Time after $\overline{\text{RAS}}$	4T'-TBD		ns
t_{CAS}	$\overline{\text{CAS}}$ Pulse Width	T'-TBD		ns
t_{ASR}	Row Address Set-up Time to $\overline{\text{RAS}}$	T'-TBD		ns
t_{RAH}	Row Address Hold Time after $\overline{\text{RAS}}$	3T'-TBD		ns
t_{ASC}	Column Address Set-up Time to $\overline{\text{CAS}}$	T'-TBD		ns
t_{RAD}	Column Address Delay Time from $\overline{\text{RAS}}$	2T'+TBD	3T'+TBD	ns
t_{CAH}	Column Address Hold Time from $\overline{\text{CAS}}$	T'-TBD		ns
t_{AR}	Column Address Hold Time from $\overline{\text{RAS}}$	4T'-TBD		ns
t_{CAL}	Column Address to $\overline{\text{CAS}}$ Lead Time	2T'-TBD		ns
t_{RAL}	Column Address to $\overline{\text{RAS}}$ Lead Time	5T'-TBD		ns
t_{WCS}	Write Command Set-up Time to $\overline{\text{CAS}}$	4T'-TBD		ns
t_{RWL}	Write Command to $\overline{\text{RAS}}$ Lead Time	9T'-TBD		ns
t_{CWL}	Write Command to $\overline{\text{CAS}}$ Lead Time	5T'-TBD		ns
t_{WCR}	Write Command Hold Time after $\overline{\text{RAS}}$	6T'-TBD		ns
t_{WCH}	Write Command Hold Time after $\overline{\text{CAS}}$	T'-TBD		ns
t_{WP}	Write Command Pulse Width	7T'-TBD		ns
t_{DS}	Data in Set-up Time to $\overline{\text{CAS}}$ Rising Edge	T'-TBD		ns
t_{DH}	Data in Hold Time from $\overline{\text{CAS}}$ Rising Edge	T'+TBD		ns
t_{OED}	$\overline{\text{OE}}$ High before Data Valid	8T'-TBD		ns
t_{OEH}	$\overline{\text{OE}}$ High Hold Time after Write Command ($\overline{\text{WE}}$ falling edge)	12T'-TBD		ns
t_{DHR}	Data Hold Time after $\overline{\text{RAS}}$	5T'+TBD		ns

3250-14.TBL

Note : 1. The timing are given in terms of the clock period of the memory subsystem clock (T').

Page Mode Read Cycle

Symbol	Parameter	Min.	Max.	Units
t_{RC}	Read Cycle Time (1)	12T'		ns
t_{RP}	$\overline{\text{RAS}}$ Precharge Time	4T'-TBD		ns
t_{RASP}	$\overline{\text{RAS}}$ Pulse Width	8T'-TBD		ns
t_{CRP}	$\overline{\text{CAS}}$ to $\overline{\text{RAS}}$ Precharge Time	5T'-TBD		ns
t_{RCD}	$\overline{\text{RAS}}$ Low to $\overline{\text{CAS}}$ Low Delay Time	3T'-TBD	4T'+TBD	ns
t_{PC}	Fast Page Mode Read Cycle Time	2T'		ns
t_{CP}	$\overline{\text{CAS}}$ Precharge Time	T'-TBD		ns
t_{RHCP}	$\overline{\text{RAS}}$ Hold Time after $\overline{\text{CAS}}$ Precharge	T'-TBD		ns
t_{RSH}	$\overline{\text{RAS}}$ Hold Time after $\overline{\text{CAS}}$	2T'-TBD		ns
t_{CSH}	$\overline{\text{CAS}}$ Hold Time after $\overline{\text{RAS}}$	4T'-TBD		ns
t_{CAS}	$\overline{\text{CAS}}$ Pulse Width	T'-TBD		ns
t_{ASR}	Row Address Set-up Time to $\overline{\text{RAS}}$	T'-TBD		ns
t_{RAH}	Row Address Hold Time after $\overline{\text{RAS}}$	3T'-TBD		ns
t_{ASC}	Column Address Set-up Time to $\overline{\text{CAS}}$	T'-TBD		ns
t_{RAD}	Column Address Delay Time from $\overline{\text{RAS}}$	2T'+TBD	3T'+TBD	ns
t_{CAH}	Column Address Hold Time from $\overline{\text{CAS}}$	T'-TBD		ns
t_{AR}	Column Address Hold Time from $\overline{\text{RAS}}$	4T'-TBD		ns
t_{CAL}	Column Address to $\overline{\text{CAS}}$ Lead Time	2T'-TBD		ns
t_{RAL}	Column Address to $\overline{\text{RAS}}$ Lead Time	5T'-TBD		ns
t_{DS}	Data in Set-up Time to $\overline{\text{CAS}}$ Rising Edge	TBD		ns
t_{DH}	Data in Hold Time from $\overline{\text{CAS}}$ Rising Edge	0		ns
t_{RCS}	Read Command to $\overline{\text{CAS}}$ Set-up Time	4T'-TBD		ns
t_{RCH}	Read Command from $\overline{\text{CAS}}$ Hold Time	0		ns
t_{RRH}	Read Command from $\overline{\text{RAS}}$ Hold Time	-T'		ns
t_{OCH}	$\overline{\text{CAS}}$ from $\overline{\text{OE}}$ Hold Time	4T'-TBD		ns
t_{OES}	$\overline{\text{RAS}}$ from $\overline{\text{OE}}$ Hold Time	4T'-TBD		ns

3250-15.TBL

Note : 1. The timing are given in terms of the clock period of the memory subsystem clock (T').

XIII.3 - Microcontroller Interface (video)

Timing measurements are made with respect to thresholds of 1.5V.

Figure 77 : Single Byte Register Read Cycle (video)

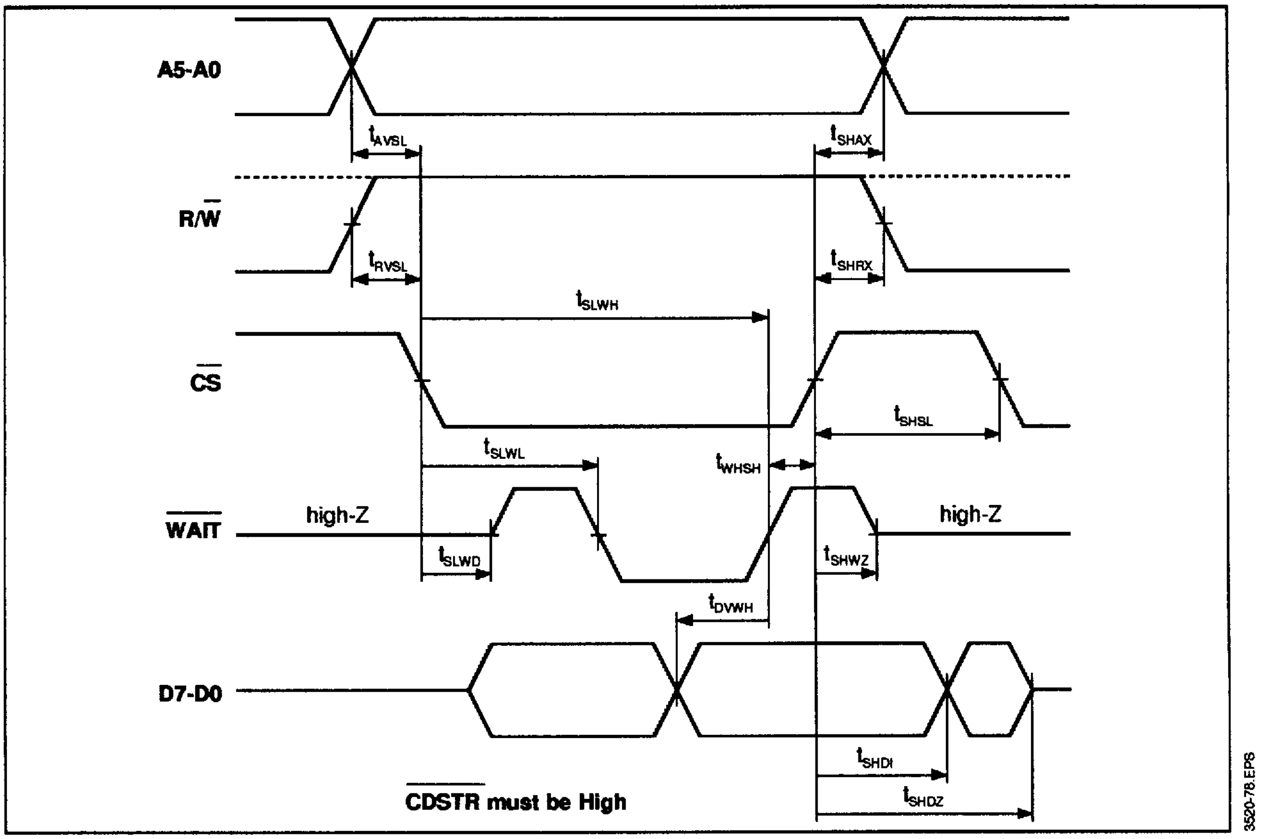


Figure 78 : Byte Register Read Cycle (video)

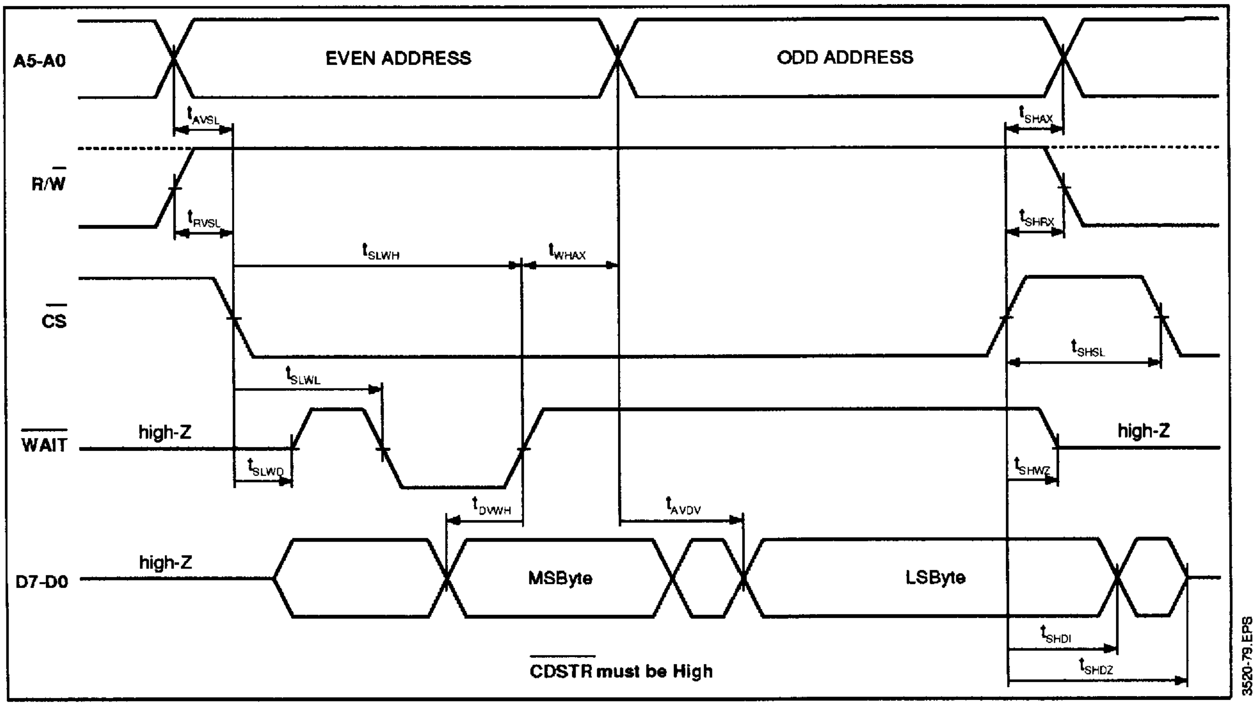
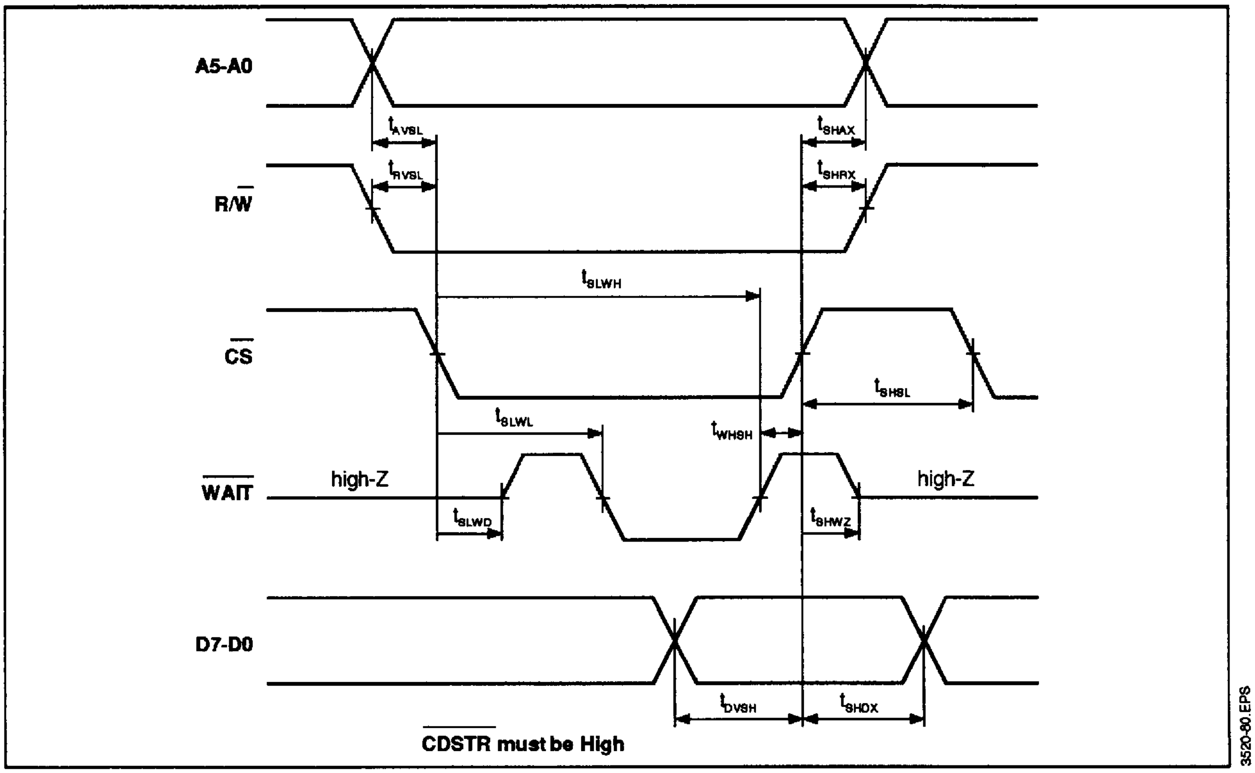


