查询CS89712供应商

CS89712

High-Performance, Low-Power System-on-Chip with 10BASE-T Ethernet Controller

Features

- ARM720T (ARM7 TDMI) processor
  - 8 Kbytes of four-way set-associative cache
  - MMU with 64-entry TLB
  - Write Buffer
  - Thumb code support enabled
- Dynamically clocked at 18, 36, 49 or 74 MHz
- 10 Mbit Ethernet Controller with integrated PHY
- Comprehensive Suite of Software Drivers
- On-Chip Transmit and Receive RAM Buffers
- 10BASE-T Port with Analog Filters provides automatic polarity detection and correction
- Programmable Transmit Features:
  - Automatic Re-transmission on Collision
  - Automatic Padding and CRC Generation
- Programmable Receive Features:
  - Early Interrupts for Frame Pre-Processing
  - Automatic Rejection of Erroneous Packets

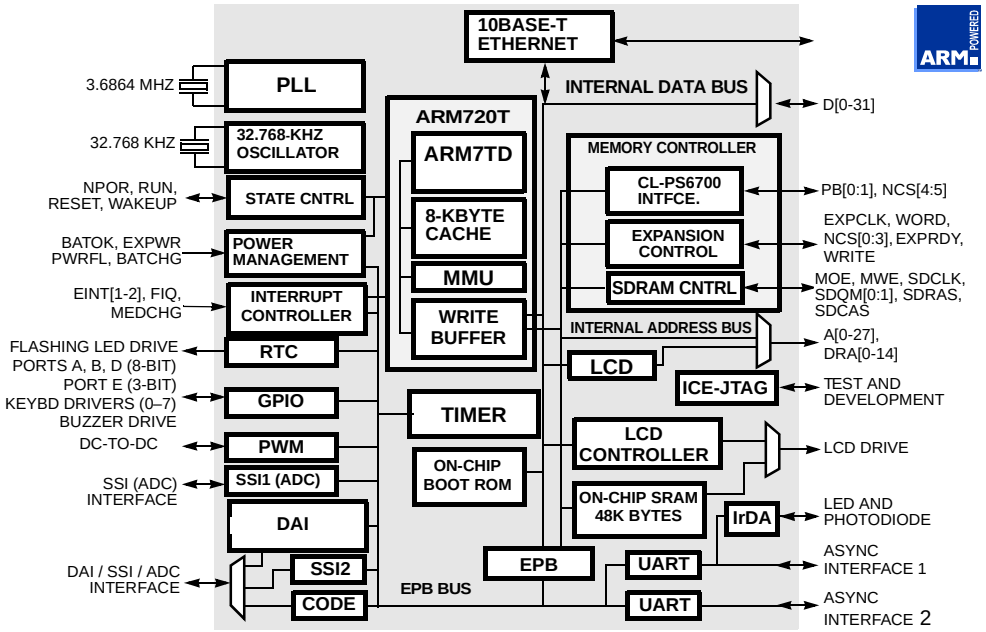
Description

The low-power high-performance CS89712 is designed for ultra-low-power communication applications such as VoIP telephones, industrial control, data acquisition, special purpose servers and RF to Ethernet bridges. The core-logic functionality of the device is built around an ARM720T processor with 8 Kbytes of four-way set-associative unified cache and a write buffer. Incorporated into the ARM720T is an enhanced memory management unit (MMU) which allows for support of sophisticated operating systems like embedded Linux.

The CS89712 Ethernet port includes on-chip RAM and 10BASE-T transmit and receive filters.

ORDERING INFO

CS89712-CB	0 to 70° C 256 Ball PBGA 17x17 mm



Preliminary Product Information | This document contains information for a new product. Cirrus Logic reserves the right to modify this product without notice.

CIRRUS LOGIC®
P.O. Box 17847, Austin, Texas 78760
(512) 445 7222 FAX: (512) 445 7581
http://www.cirrus.com

Copyright © Cirrus Logic, Inc. 2001
(All Rights Reserved)

Table Of Contents

1. OVERVIEW 4
2. FUNCTIONAL DESCRIPTION 7
2.1 CPU Core 7
2.2 State Control 7
2.3 Power-Up Sequence 10
2.4 Resets 10
2.5 Ethernet Port Reset and Initialization 11
2.6 Ethernet EEPROM Configurations 12
2.7 Clocks 16
2.8 Interrupt Controller 17
2.9 Boot ROM 21
2.10 Memory Map 21
2.11 Memory and I/O Expansion Interface 23
2.12 SDRAM Controller 23
2.13 SDRAM Initialization 26
2.14 CL-PS6700 PC Card Interface 26
2.15 Endianness 29
2.16 Internal UARTs and SIR Encoder 30
2.17 Synchronous Serial Interfaces 31
2.18 LCD Controller 39
2.19 Timer Counters 41
2.20 Real-Time Clock 43
2.21 Dedicated LED Flasher 43
2.22 PWM Interfaces 43
2.23 Ethernet Port Architecture 44
2.24 Ethernet Port Functional Description 45
2.25 Programming the EEPROM 46
2.26 Ethernet LEDs 48
2.27 Media Access Control Engine 48
2.28 Encoder/Decoder (ENDEC) 53
2.29 10BASE-T Transceiver 54
2.30 Basic Transmit Operation 56
2.31 Basic Receive Operation 56
2.32 Managing Interrupts & Status Queue 57
2.33 Basic Receive Operation 57
2.34 Receive Frame Address Filtering 63
2.35 Transmit Operation 66
2.36 Full Duplex Considerations 69
2.37 Auto-Negotiation Considerations 69

Contacting Cirrus Logic Support

For a complete listing of Direct Sales, Distributor, and Sales Representative contacts, visit the Cirrus Logic web site at:
http://www.cirrus.com/corporate/contacts/sales.cfm

Preliminary product information describes products which are in production, but for which full characterization data is not yet available. Advance product information describes products which are in development and subject to development changes. Cirrus Logic, Inc. has made best efforts to ensure that the information contained in this document is accurate and reliable. However, the information is subject to change without notice and is provided “AS IS” without warranty of any kind (express or implied). No responsibility is assumed by Cirrus Logic, Inc. for the use of this information, nor for infringements of patents or other rights of third parties. This document is the property of Cirrus Logic, Inc. and implies no license under patents, copyrights, trademarks, or trade secrets. No part of this publication may be copied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photographic, or otherwise) without the prior written consent of Cirrus Logic, Inc. Items from any Cirrus Logic website or disk may be printed for use by the user. However, no part of the printout or electronic files may be copied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photographic, or otherwise) without the prior written consent of Cirrus Logic, Inc.Furthermore, no part of this publication may be used as a basis for manufacture or sale of any items without the prior written consent of Cirrus Logic, Inc. The names of products of Cirrus Logic, Inc. or other vendors and suppliers appearing in this document may be trademarks or service marks of their respective owners which may be registered in some jurisdictions. A list of Cirrus Logic, Inc. trademarks and service marks can be found at http://www.cirrus.com.

3. REGISTER SET 70
3.1 Internal Registers 70
3.2 Accessing Ethernet Port Registers 73
3.3 Ethernet Port Internal Memory Map 77
3.4 I/O Port Data Registers 78
3.5 System Control Registers 79
3.6 Interrupt Registers 89
3.7 Expansion Memory Configuration Registers 92
3.8 Timer / Counter Registers 95
3.9 Miscellaneous Registers 95
3.10 UART Registers 98
3.11 LCD Registers 100
3.12 SSI Register 103
3.13 End Of Interrupt Locations 104
3.14 State Control Registers 105
3.15 SS2 Registers 106
3.16 DAI Registers 106
3.17 Ethernet Bus Interface Registers 117
3.18 Ethernet Port Status/Control Registers 117
4. TEST & DEBUG MODES 137
4.1 Entering test modes 137
4.2 Boundary Scan 139
4.3 In-Circuit Emulation 140
5. MECHANICAL INFORMATION 142
5.1 256-PBGA Pin Diagram 142
5.2 256-Ball PBGA Ball Listing 143
5.3 External Signal Functions 147
5.4 Output Bi-Directional Pins 151
5.5 256 PBGA Package Dimensions 153
6. ELECTRICAL/THERMAL INFO 154
6.1 Absolute Maximum Ratings 154
6.2 DC Characteristics 154
6.3 AC Characteristics 157
6.4 I/O Buffer Strength & Characteristics 169
7. ORDERING INFORMATION 170

1. Overview

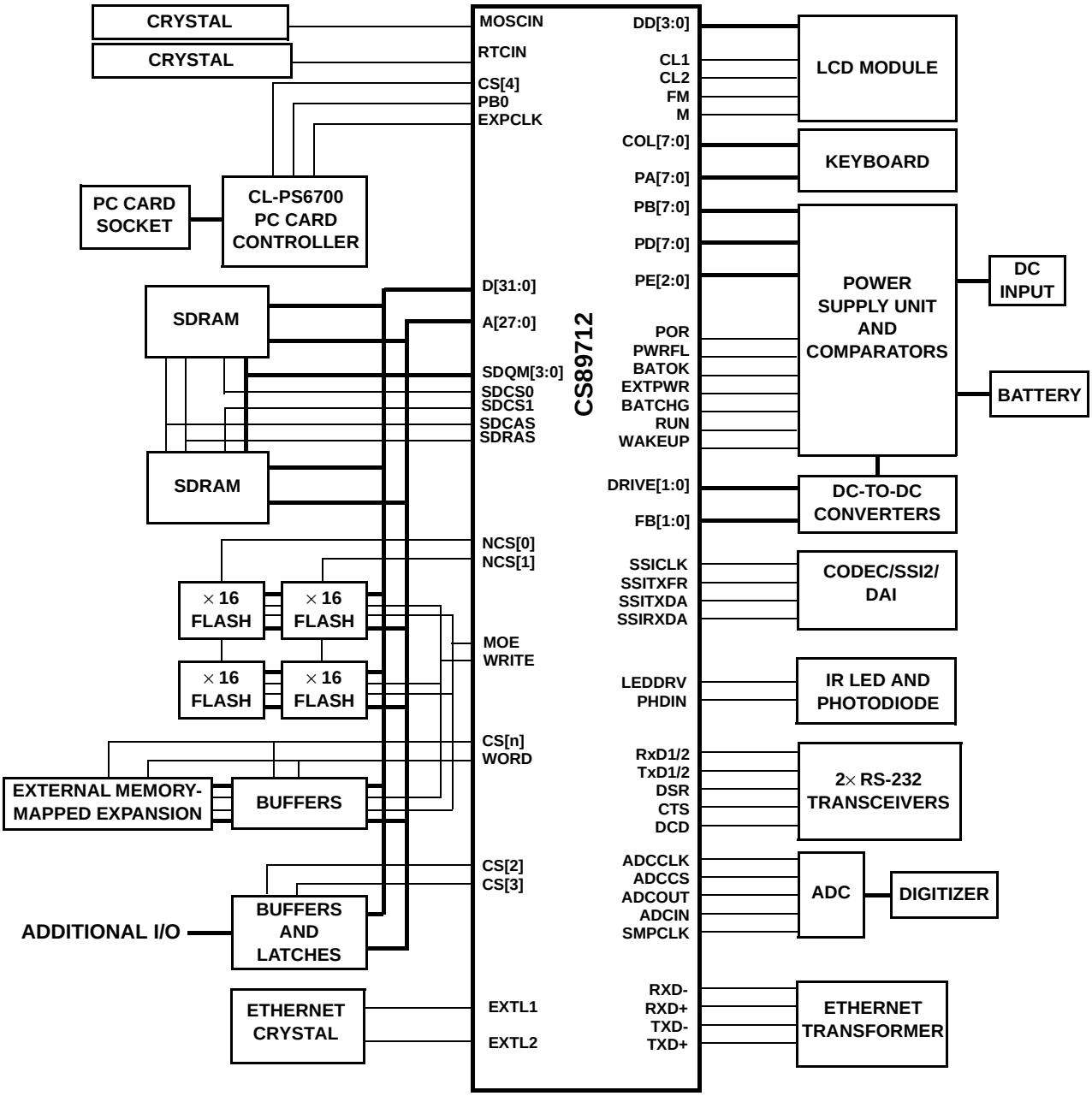


Figure 1. A CS89712–Based System

The CS89712 contains a single-chip embedded controller designed to be used in low-cost and ultra-low-power applications. Operating at 74 MHz, the CS89712 delivers about 66 Dhrystone 2.1 MIPS sustained (74 MIPS peak).

The CS89712 contains the following features:

- ARM720T processor with:
  - ARM7TDMI CPU core (supporting the Thumb instruction set and with enhanced multiplier) running at a dynamic clock speeds of 18, 36, 49, or 74 MHz
  - Advanced power management
  - Memory Management Unit compatible with the ARM710 core (and a 64-entry translation lookaside buffer) with added support for Windows CE
  - 8 kbytes of unified instruction/data cache with a four-way set associative controller
  - Write buffer
  - JTAG, core debug and full embedded ICE
- Full 10BaseT Ethernet port, with all the analog & digital circuitry needed for a complete Ethernet circuit, having:
  - Media Access Control (MAC) IEEE 802.3 compliant full-duplex engine. It handles all aspects of Ethernet frame transmission and reception, including: collision detection, preamble generation and detection, and CRC generation and test. Features include automatic retransmission on collision, and automatic padding of transmitted frames.
  - 4 kbyte page of on-chip memory, eliminating external memory chips.
  - serial EEPROM interface allowing configuration information storage for automatic load at power-up.
  - A Manchester encoder/decoder, clock recovery circuit, and 10BASE-T transceiver. It provides on-chip LED drivers for link status, bus status, and Ethernet line activity.
  - 10BASE-T transceiver including drivers, receivers, and analog filters, for direct connection to low-cost isolation transformers.
  - very low noise emission, shortening EMI testing and qualification time.
- 48k bytes of on-chip SRAM sharable between the LCD controller and general applications
- Low power operation. Typical power dissipated is 270 mW at 74 MHz in the Operating State and 160 mW in the Idle State (clock to the CPU stopped, everything else running), with <150 uW in the Standby State (realtime clock ‘on’, everything else stopped). The Ethernet block has a Software Suspend state, disabling the receiver and dropping current to the microampere range.
- Advanced audio decoder / decompression supports multiple audio decompression algorithms at all standard sample & bit rates. MPEG 1, 2, and 2.5 layer 3 audio decoding is supported, including ISO compliant MPEG 1 and 2 layer 3 support. Adaptive bit rates are supported.
- Up to 64 MHz of SDRAM can operate at up to 36.864 MHz with 16- or 32-bit wide accesses.
- ROM / SRAM / FLASH Memory controller decodes up to 5 separate memory segments each up to 256 Mbytes. Each segment can be configured as 8, 16, or 32 bits wide with page-mode access support and programmable access times. Supports removable FLASH card interface for addition of expansion FLASH modules.
- 27 general-purpose I/O bits; three 8-bit and one 3-bit port support scanning keyboard matrix.
- Digital Audio Interface (DAI) for interfacing to CD-quality DACs and CODECs.
- Interrupt controller.
- IrDA 115.2 kbps SIR protocol controller.

- LCD controller interfaces directly to a single-scan panel monochrome LCD. Panel width size is programmable from 32 to 1024 pixels in 16-pixel increments. Video frame buffer size programmable up to 128 kbytes with 1, 2, or 4 bits per pixel supports 15-level grayscale operation.
- Programmable frame buffer address allows a system with only internal SRAM for memory.
- On-chip boot ROM programmed with serial load boot sequence.
- Two 16-bit general purpose timer counters.
- 32-bit Real-Time Clock and comparator.
- Dedicated LED flasher pin driven from the RTC with programmable duty ratio (multiplexed with a GPIO pin).
- Two 16550 type UARTs:
  - support bit rates up to 115.2 kbps
  - contain two 16-byte FIFOs for TX and RX
  - UART1 supports modem control signals
- Two synchronous serial interfaces for Microwire (128 kbps) or SPI peripherals such as ADCs, one supporting both master/slave mode and the other supporting master mode only.
- PWM interface provides two 96 kHz clocks with programmable 1/16 to 15/16 duty cycle for driving a DC to DC converter.
- An interface to one or two Cirrus Logic CL-PS6700 PC Card controller devices to support two PC Card slots.
- Oscillator and phase-locked loop (PLL) to generate the core clock speeds of 18.432 MHz, 36.864 MHz, 49.152 MHz, and 73.728 MHz from an external 3.6864 MHz crystal.
- A low-power 32.768 kHz oscillator.
- Suite of software drivers for immediate use with most industry standard network operating systems. In addition, complete evaluation kits and manufacturing packages significantly reduce production cost and time.
- Commercial 0 - 70C operating temperature.

The CS89712 design is optimized for low power dissipation and is fabricated on a fully static 0.25 micron CMOS process. It is available in a 256-ball PBGA package.

A maximum configured system using the CS89712 is shown in Figure 1. This system assumes all of the DRAMs and ROMs are 16-bit wide devices. The keyboard may be connected to more GPIO bits than shown to allow greater than 64 keys, however these extra pins will not be wired into the WAKEUP pin functionality. Note that only one of the CODEC, SSI2, or DAI interfaces may be used at a time.

2. FUNCTIONAL DESCRIPTION

2.1 CPU Core

The ARM720T consists of an ARM7TDMI 32-bit RISC processor, a unified 8 kbyte cache, and a memory management unit (MMU). The cache is four-way set associative organized as 512 lines with each line being 16 bytes. The cache is directly connected to the ARM7TDMI, and therefore caches the virtual address from the CPU. When the cache misses, the MMU translates the virtual address into a physical address. A 64-entry translation lookaside buffer (TLB) is utilized to speed the address translation process and reduce bus traffic necessary to read the page table. The MMU saves power by only translating cache misses.

See the ARM720T Data sheet for a complete description of the various logic blocks that make up the processor, as well as all internal registers.

2.2 State Control

The CS89712 supports the following Power Management States: Operating, Idle, and Standby (see Figure 2). There is also a state called the Doze State, however it is a temporary execution state. The normal program execution state is the Operating State, which is a full performance state where all of the clocks and peripheral logic are enabled. The Idle State is the same as the Operating State with the exception of the CPU clock being halted, and only an external interrupt will return it back to the Operating State. The Standby State has the lowest power consumption of the three states. By selecting this mode the main oscillator shuts down, leaving only the Real-Time Clock and its associated logic powered. When the CS89712 is in Standby all power and ground pins should remain connected to power and ground in order to have a proper system wake-up. The only state that Standby can transition to is the Operating State.

2.2.1 Standby State

The Standby State equates to the system being switched "off" (i.e., no display, and the main oscillator is shut down). The PLL will be shut down.

In the Standby State, all the system memory and state is maintained and the system time is kept up-to-date. The PLL/on-chip oscillator or external oscillator is disabled and the system is static, except for the low-power watch crystal (32 kHz) oscillator and divider chain to the RTC and LED flasher. The RUN signal is driven low, therefore this signal can be used externally in the system to power down other system modules.

Whenever the CS89712 is in the Standby State, the external address and data buses are forced low internally by the RUN signal. This is done to prevent peripherals that are powered down from draining

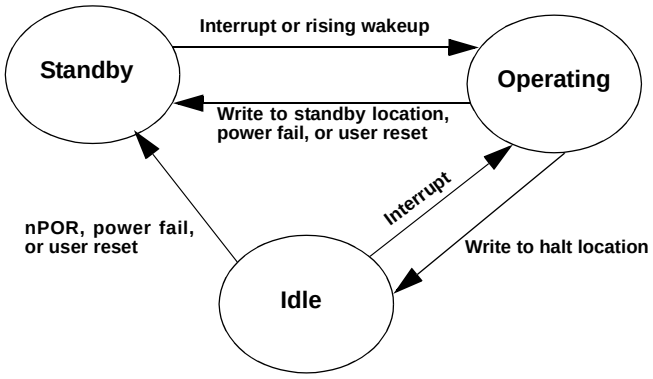


Figure 2. State Diagram

Address (W/B)	Operating	Idle	Standby	nPOR RESET	nURESET RESET
SDRAM Control	On	On	SELFREF	Off	N/A
UARTs	On	On	Off	Reset	Reset
LCD FIFO	On	On	Reset	Reset	Reset
LCD	On	On	Off	Reset	Reset
ADC Interface	On	On	Off	Reset	Reset
SSI2 Interface	On	On	Off	Reset	Reset
DAI Interface	On	On	Off	Reset	Reset
CODEC	On	On	Off	Reset	Reset
Timers	On	On	Off	Reset	Reset
RTC	On	On	On	On	On
LED Flasher	On	On	On	Reset	Reset
DC-to-DC	On	On	Off	Reset	Reset
CPU	On	Off	Off	Reset	Reset
Interrupt Control	On	On	On	Reset	Reset
PLL/CLKEN Signal	On	On	Off	Off	Off

Table 1. Peripheral Status in Different Power Management States

current. Also, the internal peripheral's signals get set to their Reset State.

When first powered, or reset by the nPOR (Power On Reset, active low) signal, the CS89712 is forced into the Standby State. This is known as a cold reset, and when leaving the Standby State after a cold reset, external wake up is the only way to wake up the device. When leaving the Standby State after non-cold reset conditions (i.e., the software has forced the device into the Standby State), the transition to the Operating State can be caused by a rising edge on the WAKEUP input signal or by an enabled interrupt. Normally, when entering the Standby State from the Operating State, the software will leave some interrupt sources enabled.

Note: The CPU cannot be awakened by the TINT, WEINT, and BLINT interrupts when in the Standby State.

Typically, software writes to the Standby internal memory location to cause the transition from the Operating State to the Standby State. Before entering the Standby State, if external I/O devices (such as the CL-PS6700s connected to nCS[4] or nCS[5]) are in use, the software must ensure that they are idle before writing to the Standby State location.

Before entering the Standby State, the software must properly disable the DAI. Failing to do so will result in higher than expected power consumption in the Standby State, as well as unpredictable operation of the DAI. The DAI can be re-enabled after transitioning back to the Operating State.

The system can also be forced into the Standby State by hardware if the nPWRFL or nURESET inputs are forced low. The only exit from the Standby State is to the Operating State.

The system will only transition to the Operating State from the Standby State under the following conditions: when the nPWRFL input pin is high, when the nEXTPWR input pin is low, or when the BATOK input pin is high. This prevents the system from starting when the power supply is inadequate

(i.e., the main batteries are low), corresponding to a low level on nPWRFL or BATOK.

From the Standby State, if the WAKEUP signal is applied with no clock except the 32 kHz clock running, the CS89712 will be initialized into a state where it is ready to start and is waiting for the CPU to start receiving its clock. The CPU will still be held in reset at this point. After the first clock is applied, there will be a delay of about eight clock cycles before the CPU is enabled. This delay allows the CPU clock to settle.

2.2.1.1 UART in Standby State

During the Standby State, the UARTs are disabled and cannot detect any activity (i.e., start bit) on the receiver. If this functionality is required then this can be accomplished in software by the following method:

1) Permanently connect the RX pin to one of the active low external interrupt pins.
2) Ensure that on entry to the Standby State, the chosen interrupt source is not masked, and the UART is enabled.
3) Send a preamble that consists of one start bit, 8 bits of zero, and one stop bit. This will cause the CS89712 to wake and execute the enabled interrupt vector.

The UART will automatically be re-enabled when the processor re-enters the Operating State, and the preamble will be received. Since the UART was not awake at the start of the preamble, the timing of the sample point will be off-center during the preamble byte. However, the next byte transmitted will be correctly aligned. Thus, the actual first real byte to be received by the UART will be correct.

2.2.2 Idle State

If in the Operating State, the Idle State can be entered by writing to a special internal memory location (HALT) in the CS89712. If an interrupt occurs, the CS89712 will return immediately back to the Operating State and execute the next instruction. The WAKEUP signal can not be used to exit the Idle State. It is only used to exit the Standby State.

In the Idle State, the device functions just like it does when in the Operating State. However, the CPU clock is halted while it waits for an event such as a key press to generate an interrupt. The PLL always remains active in the Idle State.

2.2.3 Keyboard Interrupt Wakeup

For the case of the keyboard interrupt, the following options are available and are selectable according to bits 1 and 3 of the SYSCON2 register (refer to Section 3.5.2 for register details).

- If the KBWEN bit (SYSCON2 bit 3) is set low, then a keypress will cause a transition from a power saving state only if the keyboard interrupt is non-masked (i.e., the interrupt mask register 2 (INTMR2 bit 0) is high).
- When KBWEN is high, a keypress will cause the device to wake up regardless of the state of the interrupt mask register. This is called the “Keyboard Direct Wakeup’ mode. In this mode, the interrupt request may not get serviced. If the interrupt is masked (i.e., the interrupt mask register 2 (INTMR2 bit 0) is low), the processor simply starts re-executing code from where it left off before it entered the power saving state. If the interrupt is non-masked, then the processor will service the interrupt.
- When the KBD6 bit (SYSCON2 bit 1) is low, all 8 Port A inputs are OR’ed together to produce the internal wakeup signal and keyboard interrupt request. This is the default reset state.
- When the KBD6 bit (SYSCON2 bit 1) is high, only the lowest 6 bits of Port A are OR’ed together to produce the internal wakeup signal and keyboard interrupt request. The two most significant bits of Port A are available as GPIO when this bit is set high.

When both KBWEN and INTMR2 bit 0 are low, the device can be awakened only by the external WAKEUP pin or another enabled interrupt source. The keyboard interrupt capability allows use of a polled and/or interrupt-driven keyboard routine.

Notes:The keyboard interrupt is NOT deglitched.

2.2.4 Ethernet Port Software Suspend

The Ethernet port power features work in a different manner than detailed above. Suspend mode-may be entered via software. During this mode, all internal Ethernet circuits are shut off except the I/O Base Address register (Ethernet Port offset address 0020h) and the SelfCTL register.

To enter Suspend mode, the SWSuspend bit (SelfCTL Register, bit 8) is set. To exit SW Suspend, software must write to the CS89712 Ethernet (used only to wake the Ethernet port, the Write data is ignored). Upon exit, the CS89712 Ethernet performs a complete reset, and then goes through a normal initialization procedure.

2.3 Power-Up Sequence

The following sequence should be followed to ensure proper start up. If any of the timing sequences recommended below are violated, then the part may not start up properly, requiring a hard reset to recover.

1) Upon power, the signal nPOR must be held active (LOW) for a minimum of 100us, after V_{DD} has become settled.

2) After nPOR goes HIGH, the CS89712 will enter the Standby State (and only this state). In this state, the PLL and CPU are not enabled. The only method that can be used to allow the CS89712 to exit the Standby State into the Operating State is by the WAKEUP signal going active (HIGH).

   Note: It is not a requirement to use the nURESET signal. If not used, the nURESET signal must be HIGH, and it must have gone HIGH prior to nPOR going HIGH. This is due to the fact that nURESET is latched into the device by the rising edge of nPOR. When nURESET is LOW on the rising edge of nPOR, it can force the device into one of its Test Mode states.

3) After nPOR goes HIGH, the WAKEUP signal cannot be detected as going HIGH, until after at least two seconds. After two seconds, the WAKEUP signal can become active, and it must be HIGH for at least 125 us.

4) Before the WAKEUP signal is detected internally, it must go through a deglitching circuit. This is why is must be active for at least 125us. Then the PLL gets enabled. WAKEUP is ignored immediately after waking up the system. It also ignores it while in the Idle or Operating State. It can constantly toggle with no affect on the device. It will only be read again if nPOR goes low and then high again, or if software has forced the device back into the Standby State.

5) A maximum of 250 msec will pass before the CPU starts to fetch the first instruction.

2.4 Resets

There are three asynchronous resets to the CS89712: nPOR (Power On Reset), nPWRFL, and nURESET. If any of these are active, a system reset is generated internally. This will reset all internal registers in the CS89712 except the RTC data and match registers. These registers are only cleared by nPOR allowing the system time to be preserved through a user reset or power fail condition.

NOTE: The Ethernet Port has different reset conditions and considerations than described in this section. Refer to the following section for resetting the Ethernet Port.

Any reset will also reset the CPU and cause it to start execution at the reset vector when the CS89712 returns to the Operating State.

Three signals are used to internally reset storage elements. These are nPOR, nSYSRES (System Reset) and nSTBY. nPOR is an external signal. nSTBY is equivalent to the external RUN signal.

nPOR is the highest priority reset signal. When active (low), it will reset all storage elements in the CS89712. nPOR active forces nSYSRES and nSTBY active. nPOR will only be active after the CS89712 is first powered up and not during any other resets. nPOR active clears all flags in the status register except for the cold reset flag (CLDFLG) bit (SYSFLG, bit 15), which is set.

nSYSRES is generated internally in the CS89712 if either nPOR, nPWRFL, or nURESET are active. It is the second highest priority reset signal, used to asynchronously reset most internal registers. nSYSRES activation forces nSTBY and RUN low, and resets the CS89712 leaving it in the Standby State.

The nSTBY and RUN signals are high when the CS89712 is in the Operating or Idle States and low when in the Standby State. The main system clock is valid when nSTBY is high. The nSTBY signal will disable any peripheral block that is clocked from the master clock source (i.e., everything except for the RTC). However, when in Snooze State, the LCD controller and the DC to DC converter interface peripherals will NOT be disabled.

In general, a system reset will clear all registers and nSTBY will disable all peripherals that require a main clock, with the exception of the Snooze State operation as described above. The following peripherals are always disabled by a low level on nSTBY: two UARTs and IrDA SIR encoder, timer counters, telephony codec, and the two SSI interfaces. In addition, when in the Standby State, the LCD controller and PWM drive are also disabled.

2.5 Ethernet Port Reset and Initialization

Different considerations apply to resetting and initializing the Ethernet Port.

2.5.1 Reset

Three different conditions cause the Ethernet port to reset its Ethernet internal registers and circuits.

2.5.1.1 Power-Up Reset

When power is applied, the Ethernet port maintains reset until the voltage at the supply pins reaches approximately 2.5 V. The Ethernet port comes out of reset once Vcc is greater than approximately 2.5 V and the crystal oscillator has stabilized.

2.5.1.2 Software Initiated Reset

There is a chip-wide reset whenever the RESET bit (SelfCTL Register, Bit 6) is set.

2.5.1.3 Software Suspend

Whenever the Ethernet port enters Software Suspend mode, all registers and circuits are reset. Upon exit, there is a chip-wide reset.

2.5.2 Allowing Time for Reset Operation

After a reset, the Ethernet port goes through a self configuration. This includes calibrating on-chip analog circuitry, and reading EEPROM for validity and configuration. Time required for the reset calibration is typically 10 ms. Software drivers should not access registers internal to the CS89712 Ethernet during this time. When calibration is done, bit INITD in the Self Status Register is set indicating that initialization is complete, and the SIBUSY bit in the same register is cleared indicating the EEPROM is no longer being read.

2.5.3 Initialization

After each reset (except EEPROM Reset), the CS89712's Ethernet port checks the sense of the EEDataIn pin to see if an external EEPROM is present. EEDI high indicates presence of an EEPROM and the Ethernet port automatically loads the configuration data stored in the EEPROM into its internal registers (see next section). If EEDI is low, an EEPROM is not present and the Ethernet port resets with the register values in Table 2.

An optional low-cost serial EEPROM can be used to store configuration information that is automati-

cally loaded into the Ethernet port after each reset (except EEPROM reset).

The CS89712 Ethernet operates with any of six standard EEPROMs shown in Table 3.

Ethernet Port Address	Register Contents	Register Descriptions
0020h	0300h	I/O Base Address
0022h	XXXX XXXX XXXX X100	Interrupt Number
0102h	0003h	RxCFG Register
0104h	0005h	RxCTL Register
0106h	0007h	TxCFG Register
0108h	0009h	TxCMD Register
010Ah	000Bh	BufCFG Register
010Ch	Undefined	Reserved
010Eh	Undefined	Reserved
0110h	Undefined	Reserved
0112h	00013h	LineCTL Register
0114h	0015h	SelfCTL Register
0116h	0017h	BusCTL Register
0118h	0019h	TestCTL Register

Table 2. Default Configuration

Note: I/O base address is unaffected by Software Suspend mode.

EEPROM Type	Size (16-bit words)
‘C46 (non-sequential)	64
‘CS46 (sequential)	64
‘C56 (non-sequential)	128
‘CS56 (sequential)	128
‘C66 (non-sequential)	256
‘CS66 (sequential)	256

Table 3. Supported EEPROM Types

2.6 Ethernet EEPROM Configurations

2.6.1 EEPROM Interface

The EEPROM interface uses the signals shown in Table 4.

Ethernet port Pin	Ethernet port Function	EEPROM Pin
EECS	EEPROM Chip Select	Chip Select
EESK	1 MHz EEPROM Serial Clock output	Clock
EEDO	EEPROM Data Out (data to EEPROM)	Data In
EEDI	EEPROM Data in (data from EEPROM)	Data Out

Table 4. EEPROM Interface

2.6.2 EEPROM Memory Organization

If an EEPROM is used to store initial configuration information for the Ethernet port, the EEPROM is organized in one or more blocks of 16-bit words. The first block in EEPROM, referred to as the Configuration Block, is used to configure the Ethernet port after reset. An example of a typical Configuration Block is shown in Table 5. Additional blocks containing user data may be stored in the EEPROM. However, the Configuration Block must always start at address 00h and be stored in contiguous memory locations.

2.6.3 Reset Configuration Block

The first block in EEPROM, the Reset Configuration Block, is used to automatically program the Ethernet port with an initial configuration after a reset. Additional user data may also be stored in the EEPROM if space is available. The additional data are stored as 16-bit words and can occupy any EEPROM address space beginning immediately after the end of the Reset Configuration Block up to address 7Fh, depending on EEPROM size. This additional data can only be accessed through software control (refer to Section 2.24 for more information). Address space 80h to AFh is reserved.

Word Address	Value	Description
FIRST WORD in DATA BLOCK		
00h	A120h	Configuration Block Header. The high byte, A1h, indicates a ‘C46 EEPROM is attached. The Link Byte, 20h, indicates the number of bytes to be used in this block of configuration data.
FIRST GROUP of WORDS		
01h	2020h	Group Header for first group of words. Three words to be loaded, beginning at 0020h in Ethernet Port memory.
02h	0300h	I/O Base Address
03h	0003h	Interrupt Number
04h	0001h	DMA Channel Number
SECOND GROUP of WORDS		
05h	502Ch	Group Header for second group of words. Six words to be loaded, beginning at 002Ch in Ethernet Port memory.
06h	E000h	Memory Base Address - low word
07	000Fh	Memory Base Address - high word
08h	0000h	Boot PROM Base Address - low word
09h	000Dh	Boot PROM Base Address - high word
0Ah	C000h	Boot PROM Address Mask - low word
0Bh	000Fh	Boot PROM Address Mask - high word
THIRD GROUP of WORDS		
0Ch	2158h	Group Header for third group of words. Three words to be loaded, beginning at 0158 in Ethernet Port memory.
0Dh	0010h	Individual Address - Octet 0 and 1
0Eh	0000h	Individual Address - Octet 2 and 3
0Fh	0000h	Individual Address - Octet 4 and 5
CHECKSUM Value		
10h	2800h	The high byte, 28h, is the Checksum Value. In this example, the checksum includes word addresses 00h through 0Fh. The hexadecimal sum of the bytes is D8h, resulting in a 2’s complement of 28h. The low byte, 00h, provides a pad to the word boundary.

Note: FFFFh is a special code indicating that there are no more words in the EEPROM.

Table 5. EEPROM Configuration Block Example

2.6.3.1 Reset Configuration Block Structure

The Reset Configuration Block is a block of contiguous 16-bit words starting at EEPROM address 00h. It can be divided into three logical sections: a header, one or more groups of configuration data words, and a checksum value. All words in the Reset Configuration Block are read sequentially by the Ethernet port after each reset, starting with the header and ending with the checksum. Each group of configuration data is used to program an Ether-

net Port register (or set of Ethernet Port registers in some cases) with an initial non-default value.

2.6.3.2 Reset Configuration Block Header

The header (first word of the block located at EEPROM address 00h) specifies the type of EEPROM used, if a Reset Configuration block is present, and if so, how many bytes of configuration data are stored in the Reset Configuration Block.

2.6.3.3 Determining the EEPROM Type

The LSB of the high byte of the header indicates the type of EEPROM attached: sequential or non-sequential. An LSB of 0 (XXXX-XXX0) indicates a sequential EEPROM, with a 1 (XXXX-XXX1) indicating non-sequential EEPROM. The Ethernet port functions with either type of EEPROM. The Ethernet port will automatically generate sequential addresses while reading the Reset Configuration Block if a non-sequential EEPROM is used.

