DATA SHEET

SAA7385
Error correction and host interface IC for CD-ROM (SEQUOIA)

Preliminary specification
File under Integrated Circuits, IC01

1996 Jun 19

Philips
Semiconductors

PHILIPS

CONTENTS

1 FEATURES

1.1 General

- Single chip digital solution for an 8 × speed CD-ROM controller chip
- 10 Mbytes/s NCR53CF94 equivalent SCSI controller included
- High-speed 80C32 microcontroller with 256 × 8 scratch-pad SRAM included
- High performance CD-ROM interface logic
- 128 pin QFP package.

1.2 53CF94 SCSI controller

- Separate clock input to allow operation up to the maximum 10 Mbytes/s
- Fast synchronous SCSI-2 compatible
- 24-bit transfer counter for single transfers up to 16 Mbytes
- High-speed 16-bit DMA interface to the buffer manager DRAM
- On-chip 48 mA SCSI drivers
- Software compatible with members of the 53C90 family
- Allows for SCAM support.

1.3 80C32 high-speed microcontroller

- 33.87 MHz full system speed operation
- Three timers/event counters
- Programmable full duplex serial channel
- Eight general purpose microcontroller I/O pins
- External program ROM.

1.4 Front-end interface logic

- Full 8 × speed hardware operation
- Block decoder
- Sector sequencer
- CRC checking of Mode 1 and Mode 2, Form 1 sectors
- 212 ms watch-dog timer
- Sub-code interface with synchronization
- C-flag interface for absolute time stamp.

1.5 Buffer controller

- Ten level arbitration logic
- Utilizes low cost 70 ns DRAMs
- Page mode DRAM access for high-speed error correction and SCSI data transfer
- Data organization by 3 kbyte frames
- 256 kbyte or 1 Mbyte DRAM supported.

1.6 Hardware third-level error correction

- Third-level correction provides superior performance in unfavourable conditions
- Full hardware error correction to reduce microcontroller overhead
- Corrections are automatically written to the DRAM frame buffer.

1.7 Additional product support

- All control registers mapped into 80C32 special function memory space
- Dedicated S2B interface UART
- Input clock synthesizer
- Red book audio pass through.

2 GENERAL DESCRIPTION

The SAA7385 is a high integration ASIC that incorporates all of the digital electronics necessary to connect a CD decoder to a SCSI host. An 80C32 microcontroller and a 53CF94 SCSI controller are embedded in the ASIC. The following functions are supported:

- Input clock doubler
- Block decoder
- CRC checking of Mode 1 and Mode 2, Form 1 sectors
- Red book audio pass through to SCSI
- Buffer manager
- Third-level error correction
- Sub-code and Q-channel support
- Dedicated S2B interface UART
- Embedded 80C32 microcontroller
- Embedded 53CF94 SCSI controller.

The SAA7385 uses a 33.8688 MHz clock and is capable of accepting data at eight times (n = 8 or 1.4 Mbytes/s) the normal CD-ROM data rate.

Third level error correction hardware is included to improve the correction efficiency of the system. The buffer manager hardware utilizes a ten-level arbitration unit and can stop the clock to the microcontroller to emulate a wait condition when necessary.

The SAA7385 comprises five major functional blocks:

- The 80C32 microcontroller is an industry standard core
- The 53CF94 is an industry standard core
- The front-end block connects to the external CD-60 based decoder and fully processes the incoming data stream to provide bytes of data that are stored in the external buffer
- The buffer manager block provides the address generation and timing control for the external DRAM buffer
- The ECC block performs the error correction functions in hardware on the data in the DRAM buffer.

Supply of this Compact Disc IC does not convey an implied license under any patent right to use this IC in any Compact Disc application.

3 QUICK REFERENCE DATA

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
V_{DD}	digital supply voltage	4.5	5.0	5.5	V
T_{amb}	operating ambient temperature	0	–	70	°C
T_{stg}	storage temperature	–55	–	+150	°C

4 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
SAA7385GP	SQFP128	plastic quad flat package; 128 leads (lead length 1.6 mm); body 14 × 20 × 2.8 mm	SOT387-2

5 BLOCK DIAGRAM

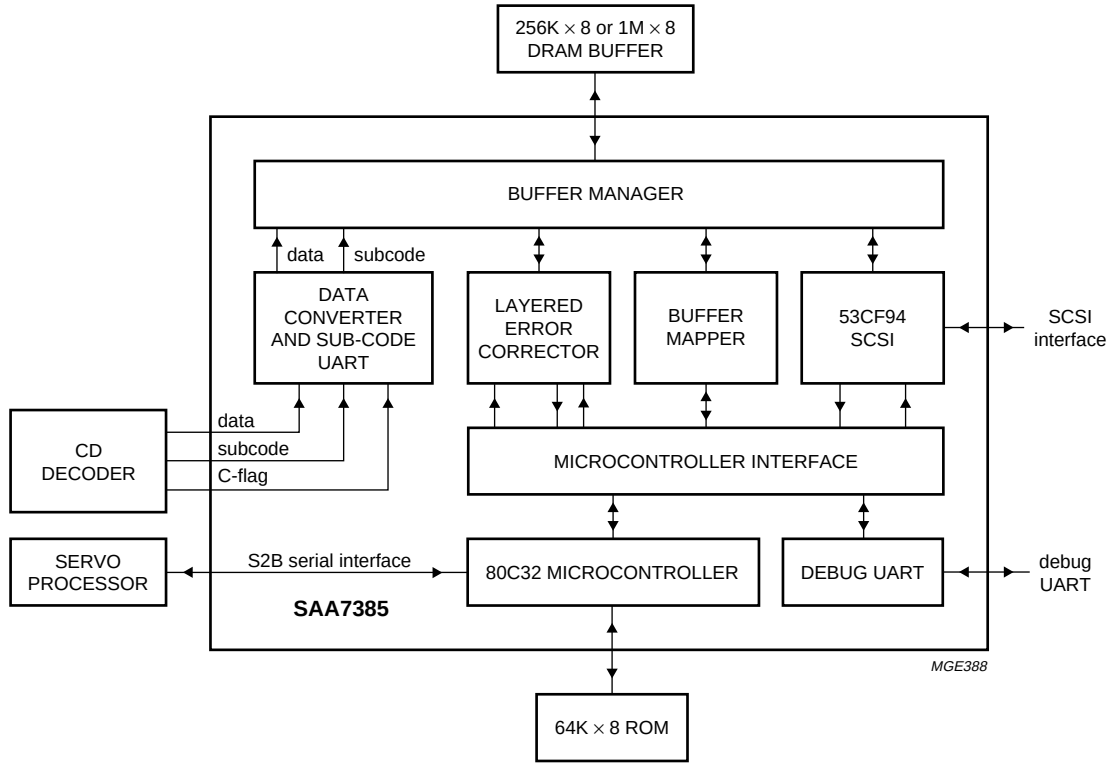


Fig.1 Block diagram (simplified).

6 PINNING

All input, output and bidirectional signals are TTL level unless otherwise stated (Pull-Down = PD25 = 25 μA; Pull-Up = PU25 = 25 μA, PU400 = 400 μA; Slew = S2 = 2 mA, S4 = 4 mA; CMOS slew = CMOS S2 = CMOS 2 = 2 mA; SCSI pad = SCSI = 48 mA).

SYMBOL	PIN	I/O	PAD	DESCRIPTION
DA2	1	O	S4	DRAM address bus; bit DA2
DA3	2	O	S4	DRAM address bus; bit DA3
DA4	3	O	S4	DRAM address bus; bit DA4
V_{SS1}	4	–	–	ground 1
DA5	5	O	S4	DRAM address bus; bit DA5
DA6	6	O	S4	DRAM address bus; bit DA6
DA7	7	O	S4	DRAM address bus; bit DA7
DA8	8	O	S4	DRAM address bus; bit DA8
DA9	9	O	S4	DRAM address bus; bit DA9
V_{DD1}	10	–	–	power supply 1

SYMBOL	PIN	I/O	PAD	DESCRIPTION
$\overline{RAS}$	11	O	S4	DRAM row address section; active LOW
$\overline{CAS}$	12	O	S4	DRAM column address selection; active LOW
$\overline{DWR}$	13	O	S4	DRAM write; active LOW
$\overline{DOE}$	14	O	S4	DRAM output enable; active LOW
V_{SS2}	15	–	–	ground 2
DD0	16	I/O	4 mA, Schmitt, PD25	DRAM data bus; bit DD0
DD1	17	I/O	4 mA, Schmitt, PD25	DRAM data bus; bit DD1
DD2	18	I/O	4 mA, Schmitt, PD25	DRAM data bus; bit DD2
DD3	19	I/O	4 mA, Schmitt, PD25	DRAM data bus; bit DD3
V_{DD2}	20	–	–	power supply 2
DD4	21	I/O	4 mA, Schmitt, PD25	DRAM data bus; bit DD4
DD5	22	I/O	4 mA, Schmitt, PD25	DRAM data bus; bit DD5
DD6	23	I/O	4 mA, Schmitt, PD25	DRAM data bus; bit DD6
DD7	24	I/O	4 mA, Schmitt, PD25	DRAM data bus; bit DD7
V_{SS3}	25	–	–	ground 3
$\overline{LED}$	26	O	24 mA, CMOS test	panel LED; active LOW; WTGCTL(4)
$\overline{TRAYSW}$	27	I	Schmitt, PU25	active LOW when tray is in
$\overline{EJECT}$	28	I	Schmitt, PU25	opens tray; active LOW
LQDATA	29	O	2 mA	serial data to DAC
LWCLK	30	O	2 mA	word strobe to DAC
V_{SS4}	31	–	–	ground 4
SCLK	32	O	2 mA	data serial clock
V_{SS5}	33	–	–	ground 5
SYSRES	34	O	2 mA, PU25	system reset; OR of $\overline{POR}$, $\overline{SCSIRST}$ and watch-dog timer
CFLAG	35	I	Schmitt, PU400	C1 and C2 status
$\overline{CPR}$	36	O	2 mA	S2B interface ready to accept data; active LOW
$\overline{SPR}$	37	I	Schmitt	S2B interface ready to send data; active LOW
$\overline{SKIPFWD}$	38	I	Schmitt, PU25	skip forwards; active LOW; RDSW(3)
$\overline{SKIPBACK}$	39	I	Schmitt, PU25	skip backwards; active LOW; RDSW(2)
SCSICLK	40	I	standard	SCSI interface clock
V_{DD3}	41	–	–	power supply 3
AD0	42	I/O	S4, Schmitt	microcontroller multiplexed data bus; bit AD0
AD1	43	I/O	S4, Schmitt	microcontroller multiplexed data bus; bit AD1
AD2	44	I/O	S4, Schmitt	microcontroller multiplexed data bus; bit AD2
AD3	45	I/O	S4, Schmitt	microcontroller multiplexed data bus; bit AD3
AD4	46	I/O	S4, Schmitt	microcontroller multiplexed data bus; bit AD4
AD5	47	I/O	S4, Schmitt	microcontroller multiplexed data bus; bit AD5
AD6	48	I/O	S4, Schmitt	microcontroller multiplexed data bus; bit AD6
AD7	49	I/O	S4, Schmitt	microcontroller multiplexed data bus; bit AD7
V_{SS6}	50	–	–	ground 6
LA0	51	O	CMOS S2, PU25	EPROM latched lower address; bit LA0

SYMBOL	PIN	I/O	PAD	DESCRIPTION
LA1	52	O	CMOS S2, PU25	EPROM latched lower address; bit LA1
LA2	53	O	CMOS S2, PU25	EPROM latched lower address; bit LA2
LA3	54	O	CMOS S2, PU25	EPROM latched lower address; bit LA3
V_{DD4}	55	–	–	power supply 4
LA4	56	O	CMOS S2, PU25	EPROM latched lower address; bit LA4
LA5	57	O	CMOS S2, PU25	EPROM latched lower address; bit LA5
LA6	58	O	CMOS S2, PU25	EPROM latched lower address; bit LA6
LA7	59	O	CMOS S2, PU25	EPROM latched lower address; bit LA7
V_{SS7}	60	–	–	ground 7
A8	61	O	CMOS S2, PU25	EPROM upper address; bit A8
A9	62	O	CMOS S2, PU25	EPROM upper address; bit A9
A10	63	O	CMOS S2, PU25	EPROM upper address; bit A10
A11	64	O	CMOS S2, PU25	EPROM upper address; bit A11
A12	65	O	CMOS S2, PU25	EPROM upper address; bit A12
A13	66	O	CMOS S2, PU25	EPROM upper address; bit A13
A14	67	O	CMOS S2, PU25	EPROM upper address; bit A14
A15	68	O	CMOS S2, PU25	EPROM upper address; bit A15
$\overline{PSEN}$	69	O	CMOS 2, PU25	program store enable; active LOW
V_{SS8}	70	–	–	ground 8
$\overline{IO}$	71	I/O	SCSI	SCSI phase signal, active LOW
$\overline{REQ}$	72	I/O	SCSI	SCSI request, active LOW
$\overline{CD}$	73	I/O	SCSI	SCSI phase signal, active LOW
$\overline{SEL}$	74	I/O	SCSI	SCSI select, active LOW
V_{SS9}	75	–	–	ground 9
$\overline{MSG}$	76	I/O	SCSI	SCSI phase signal, active LOW
$\overline{ACK}$	77	I/O	SCSI	SCSI acknowledge, active LOW
$\overline{BSY}$	78	I/O	SCSI	SCSI busy, active LOW
V_{SS10}	79	–	–	ground 10
$\overline{ATN}$	80	I/O	SCSI	output in initiator mode; input in target mode, active LOW
V_{DD5}	81	–	–	power supply 5
$\overline{SDP}$	82	I/O	SCSI	SCSI parity, active LOW
SD7	83	I/O	SCSI	SCSI data bus; bit SD7
SD6	84	I/O	SCSI	SCSI data bus; bit SD6
SD5	85	I/O	SCSI	SCSI data bus; bit SD5
V_{SS11}	86	–	–	ground 11
SD4	87	I/O	SCSI	SCSI data bus; bit SD4
SD3	88	I/O	SCSI	SCSI data bus; bit SD3
SD2	89	I/O	SCSI	SCSI data bus; bit SD2
SD1	90	I/O	SCSI	SCSI data bus; bit SD1
SD0	91	I/O	SCSI	SCSI data bus; bit SD0
V_{SS12}	92	–	–	ground 12

SYMBOL	PIN	I/O	PAD	DESCRIPTION
RXS2B	93	I	Schmitt, PU25	S2B interface receive
TXS2B	94	O	4 mA	S2B interface transmit
$\overline{\text{TRAYIN}}$	95	I/O	4 mA, PD25	tray extend control; active LOW (general purpose signal)
$\overline{\text{TRAYOUT}}$	96	I/O	4 mA, PD25	tray retract control; active LOW (general purpose signal)
$\overline{\text{SCSIRST}}$	97	I	Schmitt	SCSI reset, active LOW; also causes a system reset
$\overline{\text{POR}}$	98	I	CMOS	power-on reset; active LOW
V_{DD6}	99	–	–	power supply 6
UC_PORT1.7	100	I/O	CMOS 2, PU25	drive speed select; microcontroller port 1.7
RAB_MUSB	101	I/O	CMOS 2, PU25	RD/WR, acknowledge; microcontroller port 1.2
NRST_SEQ	102	I/O	CMOS 2, PU25	reset to engine; microcontroller port 1.5
UC_PORT1.4	103	I/O	CMOS 2, PU25	general purpose microcontroller I/O port; port 1.4
UC_PORT1.3	104	I/O	CMOS 2, PU25	general purpose microcontroller I/O port; port 1.3
UC_PORT1.1	105	I/O	CMOS 2, PU25	general purpose microcontroller I/O port; port 1.1
$\overline{\text{HOMESW}}$	106	I/O	2 mA, PU25	actuator sled home; active LOW; microcontroller port 1.0
$\overline{\text{PLAY}}$	107	I	Schmitt	laser on and focused status; active LOW; RDSW(4)
UC_PORT1.6	108	I/O	CMOS 2, PU25	general purpose microcontroller I/O port; port 1.6
V_{SS13}	109	–	–	ground 13
GPI1	110	I	Schmitt, PU25	general purpose input; microcontroller port 3.4
GPI2	111	I	Schmitt, PU25	general purpose input; microcontroller port 3.5
$\overline{\text{KILL}}$	112	I	Schmitt, PU25	shut off audio; active LOW
TXICE	113	O	4 mA	debug UART output; from 80C32 serial port
RXICE	114	I	Schmitt, PU25	debug UART input; to 80C32 serial port
RXSUB	115	I	Schmitt, PU25	sub-code input
V_{DD7}	116	–	–	power supply 7
OSCIN	117	I	standard	master input clock; 34 or 16 MHz
V_{SS14}	118	–	–	ground 14
CLAB	119	I	Schmitt	clock
V_{SS15}	120	–	–	ground 15
DAAB	121	I	Schmitt	data
WSAB	122	I	Schmitt	word strobe
EFAB	123	I	Schmitt	error flag
CLK34	124	O	2 mA	34 MHz output clock
TEST	125	I	Schmitt, PD25	test pin; must be ground
V_{SS16}	126	–	–	ground 16
DA0	127	O	S4	DRAM address bus; bit DA0
DA1	128	O	S4	DRAM address bus; bit DA1

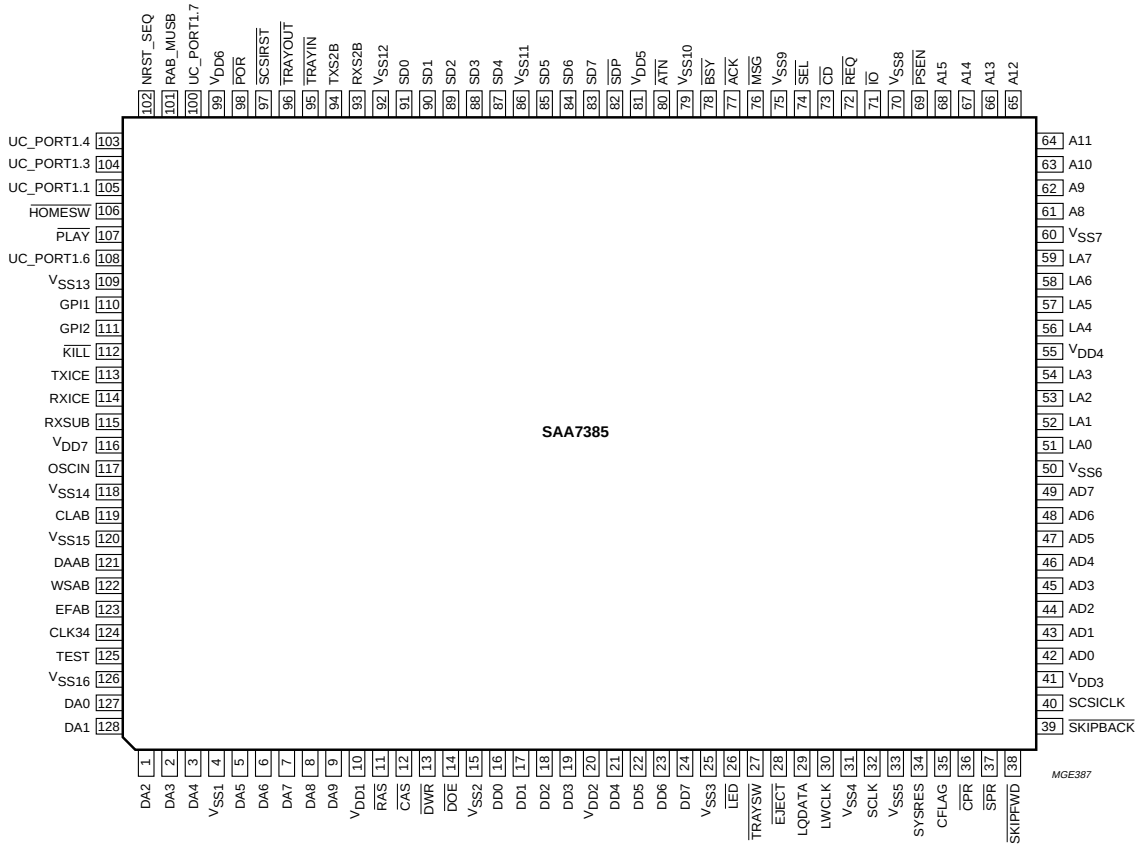


Fig.2 Pin configuration.