Figure 79 : Register Write Cycle (video)



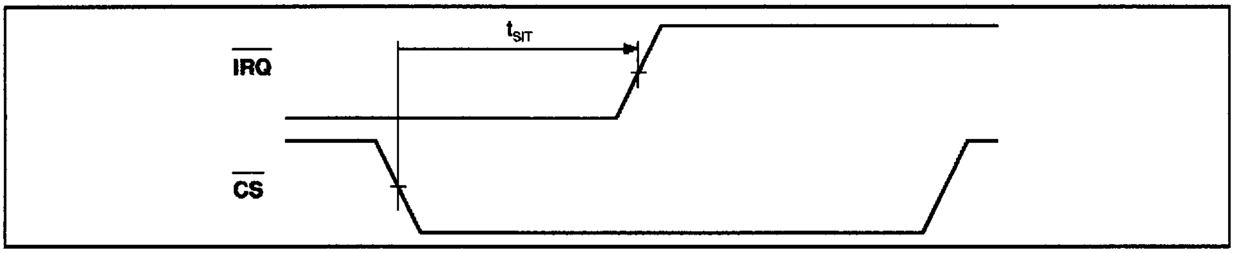
Symbol	Parameter	Min.	Max.	Units
t_{AVSL}	Address to $\overline{CS}$ Set-up Time	5		ns
t_{RVSL}	R/W to $\overline{CS}$ Set-up Time	5		ns
t_{SHAX}	Address from $\overline{CS}$ Hold Time	5		ns
t_{SHRX}	R/W from $\overline{CS}$ Hold Time	5		ns
t_{SLWD}	$\overline{CS}$ Low to $\overline{WAIT}$ On		10	ns
t_{SLWL}	$\overline{CS}$ Low to $\overline{WAIT}$ Low		25	ns
t_{SLWH}	$\overline{CS}$ Low to $\overline{WAIT}$ High		50	ns
t_{SHWZ}	$\overline{CS}$ High to $\overline{WAIT}$ Off (hi-Z)		10	ns
t_{SHSL}	$\overline{CS}$ High to $\overline{CS}$ or $\overline{CDSTR}$ Low Again	20		ns
t_{WHSH}	$\overline{CS}$ High from $\overline{WAIT}$ High Hold Time	0		ns
t_{DVWH}	Data Valid before $\overline{WAIT}$ High	5		ns
t_{SHDI}	$\overline{CS}$ High to Data Invalid	10		ns
t_{SHDZ}	$\overline{CS}$ High to Data Off (hi-Z)		20	ns
t_{WHAX}	Address from $\overline{WAIT}$ High Hold Time (first byte)	0		ns
t_{AVDV}	Address Valid to Data Valid (second byte)		20	ns
t_{DVSH}	Data Valid to $\overline{CS}$ High Set-up Time	15		ns
t_{SHDX}	Data from $\overline{CS}$ High Hold Time	5		ns

3250-16.TBL

XIII.4 - Interrupt Acknowledge (video)

Timing measurements are made with respect to thresholds of 1.5V.

Figure 80 : Interrupt Acknowledge (video)



3520-81.EPS

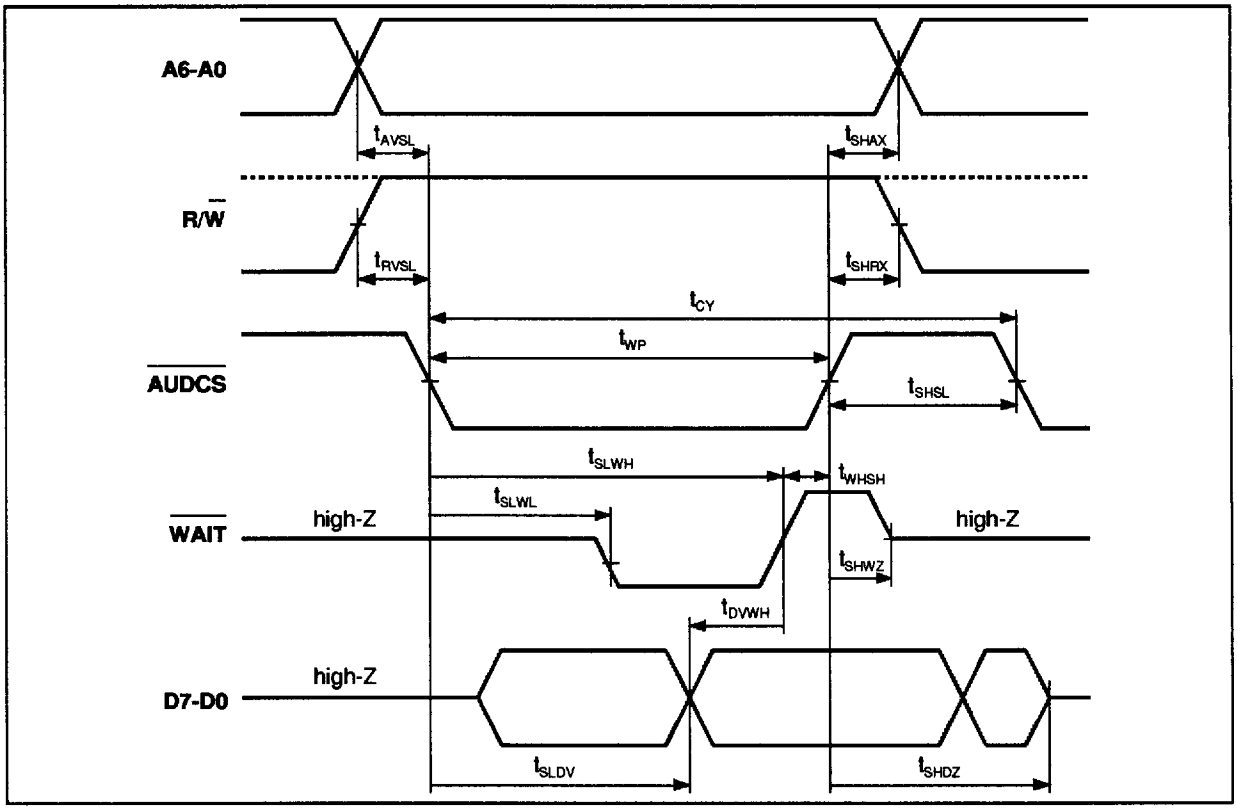
Symbol	Parameter	Min.	Max.	Units
t_{SIT}	$\overline{CS}$ Low to $\overline{IRQ}$ High		50	ns

3250-17.TBL

XIII.5 - Microcontroller Interface (audio)

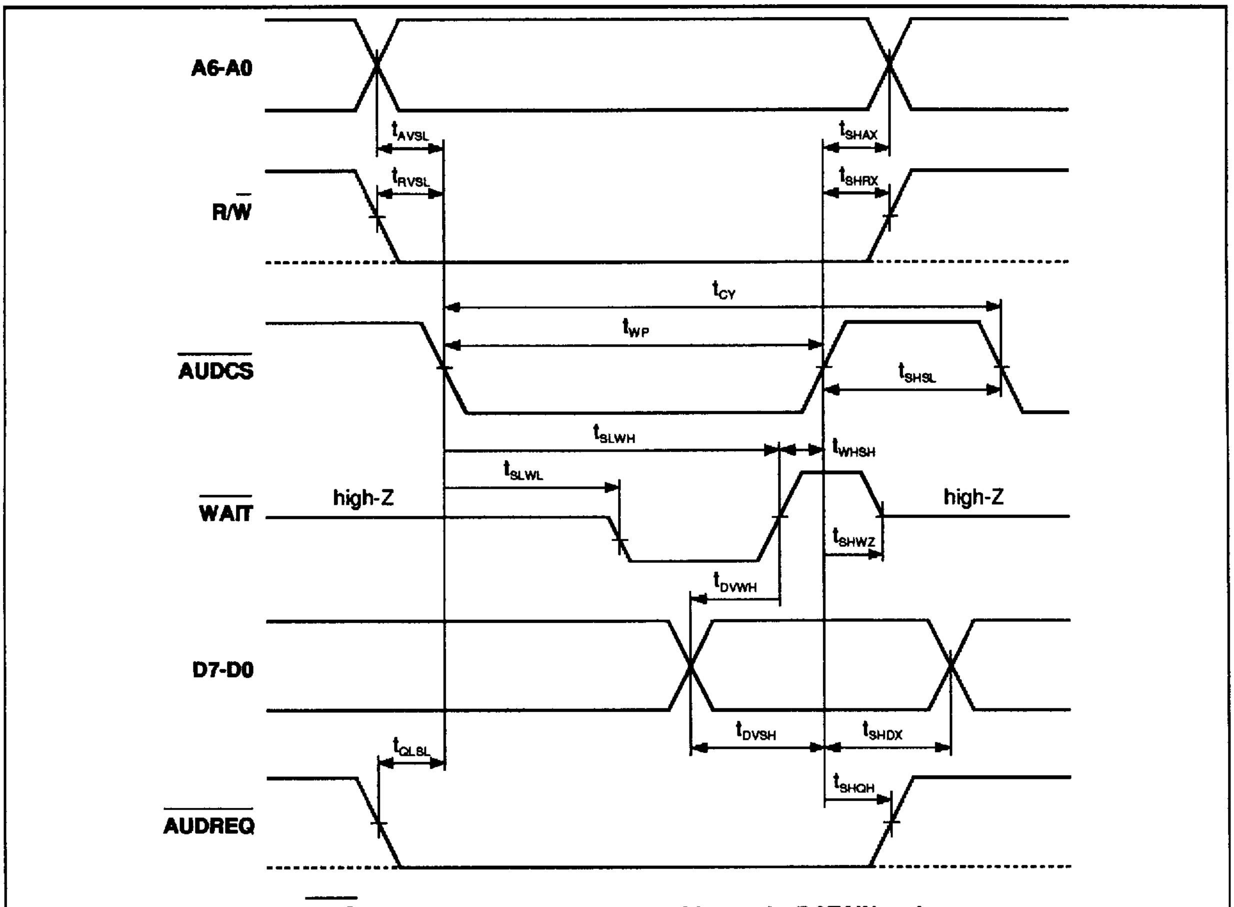
Timing measurements are made with respect to thresholds of 1.5V.

Figure 81 : Read Cyde (audio)



3520-82.EPS

Figure 82 : Write Cycle (audio)



The $\overline{\text{REQ}}$ signal is only required when writing to the DATAIN register

3520-83.EPS

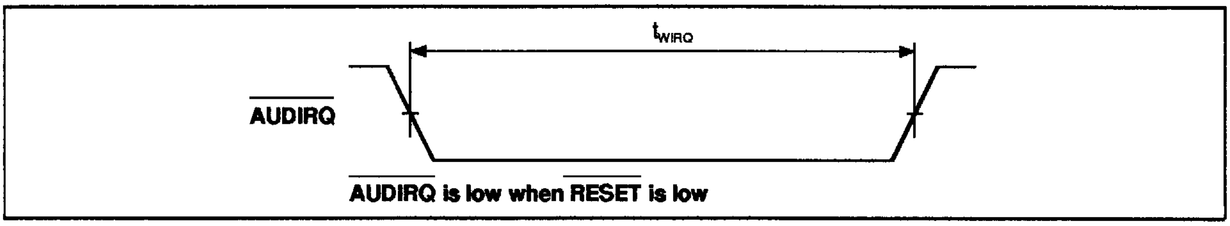
Symbol	Parameter	Min.	Max.	Units
t_{AVSL}	Address to $\overline{\text{AUDCS}}$ Low Set-up Time	15		ns
t_{RVSL}	R/$\overline{\text{W}}$ to $\overline{\text{AUDCS}}$ Low Set-up Time	15		ns
t_{SHAX}	Address from $\overline{\text{AUDCS}}$ High Hold Time	25		ns
t_{SHRX}	R/$\overline{\text{W}}$ from $\overline{\text{AUDCS}}$ High Hold Time	25		ns
t_{CY}	Read or Write Cycle Time	100		ns
t_{SLWL}	$\overline{\text{AUDCS}}$ Low to $\overline{\text{WAIT}}$ Low		20	ns
t_{SLWH}	$\overline{\text{AUDCS}}$ Low to $\overline{\text{WAIT}}$ High (1) (2)		45 80	ns
t_{WHSH}	$\overline{\text{WAIT}}$ High from $\overline{\text{AUDCS}}$ High Set-up Time	0		ns
t_{SHWZ}	$\overline{\text{AUDCS}}$ High to $\overline{\text{WAIT}}$ Disabled		20	ns
t_{DVWH}	Data Valid before $\overline{\text{WAIT}}$ High	5		ns
t_{SLDV}	$\overline{\text{AUDCS}}$ Low to Data Invalid (1) (2)	5 5	40 75	ns
t_{SHDZ}	$\overline{\text{AUDCS}}$ High to Data Off (hi-Z)		10	ns
t_{DVSH}	Data Valid to $\overline{\text{AUDCS}}$ High Set-up Time	25		ns
t_{SHDX}	Data from $\overline{\text{AUDCS}}$ High Hold Time			ns
t_{QLSL}	$\overline{\text{AUDREQ}}$ Low to $\overline{\text{AUDDCS}}$ Low Set-up Time	25		ns
t_{SHQH}	$\overline{\text{AUDDCS}}$ High to $\overline{\text{AUDREQ}}$ High (3)		80	ns

3250-18.TBL

Notes : 1. When accessing any register except DATAIN.
2. When accessing DATIN register. $\overline{\text{WAIT}}$ will remain low if PCM output is stopped and the buffers are full.
3. One more byte can be strobed in after $\overline{\text{REQ}}$ going high signals that the input buffer is full.