2.6.3.4 EEPROM Reset Configuration Block

The read-out of either a binary 101X-XXX0 or 101X-XXX1 from the high byte of the header indicates the presence of configuration data. Any other readout value terminates initialization from the EEPROM. If an EEPROM is attached but not used for configuration, the high byte of the first word should be programmed with 00h in order to ensure that the Ethernet port will not attempt to read configuration data from the EEPROM.

2.6.3.5 Determining Number of Bytes in the Reset Configuration Block

The low byte of the Reset Configuration Block header is known as the link byte. The value of the Link Byte represents the number of bytes of configuration data in the Reset Configuration Block. The two bytes used for the header are excluded when calculating the Link Byte value.

For example, a Reset Configuration Block header of A104h indicates a non-sequential EEPROM programmed with a Reset Configuration Block containing 4 bytes of configuration data. This Reset Configuration Block occupies 6 bytes (3 words) of EEPROM space (2 bytes for the header and 4 bytes of configuration data).

2.6.4 Groups of Configuration Data

Configuration data is arranged as groups of words. Each group contains one or more words of data that are to be loaded into Ethernet Port registers. The first word of each group is referred to as the Group Header. The Group Header indicates the number of words in the group and the address of the Ethernet Port register where the first data word in the group is to be loaded. Any remaining words in the group are stored in successive Ethernet Port registers.

2.6.4.1 Group Header

Bits F through C of the Group Header specify the number of words in each group that are to be transferred to Ethernet Port registers (see Figure 3 for the format). This value is two less than the total number of words in the group, including the Group Header. For example, if bits F through C contain 0001, there are three words in the group (a Group Header and two words of configuration data).

Bits 8 through 0 of the Group Header specify a 9-bit Ethernet Port Address. This address defines the Ethernet Port register that will be loaded with the first word of configuration data from the group. Bits B though 9 of the Group Header are forced to 0, restricting the destination address range to the first 512 bytes of Ethernet Port memory.

2.6.5 Reset Configuration Block Checksum

A checksum is stored in the high byte position of the word immediately following the last group of data in the Reset Configuration Block. (The EEPROM address of the checksum value can be determined by dividing the value stored in the Link Byte by two.) The checksum value is the 2's complement of the 8-bit sum (any carry out of eighth bit is

ignored) of all the bytes in the Reset Configuration Block, excluding the checksum byte. This sum includes the Reset Configuration Block header at address 00h. Since the checksum is calculated as the 2's complement of the sum of all preceding bytes in the Reset Configuration Block, a total of 0 should result when the checksum value is added to the sum of the previous bytes.

2.6.6 EEPROM Example

Table 5 shows an example of a Reset Configuration Block stored in a C46 EEPROM. Note that little-endian word ordering is used, i.e., the least significant word of a multiword datum is located at the lowest address.

2.6.7 EEPROM Read-out

If the EEDI pin is asserted high at the end of reset, the Ethernet port reads the first word of EEPROM data by:

1) Asserting EECS.
2) Clocking out a Read-Register-00h command on EEDO (EESK provides a 1 MHz serial clock signal).
3) Clocking the data in on EEDI.

If the EEDI pin is low at the end of the reset signal, the Ethernet port does not perform an EEPROM read-out (uses its default configuration).

2.6.7.1 Determining EEPROM Size

The Ethernet port determines the size of the EEPROM by checking the sense of EEDI on the tenth rising edge of EESK. If EEDI is low, the EEPROM is a 'C46 or 'CS46. If EEDI is high, the EEPROM is a 'C56, 'CS56, 'C66, or 'CS66.

2.6.7.2 Loading Configuration Data

The Ethernet port reads in the first word from the EEPROM to determine if configuration data is contained in the EEPROM. If configuration data is not stored in the EEPROM, the Ethernet port terminates initialization from EEPROM and operates using its default configuration (See Table 2). If configuration data is stored in EEPROM, the Ethernet port automatically loads all configuration data stored in the Reset Configuration Block into its internal Ethernet Port registers.

2.6.8 EEPROM Read-out Completion

Once all the configuration data are transferred to the appropriate Ethernet Port registers, the Ethernet port performs a checksum calculation to verify the Reset Configuration Blocks data are valid. If the resulting total is 0, the read-out is considered valid. Otherwise, the Ethernet port initiates a partial reset to restore the default configuration.

If the read-out is valid, the EEPROMOK bit (SelfST register, bit A) is set. EEPROMOK is

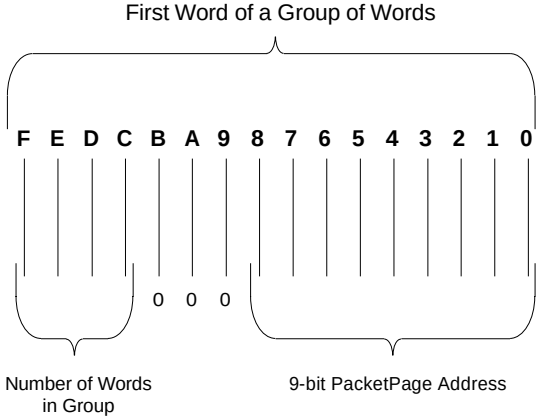


Figure 3. Group Header

cleared if a checksum error is detected. In this case, the Ethernet port performs a partial reset and is restored to its default. Once initialization is complete (configuration loaded from EEPROM or reset to default configuration) the INITD bit (SelfST register, bit 7) is set .

2.7 Clocks

The clock source is the on-chip PLL, enabled by strapping Port E pin 2 (PE[2]) low. This pin's state is latched at the rising edge of nPOR (power-up). After power-up, PE[2] can be used as a GPIO.

The CS89712 contains several separate sections of logic, each clocked according to its own clock frequency requirements. See each peripheral device section for more details. The section below describes the clocking for both the ARM720T and address/data bus.

2.7.1 On-Chip PLL

The ARM720T clock can be programmed to 18.432 MHz, 36.864 MHz, 49.152 MHz, or 73.728 MHz with the PLL running at 147456 MHz, twice the highest possible CPU clock frequency. The PLL uses an external 3.6864 MHz crystal. By default, the address/data buses run at 18.432 MHz.

When the clock frequency is selected to be 36 MHz, both the ARM720T and the address/data buses are clocked at 36 MHz. When the clock frequency is selected higher than 36 MHz, only the ARM720T gets clocked at this higher speed. The address/data will be fixed at 36 MHz. The clock frequency used is selected by programming the CLKCTL[1:0] bits in the SYSCON3 register. The clock frequency selection does not effect the EPB (external peripheral bus). Therefore, all the peripheral clocks are fixed, regardless of the clock speed selected for the ARM720T.

Note: After modifying the CLKCTL[1:0] bits, the next instruction should always be a 'NOP'.

2.7.1.1 Characteristics of the PLL Interface

When connecting a crystal to the on-chip PLL interface pins (i.e. MOSCIN and MOSCOUT), the crystal and circuit should conform to the following requirements:

- A 3.6864 MHz fundamental mode crystal should be used.
- A start-up resistor is not necessary, since one is provided internally.
- Start-up loading capacitors may be placed on each side of the external crystal and ground. Their value should be in the range of 10 pF. However, their values should be selected based upon the crystal specifications. The total sum of the capacitance of the traces between the CS89712's clock pins, the capacitors, and the crystal leads should be subtracted from the crystal's specifications when determining the values for the loading capacitors.
- The crystal frequency drift should be less than 100 ppm over the operating temperature range.

Alternatively, a digital clock source can be used to drive the MOSCIN pin of the CS89712. With this approach, the voltage levels of the clock source should match that of the V_{DD} supply for the CS89712's pads (i.e. the supply voltage level used to drive all of the non-V_{DD} core pins on the CS89712). The output clock pin (i.e., MOSCOUT) should be left floating.

2.7.2 *Dynamic Clock Switching*

The clock frequency used for the CPU and the buses is controlled by programming the CLKCTL[1:0] bits in the SYSCON3 register. When this register is written, clock switching logic waits until the clock that is currently in use and the newly programmed clock source are both low, and then switches from the previous clock frequency to the new clock without a glitch on the clocks.

2.7.3 *Ethernet Port Clock Oscillator*

A 20 MHz quartz crystal or CMOS clock input is required by the Ethernet port. If a CMOS clock input is used, it should be connected the to XTAL1 pin, with the XTAL2 pin left open. The clock signal should be 20 MHz ±0.01% with a duty cycle between 40% and 60%. The specifications for the crystal are described in Section 5.3.

2.8 Interrupt Controller

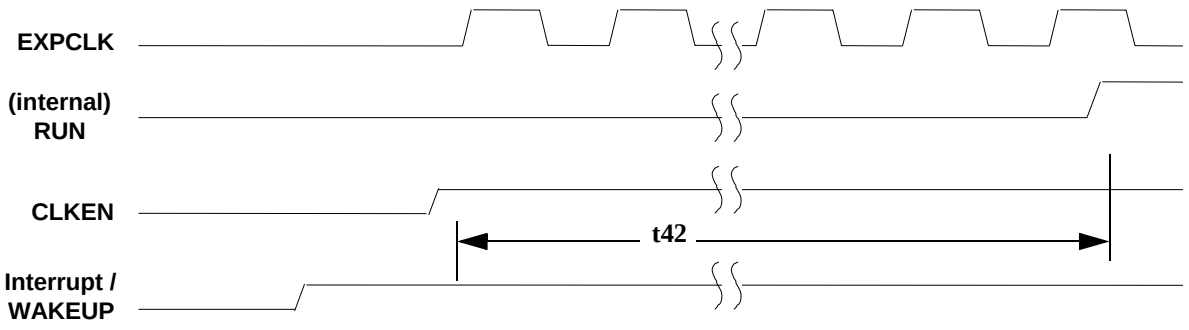
When unexpected events arise during the execution of a program (i.e., interrupt or memory fault) an exception is usually generated. When these exceptions occur at the same time, a fixed priority system determines the order in which they are handled. Table 6 shows the priority order of the exceptions.

Priority	Exception
Highest	Reset
.	Data Abort
.	FIQ
.	IRQ
.	Prefetch Abort
Lowest	Undefined Instruction, Software Interrupt

Table 6. Exception Priority Handling

The CS89712 interrupt controller has two interrupt types: interrupt request (IRQ) and fast interrupt request (FIQ). The interrupt controller has the ability to control interrupts from 22 different FIQ and IRQ sources. Of these, seventeen are mapped to the IRQ input and five sources are mapped to the FIQ input. FIQs have a higher priority than IRQs. If two interrupts are received from within the same group (IRQ or FIQ), the order in which they are serviced must be resolved in software. All interrupts are level sensitive; that is, they must conform to the following sequence:

1) The interrupting device (either external or internal) asserts the appropriate interrupt.
2) If the appropriate bit is set in the interrupt mask register, then either a FIQ or an IRQ will be as-



Note: t42=0.125 sec. to 0.25 sec.

Figure 4. CLKEN Timing Exiting the Standby State

serted by the interrupt controller. (A description for each bit in this register can be found in Section 3.6.1.

3) If interrupts are enabled the processor will jump to the appropriate address.

4) Interrupt dispatch software reads the interrupt status register to establish the source(s) of the interrupt and calls the appropriate interrupt service routine(s).

5) Software in the interrupt service routine will clear the interrupt source by some action specific to the device requesting the interrupt (i.e., reading the UART RX register).

The interrupt service routine may then re-enable interrupts, and any other pending interrupts will be serviced in a similar way. Alternately, it may return to the interrupt dispatch code, which can check for any more pending interrupts and dispatch them accordingly. The "End of Interrupt" type interrupts are latched. All other interrupt sources (i.e., external interrupt source) must be held active until its respective service routine starts executing. See Section 3.13, *"End Of Interrupt Locations"* for more details.

Table 7, Table 8, and Table 9 show the names and allocation of interrupts in the CS89712.

Interrupt	Bit in INTMR1 and INTSR1	Name	Comment
FIQ	0	EXTFIQ	External fast interrupt input (nEXTFIQ pin)
FIQ	1	BLINT	Battery low interrupt
FIQ	2	WEINT	Tick Watchdog expired interrupt
FIQ	3	MCINT	Media changed interrupt
IRQ	4	CSINT	Codec sound interrupt
IRQ	5	EINT1	External interrupt input 1 (nEINT[1] pin)
IRQ	6	EINT2	External interrupt input 2 (nEINT[2] pin)
IRQ	8	TC1OI	TC1 underflow interrupt
IRQ	9	TC2OI	TC2 underflow interrupt
IRQ	10	RTCMI	RTC compare match interrupt
IRQ	11	TINT	64 Hz tick interrupt
IRQ	12	UTXINT1	Internal UART1 transmit FIFO empty interrupt
IRQ	13	URXINT1	Internal UART1 receive FIFO full interrupt
IRQ	14	UMSINT	Internal UART1 modem status changed interrupt
IRQ	15	SSEOTI	Synchronous serial interface 1 end of transfer interrupt

Table 7. Interrupt Allocation in the First Interrupt Register

Interrupt	Bit in INTMR2 and INTSR2	Name	Comment
IRQ	0	KBDINT	Key press interrupt
IRQ	1	SS2RX	Master / slave SSI 16 bytes received
IRQ	2	SS2TX	Master / slave SSI 16 bytes transmitted
IRQ	12	UTXINT2	UART2 transmit FIFO empty interrupt
IRQ	13	URXINT2	UART2 receive FIFO full interrupt

Table 8. Interrupt Allocation in the Second Interrupt Register

Interrupt	Bit in INTMR3 and INTSR3	Name	Comment
FIQ	0	DAIINT	DAI interface interrupt

Table 9. Interrupt Allocation in the Third Interrupt Register

2.8.1 Interrupt Latencies

2.8.1.1 Operating State

The ARM720T core checks for a low level on its FIQ and IRQ inputs at each instruction boundary. The interrupt latency is therefore directly related to the amount of time it takes to complete execution of the current instruction when the interrupt condition is detected. First, there is a one to two clock cycle synchronization penalty. For the case where the CS89712 is operating with a 16-bit external memory system, and the program stored in one wait state FLASH memory, the worst-case interrupt latency is 251 clock cycles. This includes a delay for cache line fills for instruction prefetches, and a data abort occurring at the end of the LDM instruction, and the LDM being non-quad word aligned. In addition, the worst-case interrupt latency assumes that LCD DMA cycles to support a panel size of 320 x 240 at 4 bits-per-pixel, 60 Hz refresh rate, is in progress. This would give a worst-case interrupt latency of about 3.4 µs for 74 MHz operation. For operation at different frequencies and/or with 32 bit wide external memory, the latency will change accordingly.

For the nMEDCHG signal, this figure is substantially increased by the maximum time required to pass through the deglitcher, approximately 125 µs (2 cycles of the 16.384 kHz clock derived from the RTC oscillator). This results in an absolute worst-case latency of approximately 128 µs. Refer to Table 10 for a summary.

All the serial data transfer peripherals included in the CS89712 (except for the master-only SSI1) have local buffering to ensure a reasonable interrupt latency response requirement for the OS of < 1 ms. This assumes that the design data rates do not exceed the data rates described in this specification. If the OS cannot meet this requirement, there will be a risk of data over/underflow occurring.

2.8.1.2 Idle State

When leaving the Idle State as a result of an interrupt, the CPU clock is restarted after approximately two clock cycles. However, there is still potentially up to a 251 clock latency as described in the first section above, unless the code is written to include at least two single cycle instructions immediately after the write to the IDLE register (in which case the latency drops to a few microseconds). This is important, as the Idle State can only be left because of a pending interrupt, which has to be synchronized by the processor before it can be serviced.

2.8.1.3 Standby State

In the Standby State, the latency will depend on whether the system clock is shut down and if the FASTWAKE bit in the SYSCON3 register is set. If the system is configured to run from the internal PLL clock, then the PLL will always be shut down when in the Standby State. In this case, if the FASTWAKE bit is cleared, then there will be a latency of between 0.125 sec to 0.25 sec. If the FASTWAKE bit is set, then there will be a latency of between 250 µsec to 500 µsec.

Whenever the CS89712 is in the Standby State, the external address and data buses are driven low. The RUN signal is used internally to force these buses to be driven low. This prevents de-powered peripherals from draining current. Also, the internal peripheral's signals are set to their Reset State.

2.8.1.4 Snooze State

In Snooze State, the latency will be reduced to the same as for the Idle State described above. This is true at any frequency because the PLL or external clock source is not stopped. All clocks except the minimum required for LCD refresh from the internal SRAM are disabled to save further power.

To drastically reduce the potential worst case latency when leaving Snooze State to a few microseconds, ensure that the code contains two single cycle

instructions immediately after the write to the SNOOZE register location.

2.8.1.5 Doze State

Since Doze State can be considered a preliminary state between Snooze State and Operating State, the only requirement for existing this state into the Operating State is for a few instructions to be executed. Therefore, the latency is based solely upon the time required to execute these instructions.

Table 10 summarizes the five external interrupt sources and their effect on the processor interrupts.

2.9 Boot ROM

The 128 bytes of on-chip Boot ROM contain an instruction sequence that initializes the device and then configures UART1 to receive 2048 bytes of serial data that will then be placed in the on-chip SRAM. Once the download is complete, execution jumps to the start of the on-chip SRAM. This would allow, for example, code to be downloaded to program system FLASH during a product's manufacturing process. See Section , *"Appendix B: Boot Code"* for details of the ROM Boot Code with comments to describe the stages of execution.

Selection of the Boot ROM option is determined by the state of the nMEDCHG pin during a power on reset. If nMEDCHG is high while nPOR is active, then the CS89712 will boot from an external memory device connected to CS[0] (normal boot mode). If nMEDCHG is low, then the boot will be from the on-chip ROM. Note that in both cases, following the de-assertion of nPOR, the CS89712 will be in the Standby State and require a low-to-high transition on the external WAKEUP pin in order to actually start the boot sequence.

The effect of booting from the on-chip Boot ROM is to reverse the decoding for all chip selects internally. Table 11 shows this decoding. The control signal for the boot option is latched by nPOR, which means that the remapping of addresses and bus widths will continue to apply until nPOR is asserted again. After booting from the Boot ROM, the contents of the Boot ROM can be read back from address 0x0000.0000 onwards, and in normal state of operation the Boot ROM contents can be read back from address range 0x7000.0000.

2.10 Memory Map

The lower 2 GByte of the address space is allocated to memory. The 512 MBytes of address space from

Interrupt Pin	Input State	Operating State Latency	Idle State Latency	Standby State Latency
nEXTFIQ	Not deglitched; must be active for 251 clock cycles to ensure detection	Worst-case 3.4 µsec at 74 MHz	Worst-case 251 clocks: if only single cycle instructions, less than 1 µsec	Including PLL / osc. settling time, ~ 0.25 sec when FASTWAKE = 0, or approx. 500 µsec when FASTWAKE = 1
nEINT1–2	Not deglitched	Worst-case 3.4 µsec at 74 MHz	As above	As above
nMEDCHG	Deglitched by 16 kHz clock; must be active for at least 125 µs to be detected	Worst-case latency of 128 µsec at 74 MHz	Worst-case 80 µsec: if only single cycle instructions, 125 µsec	As above

Table 10. External Interrupt Source Latencies

Address Range	Chip Select
0000.0000–0FFF.FFFF	CS[7] (Internal only)
1000.0000–1FFF.FFFF	CS[6] (Internal only)
2000.0000–2FFF.FFFF	nCS[5]
3000.0000–3FFF.FFFF	nCS[4]
4000.0000–4FFF.FFFF	nCS[3]
5000.0000–5FFF.FFFF	nCS[2]
6000.0000–6FFF.FFFF	nCS[1]
7000.0000–7FFF.FFFF	nCS[0]

Table 11. Chip Select Address Ranges After Boot From On-Chip Boot ROM

0xC000.0000 to 0xDFFF.FFFF is allocated to SDRAM. The 1.5 GByte, less 8 kbytes for internal registers, is not accessible in the CS89712. The MMU should be programmed to generate an abort exception for access to this area.

Internal peripherals are addressed through a set of internal registers from address 0x8000.0000 to 0x8000.3FFF.

Table 12 shows how the 4-Gbyte address range of the ARM720T processor (as configured within this chip) is mapped. The memory map shown assumes that two CL-PS6700 PC Card controllers are connected. If this functionality is not required, then the nCS[4] and nCS[5] memory is available. The external boot ROM is not fully decoded (i.e., the boot code will repeat within the 256 Mbyte space from 0x7000.0000 to 0x8000.0000).

When booted from on chip boot ROM, the SRAM is fully decoded up to 128 kbytes. Access to any location above this range will wrap within the range.

Address	Contents	Size
0xF000.0000	Reserved	256 Mbytes
0xE000.0000	Reserved	256 Mbytes
0xD000.0000	Reserved	256 Mbytes
0xC000.0000	SDRAM	64 Mbytes
0x8000.4000	Unused	~1 Gbyte
0x8000.2000	Internal registers	8 kbytes
0x8000.0000	Internal registers	8 kbytes
0x7000.0000	Boot ROM (nCS[7])	128 bytes
0x6000.0000	SRAM (nCS[6])	48k bytes
0x5000.0000	PCMCIA-1 (nCS[5])	4 x 64 Mbytes
0x4000.0000	PCMCIA-0 (nCS[4])	4 x 64 Mbytes
0x3000.0000	Expansion (nCS[3])	256 Mbytes
0x2000.0000-0x2000.02FF	Expansion (nCS[2])	256 Mbytes
0x2000.0300-0x2000.030F	Ethernet Port (on nCS[2])	
0x2000.0310-0x2FFF.FFFF	Expansion (nCS[2]) cont.	
0x1000.0000	ROM Bank 1 (nCS[1])	256 Mbytes
0x0000.0000	ROM Bank 0 (nCS[0])	256 Mbytes

Table 12. CS89712 Memory Map in External Boot Mode

2.11 Memory and I/O Expansion Interface

Six separate linear memory or expansion segments are decoded by the CS89712, two of which can be reserved for two PC Cards, each interfacing to a separate single CL-PS6700 device. Each segment is 256 Mbytes in size. Two additional segments (in addition to these six) are dedicated to the on-chip SRAM and ROM. The on-chip ROM space is fully decoded, and the SRAM space is decoded up to the maximum size of the video frame buffer programmed in the LCDCON register (128 kbytes). Beyond this address range the SRAM space is not fully decoded (i.e., any accesses beyond 128 kbyte range get wrapped around to within 128 kbyte range). Any of the six segments are configured to interface to a conventional SRAM-like interface, and can be individually programmed to be 8-, 16-, or 32-bits wide, to support page mode access, and to execute from 1 to 8 wait states for non-sequential accesses and 0 to 3 for burst mode accesses. The zero wait state sequential access feature is designed to support burst mode ROMs. For writable memory devices which use the nMWE pin, zero wait state sequential accesses are not permitted and one wait state is the minimum which should be programmed in the sequential field of the appropriate MEMCFG register. Bus cycles can also be extended using the EXPRDY input signal.

Page mode access is accomplished by setting SQAEN = 1, enabling accesses of one random address followed by three sequential addresses, etc., while keeping nCS asserted. These sequential bursts can be up to four words long before nCS is released to allow DMA and refreshes to take place. This can significantly improve bus bandwidth to devices such as ROMs which support page mode. When SQAEN = 0, all accesses to memory are by random access without nCS being de-asserted between accesses. Again nCS is de-asserted after four consecutive accesses to allow DMA.

Bits 5 and 6 of the SYSCON2 register independently enable the interfaces to the CL-PS6700 (PC Card slot drivers). When either of these interfaces are enabled, the corresponding chip select (nCS4 and/or nCS5) becomes dedicated to that CL-PS6700 interface. The state of SYSCON2 bit 5 determines the function of chip select nCS4 (i.e., CL-PS6700 interface or standard chip select functionality); bit 6 controls nCS5 in a similar way. There is no interaction between these bits.

For applications that require a display buffer smaller than 48k bytes, the on-chip SRAM can be used as the frame buffer.

Before entering the Snooze State, the SRAM at 0x6000.0000 must be updated, under program control, with data to be displayed during the Snooze State. In a system using the on-chip SRAM as the frame buffer in normal operation, Snooze State can be entered without requiring any data transfer first, assuming data is stored in the on-chip SRAM in 1-bit -per-pixel format.

The width of the boot device can be chosen by selecting values of PE[1] and PE[0] during power on reset. The inputs in Table 13 are latched by the rising edge of nPOR to select the boot option.

PE[1]	PE[0]	Boot Block (nCS0)
0	0	32-bit
0	1	8-bit
1	0	16-bit
1	1	Undefined

Table 13. Boot Options

2.12 SDRAM Controller

The SDRAM controller provides all the connections to directly interface to up to two banks of SDRAM, and the width of the memory interface is programmable from 16 to 32 bits wide. Both banks have to be of the same width. Each of the two banks supported can be up to 256 Mbits in size. The sig-

nals nRAS nCAS, and nWE are provided for SDRAM. Two chip selects are provided for supporting up to 2 rows of SDRAMs. The SDRAM devices are put into self-refresh mode when the SDRAM controller is put into standby. The SDRAM clock is halted as well.

The controller supports read, write, refresh, precharge and mode register write requests to the SDRAM. Data is transferred to and from the SDRAM as unbroken quad accesses (either quad word or for 16 bit memory, quad halfword), which is a convenient data packet size for the ARM cache line fills. For the CPU to read smaller than a quad access, the SDRAM controller will discard the extra data. For CPU writes smaller than a quad access, the SDQM pins (SDRAM data byte mask selects) are used to force the SDRAMs to ignore invalid data. For CPU access sizes larger than a quad access, multiple quad accesses are issued to the SDRAM.

The SDRAM controller can access a total memory size of 2-64 Mbytes. Each individual SDRAM should be NEC or compatible SDRAM memory in sizes of 16-256 Mbits, arranged as shown in Table 14 and Table 15.

Chip selects for row 1 SDRAMs should be connected to nSDCS[0]. If row 2 is used, these devices should connect to nSDCS[1].

For 32-bit memory access, four SDQM data byte mask selects are provided to control individual byte lanes within each row. For 16-bit memory access only, SDQM[1:0] are used. For a 32-bit memory access configuration with each row containing two 16-bit wide SDRAMs, the high order SDRAM should have UDQM (upper SDQM) connected to SDQM[3] and LDQM (lower SDQM) connected to SDQM[2]. The low order SDRAM follows the same convention: USDQM is connected to SDQM[1], and LDQM is connected to SDQM[0].

Memory address line multiplexing is done internally so that the address mapping is contiguous. Table 16 indicates how the SDRAM address pins are connected to the CPU's address pins. Note that small SDRAM devices will not use all of these pins. For example, A[12:11] may not be required. However, the bank select pins BA[1:0], are required by all SDRAMs. Smaller devices may only have one bank, so BA1 may not be needed.

SDRAM details		Arrangement of SDRAMs (C = # Columns of SDRAM, R = # Rows of SDRAM, D = # of SDRAMs)														
		4 Mbytes			8 Mbytes			16 Mbytes			32 Mbytes			64 Mbytes		
Density (Mbits)	Width (bits)	C	R	D	C	R	D	C	R	D	C	R	D	C	R	D
16	4							8	1	8						
	8				4	1	4									
	16	2	1	2												
64	4													8	1	8
	8										4	1	4	4	2	8
	16							2	1	2	2	2	4			
	32				1	1	1	1	2	2						
128	4															
	8													4	1	4

Table 14. SDRAM Configurations (SDRAM 32-Bit Memory Interface)

	16										2	1	2	2	2	4
256	4															
	8															
	16													2	1	2

Table 14. SDRAM Configurations (SDRAM 32-Bit Memory Interface)

SDRAM Details		Arrangement of SDRAMs (C = # Columns of SDRAM, R = # Rows of SDRAM, D = # of SDRAMs)																	
		2 Mbytes			4 Mbytes			8 Mbytes			16 Mbytes			32 Mbytes			64 Mbytes		
Density (Mbits)	Width (bits)	C	R	D	C	R	D	C	R	D	C	R	D	C	R	D	C	R	D
16	4							4	1	4									
	8				2	1	2												
	16	1	1	1															
64	4													4	1	4	4	2	8
	8										2	1	2	2	2	4			
	16							1	1	1	1	2	2						
128	4																4	1	4
	8													2	1	2	2	2	4
	16										1	1	1	1	2	2			
256	4																		
	8																2	1	2
	16													1	1	1	1	2	2

Table 15. SDRAM Configurations (SDRAM 16-Bit Memory Interface)

SDRAM Address Pins	CS89712 Pin Names
A0.	a27/dra0
A1.	a26/dra1
A2.	a25/dra2
A3.	a24/dra3
A4.	a23/dra4
A5.	a22/dra5
A6.	a21/dra6
A7.	a20/dra7
A8.	a19/dra8
A9.	a18/dra9
A10.	a17/dra10
A11.	a16/dra11
A12.	a15/dra12
BA0.	a14/dra13
BA1.	a13/dra14

Table 16. SDRAM Address Pin Connections

2.13 SDRAM Initialization

The SDRAM is initialized in the power-on sequence as follows:

1) To stabilize internal circuits when power is applied, a 200+ µs pause must precede any signal toggling.
2) After the pause, all banks must be precharged using the Precharge command (including the precharge all banks command).
3) Once the precharge is complete, and the minimum tRP is satisfied, the mode register can be programmed. After the mode register set cycle, tRSC (2 CLK minimum) pause must be satisfied as well. (Only required for NEC SDRAM)
4) Eight or more refresh cycles must be performed.

2.14 CL-PS6700 PC Card Interface

Two of the expansion memory areas are dedicated to supporting up to two CL-PS6700 PC Card controller devices. These are selected by nCS4 and nCS5 (must first be enabled by bits 5 and 6 of SYSCON2). For efficient, low power operation, both address and data are carried on the lower 16 bits of the CS89712 data bus. Accesses are initiated by a write or read from the area of memory allocated for nCS4 or nCS5. The memory map within each of these areas is segmented to allow different types of PC Card accesses to take place, for attribute, I/O, and common memory space. The CL-PS6700 internal registers are memory mapped within the address space as shown in Table 17.

Note: Due to the operating speed of the CL-PS6700, this interface is supported only for a processor speed of 18 MHz.

A complete description of the protocol and AC timing characteristics can be found in the CL-PS6700 data sheet. A transaction is initiated by an access to the nCS4 or nCS5 area. The chip select is asserted, and on the first clock, the upper 10 bits of the PC Card address, along with 6 bits of size, space, and slot information are put out onto the lower 16 bits

Access Type	Addresses for CL-PS6700 Interface 1	Addresses for CL-PS6700 Interface 2
Attribute	0x4000.0000–0x43FF.FFFF	0x5000.0000– 0x53FF.FFFF
I/O	0x4400.0000–0x47FF.FFFF	0x5400.0000–0x57FF.FFFF
Common memory	0x4800.0000–0x4BFF.FFFF	0x5800.0000–0x5BFF.FFFF
CL-PS6700 registers	0x4C00.0000–0x4FFF.FFFF	0x5C00.0000–0x5FFF.FFFF

Table 17. CL-PS6700 Memory Map

of the CS89712's data bus. Only word (i.e., 4-byte) and single-byte accesses are supported, and the slot field is hardcoded to 11, since the slot field is defined as a 'Reserved field' by the CL-PS6700. The chip selects are used to select the device to be accessed. The space field is made directly from the A26 and A27 CPU address bits, according to the decode shown in Table 18. The size field is forced to 11 if a word access is required, or to 00 if a byte access is required. This avoids the need to configure the interface after a reset. On the second clock cycle, the remaining 16 bits of the PC Card address are multiplexed out onto the lower 16 bits of the data bus. If the transaction selected is a CL-PS6700 register transaction, or a write to the PC Card (assuming there is space available in the CL-PS6700's internal write buffer) then the access will continue on the following two clock cycles. During these following two clock cycles the upper and lower halves of the word to be read or written will be put onto the lower 16 bits of the main data bus.

The 'ptype' signal on the CL-PS6700s should be connected to the CS89712's WRITE output pin. During PC Card accesses, the polarity of this pin changes, and it becomes low to signify a write and high to signify a read. It is valid with the first half word of the address. During the second half word of the address, it is always forced high to indicate to the CL-PS6700 that the CS89712 has initiated either the write or read.

The PRDY signals from each of the two CL-PS6700 devices are connected to Port B bits 0 and 1, respectively. When the PC CARD1 or PC CARD2 control bits in the SYSCON2 register are de-asserted, these port bits are available for GPIO. When asserted, these port bits are used as the PRDY signals. When the PRDY signal is de-asserted (i.e., low), it indicates that the CL-PS6700 is busy accessing its card. If a PC CARD access is attempted while the device is busy, the PRDY signal will cause the CS89712's CPU to be stalled. The CS89712's CPU will have to wait for the card to become available. DMA transfers to the LCD can still continue in the background during this period of time (as described below). The CS89712 can access the registers in the CL-PS6700, regardless of the state of the PRDY signal. If the CS89712 needs to access the PC CARD via the CL-PS6700, it waits until the PRDY signal is high before initiating a transfer request. Once a request is sent, the PRDY signal indicates if data is available.

In the case of a PC Card write, writes can be posted to the CL-PS6700 device, with the same timing as CL-PS6700 internal register writes. Writes will normally be completed by the CL-PS6700 device independent of the CS89712 processor activity. If a posted write times out, or fails to complete for any other reason, then the CL-PS6700 will issue an interrupt (i.e., a WR_FAIL interrupt). In the case where the CL-PS6700 write buffer is already full, the PRDY signal will be de-asserted (i.e., driven low) and the transaction will be stalled pending an available slot in the buffer. In this case, the CS89712's CPU will be stalled until the write can

Space Field Value	PC CARD Memory Space
00	Attribute
01	I/O
10	Common memory
11	CL-PS6700 registers

Table 18. Space Field Decoding

be posted successfully. While the PRDY signal is de-asserted, the chip select to the CL-PS6700 will be de-asserted and the main bus will be released so that DMA transfers to the LCD controller can continue in the background.

In the case of a PC Card read, the PRDY signal from the CL-PS6700 will be de-asserted until the read data is ready. At this point, it will be reasserted and the access will be completed in the same way as for a register access. In the case of a byte access, only one 16-bit data transfer will be required to complete the access. While the PRDY signal is de-asserted, the chip select to the CL-PS6700 will be de-asserted, and the main bus will be released so that DMA transfers to the LCD controller can continue in the background.

The CS89712 will re-arbitrate for the bus when the PRDY signal is reasserted to indicate that the read or write transaction can complete. The CPU will stall until the PC Card access is completed.

A card read operation may be split into a request cycle and a data cycle, or it may be combined into a single request/data transfer cycle. This depends on whether the requested data is available in the internal CL-PS6700 prefetch buffer.

The request portion of the cycle, for a card read, is similar to the request phase for a card write (described above). If the requested data is available in the prefetch buffer, the CL-PS6700 asserts the PRDY signal before the rising edge of the third clock and the CS89712 continues the cycle to read the data. Otherwise, the PRDY signal is de-asserted, and the request cycle is stalled. The CS89712 may then allow the DMA address generator to gain control of the bus, to allow LCD refreshes to continue. When the CL-PS6700 is ready with the data, it asserts the PRDY signal. The CS89712 then arbitrates for the bus and, once the request is granted, the suspended read cycle is resumed. The CS89712 resumes the cycle by asserting the appropriate chip select, and data is transferred on the next two clocks if a word read (one clock if a byte read).

There is no support within the CS89712 for detecting time-outs. The CL-PS6700 device must be programmed to force the cycle to be completed (with invalid data for a read) and then generate an interrupt if a read or write access has timed out (i.e., RD_FAIL or WR_FAIL interrupt). The system software can then determine which access was not successfully completed by reading the status registers within the CL-PS6700.

The CL-PS6700 has support for DMA data transfers. However, DMA is supported only by software emulation because the DMA address generator built into the CS89712 is dedicated to the LCD controller interface. If DMA is enabled within the CL-PS6700, it will assert its PDREQ signal to make a DMA request. This can be connected to one of the CS89712's external interrupts and be used to interrupt the CPU for servicing the DMA request.

Each of the CL-PS6700 devices can generate an interrupt PIRQ. Since the PIRQ signal is an open drain on the CL-PS6700 devices, two CL-PS6700 devices may be wired OR'ed to the same interrupt. The circuit can then be connected to one of the CS89712's active low external interrupt sources. On the receipt of an interrupt, the CPU can read the interrupt status registers on the CL-PS6700 devices to determine the cause of the interrupt.

All transactions are synchronized to the EXPCLK output from the CS89712 in 18.432 MHz mode. The EXPCLK should be permanently enabled, by setting the EXCKEN bit in the SYSCON1 register, when the CL-PS6700 is used. The reason for this is that the PC Card interface and CL-PS6700 internal write buffers need to be clocked after the CS89712 has completed its bus cycles.