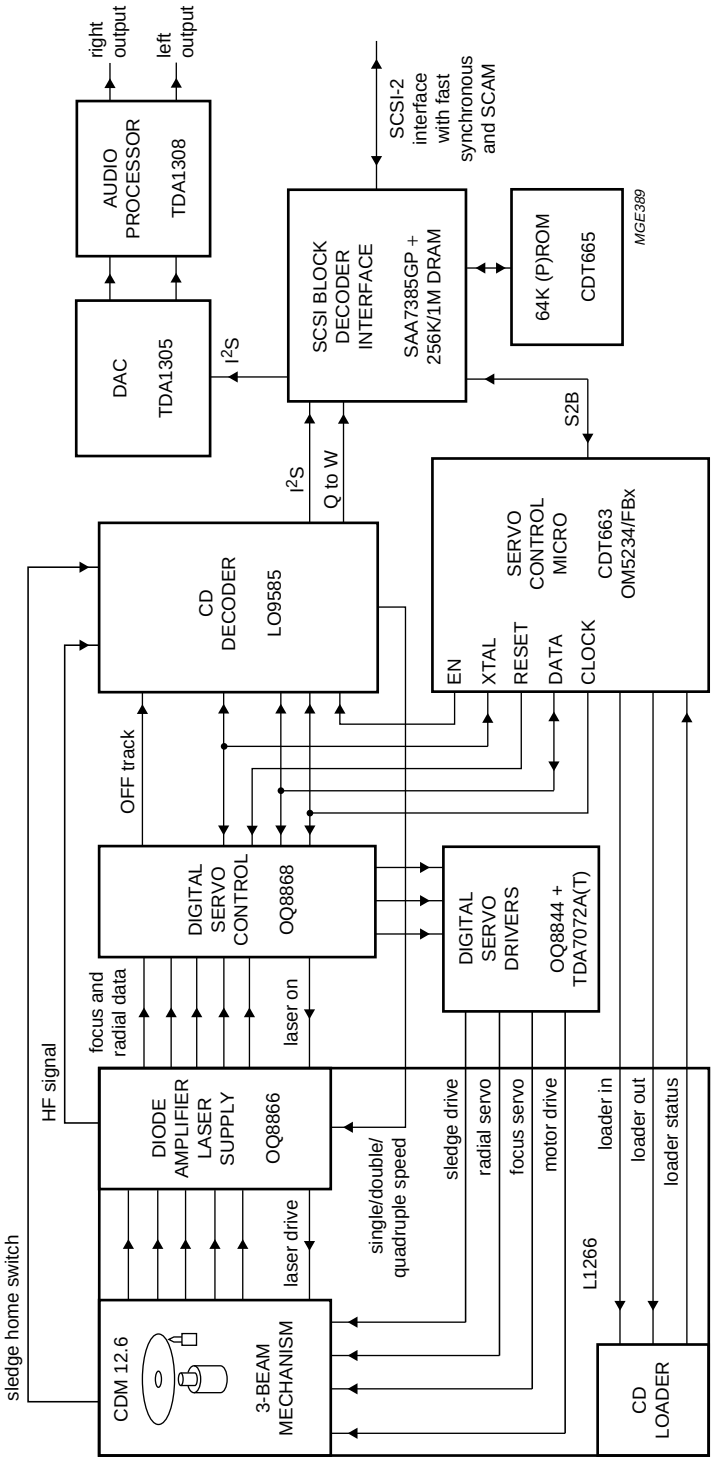


Fig.3 Example of CD-ROM system with SCSI-2 interface.

7 FUNCTIONAL DESCRIPTION

7.1 80C32 microcontroller

The standard specification for details of the operation for this part may be found in any data sheet covering the 80C32 microcontroller. The one deviation from a normal 80C32 is the addition of all of the control registers for the special function register map for the 80C32. All of the SAA7385 control registers, including the 53CF94 control registers appear within this space.

7.2 53CF94 fast SCSI controller

The details of operation of this block may be found in the *"53CF94 data manual"*. Two deviations from the operation of a normal 53CF94 have been made. The first is that the part supports single-ended SCSI bus operation only. The second deviation is the additional feature of mapping the control registers into the 80C32 special function register map as previously mentioned.

7.3 Input clock doubler

To facilitate compatibility of the SAA7385 with the maximum number of CD decoders, a clock doubler has been included. This clock doubler may take a 16.9344 MHz clock and double this when requested to do so by the microcontroller. Logic has been included to remove the possibility of a 'runt' clock pulse when the doubler is engaged. Once engaged, the only way to disengage it is via a reset condition.

7.4 Front-end

The front-end is comprised of many sub-sections.

7.4.1 BLOCK DECODER

The block decoder first reverses the bits of each received byte and then runs them through a linear feedback shift register to be de-scrambled. The polynomial used to de-scramble the serial data is as follows: $X^{15} + X + 1$

It also detects and tests the synchronization field and will start the data clock when commanded. The de-scrambled header is assembled into four registers (MODE, MINS, SECS and FRMS) with header ready and header error status (see HDRRDY and HDRERR in RDDSTAT). The data clock does not have to be enabled to receive valid headers.

Also included in this section is the logic required to decide when to start collecting data and sub-code information taken from the synchronization signal.

7.4.2 SECTOR SEQUENCER

The sector sequencer de-serializes the data and error flags from the block decoder and determines when to:

- Write data to the buffer
- Write flags to the buffer
- Test the header to determine the Mode
- Test the sub-header to determine the Form
- Test the CRC
- End the sector and write the status byte to the buffer.

Included in the sector sequencer is the CRC generator which checks each Yellow Book or Green Book sector as it is shifted into the SAA7385 in accordance with the following polynomial:

$X^{32} + X^{31} + X^{16} + X^{15} + X^4 + X^3 + X + 1$

The status of each sector is saved and written to the buffer at the end of the sector.

7.4.3 SUB-CODE RECEIVE AND Q-CHANNEL EXTRACTOR

A UART which samples asynchronous bits on a 24 clocks per bit basis is included. This is required because Philips decoders output the sub-code data at nominally 24 clocks per bit, but not synchronized to the data. Also included is a sub-code synchronization detector which senses the beginning of each new sector of sub-code information. The serial sub-code information is assembled into bytes in the following order:

Data bits 7 to 0 = 0, Q, R, S, T, U, V and W.

As each byte is assembled, it is sent to the buffer manager to be written to the DRAM buffer. At the same time, the Q-channel bits are assembled into bytes and sent to the buffer. All Q-channel bytes except CRC are sorted in registers for use by the microcontroller. The track, mode, minutes, seconds and frames bytes (RDTK, RDMD, RDMN, RDSC and RDFM) are also stored in registers for use by the microcontroller. The Q-channel CRC (last two bytes) is checked just before the end of the sub-code sector. If the CRC check fails, BADQ in RDDSTAT is available to the microcontroller and is written into the buffer at the end of the sector. When the five Q-channel registers have been updated, QFRMRDY in RDDSTAT is set. The five Q-channel registers are valid while QFRMRDY is set. In the audio mode, HDRRDY in RDDSTAT generates this interrupt, but the QFRMRDY bit will still be available as status to the microcontroller.

7.4.4 C-FLAG RECEIVER

The C-flag bits, or corrector flags, are also 24 data clocks long and reception of these bits is achieved using the same method as for the sub-code; in this event, the C-flag data is synchronized to the data. The difference is that only one bit is used; F1, the absolute time synchronization information. When in audio mode and ENABRED in FECTL is set, receipt of F1 set will start the internal data clock after the next rising edge of word strobe (WSAB) which is the left channel sample when the CD decoder is programmed for EIAJ audio format. When in audio mode, the Q-channel information provides the MSF address and the F1 flag provides the start of frame information; together these provide an absolute byte address on the disc.

7.4.5 S2B UART

This UART is provided for remote debugging of the firmware. It is hard-wired for one start-bit, eight data bits, a parity bit and one stop bit. Parity testing can be programmed to be either odd parity or even parity. Parity error and over-run status are provided via PE and OVRRUN in S2BSTAT. Selectable baud rates of 31.25, 62.5 and 187.5 kbaud are available via ICESEL1 and ICESEL0 in BRGSEL.

7.4.6 WATCH-DOG TIMER

A pair of counters are included which output a 967 µs reset pulse to the entire chip and the SYSRES pin if the timer is not reset during the 212 ms time-out period.
The watch-dog timer is reset by setting RWMD in FECTL HIGH then LOW. If RWMD is left HIGH, the watch-dog function is disabled.

7.4.7 GLUE LOGIC (GLIC)

The final block of logic in the front-end consists of: a programmable, linear pulse-width modulator for spindle-motor control; an address de-multiplexer for the address/data bus of the microcontroller; plus audio multiplexing and muting circuitry for full control of Red Book audio data to an external Digital-to-Analog Converter (DAC).

7.4.8 BUFFER MANAGER

The buffer manager provides the arbitration for the different processes that wish to access the DRAM buffer. These processes include the front-end, microcontroller requests, ECC accesses, SCSI interface requests and DRAM refreshing. The DRAM control logic will start an access on the next rising edge of the clock after a request is received. If two or more requests are pending then the priority is as follows:

1. Front-end (highest priority)
2. Microcontroller requests
3. SCSI interface requests
4. ECC requests (lowest priority).

A refresh cycle is required every 15.6 µs and will be granted priority for one access. A burst access by ECC or SCSI will only be interrupted by a higher priority access request.

In addition to the priority logic, logic is required for the front-end sources of data. The priority is: frame data (highest), flag data, sub-code data, Q-channel data and finally status byte. All front-end sources are granted priority over the SCSI logic, ECC, refresh and data will be written into the frame store during the next cycle. However, the microcontroller has priority over the lower three front-end sources and will be granted an access after front-end frame data or flag data is written to memory.

The required timing (see Figs 4 to 11) operate with the industry standard 70 ns DRAMs. The interface is designed to operate with one or two DRAMs using: 256 kbit $\times$ 4 or 1 Mbit $\times$ 4 devices. If a single DRAM is connected, all access cycles require a page mode cycle to load both the high and the low nibble of data. With a byte-wide memory attached, a single byte cycle takes five clock cycles of 29.5 ns each, totalling 147.5 ns. In nibble mode, a single byte cycle takes 236 ns.

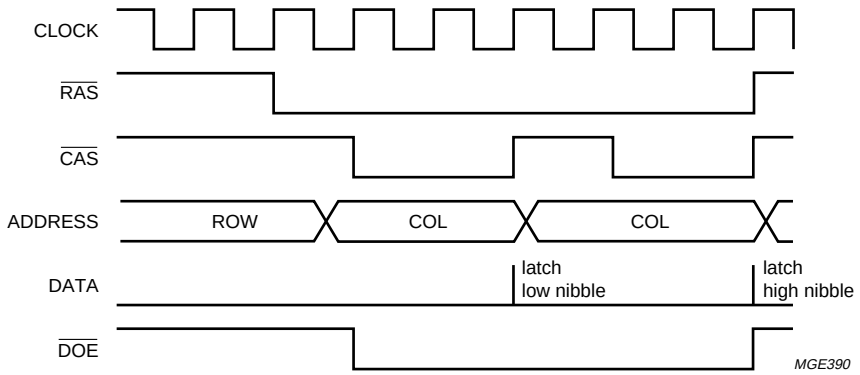


Fig.4 Nibble access read cycle.

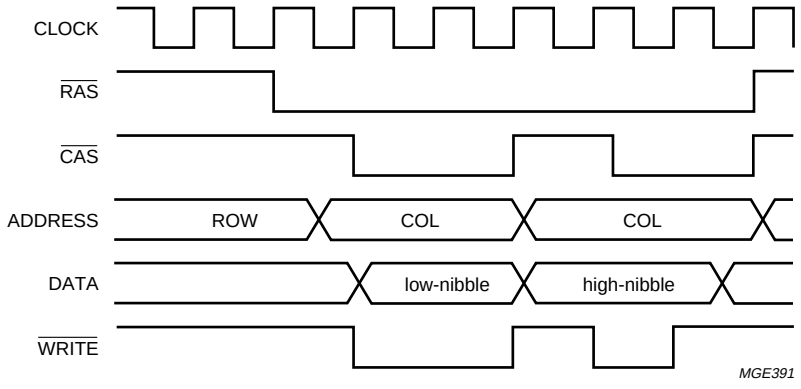


Fig.5 Nibble access write cycle.

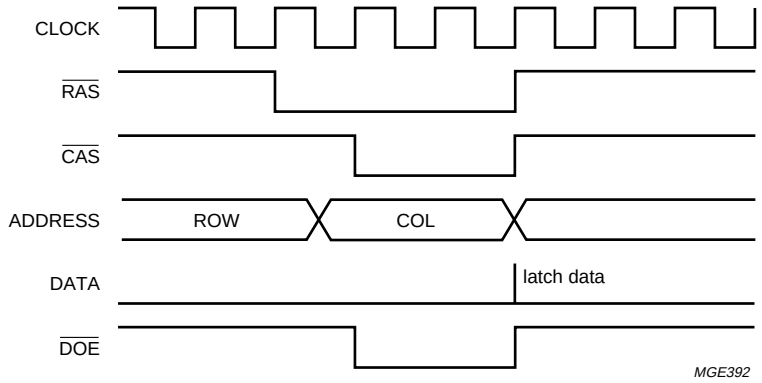


Fig.6 Byte mode single access read cycle.

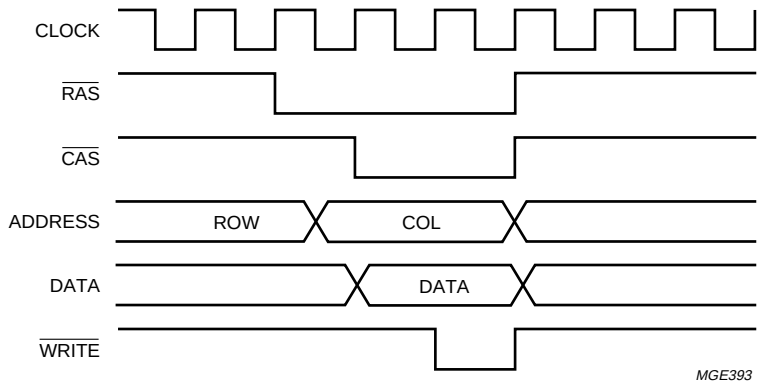


Fig.7 Byte mode single access write cycle.

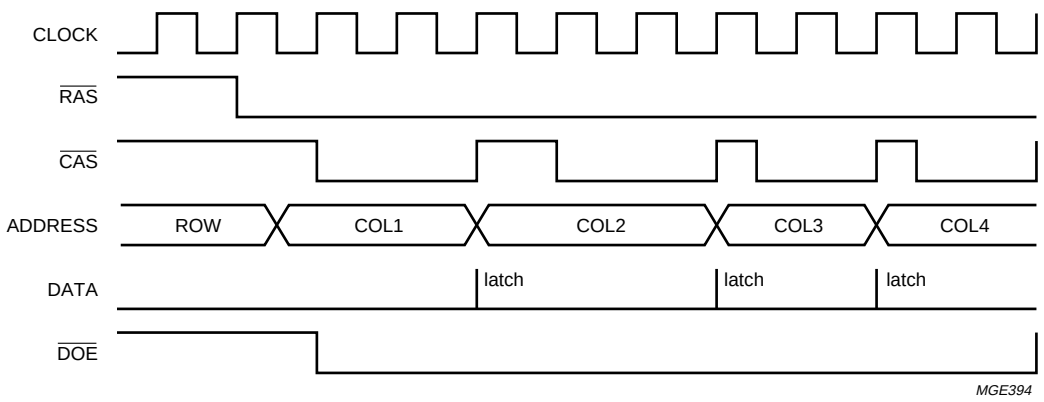


Fig.8 ECC burst access read cycle.

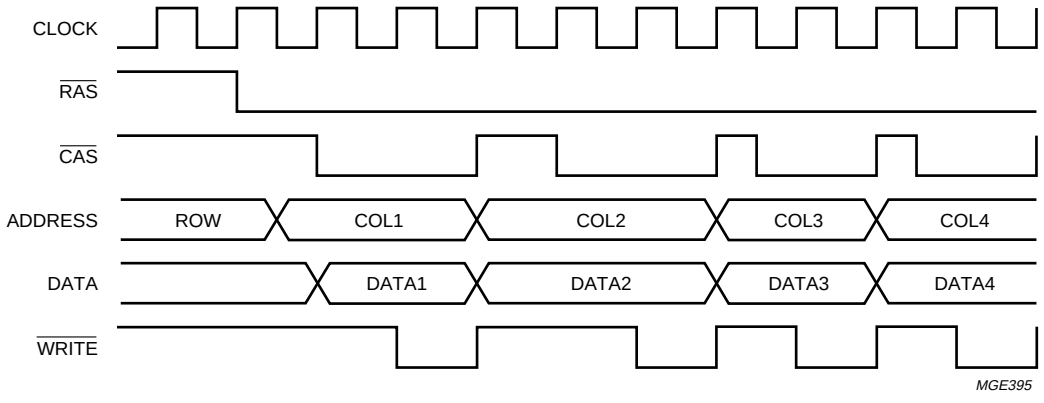


Fig.9 ECC burst access write cycle.