XIII.6 - Interrupt Request (audio)

Figure 83 : Interrupt Request (audio)


t_{WIRQ}
$\overline{\text{AUDIRQ}}$
$\overline{\text{AUDIRQ}}$ is low when $\overline{\text{RESET}}$ is low

3520-84.EPS

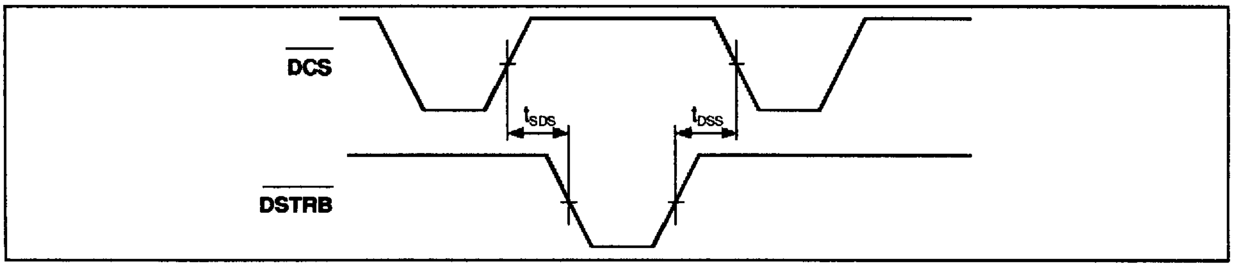
Symbol	Parameter	Min.	Max.	Units
t_{WIRQ}	$\overline{\text{AUDIRQ}}$ Pulse Width (1)		100	ns

3250-19.TBL

Note : 1. $\overline{\text{IRQ}}$ pulse is generated when a bit becomes set in register INTR.

XIII.7 - Relative Timing of $\overline{\text{AUDCS}}$ and $\overline{\text{AUDSTR}}$

Figure 84 : $\overline{\text{AUDCS}}$, $\overline{\text{AUDSTR}}$



3520-85.EPS

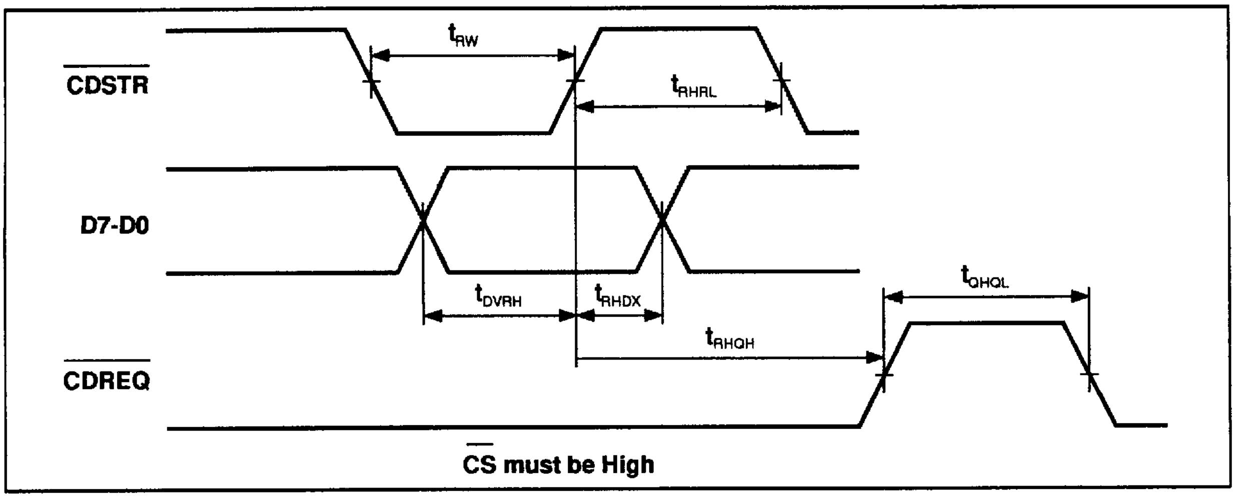
Symbol	Parameter	Min.	Max.	Units
t_{SDS}	$\overline{\text{AUDCS}}$ High before $\overline{\text{AUDSTR}}$ Low	50		ns
t_{DSS}	$\overline{\text{AUDSTR}}$ High before $\overline{\text{AUDCS}}$ Low	50		ns

3250-20.TBL

XIII.8 - Compressed Data Write Cycle (video)

Timing measurements are made with respect to thresholds of 1.5V.

Figure 85 : Compressed Data Write (video)


$\overline{\text{CDSTR}}$
D7-D0
$\overline{\text{CDREQ}}$
t_{RW}
t_{RHRL}
t_{DVRH}
t_{RHDX}
t_{RHQH}
t_{QHQL}
$\overline{\text{CS}}$ must be High
3520-86.EPS

Symbol	Parameter	Min.	Max.	Units
t_{RW}	$\overline{\text{CDSTR}}$ Pulse Width	15		ns
t_{RHRL}	$\overline{\text{CDSTR}}$ High to $\overline{\text{CDSTR}}$ or $\overline{\text{CS}}$ Low Again (1)	20		ns
t_{DVRH}	D7-D0 Set-up Time to $\overline{\text{CDSTR}}$ Rising Edge	15		ns
t_{RHDX}	D7-D0 Hold Time from $\overline{\text{CDSTR}}$ Rising Edge	5		ns
t_{RHQH}	$\overline{\text{CDREQ}}$ High from $\overline{\text{CDSTR}}$ High (2, 3)		40	ns
t_{QHQL}	$\overline{\text{CDREQ}}$ High to Low Again : Normal Mode 8-Mbit Mode		 180 300	Primary Clock Cycles

3250-21.TBL

Notes : 1. If $\overline{\text{CDREQ}}$ handshake not used.

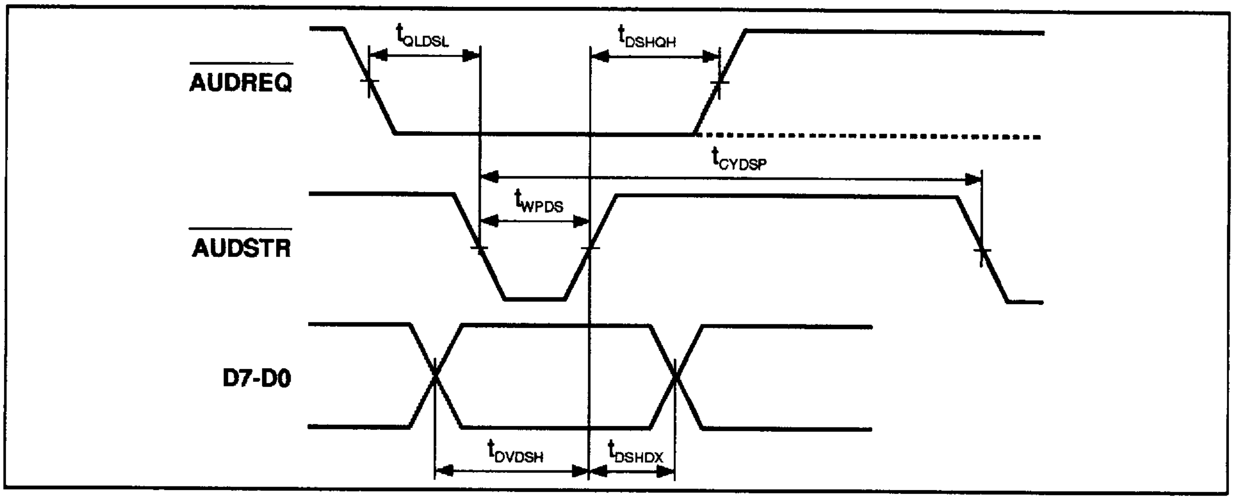
2. $\overline{\text{CDREQ}}$ goes high if the write caused CD FIFO to become almost full.

3. $\overline{\text{CDREQ}}$ becoming high indicates that only 3 more bytes can be written. For the CD FIFO to become empty there is a maximum delay of 350 primary clock cycles (or 580 in 8-Mbit mode) after the rising edge of CDSTR.

XIII.9 - Compressed Data Input (audio)

Timing measurements are made with respect to thresholds of 1.5V.

Figure 86 : Compressed Data Write (audio)



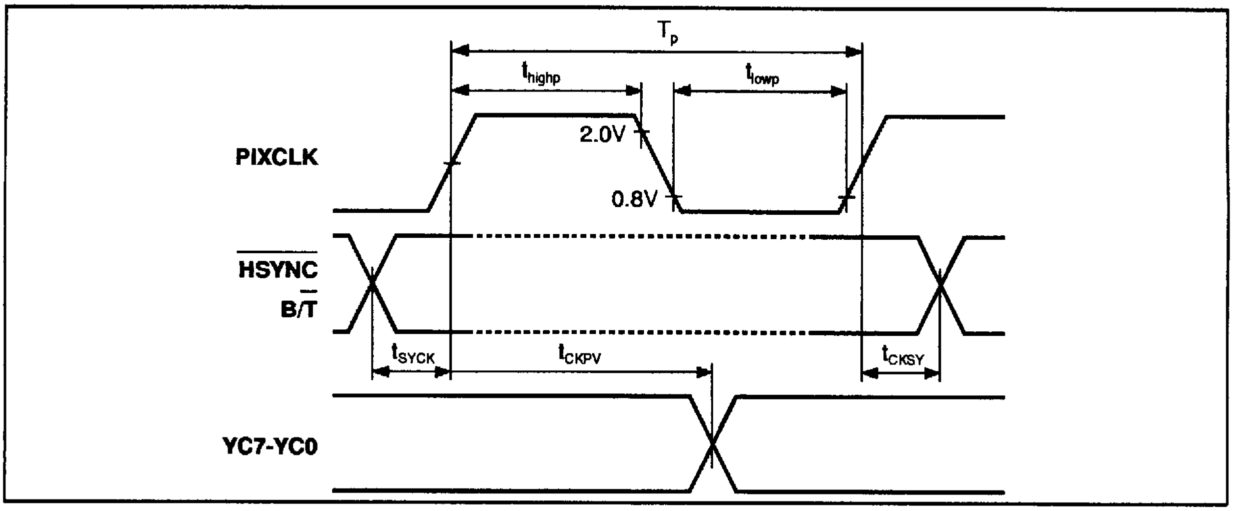
Symbol	Parameter	Min.	Max.	Units
t_{QLDSL}	$\overline{\text{AUREQ}}$ Low to $\overline{\text{AUDSTR}}$ Low Set-up Time	25		ns
t_{DSHQH}	$\overline{\text{AUDSTR}}$ High to $\overline{\text{AUDREQ}}$ High (1)		80	ns
t_{WPDS}	$\overline{\text{AUDSTR}}$ Low Time	25		ns
t_{CYDSP}	$\overline{\text{AUDSTR}}$ Cycle Time (2)	400		ns
t_{DVDSH}	D Set-up Time to $\overline{\text{AUDSTR}}$ Rising Edge	25		ns
t_{DSHDX}	D Hold Time from $\overline{\text{AUDSTR}}$ Rising Edge	0		ns

Notes : 1. One more byte can be strobed in after $\overline{\text{REQ}}$ going high signals that the input buffer is full.
2. This corresponds to a maximum input rate of 2.5Mbyte/s.

XIII.10 - Video Interface Timing

Timing measurements are made with respect to thresholds of 1.5V.