A GPIO signal from the CS89712 can be connected to the PSLEEP pin of the CL-PS6700 devices to allow them to be put into a power saving state before the CS89712 enters the Standby State. It is essen-

Address (W/B)	Data in Memory (as seen by the CS89712)	Byte Lanes to Memory / Ports / Registers								R0 Contents	
		Big Endian Memory				Little Endian Memory					
		7:0	15:8	23:16	31:24	7:0	15:8	23:16	31:24	Big Endian	Little Endian
Word + 0 (W)	11223344	44	33	22	11	44	33	22	11	11223344	11223344
Word + 1 (W)	11223344	44	33	22	11	44	33	22	11	44112233	44112233
Word + 2 (W)	11223344	44	33	22	11	44	33	22	11	33441122	33441122
Word + 3 (W)	11223344	44	33	22	11	44	33	22	11	22334411	22334411
Word + 0 (H)	11223344	dc	dc	22	11	44	33	dc	dc	00001122	00003344
Word + 1 (H)	11223344	dc	dc	22	11	44	33	dc	dc	22000011	44000033
Word + 2 (H)	11223344	44	33	dc	dc	dc	dc	22	11	00003344	00001122
Word + 3 (H)	11223344	44	33	dc	dc	dc	dc	22	11	44000033	22000011
Word + 0 (B)	11223344	dc	dc	dc	11	44	dc	dc	dc	00000011	00000044
Word + 1 (B)	11223344	dc	dc	22	dc	dc	33	dc	dc	00000022	00000033
Word + 2 (B)	11223344	dc	33	dc	dc	dc	dc	22	dc	00000033	00000022
Word + 3 (B)	11223344	44	dc	dc	dc	dc	dc	dc	11	00000044	00000011

Note: dc = don't care

Table 19. Effect of Endianness on Read Operations

tial that software monitors the appropriate status registers within the CL-PS6700s to ensure that there are no pending posted bus transactions before the Standby State is entered. Failure to do this will result in incomplete PC Card accesses.

2.15 Endianness

The CS89712 uses a little endian configuration for internal registers. However, it is possible to connect the device to a big endian external memory system. The big-endian / little-endian bit in the ARM720T control register sets whether the CS89712 treats words in memory as being stored in big endian or little endian format. Memory is viewed as a linear collection of bytes numbered upwards from zero. Bytes 0 to 3 hold the first stored word, bytes 4 to 7 the second, and so on. In the little endian scheme, the lowest numbered byte in a word is considered to be the least significant byte of the word and the highest numbered byte is the most significant. Byte 0 of the memory system should be connected to D[7:0] in this case. In the big endian scheme the most significant byte of a word is stored at the lowest numbered byte, and the least significant byte is stored at the highest numbered byte. Therefore, byte 0 of the memory system should be connected to D[31:24]. Load and store are the only instructions affected by the Endianness.

Table 19 and Table 20 demonstrate the behavior of the CS89712 for read and write operations, including the effect of performing non-aligned word accesses. The register definition section defines the behavior of the internal CS89712 registers in the big endian mode in more detail. For further information, refer to ARM Application Note 61, "*Big and Little Endian Byte Addressing*".

Address (W/B)	Register Contents	Byte Lanes to Memory / Ports / Registers							
		Big Endian Memory				Little Endian Memory			
		7:0	15:8	23:16	31:24	7:0	15:8	23:16	31:24
Word + 0 (W)	11223344	44	33	22	11	44	33	22	11
Word + 1 (W)	11223344	44	33	22	11	44	33	22	11
Word + 2 (W)	11223344	44	33	22	11	44	33	22	11
Word + 3 (W)	11223344	44	33	22	11	44	33	22	11
Word + 0 (H)	11223344	44	33	44	33	44	33	44	33
Word + 1 (H)	11223344	44	33	44	33	44	33	44	33
Word + 2 (H)	11223344	44	33	44	33	44	33	44	33
Word + 3 (H)	11223344	44	33	44	33	44	33	44	33
Word + 0 (B)	11223344	44	44	44	44	44	44	44	44
Word + 1 (B)	11223344	44	44	44	44	44	44	44	44
Word + 2 (B)	11223344	44	44	44	44	44	44	44	44
Word + 3 (B)	11223344	44	44	44	44	44	44	44	44

Note: Bold indicates active byte lane.

Table 20. Effect of Endianness on Write Operations

2.16 Internal UARTs and SIR Encoder

The CS89712 contains two built-in UARTs that offers similar functionality to National Semiconductor's 16C550A device. Both UARTs can support bit rates of up to 115.2 kbits/s and include two 16-byte FIFOs: one for receive and one for transmit.

One of the UARTs (UART1) supports the three modem control input signals CTS, DSR, and DCD. The additional RI input, and RTS and DTR output modem control lines are not explicitly supported but can be implemented using GPIO ports in the CS89712. UART2 has only the RX and TX pins.

UART operation and line speeds are controlled by the UBLCR1 (UART bit rate and line control). Three interrupts can be generated by UART1: RX, TX, and modem status interrupts. Only two can be generated by UART2: RX and TX. The RX interrupt is asserted when the RX FIFO becomes half full or if the FIFO is non-empty for longer than three character length times with no more characters being received. The TX interrupt is asserted if the TX FIFO buffer reaches half empty. The modem status interrupt for UART1 is generated if any of the modem status bits change state. Framing and parity errors are detected as each byte is received and pushed onto the RX FIFO. An overrun error generates an RX interrupt immediately. All error bits can be read from the 11-bit wide data register. The FIFOs can also be programmed to be one byte depth only (i.e., like a conventional 16450 UART with double buffering).

The CS89712 also contains an IrDA (Infrared Data Association) SIR protocol encoder as a post-processing stage on the output of UART1. This encoder can be optionally switched into the TX and RX signals of UART1, so that these can be used to drive an infrared interface directly. If the SIR protocol encoder is enabled, the UART TXD1 line is held in the passive state and transitions of the RXD1 line will have no effect. The IrDA output pin

is LEDDRV, and the input from the photodiode is PHDIN. Modem status lines will cause an interrupt (which can be masked) irrespective of whether the SIR interface is being used.

Both the UARTs operate in a similar manner to the industry standard 16C550A. When CTS is deasserted on the UART, the UART does not stop shifting the data. It relies on software to make an appropriate response to the interrupt generated.

Baud rates supported for both the UARTs are dependent on frequency of operation. When operating from the internal PLL, the interface supports various baud rates from 115.2 kbits/s downwards. The master clock frequency is chosen so that most of the required data rates are obtainable exactly.

2.17 Synchronous Serial Interfaces

The CS89712 has the synchronous serial interfaces shown in Table 21. Three sets of serial interface (DAI, CODEC, and SSI2) pins are multiplexed together. On power up, both the DAISEL and SERSEL register bits are low, enabling the master / slave SSI2 to these pins (and configuring it for slave mode operation to avoid external contention).

Table 22 contains pin definition information for the three multiplexed interfaces.

The internal names given to each of the three interfaces are unique to help differentiate them from each other. The sections below that describe each of the three interfaces will use their respective unique internal pin names for clarity.

Type	Comments	Referred To As	Max. Transfer Speed
SPI / Microwire 1	Master mode only	ADC Interface	128 kbits/s
SPI / Microwire 2	Master / slave mode	SSI2 Interface	512 kbits/s
DAI Interface	CD quality DACs and ADCs	DAI Interface	1.536 Mbits/s
CODEC Interface		CODEC Interface	64 kbits/s

Table 21. Serial Interface Options

BGA Ball	External Pin Name	SSI2 Slave Mode (Internal Name)	SSI2 Master Mode	CODEC Internal Name	DAI Internal Name	Strength
	SSICLK	SSICLK = serial bit clock; Input	Output	PCMCLK = Output	SCLK = Output	1
	SSITXFR	SSITXFR = TX frame sync; Input	Output	PCMSYNC = Output	LRCK = Output	1
	SSITXDA	SSITXDA = TX data; Output	Output	PCMOUT = Output	SDOUT = Output	1
	SSIRXDA	SSIRXDA = RX data; Input	Input	PCMIN = Input	SDIN = Input	
	SSIRXFR	SSIRXFR = RX frame sync; Input	Output	p/u (use a 10k pull-up)	MCLK	1

Table 22. Serial-Pin Assignments

2.17.1 Codec Sound Interface

The codec interface allows direct connection of a telephony type codec to the CS89712. It provides all the necessary clocks and timing pulses. It also performs a parallel to serial conversion or vice versa on the data stream to or from the external codec device. The interface is full duplex and contains two separate data FIFOs (16 deep by 8-bits, one for the receive data, another for the transmit data).

Data is transferred to or from the codec at 64 kbits/s. The data is either written to or read from the appropriate 16-byte FIFO. If enabled, a codec interrupt (CSINT) will be generated after every 8 bytes are transferred (FIFO half full/empty). This means the interrupt rate will be every 1 msec, with a latency of 1 msec.

Transmit and receive modes are enabled by asserting high both the CDENRX and CDENTX codec enable bits in the SYSCON1 register.

Note: Both the CDENRX and CDENTX enable bits should be asserted in tandem for data to be transmitted or received. The reason for this is that the interrupt generation will occur 1 msec after one of the FIFOs is enabled. For example: If the receive FIFO gets enabled first and the transmit FIFO at a later time, the interrupt will occur 1 msec after the receive FIFO is enabled. After the first interrupt occurs, the receive FIFO will be half full. However, it will not be possible to know how full the transmit FIFO will be since it was enabled at a later time. Thus, it is possible to unintentionally overwrite data already in the transmit FIFO (See **Figure 5**).

2.17.1.1 Codec Interrupt Timing

After the CDENRX and CDENTX enable bits are asserted, the corresponding FIFOs become enabled. When both FIFOs are disabled, the FIFO status flag CRXFE is set and CTXFF is cleared so that the FIFOs appear empty. Additionally, if the CDENTX bit is low, the PCMOUT output is disabled. Asserting either of the enable bits causes the sync and interrupt generation logic to activate; otherwise they are disabled to conserve power.

Data is loaded into the transmit FIFO by writing to the CODR register. At the beginning of a transmit cycle, this data is loaded into a shift/load register.

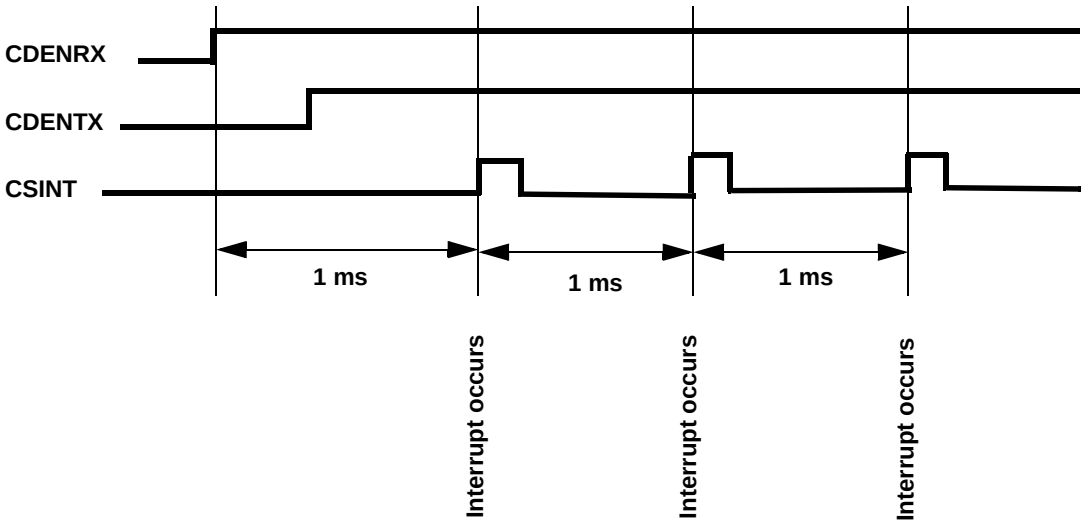


Figure 5. Codec Interrupt Timing

Just prior to the byte being transferred out, PCMSYNC goes high for one PCMCLK cycle. Then the data is shifted out serially to PCMOUT, MSB first, (with the MSB valid at the same time PCMSYNC is asserted). Data is shifted on the rising edge of the PCMCLK output.

Receiving of data is performed by taking data in serially through PCMIN, again MSB first, shifting it through the shift/load register and loading the complete byte into the receive FIFO. If there is no data available in the transmit FIFO, then a zero will be loaded into the shift/load register. Input data is sampled on the falling edge of PCMCLK. Data is read from the CODR register.

Note: After data is transmitted, the speaker amplifier should be turned off to avoid audible noise. This is needed because the CS89712 will continue to transmit data from the FIFO even though it is empty, thus causing noise. This will occur even when receiving.

2.17.2 Digital Audio Interface

The DAI interface provides a high quality digital audio connection to DAI compatible audio devices. The DAI is a subset of I2S audio format that is supported by a number of manufacturers.

The DAI interface produces one 128-bit frame at the audio sample frequency using a bit clock and frame sync signal. Digital audio data is transferred, full duplex, via separate transmit and receive data lines. The bit clock frequency is programmable to 64 fs or 128 fs. The sample frequency (fs) is now programmable from 8-48Khz using either the on-chip PLL (73.728MHz) or the external 11.2896 Mhz clock.

The DAI interface contains separate transmit and receive FIFO's. The transmit FIFO's are 8 audio samples deep and the receive FIFO's are 12 audio samples deep.

DAI programming centers around the selection of the desired sample frequency (fs). All three clocks (MCLK, LRCK, SCLK) become a multiple of the selected sample frequency as illustrated on the previous page. The DAI share the same output with the CODEC and SSI as shown in Figure 6. Please see Table 23 for the MUX programming matrix.

2.17.2.1 DAI Operation

Following reset, the DAI logic is disabled. To enable the DAI, the applications program should first clear the emergency underflow and overflow status bits, which are set following the reset, by writing a 1 to these register bits (in the DAISR register). Next, the DAI control register should be programmed with the desired mode of operation using a word write. The transmit FIFOs can either be "primed" by writing up to eight 16-bit values each, or can be filled by the normal interrupt service routine which handles the DAI FIFOs. Finally, the FIFOs for each channel must be enabled via writes to DAIDR2. At this point, transmission/reception of data begins on the transmit (SDOUT) and re-

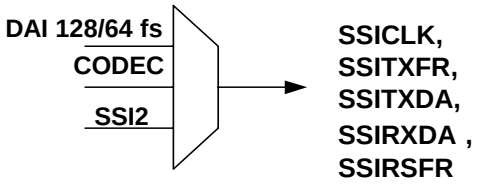


Figure 6. Portion of the CS89712 Block Diagram Showing Multiplexed Feature

ceive (SDIN) pins. This is synchronously controlled by either the PLL or the external clock. These fixed frequencies pass through a programmable divider network which will create the appropriate values for SCLK, LRCLK, and MCLK for the desired sample frequency. Examples of sample frequencies are shown in Table 24. Register DAI64Fs enables/disables the bit clock frequency of 64 fs (and the other features as shown in Figure 7), but must be complemented by SYSCON3 bit 9 which will enable/disable 128 fs. To enable one rate, you must disable the other.

2.17.2.2 DAI Frame Format

Each DAI frame is 128 bits long and comprises one audio sample. Of this 128-bit frame, only 32 bits are used for digital audio data. The remaining bits are output as zeros. The LRCK signal is used as a

FEATURE	SYSCON3	DAIR (DAI)	DAI64 fs	SYSCON2
DAI –128 fs	DAISEL[3] (H)	DAIEN[16] (H)	I2SF64[0] (L)	(X)
	128Fs[9] (H)			
DAI-64 fs	DAISEL[3] (H)	DAIEN[16] (H)	I2SF64[0] (H)	(X)
	128Fs[9] (L)			
SSI2	DAISEL[3] (L)	DAIEN[16] (L)	(X)	SERSEL[0] (L)
CODEC	DAISEL[3] (L)	DAIEN[16] (L)	(X)	SERSEL[0] (H)

Table 23. Matrix for Programming the MUX

Note: To connect the port to any of the 4 features shown above, a minimum software configuration shown in the table above must be observed. Each register column contains the bit name (bit #) that must be cleared or set for each feature as shown in the column. This table does not complete the programming for each of the features, but allows access to the port only. The interrupt masks for these features will have to be programmed as well.

128 fs Audio Bit Clock (MHz)	64 fs Audio Bit Clock (MHz)	Clock Source (MHz)	Sample Frequency (KHz)	128 fs Divisor (AUDDIV)	64 fs Divisor (AUDDIV)
1.0240	0.5120	73.728	8	36	72
1.4112	0.7056	11.2896	11.025	8	16
1.5360	0.7680	73.728	16	18	36
2.8224	1.4112	11.2896	22.025	4	8
3.0720	1.5360	73.728	24	12	24
4.0960	2.0480	73.728	32	9	18
5,6448	2.8224	11.2896	44.1	2	4
6.1440	3.0720	73.728	48	6	12

Table 24. Relationship between Audio Clocks / Clock Source / Sample Frequencies

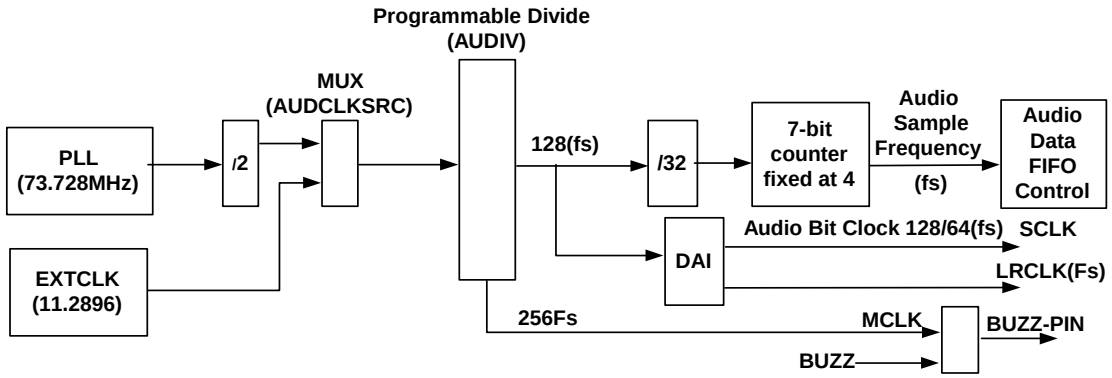


Figure 7. Digital Audio Clock Generation

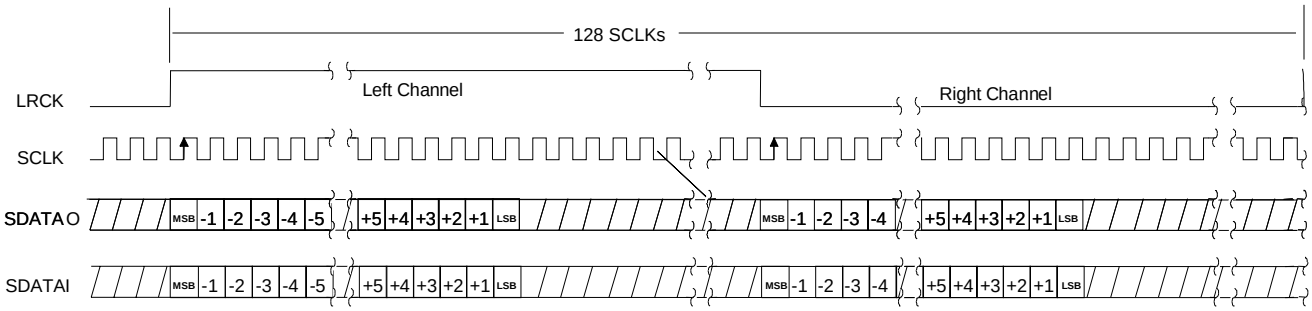


Figure 8. CS89712 - Digital Audio Interface Timing – MSB / Left Justified format

frame synchronization. Each transition of LRCK delineates the left and right halves of an audio sample. When LRCK transitions from high-to-low the next 16 bits make up the right side of a sample. When LRCK transitions from low-to-high the next 16 bits make up the left side of a sample.

2.17.2.3 DAI Signals

MCLK is used as an input to the CS89712 for generating the DAI timing. This signal is also usually used as an input to a DAC/ADC as an oversampled clock. This signal is fixed at 256 times the audio sample frequency.

The SCLKbit clock is used as the bit clock input into the DAC/ADC. This signal is fixed at 128 or 64 times the audio sample frequency.

LRCK is used as a frame synchronization input to the DAC/ADC. This signal is fixed at the audio sample frequency. This signal is clocked out on the negative going edge of SCLK.

SDOUT is used for sending playback data to a DAC. This signal is clocked out on the negative going edge of the SCLK output.

SDIN is used for receiving record data from an ADC. This signal is latched by the CS89712 on the positive going edge of SCLK.

2.17.3 ADC Interface - SSI1 Master Only

The first synchronous serial interface allows interfacing to the following peripheral devices:

- In the default mode, the device is compatible with the MAXIM MAX148/9 in external clock mode. Similar SPI- or Microwire-compatible devices can connect directly to the CS89712.
- In the extended mode and with negative-edge triggering selected (the ADCCON and ADCCKNSEN bits are set, respectively, in the SYSCON3 register), this device can be interfaced to Analog Devices' AD7811/12 chip using nADCCS as a common RFS/TFS line.
- Other features of the devices, including power management, can be utilized by software and the use of the GPIO pins.

The clock output frequency is programmable and only active during data transmissions to save power. There are four output frequencies selectable. The required frequency is selected by programming the corresponding bits 16 and 17 in the SYSCON1 register. The sample clock (SMPCLK) always runs at twice the frequency of the shift clock (ADCCLK). The output channel is fed by an 8-bit shift register when the ADCCON bit of SYSCON3 is clear. When ADCCON is set, up to 16 bits of configuration command can be sent, as specified in the SYNCIO register. The input channel is captured by a 16-bit shift register. The clock and synchronization pulses are activated by a write to the output shift register. During transfers the SSIBUSY (synchronous serial interface busy) bit in the system status flags register is set. When the transfer is complete and valid data is in the 16-bit read shift register, the SSEOTI interrupt is asserted and the SSIBUSY bit is cleared.

An additional sample clock (SMPCLK) can be enabled independently and is set at twice the transfer clock frequency.

This interface has no local buffering capability and is only intended to be used with low bandwidth interfaces, such as for a touch-screen ADC interface.

2.17.4 SSI2 with Master / Slave operation

A second SPI / Microwire interface with full master/slave capability is provided by the CS89712. Data rates in slave mode are theoretically up to 512 kbits/s, full duplex, although continuous operation at this data rate will give an interrupt rate of 2 kHz, too fast for many operating systems. This would require a worst-case interrupt response time of less than 0.5 msec and would cause loss of data through TX underruns and RX overruns.

The interface is fully capable of being clocked at 512 kHz when in slave mode. However, it is anticipated that external hardware will be used to frame the data into packets. Therefore, although the data would be transmitted at a rate of 512 kbits/s, the sustained data rate would in fact only be 85.3 kbits/s (i.e., 1 byte every 750 µsec). At this

SYSCON1 bit 17	SYSCON1 bit 16	18.432–73.728 MHz Operation ADCCLK Frequency (kHz)
0	0	4
0	1	16
1	0	64
1	1	128

Table 25. ADC Interface Operation Frequencies

data rate, the required interrupt rate will be greater than 1 msec, which is acceptable.

There are separate half-word-wide RX and TX FIFOs (16 half-words each) and corresponding interrupts which are generated when the FIFO's are half-full or half-empty as appropriate. The interrupts are called SS2RX and SS2TX, respectively. Register SS2DR is used to access the FIFOs.

There are five pins to support this SSI port: SSIRXDA, SSITXFR, SSICLK, SSITXDA, and SSIRXFR. The SSICLK, SSIRXDA, SSIRXFR, and SSITXFR signals are inputs and the SSITXDA signal is an output in slave mode. In the master mode, SSICLK, SSITXDA, SSITXFR, and SSIRXFR are outputs, and SSIRXDA is an input. Master mode is enabled by writing a one to the SS2MAEN bit (SYSCON2[9]). When the master / slave SSI is not required, it can be disabled to save power by writing a zero to the SS2TXEN and the SS2RXEN bits (SYSCON2[4] [7]). When set, these two bits independently enable the transmit and receive sides of the interface. The master/slave SSI is synchronous, full duplex, and capable of supporting serial data transfers between two nodes. Although the interface is byte-oriented, data is loaded in blocks of two bytes at a time. Each data byte to be transferred is marked by a frame sync pulse, lasting one clock period, and located one clock prior to the first bit being transferred. Direction of the SSI2 ports, in slave and master mode, is shown in Figure 9.

Data on the link is sent MSB first and coincides with an appropriate frame sync pulse, one clock in duration, located one clock prior to the first data bit (MSB). It is not possible to send data LSB first.

When operating in master mode, the clock frequency is selected to be the same as the ADC interface's (master mode only SSI1) — that is, the frequencies are selected by the same bits 16 and 17 of the SYSCON1 register (i.e., the ADCKSEL bits). Thus, the maximum frequency in master mode is 128 kbits/s. The interface will support continuous transmission at this rate assuming that the OS can respond to the interrupts within 1 msec to prevent over/underruns.

Note: To allow synchronization to the incoming slave clock, the interface enable bits will not take effect until one SSICLK cycle after they are written and the value read back from SYSCON2. The enable bits reflect the real status of the enables internally. Hence, there will be a delay before the new value programmed to the enable bits can be read back.

The timing diagram for this interface can be found in Section 6.3.

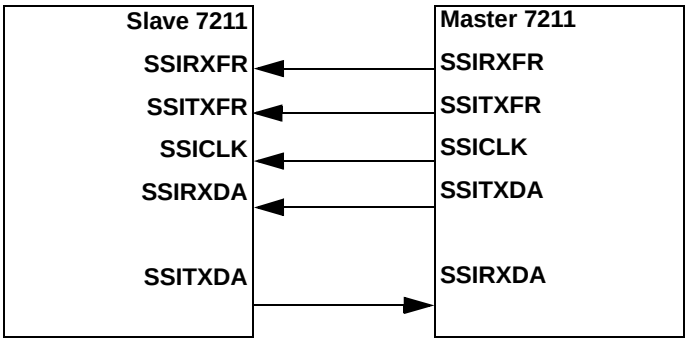


Figure 9. SSI2 Port Directions in Slave and Master Mode

2.17.4.1 Read Back of Residual Data

All writes to the transmit FIFO must be in half-words (i.e., in units of two bytes at a time). On the receive side, it is possible that an odd number of bytes will be received. Bytes are always loaded into the receive FIFO in pairs. Consequently, in the case of a single residual byte remaining at the end of a transmission, it will be necessary to read the byte separately. This is done by reading the status of two bits in the SYSFLG2 register to determine the validity of the residual data. These two bits (RESVAL, RESFRM) are both set high when a residual is valid. RESVAL is cleared on either a new transmission or on reading of the residual bit by software. RESFRM is cleared only on a new transmission. By popping the residual byte into the RX FIFO and then reading the status of these bits it is possible to determine if a residual bit has been correctly read.

Figure 10 illustrates this procedure. The sequence is as follows: read the RESVAL bit, if this is a 0, no action needs to be taken. If this is a 1, then pop the residual byte into the FIFO by writing to the SS2POP location. Then read back the two status bits RESVAL and RESFRM. If these bits read back 01, then the residual byte popped into the FIFO is valid and can be read back from the SS2DR register. If the bits are not 01, then there has been another transmission received since the residual read procedure has been started. The data item that has been popped to the top of the FIFO will be invalid and should be ignored. In this case, the correct byte will have been stored in the most significant byte of the next half-word to be clocked into the FIFO.

Note: All the writes / reads to the FIFO are done word at a time (data on the lower 16 bits is valid and upper 16 bits are ignored).

Software manually pops the residual byte into the RX FIFO by writing to the SS2POP location (the value written is ignored). This write will strobe the RX FIFO write signal, causing the residual byte to be written into the FIFO.

2.17.4.2 Support for Asymmetric Traffic

The interface supports asymmetric traffic (i.e., unbalanced data flow). This is accomplished through separate transmit and receive frame sync control lines. In operation, the receiving node receives a byte of data on the eight clocks following the assertion of the receive frame sync control line. In a similar fashion, the sending node can transmit a byte of data on the eight clocks following the assertion of the transmit frame sync pulse. There is no correlation in the frequency of assertions of the RX and TX frame sync control lines (SSITXFR and SSIRXFR). Hence, the RX path may bear a greater data throughput than the TX path, or vice versa.

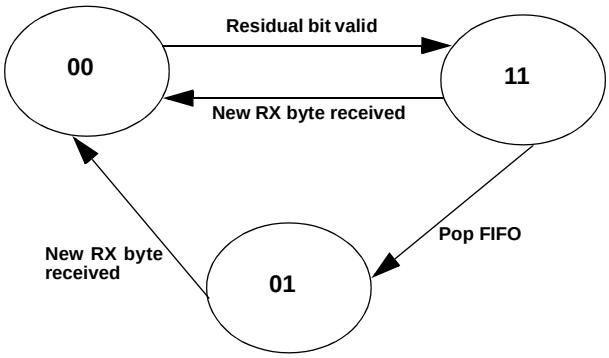


Figure 10. Residual Byte Reading

Both directions, however, have an absolute maximum data throughput rate determined by the maximum possible clock frequency, assuming that the interrupt response of the target OS is quick enough.

2.17.4.3 Continuous Data Transfer

Data bytes may be sent/received in a contiguous manner without interleaving clocks between bytes. The frame sync control line(s) are eight clocks apart and aligned with the clock representing bit D0 of the preceding byte (i.e. one bit before the MSB).

2.17.4.4 Discontinuous Clock

In order to save power during the idle times, the clock line is put into a static low state. The master is responsible for putting the link into the Idle State. The Idle State will begin one clock, or more, after the last byte transferred and will resume at least one clock prior to the first frame sync assertion. To disable the clock, the TX section is turned off.

In Master mode, the CS89712 does not support the discontinuous clock.

2.17.4.5 Error Conditions

RX FIFO overflows are detected and conveyed via a status bit in the SYSFLG2 register. This register should be accessed at periodic intervals by the application software. The status register should be read each time the RX FIFO interrupts are generated. At this time the error condition (i.e., overrun flag) will indicate that an error has occurred but cannot convey which byte contains the error. Writing to the SRXEOF register location clears the overrun flag. TX FIFO underflow condition is detected and conveyed via a bit in the SYSFLG2 register, which is accessed by the application software. A TX underflow error is cleared by writing data to be transmitted to the TX FIFO.

2.17.4.6 Clock Polarity

Clock polarity is fixed. TX data is presented on the bus on the rising edge of the clock. Data is latched into the receiving device on the falling edge of the clock. The TX pin is held in a tristate condition when not transmitting.

2.18 LCD Controller

The LCD controller provides all the necessary control signals to interface directly to a single panel multiplexed LCD.

The panel size is programmable and can be any width (line length) from 32 to 1024 pixels in 16-pixel increments. The total video frame buffer size is programmable up to 128 kbytes. This equates to a theoretical maximum panel size of 1024 x 256 pixels in 4 bits-per-pixel mode. The video frame buffer can be located in any portion of memory controlled by the chip selects. Its start address will be fixed at address 0x0000.0000 within each chip select. The start address of the LCD video frame buffer is defined in the FBADDR[3:0] register. These bits become the most significant nibble of the external address bus. The default start address is 0xC000.0000 (FBADDR = 0xC).

A system built using the on-chip SRAM (OCSR), will then serve as the LCD video frame buffer and miscellaneous data store. The LCD video frame buffer start address should be set to 0x6 in this option. Programming of the register FBADDR is only permitted when the LCD is disabled (this is to avoid possible cycle corruption when changing the register contents while a LCD DMA cycle is in progress). There is no hardware protection to prevent this. It is necessary to disable the LCD controller before reprogramming the FBADDR register. Full address decoding is provided for the OCSR, up to the maximum video frame buffer size programmable into the LCDCON register. Beyond this, the address is wrapped around. The frame buffer start address must not be programmed to 0x4 or 0x5 if either CL-PS6700 interface is in use (PCMEN1 or PCMEN2 bits in the SYSCON2 register are enabled). FBADDR should never be programmed to

0x7 or 0x8, as these are the locations for the on-chip Boot ROM and internal registers.

During Snooze State, the shift clock (CL2), FRM, and line (CL1) signals are on for the entire display. The SNZDISP register is used to disable the data path through the LCD controller after the required data has been displayed, to save further power. After the word address stored in the SNZDISP register is reached, the data output pins DD[3:0] will be blanked to 0 or 1 as defined by the SNZPOL bit, which is at Bit 10 of the SYSCON2 register. Sections of the SRAM not used for the display data in Snooze State can be used for other data storage.

In Snooze State, the LCD controller (if enabled via the LCDEN bit in the SYSCON1 register) will automatically fetch data from the on-chip SRAM in 1-bit-per-pixel mode. Before entering Snooze State, the required display buffer must be transferred into the on-chip SRAM in a 1-bit-per-pixel format. On entry to Snooze State, the video frame size field is reinterpreted for 1-bit-per-pixel data, and the grey scale mode bits are ignored.

On exit from Snooze State, the CS89712 enters the Doze State. In Doze State, all of the CS89712, except the LCD controller, is operating normally. The DRAM is taken out of self-refresh and normal CAS before RAS (CBR) refreshes start. The CPU is active and takes interrupts at normal speed. In Doze State, display data continues to be fetched from the OCSR, as for the Snooze State. DMA for the display is active only while the number of lines programmed into the SNZDISP register are displayed and DMA/CPU arbitration is carried out during this time. For the rest of the time, the LCD controller displays “pixel fill” data on the LCD. During the Doze State, if some of the OCSR memory space is not being used to store the video buffer, the remaining section can be used by the CPU for general purpose data storage. The remaining section is fully address decoded.

Note: The only way to enter the Doze State is by exit from the Snooze State. Also, the Snooze State cannot exit directly to the Operating State; it must go through the Doze State first.

In an application, the CS89712 would spend most of its time in snooze mode. On interrupt or wake-up, it moves into Doze State. At this point the OS identifies the cause of the interrupt and decides whether it can stay in Doze State (e.g., update a clock on the display) or if it is woken up because the user wants to perform a function requiring the full display. In the latter case, the OS will set the LCDSNZE bit low and the CS89712 will switch to the Operating State. In the Operating State, the display will be automatically switched to the main frame buffer on the next frame sync. The full LCD controller is used and data fetched from the buffer pointed to by the FBADDR register (if LCDSNZE is low). To ensure correct synchronization it is not possible to program the LCDSNZE bit to high with software, this is done automatically as part of the process of entering the Snooze State. It can, however, be set low from software. This is how the display is changed back to the main display after exit from the Snooze State, if this is required.

It is likely that system software would normally wake up from the Snooze State for two reasons: 1) to perform minor OS functions like updating a time display or polling the keyboard, and 2) to wake up completely because the user has pressed a key. In the former case, the LCDSNZE bit would be left high and the Snooze State re-entered by writing to the SNOOZE location in the normal way. The chip would continue to output data from the on-chip SRAM throughout (which could have been updated while out of Snooze State). In the second case, software should write to the LCDSNZE location soon after exiting from Snooze State. Then the LCD controller will be re-enabled and the display cleanly switched across to the main frame buffer as pointed to by the address in the FBADDR register.

The screen is mapped to the video frame buffer as one contiguous block where each horizontal line of

pixels is mapped to a set of consecutive bytes or words in the video RAM. The video frame buffer can be accessed word wide as pixel 0 is mapped to the LSB in the buffer such that the pixels are arranged in a little endian manner.

The pixel bit rate, and hence the LCD refresh rate, can be programmed from 18.432 MHz to 576 kHz. The LCD controller is programmed by writing to the LCD control register (LCDCON). The LCDCON register should not be reprogrammed while the LCD controller is enabled.

The LCD controller also contains two 32-bit palette registers, which allow any 4-, 2-, or 1-bit pixel value to be mapped to any of the 15 grayscale values available. The palette registers are bypassed in Snooze State.

The required DMA bandwidth to support a ½ VGA panel displaying 4 bits-per-pixel data at an 80 Hz refresh rate is approximately 6.2 Mbytes/sec. Assuming the frame buffer is stored in a 32-bit wide the maximum theoretical bandwidth available is 86 Mbytes/sec at 36.864 MHz.