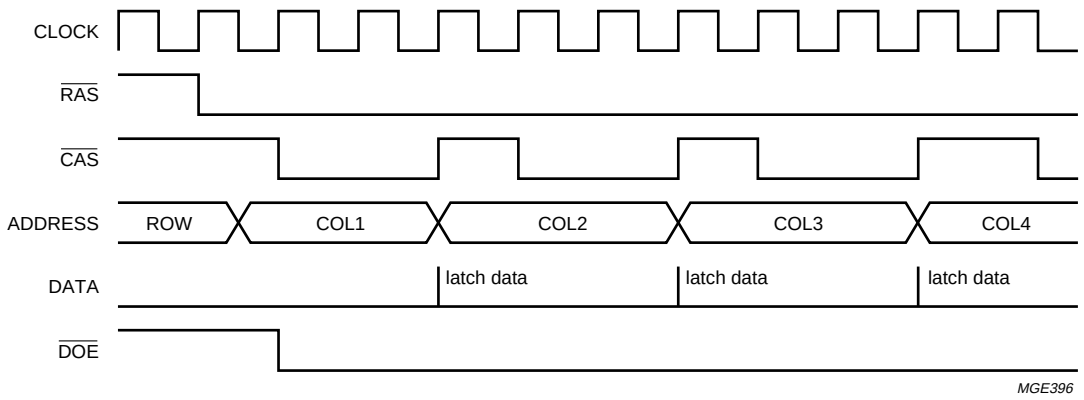


Fig.10 SCSI standard burst access read cycle.

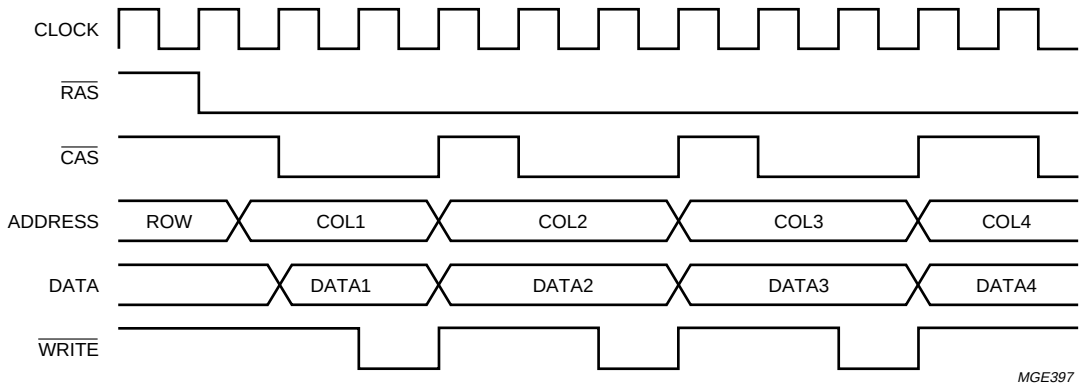


Fig.11 SCSI standard burst access write cycle.

8 MICROCONTROLLER INTERFACE

8.1 Microcontroller interface status register

Table 1 NUM_COR register: 0xF08E

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
NUM_COR	R	NUM_COR7 to NUM_COR0							

Register 0xF08E indicates the number of corrections performed during the most recently executed CORRECT_P_SYNDROMES or CORRECT_Q_SYNDROMES command. Note that NUM_COR is only valid after completion of the CORRECT_P_SYNDROMES or CORRECT_Q_SYNDROMES command, and becomes invalid upon execution of any other command.

Table 2 ECC_STATUS register: 0xF086

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
ECC_STATUS	R	–	–	–	FLG_EQ0	CRC_EQ0	PS_EQ0	QS_EQ0	ECC_ACT

Register 0xF086 provides status information on the current or last ECC command.

Table 3 ECC_STATUS definitions

MNEMONIC	DESCRIPTION
ECC_ACT	asserted while a command other than ASSERT_ABORT or RELEASE_ABORT remains active
QS_EQ0	asserted when all Q syndromes are zero
PS_EQ0	asserted when all P syndromes are zero
CRC_EQ0	asserted when the CRC remainder calculated by the CRC_CALCULATE command is all zeros
FLG_EQ0	asserted when all flag bytes in ECC RAM are zero

8.2 Microcontroller interface command register

Table 4 ECCCTL register: 0xF085

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
ECCCTL	R/W	–	–	–	–	ECC_COMMAND3 to ECC_COMMAND0			

The ECC_COMMAND definitions are explained in Table 5.

Table 5 Definitions of ECC_COMMAND3 to ECC_COMMAND0

EEC_COMMAND	DESCRIPTION
0000	ASSERT_ABORT
0001	RELEASE_ABORT
0010	CALCULATE_SYNDROMES (not Mode 2, Form 1)
0011	CALCULATE_SYNDROMES (Mode 2, Form 1)
0100	CRC_RECALCULATE (not Mode 2, Form 1)
0101	CRC_RECALCULATE (Mode 2, Form 1)
0110	COPY_RESULTS (not Mode 2, Form 1)
0111	COPY_RESULTS (Mode 2, Form 1)
1000	CORRECT_P_SYNDROMES
1001	CORRECT_Q_SYNDROMES
1100	TEST_ECC_ROM
1101	TEST_ECC_RAM_READ
1110	TEST_ECC_RAM_WRITE

Table 6 Command descriptions

COMMAND	DESCRIPTION
ASSERT_ABORT	Terminates any currently active operation and re-initializes the ECC logic. Remains in reset state until occurrence of the RELEASE_ABORT command. At power-on reset, the ECC is in the ASSERT_ABORT state. All microcontroller status bits are reset when the ECC is in the ASSERT_ABORT state.
RELEASE_ABORT	Terminates the ASSERT_ABORT command and enables activation of other commands.
CRC_RECALCULATE	Calculate CRC remainder buffer data, storing result in ECC RAM and updating microcontroller status bit CRC_EQ0. Mode 2, Form 1 uses address 16 : 2075, or 0 : 2067; note 1.
CALCULATE_SYNDROMES	Prepares buffer for correction, calculates P and Q syndromes, and copies error flags and CRC remainder from buffer to ECC RAM. The microcontroller status bits PS_EQ0, QS_EQ0 and FLAGS_EQ0 are updated at the end of this operation. 1. Copy header from buffer to ECC RAM (Mode 2, Form 1 only) 2. Write to the buffer. Not Mode 2, Form 1: Address 0 → 0x00; Address 1 : 10 → 0xFF; Address 11 → 0x00; Address 2068 : 2075 → 0x00 Mode 2, Form 1: Address 0 → 0x00; Add 1 : 10 → 0xFF; Add 11 : 15 → 0x00 3. Read header and frame data from buffer to calculate P and Q syndromes psyn[0 : 85].s1, psyn[0 : 85].s0, qsyn[0 : 51].s1 and qsyn[0 : 51].s0, storing results in ECC RAM; see Table 76 4. Copy error flags from buffer to ECC RAM 5. Copy CRC remainder from buffer to ECC RAM 6. Update microcontroller status bits PS_EQ0, QS_EQ0 and FLAGS_EQ0.

COMMAND	DESCRIPTION
COPY_RESULTS	Copies current ECC RAM contents to the buffer memory: 1. Copy header flags from ECC RAM to buffer (Mode 2, Form 1 only) 2. Copy error Flags from ECC RAM to buffer 3. Copy CRC remainder from ECC RAM to buffer 4. Copy P syndromes from ECC RAM to buffer 5. Copy Q syndromes from ECC RAM to buffer.
CORRECT_P_SYNDROMES	Scan all P syndromes and perform P-syndrome calculation. The microcontroller status bits PS_EQ0, QS_EQ0 and FLAGS_EQ0 are updated at the end of this operation.
CORRECT_Q_SYNDROMES	Scan all Q syndromes and perform Q-syndrome calculation. The microcontroller status bits PS_EQ0, QS_EQ0 and FLAGS_EQ0 are updated at the end of this operation.
TEST_ECC_ROM	Read each exponent and log in the alpha ROM to the NUM_COR register. This command may only be terminated by the ASSERT_ABORT command.
TEST_ECC_RAM_READ	Read ECC RAM addresses 0 : 591 and copy to buffer addresses 0 : 591.
TEST_ECC_RAM_WRITE	Read buffer addresses 0 : 591 and copy to ECC RAM addresses 0 : 591.

Note

1. 16 : 2075 and 0 : 2067 are address frame offsets. The frame buffer organization is shown in Table 75.

8.3 Microcontroller interrupts

An interrupt pulse is generated upon completion of any of the following commands:

- CALCULATE_SYNDROMES (not Mode 2, Form 1)
- CALCULATE_SYNDROMES (Mode 2, Form 1)
- CRC_RECALCULATE (not Mode 2, Form 1)
- CRC_RECALCULATE (Mode 2, Form 1)
- COPY_RESULTS (not Mode 2, Form 1)
- COPY_RESULTS (Mode 2, Form 1)
- CORRECT_P_SYNDROMES
- CORRECT_Q_SYNDROMES
- TEST_ECC_ROM
- TEST_ECC_RAM_READ
- TEST_ECC_RAM_WRITE.

If a command is aborted by the ASSERT_ABORT command, a spurious interrupt may be generated within five clock cycles of the ASSERT_ABORT command.

Table 7 Command execution times

COMMAND	CYCLES	TIME (μs) at 33 MHz	MEMORY ACCESSES
CALCULATE_SYNDROMES (not Mode 2, Form 1)	5604	186.8	2658
CALCULATE_SYNDROMES (Mode 2, Form 1)	5600	186.7	2654
CRC_RECALCULATE (not Mode 2, Form 1)	4136	137.9	2068
CRC_RECALCULATE (Mode 2, Form 1)	4120	137.3	2060
COPY_RESULTS (not Mode 2, Form 1)	1148	38.3	574
COPY_RESULTS (Mode 2, Form 1)	1156	38.5	578
CORRECT_P_SYNDROMES (maximum addition per correction)	1466 157	48.9 5.2	0 2
CORRECT_Q_SYNDROMES (maximum addition per correction)	888 167	29.6 5.6	0 2
TEST_ECC_RAM_READ	1184	39.5	592
TEST_ECC_RAM_WRITE	1184	39.5	592

All times indicated reflect two clock cycles per memory access for all accesses other than P and Q corrections. P and Q corrections reflect seven clock cycles per memory access. Execution times will be extended due to refresh timing, other buffer traffic, and configuration of nibble-wide memory.

8.3.1 INTERRUPT REGISTER DEFINITIONS

Two registers are used to control the operation of the interrupt logic. The register INTRMSK allows each interrupt to be enabled or disabled. INTRMSK and INTRFLG are cleared on reset to initially disable and clear all interrupts; the output latch controlling the INT line is set on a reset; this must be cleared by writing 0x00 to INTRFLG. To enable an interrupt, the bit that corresponds to the interrupt in INTRFLG must be set. The INTRFLG register shows the status of the interrupts. If any bit is HIGH then an interrupt has occurred since the last time the bit was cleared. Writing a zero to any bit location in INTRFLG will clear the corresponding interrupt. If a masked interrupt occurs, the microcontroller can still detect the occurrence because the event is still posted in INTRFLG.

Table 8 Interrupt mask register: 0xF0FB

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
INTRMSK	R/W	MASK7	MASK6	MASK5	MASK4	MASK3	MASK2	MASK1	MASK0

Each bit in register 0xF0FB corresponds to the interrupt at the same bit location in register 0xF0FC. To enable an interrupt, the bit in this register must be set HIGH.

Table 9 Interrupt flag register: 0xF0FC

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
INTRFLG	R/W	–	FETXINT	FERXINT	ECC_COR	FE_HDR	FE2352	STR_LST	FRM_STR

If any bit is set in this register (Table 9) then an interrupt may be sent to the microcontroller. Table 10 shows when the interrupts are asserted; assuming the corresponding mask bit is set.

Table 10 INTRFLG field descriptions

FIELD	DESCRIPTION
FRM_STR	set one when one complete frame is stored
STR_LST	set at the start of the last frame
FE_2352	set if the front-end data exceeds 2352 bytes
FE_HDR	front-end interrupt for header (or Q channel) ready
ECC_COR	ECC interrupt for correction complete
RFERXINT	front-end interrupt for receive ready
FETXINT	front-end interrupt for transmit ready

8.4 Microcontroller RAM organization

MICFRM# is used to determine the frame address for the microcontroller access through the frame window 0x8000 to 0x8FFF. To obtain the actual byte location within the buffer RAM, the lower 12 bits of the microcontroller address form the relative offset and hence the absolute address is found.

Note that the microcontroller has the option of addressing memory in a linear fashion using the 32 kbyte address space between 0x000 and 0x7FFF. If this 32 kbyte page is used, the PAGEREG must be programmed with the required page address. PAGEREG is used to select the required page when the microcontroller makes a linear access to the buffer memory using the address space 0x7000 to 0x7FFF. The actual address is the fifteen LSBs from the microcontroller plus 32768 times the value in PAGEREG.

Table 11 Microcontroller frame number address registers: 0xF0F6 and 0xF0F7

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
MICFRM#	R/W	NUM7	NUM6	NUM5	NUM4	NUM3	NUM2	NUM1	NUM0
MICFRM#	R/W	–	–	–	–	–	–	–	NUM8

Registers 0xF0F6 and 0xF0F7 provide the frame number address for the microcontroller access to memory. The counter associated with these registers is loaded after the most significant byte is written; the least significant byte must be written first to ensure that the counter is loaded correctly. If a DRAM access is in progress that uses the address from the counter, the update will be delayed until the access is complete.

Table 12 Microcontroller address page register: 0xF0FF

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
PAGEREG	R/W	–	–	PAGE_EN(1)	MA_19	MA_18	MA_17	MA_16	MA_15

Note

1. PAGE_EN is disconnected.

Register 0xF0FF is used by the buffer manager for the upper address lines when the microcontroller addresses non-frame memory. These registers overlap frame memory, so register 0xF0FF must be programmed with an address in the top part of the memory if no overlap is required. The microcontroller page address line is selected from this register. The outputs are used directly to control DRAM access cycles, and will affect any current DRAM cycle in progress.

It is possible to access three contiguous frames from the microcontroller by reading the three data sector windows, 0x8000 to 0x8FFF, 0x9000 to 0x9FFF and 0xA000 to 0xAFFF. This function is required for the decoding of the sub-code information. If the ‘next’ frame wraps past the last frame pointer (LASTFRM) then the pointers are modified to wrap back to the start pointer onwards (FEFRM#); this section is transparent to the microcontroller.

Table 13 Program memory control register: 0xF09F

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
PRGMEM	R/W	DATAPRG	EN_WIN	SEL_TOP	INV_A15	–	–	–	–

Register 0xF09F controls a system test feature where an SRAM is used for the 80C32 external program memory; note DATAPRG must be set for any of these features to be enabled.

Table 14 PRGMEM field descriptions

FIELD	LOGIC	DESCRIPTION
INV_A15	0	normal operation
	1	invert A(15) to program memory for data memory access
SEL_TOP	0	select bottom 32 kbyte window
	1	select top 32 kbyte window
EN_WIN	0	windowing disabled
	1	32 kbyte windows are enabled and SEL_TOP is used to determine window selected
DATAPRG	0	normal operation
	1	data memory is mapped to program memory and data memory DRAM accesses are disabled

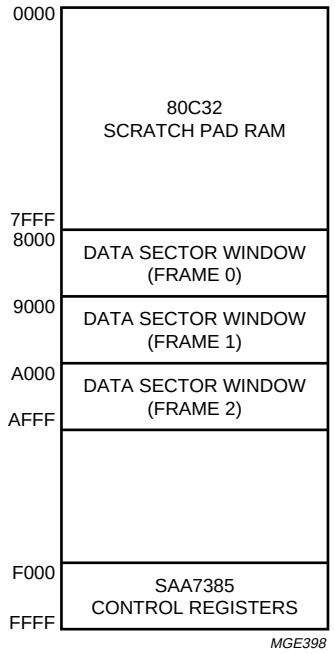


Fig.12 Address map for the microcontroller.

Table 15 SAA7385 address map details for the 80C32

ADDRESS	FUNCTION
0000 to 7FFF	This 32 kbyte window is used to address and portion the DRAM buffer. It is intended for non-frame mapped memory to be addressed through this window. The upper page address bits (to address the full range of the DRAM buffer) are set by the linear address page register (PAGEREG).
8000 to 8FFF	All accesses to frame memory use this window to read or write to the buffer memory. The actual address to the DRAM buffer is Micro Frame Number (MICFRM#) times 3 k plus the 12 LSBs from the 80C32.
9000 to 9FFF	This frame window is identical to the frame 0 window with the exception that one is added to the value from the Micro Frame Number (MICFRM#).
A000-AFFF	This frame window is identical to the frame 0 window with the exception that two is added to the value from the Micro Frame Number (MICFRM#).
B000-EFFF	Not used; outputs are driven LOW
F000-FFFF	SAA7385 control registers

9 FRONT PANEL AND MISCELLANEOUS CONTROL SIGNALS

This chapter describes the various SAA7385 control signals; front panel and basic engine signals, jumper settings and use of the general purpose signals.

Table 16 Start clock doubler: 0xF091

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
CLKSEL	W	–	–	–	–	–	–	–	–

A write of any value to this address will engage the clock doubler. The state of the doubler may be obtained by reading C_34_16 in BRGSEL (see Table 30). If this bit is set then the clock doubler is engaged. On power-on, the clock doubler is disabled. Once the clock doubler is engaged, it can only be reset by one of the reset sources; a power-on reset, a SCSI reset or a reset from the watch-dog timer. The clock doubler must not be engaged when a 33.8688 MHz clock is connected to OSCIN (pin 117).

Table 17 Frequency synthesizer test register: 0xF0D8; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
FSTEST	R/W	–	–	USE_IN	CPSEL	FVCOD	FS_LOCK	–	–

Note

1. Register 0xF0D8 is used for IC-level testing and to power down the frequency synthesizer. Only bit USE_IN should be asserted in normal operation.