Figure 87 : Video Interface



3520-88.EPS

Symbol	Parameter	Min.	Max.	Units
T_p	Pel Clock Period (1)	35		ns
t_{highp}	Pel Clock High Time	14		ns
t_{lowp}	Pel Clock Low Time	14		ns
t_{rp}	Pel Clock Rise Time			ns
t_{fp}	Pel Clock Fall Time			ns
t_{SYCK}	$\overline{\text{HSYNC}}$, B/$\overline{\text{T}}$ Set-up Time to PIXCLK Rising Edge	10		ns
t_{CKSY}	$\overline{\text{HSYNC}}$, B/$\overline{\text{T}}$ from PIXCLK Rising Edge Hold Time	0		ns
t_{CKPV}	PIXCLK to YC7-YC0 Valid	4	25	ns

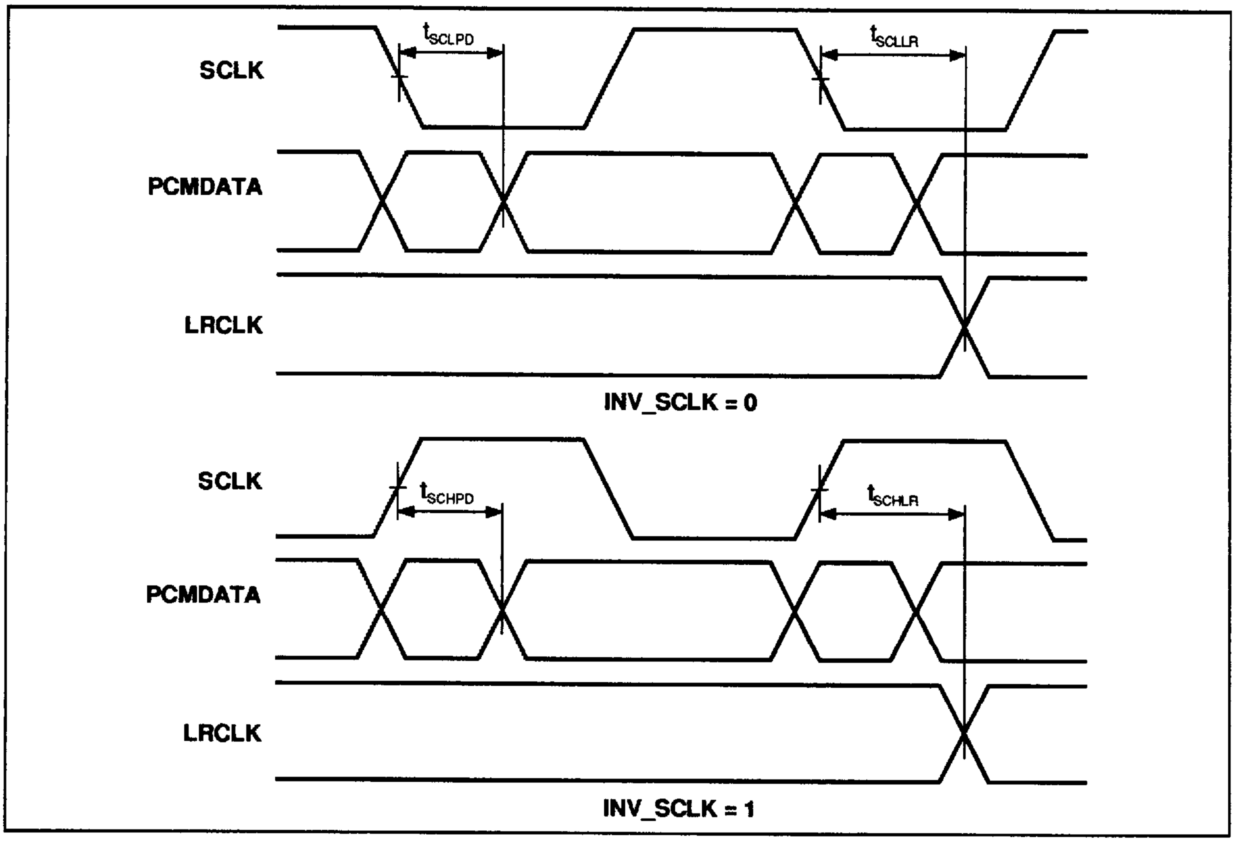
3250-23.TBL

Note : 1. This corresponds to a PIXCLK frequency of 28.6MHz.

XIII.11 - PCM Data Output

Timing measurements are made with respect to thresholds of 1.5V.

Figure 88 : PCM Data Output



3520-89.EPS

Symbol	Parameter	Min.	Max.	Units
T_{SCLPD}	SC Low to PCMDATA Valid		50	ns
t_{SCLLR}	SC Low to LRCLK		50	ns

3250-24.TBL

XIV - VIDEO REGISTERS

XIV.1 - Register Map

Address (hex)	MSByte	LSByte	Address (hex)
00	HDF (R)		01
02		CMD (R/W)	03
04	GCF (R/W)		05
06	CTL (R/W) PLL (R/W)		07
08	STA (R)		09
0A	ITM (R/W)		0B
0C	ITS (R)		0D
0E	INS1 (R/W) INS2 (R/W)		0F
10	MRF (R) BMS (R/W)	MWF (W)	11
12	▨	▨	13
14	MRP (R/W)		15
16	▨	▨	17
18	MWP (R/W)		19
1A	▨ DFP (R/W)		1B
1C	▨ RFP (R/W)		1D
1E	▨ FFP (R/W)		1F
20	▨ BFP/FBP (R/W)		21
22	▨ BBL (R)		23
24	▨ BBS (R/W) BBG (R/W)		25
26	▨ BBT (R/W)		27
28	▨	DFW (R/W) XFW (R/W)	29
2A	▨ DFS (R/W) XFS (R/W)		2B
2C	YDO (R/W)	XDO (R/W)	2D
2E	YDS (R/W)	XDS (R/W)	2F
30	▨ OBP (R/W)		31
32	▨ OTP (R/W)		33
34	LSO (R/W)	LSR (R/W)	35
36	CSO (R/W)	CSR (R/W)	37
38	DCF (R/W)		39
3A	PSV (R/W)		3B
3C	QMW (W)	QMW (W)	3D
3E	▨	▨	3F

XIV.2 - Register Descriptions

Registers are listed in alphabetical order.

All addresses are in hexadecimal.

All unspecified bits of the register map are reserved. Only the value 0 must be written to any of these bits. The values which are read from these bits are undefined.

The reset state is the state existing after a hard reset.

Synchronization

There are two types of register : synchronized and unsynchronized.

Synchronized registers only change value in response to an internal event, either DSYNC or VSYNC, depending on the register. These registers are double-banked ; during the write cycle the new value is loaded into a master register, and on the occurrence of the synchronizing event this value is loaded into a slave register, at which time the new value is available to the circuit. When a synchronized register is read, the value returned is that held in the master register.

Unsynchronized registers change their value immediately they are written to. With the exception of the BBT, BMS, CTL and ITM registers and bit CMD.HDS, the unsynchronized registers are simple latches which are open when the signals $\overline{\text{CS}}$ and R/$\overline{\text{W}}$ are both low.

BBG - Start of Bit Buffer

13	0
BBG[13:0]	

Address :	24-25
Type :	R/W
Reset State :	0
Synchronization :	None

Description

The register holds the starting address of the bit buffer, defined in units of 2 kbits (32 words). If the bit buffer starts at address 0, then this register does not need to be set up, since its reset state is 0. A soft reset must be done immediately after the loading of this register in order for the value to be taken into account. In other words it must only be changed before the first compressed data of a new sequence is input, and never during the decoding of a sequence.

This register shares the same addresses as the BBS register. Bit CMD.BBG selects which of the two registers is accessible at these addresses. When this bit is set, the BBG register is selected.

BBL - Bit Buffer Level

13	0
BBL[13:0]	

Address :	22-23
Type :	R
Reset State :	0

Description

This register holds the current level of occupation of the bit buffer, defined in units of 2 kbits (32 words). It can be read at any time for the monitoring of the bit buffer level. When BBL is greater than or equal to the value held in the BBT register, the status bit STA.BBF becomes set. When BBL is zero, the status bit STA.BBE becomes set. These conditions can be used to cause interrupts.

BBS - Bit Buffer Stop Address

13	0
BBS[13:0]	

Address :	24-25
Type :	R/W
Reset State :	0
Synchronization :	None

Description

This register holds the address of the top of the bit buffer, defined in units of 2 kbits (32 words). The space allocated to the bit buffer starts at the address defined by the BBG register, or, by default, 0. The end address of the bit buffer is :

$$(32 \times BBS) + 31$$

BBS must only be changed before the first compressed data of a new sequence is input, and never during the decoding of a sequence.

This register shares the same addresses as the BBG register. Bit CMD.BBG selects which of the two registers is accessible at these addresses. When this bit is reset, the BBS register is selected.

BBT - Bit Buffer Threshold

13	0
BBT[13:0]	

Address :	26-27
Type :	R/W
Reset State :	0
Synchronization :	None

Description

This register holds the level of occupancy of the bit buffer, in units of 2 kbits (32 words), which when reached causes the status bit STA.BBF to become set, i.e. if BBL ≥ BBT, STA.BBF is set.

This threshold would normally be used to generate a "bit buffer nearly full" interrupt.

If the bit CTL.PBO is set, then transfer of data from the CD FIFO to the bit buffer is prevented if the bit buffer level is at or above the level defined in the BBT register. If BBT is set to a value equal to the size of the bit buffer (i.e. BBT = BBS + 1 - BBG), then this automatic mechanism will ensure than overflow never occurs.

BFP/FBP - Backward Frame / Fold-back Pointer

13	0
BFP/FBP[13:0]	

Address : 20-21
Type : R/W
Reset State : 0
Synchronization : DSYNC

Description

This register holds the start address of the backward prediction frame picture buffer, defined in units of 32 words.

When the STi3520 is operating in 8-Mbit mode (bit CTL.S8M is set), this register has a dual function. If address folding is required, the register must be set up with the value of the starting address in memory, in units of 64 32-bit words, of the frame buffer area. This is the fold-back address. If address folding is not required, i.e. accesses will never go beyond the end of memory, then this register is set up with the backward prediction frame buffer address. Thus, address folding and B-picture decoding are mutually exclusive.

BMS - Block Move Size

15	0
BMS[15:0]	

Address : 10-11
Type : R/W
Reset State : 0
Synchronization : None

Description

This register holds the number of words to be moved in a block move operation. Writing to it starts a block move operation.

This register shares the same addresses as the MRF and MWF registers. Bit CMD.SBM selects whether MRF/MWF or BMS is accessible at these addresses. When this bit is set, the BMS register is selected.

CMD - Command (Write)

9	8	7	6	5	4	3	2	1	0
AVS[1:0]		SBM	BBG	SKP[1:0]		INS	QMN	QMI	HDS

Address : 02-03
Type : W
Reset State : 0
Synchronization : SKP[1:0] with VSYNC, others none. Bit HDS is stored in a master-slave register to prevent undesired launching of header searches.

Description

The following write-only bits are used to control a variety of operations.

AVS[1:0] These bits select which data is available when the CMD register is read.

AVS[1]	AVS[0]	CMD data
0	0	CDcount[15:0]
0	1	CDcount[23:16]
1	0	SCDcount[15:0]
1	1	SCDcount[23:16]

SBM Select Block Move. When this bit is set block move operation is enabled and the BMS register is accessible at addresses 10 and 11. Otherwise (the default state) the MRF and MWF registers are accessible and read/write operations to the memory are enabled.

BBG Selects registers BBG, XFS, XFW, GCF2 and PLL. When this bit is set, the BBG register is accessible at addresses 24 and 25. Otherwise (the default state) the BBS register is accessible. The registers XFS and XFW are accessible at 2A-2B and 29. Otherwise (the default state) the DFS and DFW registers are accessible. The register GCF2 is accessible at 04-05 otherwise GCF1 (default state) is accessible. The register PLL is accessible at 06-07 otherwise (default state) the CTL register is available.

SKP[1:0] These bits are part of the instruction register, and are synchronized by the signal VSYNC. They define the skipping of one or two pictures, as defined in the following table :

SKP[1]	SKP[0]	
0	0	No skip (default)
0	1	Skip one picture, decode next
1	0	Skip two pictures, decode next
1	1	Illegal

If skipping is required, then these bits must be setup as part of the instruction for the picture which will be decoded. The skipping and the decoding of the following picture represent one task.

INS This bit selects which instruction register, INS1 or INS2, is accessible. When set, INS2 is selected, when reset (the default state), INS1 is selected.

QMN, QMI These two bits are used to control access to the inverse quantizer tables.

QMN	QMI	
0	0	Tables not accessible (lock)
0	1	Select the intra table
1	0	Select the non-intra table
1	1	Enable writing to the selected table

For example, to write a new intra table, the following steps are required :
Write CMD.QMI = 1, CMD.QMN = 0
Write CMD.QMI = CMD.QMN = 1
Write 64 weights to QMW
Write CMD.QMI = CMD.QMW = 0

HDS Writing a 1 to this bit starts a header search. Completion of the header search is indicated by the setting of bit STA.SCH.

CMD - Command (Read)

15	0
CD/SCDcount[15:8]	CD/SCDcount[23:16],[7:0]

Address :	02-03
Type :	R
Reset State :	0

Description

The contents of the bit buffer input and output counters, "CDcount" and "SCDcount" can be read from addresses 02 and 03. Which data is accessible is defined by the state of bits CMD.AVS[1:0].

AVS[1]	AVS[0]	CMD data
0	0	CDcount[15:0]
0	1	CDcount[23:16]
1	0	SCDcount[15:0]
1	1	SCDcount[23:16]

CDcount[23:16] or SDCcount[23:16] is available in bits CMD[7:0]. CDcount[15:0] or SCDcount[15:0] is available in bits CMD[15:0].

CSO - SRC Chrominance Offset

7	0
CSO[7:0]	

Address :	36
Type :	R/W
Reset State :	0
Synchronization :	VSYNC

Description

This register is set up with a value calculated from the fractional part of the pan vector. If no pan vector is defined, this register can be left in its reset (default) state.