The LCD controller uses a nine stage 32-bit wide FIFO to buffer display data. The LCD controller requests new data when there are five words remaining in the FIFO. This means that for a ½ VGA display at 4 bits-per-pixel and 80 Hz refresh rate, the maximum allowable DMA latency is approximately 3.25 µsec ((5 words x 8 bits/byte) / (640 x 240 x 4 bpp x 80 Hz)) = 3.25 µsec). The worst-case latency is the total number of cycles from when the DMA request appears to when the first DMA data word actually becomes available at the FIFO. DMA has the highest priority, so it will always happen next in the system. The maximum number of cycles required is 36 from the point at which the DMA request occurs to the point at which the STM is complete, then another 6 cycles before the data actually arrives at the FIFO from the first DMA read. This creates a total of 42 cycles assuming the frame buffer is located in 32-bit wide memory. With 16-bit wide memory, the worst-case latency will double. In this case, the maximum permissible display size may be halved, to approximately 320 x 240 pixels, depending on required pixel depth and refresh rate. If 18 MHz mode is selected with 32-bit wide memory, then the worst-case latency will be 2.26 µsec (i.e., 42 cycles x 54 nsec/cycle). If 36 MHz mode is selected, and 32-bit wide, then the worst-case latency drops down to 1.49 µs. If the frame buffer is to be stored in static memory, then further calculations must be performed. This calculation is a little more complex for 36 MHz mode of operation. The total number of cycles = (12 x 4) + 7 = 55. Thus, 55 x 27 ns = ~1.49 µsec.

Figure 11 shows the organization of the video map for all combinations of bits-per-pixel.

The refresh rate is not affected by the number of bits-per-pixel; however the LCD controller fetches twice the data per refresh for 4 bits-per-pixel compared to 2 bits-per-pixel. The main reason for reducing the number of bits-per-pixel is to reduce the power consumption of the memory where the video frame buffer is mapped.

2.19 Timer Counters

Two identical timer counters are integrated into the CS89712. These are referred to as TC1 and TC2. Each timer counter has an associated 16-bit read / write data register and some control bits in the system control register. Each counter is loaded with the value written to the data register immediately. This value will then be decremented on the second active clock edge to arrive after the write (i.e., after the first complete period of the clock). When the timer counter under flows (i.e., reaches 0), it will assert its appropriate interrupt. The timer counters can be read at any time. The clock source and mode are selectable by writing to bits in the system control register. 512 kHz and 2 kHz rates are provided.

The timer counters can operate in two modes: free running or pre-scale.

2.19.1 Free Running Mode

In the free running mode, the counter will wrap around to 0xFFFF when it under flows and it will continue to count down. Any value written to TC1 or TC2 will be decremented on the second edge of the selected clock.

2.19.2 Prescale Mode

In the prescale mode, the value written to TC1 or TC2 is automatically re-loaded when the counter under flows. Any value written to TC1 or TC2 will be decremented on the second edge of the selected clock. This mode can be used to produce a programmable frequency to drive the buzzer (i.e., with TC1) or generate a periodic interrupt. The formula is F=(500 kHz) / (n+1).

2.20 Real-Time Clock

The CS89712 contains a 32-bit Real-Time Clock (RTC). This can be written to and read from in the

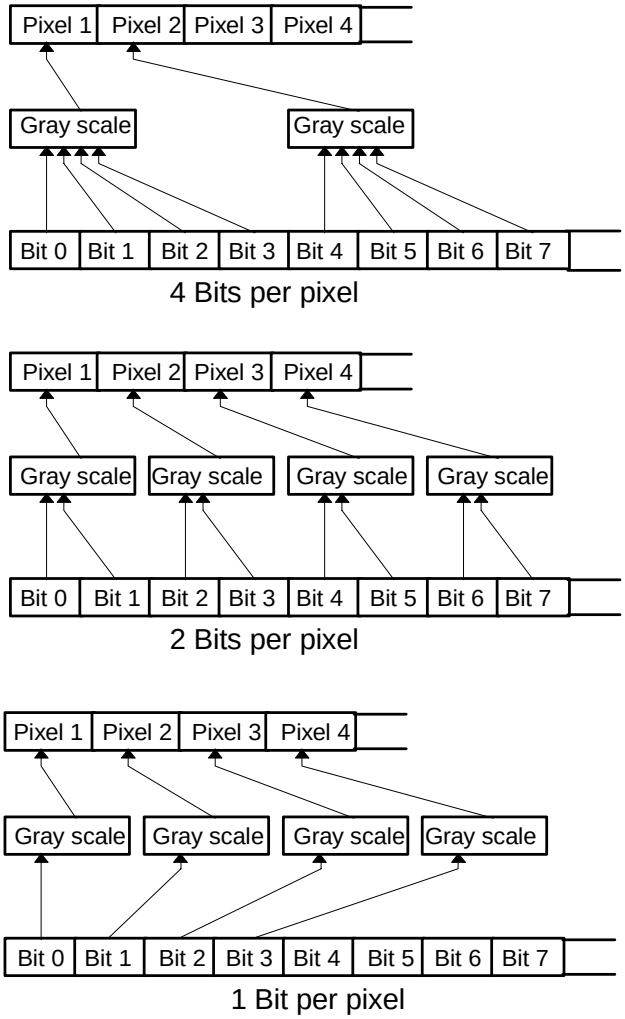


Figure 11. Video Buffer Mapping

same way as the timer counters, but is 32 bits wide. The RTC is always clocked at 1 Hz, generated from the 32.768 kHz oscillator. It also contains a 32-bit output match register, this can be programmed to generate an interrupt when the count in the RTC matches a specific value written to this register. The RTC can only be reset by an nPOR cold reset. Because the RTC data register is updated from the 1 Hz clock derived from the 32 kHz source, which is asynchronous to the main memory system clock, the data register should always be read twice to ensure a valid and stable reading. This also applies when reading back the RTCDIV field of the SYSCON1 register, which reflects the status of the six LSBs of the RTC counter.

2.20.1 RTC Interface Characteristics

When connecting a crystal to the RTC interface pins (i.e., RTCIN and RTCOUT), the crystal and circuit should conform to the following:

- The 32.768 kHz frequency should be created by the crystals fundamental tone (i.e., it should be a fundamental mode crystal)
- A start-up resistor is not necessary, since one is provided internally.
- Start-up loading capacitors may be placed on each side of the external crystal and ground. Their value should be in the range of 10 pF. However, their values should be selected based upon the crystal specifications. The total sum of the capacitance of the traces between the CS89712's clock pins, the capacitors, and the crystal leads should be subtracted from the crystal's specifications when determining the values for the loading capacitors.
- The crystal should have a maximum 5 ppm frequency drift over the chip's operating temperature range.
- The voltage for the crystal must be 2.5 V $\pm$ 0.2 V.

Alternatively, a digital clock source can be used to drive the RTCIN pin of the CS89712. With this approach, the voltage levels of the clock source should match that of the V_{DD} supply for the CS89712's pads (i.e., the supply voltage level used to drive all of the non-V_{DD} core pins on the CS89712) (i.e., RTCOUT). The output clock pin should be left floating.

2.21 Dedicated LED Flasher

The LED flasher feature enables an external pin (PD[0] / LEDFLSH) to be toggled at a programmable rate and duty ratio for connection to an LED. This module is driven from the RTCs 32.768 kHz oscillator and works in all running modes because no CPU intervention is needed once its rate and duty ratio have been configured (via the LEDFLSH register). The LED flash rate period can be programmed for 1, 2, 3, or 4 seconds. The duty ratio can be programmed such that the mark portion can be 1/16 to 16/16 of the full cycle.

2.22 PWM Interfaces

Two Pulse Width Modulator (PWM) duty ratio clock outputs are provided in the CS89712. When the device is operating from the internal PLL, the PWM will run at a frequency of 96 kHz. These signals are intended for use as drives for external DC-to-DC converters in the Power Supply Unit (PSU) subsystem. External input pins that would normally be connected to the output from comparators monitoring the external DC-to-DC converter output are also used to enable these clocks. These are the FB[1:0] pins. The duty ratio (and hence PWMs on time) can be programmed from 1 in 16 to 15 in 16. The sense of the PWM drive signal (active high or low) is determined by latching the state of this drive signal during power on reset (i.e., a pull-up on the drive signal will result in a active low drive output, and visa versa). This allows either positive or negative voltages to be generated by the external DC-to-DC converter.The DC to DC converter channels remain enabled in Snooze State. In

Snooze State these are the only peripherals apart from the LCD controller and on-chip SRAM to remain enabled. If either or both of the converters are not required, they should be switched off before entering Snooze State to save power. PWMs are disabled by writing zeros into the drive ratio fields in the PMPCON Pump Control register.

Note: To maximize power savings, the drive ratio fields should be used to disable the PWMs, instead of the FB pins. The clocks that source the PWMs are disabled when the drive ratio fields are zeroed.

2.23 Ethernet Port Overview

The Ethernet port provides a flexible set of performance features and configuration options, allowing designers to develop Ethernet circuits that meet their system requirements.

The Ethernet Port performs two basic functions: Ethernet packet transmission and reception. Before transmission or reception is possible, the Ethernet Port must be configured.

2.23.1 Configuration

The Ethernet port must be configured for packet transmission and reception at power-up or reset. Parameters must be written to its internal Configuration and Control registers. Configuration data can either be written to the Ethernet port by software or loaded automatically from an external EEPROM.

Section 2.24, *"Programming the EEPROM"* and Section 2.6, *"Ethernet EEPROM Configurations"* describe the configuration process in detail. Section 3.2.3, *"Ethernet Status/Control Registers"* provides a detailed description of the bits in the Configuration and Control Registers.

2.23.2 Packet Transmission

Packet transmission occurs in two phases. In the first phase, the Ethernet frame is moved into the Ethernet port's buffer memory. The first phase begins with the issuance of a Transmit Command. This informs the Ethernet port both that a frame is to be transmitted and when to start transmission (i.e. after 5, 381, 1021 or all bytes have been transferred). Following the Transmit Command is the Transmit Length, indicating how much buffer space is required. When buffer space is available, the Ethernet frame is written into the Ethernet port's internal memory.

In the second phase of transmission, the Ethernet port converts the frame into an Ethernet packet then transmits it onto the network. The second phase begins with the Ethernet port transmitting the preamble and Start-of-Frame delimiter as soon as the proper number of bytes has been transferred into its transmit buffer (5, 381, 1021 bytes or full frame, depending on configuration). The preamble and Start-of-Frame delimiter are followed by the Destination Address, Source Address, Length field and LLC data (all software supplied). If the frame is less than 64 bytes, including CRC, the Ethernet port adds pad bits if so configured. Finally, the Ethernet port appends the proper 32-bit CRC value.

Section 2.34, *"Transmit Operation"* provides a detailed description of packet transmission.

2.23.3 Packet Reception

Like packet transmission, packet reception occurs in two phases. In the first phase, the Ethernet port receives an Ethernet packet and stores it in on-chip memory. The first phase of packet reception begins with the receive frame passing through the analog front end and Manchester decoder where Manchester data is converted to NRZ data. Next, the preamble and Start-of-Frame delimiter are stripped off and the receive frame is sent through the address filter. If the frame's Destination Address matches the criteria programmed into the address filter, the packet is stored in the Ethernet port's internal memory. The Ethernet port then checks the CRC, and depending on the configuration, informs the processor that a frame has been received. In the second phase, the receive frame is transferred into host memory.

Section 2.32, *"Basic Receive Operation"* and Section 2.32.7, *"Receive Ethernet Port Locations"* provide a detailed description of packet reception.

2.24 Programming the EEPROM

After initialization, software can access the EEPROM through the Ethernet port by writing one of seven commands to the EEPROM Command register. Figure 12 shows the format of the EEPROM Command register.

2.24.1 EEPROM Commands

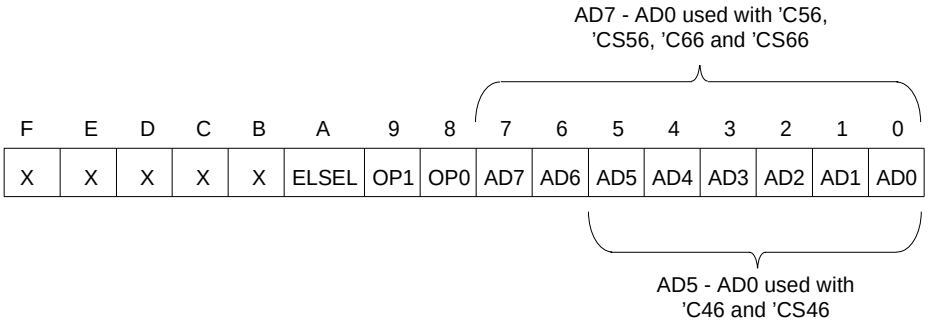
The seven commands used to access the EEPROM are: Read, Write, Erase, Erase/Write Enable, Erase/Write Disable, Erase-All, and Write-All. They are described in Table 26.

2.24.2 EEPROM Command Execution

During the execution of a command, the two Opcode bits, followed by the six bits of address (for a 'C46 or 'CS46) or eight bits of address (for a 'C56, 'CS56, 'C66 or 'CS66), are shifted out of the Ethernet port, into the EEPROM. If the command is a Write, the data in the EEPROM Data register (Ethernet Port offset address 0042h) follows. If the command is a Read, the data in the specified EEPROM location is written into the EEPROM Data register. If the command is an Erase or Erase-All, no data is transferred to or from the EEPROM Data register. Before issuing any command, the SIBUSY bit (Register 16, SelfST, bit 8) must clear. After each command has been issued, software must wait again for SIBUSY to clear.

2.24.3 Enabling Access to the EEPROM

The Erase/Write Enable command provides protection from accidental writes to the EEPROM. The software must write an Erase/Write Enable command before it attempts to write to or erase any EEPROM memory location. Once the software has finished altering the contents of the EEPROM, it must write an Erase/Write Disable command to prevent unwanted modification of the EEPROM.



Bit	Name	Description
[F:B]		Reserved
[A]	ELSEL	External Logic Select: When clear, the EECS pin is used to select the EEPROM. When set, the ELCS pin is used to select the external LA decode circuit.
[9:8]	OP1, OP0	Opcode: Indicates what command is being executed (see next section).
[7:0]	AD7 to AD0	EEPROM Address: Address of EEPROM word being accessed.

Figure 12. EEPROM Command Register Format

Command	Opcode (bits 9,8)	EEPROM Address (bits 7 to 0)	Data	EEPROM Type	Execution Time
Read Register	1,0	word address	yes	all	25 µs
Write Register	0,1	word address	yes	all	10 ms
Erase Register	1.1	word address	no	all	10 ms
Erase/Write Enable	0,0	XX11-XXXX	no	‘CS46, ‘C46	9 µs
		11XX-XXXX	no	‘CS56, ‘C56, ‘CS66, ‘C66	9 µs
Erase/Write Disable	0,0 0,0	XX00-XXXX	no	‘CS46, ‘C46	9 µs
		00XX-XXXX	no	‘CS56, ‘C56, ‘CS66, ‘C66	9 µs
Erase-All Registers	0,0 0,0	XX10-XXXX	no	‘CS46, ‘C46	10 ms
		10XX-XXXX	no	‘CS56, ‘C56, ‘CS66, ‘C66	9 µs
Write-All Register	0,0 0,0	XX01-XXXX	yes	‘CS46, ‘C46	10 ms
		01XX-XXXX	yes	‘CS56, ‘C56, ‘CS66, ‘C66	10 ms

Table 26. EEPROM Commands

2.24.4 Writing and Erasing the EEPROM

To write data to the EEPROM, the software must execute the following series of commands:

1) Issue an Erase/Write Enable command.
2) Load the data into the EEPROM Data register.
3) Issue a Write command.
4) Issue an Erase/Write Disable command.

During the Erase command, the Ethernet port writes FFh to the specified EEPROM location. During the Erase-All command, the Ethernet port writes FFh to all locations.

2.25 Ethernet LEDs

The Ethernet port provides three output pins that can be used to control LEDs or external logic.

2.25.1 LANLED

$\overline{\text{LANLED}}$ goes low whenever the Ethernet port transmits or receives a frame, or when it detects a collision. $\overline{\text{LANLED}}$ remains low until there has been no activity for 6 ms (i.e. each transmission, reception, or collision produces a pulse lasting a minimum of 6 ms).

2.25.2 $\overline{\text{LINKLED}}$ or $\overline{\text{HC0}}$

$\overline{\text{LINKLED}}$ or $\overline{\text{HC0}}$ can be controlled by either the Ethernet port or the software. When controlled by the Ethernet port, $\overline{\text{LINKLED}}$ is low whenever the Ethernet port receives valid 10BASE-T link pulses. To configure this pin for software control, the HC0E bit (SelfCTL register, Bit C) must be clear. When controlled by the software, $\overline{\text{LINKLED}}$ is low when the HCB0 bit (SelfCTL register, Bit E) is set. To configure it for software control, the HC0E bit must be set. Table 27 summarizes this operation.

HC0E (Bit C)	HCB0 (Bit E)	Pin Function
0	N/A	Pin configured as $\overline{\text{LINKLED}}$: Output is low when valid 10BASE-T link pulses are detected. Output is high if valid link pulses are not detected
1	0	Pin configured as $\overline{\text{HC0}}$: Output is high
1	1	Pin configured as $\overline{\text{HC0}}$: Output is low

Table 27. $\overline{\text{LINKLED}}$/$\overline{\text{HC0}}$ Pin Operation

2.25.3 LED Connection

Each LED output is capable of sinking 10 mA to drive an LED directly through a series resistor. The output voltage of each pin is less than 0.4 V when the pin is low.

2.26 Media Access Control Engine

2.26.1 Overview

The CS89712's Ethernet Media Access Control (MAC) engine is fully compliant with the IEEE 802.3 Ethernet standard (ISO/IEC 8802-3, 1993). It handles all aspects of Ethernet frame transmission and reception, including: collision detection, preamble generation and detection, and CRC generation and test. Programmable MAC features include automatic retransmission on collision, and padding of transmitted frames. The primary functions of the MAC are: frame encapsulation and decapsulation; error detection and handling; and, media access management.

2.26.2 Frame Encapsulation/Decapsulation

The Ethernet port's MAC engine automatically assembles transmit packets and disassembles receive packets. It also determines if transmit and receive frames are of legal minimum size.

2.26.2.1 Transmission

Once the proper number of bytes have been transferred to the Ethernet port's memory (either 5, 381, 1021 bytes, or full frame), and providing that access to the network is permitted, the MAC automatically transmits the 7-byte preamble (1010101b...), followed by the Start-of-Frame Delimiter (SFD, 10101011b), and then the serialized frame data. It then transmits the Frame Check Sequence (FCS). The data after the SFD and before the FCS (Destination Address, Source Address, Length, and data field) is supplied by the software. FCS generation by the Ethernet port may be disabled by setting the InhibitCRC bit (TxCMD register, bit C).

Figure 13 shows the Ethernet frame format.

2.26.2.2 Reception

The MAC receives the incoming packet as a serial stream of NRZ data from the Manchester encoder/decoder. It begins by checking for the SFD. Once the SFD is detected, the MAC assumes all subsequent bits are frame data. It reads the DA and compares it to the criteria programmed into the address filter (see Section 2.32.7, *"Receive Ethernet Port Locations"* for a description of Address Filtering). If the DA passes the address filter, the frame is loaded into the Ethernet port's memory. If the BufferCRC bit (RxCFG register, bit B) is set, the received FCS is also loaded into memory. Once the entire packet has been received, the MAC validates the FCS. If an error is detected, the CRCerror bit (RxEvent register, Bit C) is set.

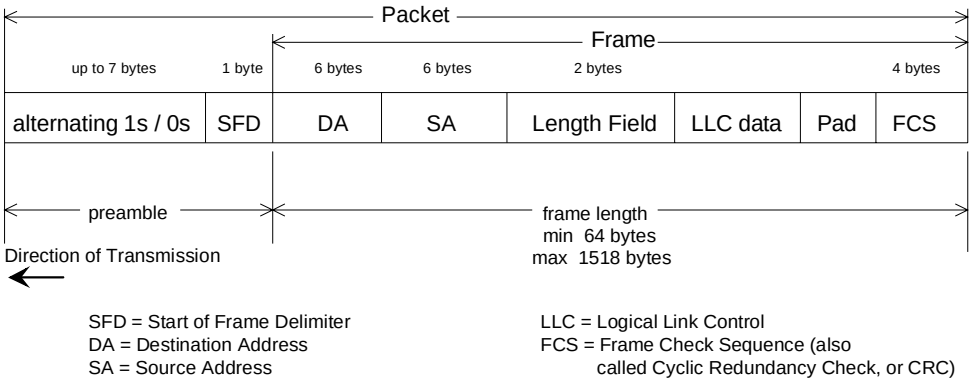


Figure 13. Ethernet Frame Format

2.26.2.3 Enforcing Minimum Frame Size

The MAC provides minimum frame size enforcement of both transmit and receive packets. When the TxPadDis bit (TxCMD register, Bit D) is clear, transmit frames will be padded with additional bits to ensure that the receiving station receives a legal frame (64 bytes, including CRC). When TxPadDis is set, the Ethernet port will not add pad bits and will transmit frames less that 64 bytes. If a frame is received that is less than 64 bytes (including CRC), the Runt bit (RxEvent register, Bit D) will be set indicating the arrival of an illegal frame.

2.26.3 Transmit Error Detection/Handling

The MAC engine monitors Ethernet activity and reports and recovers from a number of error conditions. For transmission, the MAC reports the following errors in TxEvent (Register 8) and BufEvent (Register C):

2.26.3.1 Out-of-Window (Late) Collision

If a collision is detected after the first 512 bits have been transmitted, the MAC reports a late collision by setting the Out-of-window bit (TxEvent register, Bit 9). The MAC then forces a bad CRC and terminates the transmission. If the Out-of-windowiE bit (TxCFG register, Bit 9) is set, an interrupt is generated. A late collision may indicate an illegal network configuration.

2.26.3.2 Jabber Error

If a transmission continues longer than about 26 ms, the MAC disables the transmitter and sets the Jabber bit (TxEvent register, Bit A). The output of the transmitter returns to idle and remains there until the software issues a new Transmit Command. If the JabberiE bit (TxCFG register, Bit A) is set, an interrupt is generated. A Jabber condition indicates a possible error in the Ethernet port transmit function. To prevent possible network faults, the software should clear the transmit buffer. Possible options include:

- Reset the chip with either software or hardware reset (see Section 2.24, *"Programming the EEPROM"*).
- Issue a Force Transmit Command by setting the Force bit (TxCMD register, bit 8).
- Issue a Transmit Command with the TxLength field set to zero.

2.26.3.3 Transmit Collision

The MAC counts the number of times an individual packet must be retransmitted due to network collisions. The collision count is stored in bits B through E of the TxEvent register. If the packet collides 16 times, transmission of that packet is terminated and the 16coll bit (TxEvent register, Bit F) is set. If the 16colliE bit (TxCFG register, Bit F) is set, an interrupt is generated on the 16th collision. A running count of transmit collisions is recorded in the TxCOL register.

2.26.3.4 Transmit Underrun

If the Ethernet port starts transmission of a packet but runs out of data before reaching the end of frame, the TxUnderrun bit (BufEvent register, Bit 9) is set. The MAC then forces a bad CRC and terminates the transmission. If the TxUnderruniE bit (BufCFG bit 9) is set, an interrupt is generated.

2.26.4 Receive Error Detection/Handling

The following receive errors are reported in the RxEvent register:

2.26.4.1 CRC Error

If a frame is received with a bad CRC, the CRCerror bit (RxEvent register, Bit C) is set. If the CRCerrorA bit (RxCTL register, Bit C) is set, the frame will be buffered by Ethernet port. If the CRCerroriE bit (RxCFG register. Bit C) is set, an interrupt is generated.

2.26.4.2 Runt Frame

If a frame is received that is shorter than 64 bytes, the Runt bit (RxEvent register, Bit D) is set. If the RuntA bit (RxCTL register, Bit D) is set, the frame will still be buffered by Ethernet port. If the RuntiE bit (RxCFG bit D) is set, an interrupt is generated.

2.26.4.3 Extra Data

If a frame is received that is longer than 1518 bytes, the Extradata bit (RxEvent register, Bit E) is set. If the ExtradataA bit (RxCTL register, Bit E) is set, the first 1518 bytes of the frame will still be buffered by Ethernet port. If the ExtradataiE bit (RxCFG register Bit E) is set, an interrupt is generated.

2.26.4.4 Dribble Bits and Alignment Error

Under normal operating conditions, the MAC may detect up to 7 additional bits after the last full byte of a receive packet. These bits, known as dribble bits, are ignored. If dribble bits are detected, the Dribblebit bit (RxEvent register, Bit 7) is set. If both the Dribblebits bit and CRCerror bit (RxEvent register Bit C) are set at the same time, an alignment error has occurred.

2.26.5 Media Access Management

The Ethernet network topology is a single shared medium with several attached stations. The Ethernet protocol is designed to allow each station equal access to the network at any given time. Any node can attempt to gain access to the network by first completing a deferral process (described below) after the last network activity, and then transmitting a packet that will be received by all other stations. If two nodes transmit simultaneously, a collision occurs and the colliding packets are corrupted. Two primary tasks of the MAC are to avoid network collisions, and then recover when they occur.

2.26.5.1 Collision Avoidance

The MAC continually monitors network traffic by checking for the presence of carrier activity (carrier activity is indicated by the assertion of the internal Carrier Sense signal generated by the ENDEC). If carrier activity is detected, the network is assumed busy and the MAC must wait until the current packet is finished before attempting transmission. The Ethernet port supports two schemes for determining when to initiate transmission: Two-Part Deferral, and Simple Deferral. Selection of the deferral scheme is determined by the 2-partDefDis bit (LineCTL register bit D). If the 2-partDefDis bit is clear, the MAC uses a two-part deferral process defined in section 4.2.3.2.1 of the Ethernet standard (ISO/IEC 8802-3, 1993). If the 2-partDefDis bit is set, the MAC uses a simplified deferral scheme. Both schemes are described below:

2.26.5.2 Two-Part Deferral

In the two-part deferral process, the 9.6 µs Inter Packet Gap (IPG) timer is started whenever the internal Carrier Sense signal is deasserted. If activity is detected during the first 6.4 µs of the IPG timer, the timer is reset and then restarted once the activity has stopped. If there is no activity during the first 6.4 µs of the IPG timer, the IPG timer is allowed to time out (even if network activity is detected during the final 3.2 µs). The MAC then begins transmission if a transmit packet is ready and if it is not in Backoff (Backoff is described later in this section). If no transmit packet is pending, the MAC continues to monitor the network. If activity is detected before a transmit frame is ready, the MAC defers to the transmitting station and resumes monitoring the network.

The two-part deferral scheme was developed to prevent the possibility of the IPG being shortened due to a temporary loss of carrier. Figure 14 diagrams the two-part deferral process.

2.26.5.3 Simple Deferral

In the simple deferral scheme, the IPG timer is started whenever Carrier Sense is deasserted. Once the IPG timer is finished (after 9.6 µs), if a transmit

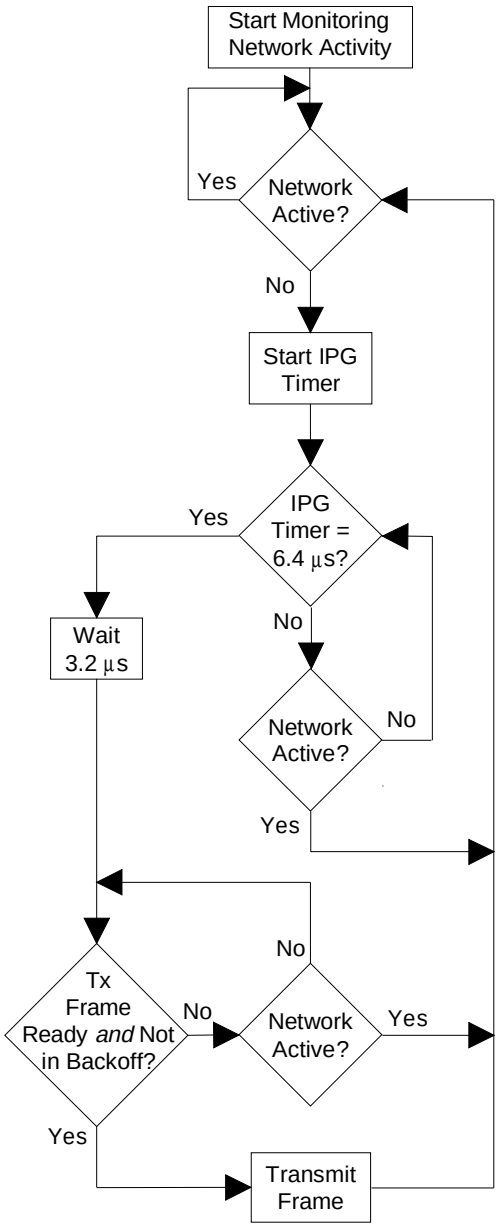


Figure 14. Two-Part Deferral

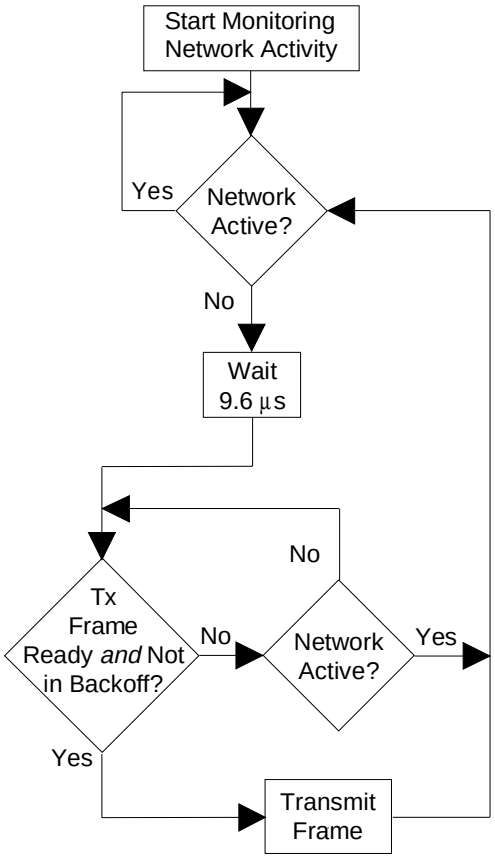


Figure 15. Simple Deferral

frame is pending and if the MAC is not in Backoff, transmission begins the 9.6 µs IPG). If no transmit packet is pending, the MAC continues to monitor the network. If activity is detected before a transmit frame is ready, the MAC defers to the transmitting station and resumes monitoring the network. Figure 15 diagrams the simple deferral process.

2.26.5.4 Collision Resolution

If a collision is detected while the Ethernet port is transmitting, the MAC responds in one of three ways depending on whether it is a normal collision (within the first 512 bits of transmission) or a late collision (after the first 512 bits of transmission):

2.26.5.5 Normal Collisions

If a collision is detected before the end of the preamble and SFD, the MAC finishes the preamble and SFD, transmits the jam sequence (32-bit pattern of all 0's), and then initiates Backoff. If a collision is detected after the transmission of the preamble and SFD but before 512 bit times, the MAC immediately terminates transmission, transmits the jam sequence, and then initiates Backoff. In either case, if the Onecoll bit (TxCMD register bit 9) is clear, the MAC will attempt to transmit a packet a total of 16 times (the initial attempt plus 15 retransmissions) due to normal collisions. On the 16th collision, it sets the 16coll bit (TxEvent register bit F) and discards the packet. If the Onecoll bit is set, the MAC discards the packet without attempting any retransmission.

2.26.5.6 Late Collisions

If a collision is detected after the first 512 bits have been transmitted, the MAC immediately terminates transmission, transmits the jam sequence, discards the packet, and sets the Out-of-window bit (TxEvent bit 9). The Ethernet port does not initiate backoff or attempt to retransmit the frame. For additional information about Late Collisions, see Section 2.26.3.1, *"Out-of-Window (Late) Collision"* in this section.

2.26.5.7 Backoff

After the MAC has completed transmitting the jam sequence, it must wait, or "Back off", before attempting to transmit again. The amount of time it must wait is determined by one of two Backoff algorithms: the Standard Backoff algorithm (ISO/IEC 4.2.3.2.5) or the Modified Backoff algorithm. The algorithm used is selected by the ModBackoffE bit (LineCTL register bit B).

2.26.5.8 Standard Backoff

The Standard Backoff algorithm, also called the "Truncated Binary Exponential Backoff", is described by the equation:

$$0 \le r \le 2^k$$

where r (a random integer) is the number of slot times the MAC must wait (1 slot time = 512 bit times), and k is the smaller of n or 10, where n is the number of retransmission attempts.

2.26.5.9 Modified Backoff

The Modified Backoff is described by the equation:

$$0 \le r \le 2^k$$

where r (a random integer) is the number of slot times the MAC must wait, and k is 3 for $n < 3$ and k is the smaller of n or 10 for $n \ge 3$, where n is the number of retransmission attempts.

The advantage of the Modified Backoff algorithm over the Standard Backoff algorithm is that it reduces the possibility of multiple collisions on the first three retries. The disadvantage is that it extends the maximum time needed to gain access to the network for the first three retries.

The software may choose to disable the Backoff algorithm altogether by setting the DisableBackoff bit (TestCTL register bit B). When disabled, the Ethernet port only waits the 9.6 µs IPG time before starting transmission.

2.27 Encoder/Decoder (ENDEC)

The Ethernet port's integrated encoder/decoder (ENDEC) circuit is compliant with the relevant portions of section 7 of the Ethernet standard (ISO/IEC 8802-3, 1993). Its primary functions include: Manchester encoding of transmit data; informing the MAC when valid receive data is present (Carrier Detection); and, recovering the clock and NRZ data from incoming Manchester-encoded data.

Figure 16 provides a diagram of the ENDEC and its interface to the MAC and 10BASE-T transceiver.

2.27.1 Encoder

The encoder converts NRZ data from the MAC and a 20 MHz Transmit Clock signal into a serial stream of Manchester data. The Transmit Clock is produced by an on-chip oscillator circuit that is driven by either an external 20 MHz quartz crystal or a TTL-level CMOS clock input. The encoded signal is routed to the 10BASE-T transceiver.

2.27.2 Carrier Detection

The internal Carrier Detection circuit informs the MAC that valid receive data is present by asserting the internal Carrier Sense signal as soon it detects a valid bit pattern (1010b or 0101b for 10BASE-T. During normal packet reception, Carrier Sense remains asserted while the frame is being received, and is deasserted 1.3 to 2.3 bit times after the last low-to-high transition of the End-of-Frame (EOF) sequence. Whenever the receiver is idle (no receive activity), Carrier Sense is deasserted. The CRS bit (LineST register bit E) reports the state of the Carrier Sense signal.

2.27.3 Clock and Data Recovery

When the receiver is idle, the phase-lock loop (PLL) is locked to the internal clock signal. The assertion of the Carrier Sense signal interrupts the PLL. When it restarts, it locks on the incoming data. The receive clock is then compared to the incoming data at the bit cell center and any phase difference is corrected. The PLL remains locked as long as the receiver input signal is valid. Once the PLL has locked on the incoming data, the ENDEC converts the Manchester data to NRZ and passes the decoded data and the recovered clock to the MAC for further processing.

2.28 10BASE-T Transceiver

The Ethernet port includes an integrated 10BASE-T transceiver that is compliant with the relevant portions of section 14 of the Ethernet standard (ISO/IEC 8802-3, 1993). It includes all analog and digital circuitry needed to interface the Ethernet port directly to a simple isolation transformer (see the Characteristics/Specifications section for a connection diagram). Figure 17 provides a block diagram of the 10BASE-T transceiver.

2.28.1 10BASE-T Filters

The CS89712's 10BASE-T transceiver includes integrated low-pass transmit and receive filters, elim-

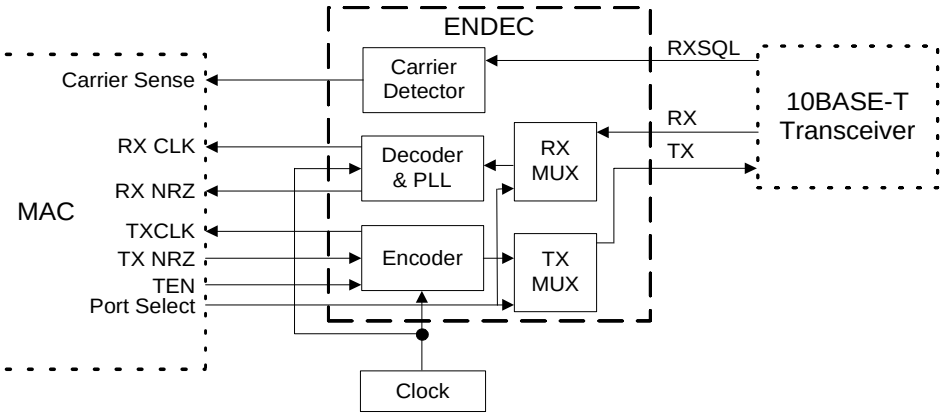


Figure 16. ENDEC

inating the need for external filters or a filter/transformer hybrid. On-chip filters are gm/c implementations of fifth-order Butterworth low-pass filters. Internal tuning circuits keep the gm/c ratio tightly controlled, even when large temperature, supply, and IC process variations occur. The nominal 3 dB cutoff frequency of the filters is 16 MHz, and the nominal attenuation at 30 MHz (3rd harmonic) is -27 dB.