Table 18 FSTEST field description

FIELD	LOGIC	DESCRIPTION
FS_LOCK	0	normal operation
	1	3-state $\overline{\text{LED}}$ and switch LQDATA to FS_LOCK
FVCOD	0	normal operation
	1	test mode for FVCOD from the synthesizer
CPSEL	0	normal operation
	1	test mode for CPSEL from the synthesizer
USE_IN	0	use internal synthesizer
	1	power down the synthesizer and operate off a 33.87 MHz input clock

Table 19 Disconnected pulse-width modulator control: 0xF0B3; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
WTPWM	R/W	–	–	–	–	–	–	–	–

Note

1. Register 0xF0B3 is disconnected.

Table 20 General logic control register: 0xF0B9; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
WTGCTL	W	–	–	PWMSEL	$\overline{\text{LED}}$	LA_MUTE	RA_MUTE	CHANNEL1	CHANNEL0

Note

1. Register 0xF0B9 controls the audio mixing and the LED.

Table 21 WTGCTL field description

FIELD	LOGIC	DESCRIPTION
CHANNELS	00	mute
	01	right data sent to both channels
	10	left data sent to both channels
	11	stereo
RA_MUTE	–	right channel digital mute
LA_MUTE	–	left channel digital mute
$\overline{\text{LED}}$	–	active LOW control for the light emitting diode
PWMSEL	–	PWM is disconnected

Table 22 Drive switches register: 0xF0BA; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
RDSW	R	–	–	–	$\overline{\text{PLAY}}$	$\overline{\text{SKIPFWD}}$	$\overline{\text{SKIPBACK}}$	$\overline{\text{EJECT}}$	$\overline{\text{TRAYSW}}$

Note

1. Register 0xF0BA is used for sensing the drive switches.

Table 23 RDSW field description

FIELD	LOGIC	DESCRIPTION
$\overline{\text{TRAYSW}}$	0	tray position in
	1	tray position out
$\overline{\text{EJECT}}$	–	user is requesting the drive tray to open (active LOW)
$\overline{\text{SKIPBACK}}$	–	user is requesting a track skip backwards (active LOW)
$\overline{\text{SKIPFWD}}$	–	user is requesting a track skip forwards (active LOW)
$\overline{\text{PLAY}}$	–	user is requesting the drive to play (active LOW)

Table 24 Jumper status register: 0xF0C9; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
RDJMPRS	R	JUMPER7 to JUMPER0							

Note

1. The bit fields for the jumpers are explained in Table 25.

Table 25 RDJMPRS field description

FIELD	DESCRIPTION
JUMPER7 to JUMPER0	Indicates the value of the DRAM data bus on power-up. The data bus may be pulled HIGH or LOW using weak pull-ups and pull-downs hence up to eight jumper settings are accommodated.

Table 26 General purpose bits: 0xF0C2; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
GPIOCTL	R/W	GPDAT4	GPDIR4	GPDAT3	GPDIR3	GPDAT2	GPDIR2	GPDAT1	GPDIR1

Note

1. Register 0xF0C2 controls the direction and output state of the general purpose I/O bits on the SAA7385. Reading the GPIO direction bits reflects the last value that was written to the register. The four GPIO data bits shows the current value of the input signals in the input mode. In the output mode, the last value written to the output latches is that which is read back.

Table 27 GPIOCTL field description

FIELD	DESCRIPTION
GPDIR1	General purpose bit direction control. Default LOW puts GPIO1 into the input mode, setting this HIGH puts GPIO1 in output mode.
GPDAT1	GPIO1 data bit.
GPDIR2	General purpose bit direction control. Default LOW puts GPIO2 into the input mode, setting this HIGH puts GPIO2 in output mode.
GPDAT2	GPIO2 data bit.
GPDIR3	General purpose bit direction control. Default LOW puts GPIO3 into the input mode, setting this HIGH puts GPIO3 in output mode.
GPDAT3	GPIO3 data bit.
GPDIR4	General purpose bit direction control. Default LOW puts GPIO4 into the input mode, setting this HIGH puts GPIO4 in output mode.
GPDAT4	GPIO4 data bit.

9.1 S2B UART registers

This section describes the registers used for the S2B UART control.

Table 28 S2B UART transmit, receive and status buffer: 0xF0A1, F0A2 and F0A3

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
WTS2B[1]	W	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
RDS2B	R	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
S2BSTAT	R	–	–	$\overline{SPR}$	$\overline{CPR}$	TXDRDY	PE	OVRRUN	RXDRDY

Note

1. WTS2B is for the transmit data byte from the S2B UART and RDS2B is for the receive data byte from the S2B UART.

Table 29 S2BSTAT field description

FIELD	DESCRIPTION
RXDRDY	A logic 1 indicates that the receive data is valid.
OVRRUN	A logic 1 indicates that the data in the receive buffer was not read before it was over written by the next byte.
PE	A logic 1 indicates that a parity error was detected in the receive data byte; this is usually caused by the wrong baud rate.
TXDRDY	A logic 1 indicates that the transmit data buffer is empty and ready for another byte.
$\overline{CPR}$	S2B handshake bit which may be interpreted as ‘clear to send’; this is generated automatically by the UART. It is asserted whenever the UART receiver is ready for a byte and negated as soon as the stop bit is shifted in. It is again asserted as soon as the received byte is read by the 80C32.
$\overline{SPR}$	S2B handshake bit which may be interpreted as ‘request to send’; this is received from the CD-ROM engine UART transmitter and will generate an interrupt to the 80C32 if the TXRDY bit is set and the interrupt is not masked.

Table 30 Baud rate generator control: 0xF0C0; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
BRGSEL	R/W	C_34_16	LOCK	EVENPAR	INVSUBC	INVQ	–	ICESEL1	ICESEL0

Note

1. Register 0xF0C0 controls the S2B UART baud rate and selective inversion of the sub-code information. Control over the parity and the clock doubler is also included together with the ability to invert the sub-code and Q-channel information.

Table 31 BRGSEL field description

FIELD	LOGIC	DESCRIPTION
ICESEL1 to 0	00	31.25 kbaud transfer rate
	01	62.5 kbaud transfer rate
	10	187.5 kbaud transfer rate
	11	not specified
INVQ	–	inverts all Q-channel information if set
INVSUBC	–	inverts all sub-code information if set
EVENPAR	–	selects even parity for S2B UART is set
LOCK	–	read only information; indicates clock synthesizer is stable (after reset) and it is ready to set C_34_16
C_34_16	–	once LOCK is HIGH, asserting this bit engages the clock doubler

Table 32 UART special control register: 0xF09E; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
UARTCTL	R/W	EXTUART	UARTCNT	DIVIDE5	DIVIDE4	DIVIDE3	DIVIDE2	DIVIDE1	DIVIDE0

Note

1. Register 0xF09E allows the 80C32 UART clock to be derived from 16.945 MHz. This external UART clock is required for reliable operation of the UART if the 80C32 is used for other functions during the transfer.

Table 33 UARTCTL field description

FIELD	LOGIC	DESCRIPTION
DIVIDE5 to 0	–	value 0 produces a 0.264 MHz clock and 58 produces a 2.82 MHz clock for the UART; this is the maximum accepted by the 80C32, a smaller number is required for guaranteed operation e.g. 15
UARTCNT	0	normal UART data input sampled by the external clock
	1	select a UART data input sampled by the clock from the internal counter
EXTUART	0	use external UART clock; disables internal clock
	1	switch external UART clock input from pin to this internal counter

9.2 Miscellaneous control registers

Table 34 53CF90 direction and audio mode control: 0xF0C1; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
AUSWP	R/W	TEST	–	–	–	OVER4X	BSB	–	–

Note

1. Register 0xF0C1 controls the audio mode byte swapping and a test mode bit.

Table 35 WTDIR field descriptions

FIELD	DESCRIPTION
BSB	Byte swap bit. Defaults to swapping the most significant byte and least significant byte in the audio mode such that the least significant byte of all audio samples is stored at even addresses in the DRAM. Setting this HIGH causes the audio data to be stored in the same way as in the data mode.
OVER4X	4× over-sampling bit selection; default LOW select transmit, or no over-sampling, mode for the sub-code and C-flag UARTs. Setting this bit HIGH will cause the sub-code and C-flag data to be sampled at one quarter the data rate allowing Q-channel information to be correctly stored in the registers while the CD-60 is outputting audio data at 4× over-sampling.
TEST	Enables internal signals to be multiplexed out when the TEST pin (pin 125) is HIGH.

Table 36 SCSI mode control register: 0xF0FD; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
SCSIMOD	R/W	–	–	–	OFF_ADR	OFF_END	OFF_STR	RD_BUF	BYT/PAG

Note

1. Register 0xF0FD controls the operation of the interface to the SCSI controller. The outputs of these registers are used to directly control DRAM access cycles, and will affect any current DRAM cycle in progress.

Table 37 SCSIMOD field description

FIELD	LOGIC	DESCRIPTION
BYT/PAG	0	SCSI DRAM byte mode access
	1	SCSI DRAM page mode access
RD_BUF	0	SCSI read/write control; read from buffer memory
	1	SCSI read/write control; write to buffer memory
OFF_STR	0	SCSI offset start A/B control; select A registers
	1	SCSI offset start A/B control; select B registers
OFF_END	0	SCSI offset end A/B control; select A registers
	1	SCSI offset end A/B control; select B registers
OFF_ADR	0	SCSI transfers use only A registers
	1	SCSI transfers use A and B registers

Table 38 DRAM selection/test mode: 0xF0FE; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
DRAMSEL	R/W	TEST	RESTEST3 to RESTEST0				DBL_SPD	1_MEG	2_DRAMS

Note

1. After power-up or reset, DRAMSEL should be the first register that is programmed. This register is used to select the number and the type of DRAMs used. The output of this register is used to control the DRAM access directly and will affect any current DRAM cycle.

Table 39 SCSIMOD field description

FIELD	LOGIC	DESCRIPTION
2_DRAMS	0	single DRAM used
	1	two DRAMs used
1_MEG	0	256 k $\times$ 4
	1	1 M $\times$ 4
DBL_SPD	0	single speed refresh; condition after reset
	1	double speed refresh; only set if system is running at 0.5 master clock speed
RESTEST3 to RESTEST0	–	reserved for test: enable DRAM access test; switch multiplexer control, enable interrupts
TEST	0	normal operation
	1	test mode; read back of RESTEST3 to RESTEST0 is gated by this bit

10 FRONT-END

This Chapter explains the information of the front-end circuitry.

10.1 Minute Second Frame (MSF) addressing and header information

Table 40 Header mode and MSF from block decoder: 0xF092, F093, F09A and F09B

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
MODE	R	MODE7 to MODE0							
MINS	R	MINUTES7 to MINUTES0							
SECS	R	SECONDS7 to SECONDS0							
FRMS	R	FRAME7 to FRAME0							

These registers contain the mode, minute, second and frame information from the header when in data mode. This data is valid whenever the HDDRDY bit in the RDDSTAT register is set. In audio mode, the MSF address is taken from the Q-channel information.

Table 41 Q-channel information: 0xF0A9, F0AA, F0AB, F0B1 and F0B2

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
RDTK	R	TRACK7 to TRACK0							
RDMD	R	MODE7 to MODE0							
RDMN	R	ABSMIN7 to ABSMIN0							
RDSC	R	ABSSEC7 to ABSSEC0							
RDFM	R	ABSFRM7 to ABSFRM0							

These registers contain the information taken from the raw sub-channel information from the CD decoder. Due to the fact that this data has not had any error correction applied to it, it is necessary to perform a CRC check for validity. Twelve bytes of Q-channel information are assembled from each sector of data; the last two bytes contain the CRC parity. Therefore the validity of the contents of these registers can only be determined after the last bit has been loaded and checked.

Table 42 Times from QCHRDY to BADQ (RDDSTAT)

SPEED	TIME (µs)
n = 1	2177
n = 2	1089
n = 4	545
n = 6	363
n = 8	273

For example, at the n = 4 data rate, the BADQ flag (in RDDSTAT) should be checked 545 µs after the QFRMRDY interrupt (from RDDSTAT) is asserted. If BADQ is LOW then the contents of the Q-channel registers are valid; otherwise the CRC check failed and the Q-channel information may be incorrect. If the data clock is running (ECMD LOW or ENABRED HIGH) then BADQ will be valid until the end of the sector; otherwise BADQ is valid until the end of the next Q frame.

10.2 Front-end status and control

Table 43 Front-end control: 0xF0BB; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
FECTL	R/W	SIM_EOF	RSMD	BREAK	RWMD	ENABRED	AUDMODE	$\overline{\text{SYNASYN}}$	$\overline{\text{ECMD}}$

Note

1. Register 0xF0BB controls the front-end of the SAA7385. The naming convention used here is similar to that used in the block decoders.

Table 44 FECTL field description

FIELD	LOGIC	DESCRIPTION
$\overline{\text{ECMD}}$	0	Data is shifted in and stored when the next synchronization pattern is detected; ($\overline{\text{SYNASYN}}$ = 1 and AUDMODE = 0).
	1	Data flow stop just before next synchronization pattern. $\overline{\text{ECMD}}$ is set on a reset condition; ($\overline{\text{SYNASYN}}$ = 1).
$\overline{\text{SYNASYN}}$		Synchronous/asynchronous selection; this controls the method by which data is started and stopped by the block decoder, only operates in data mode.
	0	Causes a ‘panic stop’. A partial frame will reside in current and subsequent buffers unless SIM_EOF is set then cleared; (ECMD = 1).
	1	Data is started and stopped on frame boundaries (on synchronization patterns).
AUDMODE	0	Data mode. Cleared on reset.
	1	Audio mode, where the bit clock is shifted to accommodate EIAJ format. HQRDY in INTRFLG follows HDRRDY in data mode and QFRMDRY in audio mode.
ENABRED		Enable red book to data path; while in audio mode, this is equivalent to $\overline{\text{ECMD}}$ in the data mode. No asynchronous stop is provided in the audio mode.
	0	Data flow will stop when the next F1 C-flag is detected. Cleared on a reset condition.
	1	Red book data is input to buffer after the detection of the next F1 C-flag.
RWMD	–	This must be pulsed HIGH then LOW every 212 ms to prevent the watch-dog timer from resetting the SAA7385 and the drive. The length of the reset pulse is 967 μs. If RWMD is set, the watch-dog timer is disabled.
BREAK	–	When set, the S2B UART transmitter output is held HIGH.
RSMD	–	When the pulse is HIGH then LOW, the block decoder begins to search for a synchronization pattern in the data bitstream. Once a synchronization pattern is found, MODE, MINS, SECS, and FRMS become valid.
SIM_EOF	–	This provides a firmware reset to the frame sequencer and parts of the buffer manager. This would be required if an asynchronous stop of the data stream occurs. Pulsing this HIGH then LOW resets all counters and establishes a ‘beginning of frame’ state. DCOACT in RDDSTAT must be LOW to allow SIM_EOF to have any effect. If SIM_EOF is set, no data or sub-code is stored in the buffer.

Table 45 Read status register: 0xF0C3; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
RDDSTAT	R	DCOTACT	BADQ	QFRMRDY	HDRRDY	HDRERR	CRCERR	DATERR	SYNCERR

Note

1. The information in register 0xF0C3 is a copy of the status byte written to the data buffer at the end of every frame. SYNCERR, DATERR and CRCERR are essentially unusable since they are valid only long enough to be written to the buffer.

Table 46 RDDSTAT field description

FIELD	LOGIC	DESCRIPTION
SYNCERR	0	Good synchronization detected (valid for 120 ns at the end of a sector).
DATERR	0	Good data (valid for 120 ns at the end of a sector).
CRCERR	0	Good CRC (valid for 120 ns at the end of a sector).
HDRERR	0	Good header. If the automatic storage is selected, assertion of HDRERR inhibits data storage.
	1	EFAB during reception of header (valid while HDRRDY set). If the automatic storage is selected, assertion of HDRERR inhibits data storage.
HDRRDY	–	When set, a valid header is available. If the header is not read within a frame time, this remains set until the next synchronization pattern and will be set again when the next header arrives. It is cleared when any of the header bytes are read. This bit generates an interrupt to the microcontroller when in data mode.
QFRMRDY	–	When set, all ten Q-channel bytes are received waiting to be read (BADQ is known). It is reset at the end of frame or when any of the Q-channel bytes are read. This bit generates an interrupt to the microcontroller when in audio mode.
BADQ	–	If Q-channel information failed CRC then BADQ is set. It is reset on next good CRC check or on end of frame if DCOACT is running. If DCOACT is not running (i.e. audio mode) BADQ is reset on next detection of sub-code gap. If AUTOSTR in WTDIR is selected, assertion of BADQ inhibits audio data storage.
DCOTACT	–	Set when data is being shifted an and stored in the buffer: this will remain HIGH for the entire transmission.

11 BUFFER MANAGER

11.1 Front-end to buffer manager interface

The buffer manager interface to the front-end is write only with no handshaking. The front-end passes one byte of data and a write strobe to the buffer manager; this byte will be one of five types of data (see Table 47). The data byte is latched and the interface is given the highest priority thus no wait signal is required. The other signals passed from the front-end logic are an end-of-frame strobe (which is the same as the status byte write strobe), a software-generated reset pulse (used to reset the front-end counters), and a reset pulse for the Q-channel and sub-code offset counters.

The buffer manager provides the remainder of the logic to write the data into the RAM and keep track of the frame addresses and offset addresses. This logic consists of a 12-bit frame offset counter FEOFF, for data and an 9-bit frame counter; this is a relative frame number and is not related to the CD-ROM frame number. Offset counters are also provided for the four other types of data. The other offset address generators are based on fixed addresses, and they will be loaded with the start address at the beginning of each frame. The five types of data from the front-end are loaded into the frame map as shown in Table 47.

Table 47 Data types from the front-end

START	END	LENGTH	DATA TYPE
0x000	0x92F	0x930	header, data and parity
0x930	0x93F	0x010	Q-channel
0x940	0x99F	0x060	sub-channel
0x9A0	0xAC5	0x126	error flags
0xBDE	0xBDE	0x001	status byte

Initially the front-end frame counter and all of the offset counters are cleared by reset or loaded with the contents of FEFRM# when the last frame as specified by LASTFRM is filled; therefore FEFRM# should be loaded with the required starting frame number. The data frame offset counter, FEFRMOFF, may be loaded for test purposes, but is cleared at the end of each frame.

LASTFRM establishes the limit of the frame memory. This register should be loaded with the required number of frames; the amount of memory used is 3 kbytes times the number of frames. The front-end frame address counter uses this value to determine the correct location to clear the counter.

Once the data load process starts, the offset counter (FEFRMOFF) increments after each byte is written into memory. This process continues until an end of frame signal is received from the front-end logic. If an error occurs and the offset counter increments past the maximum 2352, an interrupt will be issued to the microcontroller.

Table 48 Front-end frame offset: 0xF0E2, F0E3

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
FEFRMOFF	R/W	OFFSET7 to OFFSET0							
FEFRMOFF	R/W	–	–	–	–	OFFSET11 to OFFSET8			

This register allows the front-end frame offset counter to be read and reloaded. The counter associated with these registers is loaded after the most significant byte is written; the least significant byte must be written first to ensure that the counter is loaded correctly. If a DRAM access is in progress that uses the address from the counter, the update will be delayed until the access is complete. This counter is cleared on reset or after each frame is loaded into buffer memory. Therefore, this register should not be loaded during normal operation.