The method of calculation of the CSO value is given in the PSV register description.

CSR - SRC Chrominance Resolution

7	0
CSR[7:0]	

Address :	37
Type :	R/W
Reset State :	0
Synchronization :	VSYNC

Description

The register holds the upsampling factor of the chrominance SRC (sample rate converter). This value must always be equal to that loaded into the LSR register.

The method of calculation of this value is given in the LSR register description.

CTL - Control

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
35A	DEC	S8M	PBO	MP2	HRD	EPR	CBC	EC3	EC2	ECK	EDI	EVI	PRS	SRS	EDC

Address :	06-07
Type :	R/W
Reset State :	0
Synchronization :	None

This is implemented as a master-slave register in order to prevent problems of erroneous commands being issued during the write cycle.

Description

35A Enable STi3500A features. When this bit is set, the additional functions of the STi3500A which are not compatible with the STi3500 are activated. When this bit is reset, the STi3520 behaves in a manner which is compatible with the STi3500.
The features which are enabled by this bit are :
- GCF.DFA[7:0] is defined.
- DFW, DFS are synchronized to DSYNC.
- YC output never has values 00h and FFh.
- PSV.V[7:1]and PSV.H[8] are defined.
- CSO, CSR, DCF.PXD, DCF.EOS, DCF.DSR, LSO, LSR, XDO, XDS, YDO, YDS are synchronized to VSYNC.
- XDO[9,8] and XDS[9,8] are defined.

DEC Disable Error Concealment. When this bit is set, automatic error concealment is disabled. This should only be done for debugging purposes, since the occurrence of a syntax error with error concealment disabled will cause decoding to stop. Recovery is only possible with a pipeline or a soft reset.

S8M Select 8-Mbit memory mode. When this bit is set, 8-Mbit memory mode is entered.

PBO Prevent bit Buffer Overflow. When this bit is set, bit buffer overflow (and thus the loss of data) is prevented by disabling the transfer of data from the compressed data FIFO to the bit buffer whenever the bit buffer level reaches the threshold defined in the BBT register.

MP2 MPEG-2 mode. When this bit is set, the STi3520 expects an MPEG-2 video bitstream. If it is reset, then an MPEG-1 bitstream is expected.

HRD Half-Resolution Display. This bit must be set if the decoded picture vertical resolution is half that of the display, i.e. the whole decoded picture is displayed in both fields. (This bit determines whether buffer overwriting can commence after 8 (CTL.HRD = 0) or 16 (CTL.HRD = 1) lines).

EPR Enable Pipeline Reset. When this bit is set, a pipeline reset is automatically generated if more than 64 samples are decoded in a block (i.e. in case of pipeline error).

CBC Circular Buffer Control. When this bit is set, the memory wraps at DFA and the overwrite mode is modified as described in section X.6.5.1 "Overwrite Mechanism".

EC3 Enable "Clock 3". When this bit is reset, the internal "clock 3" is disabled. This bit must be set for normal operation and for reduced power mode. It is reset in low power mode. (Power-down modes are defined in section X.1.2, "Power-Down Modes").

EC2 Enable "Clock 2". When this bit is reset, the internal "clock 2" is disabled. This bit must be set for normal operation, and reset in reduced and low power modes. (Power-down modes are defined in section X.1.2, "Power-Down Modes").

ECK Enable Clocks. When this bit is reset, all internal clocks are disabled. This bit must be set for normal operation and for reduced power mode. It is reset in low power mode. (Power-down modes are defined in section X.1.2, "Power-Down Modes").

EDI Enable DRAM Interface. When this bit is reset the DRAM interface (DD63-DD0, AA8-AA0, RAS1, RAS0, CAS, OE and WE) and the signal CDREQ are put into their high impedance state. This bit must be set for normal operation and for reduced power mode. It is reset in low power mode. (Power-down modes are defined in section X.1.2, "Power-Down Modes").

EVI Enable Video Interface. When this bit is reset the video interface (YC7 -YC0) is put into its high impedance state and the internal PIXCLK disabled. This bit must be set for normal operation and for reduced power mode (if the display interface is used). It is reset in low power mode. (Power-down modes are defined in section X.1.2, "Power-Down Modes").

PRS Pipeline Reset. In order to generate a pipeline reset, this bit must be kept set for a duration of at least 3 primary clock cycles (55ns with a 55MHz primary clock). The effect of a pipeline reset is described in section X.1.1, "Resets" .

SRS Soft Reset. In order to generate a soft reset, this bit must be kept set for a duration of at least 40 primary clock cycles (730ns with a 55MHz primary clock). The effect of a soft reset is described in section X.1.1, "Resets" .

EDC Enable Decoding. This bit must be set to enable the decoding pipeline to run. It's action is asynchronous and independent of the task control state machine (see Figure 29). If EDC is reset while the pipeline is idle, then it is still possible for the next instruction to be executed on the next VSYNC, and for a DSYNC to be generated. However, in this case, the decoding task will not start until EDC is set.

DCF - Display Configuration

15 14	13	12	11 9	8	7	6	5	4	3	2 0
OAD[1:0]	OAM	XYE	DAM[2:0]	FLD	USR	PXD	EVD	EOS	DSR	VFC[2:0]

Address : 38-39
Type : R/W
Reset State : 0
Synchronization : XYE, USR : None
Others : VSYNC

Description

OAD[1:0] OSD Active signal Delay. These bits are used to define the delay of the OSD active signal corresponding to output OSD pixels. Their programming is described in section X.4.7.3, "OSD Specification : STi3520 Mode".

OAM OSD Active signal Mode. When this bit is set, OSD active signal is an input. When it is reset, OSD active signal is an output. Note that OSD active signal must never be driven when CTL.EVI = 1 and DCF.OAM = 0.

XYE Select XDO/XDS Extensions. When this bit is set, register bits XD0[9,8] and XDS[9,8] are accessible at addresses 2Dh and 2Fh. When is it reset, XDO[7:0] and XDS[7:0] are accessible.

DAM[2:0] Display Access Mode. These bits are only active when bit DCF.USR is set. They are used in conjunction with the bit DCF.FLD. Their use is described in section X.4.9, "Displayed Field Sequence Control".

FLD Field bit. This bit is only active when bit DCF.USR is set. It defines which field is to be displayed : 1 for the bottom field, 0 for the top.
Bits DCF.DAM[2:0] and DCF.FLD must be programmed in conjunction with the bits DCF.VFC[2:0], as defined in the table appearing in section X.4.9, "Displayed Field Sequence Control".

USR Enable user field sequence control. When this bit is set, bits DCF.DAM[2:0] and DCF.FLD can be programmed field-by-field to enable a user-defined display sequence. Otherwise, these bits have no effect, and the fields are displayed in a sequence determined by the B/T signal.

PXD Add one PIXCLK Delay. When this bit is set the active video is delayed by one PIXCLK cycle with respect to HSYNC. This is to allow its horizontal position to be defined more precisely than is possible with XDO and XDS, which have a resolution of 2 PIXCLK cycles. Changing the value of PXD also has the effect of inverting the phasing of the Y/C output samples with respect to HSYNC.

EVD Enable Video Display. When this bit is reset, the video output has a constant value of Y = 16, C_B = C_R = 128. OSD is still displayed.

EOS Enable OSD. When this bit is set, the OSD (on-screen display) bitmap defined in the top and bottom field OSD buffers is displayed over the picture.

DSR Disable SRC. When this bit is set, both luminance and chrominance SRCs (sample rate converters) are disabled. In this case no horizontal filtering can occur, as would be required when the horizontal resolution of the decoded picture is equal to the horizontal resolution of the display.

VFC[2:0] Vertical Filter Configuration. These bits define the vertical filter mode.

VFC[2]	VFC[1]	VFC[0]	Mode
0	0	0	Full resolution, chrominance line repeat with interpolation
0	0	1	Full resolution, chrominance line repeat
0	1	0	Full resolution, chrominance field repeat with interpolation
0	1	1	Full resolution, chrominance field repeat
1	0	0	Half resolution, chrominance interpolation
1	0	1	Half resolution, chrominance repeat
1	1	0	Half resolution, luminance interpolation
1	1	1	Illegal

The vertical filtering modes are described in detail in section X.4.5, "Vertical Filter".

DFP - Displayed Frame Pointer

13 0

DFP[13:0]

Address : 1A-1B
Type : R/W
Reset State : 0
Synchronization : VSYNC

Description

This register holds the start address, defined in units of 32 words, of the frame which is currently being displayed. When a new value is written this is used at the start of the next field.

When DFP is set to same value as RFP (i.e. the decoder is writing the reconstructed picture into the buffer which is being displayed), bit INS1.OVW must be set.

DFS - Decoded Frame Size

13 0

DFS[13:0]

Address : 2A-2B
Type : R/W
Reset State : 0
Synchronization : DSYNC

Description

This register is set up with a value equal to the number of macroblocks in the decoded picture. This is derived from the horizontal_size and vertical_size values transmitted in the sequence header.

DFW - Decoded Frame Width

7 0

DFW[7:0]

Address : 29
Type : R/W
Reset State : 0
Synchronization : DSYNC

Description

This register is set up with a value equal to the width in macroblocks of the decoded picture. This is derived from the horizontal_size value transmitted in the sequence header.

FFP - Forward Frame Pointer

13 0

FFP[13:0]

Address : 1E-1F
Type : R/W
Reset State : 0
Synchronization : DSYNC

Description

This register holds the start address of the forward prediction frame picture buffer, defined in units of 32 words.

GCF1 - General Configuration

15 … 8	7	6 … 0
DFA[7:0]	M20	RFI[6:0]

Address : 04-05
Type : R/W
Reset State : 0
Synchronization : DFA[7:0] synchronized to DSYNC
Others : None

Description

DFA[7:0] DFS Adjustment. 4 x DFA[7:0] defines the offset, in units of macroblocks, between the end of the luminance storage area and the beginning of the chrominance storage area in the frame buffers. This offset is normally only used in 8-Mbit mode, in which it is set equal to the size in macroblocks of the picture overlap required for motion compensation minus 1, rounded up to the next highest multiple of 4. For example, if the maximum vertical vector range is -128 to 127 and the picture width is 720 pels, then the area needed for motion compensation is :
(720 x (128 + 16)/256) - 1
= 404 macroblocks
thus, DFA[7:0] = 101

M20 Select 20-Mbit memory mode. When this bit is set, 20-Mbit memory mode (one 16-Mbit bank plus one 4-Mbit bank) is entered.

RFI[6:0] DRAM Refresh Interval. This is defined in units of 24 primary clock periods. For example if each row must be refreshed every 8 ms, with a primary clock of 55MHz, the following value must be stored in GCF.RFI[6:0]:
$(8 \times 10^{-3}/512) \times (55 \times 10^{6}/24) \rightarrow 35$

GCF2 - General Configuration

15 … 8	7	6	5	4	3	2	1	0
XFA[7:0]		NPD	MRS	SQF	SGR	CLK	HPD	SDR

Address : 04-05
Type : R/W
Reset State : 0
Synchronization : XFA[7:0] synchronized to VSYNC
Others : None

Description

XFA[7:0] XFS Adjustment. 4 x XFA[7:0] defines the offset, in units of macroblocks, between the end of the luminance storage area and the beginning of the chrominance storage area in the frame buffers. This offset is normally only used in 8-Mbit mode, in which it is set equal to the size in macroblocks of the picture overlap required for motion compensation minus 1, rounded up to the next highest multiple of 4. For example, if the maximum vertical vector range is -128 to 127 and the picture width is 720 pels, then the area needed for motion compensation is :
(720 x (128 + 16)/256) - 1
= 404 macroblocks
thus, XFA[7:0] = 101

NPD Not Power Down. When this bit is reset the signal DQM is forced high turning off the output buffers of the memory and CKE is forced low disabling the internal clock in the DRAM. The memory is in power down mode. Setting this bit will re-enable the clock and other inputs asynchronously. A minimum setup time must be respected before any command is issued to the memory.

MRS Mode Register Set. When this bit is set the special power-on-reset sequence and mode register programming is carried out.When this bit is reset the procedure is inhibited.This bit can be programmed with the SDR selection bit in the GCF2 register.

SQF Select Quadruple Frame parameter buffers. When this bit is set the buffers for DFW,DFS and DFA are separated into two banks one for the decode process and one for the display process.

SGR Segments per Row. This bit is used by the memory controller to determine how a given DRAM should be addressed. The bit indicates the internal organisation of the DRAM to the memory controller. The bit is programmed in the following manner for the listed memory configurations.

SGR	Memory Configuration
0	Standard DRAM 4 x 4-Mbit, 2 x 4-Mbit
1	EDO DRAM 524.288 x 32-bit
1	Synchronous DRAM 524.288 x 16 bit x 2 banks

CLK Memory Subsystem Clock Enable. This bit when set enables the high frequency internal clock which is used in the generation of certain strobes. The clock must be enabled in EDO and Synchronous DRAM modes.

HPD EDO DRAM mode. This bit enables the EDO DRAM mode.

SDR Synchronous DRAM mode. This bit enables the Synchronous DRAM mode. When the bit is enabled the automatic initialisation sequence for the Synchronous DRAM is launched only if the MRS bit is set.

HDF - Header Data FIFO

15 0
HDF[15:0]

Address : 00-01
Type : R
Reset State : Undefined

Description

When the start code detector has found a start code, the header data FIFO must be read in order to identify the start code and if required to obtain the header data. The start code identification procedure is described in section X.2.2, "Start Code Detection".

Before reading the header FIFO, status bit STA.HFE should be checked to ensure that it is not empty. Bit STA.HFF set indicates that the header FIFO contains at least 66 bytes of data.