2.28.2 Transmitter

When configured for 10BASE-T operation, Manchester encoded data from the ENDEC is fed into the transmitter's predistortion circuit where initial wave shaping and preequalization is performed. The output of the predistortion circuit is fed into the transmit filter where final wave shaping occurs and unwanted noise is removed. The signal then passes to the differential driver where it is amplified and driven out of the TXD+/TXD- pins.

In the absence of transmit packets, the transmitter generates link pulses in accordance with section 14.2.1.1. of the Ethernet standard. Transmitted link pulses are positive pulses, one bit time wide, typically generated at a rate of one every 16 ms. The 16 ms timer starts whenever the transmitter completes an End-of-Frame (EOF) sequence. Thus, there is a link pulse 16 ms after an EOF unless there is another transmitted packet. Figure 18 diagrams the operation of the Link Pulse Generator.

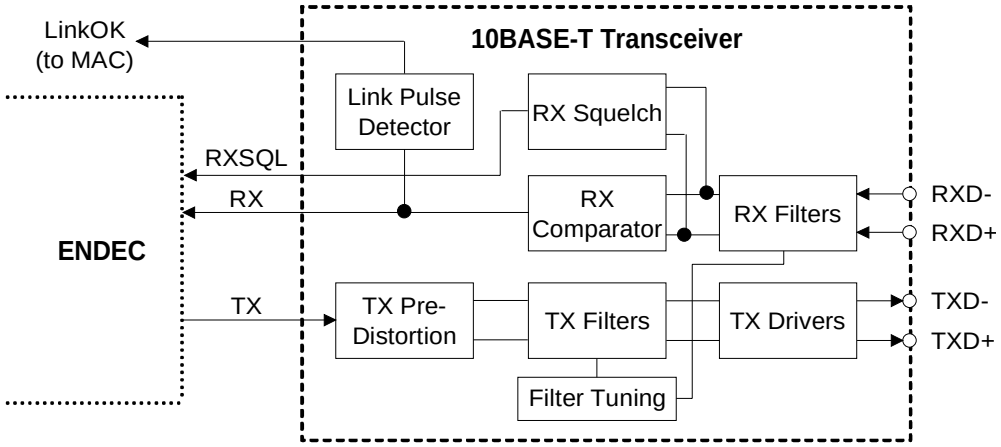


Figure 17. 10BASE-T Transceiver

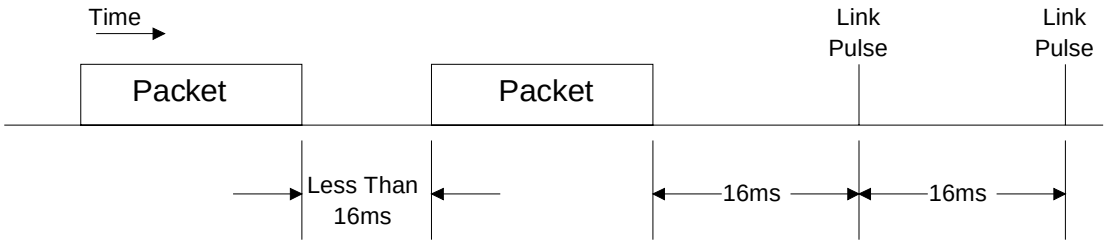


Figure 18. Link Pulse Transmission

If no link pulses are being received on the receiver, the 10BASE-T transmitter is internally forced to an inactive state unless bit DisableLT in the Test Control register is set to one.

2.28.3 Receiver

The 10BASE-T receive section consists of the receive filter, squelch circuit, polarity detection and correction circuit, and link pulse detector.

2.28.3.1 10BASE-T Squelch Circuit

This circuit determines when valid data is present on the RXD+/RXD- pair. Incoming signals passing through the receive filter are tested by the squelch circuit. Any signal with amplitude less than the squelch threshold (either positive or negative, depending on polarity) is rejected.

2.28.3.2 Extended Range

The CS89712 supports an Extended Range feature that reduces the 10BASE-T receive squelch threshold by approximately 6 dB. This allows the CS89712 to operate with 10BASE-T cables that are longer than 100 meters (100 meters is the maximum length specified by the Ethernet standard). The exact additional distance depends on the quality of the cable and the amount of electromagnetic noise in the surrounding environment. To activate this feature, the software must set the LoRxSquelch bit (LineCTL register bit E).

2.28.4 Link Pulse Detection

To prevent disruption of network operation due to a faulty link segment, the Ethernet port continually monitors the 10BASE-T receive pair (RXD+/RXD-) for packets and link pulses. After each packet or link pulse is received, an internal Link-Loss timer is started. As long as a packet or link pulse is received before the Link-Loss timer finishes (between 25 and 150 ms), the Ethernet port maintains normal operation. If no receive activity is detected, the Ethernet port disables packet transmission to prevent "blind" transmissions onto the network (link pulses are still sent while packet transmission is disabled). To reactivate transmission, the receiver must detect a single packet (the packet itself is ignored), or two link pulses separated by more than 2 to 7 ms and no more than 25 to 150 ms (see the Characteristics / Specifications section for 10BASE-T timing).

The state of the link segment is reported in the LinkOK bit (LineST register bit 7). If the HC0E bit (SelfCTL register bit D) is clear, it is also indicated by the $\overline{\text{LINKLED}}$ output pin. If the link is "good", the LinkOK bit is set and the $\overline{\text{LINKLED}}$ pin is driven low. If the link is "bad" the LinkOK bit is clear and the $\overline{\text{LINKLED}}$ pin is high. To disable this feature, the DisableLT bit (TestCTL register bit 7) must be set. If DisableLT is set, the Ethernet port will transmit and receive packets independent of the link segment.

2.28.5 Receive Polarity Detection/Correction

The Ethernet port checks the polarity of the receive half of the twisted pair cable. If the polarity is correct, the PolarityOK bit (LineST register bit C) is set. If the polarity is reversed, the PolarityOK bit is clear. If the PolarityDis bit (LineCTL register bit C) is clear, the Ethernet port automatically corrects a reversal. If the PolarityDis bit is set, the port does not correct a reversal. The PolarityOK bit and the PolarityDis bit are independent.

To detect a reversed pair, the receiver examines received link pulses and the End-of-Frame (EOF) sequence of incoming packets. If it detects at least one reversed link pulse and at least four frames in a row with negative polarity after the EOF, the receive pair is considered reversed. Data received before the correction of the reversal is ignored.

2.28.6 Collision Detection

If half-duplex operation is selected (FDX bit E is clear), the Ethernet port detects a 10BASE-T collision whenever the receiver and transmitter are active simultaneously. When a collision is present,

the Collision Detection circuit informs the MAC by asserting the internal Collision signal (see Section 2.26, *"Media Access Control Engine"* for collision handling).

2.29 Basic Transmit Operation

Transmit operations occur in the following order (using interrupts):

1) Software bids for storage of the frame by writing the Transmit Command to the TxCMD Port and the transmit frame length to the TxLength Port.
2) Software reads the BusST register to see if the Rdy4TxNOW bit (Bit 8) is set. To read the BusST register, the software must first set the Ethernet Port Pointer at the correct location by writing 0138h to the Ethernet Port Pointer Port (offset address 000Ah). It can then read the BusST register from the Ethernet Port Data Port (offset address 000Ch). If Rdy4TxNOW is set, the frame can be written. If clear, the software must wait for Ethernet port buffer memory to become available. If Rdy4TxiE (bit 8 of BufCFG) is set, an interrupt is generated when Rdy4Tx (bit 8 of BufEvent) becomes set. If the TxBidErr bit (BusST register bit 7) is set, the transmit length is not valid.
3) Once the Ethernet port is ready to accept the frame, software executes repetitive write instructions (REP OUT) to the Receive/Transmit Data Port to transfer the entire frame from host RAM to the Ethernet port's memory.

For a more detailed description of transmit, see Section 2.34, *"Transmit Operation"*.

2.30 Basic Receive Operation

Receive operations occur in the following order (in this example, interrupts are enabled to signal the presence of a valid receive frame):

1) A frame is received by the CS89712, triggering an enabled interrupt.
2) The software reads the Interrupt Status Queue Port and is informed of the receive frame.
3) The software reads the frame data by executing repetitive read instructions from the Receive/Transmit Data Port to transfer the frame from Ethernet port memory to host RAM. Preceding the frame data are the contents of the RxStatus register and the RxLength register.

For a more detailed description of receive, see Section 2.32, *"Basic Receive Operation"*.

2.31 Managing Interrupts & Status Queue

The Interrupt Status Queue (ISQ) is used by the Ethernet port to communicate Event reports. Whenever an event occurs that triggers an enabled interrupt, the Ethernet port sets the appropriate bit(s) in one of five registers, maps the contents of that register to the ISQ, and drives the selected interrupt request pin high (if an earlier interrupt is waiting in the queue, the interrupt request pin will already be high). When the software services the interrupt, it must first read the ISQ to learn the nature of the interrupt. It can then process the interrupt (the first read to the ISQ causes the interrupt request pin to go low.)

Three of the registers mapped to the ISQ are event registers: RxEvent, TxEvent, and BufEvent. The other two registers are counter-overflow reports: RxMISS and TxCOL. There may be more than one RxEvent report and/or more than one TxEvent report in the ISQ at a time. However, there may be only one BufEvent report, one RxMISS report and one TxCOL report in the ISQ at a time.

Event reports stored in the ISQ are read out in the order of priority, with RxEvent first, followed by TxEvent, BufEvent, RxMiss, and then TxCOL. The software only needs to read from one location to get the interrupt currently at the front of the queue. It is located at offset address 0008h. Each time the software reads the ISQ, the bits in the cor-

responding register are cleared and the next report in the queue moves to the front.

When the software starts reading the ISQ, it must read and process all Event reports in the queue. A read-out of a null word (0000h) indicates that all interrupts have been read.

The ISQ is read as a 16-bit word. The lower six bits (0 through 5) contain the register number (4, 8, C, 10, or 12). The upper ten bits (6 through F) contain the register contents. The software must always read the entire 16-bit word.

The active interrupt pin (INTRQx) is selected via the Interrupt Number register (Ethernet Port offset address 22h). As an additional option, all of the interrupt pins can be 3-Stated using the same registers; see Section 3.17 for more details.

An event triggers an interrupt only when the EnableIRQ bit (17) of the Bus Control register is set. After the CS89712 has generated an interrupt, the first read of the ISQ makes the INTRQ output pin go low (inactive). INTRQ remains low until the null word (0000h) is read from the ISQ, or for 1.6µs, whichever is longer.

2.32 Basic Receive Operation

2.32.1 Overview

Once an incoming packet has passed through the analog front end and Manchester decoder, it goes through the following three-step receive process:

1) Pre-Processing
2) Temporary Buffering
3) Transfer to system RAM

As shown in the figure, all receive frames go through the same pre-processing and temporary buffering phases, regardless of transfer method

Once a frame has been pre-processed and buffered, it can be accessed by the software.

2.32.2 Receive Configuration

After each reset, the CS89712 Ethernet port must be configured for receive operation. This can be done automatically using an attached EEPROM or by writing configuration commands to the CS89712's internal registers (see Section 2.6, *"Ethernet EEPROM Configurations"*). The items that must be configured include:

- which types of frames to accept;
- which receive events cause interrupts; and,

2.32.2.1 Configuring the Physical Interface

Configuring the physical interface consists of enabling the receive logic for serial reception. This is done via the LineCTL register. Bit 6 enables reception and bit E is used to reduce squelch.

2.32.2.2 Choosing Acceptable Frame Types

The RxCTL register selects which frame types will be accepted by the Ethernet port. Bits 6 through E of RxCTL are used for this. Refer to Section 2.32.7, *"Receive Ethernet Port Locations"* for a detailed description of Destination Address filtering, and Section 3.18.4 on page 119 for RxCTL details.

2.32.2.3 Selecting Interrupt Events

The RxCFG and BufCFG registers are used to determine which receive events will cause interrupts to the processor. Bits 8, C, D and E of BufCFG are used for this, and A, B, D and F of BufCFG. See Section 3.18.2 and Section 3.18.8 for details. Note that the DA filter must be passsed before there is an interrupt.

2.32.3 Receive Frame Pre-Processing

The CS89712 pre-processes all receive frames using a four step process:

1) Destination Address filtering;
2) Early Interrupt Generation;
3) Acceptance filtering; and,

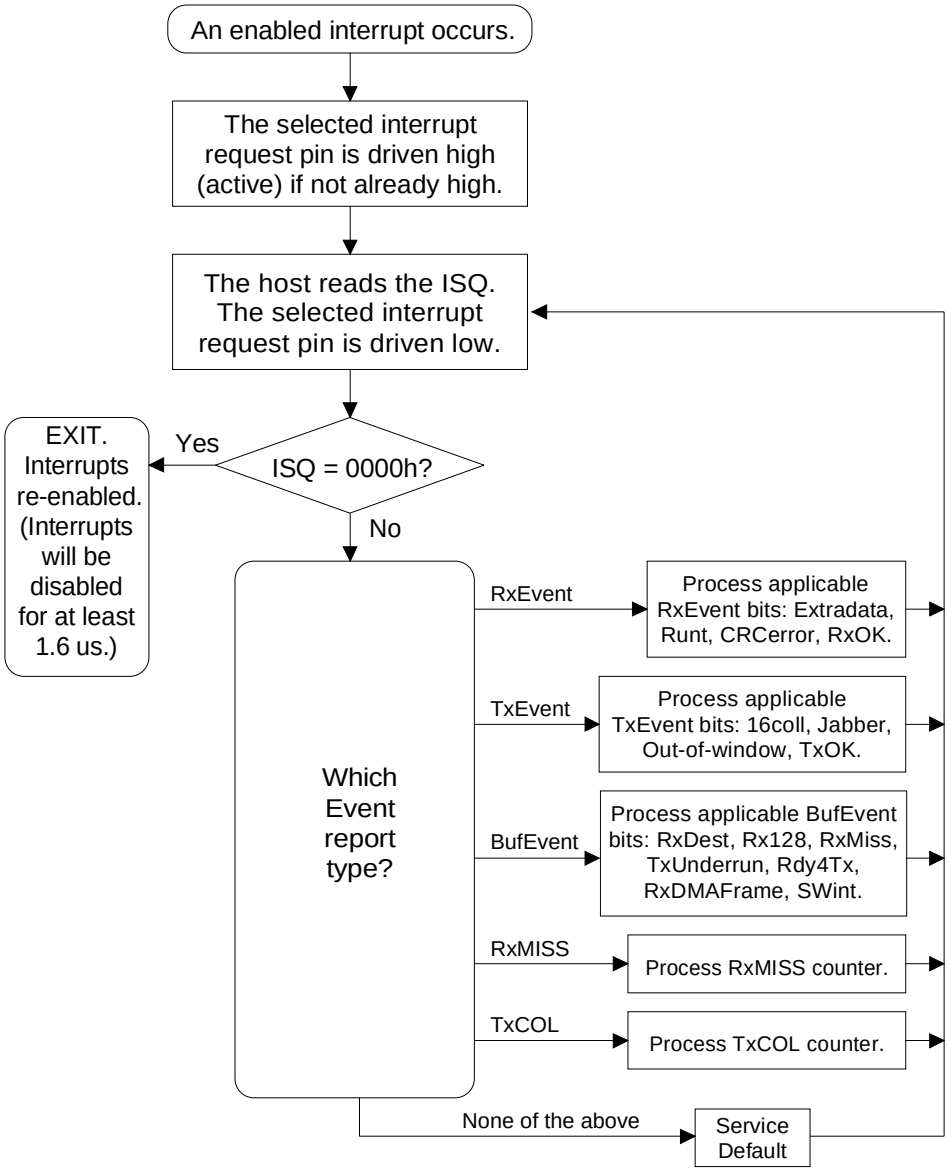


Figure 19. Interrupt Status Queue

4) Normal Interrupt Generation.

Figure 20 diagrams frame pre-processing.

2.32.3.1 Destination Address Filtering

All incoming frames are passed through the Destination Address filter (DA filter). If the frame's DA passes the DA filter, the frame is passed on for further pre-processing. If it fails the DA filter, the frame is discarded. See Section 2.32.7, *"Receive Ethernet Port Locations"* for a more detailed description of DA filtering.

2.32.3.2 Early Interrupt Generation

The Ethernet port supports the following two early interrupts for frame reception.

- **RxDest**

  The RxDest bit (bit F of BufEvent register) is set as soon as the Destination Address (DA) of the incoming frame passes the DA filter. If the RxDestiE bit (BufCFG register bit F) is set, the CS89712 generates a corresponding interrupt. Once RxDest is set, the software is allowed to read the incoming frame's DA (the first 6 bytes of the frame).

- **Rx128**

  The Rx128 bit (BufEvent register bit B) is set as soon as the first 128 bytes of the incoming frame have been received. If the Rx128iE bit (BufCFG register bit B) is set, the CS89712 generates a corresponding interrupt. Once the Rx128 bit is set, the RxDest bit is cleared and the software is allowed to read the first 128 bytes of the incoming frame. The Rx128 bit is cleared by the software reading the BufEvent register (either directly or through the Interrupt Status Queue) or by the CS89712 detecting the incoming frame's End-of-Frame (EOF) sequence.

Like all Event bits, RxDest and Rx128 are set by the whenever the appropriate event occurs. Unlike other Event bits, RxDest and Rx128 may be cleared

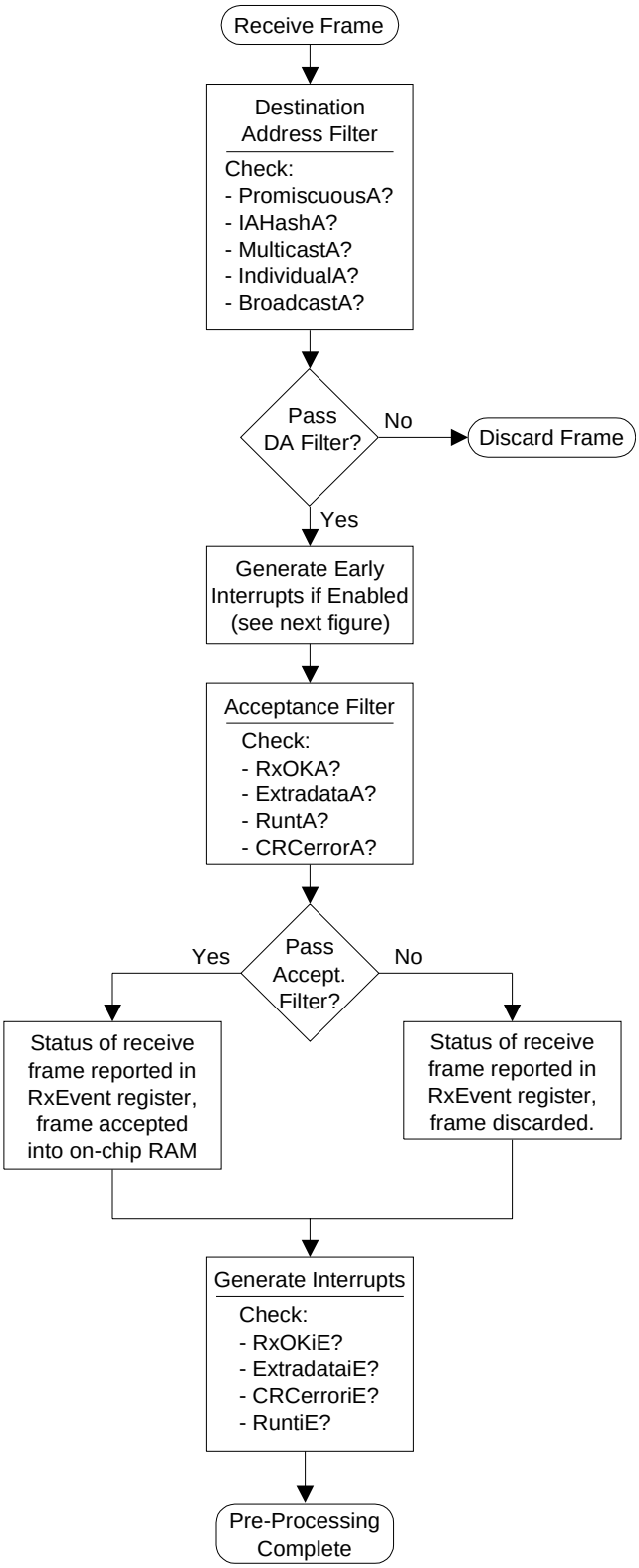


Figure 20. Receive Frame Pre-Processing

by the Ethernet port without software intervention. All other event bits are cleared only by the software reading the appropriate event register, either directly or through the Interrupt Status Queue (ISQ). (RxDest and Rx128 can also be cleared by the software reading the BufEvent register, either directly or through the Interrupt Status Queue).

2.32.3.3 Acceptance Filtering

The third step of pre-processing is to determine whether or not to accept the frame by comparing the frame with the criteria programmed into the RxCTL register. If the receive frame passes the Acceptance filter, the frame is buffered on chip. If the frame fails the Acceptance filter, it is discarded. The results of the Acceptance filter are reported in the RxEvent register.

2.32.3.4 Normal Interrupt Generation

The final step of pre-processing is to generate any enabled interrupts that are triggered by the incoming frame. Interrupt generation occurs when the entire frame has been buffered (up to the first 1518 bytes). For more information about interrupt generation, see Section 2.31, *"Managing Interrupts & Status Queue"*.

2.32.4 Buffering Held Receive Frames

If space is available, an incoming frame will be temporarily stored in on-chip RAM, where it awaits processing by the software. Although this receive frame now occupies on-chip memory, the Ethernet port does not commit the memory space to it until one of the following two conditions is true:

1) The entire frame has been received and the software has learned about the frame by reading the RxEvent register, either directly or through the ISQ.

or:

2) The frame has been partially received, causing either the RxDest bit (BufEvent register bit F) or the Rx128 bit (BufEvent register bit B) to become set, and the software has learned about the receive frame by reading the BufEvent register, either directly or through the ISQ.

When the CS89712 commits buffer space to a particular held receive frame (termed a committed received frame), no data from subsequent frames can be written to that buffer space until the frame is freed from commitment. (The committed received frame may or may not have been received error free.)

A received frame is freed from commitment by any one of the following conditions:

1) The software reads the entire frame sequentially in the order that it was received (first byte in, first byte out).

or:

2) The software reads part or none of the frame, and then issues a Skip command by setting the Skip_1 bit (RxCFG register bit 6).

or:

3) The software reads part of the frame and then reads the RxEvent register, either directly or through the ISQ, and learns of another receive frame. This condition is called an "implied Skip". Ensure that the software does not do "implied skips."

Both early interrupts are disabled whenever there is a committed receive frame waiting to be processed by the software.

There are three possible ways that the software can learn the status of a particular frame. It can:

1) Read the Interrupt Status Queue;
2) Read the RxEvent register directly ; or
3) Read the RxStatus register.

2.32.5 Receive Frame Visibility

Only one receive frame is visible to the software at a time. The receive frame's status can be read from

the RxStatus register, and its length can be read from the RxLength register.

2.32.6 Receive Frame Byte Counter

The receive frame byte counter describes the number of bytes received for the current frame. The counter is incremented in real time as bytes are received from the Ethernet. The byte counter can be used by the driver to determine how many bytes are available for reading out of the Ethernet port. Maximum Ethernet throughput can be achieved by dedicating the CPU to reading this counter, and using the count to read the frame out of the Ethernet port at the same time it is being received by the CS89712 from the Ethernet (parallel frame-reception and frame-read-out tasks).

Following an RxDest or Rx128 interrupt the register contains the number of bytes which are available to be read by the CPU. When the end of frame is reached, the count contains the final count value for the frame, including the allowance for the BufferCRC option. When this final count is read by the CPU the count register is set to zero. Therefore to read a complete frame using the byte count register, the register can be read and the data moved until a count of zero is detected. Then the RxEvent register can be read to determine the final frame status.

The sequence is as follows:

1) At the start of a frame, the byte counter matches the incoming character counter. The byte counter will have an even value prior to the end of the frame.
2) At the end of the frame, the final count, including the allowance for the CRC (if the BufferCRC option is enabled), is held until the byte counter is read.
3) When a read of the byte counter returns a count of zero, the previous count was the final count. The count may now have an odd value.
4) RxEvent should be read to obtain a final status of the frame, followed by a Skip command to complete the operation.

Note that all RxEvents should be processed before using the byte counter. The byte counter should be used following a BufEvent when RxDest or Rx128 interrupts are enabled.

2.32.7 Receive Ethernet Port Locations

The receive status/length/frame locations are read through repetitive reads from one Ethernet port at the I/O base address.

Random access is not needed. However, the first 118 bytes of the receive frame can be accessed randomly if word reads, on even word boundaries, are used. Beyond 118 bytes, the memory reads must be sequential. Byte reads, or reads on odd-word boundaries, can be performed only in sequential read mode.

The RxStatus word reports the status of the current received frame. RxEvent has the same contents as the RxStatus register, except RxEvent is cleared when read.

The RxLength (receive length) word is the length, in bytes, of the data to be transferred to the host RAM. The register describes the length from the start of Destination Address to the end of CRC, assuming that CRC has been selected (via RxCFG register bit BufferCRC). If CRC has not been selected, then the length does not include the CRC, and the CRC is not present in the receive buffer.

After the RxLength has been read, the receive frame can be read. When some portion of the frame is read, the entire frame should be read before reading the RxEvent register either directly or through the ISQ register. Reading the RxEvent register signals to the Ethernet port that the software is finished with the current frame, and wants to start processing the next frame. In this case, the current frame will no longer be accessible. The current frame will also become inaccessible if a Skip com-

mand is issued, or if the entire frame has been read. See Section 2.32, *"Basic Receive Operation"*.

2.33 Receive Frame Address Filtering

The Ethernet port is equipped with a Destination Address (DA) filter used to determine which receive frames will be accepted. The DA filter can be configured to accept the following frame types:

2.33.1 Individual Address Frames

For all Individual Address frames, the first bit of the DA is a "0" (DA[0] = 0), indicating that the address is a Physical Address. The address filter accepts Individual Address frames whose DA matches the Individual Address or whose hash-filtered DA matches one of the bits programmed into the Logical Address Filter (the hash filter is described later in this section).

2.33.2 Multicast Frames

For Multicast Frames, the first bit of the DA is a "1" (DA[0] = 1), indicating that the frame is a Logical Address. The address filter accepts Multicast frames whose hash-filtered DA matches one of the bits programmed into the Logical Address Filter (the hash filter is described later is this section). As shown in Table 28, Broadcast Frames can be accepted as Multicast frames under a very specific set of conditions.

2.33.3 Broadcast Frames

Frames with DA equal to FFFF.FFFF.FFFFh are broadcast frames. In addition, the CS89712 can be configured for Promiscuous Mode, in which case it will accept all receive frames, irrespective of DA.

2.33.4 Destination Address Filter

The DA filter is configured by five DA filter bits in the RxCTL register: IAHashA, PromiscuousA, MulticastA, IndividualA, and BroadcastA. Four of these bits are associated with four status bits in the RxEvent register: IAHash, Hashed, IndividualAdr, and Broadcast. The RxEvent register reports the results of the DA filter for a given receive frame. See Section 3.18.4 on page 119 for RxCTL details.

The IAHashA, MulticastA, IndividualA, and BroadcastA bits are used independently. As a result, many DA filter combinations are possible. For example, if MulticastA and IndividualA are set, then all frames that are either Multicast or Individual Address frames are accepted. The PromiscuousA bit, when set, overrides the other four DA bits, and allows all valid frames to be accepted. Table 29 summarizes the configuration options available for DA filtering.

It may become necessary for the software to change the Destination Address (DA) filter criteria without resetting the Ethernet port. This can be done as follows:

1) Clear SerRxON (LineCTL register bit 6) to prevent any additional receive frames while the filter is being changed.
2) Modify the DA filter bits (B, A, 9, 7, and 6) in the RxCTL register. Modify the Logical Address Filter, if necessary.
3) Set SerRxON to re-enable the receiver.

Because the receiver has been disabled, the CS89712 will ignore frames while the software is changing the DA filter.

2.33.5 Hash Filter

The hash filter is used to help determine which Multicast frames and which Individual Address frames should be accepted by the CS89712.

2.33.5.1 Hash Filter Operation

See Figure 21. The DA of the incoming frame is passed through the CRC logic, generating a 32-bit CRC value. The six most-significant bits of the CRC are latched into the 6-bit hash register (HR). The contents of the HR are passed through a 6-to-64-bit decoder, asserting one of the decoder's outputs. The asserted output is compared with a corresponding bit in the 64-bit Logical Address Filter,

Address Type of Received Frame	Erred Frame?	Passes Hash Filter?	Contents of RxEvent							
			Bits F-A					Bit 9 Hashed	Bit 8 RxOK	Bit 6 IAHash
Individual Address	no	yes	Hash Table Index					1	1	1
	no	no	ExtraData	Runt	CRC Error	Broadcast	Individual Adr	0	1	0
	yes	don't care	ExtraData	Runt	CRC Error	Broadcast	Individual Adr	0	0	0
Multicast Address	no	yes	Hash table index					1	1	0
	no	no	ExtraData	Runt	CRC Error	Broadcast	Individual Adr	0	1	0
	yes	don't care	ExtraData	Runt	CRC Error	Broadcast	Individual Adr	0	0	0
Broad-cast Address	no	yes (Note 1)	ExtraData	Runt	CRC Error (actual value X00010)	Broadcast	Individual Adr	1	1	0
	no	yes (Note 2)	ExtraData	Runt	CRC Error	Broadcast	Individual Adr	0	1	0
	no	no	ExtraData	Runt	CRC Error	Broadcast	Individual Adr	0	1	0
	yes	don't care	ExtraData	Runt	CRC Error	Broadcast	Individual Adr	0	0	0

Notes: 1. Broadcast frames are accepted as Multicast frames if and only if all the following conditions are met simultaneously:
a) the Logical Address Filter is programmed as: (MSB) 0000 8000 0000 0000h (LSB). Note that this LAF value corresponds to a Multicast Addresses of both all 1s and 03-00-00-00-00-01.
b) the Rx Control Register (register 5) is programmed to accept IndividualA, MulticastA, RxOK-only, and the following address filters were enabled: IAHashA and BroadcastA.

2. NOT (Note 1).

Table 28. Contents of RxEvent Upon Various Conditions

IAHashA	Promiscuous A	MulticastA	IndividualA	BroadcastA	Frames Accepted
0	0	0	1	0	Individual Address frames with DA matching the IA at Ethernet Port offset address 0158h
1	0	0	0	0	Individual Address frames with DA that pass the hash filter (DA[0] must be "0")
0	0	1	0	0	Multicast frames with DA that pass the hash filter (DA[0] must be "1")
0	0	0	0	1	Broadcast frames
X	1	X	X	X	All frames

Table 29. Destination Address Filtering Options

located at Ethernet Port offset address 0150h. If the decoder output and the Logical Address Filter bit match, the frame passes the hash filter and the Hashed bit (RxEvent register bit 9) is set. If the two do not match, the frame fails the filter and the Hashed bit is clear.

Whenever the hash filter is passed by a "good" frame, the RxOK bit (RxEvent register bit 8) is set and the bits in the HR are mapped to the Hash Table Index bits (RxEvent register, bits A through F).

2.33.6 Broadcast Frame Hashing Exception

Table 28 describes in detail the content of the RxEvent register for each output of the hash and address filters, and describes an exception to normal processing. That exception can occur when the hash-filter Broadcast address matches a bit in the Logical Address Filter. To properly account for this exception, the software driver should use the following test to determine if the RxEvent register contains a normal RxEvent (meaning bits E-A are used for Extra data, Runt, CRC Error, Broadcast and IndividualAdr) or a hash-table RxEvent (meaning bits F-A contain the Hash Table Index).

If bit Hashed =0, or bit RxOK=0, or (bits F-A = 02h and the destination address is all ones) then RxEvent contains a normal RxEvent, else RxEvent contained a hash RxEvent.

2.34 Transmit Operation

2.34.1 Overview

Packet transmission occurs in two phases. In the first phase, the Ethernet frame is moved into the Ethernet port's buffer memory. This phase begins with the software issuing a Transmit Command.

This informs the CS89712 that a frame is to be transmitted and tells the chip when (i.e. after 5, 381, or 1021 bytes have been transferred or after the full frame has been transferred to the CS89712) and how the frame should be sent (i.e. with or without CRC, with or without pad bits, etc.). The software follows the Transmit Command with the Transmit Length, indicating how much buffer space is required. When buffer space is available, the software writes the Ethernet frame into the Ethernet port's internal memory.

In the second phase of transmission, the Ethernet port converts the frame into an Ethernet packet then transmits it onto the network. The second phase begins with the Ethernet port transmitting the preamble and Start-of-Frame delimiter as soon as the proper number of bytes has been transferred into its transmit buffer (5, 381, 1021 bytes or full frame, depending on configuration). The preamble and Start-of-Frame delimiter are followed by the data transferred into the on-chip buffer by the software

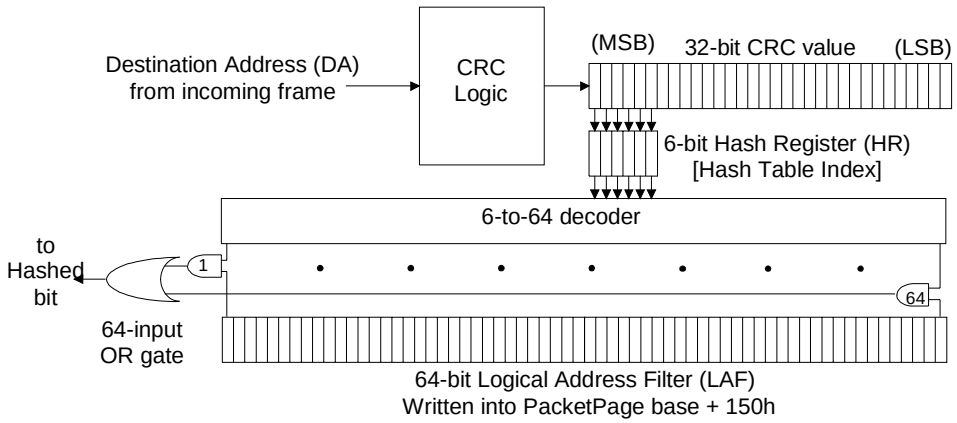


Figure 21. Hash Filter Operation

(Destination Address, Source Address, Length field and LLC data). If the frame is less than 64 bytes, including CRC, the CS89712 adds pad bits if configured to do so. Finally, the CS89712 appends the proper 32-bit CRC value.

2.34.2 Transmit Configuration

After each reset, the Ethernet port must be configured for transmit operation. This can be done automatically using an attached EEPROM, or by writing configuration commands to the Ethernet port's internal registers (see Section 2.6, *"Ethernet EEPROM Configurations"*). The items that must be configured include which physical interface to use and which transmit events cause interrupts.

2.34.2.1 Configuring the Physical Interface

Configuring the Physical Interface is accomplished via the LineCTL register. Bit 6 enables reception and bit 9 enables transmission, while bits B and D control backoff and deferral. See Section 3.18.12 on page 127 for LineCTL details.

Note that the CS89712 transmits in 10BASE-T mode when no link pulses are being received only if bit DisableLT is set in the Test Control register.

2.34.2.2 Selecting Interrupt Events

The TxCFG and BufCFG registers are used to determine which transmit events will cause interrupts. TxCFG [B:8] and F, and BufCFG [9:8] and C are used. Refer to Section 3.18.5 and Section 3.18.8 for details.

2.34.3 Changing the Configuration

When software configures these registers it does not need to change them for subsequent packet transmissions. The TxCFG or BufCFG register bits may be changed at any time. The effects of the change are noticed immediately. Any changes in the Interrupt Enable (iE) bits may affect the packet currently being transmitted.

If the LineCTL register bits are changed after initialization, the ModBackoffE bit and any receive related bit (LoRxSquelch, SerRxON) may be changed at any time.