Table 49 Front-end frame number: 0xF0E4, F0E5

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
FEFRM#	R/W	NUM7	NUM6	NUM5	NUM4	NUM3	NUM2	NUM1	NUM0
FEFRM#	R/W	–	–	–	–	–	–	–	NUM8

This register allows the front-end frame number counter to be read and reloaded. The counter associated with these registers is loaded after the most significant byte is written; the least significant byte must be written first to ensure that the counter is loaded correctly. If a DRAM access is in progress that uses the address from the counter, the update will be delayed until the access is completed. This counter is cleared on reset or when the last frame, as specified by LASTFRM, is filled. Therefore, this register should only be loaded if a non-zero start frame number is required. The frame counter is automatically cleared at the end of the frame buffer memory and thus multiple passes of a non-zero start address will require a re-load for each pass; it is not practical to do this in real-time.

Table 50 Last frame number for storage: 0xF0F8, F0F9

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
LASTFRM	R/W	NUM7	NUM6	NUM5	NUM4	NUM3	NUM2	NUM1	NUM0
LASTFRM	R/W	–	–	–	–	–	–	–	NUM8

These registers are used by the buffer manager to set the top of frame storage memory (wrap point). Any memory past this point is available for general usage by the microcontroller. The outputs of the registers are used directly to control DRAM access cycles, and will affect any current DRAM cycle in progress. Both the SCSI address counter and the front-end frame address counter use this value to determine the correct location to clear their respective frame counters.

11.2 Microcontroller to buffer manager interface

The microcontroller interface allows the microcontroller to read or write any register or the frame store memory. Frame and offset registers are used to update the counters after the most significant byte has been loaded. Frame store memory is addressed using a frame number register controller by the microcontroller. Logic is provided to allow the frame number of the last complete frame received (LSTCMPFM) from the front-end to be read by the microcontroller for the purpose of setting the microcontroller frame address. Memory beyond the last frame number is available to the microcontroller using the microcontroller bottom 32 kbyte located at 0x0000 to 0x7FFF. The 4 kbyte segment at 0x8000 to 0x8FFF is used to address the current frame memory. Also, the next frame may be accessed at 0x9000 to 0x9FFF, and the current frame plus 2 may be accessed at 0xA000 to 0xAFFF. A page register is provided to allow the microcontroller to address the complete memory range in 32 kbyte pages. All microcontroller accesses to memory are single byte read or write cycles.

All microcontroller accesses to memory will generate a wait state. If no other accesses to memory are in progress then a minimum wait state cycle will be generated. If, however, other cycles are in progress then the microcontroller is forced to wait until the lower priority access cycles finish and any high priority access cycles are completed. The worst case wait is four complete access cycles; a total of 20 clock cycles in byte mode and 32 cycles in nibble mode.

Table 51 Last complete frame number: 0xF0E6, F0E7

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
LSTCMPFM	R	NUM7	NUM6	NUM5	NUM4	NUM3	NUM2	NUM1	NUM0
LSTCMPFM	R	–	–	–	–	–	–	–	NUM8

11.3 ECC to buffer manager interface

The ECC logic is able to access the buffer manager frame memory in either byte or burst mode. The ECC logic provides an offset address and uses a frame address programmed by the microcontroller, ECCFRM#. The logic can write a single byte or variable number of bytes. In the event of an access to a variable number of bytes, the ECC logic will assert the signal BURST and EREQ to indicate that a large number of cycles are requested. For each read or write cycle, the buffer manager will toggle EACK HIGH for one clock cycle to indicate that one byte of data has been read from or written to the memory. A single byte cycle will be the same with the exception that BURST will remain negated (LOW). In the event of a higher priority memory access request during a burst cycle, EACK will remain LOW for the duration of the higher priority access cycle. At the end of the higher priority access, the burst cycle will resume and EACK will again toggle HIGH after each read or write is completed.

Table 52 ECC frame number address registers: 0xF0F4, F0F5; notes 1 and 2

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
ECCFRM#	R/W	NUM7	NUM6	NUM5	NUM4	NUM3	NUM2	NUM1	NUM0
ECCFRM#	R/W	–	–	–	–	–	–	–	NUM8

Notes

1. These registers provide the frame number address for ECC access to memory. The counter associated with these registers is loaded after the most significant byte is written; the least significant byte must be written first to ensure that the counter is loaded correctly. If a DRAM access is in progress that uses the address from the counter, the update will be delayed until the access is completed.
2. ECCFRM# is used to determine the frame address for all ECC operations. This register must be reloaded for each frame accessed by the ECC.

11.4 SCSI to buffer manager interface

The SCSI registers should be loaded prior to starting an SCSI transfer. The SCSIMOD register should be loaded first. BYT/PAG from this register is used to control the type of DRAM access used by the SCSI interface. If BYT/PAG is HIGH then burst mode access cycles are selected; multiple $\overline{CAS}$ access cycles are used to access data as fast as possible. RD_BUF from SCSIMOD controls the direction of data flow to the buffer memory; this bit is kept LOW to allow reading of data from the DRAM buffer. If RD_BUF is asserted then SCSI data will be written to the DRAM buffer. OFF_ADR from SCSIMOD is used to select between one and two offset mode for the SCSI transfer. OFF_ADR LOW selects single offset mode in which one block of data is transferred for each frame of the buffer.

The transfer block is specified by registers SCSIOFFS and SCSIOFFE. For each frame, the transfer will start at the address specified by SCSIOFFS and continue until the address specified by SCSIOFFE is transferred. After each block is transferred, the frame address SCSICFRM will be incremented and the transfer will continue with the same address block from the next frame. If OFF_ADR is set, then two blocks of data are transferred. In the two offset mode, both SCSIOFFS and SCSIOFFE are used to access two independent register pairs; for simplicity, these are called the A registers and the B registers. In this event, the transfer for each frame is a two step process.

First, the offset block specified by SCSIOFFS-A and SCSIOFFE-A is transferred; the transfer address range is from SCSIOFFS-A to SCSIOFFE-A and includes both the start and end addresses. After the first offset block is

transferred, the second offset block as specified by SCSIOFFS-B and SCSIOFFE-B is transferred. The frame address will not be incremented until after both offset blocks are transferred. Once both offset blocks are transferred, the frame address is incremented and again the two offset blocks are transferred for the next frame. Reading and writing of the A and the B registers is controlled by an automatic switching after the most significant bytes of the registers are written. After power-up or reset the pointer to the A registers will be selected. If the dual offset mode is selected, the A/B switch will be toggled when the most significant bytes of the registers are written; either the most significant bytes of SCSIOFFS or SCSIOFFE. Any future reads or writes will access the B registers.

The process of loading and reading the two SCSI offset address pairs can be monitored and controlled by OFF_STR and OFF_END from SCSIMOD. Reading OFF_STR shows the status of the A/B switch for the SCSIOFFS-A/B registers; reading OFF_END shows the status of the A/B switch for the SCSIOFFE-A/B registers. A write to SCSIMOD with OFF_STR LOW will clear the A/B switch for the SCSIOFFS registers; a write to SCSIMOD with OFF_END LOW will clear the A/B switch for the SCSIOFFE registers.

SCSISFRM is used to determine the starting frame address for all SCSI operations. The associated counter is automatically incremented after each frame, and is cleared when the last frame as specified by LASTFRM is transferred. To update the SCSI frame address counter, SCSISFRM must be rewritten. The current SCSI frame address is available by reading SCSICFRM. The frame counter is automatically cleared at the end of the frame buffer memory and thus multiple passes of a non-zero start address will require a re-load for each pass; it is not practical to do this in real-time.

The SCSIOFFS registers access either one or two register pairs as controlled by SCSIMOD. SCSIOFFS determines the starting offset address for a SCSI transfer.

The SCSIOFFE register accesses either one or two register pairs as controlled by SCSIMOD. SCSIOFFE determines the ending offset address for a SCSI transfer.

Remarks:

- If two offset pairs are used, the A start offset must be written last to ensure that the correct offset start address is loaded into the counter.
- In the two offset mode, reading the register after loading is not possible due to the automatic switching feature; if the A offset pair is written, and the register pair is read, the B offset pair would be read.

Table 53 SCSI offset start register (A and B): 0xF0E8, F0E9; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
SCSIOFFS	R/W	OFFSET7 to OFFSET0							
SCSIOFFS	R/W	–	–	–	–	OFFSET11 to OFFSET8			

Note

1. These registers, together with the offset end registers, allow full control over the number of frame bytes that will be transferred to the SCSI port.

Table 54 SCSI offset end registers (A and B): 0xF0EA, F0EB; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
SCSIOFFE	R/W	OFFSET7 to OFFSET0							
SCSIOFFE	R/W	–	–	–	–	OFFSET11 to OFFSET8			

Note

1. These registers together with the offset start registers, allow full control over the number of frame bytes that will be transferred to the SCSI port.

Table 55 SCSI transfer start frame number: 0xF0EC, F0ED; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
SCSISFRM	R/W	NUM7	NUM6	NUM5	NUM4	NUM3	NUM2	NUM1	NUM0
SCSISFRM	R/W	–	–	–	–	–	–	–	NUM8

Note

1. This register determines the starting frame number for an SCSI transfer. The outputs of the registers are used to directly control DRAM access cycles, and will affect any current DRAM cycle in progress. The SCSI frame pointer will wrap back to this point.

Table 56 SCSI current transfer frame: 0xF0EE, F0EF; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
SCSICFRM	R	NUM7	NUM6	NUM5	NUM4	NUM3	NUM2	NUM1	NUM0
SCSICFRM	R	–	–	–	–	–	–	–	NUM8

Note

1. This register allows the current SCSI frame transfer number to be read.

11.5 Miscellaneous buffer manager considerations

The following bandwidth limitation must be observed in normal operation:

- All burst mode operations (SCSI and ECC) take twice as long in nibble mode, and single cycle operations take 60% longer; 236 ns compared with 147.5 ns. For this reason, operation of the DRAM interface at the maximum 8 times transfer rate in nibble mode is not supported.
- In byte mode, only 833 ns is available between each data write from the front-end at the maximum 8 times transfer rate. At the end of the frame, multiple front-end byte writes may stack up and therefore it is recommended that the 80C32 avoids DRAM access at the end of the frame.

If two SCSI offset pairs are used, the A start must be written last to ensure that the correct offset start address is loaded into the counter.

In the two SCSI offset mode, reading the register after loading is not possible due to the automatic switching feature.

11.6 53CF94 related registers

In the 53CF94 SCSI controller, some registers are read only and others are write only. These share the same address and the multiplexing between the two depends on the read or write select.

This part contains only a brief description of the register definitions. For a more detailed definition, see the data sheet for the SCSI device.

Table 57 Transfer count registers: 0xF0A4, F0A5 and F0BE- 53CF94 addresses: 0x00, 01, 0E; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
TCLOW	R/W	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
TCMID	R/W	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
TCHIGH	R/W	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Note

1. These registers form the 24-bit transfer count value for DMA operations; they specify the number of bytes that are transferred.

Table 58 FIFO buffer register: 0xF0A6 - 53CF94 address 0x02; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
FIFO	R/W	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Note

1. The FIFO is a 16 $\times$ 9-bit buffer between the SCSI interface and the memory. The SCSI command packet will be present in this FIFO ready to be read by the microcontroller at the end of the SCSI command phase.

Table 59 Command register: 0xF0A7 - 53CF94 address 0x03; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
CMD	R/W	ENDMA	COMMAND6 to COMMAND0						

Note

1. The command register is a two-deep read/write register used to pass commands to the SCSI controller; up to two commands can be stacked at the same time. When ENDMA is set, the command is a DMA instruction.

Table 60 Status register and destination ID: 0xF0AC - 53CF94 address 0x04; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
STAT	R	INT	GE	PE	TC	VGC	MSG	C/D	I/O
STAT	W	–	–	–	–	–	ID2	ID1	ID0

Note

1. This register contains the flags indicating the occurrence of certain events.

Table 61 STAT field description

FIELD	DESCRIPTION
MSG, C/D, I/O	These indicate the phase on the SCSI bus.
VGC	Valid Group Code; set if the group code matches code defined in ANSI X3.131-1986.
TC	Terminate Count; set when transfer count decrements to zero.
PE	Parity Error; set if parity checking is enabled and a SCSI parity error occurs.
GE	Gross Error; set on various error conditions. These errors do not cause interrupts.
INT	Interrupt. Indicates whether device is trying to interrupt the microcontroller.
ID2 to ID0	Specifies the encode destination for selection or re-selection.

Table 62 Interrupt register and time-out register: 0xF0AD - 53CF94 address 0x05; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
INT	R	SRST	ILCMD	DIS	BS	FC	RESEL	SATN	SEL
INT	W	TIMEOUT7 to TIMEOUT0							

Note

1. This register is used in conjunction with STAT and SEQSTP to determine the cause of an interrupt.

Table 63 INT field description

FIELD	DESCRIPTION
SEL	Selected; set during selection phase if selected as a target with ATN negated.
SATN	Selected with ATN; set during selection if selected as a target with ATN asserted.
RESEL	Reselected; set during reselection phase if reselected as an initiator.
FC	Function complete; set after any target mode command has been completed.
BS	Bus service. Indicates that another device is requesting service; in target mode it is asserted whenever the initiator asserts ATN.
DIS	Disconnect. In target mode this is asserted and as a Terminate Sequence or a Command Complete Sequence command causes disconnection.
ILCMD	Illegal command; set when a reserved code is placed in CMD or when the command is from a mode group.
SRST	SCSI reset detect. This may be set if a reset on the SCSI bus is detected.
TIMEOUT7 to TIMEOUT0	Time-out period for response to selection or re-selection.

Table 64 Sequence step register and synchronous transfer register: 0xF0AE - 53CF94 address 0x06; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
SEQSTP	R	–	–	–	–	SOM	SS2	SS1	SS0
SEQSTP	W	–	–	–	TRANSPERIOD4 to TRANSPERIOD0				

Note

1. The register fields are described in Table 65.

Table 65 SEQSTP field descriptions

FIELD	DESCRIPTION
SS2 to SS0	Sequence step. Counter increments at various points in a command; may be used for error recovery.
SOM	Synchronous offset maximum. When clear, the synchronous offset has reached the maximum value.
TRANSPERIOD	Synchronous transfer period. Specifies minimum time between successive REQ or ACK pulses.

Table 66 FIFO flags and synchronous offset register: 0xF0AF - 53CF94 address 0x07; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
FIFOFLG	R	SS2	SS1	SS0	FF4	FF3	FF2	FF1	FF0
FIFOFLG	W	SYNCOFFSET7 to SYNCOFFSET0							

Note

1. The field description for the FIFO flags register is shown in Table 67.

Table 67 FIFOFLG field descriptions

FIELD	DESCRIPTION
FF	number of bytes in the FIFO
SS	duplicates of sequence step register
SYNCOFFSET	controls handshaking in synchronous transfer mode

Table 68 Configuration registers: 0xF0B4, F0B7, F0BC and F0BD - 53CD94 addresses 0x08, 0B, 0C and 0D; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
CONFIG1	R/W	SLOW	SRD	PTEST	PCHK	CTEST	MYBUSID2 to MYBUSID0		
CONFIG2	R/W	RFB	FE	EBC	DHZ	SCSI2	BPA	RPE	DPE
CONFIG3	R/W	IMRC	QTE	CDB10	FSCSI	FCLK	SRB	ADMA	T8
CONFIG4	R/W	–	–	–	–	–	EAN	TEST	BBTE

Note

1. The registers described allows the controller to be configured for the specific mode of operation.

Table 69 CONFIG1 to CONFIG4 field descriptions

FIELD	DESCRIPTION
MyBusID	ID of SCSI controller
CTEST	chip test mode enable
PCHK	enable parity checking
PTEST	parity test mode
SRD	SCSI reset reporting interrupt disable
SLOW	slow cable mode
DPE	DMA parity enable
RPE	register parity enable
BPA	target bad parity abort
SCSI2	allows support for SCSI-2 features
DHZ	DREQ high impedance
EBC	enable byte control
FE	features enable
RFB	reserve FIFO byte
T8	threshold eight
ADMA	alternate DMA mode
SRB	save residual byte
FCLK	fast clock
FSCSI	fast SCSI
CDB10	allows 10-byte group-2 commands to be recognized
QTE	queue tag enable
IMRC	ID message reserve check
BBTE	back-to-back transfer enable
TEST	transfer count test mode
EAN	enable active negation

Table 70 Clock conversion factor: 0xF0B5 - 54CF94 address 0x09; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
CLOCK	W	–	–	–	–	–	CONVERT7 to CONVERT0		

Note

1. This register must be set in accordance with the clock input frequency.

Table 71 CLOCK Field Descriptions

FIELD	DESCRIPTION
CONVERT7 to CONVERT0	clock conversion factor

Table 72 Test Register: 0xF06 - 53CF94 address 0x0A; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
TEST	R/W	–	–	–	–	–	HI-Z	INIT	TAR

Note

1. This register is enabled by setting the test mode in CONFIG1; after test mode is entered, a hardware reset or reset command must occur before normal operation may resume.