INS1 - Instruction 1

15	14	13 10	9 6	5 4	3	2	1	0
TFF	OVW	BFH[3:0]	FFH[3:0]	PCT[1] PCT[0]	SEQ	EXE	RPT	CMV

Address : 0E-0F
Type : R/W
Reset State : 0
Synchronization : "new instruction" (see Figure 23), except bits OVW, SEQ, EXE and RPT, which are updated by VSYNC

Description

This register contains 16 bits of the decoding instruction. The other bits are CMD.SKP[1:0] and INS2[15:0]. The mechanism of instruction execution is described in section X.3.4, "Decoding Task Control".

TFF This bit is set equal to the top_field_first bit of the MPEG-2 picture coding extension. It is only taken into account when decoding MPEG-2 dual-prime pictures.

OVW This bit must be set when the displayed picture and the reconstructed picture share the same buffer (i.e. DFP = RFP). It enables the overwrite mode which ensures that the reconstructed picture does not overwrite data which has not yet been displayed.

BFH[3:0] In MPEG-1 mode BFH[3] is set equal to full_pel_backward_vector of the picture header, and BFH[2:0] is set equal to backward_f_code of the picture header.
In MPEG-2 mode BFH[3:0] is set equal to backward_horizontal_f_code of the picture coding extension.

FFH[3:0] In MPEG-1 mode FFH[3] is set equal to full_pel_forward_vector of the picture header, and FFH[2:0] is set equal to forward_f_code of the picture header.
In MPEG-2 mode BFH[3:0] is set equal to forward_horizontal_f_code of the picture coding extension.

PCT[1:0] This is set equal to the two least significant bits of picture_coding_type in the picture header.

SEQ Search for next Sequence. This bit must be set while the decoder is searching for the start of a new sequence after a soft reset. It has the effect of putting the controller into a state in which VSYNCs are ignored.

EXE When this bit is not set, no decoding task is executed for one or two VSYNC periods, depending on the state of RPT.

RPT When this bit is set, the task duration is two VSYNC periods. In many applications, when the frame display rate is equal to the picture decoding rate, RPT will always be set.

CMV This bit is set equal to the concealment_motion_vectors bit of the MPEG-2 picture coding extension. It indicates that motion vectors are coded for intra macroblocks. It is only taken into account when decoding MPEG-2 pictures.

This register shares the same addresses as the INS2 register. Bit CMD.INS selects which of the two registers is accessible at these addresses. When this bit is reset, the INS1 register is selected.

INS2 - Instruction 2

15 14	13 10	9 6	5 4	3	2	1	0
PST[1] PST[0]	BFV[3:0]	FFV[3:0]	DCP[1] DCP[0]		QST FRM	AZZ	IVF

Address : 0E-0F
Type : R/W
Reset State : 0
Synchronization : "new instruction"
(see Figure 23)

Description

This register contains 16 bits of the decoding instruction. The other bits are CMD.SKP[1:0] and INS1[15:0]. The mechanism of instruction execution is described in section X.3.4, "Decoding Task Control".

In MPEG-1 mode (i.e. when CTL.MP2 is reset), INS2 is not used, and must be kept reset.

PST[1:0] This is set equal the picture_structure bits of the MPEG-2 picture coding extension. Note that code "00" also indicates frame structure, even though this value is illegal in the MPEG-2 variable.

PST[1]	PST[0]	
0	0	Frame picture
0	1	Top field
1	0	Bottom field
1	1	Frame picture

BFV[3:0] INS2.BFV[3:0] is set equal to backward_vertical_f_code of the picture coding extension.

FFV[3:0] INS2.FFV[3:0] is set equal to forward_vertical_f_code of the picture coding extension.

DCP[1:0] INS2.DCP[1:0] is set equal to intra_dc_precision of the picture coding extension. The value "11", defining a precision of 11 bits, is not allowed.

FRM This bit is set equal to the frame_pred_frame_dct bit of the picture coding extension.

QST This bit is set equal to the q_scale_type bit of the picture coding extension.

AZZ This bit is set equal to the alternate_scan bit of the picture coding extension.

IVF This bit is set equal to the intra_vlc_format bit of the picture coding extension.

ITM - Interrupt Mask

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDE	SER	BMI	HFF	RFF	WFE	PID	PER	PSD	VST	VSB	BBE	BBF	HFE	BFF	SCH

Address : 0A-0B
Type : R/W
Reset State : 0 (all interrupts disabled)
Synchronization : None

Description

Any bit set in this register will enable the corresponding interrupt. An interrupt is generated whenever at bit in the STA register changes from 0 to 1 and the corresponding mask bit is set.

ITS - Interrupt Status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDE	SER	BMI	HFF	RFF	WFE	PID	PER	PSD	VST	VSB	BBE	BBF	HFE	BFF	SCH

Address : 0C-0D
Type : R
Reset State : 0

After clocks have been enabled, the state changes to be the same as that of STA.

Description

When a bit in the STA register changes from 0 to 1, the corresponding bit in the ITS register is set, independent of the state of ITM. If any set ITS bit is unmasked, the signal $\overline{IRQ}$ is asserted. Reading the most significant byte of ITS clears it, leaving $\overline{IRQ}$ in its de-asserted (high) state.

See section VI.4, "Interrupts (video)" for more information on interrupt handling.

LSO - SRC Luminance Offset

7	0
LSO[7:0]	

Address : 34
Type : R/W
Reset State : 0
Synchronization : VSYNC

Description

This register is set up with a value calculated from the fractional part of the pan vector. If no pan vector is defined, this register can be left in its reset (default) state.

The method of calculation of the LSO value is given in the PSV register description.

LSR - SRC Luminance Resolution

7	0
LSR[7:0]	

Address : 35
Type : R/W
Reset State : 0
Synchronization : VSYNC

Description

This register holds the upsampling factor of the luminance SRC (sample rate converter).

The upsampling factor is equal to 256/LSR. Below are given some examples of upsampling factors, where in each case the displayed picture has a nominal width of 720 pels*. Also shown are the numbers of valid pels generated, "N", calculated as shown in section X.4.4.1, "Sample Rate Converter".

* Displayed picture widths other than 720 can of course be supported.

Decoded Picture Width	LSR	N
640	228	715
640	227	718
544	193	717
544	192	721
480	170	717
480	169	722
352	125	713
352	124	719
704	250	717
704	249	720

MRF - Memory Read FIFO

7	0
MRF[7:0]	

Address : 10
Type : R
Reset State : Undefined

Description

The memory read FIFO data is read from this address. The mechanism of data transfer from the memory into the FIFO is described in section VII.5, "Memory Read and Writethrough the Microcontroller Interface". Each 64-bit (or 32-bit, when in 8-Mbit mode) word is accessed most significant-byte-first.

This register shares the same address as the most significant byte of the BMS register. Bit CMD.SBM selects whether MRF or BMS is accessible at this address. When this bit is reset, the MRF register is selected.

MRP - Memory Read Pointer

18	0
MRP[18:0]	

Address : 13-14-15
Type : R/W
Reset State : Undefined
Synchronization : None

Description

This register holds the address of the word which will be transferred next from the memory into the memory read FIFO. It is incremented each time a word is transferred from the memory to the FIFO. It is thus not the address of the word currently available in the FIFO, but one or two addresses ahead.

MRP is only set up at the beginning of a memory read sequence, at which time the memory read FIFO is cleared. The mechanism of data transfer from the memory into the FIFO is described in section VII.5,"Memory Read and Writethrough the Microcontroller Interface".

MWF - Memory Write FIFO

7	0
MWF[7:0]	

Address : 11
Type : W
Reset State : Undefined
Synchronization : None (not a register)

Description

The memory write FIFO data is written to this address. The mechanism of data transfer from the FIFO to the memory is described in section VII.5, "Memory Read and Write through the Microcontroller Interface". Each 64-bit (or 32-bit, when in 8-Mbit mode) word must be written most significant-byte-first.

This register shares the same address as the least significant byte of the BMS register. Bit CMD.SBM selects whether MWF or BMS is accessible at this address. When this bit is reset, the MWF register is selected.

MWP - Memory Write Pointer

18	0
MWP[18:0]	

Address : 17-18-19
Type : R/W
Reset State : Undefined
Synchronization : None

Description

This register holds the address of the word which will be transferred next from the memory write FIFO into the memory. It is incremented each time a word is transferred from the FIFO to the memory. It is thus not the address of the word last written to the FIFO, but one or two addresses behind.

MWP is only set up at the beginning of a memory write sequence, at which time the memory write FIFO is cleared. The mechanism of data transfer from the FIFO to the memory is described in section VII.5, "Memory Read and Write through the Microcontroller Interface".

OBP - OSD Bottom Field Pointer*

13	0
OBP[13:0]	

Address : 30-31
Type : R/W
Reset State : 0
Synchronization : VSYNC bottom

Description

The register holds the start address, in units of 32 words, of the current OSD specification buffer for the bottom field. This specification will be decoded during bottom fields when OSD is enabled (bit DCT.EOS is set).

* Formerly called OEP.

OTP - OSD Top Field Pointer*

13	0
OTP[13:0]	

Address : 32-33
Type : R/W
Reset State : 0
Synchronization : VSYNC top

Description

The register holds the start address, in units of 32 words, of the current OSD specification buffer for the top field. This specification will be decoded during top fields when OSD is enabled (bit DCF.EOS is set).

* Formerly called OOP.

PLL - Phased Locked Loop Configuration Register

11	10	9	8	7	6	5 2	1 0
SPT		SMC	SAC	SVC	SRC	M[3:0]	S[1:0]

Address : 06-07
Type : R/W
Reset State : 0
Synchronization : None

S[1:0] Audio clock divider. This divider must be set to keep the audio decoder clock in the range 24 - 30MHz. The PLL output frequency can be calculated as a function of M and the reference frequency. The audio clock divider must then be set for a division by 8 , 9 or 10 to respect the clock frequency limits.

S[1:0]	Division Factor
00	10
01	9
10	8
11	8

M[3:0] PLL Multiplication Factor, M. These bits control the reference frequency input to the clock generator's PLL. The actual multiplication factor is M + 6. This bit corresponds to "M" in E. 2 in section IX.3.

SRC Select Reference Clock. This bit when set selects an external clock for the reference clock. The default (reset state) is pixclk.

SVC Select Video Clock. This bit when set selects the video decoder clock generated by the PLL. The default (reset state) is that the video decoder clock is generated externally.

SAC Select Audio Clock. This bit when set selects the audio clock generated by the PLL. The default (reset state) is that the audio decoder clock is generated externally.

SMC Select Memory subsystem Clock. This bit when set allows an externally generated memory clock to be connected to Pin 61 in both the EDO and Synchronous DRAM modes. The clock, if supplied externally, must be locked to the video decoder clock which must also be supplied externally. For more details and a schematic see section VII.3. The default condition is that the clock is generated by the PLL.

SPT Select PLL Test mode. This bit should be programmed to zero.

PSV - Pan/Scan Vector

15 9	8 0
V[7:1]	H[8:0]

Address : 3A-3B
Type : R/W
Reset State : 0
Synchronization : VSYNC

Description

V[7:1] Vertical Offset. The vertical component of the pan/scan vector is programmed into V[7:1], in units of 2 field lines, or 4 frame lines. For example, if V[7:1] is set to the value 16, then the 32 top lines of every field would not be output to the display.

H[8:0] Horizontal Offset. These bits are set up with the integer part of the horizontal component of the pan/scan vector. This number defines, in the decoded picture, the location of the first displayed luminance sample relative to the first luminance sample in the line. The LSO and CSO registers are set up with the fractional part of the pan vector, as follows :
PSV = ⌊pan vector⌋
where " ⌊x⌋ " indicates the integer part of x.
LSO = 256 x (pan vector - PSV)
CSO = LSO/2 (+ 1 if PSV is odd)
Refer to section X.4.4.1, "Sample Rate Converter" for more details.

QMW - Quantization Matrix Data

7 0
Q[7:0]

Address : 3C or 3D
Type : W
Reset State : Undefined
Synchronization : None (not a register)

Description

To either of these addresses are written the quantization coefficients in the order in which they appear in the bitstream, i.e. in zig-zag order. Which matrix (intra or non-intra) is written is defined by bits CMD.QMN and CMD.QMI.

There are no built-in default quantization matrices.

RFP - Reconstructed Frame Pointer

13 0
RFP[13:0]

Address : 1C-1D
Type : R/W
Reset State : 0
Synchronization : DSYNC

Description

This register holds the start address of the reconstructed (decoded) frame picture buffer, defined in units of 32 words.

STA - Status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDE	SER	BMI	HFF	RFF	WFE	PID	PER	PSD	VST	VSB	BBE	BBF	HFE	BFF	SCH

Address :	08-09
Type :	R
Reset State :	0

After clocks have been enabled, the state changes to :

PDE	=	0
SER	=	0
BMI	=	1
HFF	=	0
RFF	=	1
WFE	=	1
PID	=	1
PER	=	0
PSD	=	0
VST	=	0
VSB	=	0
BBE	=	1
BBF	=	1
HFE	=	1
BFF	=	0
SCH	=	0

Description

This register contains a set of bits which represent the status of the decoder at any instant. Any change from 0 to 1 of any of these bits sets the corresponding bit of the ITS register, and can thus potentially cause an interrupt.

The status vector is sampled internally at the start of the read cycle accessing the most significant byte of STA (address 08). VST, VSB and PSD are pulses and are unlikely ever to be read as a 1.