2.34.4 Enabling CRC Generation/Padding

Whenever the software issues a Transmit Request command, it must indicate whether or not the Cyclic Redundancy Check (CRC) value should be appended to the transmit frame, and whether or not pad bits should be added (if needed). Bits C and D of TxCMD are used, refer to Table 3.18.7 on page 122.

2.34.5 Individual Packet Transmission

Whenever the software has a packet to transmit, it must issue a Transmit Request to the Ethernet port consisting of the following three operations in the exact order shown:

1) The software must write a Transmit Command to the TxCMD register. The contents of the TxCMD register may be read back from the TxCMD register.
2) The software must write the frame's length to the TxLength register.
3) Software must read the BusST register.

The information written to the TxCMD register tells the Ethernet port how to transmit the next frame. Appropriate bits in TxCMD are 6:9, C and D. Refer to Table 3.19 on page 133.

For each individual packet transmission, software must issue a *complete* Transmit Request. Furthermore, the software must write to the TxCMD register before each packet transmission, even if the contents of the TxCMD register do not change.

2.34.6 Transmit in Poll Mode

In poll mode, Rdy4TxiE bit (BufCFG register bit 8) must be clear (Interrupt Disabled). The transmit operation occurs in the following order:

1) Software bids for frame storage by writing the Transmit Command to the TxCMD register.
2) Software writes the transmit frame length to the TxLength register. If the transmit length is erroneous, the command is discarded and the TxBidErr bit (BusST register bit 7) is set.
3) The BusST register is read. The software must first set the Ethernet Port Pointer at the correct location by writing 0138h to the Ethernet Port Pointer Port. Software can then read the BusST register from the Ethernet Port Data Port.
4) After the read, Rdy4TxNOW (bit 8) is checked. If set, the frame can be written. If clear, the software must continue reading the BusST register and checking Rdy4TxNOW until set.

When the CS89712 is ready to accept the frame, software transfers the frame from host RAM to Ethernet port memory Receive/Transmit Data Port.

2.34.7 Transmit in Interrupt Mode

In interrupt mode, Rdy4TxiE bit (BufCFG register bit 8) must be set for transmit operation. Transmit operation occurs in the following order:

1) The software bids for frame storage by writing the Transmit Command to the TxCMD register.
2) The software writes the transmit frame length to the TxLength register. If the transmit length is erroneous, the command is discarded and the TxBidErr, bit 7, in BusST register is set.
3) The BusST register is read. If the Rdy4TxNOW bit is set, the frame can be written to Ethernet port memory. If Rdy4TxNOW is clear, software will have to wait for the Ethernet port buffer memory to become available at which time an interrupt is generated. On interrupt, the interrupt service routine reads ISQ register and checks the Rdy4Tx bit (bit 8). If Rdy4Tx is clear then the software waits for the next interrupt. If Rdy4Tx is set, then the Ethernet port is ready to accept the frame.
4) When the Ethernet port is ready to accept the frame, the software transfers the entire frame from host memory to Ethernet port memory to Receive/Transmit Data Port.

2.34.8 Completing Transmission

When the CS89712 successfully completes transmitting a frame, it sets the TxOK bit (TxEvent register bit 8). If the TxOKiE bit (TxCFG register bit 8) is set, an interrupt is generated.

2.34.9 Rdy4TxNOW vs. Rdy4Tx

The Rdy4TxNOW bit (BusST register bit 8) is used to indicate the Ethernet port is ready to accept a frame for transmission. This bit is used during the Transmit Request process or after the Transmit Request process to signal the software that space has become available when interrupts are not being used (i.e. Rdy4TxiE, bit 8 of Register B, is not set). Also, the Rdy4Tx bit is used with interrupts and requires the Rdy4TxiE bit be set.

2.34.10 Committing Buffer Space to a Frame

When the software issues a transmit request, the Ethernet port checks the length of the transmit frame to see if there is sufficient on-chip buffer space. If there is, the Ethernet port sets the Rdy4TxNOW bit. If not, and the Rdy4TxiE bit is set, the Ethernet port waits for buffer space to free up and then sets the Rdy4Tx bit.

Even though transmit buffer space may be available, the Ethernet port does not commit buffer space to a transmit frame until all of the following are true:

1) The software must issues a Transmit Request;
2) The Transmit Request must be successful; and,
3) Either the software reads that the Rdy4TxNOW bit (BusST register bit 8) is set, or the software reads that the Rdy4Tx bit (BufEvent register bit 8) is set.

If the CS89712 commits buffer space to a particular transmit frame, it will not allow subsequent frames to be written to that buffer space as long as the transmit frame is committed.

After buffer space is committed, the frame is subsequently transmitted unless any of the following occur:

1) The software completely writes the frame data, but transmission failed on the Ethernet line. There are three such failures, and these are indicated by three bits in the TxEvent register: 16coll, Jabber, or Out-of-Window.

or:

2) The software aborts the transmission by setting the Force (TxCMD register bit 8) bit. In this case, the committed transmit frame, as well as any yet-to-be-transmitted frames queued in the on-chip memory, are cleared and not transmitted. The software should make TxLength = 0 when using the Force bit.

or:

3) There is a transmit under-run while the Tx-Underrun bit (BufEvent register bit 9) is set.

Successful transmission is indicated when the TxOK bit (TxEvent register bit 8) is set.

2.34.11 Transmit Frame Length

The length of the frame transmitted is determined by the value written into the TxLength register during the Transmit Request. The length of the transmit frame may be modified by the configuration of the TxPadDis bit (TxCMD register bit D) and the InhibitCRC bit (TxCMD register bit C). Table 30 defines how these bits affect the length of the transmit frame, and details which frames will be sent.

2.35 Full Duplex Considerations

The driver should not bid to transmit a long frame (i.e., a frame greater than 118 bytes) if the prior transmit frame is still being transmitted. The end of the transmission of this prior frame is indicated by a TxOK bit being set in the TxEvent register.

TxCMD register		Software specified transmit length at 0146h (in bytes)			
TxPadDis (Bit D)	InhibitCRC (Bit C)	3 < TxLength < 60	60 < TxLength < 1514	1514 < TxLength < 1518	TxLength > 1518
0	0	Pad to 60 and add CRC	Send frame and add CRC [Normal Mode]	Will not send	Will not send
0	1	Pad to 60 and send without CRC	Send frame without CRC	Send frame without CRC	Will not send
1	0	Send without pads, and add CRC	Send frame and add CRC	Will not send	Will not send
1	1	Send without pads and without CRC	Send frame without CRC	Send frame without CRC	Will not send

Notes: 1. If the TxPadDis bit is clear and InhibitCRC is set and the CS89712 is commanded to send a frame of length less than 60 bytes, the CS89712 pads.

2. The CS89712 will not send a frame with TxLength less than 3 bytes.

Table 30. Transmit Frame Length

2.36 Auto-Negotiation Considerations

The original IEEE 802.3 specification requires the MAC to wait until 4 valid link-pulses are received before asserting Link-OK. Any time an invalid link-pulse is received, the count is restarted. When auto-negotiation occurs, a transmitter sends FLP (Fast Link Pulse) bursts instead of the original IEEE 802.3 NLP (Normal Link Pulses).

If the hub is attempting to auto-negotiate with the CS89712, the CS89712 will never get more than 1 "valid" link pulse (valid NLP). This is not a problem if the CS89712 is already sending link-pulses, because when the hub receives NLPs from the CS89712, the hub is required to stop sending FLPs and start sending NLPs. The NLP transmitted by the hub will put the CS89712 into Link-OK.

However, if the CS89712 is in Auto-Switch mode, the CS89712 will never send any link-pulses, and the hub will never change from sending FLPs to sending NLPs.

3. REGISTER SET

The 89712 contains multiple register ranges.

An 8 kbyte segment of memory in the range 0x8000.0000 to 0x8000.3FFF is for registers that control non-Ethernet functions. Section 3.1 gives an overview of these while Sections 3.3 through 3.16 provide register bit details.

Ethernet port registers are accessed through two separate ranges: a 16 byte window of eight registers and a 4 Kbyte page of registers. Section 3.2 explains Ethernet Port register access, and Sections 3.17 to 3.20 give bit details.

3.1 Non-Ethernet Registers

Table 31 shows the internal non-Ethernet registers of the CS89712 when the CPU is configured to a little endian memory system. Table 32 shows the differences that occur when the CPU is configured to a big endian memory system for byte-wide access to Ports A, B, and D. All the internal registers are inherently little endian (i.e., the least significant byte is attached to bits 7 to 0 of the data bus). Hence, the system Endianness affects the addresses required for byte accesses to the internal registers, resulting in a reversal of the byte address required to read/write a particular byte within a register.

There is no effect on the register addresses for word accesses. Bits A[1:0] of the internal address bus are only decoded for Ports A, B, and D (to allow read/write to individual ports). For all other registers, bits A[1:0] are not decoded, so that byte reads will return the whole register contents onto the CS89712's internal bus, from where the appropriate byte (according to the endianness) will be read. To avoid the additional complexity, it is preferable to perform all internal register accesses as word operations, except for ports A to D which are explicitly designed to operate with byte accesses, as well as with word accesses.

Writes to bits that are not explicitly defined in the internal area are legal and will have no effect. Reads from bits not explicitly defined in the internal area are legal but will read undefined values. All the internal addresses should only be accessed as 32-bit words and are always on a word boundary, except for the PIO port registers, which can be accessed as bytes. Address bits in the range A[0:5] are not decoded (except for Ports A–D), this means each internal register is valid for 64 bytes (i.e., the SYSFLG1 register appears at locations 0x8000.0140 to 0x8000.017C). There are some gaps in the register map but registers located next to a gap are still only decoded for 64 bytes.

The GPIO port registers are byte-wide and can be accessed as a word but not as a half-word. These registers additionally decode A[1:0].

Note: All byte-wide registers should be accessed as words (except Port A to Port D registers, which are designed to work in both word and byte modes).
All register bit alignment starts from the LSB of the register (i.e., they are all right shift justified).
The registers which interact with the 32 kHz clock or which could change during readback (i.e., RTC data registers, SYSFLG1 register (lower 6-bits only), the TC1D and TC2D data registers, port registers, and interrupt status registers), should be read twice and compared to ensure that a stable value has been read.

All internal registers are reset to zero by a system reset (i.e., nPOR, nURESET, or nPWRFL signals becoming active), except for the DRAM refresh period register (DPFPR), the Real-Time Clock data register (RTCDR), and the match register (RTCMR), which are only reset by nPOR becoming active. This ensures that the DRAM contents and system time are preserved through a user reset or power fail condition.

Address	Name	Default	RD/WR	Size	Comments
0x8000.0000	PADR	0	RW	8	Port A data register.
0x8000.0001	PBDR	0	RW	8	Port B data register.
0x8000.0002	—		—	8	Reserved.
0x8000.0003	PDDR	0	RW	8	Port D data register.
0x8000.0040	PADDR	0	RW	8	Port A data direction register.
0x8000.0041	PBDDR	0	RW	8	Port B data direction register.
0x8000.0042	—		—	8	Reserved.
0x8000.0043	PDDDR	0	RW	8	Port D data direction register.
0x8000.0080	PEDR	0	RW	3	Port E data register.
0x8000.00C0	PEDDR	0	RW	3	Port E data direction register.
0x8000.0100	SYSCON1	0	RW	32	System control register 1.
0x8000.0140	SYSFLG1	0	RD	32	System status flags register 1.
0x8000.0180	MEMCFG1	0	RW	32	Expansion memory configuration register 1.
0x8000.01C0	MEMCFG2	0	RW	32	Expansion memory configuration register 2.
0x8000.0200		0	RW	32	Reserved.
0x8000.0240	INTSR1	0	RD	32	Interrupt status register 1.
0x8000.0280	INTMR1	0	RW	32	Interrupt mask register 1.
0x8000.02C0	LCDCON	0	RW	32	LCD control register.
0x8000.0300	TC1D	0	RW	16	Read / Write register sets and reads data to TC1.
0x8000.0340	TC2D	0	RW	16	Read / Write register sets and reads data to TC2.
0x8000.0380	RTCDR	—	RW	32	Real Time Clock data register.
0x8000.03C0	RTCMR	—	RW	32	Real Time Clock match register.
0x8000.0400	PMPCON	0	RW	12	PWM pump control register.
0x8000.0440	CODR	0	RW	8	CODEC data I/O register.
0x8000.0480	UARTDR1	0	RW	16	UART1 FIFO data register.
0x8000.04C0	UBLCR1	0	RW	32	UART1 bit rate and line control register.
0x8000.0500	SYNCIO	0	RW	32	Synchronous serial I/O data register for master only SSI.
0x8000.0540	PALLSW	0	RW	32	Least significant 32-bit word of LCD palette register.
0x8000.0580	PALMSW	0	RW	32	Most significant 32-bit word of LCD palette register.
0x8000.05C0	STFCLR	—	WR	—	Write to clear all start up reason flags.
0x8000.0600	BLEOI	—	WR	—	Write to clear battery low interrupt.
0x8000.0640	MCEOI	—	WR	—	Write to clear media changed interrupt.
0x8000.0680	TEOI	—	WR	—	Write to clear tick and watchdog interrupt.
0x8000.06C0	TC1EOI	—	WR	—	Write to clear TC1 interrupt.
0x8000.0700	TC2EOI	—	WR	—	Write to clear TC2 interrupt.
0x8000.0740	RTCEOI	—	WR	—	Write to clear RTC match interrupt.

Table 31. CS89712 Internal Registers (Little Endian Mode)

Address	Name	Default	RD/WR	Size	Comments
0x8000.0780	UMSEOI	—	WR	—	Write to clear UART modem status changed interrupt.
0x8000.07C0	COEOI	—	WR	—	Write to clear CODEC sound interrupt.
0x8000.0800	HALT	—	WR	—	Write to enter the Idle State.
0x8000.0840	STDBY	—	WR	—	Write to enter the Standby State.
0x8000.0880–0x8000.0FFF	Reserved				Write will have no effect, read is undefined.
0x8000.1000	FBADDR	0xC	RW	4	LCD frame buffer start address.
0x8000.1100	SYSCON2	0	RW	16	System control register 2.
0x8000.1140	SYSFLG2	0	RD	24	System status register 2.
0x8000.1240	INTSR2	0	RD	16	Interrupt status register 2.
0x8000.1280	INTMR2	0	RW	16	Interrupt mask register 2.
0x8000.12C0–0x8000.147F	Reserved				Write will have no effect, read is undefined. .
0x8000.1480	UARTDR2	0	RW	16	UART2 Data register.
0x8000.14C0	UBLCR2	0	RW	32	UART2 bit rate and line control register.
0x8000.1500	SS2DR	0	RW	16	Master / slave SSI2 data register.
0x8000.1600	SRXEOF	—	WR	—	Write to clear RX FIFO overflow flag.
0x8000.16C0	SS2POP	—	WR	—	Write to pop SSI2 residual byte into RX FIFO.
0x8000.1700	KBDEOI	—	WR	—	Write to clear keyboard interrupt.
0x8000.1800	Reserved	—	WR	—	Do not write to this location. A write will cause the processor to go into an unsupported power state.
0x8000.1840–0x8000.1FFF	Reserved	—			Write will have no effect, read is undefined.
0x8000.2000	DAIR	0	RW	32	DAI control register.
0x8000.2040	DAIR0	0	RW	32	DAI data register 0.
0x8000.2080	DAIDR1	0	RW	32	DAI data register 1.
0x8000.20C0	DAIDR2	0	WR	21	DAI data register 2.
0x8000.2100	DAISR	0	RW	32	DAI status register.
0x8000.2200	SYSCON3	0	RW	16	System control register 3.
0x8000.2240	INTSR3	0	RD	32	Interrupt status register 3.
0x8000.2280	INTMR3	0	RW	8	Interrupt mask register 3.
0x8000.22C0	LEDFLSH	0	RW	7	LED Flash register.

Table 31. CS89712 Internal Registers (Little Endian Mode) (Continued)

Big Endian Mode	Name	Default	RD/WR	Size	Comments
0x8000.0003	PADR	0	RW	8	Port A Data register

Table 32. CS89712 Internal Registers (Big Endian Mode)

Big Endian Mode	Name	Default	RD/WR	Size	Comments
0x8000.0002	PBDR	0	RW	8	Port B Data register
0x8000.0001	—		—	8	Reserved
0x8000.0000	PDDR	0	RW	8	Port D Data register
0x8000.0043	PADDR	0	RW	8	Port A data Direction register
0x8000.0042	PBDDR	0	RW	8	Port B Data Direction register
0x8000.0041	—		—	8	Reserved
0x8000.0040	PDDDR	0	RW	8	Port D Data Direction register
0x8000.0083	PEDR	0	RW	3	Port E Data Register
0X8000.00C3	PEDDR	0	RW	3	Port E Data Direction register

Table 32. CS89712 Internal Registers (Big Endian Mode)

1. The following register descriptions refer to Little Endian Mode Only.

3.2 Accessing Ethernet Port Registers

Registers for the Ethernet port are accessed through two memory ranges; first, a 16-byte window of eight 16-bit registers (shown in Figure 33) located at address 0x2000.0300; and additional registers in a 4 Kbyte internal memory page listed in Figure 36. The registers at 0x2000.0300 are always immediately accessable, however registers mapped into the 4 Kbyte page must be accessed through an index using the Ethernet Port pointer and Ethernet Data Ports.

This is done by writing the offset of the target register to the Ethernet Port Pointer. For example, the EEPROM data register has an offset of 0042h. The contents of the target register are then mapped into the Ethernet Data Port.

If the software needs to access a sequential block of registers, the MSB of the Ethernet Port address of the first word to be accessed should be set to "1". The Ethernet Port Pointer will then move to the next word location automatically, eliminating the need to setup the Ethernet Port Pointer between successive accesses (see Figure 22).

Memory Location	Type	Description
0x2000.0300	Read/Write	Receive/Transmit Data (Port 0)
0x2000.0302	Read/Write	Receive/Transmit Data (Port 1)
0x2000.0304	Write-only	TxCMD (Transmit Command)
0x2000.0306	Write-only	TxLength (Transmit Length)
0x2000.0308	Read-only	Interrupt Status Queue
0x2000.030A	Read/Write	Ethernet Port Pointer
0x2000.030C	Read/Write	Ethernet Port Data (Port 0)
0x2000.030E	Read/Write	Ethernet Port Data (Port 1)

Table 33. Ethernet Port Register Window

3.2.1 Ethernet Port Register Window

This section refers to the eight 2-byte registers residing in the 16-byte window at 0x2000.3000. These registers are always immediately available.

3.2.1.1 Receive/Transmit Data Ports 0 & 1

These two ports are used when transferring transmitting/receving data to/from the CS89712. Port 0 is used for 16-bit operations and Ports 0 and 1 are for 32-bit operations (lower-order word in Port 0).

3.2.1.2 TxCMD Port

Software writes the Transmit Command (TxCMD) to this port at the start of each transmit operation. The Transmit Command indicates that the software has a frame to be transmitted, as well as how that frame should be transmitted. See Section 3.2.3, *"Ethernet Status/Control Registers"* for more information.

3.2.1.3 TxLength Port

The length of the frame to be transmitted is written here immediately after the Transmit Command is written.

3.2.1.4 Interrupt Status Queue Port

This port contains the current value of the Interrupt Status Queue (ISQ). For a more detailed description of the ISQ, see Section 2.31, *"Managing Interrupts & Status Queue"*.

3.2.1.5 Ethernet Port Pointer

The Ethernet Port Pointer is written in order to access any of the Ethernet port indexed registers (which reside in the 4 Kbyte memory page). The first 12 bits (bits 0 through B) of the pointer provide the offset of the target register to be accessed during the current operation. The next three bits (C, D, and E) are read-only and will always read as 011b. Any convenient value may be written to these bits when writing to the Ethernet Port Pointer Port. The last bit (Bit F) indicates whether or not the Ethernet Port Pointer should be auto-incremented to the next word location. Figure 22 shows the structure of the Ethernet Port Pointer.

3.2.1.6 Ethernet Port Data Ports 0 and 1

The Ethernet Port Data Ports are used to transfer data to and from any of the CS89712's internal registers. Port 0 is used for 16-bit operations and Port 0 and 1 are used for 32-bit operations (lower-order word in Port 0).

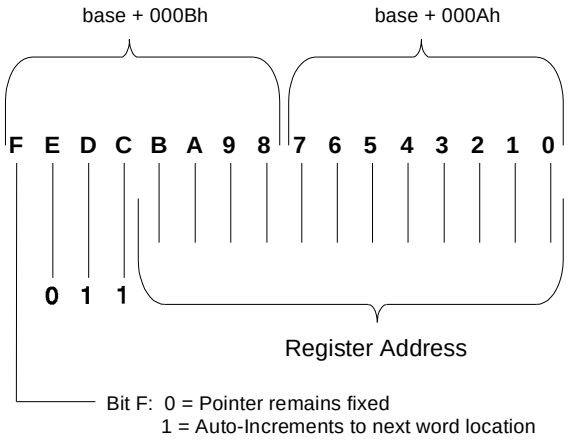


Figure 22. Ethernet Port Pointer

3.2.2 Ethernet Port Indexed Registers

Central to the Ethernet port architecture is a 4 Kbyte page of integrated RAM, which is used for temporary storage of transmit and receive frames, and for additional registers. These registers are accessed by use with the Ethernet Data Pointer and Data Ports. These registers are organized into the following sections:

3.2.2.1 Bus Interface Registers

The Bus Interface Registers contain Ethernet Port interrupt enables, EEPROM control and data, and receive frame information.

3.2.2.2 Status and Control Registers

The Status and Control registers are the primary means of controlling and reading status of the Ethernet port. They are detailed in Section 3.2.3, *"Ethernet Status/Control Registers"*.

3.2.2.3 Initiate Transmit Registers

The TxCMD/TxLength registers are used to initiate Ethernet frame transmission. These are detailed in Section 3.19, *"Initiate Transmit Registers"*. (Also see Section 2.34, *"Transmit Operation"* for a description of frame transmission.)

3.2.2.4 Address Filter Registers

The Filter registers store the Individual Address filter and Logical Address filter used by the Destination Address filter. These registers are described in more detail in Section 3.20, *"Address Filter Registers"*. For a description of the DA filter, see Section 2.32.7, *"Receive Ethernet Port Locations"*.

3.2.2.5 Receive/Transmit Frame Locations

The Receive and Transmit Frame Ethernet Port locations are used to transfer Ethernet frames to and from the host RAM. The software simply writes to and reads from these locations and internal buffer memory is dynamically allocated between transmit and receive as needed. This provides more efficient use of buffer memory and better overall network performance. As a result of this dynamic allocation, only one receive frame and one transmit frame are directly accessible.

3.2.3 Ethernet Status/Control Registers

The Status and Control registers are the primary registers used to control and check the status of the Ethernet port in the CS89712. They are organized into two groups: Configuration/Control Registers and Status/Event Registers. All Status and Control Registers are 16-bit words as shown in Figure 23. Bit 0 indicates whether it is a Configuration/Control Register (Bit 0 = 1) or a Status/Event Register (Bit 0 = 0). Bits 0 through 5 provide an internal address code that describes the exact function of the register. Bits 6 through F are the actual Configuration/Control and Status/Event bits.

3.2.4 Configuration and Control Registers

Configuration and Control registers are used to setup the following:

- how frames will be transmitted and received;
- which frames will be transmitted and received;
- which events will cause interrupts to the processor; and,
- Configuration of the Ethernet physical interface.

These registers are read/write and are designated by odd numbers (e.g. Register 1, Register 3, etc.).

The Transmit Command Register (TxCMD) is a special type of register. It appears in two separate locations in the Ethernet Port memory map. The first location, Ethernet Port offset address 0108h, is within the block of Configuration/Control Registers and is read-only. The second location, Ethernet Port offset address 0144h, is where the actual transmit commands are issued and is write-only. See Section 3.2.7, *"Status/Control Register Summary"* and Section 2.34, *"Transmit Operation"* for a more detailed description of the TxCMD register.

3.2.5 Status and Event Registers

Status and Event registers report the status of transmitted and received frames, as well as information about the configuration of the CS89712. They are read-only.

The Interrupt Status Queue (ISQ) is a special type of Status/Event register. It is located at Ethernet Port offset address 0120h and is the first register the software reads when responding to an Interrupt.

A more detailed description of the ISQ can be found in Section 2.31, *"Managing Interrupts & Status Queue"*.

Three 10-bit counters are included with the Status and Event registers. RxMISS counts missed receive frames, TxCOL counts transmit collisions.

Table 34 summarizes Ethernet Port Register types.

3.2.6 Status and Control Bit Definitions

This section provides a description of the special bit types used in the Status and Control registers.

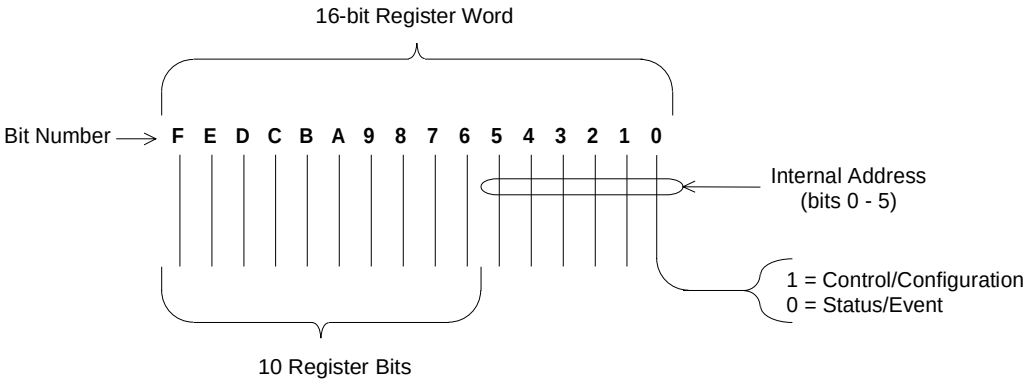


Figure 23. Status and Control Register Format

Suffix	Type	Description	Comments
CMD	Read/Write	**Command:** Written once per frame to initiate transmit.	
CFG	Read/Write	**Configuration:** Written at setup and used to determine what frames will be transmitted and received and what events will cause interrupts.	
CTL	Read/Write	**Control:** Written at setup and used to determine what frames will be transmitted and received and how the physical interface will be configured.	
Event	Read-only	**Event:** Reports the status of transmitted and received frames.	cleared when read
ST	Read-only	**Status:** Reports information about the configuration of the CS89712.	
	Read-only	**Counters:** Counts missed receive frames and collisions. Provides time domain for locating coax cable faults.	cleared when read

Table 34. Ethernet Port Register Types

Section 3.2.7, *"Status/Control Register Summary"* provides a detailed description of each register.

3.2.6.1 Act-Once Bits

There are four bits that cause a certain action only once when set. These "Act-Once" bits are: Skip_1 (RxCFG register bit 6), RESET (SelfCTL register bit 6), ResetRxDMA (BusCTL register bit 6), and SWint-X (BufCFG register bit 6). To cause the action again, the software must set the bit again. Act-Once bits are always read as clear.

3.2.6.2 Temporal Bits

Temporal bits are bits that are set and cleared by the Ethernet port automatically. This includes all status bits in the three status registers (LineST, SelfST, and BusST), the RxDest bit (BufEvent bit F), and the Rx128 bit (BufEvent bit B). Like all Event bits, RxDest and Rx128 are cleared when read by the software.

3.2.6.3 Interrupt Enable Bits and Events

Interrupt Enable bits end with the suffix iE and are located in three Configuration registers: RxCFG, TxCFG, and BufCFG. Each Interrupt Enable bit corresponds to a specific event. If an Interrupt Enable bit is set and its corresponding event occurs, the Ethernet port generates an interrupt.

The bits that report when various events occur are located in three Event registers and two counters. The Event registers are RxEvent, TxEvent, and BufEvent. The counters are RxMISS and TxCOL. Each Interrupt Enable bit and its associated Event are identified in Table 35.

An Event bit will be set whenever the specified event happens, whether or not the associated Interrupt Enable bit is set. All Event registers are cleared upon read-out by the software.

3.2.6.4 Accept Bits

There are nine Accept bits located in the RxCTL register, each of which is followed by the suffix A.

Interrupt Enable Bit (register name)	Event Bit or Counter (register name)
ExtradataiE (RxCFG)	Extradata (RxEvent)
RuntiE (RxCFG)	Runt (RxEvent)
CRCerroriE (RxCFG)	CRCerror (RxEvent)
RxOKiE (RxCFG)	RxOK (RxEvent)
16colliE (TxCFG)	16coll (TxEvent)
AnycolliE (TxCFG)	"Number-of Tx-collisions" counter is incremented (TxEvent)
JabberiE (TxCFG)	Jabber (TxEvent)
Out-of-windowiE (TxCFG)	Out-of-window (TxEvent)
TxOKiE (TxCFG)	TxOK (TXEvent)
MissOvfloiE (BufCFG)	RxMISS counter overflows past 1FFh
TxColOvfloiE (BufCFG)	TxCOL counter overflows past 1FFh
RxDestiE (BufCFG)	RxDest (BufEvent)
Rx128iE (BufCFG)	Rx128 (BufEvent)
RxMissiE (BufCFG)	RxMISS (BufEvent)
TxUnderruniE (BufCFG)	TxUnderrun (BufEvent)
Rdy4TxiE (BufCFG)	Rdy4Tx (BufEvent)

Table 35. Interrupt Enable Bits and Events

Accept bits indicate which types of frames will be accepted by the CS89712. Four of these bits have corresponding Interrupt Enable (iE) bits. An Accept bit and an Interrupt Enable bit are independent operations. It is possible to set either, neither, or both bits. The four corresponding pairs of bits are:

IE Bit in RxCFG	A Bit in RxCTL
ExtradataiE	ExtradataA
RuntiE	RuntA
CRCerroriE	CRCerrorA
RxOKiE	RxOKA

If one of the above Interrupt Enable bits is set and the corresponding Accept bit is clear, the CS89712

generates an interrupt when the associated receive event occurs, but then does not accept the receive frame (the receive frame length is set to zero).

The other five Accept bits in RxCTL are used for destination address filtering (see Section 2.32.7, *"Receive Ethernet Port Locations"*). The Accept mechanism is explained in more detail in Section 2.32, *"Basic Receive Operation"*.

3.2.7 Status/Control Register Summary

This section gives a detailed description of each Status and Control register. Bits marked "RSVD" are reserved and must be written with a zero for proper operation of the device.

3.3 Ethernet Port 4 Kbyte Memory Register Map

The following Table shows the CS89712 Ethernet Port internal register map:

Internal Offset	# of Bytes	Type	Description	Cross Reference
Bus Interface Registers				
0000h	4		Reserved	
0004h	28	-	Reserved	Note 2
0022h	2	RW	Master Interrupt Enable	Section 3., *"REGISTER SET"*, Section 3.17, *"Ethernet Bus Interface Registers"*
0038h	8	-	Reserved	Note 2
0040h	2	RW	EEPROM Command	Section 2.24, *"Programming the EEPROM"*, Section 3.17, *"Ethernet Bus Interface Registers"*
0042h	2	RW	EEPROM Data	Section 2.24, *"Programming the EEPROM"*, Section 3.17, *"Ethernet Bus Interface Registers"*
0044h	12	-	Reserved	Note 2
0050h	2	RD	Received Frame Byte Counter	Section 3.17, *"Ethernet Bus Interface Registers"*, Section 2.32, *"Basic Receive Operation"*
0052h	174	-	Reserved	Note 2
Status and Control Registers				
0102	2	RW	Receive Configuration	Section 3.2.3, *"Ethernet Status/Control Registers"*
0104	2	RW	Receive Control	see above
0106	2	RW	Transmit Configuration	see above
0108	2	RW	Transmit Command	see above
010A	2	RW	Buffer Configuration	see above
0112	2	RW	Line Control	see above
0114	2	RW	Self Control	see above
0116	2	RW	Bus Control	see above
0118	2	RW	Test Control	see above
0120	2	RD	Interrupt Status Queue	
0124	2	RD	Receive Event	
0124	2	RD	alternate Receive Event	
0128	2	RD	Transmit Event	
012C	2	RD	Buffer Event	
0130	2	RD	Receive Miss	
0132	2	RD	Transmit Collision	
0134	2	RD	Line Status	
0136	2	RD	Self Status	

Table 36. Ethernet Port 4 Kbyte Memory Register Address Map

Internal Offset	# of Bytes	Type	Description	Cross Reference
0138	2	RD	Bus Status	
0140h	4	-	Reserved	Note 2
Initiate Transmit Registers				
0144h	2	WR	TxCMD (transmit command)	Section 3.19, *"Initiate Transmit Registers"*, Section 2.34, *"Transmit Operation"*
0146h	2	WR	TxLength (transmit length)	Section 3.19, *"Initiate Transmit Registers"*, Section 2.34, *"Transmit Operation"*
0148h	8	-	Reserved	Note 2
Address Filter Registers				
0150h	8	RW	Logical Address Filter (hash table)	Section 3.20, *"Address Filter Registers"*, Section 2.32.7, *"Receive Ethernet Port Locations"*
0158h	6	RW	Individual Address	Section 3.20, *"Address Filter Registers"*, Section 2.32.7, *"Receive Ethernet Port Locations"*
015Eh	674	-	Reserved	Note 2
Frame Location				
0400h	2	RD	RXStatus (receive status)	Section 3.18, *"Ethernet Port Status/Control Registers"*, Section 2.32, *"Basic Receive Operation"*
0402h	2	Read-only	RxLength (receive length, in bytes)	Section 3.18, *"Ethernet Port Status/Control Registers"*, Section 2.32, *"Basic Receive Operation"*
0404h	-	Read-only	Receive Frame Location	Section 3.18, *"Ethernet Port Status/Control Registers"*, Section 2.32, *"Basic Receive Operation"*
0A00	-	Write-only	Transmit Frame Location	Section 3.18, *"Ethernet Port Status/Control Registers"*, Section 2.34, *"Transmit Operation"*

Table 36. Ethernet Port 4 Kbyte Memory Register Address Map (Continued)

Notes: 1. All registers are accessed as 16-bit only.
2. Read operation from the reserved location provides undefined data. Writing to a reserved location or undefined bits may result in unpredictable operation.

3.4 I/O Port Data Registers

3.4.1 PADR Port A Data Register (address 0x8000.0000)

Values written to this 8-bit read / write register will be output on Port A pins if the corresponding data direction bits are set high (port output). Values read from this register reflect the external state of Port A, not necessarily the value written to it. All bits are cleared by a system reset.

3.4.2 PBDR Port B Data Register (address 0x8000.0001)

Values written to this 8-bit read / write register will be output on Port B pins if the corresponding data direction bits are set high (port output). Values read from this register reflect the external state of Port B, not necessarily the value written to it. All bits are cleared by a system reset.

3.4.3 PDDR Port D Data Register (address 0x8000.0003)

Values written to this 8-bit read / write register will be output on Port D pins if the corresponding data direction bits are set low (port output). Values read from this register reflect the external state of Port D, not necessarily the value written to it. All bits are cleared by a system reset.

3.4.4 PADDR Port A Data Direction Register (address 0x8000.0040)

Bits set in this 8-bit read / write register will select the corresponding pin in Port A to become an output, clearing a bit sets the pin to input. All bits are cleared by a system reset.

3.4.5 PBDDR Port B Data Direction Register (address 0x8000.0041)

Bits set in this 8-bit read / write register will select the corresponding pin in Port B to become an output, clearing a bit sets the pin to input. All bits are cleared by a system reset.

3.4.6 PDDDR Port D Data Direction Register (address 0x8000.0043)

Bits cleared in this 8-bit read / write register will select the corresponding pin in Port D to become an output, setting a bit sets the pin to input. All bits are cleared by a system reset so that Port D is output by default.

3.4.7 PEDR Port E Data Register (address 0x8000.0080)

Values written to this 3-bit read / write register will be output on Port E pins if the corresponding data direction bits are set high (port output). Values read from this register reflect the external state of Port E, not necessarily the value written to it. All bits are cleared by a system reset.

3.4.8 PEDDR Port E Data Direction Register (address 0x8000.00C0)

Bits set in this 3-bit read / write register will select the corresponding pin in Port E to become an output, while the clearing bit sets the pin to input. All bits are cleared by a system reset so that Port E is input by default.

3.5 System Control Registers

3.5.1 SYSCON1 The System Control Register 1 (address 0x8000.0100)

23:21	20	19	18	17:16	15
Reserved	IRTXM	WAKEDIS	EXCKEN	ADCKSEL	SIREN
14	**13**	**12**	**11**	**10**	**9**
CDENRX	CDENTX	LCDEN	DBGEN	BZMOD	BZTOG
8	**7**	**6**	**5**	**4**	**3:0**
UART1EN	TC2S	TC2M	TC1S	TC1M	Keyboard scan

The system control register is a 21-bit read / write register which controls all the general configuration of the CS89712, as well as modes etc. for peripheral devices. All bits in this register are cleared by a system reset. The bits in the system control register SYSCON1 are defined in Table 37.