Table 73 TEST field descriptions

FIELD	DESCRIPTION
TAR	target mode
INIT	initiator mode
HI-Z	all outputs set to high impedance

Table 74 FIFO Bottom: 0xFBF - 53CF94 address 0x0F; note 1

MNEMONIC	R/W	DATA BYTE							
		7	6	5	4	3	2	1	0
FIFOBTM	W	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Note

1. This register is used during initiator synchronous data in to align 16-bit DMA transfers to word boundaries.

12 FRAME BUFFER ORGANIZATION

Table 75 Frame buffer organization

DECIMAL			HEXADECIMAL			DATA
START	END	LEN	START	END	LEN	
0	11	12	000	00B	00C	synchronization field
12	15	4	00C	00F	004	header
16	2063	2048	010	80F	800	frame data
2064	2067	4	810	813	004	CRC parity
2068	2075	8	814	81B	008	padding
2076	2247	172	81C	8C7	0AC	P parity
2248	2351	104	8C8	92F	068	Q parity
2352	2367	16	930	93F	010	Q channel
2368	2463	96	940	99F	060	sub-channel
2464	2757	294	9A0	AC5	126	error flags
2758	2761	4	AC6	AC9	004	CRC remainder
2762	2933	172	ACA	B75	0AC	P syndromes
2934	3037	104	B76	BDD	068	Q syndromes
3038	3038	1	BDE	BDE	001	status

Table 76 ECC RAM organization

DEC	HEX	BYTE NUMBER			
		3	2	1	0
000	000	psyn[00].s1	psyn[00].s0	qsyn[00].s1	qsyn[00].s0
↓	↓	↓	↓	↓	↓
204	0CC	psyn[51].s1	psyn[51].s0	qsyn[51].s1	qsyn[51].s0
208	0D0	psyn[52].s1	psyn[52].s0	flags[001]	flags[000]
↓	↓	↓	↓	↓	↓
340	154	psyn[85].s1	psyn[85].s0	flags[067]	flags[066]
344	158	flags[071]	flage[070]	flags[069]	flags[068]
↓	↓	↓	↓	↓	↓
564	234	flags[291]	flage[290]	flags[289]	flags[288]
568	238	unused[1]	unused[0]	flags[293]	flags[292]
572	23C	crc_rem[3]	crc_rem[2]	crc_rem[1]	crc_rem[0]
576	240	header[3]	header[2]	header[1]	header[0]
580	244	ecc_reg[03]	ecc_reg[02]	ecc_reg[01]	ecc_reg[00]
584	248	ecc_reg[07]	ecc_reg[06]	ecc_reg[05]	ecc_reg[04]
588	588	ecc_reg[11]	ecc_reg[10]	ecc_reg[09]	ecc_reg[08]

13 SUMMARY OF CONTROL REGISTER MAP

Table 77 Control register map for the SAA7385

ADDRESS	MNEMONIC	READ/WRITE	DESCRIPTION
F085	ECCCTL	R/W	ECC control register
F086	ECCSTAT	R	ECC status register
F08E	NUM_COR	R	ECC register for the number of corrections
F091	CLKSEL	W	start the clock synthesizer (doubler)
F092	MODE	R	header mode byte from block decoder
F093	FRMS	R	header frame byte from block decoder
F09A	SECS	R	header seconds byte from block decoder
F09B	MINS	R	header minutes byte from block decoder
F09E	UARTCTL	R/W	UART special control registers
F09F	PRGMEM	R/W	program memory control registers
F0A1	WTS2B	W	S2B UART transmit buffer
F0A2	RDS2B	R	S2B UART receive buffer
F0A3	S2BSTAT	R	S2B UART status register
F0A4	TCLOW	R/W	53CF94 (00); transfer counter low
F0A5	TCMID	R/W	53CF94 (01); transfer counter middle
F0A6	FIFO	R/W	53CF94 (02); FIFO
F0A7	CMD	R/W	53CF94 (03); command register
F0A9	RDTK	R	Q channel track number
F0AA	RDMD	R	Q channel mode number
F0AB	RDMN	R	Q channel minutes number (absolute)
F0AC	STAT	R/W	53CF94 (04); status register and destination ID
F0AD	INT	R/W	53CF94 (05); interrupt register and time-out
F0AE	SEQSTP	R/W	53CF94 (06); sequence step and synchronous transfer period
F0AF	FIFOFLG	R/W	53CF94 (07); FIFO flags and synchronous offset
F0B1	RDSC	R	Q channel seconds (absolute)
F0B2	RDFM	R	Q channel frames (absolute)
F0B3	WTPWM	R/W	pulse width modulator duty cycle select
F0B4	CONFIG1	R/W	53CF94 (08); configuration register number 1
F0B5	CLOCK	W	53CF94 (09); clock conversion factor
F0B6	TEST	W	53CF94 (0A); test mode
F0B7	CONFIG2	R/W	53CF94 (0B); configuration register number 2
F0B9	WTGCTL	W	GLIC control registers (audio control)
F0BA	RDSW	R	drive control switches register
F0BB	FECTL	R/W	front-end control register
F0BC	CONFIG3	R/W	53CF94 (0C); configuration register number 3
F0BD	CONFIG4	R/W	53CF94 (0D); configuration register number 4
F0BE	TCHIGH	R/W	53CF94 (0E); transfer count high and SCSI ID
F0BF	FIFOBTM	W	53CF94 (0F); FIFO bottom
F0C0	BRGSEL	R/W	baud rate generator select register

ADDRESS	MNEMONIC	READ/WRITE	DESCRIPTION
F0C1	AUSWP	R/W	audio byte swap control
F0C2	GPIOCTL	R/W	general purpose bits control register
F0C3	RDDSTAT	R	data status register
F0C9	RDJMPRS	R	option jumper register
F0D8	FSTEST	R/W	frequency synthesizer test register
F0E2	FEFRMOFF	R/W	front-end 8 LSBs; frame offset
F0E3	FEFRMOFF	R/W	front-end 4 MSBs; frame offset (bit 0 to bit 3)
F0E4	FEFRM#	R/W	front-end 8 LSBs of the frame
F0E5	FEFRM#	R/W	front-end 3 MSBs of the frame (bit 0 to bit 2)
F0E6	LSTCMPFM	R	8 LSBs; last complete frame number
F0E7	LSTCMPFM	R	3 MSBs; last complete frame number (bit 0 to bit 2)
F0E8	SCSIOFFS	R/W	SCSI 8 LSBs; offset start (A and B)
F0E9	SCSIOFFS	R/W	SCSI 4 MSBs; offset start (bit 0 to bit 3)
F0EA	SCSIOFFE	R/W	SCSI 8 LSBs; offset end (A and B)
F0EB	SCSIOFFE	R/W	SCSI 4 MSBs; offset end (bit 0 to bit 3)
F0EC	SCSISFRM	R/W	SCSI 8 LSBs; start transfer frame number
F0ED	SCSISFRM	R/W	SCSI 3 MSBs; start frame number (bit 0 to bit 2)
F0EE	SCSICFRM	R/W	SCSI 8 LSBs; current frame number
F0EF	SCSICFRM	R/W	SCSI 3 MSBs; current frame number (bit 0 to bit 2)
F0F4	ECCFRM#	R/W	ECC 8 LSBs; frame number (frame address)
F0F5	ECCFRM#	R/W	ECC 3 MSBs; frame number (bit 0 to bit 2)
F0F6	MICFRM#	R/W	microcontroller 8 LSBs; frame number (frame address)
F0F7	MICFRM#	R/W	microcontroller 3 MSBs; frame number (bit 0 to bit 2)
F0F8	LASTFRM	R/W	last frame number for storage 8 LSBs
F0F9	LASTFRM	R/W	last frame number 3 MSBs (bit 0 to bit 2)
F0FB	INTRMSK	R/W	interrupt mask register
F0FC	INTRFLG	R/W	interrupt flag register
F0FD	SCSIMOD	R/W	SCSI mode control
F0FE	DRAMSEL	R/W	DRAM selection/test mode register.
F0FF	PAGEREG	R/W	80C32 linear address page register

14 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
V_{DD}	digital supply voltage	–0.5	+7	V
$V_{i(max)}$	maximum input voltage on any pin	V_{SS} – 0.5	V_{DD} + 0.5	V
V_o	output voltage on any output	–0.5	+7	V
T_{stg}	storage temperature	–55	+150	°C

15 OPERATING CHARACTERISTICS

15.1 I²S-bus timing; data mode

V_{DD} = 4.75 to 5.25 V; V_{SS} = 0 V; T_{amb} = –10 to +70 °C; unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
I²S-bus timing (single speed × n); see Fig.13 and note 1						
CLOCK INPUT: CLAB						
T_{cy}	output clock period	sample rate = f_s	–	472.4/n	–	ns
		sample rate = 2 f_s	–	236.2/n	–	ns
		sample rate = 4 f_s	–	118.1/n	–	ns
t_{CH}	clock HIGH time	sample rate = f_s	166/n	–	–	ns
		sample rate = $2f_s$	83/n	–	–	ns
		sample rate = $4f_s$	42/n	–	–	ns
t_{CL}	clock LOW time	sample rate = f_s	166/n	–	–	ns
		sample rate = $2f_s$	83/n	–	–	ns
		sample rate = $4f_s$	42/n	–	–	ns
INPUTS: DAAB, WSAB AND EFAB						
t_{su}	set-up time	sample rate = f_s	95/n	–	–	ns
		sample rate = $2f_s$	48/n	–	–	ns
		sample rate = $4f_s$	24/n	–	–	ns
t_h	hold time	sample rate = f_s	95/n	–	–	ns
		sample rate = $2f_s$	48/n	–	–	ns
		sample rate = $4f_s$	24/n	–	–	ns

Note

1. The I²S-bus timing is directly related to the overspeed factor ‘n’ in the normal operating mode. In the lock-to-disc mode ‘n’ is replaced by the disc speed factor ‘d’.

15.2 EIAJ timing; audio mode

V_{DD} = 4.75 to 5.25 V; V_{SS} = 0 V; T_{amb} = −10 to +70 °C; unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
EIAJ timing (single speed × n); see Fig.14 and note 1						
CLOCK INPUT: CLAB						
T_{cy}	output clock period	sample rate = f_s	–	472.4/n	–	ns
		sample rate = 2 f_s	–	236.2/n	–	ns
		sample rate = 4 f_s	–	118.1/n	–	ns
t_{CH}	clock HIGH time	sample rate = f_s	166/n	–	–	ns
		sample rate = $2f_s$	83/n	–	–	ns
		sample rate = $4f_s$	42/n	–	–	ns
t_{CL}	clock LOW time	sample rate = f_s	166/n	–	–	ns
		sample rate = $2f_s$	83/n	–	–	ns
		sample rate = $4f_s$	42/n	–	–	ns
INPUTS: DAAB, WSAB AND EFAB						
t_{su}	set-up time	sample rate = f_s	95/n	–	–	ns
		sample rate = $2f_s$	48/n	–	–	ns
		sample rate = $4f_s$	24/n	–	–	ns
t_h	hold time	sample rate = f_s	95/n	–	–	ns
		sample rate = $2f_s$	48/n	–	–	ns
		sample rate = $4f_s$	24/n	–	–	ns

Note

1. The EIAJ timing is directly related to the overspeed factor ‘n’ in the normal operating mode. In the lock-to-disc mode ‘n’ is replaced by the disc speed factor ‘d’.

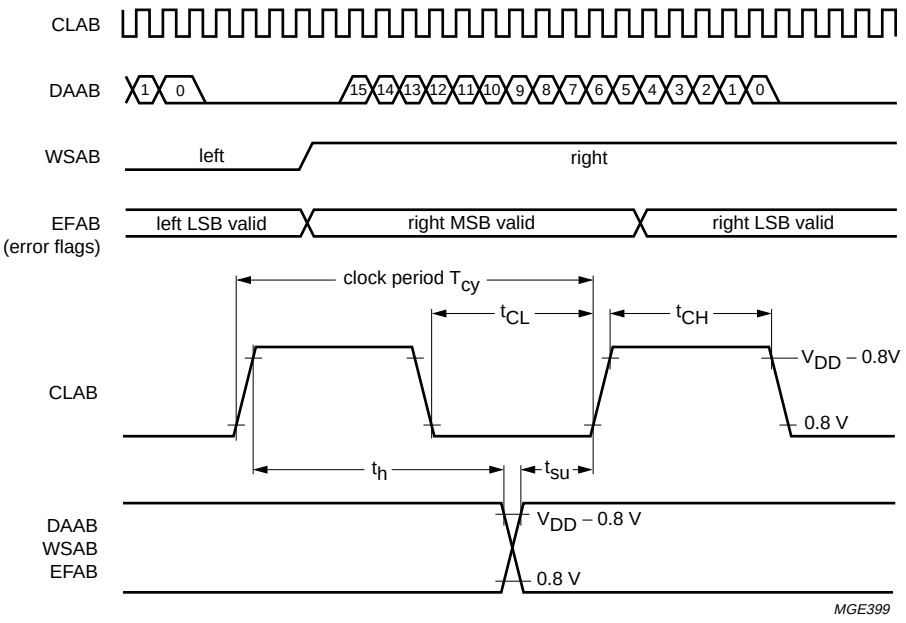


Fig.13 I^2S-bus timing diagram.

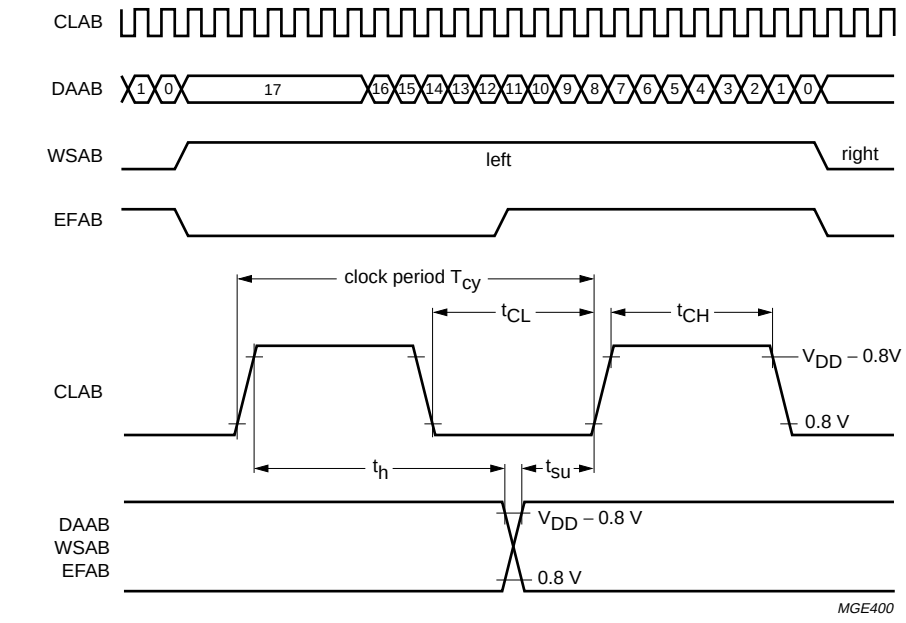


Fig.14 EIAJ timing diagram.

15.3 R-W timing (see Fig.15)

The data from sub-code R-W may be read via the V4 pin from the CD-decoder and has a format similar to RS232. The sub-code synchronization word is formatted by a pause of 200 µs minimum. Each sub-code byte starts with a logic 1 followed by seven bits (Q to W). The gap between bytes is variable between 1.3 and 90 µs.

15.4 C-flag timing (see Fig.16)

A 1-bit flag signal is input to the CFLAG pin. This signal shows the status of the error corrector and interpolator and is updated every frame.

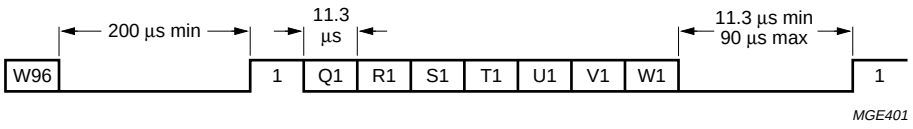


Fig.15 Sub-code formatting and timing from the V4 pin.

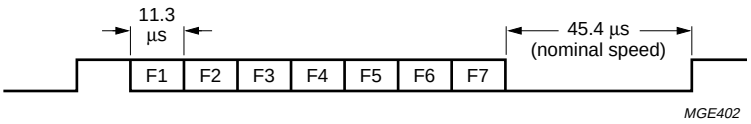


Fig.16 C-flag output timing.

15.5 S2B interface timing

The S2B serial interface consists of four lines (see Fig.17):

Transmit data (TXD)

Receive data (RXD)

Data path ready to accept data; active LOW ($\overline{CPR}$)

Basic engine ready to accept data; active LOW ($\overline{SPR}$).

These are used for communication. TXD and $\overline{CPR}$ for sending acknowledges and information data to the data path and RXD and $\overline{SPR}$ for receiving commands and parameters from the data path. The data is transferred frame-wise and asynchronously.

A data frame is proceeded by a start-bit (active LOW), followed by the actual data byte, and again followed by a parity bit (even parity), and a stop bit (active HIGH), see Fig.18. In total, eleven bits per frame are incorporated.

The interface is full duplex, meaning data frames may be transmitted and received simultaneously.

The bit-rate is selectable:

187.5 kbits/s with a 2.6% error

62.5 kbits/s with a 0.4% error

31.25 kbits/s with a 0.4% error.

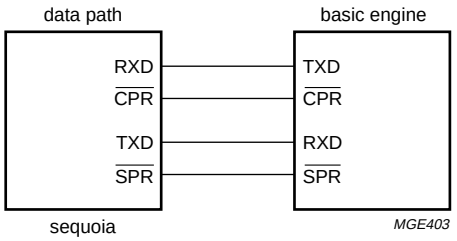


Fig.17 S2B interface.

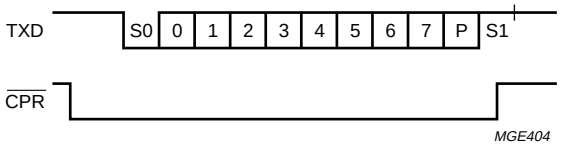


Fig.18 S2B Timing.

15.6 SCSI interface timing

V_{DD} = 4.75 to 5.25 V; V_{SS} = 0 V; T_{amb} = −10 to +70 °C; unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
Target asynchronous send; see Fig.19					
t_1	Data set-up time to $\overline{REQ}$ LOW		60	–	ns
t_2	$\overline{ACK}$ LOW to $\overline{REQ}$ HIGH		–	40	ns
t_3	Data hold time from $\overline{ACK}$ LOW	FIFO not empty	5	–	ns
t_4	$\overline{ACK}$ HIGH to $\overline{REQ}$ LOW	data already set-up	–	40	ns
Target asynchronous receive; see Fig.20					
t_1	$\overline{ACK}$ LOW to $\overline{REQ}$ HIGH		–	40	ns
t_2	$\overline{ACK}$ HIGH to $\overline{REQ}$ LOW	FIFO not full	–	40	ns
t_3	Data set-up time to $\overline{ACK}$ LOW		0	–	ns
t_4	Data hold time from $\overline{REQ}$ HIGH		0	–	ns
Fast SCSI-2 single-ended transfers (10 Mbytes/s)					
TARGET SYNCHRONOUS OUTPUT; see Fig.21					
t_1	$\overline{REQ}$ or $\overline{ACK}$ assertion period		35	–	ns
t_2	$\overline{REQ}$ or $\overline{ACK}$ negation period		35	–	ns
t_3	Data set-up time to $\overline{REQ}$ or $\overline{ACK}$ LOW		33	–	ns
t_4	Data hold time from $\overline{ACK}$ or $\overline{REQ}$ LOW		45	–	ns
TARGET SYNCHRONOUS INPUT; see Fig.22					
t_1	$\overline{REQ}$ or $\overline{ACK}$ assertion period		20	–	ns
t_2	$\overline{REQ}$ or $\overline{ACK}$ negation period		20	–	ns
t_3	Data set-up time to $\overline{REQ}$ or $\overline{ACK}$ LOW		0	–	ns
t_4	Data hold time from $\overline{ACK}$ or $\overline{REQ}$ LOW		10	–	ns

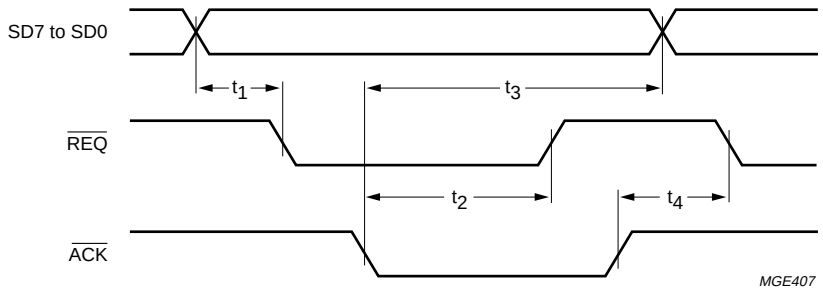


Fig.19 Target asynchronous send signal transitions.