The status bits have the following significance :

PDE Picture Decoding Error or underflow error. This bit is set when less than the programmed number of macroblocks (defined by DFS) have been decoded, either due to a data or a programming error. Decoding is halted automatically when this error condition is detected. This bit is reset by all 3 types of reset.

SER Severe Error or overflow error. This bit is set when more than the programmed number of macroblocks (defined by DFS) have been decoded, either due to a data or a programming error. Decoding is halted automatically when this error condition is detected. This bit is reset by all 3 types of reset.

BMI Block Move Idle. This bit is set when a block move operation has terminated. It is automatically reset at the start of a block move.

HFF Header FIFO Full. This bit is set when the header FIFO contains at least 66 bytes.

RFF Read FIFO Full. This bit is set when the memory read FIFO contains 16 bytes.

WFE Write FIFO Empty. This bit is set when the memory write FIFO is empty.

PID Pipeline Idle. This bit is set when the STi3520 is not in the course of decoding a picture, i.e. when the pipeline is inactive. It becomes low when the decoding of a picture starts and high when picture decoding is complete.

PER Pipeline Error. This bit is set when due to a data error, more than 64 coefficients are reconstructed for a block. If bit CTL.EPR is set, a pipeline reset is generated. This bit is reset by all 3 types of reset.

PSD Pipeline Starting to Decode. This bit is set for a short period at the instant the pipeline starts decoding a picture.

VST VSYNC Top. This bit is set for a short time at the beginning of the top field, corresponding to the falling edge of the B/$\overline{T}$ signal.

VSB VSYNC Bottom. This bit is set for a short time at the beginning of the bottom field, corresponding to the rising edge of the B/$\overline{T}$ signal.

BBE Bit Buffer Empty. This bit is set when the bit buffer contains no data.

BBF Bit Buffer Full. This bit is set when the bit buffer level (= BBL) is greater than or equal to the value loaded into the BBT register.

HFE Header FIFO Empty. This bit is set when the header FIFO is empty.

BFF Compressed Data (bitstream) FIFO Full. This bit is set when the CD FIFO is full. This bit is equivalent to the signal $\overline{\text{CDREQ}}$.

SCH Start Code Hit. This bit is set whenever the first 16-bit word available in the header FIFO contains one of the start codes recognised (see section X.2.2, "Start Code Detection"). While data is being read from the header FIFO, this bit can be tested to determine whether the next word contains a start code.

XDO - Display X Offset

1	0
XDO[9, 8]	

7	0
XDO[7:0]	

Address :	2D
Type :	R/W
Reset State :	0
Synchronization :	VSYNC

Description

This register is set up with a number defining the first active video sample of each line. Preceding pels are output as black (unless OSD is defined for this region).

When bit DCF.XYE is set, XDO[9,8] is accessible. When bit DCF.XYE is reset, XDO[7:0] is accessible.

XDO programming is described in section X.4.3, "Setting up the Display".

XDS - Display X End

1	0
XDS[9, 8]	

7	0
XDS[7:0]	

Address :	2F
Type :	R/W
Reset State :	0
Synchronization :	VSYNC

Description

This register is set up with a number defining the last active video sample of each line. Succeeding pels are output as black (unless OSD is defined for this region).

When bit DCF.XYE is set, XDS[9,8] is accessible. When bit DCF.XYE is reset, XDS[7:0] is accessible.

Note : When bit CTL.35A is set, care must be taken to set up XDS[9,8], since the default value is 0, while in the STi3500, in which these bits are not programmable, this value is 3.

XDS programming is described in section X.4.3, "Setting up the Display".

XFS - Decoded Frame Size

13	0
XFS[13:0]	

Address :	2A-2B
Type :	R/W
Reset State :	0
Synchronization :	VSYNC

Description

This register is set up with a value equal to the number of macroblocks in the decoded picture. This is derived from the horizontal_size and vertical_size values transmitted in the sequence header. See section X.4.6, "Decoding and Display across Sequence Boundaries".

XFW - Decoded Frame Width

7	0
XFW[7:0]	

Address :	29
Type :	R/W
Reset State :	0
Synchronization :	VSYNC

Description

This register is set up with a value equal to the width in macroblocks of the decoded picture. This is derived from the horizontal_size value transmitted in the sequenceheader. See section X.4.6, "Decoding and Display across Sequence Boundaries".

YDO - Display Y Offset

7	0
YDO[7:0]	

Address :	2C
Type :	R/W
Reset State :	0
Synchronization :	VSYNC

Description

This register is set up with a number defining the first active video line of a field. Preceding lines are output as black (unless OSD is defined for this region).

YDO programming is described in section X.4.3, "Setting up the Display".

YDS - Display Y End

7	0
YDS[7:0]	

Address :	2E
Type :	R/W
Reset State :	0
Synchronization :	VSYNC

Description

This register is set up with a number defining the last active video line of a field. Succeeding lines are output as black (unless OSD is defined for this region).

YDS programming is described in section X.4.3, "Setting up the Display".

XV - AUDIO REGISTERS

Registers are listed in alphabetical order.

All addresses are in hexadecimal.

All unspecified addresses of the register map are reserved, and must not be written to. Unspecified bits of user-accessible registers have no function.

All unspecified bits of the register map are reserved. Only the value 0 must be written to any of these bits. The values which are read from these bits are undefined.

The reset state is the state existing after a hard reset.

ANC - Ancillary Data Buffer

	7 0
09	ANC[31:24]
08	ANC[23:16]
07	ANC[15:8]
06	ANC[7:0]

Address :	09-06
Type :	R
Reset State :	Undefined

Description

The 4 8-bit ancillary data registers constitute a 32-bit FIFO which holds the ancillary data extracted from audio frames. The first bit of ancillary data received is stored in bit ANC[0].

The extraction of ancillary data in ANC is started by enabling interrupt 7. An interrupt 7 is generated when 32 bits have been written into ANC, i.e. when it is full.

When ANC[31:24] is read, ANC_AV is cleared and the ancillary data buffer is reinitialized.

ANC_AV - Ancillary Data Available

5 0
ANC_AV[5:0]

Address :	6C
Type :	R
Reset State :	0

Description

This register holds the number of bits available in the ancillary data buffer, ANC[31:0].

It is cleared by reading ANC[31:24].

ATTEN_L - Left Channel Attenuation

5 0
ATTEN_L[5:0]

Address :	1E
Type :	R/W
Reset State :	Undefined

Description

This register defines the left channel attenuation in steps of 2dB. The minimum attenuation is 0dB, the maximum is 2 x 63 = 126dB.

ATTEN_R - Right Channel Attenuation

5 0
ATTEN_R[5:0]

Address :	20
Type :	R/W
Reset State :	Undefined

Description

This register defines the right channel attenuation in steps of 2dB. The minimum attenuation is 0dB, the maximum is 2 x 63 = 126dB.

AUD_ID - Audio Stream ID

4 0
AUD_ID[4:0]

Address :	22
Type :	R/W
Reset State :	Undefined

Description

This value stored in this register is only taken into account if AUD_ID_EN is set.

This register specifies the number (between 0 and 31) of the audio stream which is to be decoded. The stream number is defined in the field stream_id of the packet header. All other packets will be discarded.

If AUD_ID_EN is reset, then all audio packets are decoded.

AUD_ID_EN - Audio Stream ID Enable

Bit 0
AUD_ID_EN

Address :	24
Type :	R/W
Reset State :	Undefined

Description

If this bit is reset, then the contents of AUD_ID are ignored.

If it is set, then the register AUD_ID is taken into account.

0 : Ignore AUD_ID.
1 : Use AUD_ID.

CRC_ECM - CRC Error Concealment Mode

1	0
CRC_ECM[1:0]	

Address :	2A
Type :	R/W
Reset State :	Undefined

Description

This register defines the action which will be taken upon detection of a CRC error in an input frame.

00 : Disable CRC detection and error concealment.
01 : Mute on detection of CRC error.
10 : Illegal.
11 : Skip invalid frame.

DATAIN - Compressed Data Input

7	0
DATAIN[7:0]	

Address :	18
Type :	R/W
Reset State :	Undefined

Description

This register is provided to allow the transfer of compressed data across the microcontroller interface. When this mode of data entry is used, the signal $\overline{\text{DSTRB}}$ must be inactive.

DMPH - De-emphasis Mode

1	0
DMPH[1:0]	

Address :	46
Type :	R
Reset State :	Undefined

Description

This register is set with the value of the emphasis field of the frame currently being decoded.

00 : None.
01 : 50/15µs.
10 : Reserved value.
11 : ITU-T J.17.

FIFO_THRES - Input FIFO Threshold

7	0
FIFO_THRES[7:0]	

Address :	52
Type :	R/W
Reset State :	Undefined

Description

This value loaded into this register defines the input FIFO level at which an interrupt 12 can be generated. The level is defined as a byte address, in the range 0 to 255.

An interrupt can be generated each time the FIFO level is equal to FIFO_THRES, regardless of whether it was approached from above or below.

FREE_FORM - Free Format Frame Length

	7	0
15	FREE_FORM[15:8]	
14	FREE_FORM[7:0]	

Address :	15-14
Type :	R/W
Reset State :	Undefined

Description

When free-format decoding is used (bitrate_index = 0), the frame length, if known, can be loaded into this register, in units of bits. (In free-format, the frame length cannot be determined from bitstream parameters).

The length loaded into FREE_FORM is used in the internal synchronization algorithm.

If the frame length is not known, FREE_FORM must be loaded with zero.

HEADER - Frame Header

	7	0
61	HEADER[31:24]	
60	HEADER[23:16]	
5F	HEADER[15:8]	
5E	HEADER[7:0]	

Address : 61-5E
Type : R
Reset State : Undefined

Description

This 32-bit register contains the header of the frame currently being decoded.

This register is upd ated after interrupt 1 is enabled. An interrupt 1 is generated when a valid header has been received.

The contents are retained until HEADER[31:24] is read.

INTR - Interrupt Request Register

	6	0
1B	INTR[14:8]	
	7	0
1A	INTR[7:0]	

Address : 1B-1A
Type : R
Reset State : 0. Also cleared on restart

Description

An interrupt is signalled by a 100ns pulse on $\overline{IRQ}$ whenever one of the bits of INTR becomes set. This can only occur if the corresponding bit is set in the INTR_EN register. The INTR register is cleared on reset (assertion of $\overline{RESET}$ pin or setting of RESET register), or restart (setting the RESTART register). Also the most significant byte, and bits 3-5 of the least significant byte of INTR can be independently cleared by reading. Bits 0-2 and 7 are cleared by a different method, as indicated in the notes below the following table :

N°	Condition Signalled
14	First bit of new frame at PCM output
13	Input FIFO full
12	Input FIFO level = FIFO_THRES
11	
10	De-emphasis changed
9	Sampling frequency changed
8	PCM output buffer underflow
7	Ancillary data register full (1)
6	Not used
5	CRC error detected
4	
3	
2	Valid PTS registered (2)
1	Valid header registered (3)
0	Change in synchronization status (4)

Notes :
1. ANC[31:24] must be read in order to clear bit INTR[7] and to reinitialize the ancillary data buffer.
2. PTS[32] must be read in order to clear bit INTR[2] and to reinitialize the PTS register.
3. HEADER[31:24] must be read in order to clear bit INTR[1] and to reinitialize the HEADER register.
4. SYNC_ST must be read in order to clear bit INTR[0] and to reinitialize the SYNC_ST register.

INTR_EN - Interrupt Mask Register

	6	0
1D	INTR_EN[14:8]	
	7	0
1C	INTR_EN[7:0]	

Address : 1D-1C
Type : R/W
Reset State : 0. Also cleared on restart

Description

A one in any bit position of this register will enable the corresponding bit of the INTR register.

In addition, setting certain bits of this register have additional actions, as specified below :

- setting INTR_EN[7] enables the reading of ancillary data into ANC,
- setting INTR_EN[2] enables the updating of the PTS register,
- setting INTR_EN[1] enables the updating of the HEADER register,
- setting INTR_EN[0] enables the updating of the SYNC_ST register.

INV_LRCLK - LRCLK Polarity

Bit 0
INV_LRCLK

Address : 11
Type : R/W
Reset State : 0 after assertion of $\overline{RESET}$ Pin only

Description

This bit is used to define the polarity of the output signal LRCLK.

0 : Left channel when LRCLK = 1.
1 : Left channel when LRCLK = 0.

INV_SCLK - SCLK Polarity

Bit 0
INV_SCLK

Address : 53
Type : R/W
Reset State : 0 after assertion of $\overline{RESET}$ Pin only

Description

This bit defines the polarity of the PCM bit clock output SCLK.

0 : The LRCLK and PCMDATA outputs change on the falling edge of SCLK. The external DAC will sample LRCLK and PCMDATA on the rising edge of SCLK.

1 : The LRCLK and PCMDATA outputs change on the rising edge of SCLK. The external DAC will sample LRCLK and PCMDATA on the falling edge of SCLK.

LATENCY - Latency Selection

Bit 0
LATENCY

Address :	3C
Type :	R/W
Reset State :	Undefined

Description

This bit is used to select the latency mode of the decoder.

0 : Low latency mode. The decoding latency is less than 5 ms (see section VI.10, "Audio Decoder Latency").

1 : Illegal.

MUTE - Mute

Bit 0
MUTE

Address :	30
Type :	R/W
Reset State :	0 after assertion of $\overline{\text{RESET}}$ Pin only

Description

The action of the mute command is explained in section XI.2.3, "Play and Mute".

PACKET_SYNC - Packet Sync Mode

1	0
PACKET_SYNC[1:0]	

Address :	23
Type :	R/W
Reset State :	Undefined

Description

This register defines the packet synchronization mode.