Bit	Description
0:3	**Keyboard scan**: This 4-bit field defines the state of the keyboard column drives. The following table defines these states.

Keyboard Scan	Column
0	All driven high
1	All driven low
2–7	All high impedance (tristate)
8	Column 0 only driven high all others high impedance
9	Column 1 only driven high all others high impedance
10	Column 2 only driven high all others high impedance
11	Column 3 only driven high all others high impedance
12	Column 4 only driven high all others high impedance
13	Column 5 only driven high all others high impedance
14	Column 6 only driven high all others high impedance
15	Column 7 only driven high all others high impedance

4	**TC1M**: Timer counter 1 mode. Setting this bit sets TC1 to prescale mode, clearing it sets free running mode.
5	**TC1S**: Timer counter 1 clock source. Setting this bit sets the TC1 clock source to 512 kHz, clearing it sets the clock source to 2 kHz.
6	**TC2M**: Timer counter 2 mode. Setting this bit sets TC2 to prescale mode, clearing it sets free running mode.
7	**TC2S**: Timer counter 2 clock source. Setting this bit sets the TC2 clock source to 512 kHz, clearing it sets the clock source to 2 kHz.
8	**UART1EN**: Internal UART enable bit. Setting this bit enables the internal UART.
9	**BZTOG**: Bit to drive (i.e., toggle) the buzzer output directly when software mode of operation is selected (i.e., bit BZMOD = 0). See the BZMOD and BUZFREQ (SYSCON1) bits for more details.
10	**BZMOD**: This bit selects the buzzer drive mode. When BZMOD = 0, the buzzer drive output pin is connected directly to the BZTOG bit. This is the software mode. When BZMOD = 1, the buzzer drive is in the hardware mode. Two hardware sources are available to drive the pin. They are the TC1 or a fixed internally generated clock source. The selection of which source is used to drive the pin is determined by the state of the BUZFREQ bit in the SYSCON2 register. If the TC1 is selected, then the buzzer output pin is connected to the TC1 under flow bit. The buzzer output pin changes every time the timer wraps around. The frequency depends on what was programmed into the timer. See the description of the BUZFREQ and BZTOG bits (SYSCON2) for more details.

Table 37. SYSCON1

Bit	Description
11	**DBGEN**: Setting this bit will enable the debug mode. In this mode, all internal accesses are output as if they were reads or writes to the expansion memory addressed by nCS5. nCS5 will still be active in its standard address range. In addition, the internal interrupt request and fast interrupt request signals to the ARM720T processor are output on Port E, bits 1 and 2. Note that these bits must be programmed to be outputs before this functionality can be observed. The clock to the CPU is output on Port E, Bit 0 to delineate individual accesses. For example, in debug mode: nCS5 = nCS5 or internal I/O strobe PE0 = CLK PE1 = nIRQ PE2 = nFIQ
12	**LCDEN**: LCD enable bit. Setting this bit enables the LCD controller.
13	**CDENTX**: Codec interface enable TX bit. Setting this bit enables the codec interface for data transmission to an external codec device.
14	**CDENRX**: Codec interface enable RX bit. Setting this bit enables the codec interface for data reception from an external codec device. Note: Both CDENRX and CDENTX need to be enabled / disabled in tandem, otherwise data may be lost.
15	**SIREN**: HP SIR protocol encoding enable bit. This bit will have no effect if the UART is not enabled.
16:17	**ADCKSEL**: Microwire / SPI peripheral clock speed select. This two-bit field selects the frequency of the ADC sample clock, which is twice the frequency of the synchronous serial ADC interface clock. The table below shows the available frequencies for operation when in PLL mode. These bits are also used to select the shift clock frequency for the SSI2 interface when set into master mode.

ADCKSEL	ADC Sample Frequency (kHz) — SMPCLK	ADC Clock Frequency (kHz) — ADCCLK
00	8	4
01	32	16
10	128	64
11	256	128

18	**EXCKEN**: External expansion clock enable. If this bit is set, the EXPCLK is enabled continuously as a free running clock with the same frequency and phase as the CPU clock, assuming that the main oscillator is running. This bit should not be left set all the time for power consumption reasons. If the system enters the Standby State, the EXPCLK will become undefined. If this bit is clear, EXPCLK will be active during memory cycles to expansion slots that have external wait state generation enabled only.
19	**WAKEDIS**: Setting this bit disables waking up (exiting) from the Standby State, from either the WAKEUP input pin or a keypress after one of the following signals became active: nPWRFL, BATOK, nEXTPWR. Note: Even though a keypress will not wake the device, a keypress interrupt will still be generated, if the keyboard interrupt is not masked, and this can be used to wake the device.

Table 37. SYSCON1 (Continued)

Bit	Description
20	**IRTXM**: IrDA TX mode bit. This bit controls the IrDA encoding strategy. Clearing this bit means that each zero bit transmitted is represented as a pulse of width 3/16th of the bit rate period. Setting this bit means each zero bit is represented as a pulse of width 3/16th of the period of 115,200-bit rate clock (i.e., 1.6 µsec regardless of the selected bit rate). Setting this bit will use less power, but will probably reduce transmission distances.

Table 37. SYSCON1 (Continued)

3.5.2 SYSCON2 System Control Register 2 (address 0x8000.1100)

15	14	13	12	11:10		9	8
Reserved	BUZFREQ	CLKENSL	OSTB	Reserved		SS2MAEN	UART2EN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SS2RXEN	PC CARD2	PC CARD1	SS2TXEN	KBWEN	DRAMSZ	KBD6	SERSEL

This is an extension of SYSCON1, containing additional control for the CS89712. The bits of this second system control register are defined below. The SYSCON2 register is reset to all 0s on power up.

Bit	Description
0	**SERSEL:**The only affect of this bit is to select either SSI2 or the codec to interface to the external pins. See the table below for the selection options. **NOTE:** If the DAI bit of SYSCON3 is set, then it overrides the state of the SERSEL bit, and thus the external pins are connected to the DAI interface.
1	**KBD6**: The state of this bit determines how many of the Port A inputs are OR'ed together to create the keyboard interrupt. When zero (the reset state), all eight of the Port A inputs will generate a keyboard interrupt. When set high, only Port A bits 0 to 5 will generate an interrupt from the keyboard. It is assumed that the keyboard row lines are connected into Port A.
2	**DRAMZ**: This bit determines the width of the DRAM memory interface, where: 0=32-bit DRAM and 1=16-bit DRAM.
3	**KBWEN**: When the KBWEN bit is high, the CS89712 will awaken from a power saving state into the Operating State when a high signal is on one of Port A's inputs (irrespective of the state of the interrupt mask register). This is called the Keyboard Direct Wakeup mode. In this mode, the interrupt request does not have to get serviced. If the interrupt is masked (i.e., the interrupt mask register 2 (INTMR2) bit 0 is low), the processor simply starts re-executing code from where it left off before it entered the power saving state. If the interrupt is non-masked, then the processor will service the interrupt.
4	**SS2TXEN:** Transmit enable for the synchronous serial interface 2. The transmit side of SSI2 will be disabled until this bit is set. When set low, this bit also disables the SSICLK pin (to save power) in master mode, if the receive side is low.
5	**PC CARD1**: Enable for the interface to the CL-PS6700 device for PC Card slot 1. The main effect of this bit is to reassign the functionality of Port B, bit 0 to the PRDY input from the CL-PS6700 devices, and to ensure that any access to the nCS4 address space will be according to the CL-PS6700 interface protocol.
6	**PC CARD2**: Enable for the interface to the CL-PS6700 device for PC Card slot 2. The main effect of this bit is to reassign the functionality of Port B, bit 1 to the PRDY input from the CL-PS6700 devices and to ensure that any access to the nCS5 address space will be according to the CL-PS6700 interface protocol.

SERSEL Value	Selected Serial Device to External Pins
0	Master / slave SSI2
1	Codec

Table 38. SYSCON2

Bit	Description
7	**SS2RXEN**: Receive enable for the synchronous serial interface 2. The receive side of SSI2 will be disabled until this bit is set. When both SSI2TXEN and SSI2RXEN are disabled, the SSI2 interface will be in a power saving state.
8	**UART2EN**: Internal UART2 enable bit. Setting this bit enables the internal UART2.
9	**SS2MAEN**: Master mode enable for the synchronous serial interface 2. When low, SSI2 will be configured for slave mode operation. When high, SSI2 will be configured for master mode operation. This bit also controls the directionality of the interface pins.
10	**SNZPOL**: Snooze State LCD data polarity bit. When low, the LCD controller will put out ‘0000’ on the DD[3:0] outputs during the blanked parts of the display in Snooze State. When high, ‘1111’ will be output instead. This is to allow the connection of displays with inverse polarity. During normal Operating State, an inverse polarity display is handled by appropriate programming of the palette, and this bit will have no effect.
11	**LCDSNZE**: This bit is normally set low, but will be automatically set high on entering Snooze State. While this bit is high, data will be fetched from the on-chip SRAM for the display. When Snooze State is exited, this bit will have been set high and this will have the effect of causing the LCD controller to continue to fetch data from the on-chip SRAM, in 1-bit-per-pixel mode, irrespective of the contents of the LCDCON register. This ensures that the display does not change on exit from Snooze State, so that if the exit is for a simple update-on-interrupt operation the display need not be affected. Additional arbitration is included so that the on-chip SRAM can be written to while the LCDSNZE bit is set after Snooze State. If the exit from Snooze State has occurred because the device was to be completely woken up, including switching to the main LCD frame buffer, (whose start address is defined in the FBADDR register) then this can be achieved by writing a 0 to the LCDSNZE bit, which will cause the CL-CS89712 to start fetching DMA data from the main buffer and sync up the display at the end of the following frame. The LCDSNZE bit can never be programmed to 1 by the CPU — the value ‘1’ will be ignored.
12	**Reserved:** This bit should be set low.
13	**CLKENSL**: CLKEN select. When low, the CLKEN signal will be output on the RUN/CLKEN pin. When high, the RUN signal will be output on RUN/CLKEN.
14	**BUZFREQ**: The BUZFREQ bit is used to select which hardware source will be used as the source to drive the buzzer output pin. When BUZFREQ = 0, the buzzer signal generated from the on-chip timer (TC1) is output. When BUZFREQ = 1, a 500 Hz clock is output. See the BZMOD and the BZTOG bits (SYSCON2) for more details.

Table 38. SYSCON2 (Continued)

3.5.3 SYSCON3 System Control Register 3 (address 0x8000.2200)

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	Reserved	Reserved	ENPD67	128Fs	FASTWAKE
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VERSN[2] Reserved	VERSN[1] Reserved	VERSN[0] Reserved	ADCCKNSEN	DAISEL	CLKCTL1	CLKCTL0	ADCCON

This register allows additional control for the CS89712. The bits of this register are defined in Table 38.

Bit	Description
0	**ADCCON**: Determines whether the ADC Configuration Extension field SYNCIO(31:16) is to be used for ADC configuration data. When this bit = 0 (default state) the ADC Configuration Byte SYNCIO(7:0) only is used for compatibility with the CL-PS7111. When this bit = 1, the ADC Configuration Extension field in the SYNCIO register is used for ADC Configuration data and the value in the ADC Configuration Byte (SYNCIO(6:0)) selects the length of the data (8-bit to 16-bit).
1:2	**CLKCTL(1:0)**: Determines the frequency of operation of the processor and Wait State scaling. The table below lists the available options.

CLKCTL(1:0) Value	Processor Frequency	Memory Bus Frequency	Wait State Scaling
00	18.432 MHz	18.432 MHz	1
01	36.864 MHz	36.864 MHz	2
10	49.152 MHz	36.864 MHz	2
11	73.728 MHz	36.864 MHz	2

Note: To determine the number of wait states programmed refer to Table 46 and Table 47. Under no circumstances should the CLKCTL bits be changed using a buffered write.

3	**DAISEL**: When set selects the DAI Interface. When cleared selects either the SSI or telephony codec interface (i.e., DAISEL bit is default low).
4	**ADCCKNSEN**: When set, configuration data is transmitted on ADCOUT at the rising edge of the ADCCLK, and data is read back on the falling edge on the ADCIN pin. When clear (default), the opposite edges are used.
5:7	**VERSN[0:2]**: Additional read-only version bits — will read ‘000’
8	**Reserved.** This bit must be set to zero
9	**128Fs**: When set, this selects the 128 fs mode. Cleared by default to enable 64 fs operation.
10	**ENPD67:** Pd[6-7] control the byte mask of the SDRAM interface. Setting of this bit allows their use as GPIO bits for applications not using SDRAM.

Table 39. SYSCON3

3.5.4 SYSFLG1 — The System Status Flags Register (address 0x8000.0140)

31:30	29	28	27	26
VERID	ID	BOOTBIT1	BOOTBIT0	SSIBUSY
25	**24**	**23**	**22**	**21:16**
CTXFF	CRXFE	UTXFF1	URXFE1	RTCDIV
15	**14**	**13**	**12**	**11**
CLDFLG	PFFLG	RSTFLG	NBFLG	UBUSY1
7:4	**3**	**2**	**1**	**0**
DID	WUON	WUDR	DCDET	MCDR

The system status flags register is a 32-bit read only register, which indicates various system information. The bits in the system status flags register SYSFLG1 are defined in Table 40.

Bit	Description
0	**MCDR**: Media changed direct read. This bit reflects the INVERTED non-latched status of the media changed input.
1	**DCDET**: This bit will be set if a non-battery operated power supply is powering the system (it is the inverted state of the nEXTPWR input pin).
2	**WUDR**: Wake up direct read. This bit reflects the non-latched state of the wakeup signal.
3	**WUON**: This bit will be set if the system has been brought out of the Standby State by a rising edge on the wakeup signal. It is cleared by a system reset or by writing to the HALT or STDBY locations.
4:7	**DID**: Display ID nibble. This 4-bit nibble reflects the latched state of the four LCD data lines. The state of the four LCD data lines is latched by the LCDEN bit, and so it will always reflect the last state of these lines before the LCD controller was enabled.
8	**CTS**: This bit reflects the current status of the clear to send (CTS) modem control input to UART1.
9	**DSR**: This bit reflects the current status of the data set ready (DSR) modem control input to UART1.
10	**DCD**: This bit reflects the current status of the data carrier detect (DCD) modem control input to UART1.
11	**UBUSY1**: UART1 transmitter busy. This bit is set while UART1 is busy transmitting data, it is guaranteed to remain set until the complete byte has been sent, including all stop bits.
12	**NBFLG**: New battery flag. This bit will be set if a low to high transition has occurred on the nBATCHG input, it is cleared by writing to the STFCLR location.
13	**RSTFLG**: Reset flag. This bit will be set if the RESET button has been pressed, forcing the nURESET input low. It is cleared by writing to the STFCLR location.
14	**PFFLG**: Power Fail Flag. This bit will be set if the system has been reset by the nPWRFL input pin, it is cleared by writing to the STFCLR location.
15	**CLDFLG**: Cold start flag. This bit will be set if the CS89712 has been reset with a power on reset, it is cleared by writing to the STFCLR location.

Table 40. SYSFLG

Bit	Description
16:21	**RTCDIV**: This 6-bit field reflects the number of 64 Hz ticks that have passed since the last increment of the RTC. It is the output of the divide by 64 chain that divides the 64 Hz tick clock down to 1 Hz for the RTC. The MSB is the 32 Hz output, the LSB is the 1 Hz output.
22	**URXFE1**: UART1 receiver FIFO empty. The meaning of this bit depends on the state of the UFIFOEN bit in the UART1 bit rate and line control register. If the FIFO is disabled, this bit will be set when the RX holding register is empty. If the FIFO is enabled, the URXFE bit will be set when the RX FIFO is empty.
23	**UTXFF1**: UART1 transmit FIFO full. The meaning of this bit depends on the state of the UFIFOEN bit in the UART1 bit rate and line control register. If the FIFO is disabled, this bit will be set when the TX holding register is full. If the FIFO is enabled, the UTXFF bit will be set when the TX FIFO is full.
24	**CRXFE**: Codec RX FIFO empty bit. This will be set if the 16-byte codec RX FIFO is empty.
25	**CTXFF**: Codec TX FIFO full bit. This will be set if the 16-byte codec TX FIFO is full.
26	**SSIBUSY**: Synchronous serial interface busy bit. This bit will be set while data is being shifted in or out of the synchronous serial interface, when clear data is valid to read.
27:28	**BOOTBIT0–1**: These bits indicate the default (power-on reset) bus width of the ROM interface. See *Memory Configuration Registers* for more details on the ROM interface bus width. The state of these bits reflect the state of Port E[0:1] during power on reset, as shown in the table below.

PE[1] (BOOTBIT1)	PE[0] (BOOTBIT0)	Boot option
0	0	32-bit
0	1	8-bit
1	0	16-bit
1	1	Reserved

29	**ID**: Will always read ‘1’ for the CS89712 device.
30:31	**VERID**: Version ID bits. These 2 bits determine the version id for the CS89712. Will read ‘10’ for the initial version.

Table 40. SYSFLG (Continued)

3.5.5 SYSFLG2 System Status Register 2 (address 0x8000.1140)

23	22	21:12	11	10:7	6
UTXFF2	URXFE2	Reserved	UBUSY2	Reserved	CKMODE
5	**4**	**3**	**2**	**1**	**0**
SS2TXUF	SS2TXFF	SS2RXFE	RESFRM	RESVAL	SS2RXOF

This register is an extension of SYSFLG1, containing status bits for backward compatibility with CL-PS7111. The bits of the second system status register are defined in Table 41.

Bit	Description
0	**SS2RXOF**: Master / slave SSI2 RX FIFO overflow. This bit is set when a write is attempted to a full RX FIFO (i.e., when RX is still receiving data and the FIFO is full). This can be cleared in one of two ways: 1. Empty the FIFO (remove data from FIFO) and then write to SRXEOF location. 2. Disable the RX (affects of disabling the RX will not take place until a full SSI2 clock cycle after it is disabled)
1	**RESVAL**: Master / slave SSI2 RX FIFO residual byte present, cleared by popping the residual byte into the SSI2 RX FIFO or by a new RX frame sync pulse.
2	**RESFRM**: Master / slave SSI2 RX FIFO residual byte present, cleared only by a new RX frame sync pulse.
3	**SS2RXFE**: Master / slave SSI2 RX FIFO empty bit. This will be set if the 16 x 16 RX FIFO is empty.
4	**SS2TXFF**: Master / slave SSI2 TX FIFO full bit. This will be set if the 16 x 16 TX FIFO is full. This will get cleared when data is removed from the FIFO or the CS89712 is reset.
5	**SS2TXUF**: Master / slave SSI2 TX FIFO Underflow bit. This will be set if there is attempt to transmit when TX FIFO is empty. This will be cleared when FIFO gets loaded with data.
6	**CKMODE**: This bit reflects the status of the CLKSEL (PE[2]) input, latched on power on reset. This bit should be low.
11	**UBUSY2**: UART2 transmitter busy. This bit is set while UART2 is busy transmitting data; it is guaranteed to remain set until the complete byte has been sent, including all stop bits.
22	**URXFE2**: UART2 receiver FIFO empty. The meaning of this bit depends on the state of the UFIFOEN bit in the UART2 bit rate and line control register. If the FIFO is disabled, this bit will be set when the RX holding register contains is empty. If the FIFO is enabled, the URXFE bit will be set when the RX FIFO is empty.
23	**UTXFF2**: UART2 transmit FIFO full. The meaning of this bit depends on the state of the UFIFOEN bit in the UART2 bit rate and line control register. If the FIFO is disabled, this bit will be set when the TX holding register is full. If the FIFO is enabled, the UTXFF bit will be set when the TX FIFO is full.

Table 41. SYSFLG2

3.6 Interrupt Registers

3.6.1 INTSR1 Interrupt Status Register 1 (address 0x8000.0240)

15	14	13	12	11	10	9	8
SSEOTI	UMSINT	URXINT1	UTXINT1	TINT	RTCMI	TC2OI	TC1OI
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EINT3	EINT2	EINT1	CSINT	MCINT	WEINT	BLINT	EXTFIQ

The interrupt status register is a 32-bit read only register. The interrupt status register reflects the current state of the first 16 interrupt sources within the CS89712. Each bit is set if the appropriate interrupt is active. The interrupt assignment is given in Table 42.

Bit	Description
0	**EXTFIQ**: External fast interrupt. This interrupt will be active if the nEXTFIQ input pin is forced low and is mapped to the FIQ input on the ARM720T processor.
1	**BLINT**: Battery low interrupt. This interrupt will be active if no external supply is present (nEXTPWR is high) and the battery OK input pin BATOK is forced low. This interrupt is deglitched with a 16 kHz clock, so it will only generate an interrupt if it is active for longer than 125 µsec. It is mapped to the FIQ input on the ARM720T processor and is cleared by writing to the BLEOI location. Note: BLINT is disabled during Snooze/the Standby States.
2	**WEINT**: Tick Watch dog expired interrupt. This interrupt will become active on a rising edge of the periodic 64 Hz tick interrupt clock if the tick interrupt is still active (i.e., if a tick interrupt has not been serviced for a complete tick period). It is mapped to the FIQ input on the ARM720T processor and the TEOI location. Notes: 1. WEINT is disabled during Snooze/the Standby States. 2. Watch dog timer tick rate is 64 Hz. 3. Watchdog timer is turned off during Snooze/the Standby States.
3	**MCINT**: Media changed interrupt. This interrupt will be active after a rising edge on the nMEDCHG input pin has been detected, This input is deglitched with a 16 kHz clock so it will only generate an interrupt if it is active for longer than 125 µsec. It is mapped to the FIQ input on the ARM7TDMI processor and is cleared by writing to the MCEOI location. On power-up, the Media change pin (nMEDCHG) is used as an input to force the processor to either boot from the internal Boot ROM, or from external memory. After power-up, the pin can be used as a general purpose FIQ interrupt pin.
4	**CSINT**: Codec sound interrupt, generated when the data FIFO has reached half full or empty (depending on the interface direction). It is cleared by writing to the COEOI location.
5	**EINT1**: External interrupt input 1. This interrupt will be active if the nEINT1 input is active (low). It is cleared by returning nEINT1 to the passive (high) state.
6	**EINT2**: External interrupt input 2. This interrupt will be active if the nEINT2 input is active (low). It is cleared by returning nEINT2 to the passive (high) state.
7	**EINT3**: Interrupt input 3 (Ethernet port). This interrupt will be active if the Ethernet port requests an interrupt. It is cleared by returning EINT3 to the passive (low) state.
8	**TC1OI**: TC1 under flow interrupt. This interrupt becomes active on the next falling edge of the timer counter 1 clock after the timer counter has under flowed (reached zero). It is cleared by writing to the TC1EOI location.

Table 42. INTSR1

Bit	Description
9	**TC2OI**: TC2 under flow interrupt. This interrupt becomes active on the next falling edge of the timer counter 2 clock after the timer counter has under flowed (reached zero). It is cleared by writing to the TC2EOI location.
10	**RTCMI**: RTC compare match interrupt. This interrupt becomes active on the next rising edge of the 1 Hz Real-Time Clock (one second later) after the 32-bit time written to the Real-Time Clock match register exactly matches the current time in the RTC. It is cleared by writing to the RTCEOI location.
11	**TINT**: 64 Hz tick interrupt. This interrupt becomes active on every rising edge of the internal 64 Hz clock signal. This 64 Hz clock is derived from the 15-stage ripple counter that divides the 32.768 kHz oscillator input down to 1 Hz for the Real-Time Clock. This interrupt is cleared by writing to the TEOI location. Note: TINT is disabled / turned off during Snooze/the Standby States.
12	**UTXINT1**: Internal UART1 transmit FIFO half-empty interrupt. The function of this interrupt source depends on whether the UART1 FIFO is enabled. If the FIFO is disabled (FIFOEN bit is clear in the UART1 bit rate and line control register), this interrupt will be active when there is no data in the UART1 TX data holding register and be cleared by writing to the UART1 data register. If the FIFO is enabled this interrupt will be active when the UART1 TX FIFO is half or more empty, and is cleared by filling the FIFO to at least half full.
13	**URXINT1**: Internal UART1 receive FIFO half full interrupt. The function of this interrupt source depends on whether the UART1 FIFO is enabled. If the FIFO is disabled this interrupt will be active when there is valid RX data in the UART1 RX data holding register and be cleared by reading this data. If the FIFO is enabled this interrupt will be active when the UART1 RX FIFO is half or more full or if the FIFO is non empty and no more characters have been received for a three character time out period. It is cleared by reading all the data from the RX FIFO.
14	**UMSINT**: Internal UART1 modem status changed interrupt. This interrupt will be active if either of the two modem status lines (CTS or DSR) change state. It is cleared by writing to the UMSEOI location.
15	**SSEOTI**: Synchronous serial interface end of transfer interrupt. This interrupt will be active after a complete data transfer to and from the external ADC has been completed. It is cleared by reading the ADC data from the SYNCIO register.

Table 42. INTSR1 (Continued)

3.6.2 INTMR1 Interrupt Mask Register 1 (address 0x8000.0280)

15	14	13	12	11	10	9	8
SSEOTI	UMSINT	URXINT	UTXINT	TINT	RTCMI	TC2OI	TC1OI
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EINT3	EINT2	EINT1	CSINT	MCINT	WEINT	BLINT	EXTFIQ

This interrupt mask register is a 32-bit read / write register, which is used to selectively enable any of the first 16 interrupt sources within the CS89712. The four shaded interrupts all generate a fast interrupt request to the ARM720T processor (FIQ), this will cause a jump to processor virtual address 0000.0001C. All other interrupts will generate a standard interrupt request (IRQ), this will cause a jump to processor virtual address 0000.00018. Setting the appropriate bit in this register enables the corresponding interrupt. All bits are cleared by a system reset. Please refer to Section 3.6, *"Interrupt Registers"* for individual bit details.

3.6.3 INTSR2 Interrupt Status Register 2 (address 0x8000.1240)

15:14	13	12	11:3	2	1	0
Reserved	URXINT2	UTXINT2	Reserved	SS2TX	SS2RX	KBDINT

This register is an extension of INTSR1, containing status bits for backward compatibility with CL-PS7111. The interrupt status register also reflects the current state of the new interrupt sources within the CS89712. Each bit is set if the appropriate interrupt is active. The interrupt assignment is given in Table 43.

Bit	Description
0	**KBDINT**: Keyboard interrupt. This interrupt is generated whenever a key is pressed, from the logical OR of the first 6 or all 8 of the Port A inputs (depending on the state of the KBD6 bit in the SYSCON2 register. The interrupt request is latched and can be de-asserted by writing to the KBDEOI location. Note: KBDINT is not deglitched.
1	**SS2RX**: Synchronous serial interface 2 receives FIFO half or greater full interrupt. This is generated when RX FIFO contains 8 or more half-words. This interrupt is cleared only when the RX FIFO is emptied or one SSI2 clock after RX is disabled.
2	**SS2TX**: Synchronous serial interface 2 transmit FIFO less than half empty interrupt. This is generated when TX FIFO contains fewer than 8 byte pairs. This interrupt gets cleared by loading the FIFO with more data or disabling the TX. One synchronization clock required when disabling the TX side before it takes effect.
12	**UTXINT2**: UART2 transmit FIFO half empty interrupt. The function of this interrupt source depends on whether the UART2 FIFO is enabled. If the FIFO is disabled (FIFOEN bit is clear in the UART2 bit rate and line control register), this interrupt will be active when there is no data in the UART2 TX data holding register and be cleared by writing to the UART2 data register. If the FIFO is enabled, this interrupt will be active when the UART2 TX FIFO is half or more empty and is cleared by filling the FIFO to at least half full.
13	**URXINT2**: UART2 receive FIFO half full interrupt. The function of this interrupt source depends on whether the UART2 FIFO is enabled. If the FIFO is disabled, this interrupt will be active when there is valid RX data in the UART2 RX data holding register and be cleared by reading this data. If the FIFO is enabled, this interrupt will be active when the UART2 RX FIFO is half or more full or if the FIFO is non-empty, and no more characters have been received for a three-character time-out period, t is cleared by reading all the data from the RX FIFO.

Table 43. INSTR2

3.6.4 INTMR2 Interrupt Mask Register 2 (address 0x8000.1280)

15:14	13	12	11:3	2	1	0
Reserved	URXINT2	UTXINT2	Reserved	SS2TX	SS2RX	KBDINT

This register is an extension of INTMR1, containing interrupt mask bits for the backward compatibility with the CL-PS7111. Please refer to INTSR2 for individual bit details.

3.6.5 INTSR3 Interrupt Status Register 3 (address 0x8000.2240)

7:1	0
Reserved	DAIPINT

This register is an extension of INTSR1 and INTSR2 containing status bits for the new features of the CS89712. Each bit is set if the appropriate interrupt is active. The interrupt assignment is given in Table 44.

Bit	Description
0	**DAIINT**: DAI interface interrupt. The cause must be determined by reading the DAI status register. It is mapped to the FIQ interrupt on the ARM720T processor

Table 44. INTSR3

3.6.6 INTMR3 Interrupt Mask Register 3 (address 0x8000.2280)

7:1	0
Reserved	DAIINT

This register is an extension of INTMR1 and INTMR2, containing interrupt mask bits for the new features of the CS89712. Please refer to INTSR3 for individual bit details.

3.7 Expansion Memory Configuration Registers

3.7.1 MEMCFG1 Memory Configuration Register 1 (address 0x8000.0180)

31:24	23:16	15:8	7:0
nCS[3] configuration	nCS[2] configuration	nCS[1] configuration	nCS[0] configuration

Expansion and ROM space is selected by one of eight chip selects. One of the chip selects (nCS[6]) is used internally for the on-chip SRAM, and the configuration is hardwired for 32-bit-wide, minimum- wait-state operation. nCS[7] is used for the on-chip Boot ROM and the configuration field is hardwired for 8-bit-wide, minimum-wait-state operation. Data written to the configuration fields for either nCS[6] or nCS7 will be ignored. Two of the chip selects (nCS[4:5]) can be used to access two CL-PS6700 PC CARD controller devices, and when either of these interfaces is enabled, the configuration field for the appropriate chip select in the MEMCFG2 register is ignored. When the PC CARD1 or 2 control bit in the SYSCON2 register is disabled, then nCS[4] and nCS[5] are active as normal and can be programmed using the relevant fields of MEMCFG2, as for the other four chip selects. All of the six external chip selects are active for 256 Mbytes and the timing and bus transfer width can be programmed individually. This is accomplished by programming the six-byte-wide fields contained in two 32-bit registers, MEMCFG1 and MEMCFG2. All bits in these registers are cleared by a system reset (except for the nCS[6] and nCS[7] configurations).

The Memory Configuration Register 1 is a 32-bit read / write register which sets the configuration of the four expansion and ROM selects nCS[0:3]. Each select is configured with a 1-byte field starting with expansion select 0.

3.7.2 MEMCFG2 Memory Configuration Register 2 (address 0x8000.01C0)

31:24	23:16	15:8	7:0
(Boot ROM)	(Local SRAM)	nCS[5] configuration	nCS[4] configuration
7	**6**	**5:2**	**1:0**
CLKENB	SQAEN	Wait States Field	Bus width

The Memory Configuration Register 2 is a 32-bit read / write register which sets the configuration of the two expansion and ROM selects nCS[4:5]. Each select is configured with a 1-byte field starting with expansion select 4.

Each of the six non-reserved byte fields for chip select configuration in the memory configuration registers are identical and define the number of wait states, the bus width, enable EXPCLK output during accesses and enable sequential mode access. This byte field is defined below. This arrangement applies to nCS[0:3], and nCS[4:5] when the PC CARD enable bits in the SYSCON2 register are not set. The state of these bits is ignored for the Boot ROM and local SRAM fields in the MEMCFG2 register.

Table 45 defines the bus width field. Note that the effect of this field is dependent on the two BOOTBIT bits that can be read in the SYSFLG1 register. All bits in the memory configuration register are cleared by a system reset, and the state of the BOOTBIT bits are determined by Port E bits 0 and 1 on the CS89712 during power-on reset. The state of PE[1] and PE[0] determine whether the CS89712 is going to boot from either 32-bit-wide, 16-bit-wide or 8-bit-wide ROMs.

Table 46 shows the values for the wait states for random and sequential wait states at 18 MHz bus. At 36 MHz bus rate, the encoding becomes more complex. Table 47 preserves compatibility with the previous devices, while allowing the previously unused bit combinations to specify more variations of random and sequential wait states.

Bus Width Field	BOOTBIT1	BOOTBIT0	Expansion Transfer Mode	Port E bits 1,0 during NPOR reset
00	0	0	32-bit wide bus access	Low, Low
01	0	0	16-bit wide bus access	Low, Low
10	0	0	8-bit wide bus access	Low, Low
11	0	0	Reserved	Low, Low
00	0	1	8-bit wide bus access	Low, High
01	0	1	Reserved	Low, High
10	0	1	32-bit wide bus access	Low, High
11	0	1	16-bit wide bus access	Low, High
00	1	0	16-bit wide bus access	High, Low
01	1	0	32-bit wide bus access	High, Low
10	1	0	Reserved	High, Low
11	1	0	8-bit wide bus access	High, Low

Table 45. Values of the Bus Width Field

Note: See *AC Characteristics* for more detail on bus timing.

The memory area decoded by CS[6] is reserved for the on-chip SRAM, hence this does not require a configuration field in MEMCFG2. It is automatically set up for 32-bit-wide, no-wait-state accesses. For the Boot ROM, it is automatically set up for 8-bit, no wait state accesses.
Chip selects nCS[4] and nCS[5] are used to select two CL-PS6700 PC CARD controller devices. These have a multiplexed 16-bit wide address / data interface, and the configuration bytes in the MEMCFG2 register have no meaning when these interfaces are enabled.

Value	No. of Wait States Random	No. of Wait States Sequential
00	4	3
01	3	2
10	2	1
11	1	0

Table 46. Values of the Wait State Field at 18 MHz

Bit 3	Bit 2	Bit 1	Bit 0	Wait States Random	Wait States Sequential
0	0	0	0	8	3
0	0	0	1	7	3
0	0	1	0	6	3
0	0	1	1	5	3
0	1	0	0	4	2
0	1	0	1	3	2
0	1	1	0	2	2
0	1	1	1	1	2
1	0	0	0	8	1
1	0	0	1	7	1
1	0	1	0	6	1
1	0	1	1	5	1
1	1	0	0	4	0
1	1	0	1	3	0
1	1	1	0	2	0
1	1	1	1	1	0

Table 47. Values of the Wait State Field at 36 MHz

Bit	Description
6	**SQAEN**: Sequential access enable. Setting this bit will enable sequential accesses that are on a quad word boundary to take advantage of faster access times from devices that support page mode. The sequential access will be faulted after four words (to allow video refresh cycles to occur), even if the access is part of a longer sequential access. In addition, when this bit is not set, non-sequential accesses will have a single idle cycle inserted at least every four cycles so that the chip select is de-asserted periodically between accesses for easier debug.
7	**CLKENB**: Expansion clock enable. Setting this bit enables the EXPCLK to be active during accesses to the selected expansion device. This will provide a timing reference for devices that need to extend bus cycles using the EXPRDY input. Back-to-back (but not necessarily page mode) accesses will result in a continuous clock.

Table 48. MEMCFG

3.8 Timer / Counter Registers

3.8.1 TC1D Timer Counter 1 Data Register (address 0x8000.0300)

The timer counter 1 data register is a 16-bit read / write register which sets and reads data to TC1. Any value written will be decremented on the next rising edge of the clock.

3.8.2 TC2D Timer Counter 2 Data Register (address 0x8000.0340)

The timer counter 2 data register is a 16-bit read / write register which sets and reads data to TC2. Any value written will be decremented on the next rising edge of the clock.

3.8.3 RTCDR Real-Time Clock Data Register (address 0x8000.0380)

The Real-Time Clock data register is a 32-bit read / write register, which sets and reads the binary time in the RTC. Any value written will be incremented on the next rising edge of the 1 Hz clock. This register is reset only by nPOR.

3.8.4 RTCMR Real-Time Clock Match Register (address 0x8000.03C0)

The Real-Time Clock match register is a 32-bit read / write register, which sets and reads the binary match time to RTC. Any value written will be compared to the current binary time in the RTC, if they match it will assert the RTCMI interrupt source. This register is reset only by nPOR.