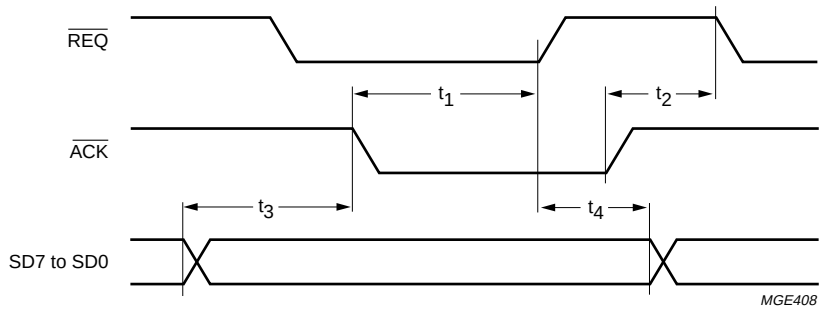


Fig.20 Target asynchronous receive signal transitions.

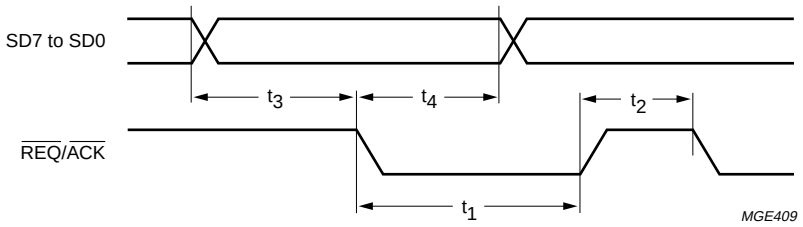


Fig.21 Target synchronous output.

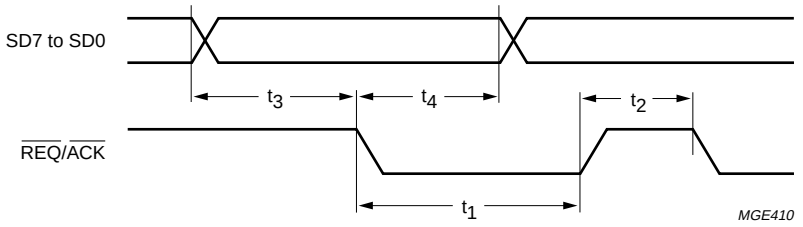


Fig.22 Target synchronous input.

15.7 Microprocessor interface

V_{DD} = 4.75 to 5.25 V; V_{SS} = 0 V; T_{amb} = −10 to +70 °C; unless otherwise specified.

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
Program memory fetch timing; see Fig.23				
t_{AVLL}	address valid to $\overline{ALE}$ LOW	15	–	ns
t_{LLAX}	address hold after $\overline{ALE}$ LOW	35	–	ns
t_{LLPL}	$\overline{ALE}$ LOW to $\overline{PSEN}$ LOW	25	–	ns
t_{PLPH}	$\overline{PSEN}$ pulse width	80	–	ns
t_{PLIV}	$\overline{PSEN}$ LOW to valid input instruction	–	65	ns
t_{PXIX}	Input instruction hold after $\overline{PSEN}$	0	–	ns
t_{PXIZ}	Input instruction float after $\overline{PSEN}$	–	30	ns
t_{AVIV}	address to valid input instruction	–	130	ns
t_{PLAZ}	$\overline{PSEN}$ low to address float	–	6	ns

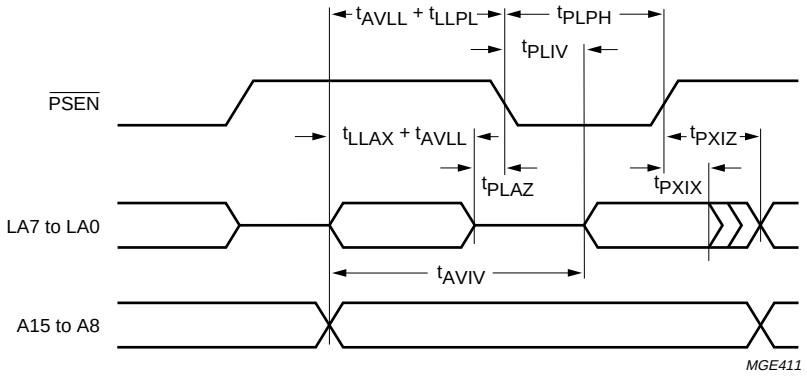


Fig.23 Program code fetch cycle.

15.8 DRAM interface (the SAA7385 is designed to operate with standard 70 ns DRAMs)

V_{DD} = 4.75 to 5.25 V; V_{SS} = 0 V; T_{amb} = −10 to +70 °C; unless otherwise specified.

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
DRAM interface timing; see Figs 24 to 28				
$t_{acc;CA}$	access time from column address	–	35	ns
$t_{hCA;RAS}$	column address hold time from $\overline{RAS}$	55	–	ns
$t_{su;CA}$	column address set-up time	0	–	ns
$t_{su;RA}$	row address set-up time	0	–	ns
$t_{acc;CAS}$	access time from $\overline{CAS}$	–	20	ns
$t_{h;CA}$	column address hold time	15	–	ns
$t_{W;CAS}$	$\overline{CAS}$ pulse width	20	10000	ns
$t_{h;CAS}$	$\overline{CAS}$ hold time (CBR refresh)	15	–	ns
t_{CASLZ}	$\overline{CAS}$ to output in low impedance	3	–	ns
t_{pCAS}	$\overline{CAS}$ precharge time	10	–	ns
$t_{acc;pCAS}$	access time from $\overline{CAS}$ precharge	–	40	ns
$t_{pCAS;RAS}$	$\overline{CAS}$ to $\overline{RAS}$ precharge time	5	–	ns
$t_{h;CAS}$	$\overline{CAS}$ hold time	70	–	ns
$t_{su;CAS}$	$\overline{CAS}$ set-up time (CBR refresh)	5	–	ns
t_{wCASL}	write command to $\overline{CAS}$ lead time	20	–	ns
$t_{h;DAT}$	data input hold time	15	–	ns
$t_{hDAT;RAS}$	data input hold time from $\overline{RAS}$	55	–	ns
$t_{su;DAT}$	data input set-up time	0	–	ns
$t_{d;OFF}$	output buffer turn off delay	3	20	ns
$t_{cy;FPR/W}$	fast page mode read or write cycle time	40	–	ns
$t_{cy;FPR-W}$	fast page mode read-write cycle time	note 1	–	ns
$t_{acc;RAS}$	access time from $\overline{RAS}$	–	70	ns
$t_{dRAS;CA}$	$\overline{RAS}$ to column address delay time	15	35	ns
$t_{h;RA}$	row address hold time	10	–	ns
$t_{W;RAS}$	$\overline{RAS}$ pulse width	70	10000	ns
$t_{W;RASFP}$	$\overline{RAS}$ pulse width (fast page mode)	70	100000	ns
$t_{CA;RASL}$	column address to $\overline{RAS}$ lead time	35	–	ns
$t_{cy;R/W}$	random read or write cycle time	130	–	ns
$t_{dRAS;CAS}$	$\overline{RAS}$ to $\overline{CAS}$ delay time	20	50	ns
$t_{su;R}$	read command set-up time	0	–	ns
$t_{hrR;CAS}$	read command hold time (referenced to $\overline{CAS}$)	0	–	ns
t_{REF}	refresh period	–	32	ns
t_{pRAS}	$\overline{RAS}$ precharge time	50	–	ns
$t_{pRAS;CAS}$	$\overline{RAS}$ to $\overline{CAS}$ precharge time	0	–	ns
$t_{hr;RAS}$	read command hold time (referenced to $\overline{RAS}$)	0	–	ns
$t_{h;RAS}$	$\overline{RAS}$ hold time	20	–	ns
$t_{cy;R-W}$	read-write cycle time	n/a[1]	–	ns

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
$t_{RASL;W}$	write command to $\overline{RAS}$ lead time	20	–	ns
t_{trans}	transition time (rise or fall)	3	50	ns
t_{hW}	write command hold time	15	–	ns
$t_{hW;RAS}$	write command hold time (referenced to $\overline{RAS}$)	55	–	ns
$t_{su;WE}$	$\overline{WE}$ command set-up time	0	–	ns
$t_{W;W}$	write command pulse width	15	–	ns
$t_{h;WE}$	$\overline{WE}$ hold time (CBR refresh)	10	–	ns
$t_{su;WE}$	$\overline{WE}$ set-up time (CBR refresh)	10	–	ns

Note

1. Not applicable.

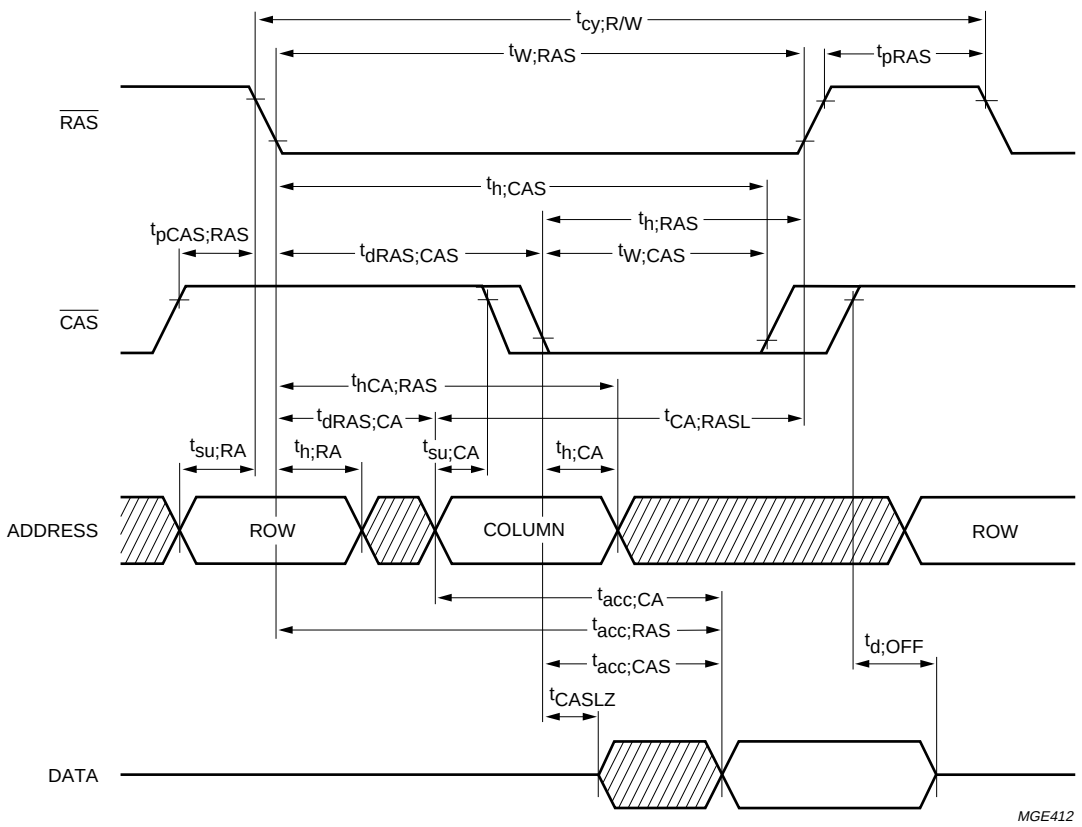


Fig.24 DRAM read cycle.

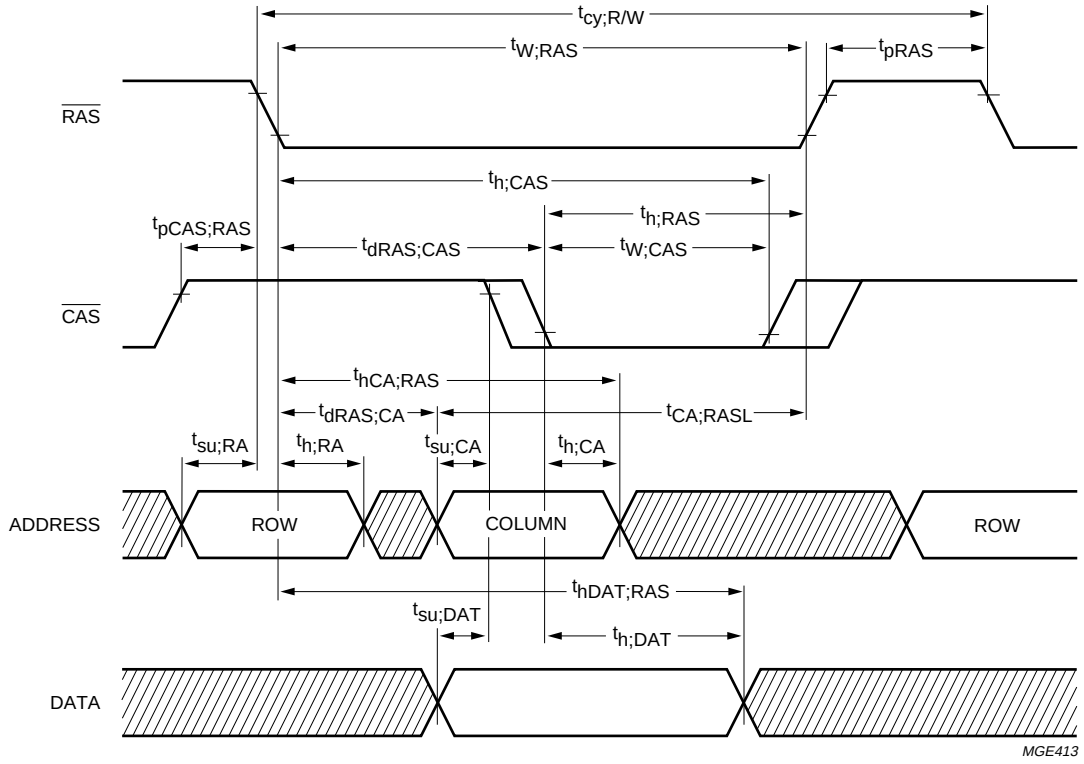


Fig.25 DRAM early write cycle.

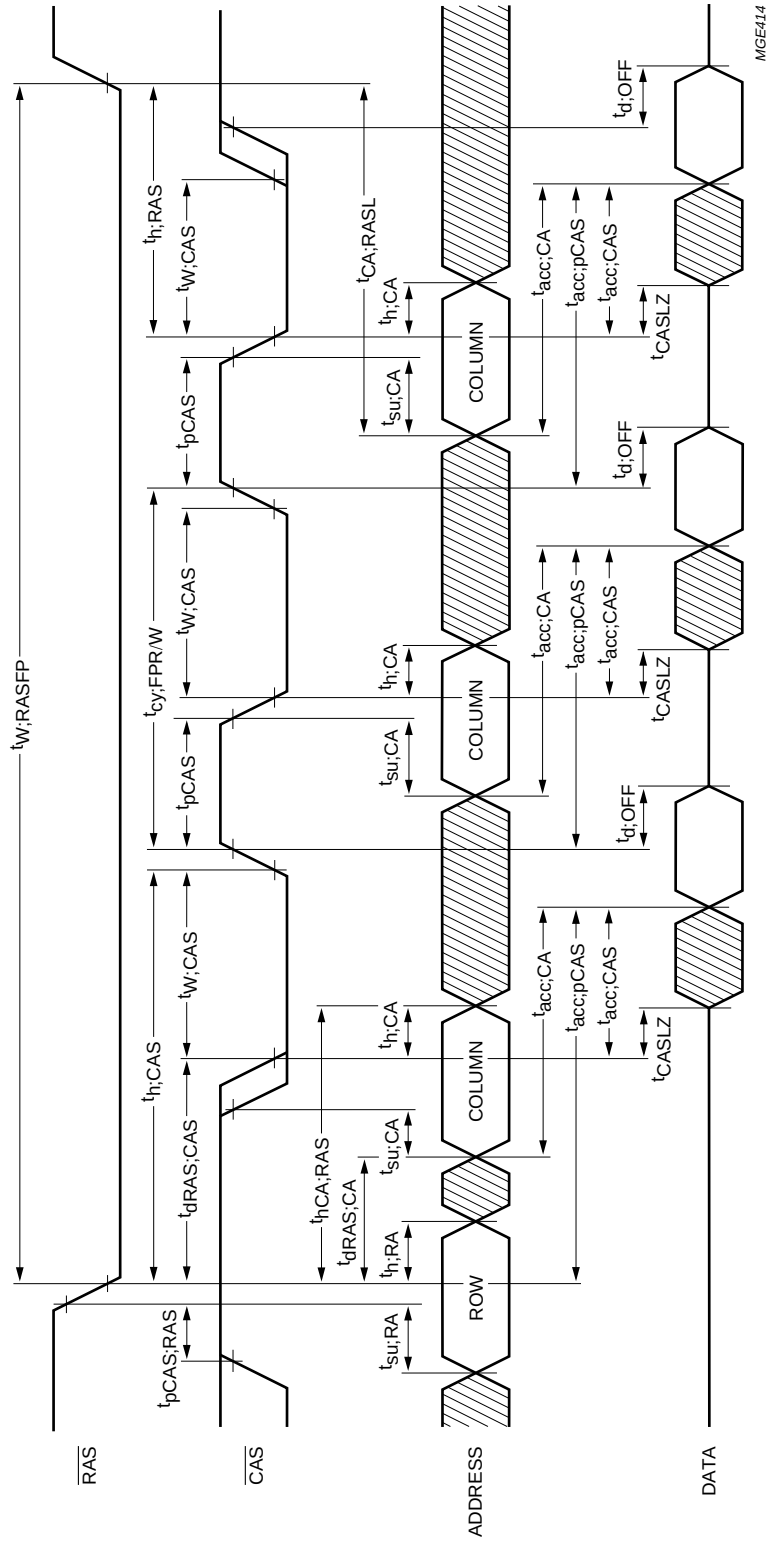


Fig.26 Fast page mode DRAM read cycle.