00 : Synchronize only on 24-bit packet_start_code_prefix. (Multiplexed audio/video bitstream).

01 : Synchronize both on 24-bit packet_start_code_prefix and first 3 bits of stream_id. (Multiplexed audio bitstream).

10 : Synchronize both on 24-bit packet_start_code_prefix and all 8 bits of stream_id. (Audio bitstream with unique id).

PCM_18 - PCM Output Precision

Bit 0
PCM_18

Address :	16
Type :	R/W
Reset State :	Undefined

Description

This bit defines the PCM output precision.

0 : 16-bit PCM data output.

1 : 18-bit PCM data output.

PCM_DIF - PCM Output Justification

(DIF : Data In Front)

Bit 0
PCM_DIF

Address :	6F
Type :	R/W
Reset State :	Undefined

Description

This bit selects whether the PCM output data is right or left justified with respect to the 32-clock frame when in 18-bit mode (i.e. when PCM_18 = 1). This bit is has no significance when in 16-bit mode (i.e. when PCM_18 = 0).

0 : 18-bit PCM data is right justified (i.e. occupies the last 18 cycles of each 32-cycle frame).

1 : 18-bit PCM data is left justified (i.e. occupies the first 18 cycles of each 32-cycle frame when PCM_FORMAT = 1, or the next 18 when PCM_FORMAT = 0).

PCM_DIV - PCM Clock Divider

5	0
PCM_DIV[5:0]	

Address : 6E
Type : R/W
Reset State : Undefined

Description

The number loaded into this register, in the range 0 to 63, defines the ratio of the frequency of the PCM bit clock, SCLK, to that of PCMCLK, according to the relationship :

$$f_{SCLK}=\frac{f_{PCMCLK}}{2\cdot(PCM\_DIV+1)}$$

For example, PCM_DIV is loaded with 0, the frequency of SCLK is one half of the frequency of PCMCLK, while if PCM_DIV is loaded with 63, the frequency of SCLK is one 128th of the frequency of PCMCLK.

The value of PCM_DIV = 16 is reserved. If this number is loaded, the divider is bypassed and the frequency of SCLK is equal to the frequency of PCMCLK.

PCM_DIV must be set up before the output of SCLK starts. This can be done by first disabling PCM outputs by de-asserting the MUTE and PLAY commands, and then writing to the PCM_DIV register. Once the register is set up, the MUTE and/or PLAY commands can be asserted. PCM_DIV cannot be changed "on the fly".

PCM_FORMAT - PCM Output Format

Bit 0
PCM_FORMAT

Address : 19
Type : R/W
Reset State : Undefined

Description

This bit is used to select the I^2S compatible PCM output format when in 18-bit left-justified mode (i.e. when PCM_18 = 1 and PCM_DIF = 1). In this mode the most- significant bit of the PCM data is output one cycle later than the change of LRCLK. PCM_FORMAT has no significance when PCM_18 = 0.

0 : I^2S compatible PCM output.
1 : Standard format (most-significant bit of data coincident with LRCLK).

PCM_FS - Sampling Frequency

1	0
PCM_FS[1:0]	

Address : 44
Type : R
Reset State : Undefined

Description

This register is loaded with the sampling frequency code extracted from the bitstream. This is equal to the frequency of the output LRCLK.

00 : 44.1kHz.
01 : 48kHz.
10 : 32kHz.
11 : Reserved.

PCM_ORD - PCM Output Bit Order

Bit 0
PCM_ORD

Address : 38
Type : R/W
Reset State : Undefined

Description

This bit determines the order of PCM data output when in 16-bit mode (i.e. when PCM_18 = 0). It has no significance when PCM_18 = 1, when data is always output most-significant bit first.

0 : Most-significant bit output first.
1 : Least-significant bit output first.

PLAY - Play

Bit 0
PLAY

Address : 2E
Type : R/W
Reset State : 0 after assertion of $\overline{RESET}$ Pin only

Description

The action of the mute command is explained in section XI.2.3, "Play and Mute".

PTS - Presentation Time Stamp

	7		0
66			32
65	PTS[31:24]		
64	PTS[23:16]		
63	PTS[15:8]]		
62	PTS[7:0]		

Address :	66-62
Type :	R
Reset State :	Undefined

Description

This 33-bit register contains the PTS associated with the frame currently being decoded.

This register is upd ated after interrupt 2 is enabled. An interrupt 2 is generated when a valid PTS has been received.

The contents are retained until PTS[32] is read.

RESET - Software Reset

Bit 0
RESET

Address :	40
Type :	R/W
Reset State :	0 (command)

Description

Writing a 0 or 1 to this bit has an equivalent function to asserting the hardware reset Pin, $\overline{\text{RESET}}$, except that the following registers are not cleared :

- INV_SCLK
- MUTE
- PLAY
- STC_INC
- STC_DIV
- STC_CTL

This bit is reset automatically after being set.

STi3520 audio decoder reset actions are described in section XI.2.2, "Initialization of the Audio Decoder".

RESTART - Restart

Bit 0
RESTART

Address :	42
Type :	R/W
Reset State :	0 (command)

Description

When this bit is set, all data buffers are flushed, and then the RESTART bit is automatically reset. In addition the INTR and INTR_EN registers are cleared.

SIN_EN - Enable Serial Input

Bit 0
SIN_EN

Address :	70
Type :	R/W
Reset State :	Undefined

Description

This bit determines whether compressed data input is serial, or parallel (input through SDATA[7:0] Pins).

0 : Parallel data input.
1 : Illegal.

SKIP - Skip Next Frame

Bit 0
SKIP

Address :	32
Type :	R/W
Reset State :	0 (command)

Description

When this bit is set, the next audio frame is skipped, and then the SKIP bit is automatically reset.

STR_SEL - Input Stream Selection

1	0
STR_SEL[1:0]	

Address :	36
Type :	R/W
Reset State :	Undefined

Description

This register defines the type of audio input bitstream expected by the STi3520.

00 : MPEG audio elementary stream.
01 : MPEG packet stream.
10 : Reserved.
11 : PCM data (select bypass mode).

SYNC_CONF - Sync Confirmation Mode

Bit 0
SYNC_CONF

Address :	25
Type :	R/W
Reset State :	Undefined

Description

This bit selects one of two options in the packet synchronization algorithm.

0 : After the first valid packet start code and stream id are found, the packet length is used to locate the next start code and stream id before synchronization is confirmed and audio decoding starts.

1 : Synchronization is confirmed when the first valid packet start code and stream id are found.

SYNC_ECM - Sync Error Concealment Mode

1 0

SYNC_ECM[1:0]

Address :	2C
Type :	R/W
Reset State :	Undefined

Description

This register defines the action which will be taken upon detection of a synchronization error.

00 : Ignore error.
01 : Mute on detection of synchronization error.
10 : Illegal.
11 : Skip invalid frame.

SYNC_LOCK - Sync Words Until Lock

1 0

SYNC_LOCK[1:0]

Address :	28
Type :	R/W
Reset State :	Undefined

Description

This register defines how many valid synchronization words after the initial one must be found before locking audio frame synchronization.

When SYNC_REG is set to its default value, the STi3520 assumes that SYNC_LOCK is set to the value 3.

Synchronization error concealment is still enabled when SYNC_LOCK has the value zero.

STC_REG - Sync Word Extension

7 0

STC_REG[7:0]

Address :	27
Type :	R/W
Reset State :	Undefined

Description

This register defines an extension to the frame synchronization word which can be used to increase the reliability of synchronization when the layer number, bit rate or sampling frequency are known. The three fields are defined in thefollowing tables.

Programming of this register is mandatory.

Layer Field

Bits 7, 6	
11	Layer I
10	Layer II
00	This field not used

Bit Rate Field

Bits 5, 2	Layer I	Layer II
0000	Free format	Free format
0001	32 kbit/s	32 kbit/s
0010	64 kbits	48 kbit/s
0011	96 kbit/s	56 kbit/s
0100	128 kbit/s	64 kbit/s
0101	160 kbit/s	80 kbit/s
0110	192 kbit/s	96 kbit/s
0111	224 kbit/s	112 kbit/s
1000	256 kbit/s	128 kbit/s
1001	288 kbit/s	160 kbit/s
1010	320 kbit/s	192 kbit/s
1011	352 kbit/s	224 kbit/s
1100	384 kbit/s	256 kbit/s
1101	416 kbit/s	320 kbit/s
1110	448 kbit/s	384 kbit/s
1111	This field not used	This field not used

Sampling Frequency Field

Bits 1, 0	
00	44.1kHz
01	48kHz
10	32kHz
11	This field not used

Examples :

- Layer II, 48kHz sampling frequency, variable bit rate : STC_REG = 10111101_2
- Parameters unknown : STC_REG = 00111111_2

SYNC_ST.VERSION - Version Register

7	0
VERSION[7:0]	

Address : 6D
Type : R

Description

This register holds the audio decoder version number, "x.y", where x is represented by bits 6-4, and y by bits 3-0. Bit 7 is always 1.

The version numbers which currently exist are :

1.0: VERSION = 1001000_2
1.1: VERSION = 10010001_2
2.0: VERSION = 10100000_2
2.1: VERSION = 10100001_2

SYNC_ST - Synchronization Status

1	0
SYNC_ST[1:0]	

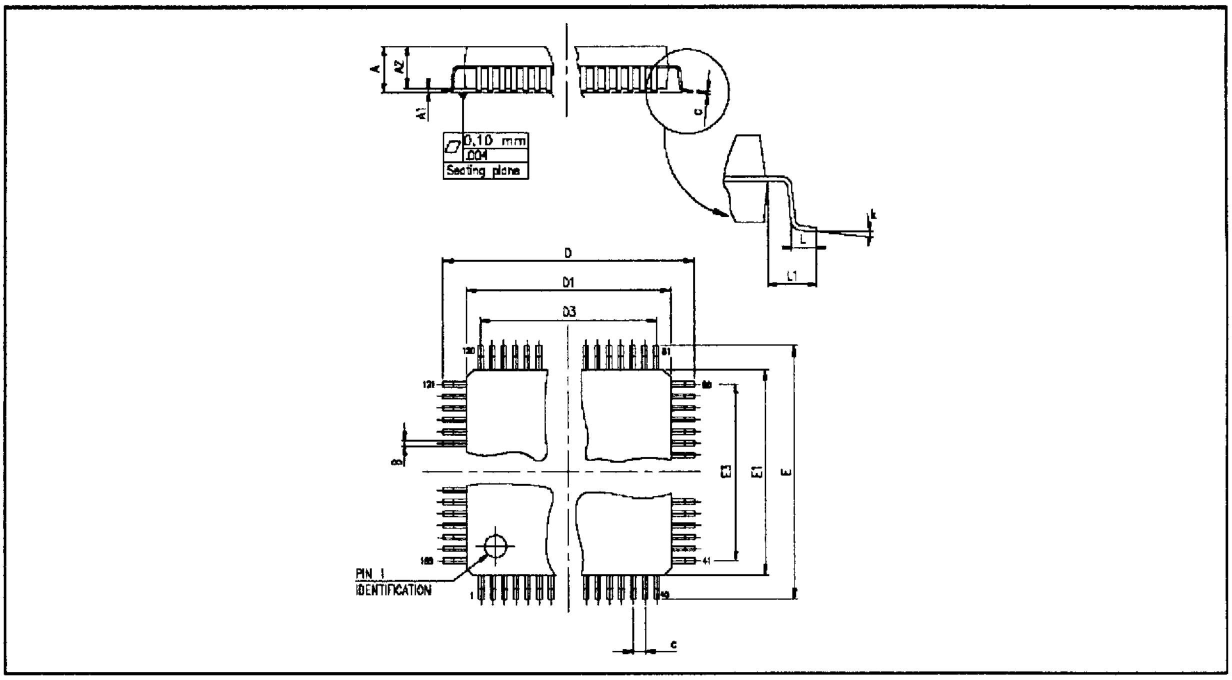
Address : 26
Type : R
Reset State : Undefined

Description

This register is loaded with the synchronization status on every synchronization cycle. This status values are :
00 : Unlocked.
01 : Attempting to recover lost synchronization.
11 : Locked.
If the status changes an interrupt 0 is generated. The status must then be read and the interrupt cleared by reading.

PACKAGE MECHANICAL DATA

160 PINS - PLASTIC QUAD FLAT PACK



PMPQF160.EPS

Dimensions	Millimeters			inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A			4.07			0.160
A1	0.25			0.010		
A2	3.17	3.42	3.67	0.125	0.135	0.145
B	0.22		0.38	0.008		0.015
C	0.13		0.23	0.005		0.009
D	30.95	31.20	31.45	1.218	1.228	1.238
D1	27.90	28.00	28.10	1.098	1.102	1.106
D3		25.35			0.998	
e		0.65			0.0256	
E	30.95	31.20	31.45	1.218	1.228	1.238
E1	27.90	28.00	28.10	1.098	1.102	1.106
E3		25.35			0.998	
L	0.65	0.80	0.95	0.026	0.031	0.037
L1		1.60			0.063	
K	0° (min.), 7° (max.)					

PQFP160.TBL

Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No licence is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of SGS-THOMSON Microelectronics.

Purchase of I^{2}C Components of SGS-THOMSON Microelectronics, conveys a license under the Philips I^{2}C Patent. Rights to use these components in a I^{2}C system, is granted provided that the system conforms to the I^{2}C Standard Specifications as defined by Philips.

SGS-THOMSON Microelectronics GROUP OF COMPANIES
Australia - Brazil - China - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco
The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.