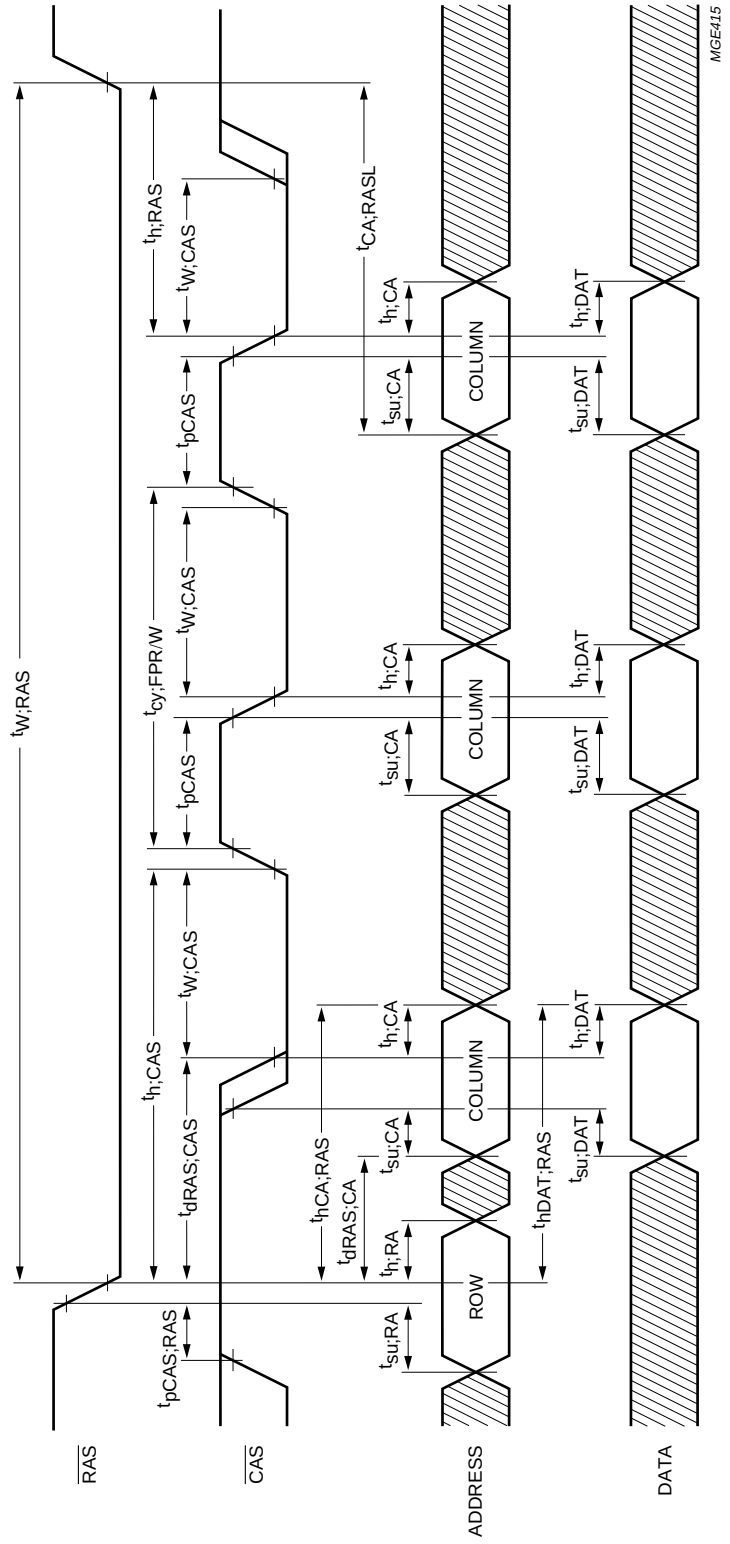


Fig.27 Fast page mode DRAM write cycle.

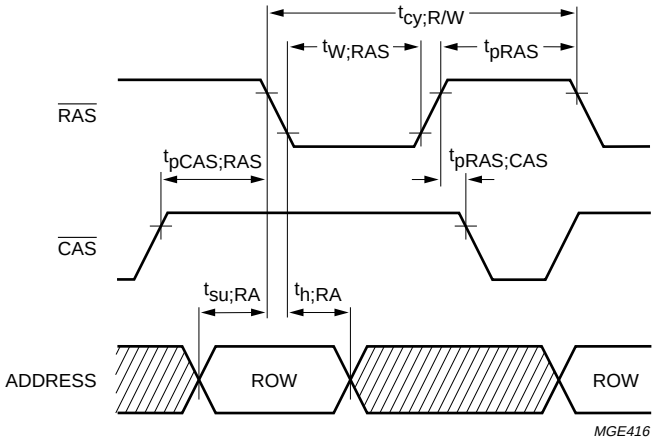
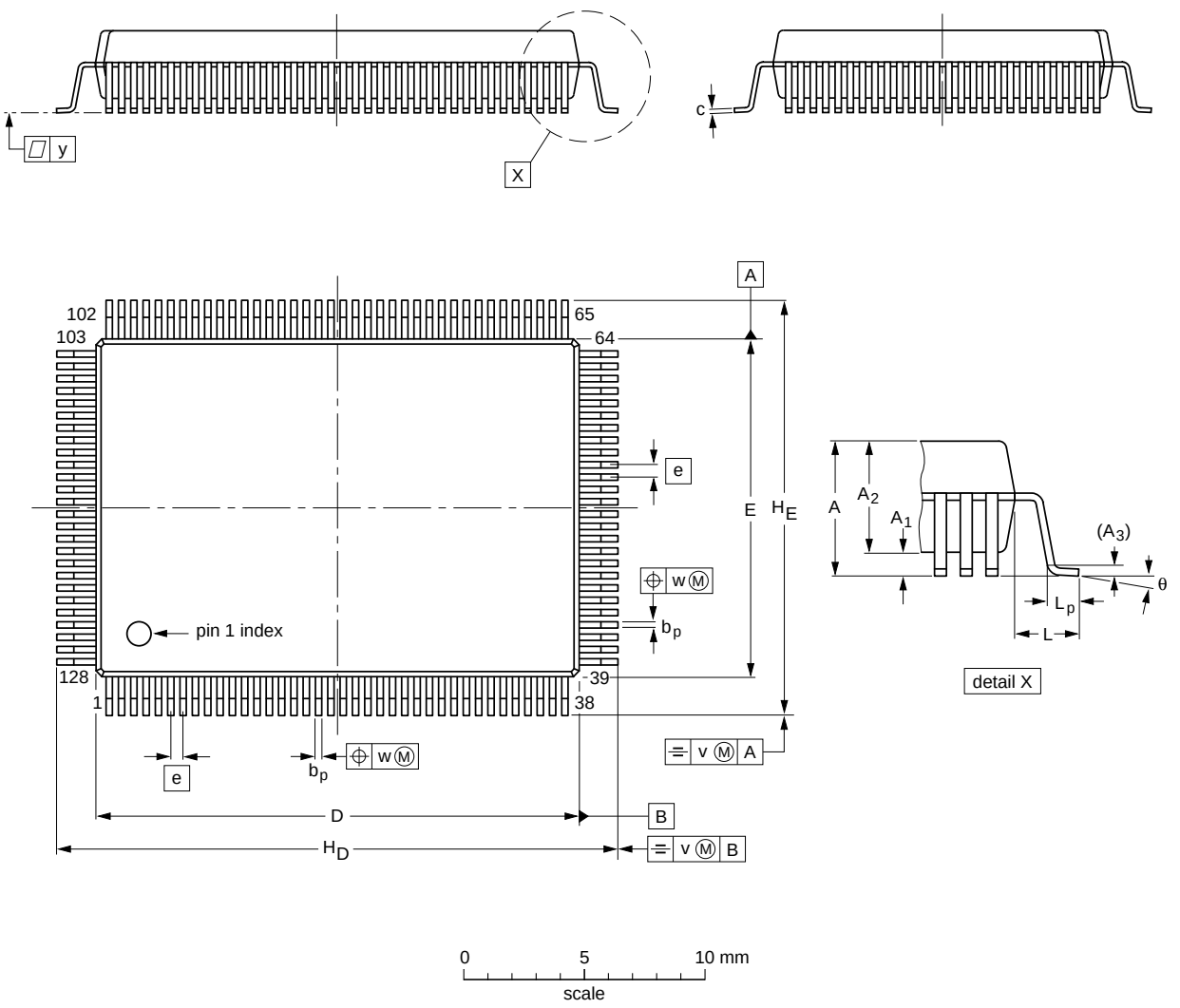


Fig.28 DRAM refresh cycle.

16 PACKAGE OUTLINE

SQFP128: plastic shrink quad flat package; 128 leads (lead length 1.6 mm); body 14 x 20 x 2.8 mm SOT387-2



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1 min.	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	v	w	y	θ
mm	3.40	0.25	3.05 2.55	0.25	0.27 0.17	0.20 0.09	20.0	14.0	0.50	23.2	17.2	1.60	0.95 0.65	0.20	0.08	0.10	7^o 0^o

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT387-2						96-03-14

17 SOLDERING

17.1 Introduction

There is no soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and surface mounted components are mixed on one printed-circuit board. However, wave soldering is not always suitable for surface mounted ICs, or for printed-circuits with high population densities. In these situations reflow soldering is often used.

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our *"IC Package Databook"* (order code 9398 652 90011).

17.2 Reflow soldering

Reflow soldering techniques are suitable for all QFP packages.

The choice of heating method may be influenced by larger plastic QFP packages (44 leads, or more). If infrared or vapour phase heating is used and the large packages are not absolutely dry (less than 0.1% moisture content by weight), vaporization of the small amount of moisture in them can cause cracking of the plastic body. For more information, refer to the Drypack chapter in our *"Quality Reference Handbook"* (order code 9397 750 00192).

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stencilling or pressure-syringe dispensing before package placement.

Several techniques exist for reflowing; for example, thermal conduction by heated belt. Dwell times vary between 50 and 300 seconds depending on heating method. Typical reflow temperatures range from 215 to 250 °C.

Preheating is necessary to dry the paste and evaporate the binding agent. Preheating duration: 45 minutes at 45 °C.

17.3 Wave soldering

Wave soldering is **not** recommended for QFP packages. This is because of the likelihood of solder bridging due to closely-spaced leads and the possibility of incomplete solder penetration in multi-lead devices.

If wave soldering cannot be avoided, the following conditions must be observed:

- **A double-wave (a turbulent wave with high upward pressure followed by a smooth laminar wave) soldering technique should be used.**
- **The footprint must be at an angle of 45° to the board direction and must incorporate solder thieves downstream and at the side corners.**

Even with these conditions, do not consider wave soldering the following packages: QFP52 (SOT379-1), QFP100 (SOT317-1), QFP100 (SOT317-2), QFP100 (SOT382-1) or QFP160 (SOT322-1).

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Maximum permissible solder temperature is 260 °C, and maximum duration of package immersion in solder is 10 seconds, if cooled to less than 150 °C within 6 seconds. Typical dwell time is 4 seconds at 250 °C.

A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

17.4 Repairing soldered joints

Fix the component by first soldering two diagonally-opposite end leads. Use only a low voltage soldering iron (less than 24 V) applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300 °C. When using a dedicated tool, all other leads can be soldered in one operation within 2 to 5 seconds between 270 and 320 °C.

18 DEFINITIONS

Data sheet status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

19 LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

Philips Semiconductors – a worldwide company

Argentina: see South America

Australia: 34 Waterloo Road, NORTH RYDE, NSW 2113,
Tel. +61 2 805 4455, Fax. +61 2 805 4466

Austria: Computerstr. 6, A-1101 WIEN, P.O. Box 213,
Tel. +43 1 60 101, Fax. +43 1 60 101 1210

Belarus: Hotel Minsk Business Center, Bld. 3, r. 1211, Volodarski Str. 6,
220050 MINSK, Tel. +375 172 200 733, Fax. +375 172 200 773

Belgium: see The Netherlands

Brazil: see South America

Bulgaria: Philips Bulgaria Ltd., Energoproject, 15th floor,
51 James Bourchier Blvd., 1407 SOFIA,
Tel. +359 2 689 211, Fax. +359 2 689 102

Canada: PHILIPS SEMICONDUCTORS/COMPONENTS,
Tel. +1 800 234 7381, Fax. +1 708 296 8556

China/Hong Kong: 501 Hong Kong Industrial Technology Centre,
72 Tat Chee Avenue, Kowloon Tong, HONG KONG,
Tel. +852 2319 7888, Fax. +852 2319 7700

Colombia: see South America

Czech Republic: see Austria

Denmark: Prags Boulevard 80, PB 1919, DK-2300 COPENHAGEN S,
Tel. +45 32 88 2636, Fax. +45 31 57 1949

Finland: Sinikalliontie 3, FIN-02630 ESPOO,
Tel. +358 615 800, Fax. +358 615 80920

France: 4 Rue du Port-aux-Vins, BP317, 92156 SURESNES Cedex,
Tel. +33 1 40 99 6161, Fax. +33 1 40 99 6427

Germany: Hammerbrookstraße 69, D-20097 HAMBURG,
Tel. +49 40 23 52 60, Fax. +49 40 23 536 300

Greece: No. 15, 25th March Street, GR 17778 TAVROS,
Tel. +30 1 4894 339/911, Fax. +30 1 4814 240

Hungary: see Austria

India: Philips INDIA Ltd, Shivsagar Estate, A Block, Dr. Annie Besant Rd.
Worli, MUMBAI 400 018, Tel. +91 22 4938 541, Fax. +91 22 4938 722

Indonesia: see Singapore

Ireland: Newstead, Clonskeagh, DUBLIN 14,
Tel. +353 1 7640 000, Fax. +353 1 7640 200

Israel: RAPAC Electronics, 7 Kehilat Saloniki St, TEL AVIV 61180,
Tel. +972 3 645 0444, Fax. +972 3 648 1007

Italy: PHILIPS SEMICONDUCTORS, Piazza IV Novembre 3,
20124 MILANO, Tel. +39 2 6752 2531, Fax. +39 2 6752 2557

Japan: Philips Bldg 13-37, Kohnan 2-chome, Minato-ku, TOKYO 108,
Tel. +81 3 3740 5130, Fax. +81 3 3740 5077

Korea: Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL,
Tel. +82 2 709 1412, Fax. +82 2 709 1415

Malaysia: No. 76 Jalan Universiti, 46200 PETALING JAYA, SELANGOR,
Tel. +60 3 750 5214, Fax. +60 3 757 4880

Mexico: 5900 Gateway East, Suite 200, EL PASO, TEXAS 79905,
Tel. +1 800 234 7381, Fax. +1 708 296 8556

Middle East: see Italy

Netherlands: Postbus 90050, 5600 PB EINDHOVEN, Bldg. VB,
Tel. +31 40 27 83749, Fax. +31 40 27 88399

New Zealand: 2 Wagener Place, C.P.O. Box 1041, AUCKLAND,
Tel. +64 9 849 4160, Fax. +64 9 849 7811

Norway: Box 1, Manglerud 0612, OSLO,
Tel. +47 22 74 8000, Fax. +47 22 74 8341

Philippines: Philips Semiconductors Philippines Inc.,
106 Valero St. Salcedo Village, P.O. Box 2108 MCC, MAKATI,
Metro MANILA, Tel. +63 2 816 6380, Fax. +63 2 817 3474

Poland: Ul. Lukiska 10, PL 04-123 WARSZAWA,
Tel. +48 22 612 2831, Fax. +48 22 612 2327

Portugal: see Spain

Romania: see Italy

Russia: Philips Russia, Ul. Usatcheva 35A, 119048 MOSCOW,
Tel. +7 095 926 5361, Fax. +7 095 564 8323

Singapore: Lorong 1, Toa Payoh, SINGAPORE 1231,
Tel. +65 350 2538, Fax. +65 251 6500

Slovakia: see Austria

Slovenia: see Italy

South Africa: S.A. PHILIPS Pty Ltd., 195-215 Main Road Martindale,
2092 JOHANNESBURG, P.O. Box 7430 Johannesburg 2000,
Tel. +27 11 470 5911, Fax. +27 11 470 5494

South America: Rua do Rocio 220 - 5th floor, Suite 51,
CEP: 04552-903-SÃO PAULO-SP, Brazil, P.O. Box 7383 (01064-970),
Tel. +55 11 821 2333, Fax. +55 11 829 1849

Spain: Balmes 22, 08007 BARCELONA,
Tel. +34 3 301 6312, Fax. +34 3 301 4107

Sweden: Kottbygatan 7, Akalla, S-16485 STOCKHOLM,
Tel. +46 8 632 2000, Fax. +46 8 632 2745

Switzerland: Allmendstrasse 140, CH-8027 ZÜRICH,
Tel. +41 1 488 2686, Fax. +41 1 481 7730

Taiwan: PHILIPS TAIWAN Ltd., 23-30F, 66,
Chung Hsiao West Road, Sec. 1, P.O. Box 22978,
TAIPEI 100, Tel. +886 2 382 4443, Fax. +886 2 382 4444

Thailand: PHILIPS ELECTRONICS (THAILAND) Ltd.,
209/2 Sanpavuth-Bangna Road Prakanong, BANGKOK 10260,
Tel. +66 2 745 4090, Fax. +66 2 398 0793

Turkey: Talatpasa Cad. No. 5, 80640 GÜLTEPE/ISTANBUL,
Tel. +90 212 279 2770, Fax. +90 212 282 6707

Ukraine: PHILIPS UKRAINE, 2A Akademika Koroleva str., Office 165,
252148 KIEV, Tel. +380 44 476 0297/1642, Fax. +380 44 476 6991

United Kingdom: Philips Semiconductors Ltd., 276 Bath Road, Hayes,
MIDDLESEX UB3 5BX, Tel. +44 181 730 5000, Fax. +44 181 754 8421

United States: 811 East Arques Avenue, SUNNYVALE, CA 94088-3409,
Tel. +1 800 234 7381, Fax. +1 708 296 8556

Uruguay: see South America

Vietnam: see Singapore

Yugoslavia: PHILIPS, Trg N. Pasica 5/v, 11000 BEOGRAD,
Tel. +381 11 825 344, Fax.+381 11 635 777

For all other countries apply to: Philips Semiconductors, Marketing & Sales Communications,
Building BE-p, P.O. Box 218, 5600 MD EINDHOVEN, The Netherlands, Fax. +31 40 27 24825

Internet: http://www.semiconductors.philips.com/ps/

Printed in The Netherlands 517021/50/01/pp64 Date of release: 1996 Jun 19 Document order number: 9397 750 00917

Let's make things better.

Philips Semiconductors

PHILIPS