3.9 Miscellaneous Registers

3.9.1 LEDFLSH Register (address 0x8000.22C0)

6	5:2	1:0
Enable	Duty ratio	Flash rate

The output is enabled whenever LEDFLSH[6] = 1. When enabled, PDDDR[0] needs to be configured as an output pin and the bit cleared to ‘0’ (See Section 3.4.6, *“PDDDR Port D Data Direction Register (address 0x8000.0043)”*). When the LED Flasher is disabled, the pin defaults to being used as Port D bit 0. Thus, this will ensure that the LED will be off when disabled.

The flash rate is determined by the LEDFLSH[1:0] bits, in the following way:

LEDFLSH[1:0]	Flash Period (sec)
00	1
01	2
10	3
11	4

Table 49. LED Flash Rates

LEDFLSH[5:2]	Duty Ratio (time on: time off)	LEDFLSH[5:2]	Duty Ratio (time on: time off)
0000	01:15	1000	09:07
0001	02:14	1001	10:06
0010	03:13	1010	11:05

Table 50. LED Duty Ratio

LEDFLSH[5:2]	Duty Ratio (time on: time off)	LEDFLSH[5:2]	Duty Ratio (time on: time off)
0011	04:12	1011	12:04
0100	05:11	1100	13:03
0101	06:10	1101	14:02
0110	07:09	1110	15:01
0111	08:08	1111	16:00 (continually on)

Table 50. LED Duty Ratio (Continued)

3.9.2 SDCONF SDRAM Control Register (address 0x8000.2300)

31:11	10	9	8:7	6:5	4:2	1:0
Reserved	SDACTIVE	CLKCTL	SDWIDTH	SDSIZE	Reserved	CASLAT

Bit	Description
1:0.	How many clock cycles after CAS before the device is ready for reading or writing. . . ‘00’ => Reserved. . . , ‘01’ => Reserved, ‘10’ => CAS Latency = 2, ‘11’ => CAS Latency = 3. . . . . The default value is ‘10’ for CAS latency = 2.
4:2.	Reserved.
6:5.	The capacity of each SDRAM. The values are: ‘00’=>16Mits, ‘01’=>64Mbits, ‘10’=>128Mbits, ‘11’=>256Mbits.
8:7.	The width of each SDRAM. ‘00’=>4bits, ‘01’=>8bits, ‘10’=>16 bits, ‘11’=>32 bits.
9.	Control over the SDRAM clock. ‘0’=> SDRAM clock is permanently enabled except when in standby mode. ‘1’=>SDRAM clock stops when the CS89712 is put into inactive mode i.e., SDACTIVE = ‘0’, or when in standby mode.
10.	Enables the SDRAM controller: ‘0’ disables, ‘1’ enables. The SDRAM controller will only initialize if SDACTIVE is set to 1. After initialization, resetting this parameter will cause the SDRAM controller to enter an inactive state. It will remain in this state until SDACTIVE is set to 1.
31:11.	Reserved.

3.9.3 SDRFPR SDRAM Refresh Period Register (address 0x8000.2340)

31:16	15:0
Reserved	REFRATE

This 16-bit read/write register sets the interval between SDRAM refresh commands. The value programmed is the interval in BLCK cycles e.g. for a 16 µs refresh period with a BCLK of 36 MHz, the following value should be used:

$16x10^{-6} * 36x10^{6} = 576$

The refresh timer is set to 256 by nPOR to ensure a refresh time of better than 16 µs. This register should not be programmed to a value below 2 otherwise the internal bus may become locked.

This register replaces DPFPR, which is no longer active. Writes to this register are ignored. Reads from this register will produce unpredictable results.

3.9.4 PMPCON Pump Control Register (address 0x8000.0400)

11:8	7:4	3:0
Drive 1 pump ratio	Drive 0 from AC source ratio	Drive 0 from battery ratio

The Pulse Width Modulator (PWM) pump control register is a 16-bit read / write register which sets and controls the variable mark space ratio drives for the two PWMs. All bits in this register are cleared by a system reset. (The top four bits are unused. They should be written as zeroes, and will read as undefined).

The state of the output drive pins is latched during power on reset, this latched value is used to determine the polarity of the drive output. The sense of the PWM control lines is summarized in Table 51.

Initial State of Drive 0 or Drive 1 During Power on Reset	Sense of Drive 0 or Drive 1	Polarity of Bias Voltage
Low	Active high	+ve
High	Active low	-ve

Table 51. Sense of PWM control lines

External input pins that would normally be connected to the output from comparators monitoring the PWM output are also used to enable these clocks. These are the FB[0:1] pins. When FB[0] is high, the PWM is disabled. The same applies to FB[1]. They are read upon power-up.

Note: To maximize power savings, the drive ratio fields should be used to disable the PWMs, instead of the FB pins. The clocks that source the PWMs are disabled when the drive ratio fields are zeroed.4.

Bit	Description
0:3	**Drive 0 from battery**: This 4-bit field controls the “on” time for the Drive 0 PWM pump while the system is powered from batteries. Setting these bits to 0 disables this pump, while setting these bits to 1 allows the pump to be driven in a 1:16 duty ratio, 2 in a 2:16 duty ratio etc. up to a 15:16 duty ratio. An 8:16 duty ratio results in a square wave of 96 kHz when operating with an 18.432 MHz master clock.
4:7	**Drive 0 from AC**: This 4-bit field controls the “on” time for the Drive 0 DC to DC pump, while the system is powered from a non-battery type power source. Setting these bits to 0 disables this pump, setting these bits to 1 allows the pump to be driven in a 1:16 duty ratio, 2 in a 2:16 duty ratio, etc. up to a 15:16 duty ratio. An 8:16 duty ratio results in a square wave of 96 kHz when operating with an 18.432 MHz master clock. Note: The CS89712 monitors the power supply input pins (i.e., BATOK and NEXTPWR) to determine which of the above fields to use.
8:11	**Drive 1 pump ratio**: This 4-bit field controls the “on” time for the drive1 PWM pump. Setting these bits to 0 disables this pump, while setting these bits to 1 allows the pump to be driven in a 1:16 duty ratio, 2 in a 2:16 duty ratio, etc. up to a 15:16 duty ratio. An 8:16 duty ratio results in a square wave of 96 kHz when operating with an 18.432 MHz master clock.

Table 52. PMPCON

3.9.5 CODR — The CODEC Interface Data Register (address 0x8000.0440)

The CODR register is an 8-bit read / write register, to be used with the codec interface. This is selected by the appropriate setting of bit 0 (SERSEL) of the SYSCON2 register. Data written to or read from this register is pushed or popped onto the appropriate 16-byte FIFO buffer. Data from this buffer is then serialized and sent to or received from the codec sound device. When the codec is enabled, the codec interrupt CSINT is

generated repetitively at 1/8th the byte transfer rate and the FIFO state can be read in the system flags register. The net data transfer rate to / from the codec device is 8 kBytes/s, giving an interrupt rate of 1 kHz.

3.9.6 STFCLR Clear all "Start Up Reason" Flags Location (address 0x8000.05C0)

A write to this location will clear all the "Start Up Reason" flags in the system flags status register SYSFLG. The 'Start Up Reason' flags should first read to determine the reason why the chip was started (i.e., a new battery was installed). Any value may be written to this location.

3.10 UART Registers

3.10.1 UARTDR1–2, UART1–2 Data Registers (address 0x8000.0480 and 0x8000.1480)

10	9	8	7:0
OVERR	PARERR	FRMERR	RX data

The UARTDR registers are 11-bit read and 8-bit write registers for all data transfers to or from the internal UARTs 1 and 2.

Data written to these registers is pushed onto the 16-byte data TX holding FIFO if the FIFO is enabled. If not it is stored in a one byte holding register. This write will initiate transmission from the UART.

The UART data read registers are made up of the 8-bit data byte received from the UART together with three bits of error status. If the FIFO is enabled, data read from this register is popped from the 16 byte data RX FIFO. If the FIFO is not enabled, it is read from a one byte buffer register containing the last byte received by the UART. If it is enabled, data received and error status is automatically pushed onto the RX FIFO. The RX FIFO is 10-bits wide by 16 deep.

Note: These registers should be accessed as words.

Bit	Description
8	**FRMERR**: UART framing error. This bit is set if the UART detected a framing error while receiving the associated data byte. Framing errors are caused by non-matching word lengths or bit rates.
9	**PARERR**: UART parity error. This bit is set if the UART detected a parity error while receiving the data byte.
10	**OVERR**: UART over-run error. This bit is set if more data is received by the UART and the FIFO is full. The overrun error bit is not associated with any single character and so is not stored in the FIFO. If this bit is set, the entire contents of the FIFO is invalid and should be cleared. This error bit is cleared by reading the UARTDR register.

Table 53. UARTDR1-2 UART1-2

3.10.2 UBRLCR1–2 UART1–2 Bit Rate and Line Control Registers (address 0x8000.04C0 and 0x8000.14C0)

31:19	18:17	16	15	14	13	12	11:0
	WRDLEN	FIFOEN	XSTOP	EVENPRT	PRTEN	BREAK	Bit rate divisor

The bit rate divisor and line control register is a 19-bit read / write register. Writing to these registers sets the bit rate and mode of operation for the internal UARTs.

Bit	Description
0:11	**Bit rate divisor**: This 12-bit field sets the bit rate. If the system is operating from the PLL clock, then the bit rate divider is fed by a clock frequency of 3.6864 MHz, which is then further divided internally by 16 to give the bit rate. The formula to give the divisor value for any bit rate when operating from the PLL clock is: Divisor = 230400 / (bit rate divisor + 1). A value of zero in this field is illegal when running from the PLL clock. The tables below show some example bit rates with the corresponding divisor value.The table below shows the bit rates available for 18.432 MHz operation.

Divisor Value	Bit Rate Running From the PLL Clock
0	—
1	115200
2	76800
3	57600
5	38400
11	19200
15	14400
23	9600
95	2400
191	1200
2094	110

12	**BREAK**: Setting this bit will drive the TX output active (high) to generate a break.
13	**PRTEN**: Parity enable bit. Setting this bit enables parity detection and generation
14	**EVENPRT**: Even parity bit. Setting this bit sets parity generation and checking to even parity, clearing it sets odd parity. This bit has no effect if the PRTEN bit is clear.
15	**XSTOP**: Extra stop bit. Setting this bit will cause the UART to transmit two stop bits after each data byte, while clearing it will transmit one stop bit after each data byte.
16	**FIFOEN**: Set to enable FIFO buffering of RX and TX data. Clear to disable the FIFO (i.e., set its depth to one byte).

Bit	Description
17:18	**WRDLEN**: This two bit field selects the word length according to the table below.

WRDLEN	Word Length
00	5 bits
01	6 bits
10	7 bits
11	8 bits

Table 54. UBRLCR1-2 UART1-2 (Continued)

3.11 LCD Registers

3.11.1 LCDCON — The LCD Control Register (address 0x8000.02C0)

31	30	29:25	24:19	18:13	12:0
GSMD2	GSMD1	AC prescale	Pixel prescale	Line length	Video buffer size

The LCD control register is a 32-bit read / write register that controls the size of the LCD screen and the operating mode of the LCD controller. Refer to the system description of the LCD controller for more information on video buffer mapping, and for details of how to program the LCD control register for use in Snooze State.

The LCDCON register should only be reprogrammed when the LCD controller is disabled.

Bit	Description
0:12	**Video buffer size:** The video buffer size field is a 13-bit field that sets the total number of bits x 128 (quad words) in the video display buffer. This is calculated from the formula: Video buffer size = (Total bits in video buffer / 128) – 1 i.e., for a 640 x 240 LCD and 4 bits-per-pixel, the size of the video buffer is equal to 614400 bits. Video buffer = 640 x 240 x 4=614400 bits Video buffer size field = (614400 / 128) – 1 = 4799 or 0x12BF hex. If Snooze State is to be used with the LCD controller enabled, then the value programmed into this register should not be less than 3. The minimum value allowed is 3 for this bit field.
13:18	**Line length**: The line length field is a 6-bit field that sets the number of pixels in one complete line. This field is calculated from the formula: line length = (Number of pixels in line / 16) – 1 i.e., for 640 x 240 LCD Line length = (640 / 16) – 1 = 39 or 0x27 hex. The minimum value that can be programmed into this register is a 1 (i.e., 0 is not a legal value).

Table 55. LCDCON

Bit	Description
19:24	**Pixel prescale**: The pixel prescale field is a 6-bit field that sets the pixel rate prescale. The pixel rate is always derived from a 36.864 MHz clock and is calculated from the formula: Pixel rate (MHz) = 36.864 / (Pixel prescale + 1) The pixel prescale value can be expressed in terms of the LCD size by the formula: When the CS89712 is operating @ 18.432 MHz: Pixel prescale = (36864000 / (Refresh Rate x Total pixels in display)) – 1 Refresh Rate is the screen refresh frequency (70 Hz to avoid flicker) The value should be rounded down to the nearest whole number and zero is illegal and will result in no pixel clock. EXAMPLE: For a system being operated in the 18.432–73.728 MHz mode, with a 640 x 240 screen size, and 70 Hz screen refresh rate desired, the LCD Pixel prescale equals 36.864E6 / (70 x 640x240) – 1 = 2.428 Rounding 2.428 down to the nearest whole number equals 2. This gives an actual pixel rate of 36.864E6 / (2+1) = 12.288 MHz, which gives an actual refresh frequency of 12.288E6 / (640x240) = 80 Hz. Note: As the CL[2] low pulse time is doubled after every CL[1] high pulse this refresh frequency is only an approximation, the accurate formula is 12.288E6 / ((640x240)+120) = 79.937 Hz.
25:29	**AC prescale**: The AC prescale field is a 5-bit number that sets the LCD AC bias frequency. This frequency is the required AC bias frequency for a given manufacturer's LCD plate. This frequency is derived from the frequency of the line clock (CL[1]). The LCD M signal will toggle after n+1 counts of the line clock (CL[1]) where n is the number programmed into the AC prescale field. This number must be chosen to match the manufacturer's recommendation. This is normally 13, but must not be exactly divisible by the number of lines in the display.
30	**GSMD1**: Grayscale mode bit number 1. Setting this bit enables 2 or 4 bits-per-pixel (01 or 11, respectively) grayscaling. (Also see the GSMD2 bit definition.) Clearing this bit enables 1 bpp (00) gray scaling only. Note: Gray scaling is always enabled when using the EP72xx LCD Controller. Direct mapping of the frame buffer bits to the LCD display is not supported. However, this can be accomplished by simply programming the palette register contents to correspond with the frame buffer bit value (i.e., for 1 bpp (00) Direct mapping program the PALLSW register nibble [3:0] with zeros, and nibble [7:4] with ones.)
31	**GSMD2**: Grayscale mode bit number 2. Setting this bit enables 4 bpp (11) gray scaling (15 gray scales.) Clearing this bit enables 2 bits-per-pixel (01) gray scaling.

Table 55. LCDCON (Continued)

3.11.2 PALLSW Least Significant Word — LCD Palette Register (address 0x8000.0580)

31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
Grayscale value for pixel value 7	Grayscale value for pixel value 6	Grayscale value for pixel value 5	Grayscale value for pixel value 4	Grayscale value for pixel value 3	Grayscale value for pixel value 2	Grayscale value for pixel value 1	Grayscale value for pixel value 0

The least and most significant word LCD palette registers make up a 64-bit read / write register which maps the logical pixel value to a physical grayscale level. The 64-bit register is made up of 16 x 4-bit nibbles, each nibble defines the grayscale level associated with the appropriate pixel value. If the LCD controller is operating in two bits-per-pixel, only the lower 4 nibbles are valid (D[15:0] in the least significant word). Similarly, one bit-per-pixel means only the lower 2 nibbles are valid (D[7:0]) in the least significant word.

3.11.3 PALMSW Most Significant Word — LCD Palette Register (address 0x8000.0540)

31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
Grayscale value for pixel value 15	Grayscale value for pixel value 14	Grayscale value for pixel value 13	Grayscale value for pixel value 12	Grayscale value for pixel value 11	Grayscale value for pixel value 10	Grayscale value for pixel value 9	Grayscale value for pixel value 8

The pixel to grayscale level assignments and the actual physical color and pixel duty ratio for the grayscale values are shown in Table 56. Note that colors 8–15 are the inverse of colors 7–0 respectively. This means that colors 7 and 8 are identical. Therefore, in reality only 15 grayscales available, not 16. The steps in the grayscale are non-linear, but have been chosen to give a close approximation to perceived linear grayscales. The is due to the eye being more sensitive to changes in gray level close to 50% gray (See *PALLSW* description).

Grayscale Value	Duty Cycle	% Pixels Lit	% Step Change
0	0	0%	11.1%
1	1/9	11.1%	8.9%
2	1/5	20.0%	6.7%
3	4/15	26.7%	6.6%
4	3/9	33.3%	6.7%
5	2/5	40.0%	5.4%
6	4/9	44.4%	5.6%
7	1/2	50.0%	0.0%
8	1/2	50.0%	5.6%
9	5/9	55.6%	5.4%
10	3/5	60.0%	6.7%
11	6/9	66.7%	6.6%
12	11/15	73.3%	6.7%
13	4/5	80.0%	8.9%
14	8/9	88.9%	11.1%
15	1	100%	

Table 56. Grayscale Value to Color Mapping

3.11.4 FBADDR LCD Frame Buffer Start Address (address 0x8000.1000)

This register contains the start address for the LCD Frame Buffer. It is assumed that the frame buffer starts at location 0x0000000 within each chip select memory region. Therefore, the value stored within the FBADDR register is only the value of the chip select where the frame buffer is located. On reset, this will be set to 0xC. The register is 4 bits wide (bits [3:0]). This register must only be reprogrammed when the LCD is disabled (i.e., setting the LCDEN bit within SYSCON2 low), or during the period after exit from Snooze State before the LCDSNZE bit has been reset (i.e. while data is still being displayed from the on-chip SRAM in 1 bit per pixel mode).

3.12 SSI Register

3.12.1 SYNCIO Synchronous Serial ADC Interface Data Register (address 0x8000.0500)

In the default mode, the bits in SYNCIO have the following meaning:

31:15	14	13	12:8	7:0
Reserved	TXFRMEN	SMCKEN	Frame length	ADC Configuration Byte

Whereas in extended mode, the following applies:

15	14	13	12:7	6:0
Reserved	TXFRMEN	SMCKEN	Frame length	ADC Configuration Length

Note: The frame length in extended mode is 6 bits wide to allow up to 16 write bits, 1 null bit and 16 read bits (= 33 cycles).

SYNCIO is a 32-bit read / write register. The data written to the SYNCIO register configures the master only SSI. In default mode, the least significant byte is serialized and transmitted out of the synchronous serial interface1 (i.e., SSI1) to configure an external ADC, MSB first. In extended mode, a variable number of bits are sent from SYNCIO[16:31] as determined by the ADC Configuration Length. The transfer clock will automatically be started at the programmed frequency and a synchronization pulse will be issued. The ADCIN pin is sampled on every positive going clock edge (or the falling clock edge, if ADCCKNSEN in SYSCON3 is set) and the result is shifted in to the SYNCIO read register.

During data transfer, the SSIBUSY bit is set high; at the end of a transfer the SSEOTI interrupt will be asserted. To clear the interrupt the SYNCIO register must be read. The data read from the SYNCIO register is the last sixteen bits shifted out of the ADC.

The length of the data frame can be programmed by writing to the SYNCIO register. This allows many different ADCs to be accommodated. The device is SPI- / Microwire-compatible (transfers are in multiples of 8 bits). However, to be compatible with some non-SPI / Microwire devices, the data written to the ADC device can be anything between 8 to 16 bits. This is user-definable per the ADC Configuration Extension section of the SYNCIO register.

Bit	Description
0:7 or 0:6	**ADC Configuration Byte**: When the ADCCON control bit in the SYSCON3 register = 0, this is the 8-bit configuration data to be sent to the ADC. When the ADCCON control bit in the SYSCON3 register = 1, this field determines the length of the ADC configuration data held in the ADC Configuration Extension field for sending to the ADC.

Table 57. SYNCIO

Bit	Description
8:12 or 7:12	**Frame length**: The Frame Length field is the total number of shift clocks required to complete a data transfer. In default mode, MAX148/9 (and for many ADCs), this is 25 = (8 for configuration byte + 1 null bit + 16 bits result). In extended mode, AD7811/12, this is 23 = (10 for configuration byte + 3 null + 10 bits result).
13	**SMCKEN**: Setting this bit will enable a free running sample clock at twice the programmed ADC clock frequency to be output on the SMPLCK pin.
14	**TXFRMEN**: Setting this bit will cause an ADC data transfer to be initiated. The value in the ADC configuration field will be shifted out to the ADC and depending on the frame length programmed, a number of bits will be captured from the ADC. If the SYNCIO register is written to with the TXFRMEN bit low, no ADC transfer will take place, but the Frame length and SMCKEN bits will be affected.
16:31	**ADC Configuration Extension**: When the ADCCON control bit in the SYSCON3 register = 0, this field is ignored for compatibility with the CL-PS7111. When the ADCCON control bit in the SYSCON3 register = 1, this field is the configuration data to be sent to the ADC. The ADC Configuration Extension field length is determined by the value held in the ADC Configuration Length field (SYNCIO[6:0]).

Table 57. SYNCIO (Continued)

3.13 End Of Interrupt Locations

The ‘End of Interrupt’ locations that follow are written to after the appropriate interrupt has been serviced. The write is performed to clear the interrupt status bit, so that other interrupts can be serviced. Any value may be written to these locations.

3.13.1 BLEOI Battery Low End of Interrupt (address 0x8000.0600)

A write to this location clears the interrupt generated by a low battery (falling edge of BATOK with nEXTPWR high).

3.13.2 MCEOI Media Changed End of Interrupt (address 0x8000.0640)

A write to this location will clear the interrupt generated by a falling edge of the nMEDCHG input pin.

3.13.3 TEOI Tick End of Interrupt Location (address 0x8000.0680)

A write to this location will clear the current pending tick interrupt and tick watch dog interrupt.

3.13.4 TC1EOI TC1 End of Interrupt Location (address 0x8000.06C0)

A write to this location will clear the under flow interrupt generated by TC1.

3.13.5 TC2EOI TC2 End of Interrupt Location (address 0x8000.0700)

A write to this location will clear the under flow interrupt generated by TC2.

3.13.6 RTCEOI RTC Match End of Interrupt (address 0x8000.0740)

A write to this location will clear the RTC match interrupt

3.13.7 UMSEOI UART1 Modem Status Changed End of Interrupt (address 0x8000.0780)

A write to this location will clear the modem status changed interrupt.

3.13.8 COEOI Codec End of Interrupt Location (address 0x8000.07C0)

A write to this location clears the sound interrupt (CSINT).

3.13.9 KBDEOI Keyboard End of Interrupt Location (address 0x8000.1700)

A write to this location clears the KBDINT keyboard interrupt.

3.13.10 SRXEOF End of Interrupt Location (address 0x8000.1600)

A write to this location clears the SSI2 RX FIFO overflow status bit.

3.14 State Control Registers

3.14.1 STDBY Enter the Standby State Location (address 0x8000.0840)

A write to this location will put the system into the Standby State by halting the main oscillator. A write to this location while there is an active interrupt will have no effect.

Notes:
1. Before entering the Standby State, the LCD Controller should be disabled. The LCD controller should be enabled on exit from the Standby State.
2. If the CS89712 is attempting to get into the Snooze/Standby State when there is a pending interrupt request, it will not enter into the low power mode. The instruction will get executed, but the processor will ignore the command.

3.14.2 SNOOZE Enter Snooze State location

A write to this location will put the system into the Snooze State. The main clock will not be stopped in this state. It is required to continue displaying a reduced LCD buffer on the first few lines of the display, and the DC pump block will continue to generate the drive signals for external DC converter circuitry. Otherwise, clocks to all parts of the device will be disabled. The device will automatically switch the DRAMs into self refresh if the RFSHEN bit is set in the DRAM refresh period register. All transitions to the Snooze State are synchronized with DRAM cycles. A write to this location while there is an active interrupt will have no effect. Before entering this state, the data to be displayed in Snooze State must be transferred under program control into the on-chip SRAM (see section on Snooze State), and the LCDCON register and frame buffer start address must be reprogrammed accordingly. On exit from Snooze State (via enabled interrupt or wakeup event) program execution will continue from the next instruction after the write to the SNOOZE location.

3.14.3 HALT Enter the Idle State Location (address 0x8000.0800)

A write to this location will put the system into the Idle State by halting the clock to the processor until an interrupt is generated. A write to this location while there is an active interrupt will have no effect.

3.14.4 SNZDISP Snooze Mode Display Size

This register contains the number of words to be displayed from the on-chip SRAM in Snooze State, minus 1. It is a 13-bit register, allowing up to 816 K words to be displayed. Because the on-chip SRAM data is always displayed at 1-bit-per-pixel in the Snooze State, the value programmed into this register is the number of pixels to be displayed divided by 32. For a half-size VGA screen the number of words to be programmed is (640 x 240 x 1)/32 = 4.8 K words. This is the maximum value that should be programmed in this register in Snooze State for a half-size VGA display. To avoid any possibility of ‘snow’ on the display when operating at high frequency, this register should always be programmed with a minimum value of 7. Correct operation is not guaranteed below this value. The number of pixels displayed should be divisible by the number of pixels in a single line of the display.

3.15 SS2 Registers

3.15.1 SS2DR Synchronous Serial Interface 2 Data (address 0x8000.1500)

This is the 16-bit-wide data register for the full-duplex master / slave SSI2 synchronous serial interface. Writing data to this register will initiate a transfer. Writes need to be word writes and the bottom 16 bits are transferred to the TX FIFO. Reads will be 32 bits as well with the lower 16 bits containing RX data, and the upper 16-bits should be ignored. Although the interface is byte-oriented, data is written in two bytes at a time to allow higher bandwidth transfer. It is up to the software to assemble the bytes for the data stream in an appropriate manner.

All reads / writes to this register must be word reads / writes.

3.15.2 SS2POP Synchronous Serial Interface 2 Pop Residual Byte (address 0x8000.16C0)

This is a write-only location which will cause the contents of the RX shift register to be popped into the RX FIFO, thus enabling a residual byte to be read. The data value written to this register is ignored. This location should be used in conjunction with the RESVAL and RESFRM bits in the SYSFLG2 register.

3.16 DAI Registers

There are five registers within the DAI Interface; one control, three data, and one status register. The control register is used to mask or unmask interrupt requests to service the DAI's FIFOs, and to select whether an on-chip or off-chip clock is used to drive the bit rate, and to enable / disable operation. The first pair of data register addresses the top of the Right Channel Transmit FIFO and the bottom of the Right Channel Receive FIFO. A read accesses the receive FIFOs, and a write the transmit FIFOs. Note that these are four physically separate FIFOs to allow full-duplex transmission. The status register contains bits which signal FIFO overrun and underrun errors and transmit and receive FIFO service requests. Each of these status conditions signal an interrupt request to the interrupt controller. The status register also flags when the transmit FIFOs are not full when the receive FIFOs are not empty.

3.16.1 DAI Control Register (address 0x8000.2000)

31-24	23	22	21	20	19	18	17	16	15-0
Reserved	Reserved	RCRM	RCTM	LCRM	LCTM	Reserved	ECS	DAIEN	Reserved

The DAI control register (DAIR) contains eight different bit fields that control various functions within the DAI interface.

Bit	Description
0-15	Reserved Must be set to 0x0404
7	Reserved
15	Reserved
16	**DAIEN:** DAI Interface Enable 0 — DAI operation disabled, control of the SDIN, SDOUT, SCLKLRCK, and LRCK pins given to the SSI2 / CODEC / DAI pin mulitiplexing logic to assign I/O pins 60-64 to another block. 1 — DAI operation enabled Note that by default, the SSI / CODEC have precedence over the DAI interface in regard to the use of the I/O pins. Nevertheless, when bit 3 (DAISEL) of register SYSCON3 is set to 1, then the above mentioned DAI ports are connected to I/O pins 60–64.
17	**ECS:** External Clock Select selects external MCLK when = 1.
18	ReservedMust be 0.
19	**LCTM:** Left Channel Transmit FIFO Interrupt Mask 0 — Left Channel Transmit FIFO half-full or less condition does not generate an interrupt (LCTS bit ignored). 1 — Left Channel Transmit FIFO half-full or less condition generates an interrupt (state of LCTS sent to interrupt controller).
20	**LCRM:** Left Channel Receive FIFO Interrupt Mask 0 — Left Channel Receive FIFO half-full or more condition does not generate an interrupt (LCRS bit ignored). 1 — Left Channel Receive FIFO half-full or more condition generates an interrupt (state of LCRS sent to interrupt controller).
21	**RCTM:** Right Channel Transmit FIFO Interrupt Mask 0 — Right Channel Transmit FIFO half-full or less condition does not generate an interrupt (RCTS bit ignored). 1 — Right Channel Transmit FIFO half-full or less condition generates an interrupt (state of RCTS sent to interrupt controller).
22	**RCRM:** Right Channel Receive FIFO Interrupt Mask 0 — Right Channel Receive FIFO half-full or more condition does not generate an interrupt (RCRS bit ignored). 1 — Right Channel Receive FIFO half-full or more condition generates an interrupt (state of RCRS sent to interrupt controller).
23	Reserved
24-31	Reserved

Table 58. DAI Control Register

3.16.1.1 DAI Enable (DAIEN)

The DAI enable (DAIEN) bit is used to enable and disable all DAI operation.

When the DAI is disabled, all of its clocks are powered down to minimize power consumption. Note that DAIEN is the only control bit within the DAI interface that is reset to a known state. It is cleared to zero to ensure the DAI timing is disabled following a reset of the device.

When the DAI timing is enabled, SCLK begins to transition and the start of the first frame is signaled by driving the LRCK pin low. The rising and falling-edge of LRCK coincides with the rising and falling-edge of SCLK. As long as the DAIEN bit is set, the DAI interface operates continuously, transmitting and receiving 128 bit data frames. When the DAIEN bit is cleared, the DAI interface is disabled immediately, causing the current frame which is being transmitted to be terminated. Clearing DAIEN resets the DAI's interface FIFOs. However DAI data register 3, the control register and the status register are not reset. Therefore, the user must ensure these registers are properly reconfigured before re-enabling the DAI interface.

3.16.1.2 DAI Interrupt Generation

The DAI interface can generate four maskable interrupts and four non-maskable interrupts, as described in the sections below. Only one interrupt line is wired into the interrupt controller for the whole DAI interface. This interrupt is the wired OR of all eight interrupts (after masking where appropriate). The software servicing the interrupts must read the status register in the DAI to determine which source(s) caused the interrupt. It is possible to prevent any DAI sources causing an interrupt by masking the DAI interrupt in the interrupt controller register.

3.16.1.3 Left Channel Transmit FIFO Interrupt Mask (LCTM)

The Left channel sample transmit FIFO interrupt mask (LCTM) bit is used to mask or enable the left channel sample transmit FIFO service request interrupt. When LATM = 0, the interrupt is masked and the state of the Left Channel Transmit FIFO service request (LCTS) bit within the DAI status register is ignored by the interrupt controller. When LCTM = 1, the interrupt is enabled and whenever LCTS is set (one) an interrupt request is made to the interrupt controller. Note that programming LCTM = 0 does not affect the current state of LCTS or the Left Channel Transmit FIFO logic's ability to set and clear LCTS; it only blocks the generation of the interrupt request.

3.16.1.4 Left Channel Receive FIFO Interrupt Mask (LARM)

The left channel sample receive FIFO interrupt mask (LCRM) bit is used to mask or enable the Left Channel Receive FIFO service request interrupt. When LCRM = 0, the interrupt is masked and the state of the left channel sample receive FIFO service request (LCRS) bit within the DAI status register is ignored by the interrupt controller. When LCRM = 1, the interrupt is enabled and whenever LCRS is set (one) an interrupt request is made to the interrupt controller. Note that programming LCRM = 0 does not affect the current state of LCRS or the Left Channel Receive FIFO logic's ability to set and clear LCRS, it only blocks the generation of the interrupt request.

3.16.1.5 Right Channel Transmit FIFO Interrupt Mask (RCTM)

The Right Channel Transmit FIFO interrupt mask (RCTM) bit is used to mask or enable the right channel transmit FIFO service request interrupt. When RCTM = 0, the interrupt is masked and the state of the Right Channel Transmit FIFO service request (RCTS) bit within the DAI status register is ignored by the interrupt controller. When RCTM = 1, the interrupt is enabled and whenever RCTS is set (one) an interrupt request is made to the interrupt controller. Note that programming RCTM = 0 does not affect the current state of RCTS or the Right Channel Transmit FIFO logic's ability to set and clear RCTS, for it only blocks the generation of the interrupt request.

3.16.1.6 Right Channel Receive FIFO Interrupt Mask (RCRM)

The Right Channel Receive FIFO interrupt mask (RCRM) bit is used to mask or enable the Right Channel Receive FIFO service request interrupt. When RCRM = 0, the interrupt is masked and the state of the Right Channel Receive FIFO service request (RCRS) bit within the DAI status register is ignored by the interrupt controller. When RCRM = 1, the interrupt is enabled, and whenever RCRS is set (one), an interrupt request is made to the interrupt controller. Note that programming RCRM = 0 does not affect the current state of RCRS or the Right Channel Receive FIFO logic's ability to set and clear RCRS, for it only blocks the generation of the interrupt request.

3.16.2 DAI Data Registers

The DAI contains three data registers: DAIDR0 addresses the top entry of the Right Channel Transmit FIFO and bottom entry of the Right Channel Receive FIFO; DAIDR1 addresses the top and bottom entry of the Left Channel Transmit and Receive FIFOs, respectively; and DAIDR2 is used to perform enable and disable the DAI FIFOs.

3.16.3 DAIDR0 DAI Data Register 0 (address 0x8000.2040)

31-16	15-0
Reserved	Bottom of Right Channel Receive FIFO

Read Access

31-16	15-0
Reserved	Top of Right Channel Transmit FIFO

Write Access

When DAI Data Register 0 (DAIDR0) is read, the bottom entry of the Right Channel Receive FIFO is accessed. As data is removed by the DAI's receive logic from the incoming data frame, it is placed into the top entry of the Right Channel Receive FIFO and is transferred down an entry at a time until it reaches the last empty location within the FIFO. Data is removed by reading DAIDR0, which accesses the bottom entry of the right channel FIFO. After DAIDR0 is read, the bottom entry is invalidated, and all remaining values within the FIFO automatically transfer down one location.

When DAIDR0 is written, the top-most entry of the Right Channel Transmit FIFO is accessed. After a write, data is automatically transferred down to the lowest location within the transmit FIFO which does not already contain valid data. Data is removed from the bottom of the FIFO one value at a time by the transmit logic, loaded into the correct position within the 64-bit transmit serial shifter, then serially shifted out onto the SDOUT pin.

Table 59 shows DAIDR0. Note that the Transmit and Receive Right Channel FIFOs are cleared when the device is reset, or by writing a zero to DAIEN (DAI disabled). Also, note that writes to reserved bits are ignored and reads return zeros.

Bit	Description
0-15	**RIGHT CHANNEL DATA:** Transmit / Receive Right Channel FIFO Data Read — Bottom of Right Channel Receive FIFO data Write — Top of Right Channel Transmit FIFO data

Bit	Description
16-31	Reserved

Table 59. DAI Data Register 0 (Continued)

3.16.4 DAIDR1 DAI Data Register 1 (address 0x8000.2080)

31-16	15-0
Reserved	Bottom of Left Channel Receive FIFO

Read Access

31-16	15-0
Reserved	Top of Left Channel Transmit FIFO

Write Access

When DAI Data Register 1 (DAIDR1) is read, the bottom entry of the Left Channel Receive FIFO is accessed. As data is removed by the DAI's receive logic from the incoming data frame, it is placed into the top entry of the Left Channel Receive FIFO and is transferred down an entry at a time until it reaches the last empty location within the FIFO. Data is removed by reading DAIDR1, which accesses the bottom entry of the left channel FIFO. After DAIDR1 is read, the bottom entry is invalidated, and all remaining values within the FIFO automatically transfer down one location.

When DAIDR1 is written, the top-most entry of the Left Channel Transmit FIFO is accessed. After a write, data is automatically transferred down to the lowest location within the transmit FIFO which does not already contain valid data. Data is removed from the bottom of the FIFO one value at a time by the transmit logic. It is then loaded into the correct position within the 64-bit transmit serial shifter then serially shifted out onto the SDOUT pin.

Table 60 shows DAIDR1. Note that the Transmit and Receive Left Channel FIFOs are cleared when the device is reset, or by writing a zero to DAIEN (DAI disabled). Also, note that writes to reserved bits are ignored and reads return zeros.

Bit	Description
0-15	**LEFT CHANNEL DATA:** Transmit / Receive Left Channel FIFO Data Read — Bottom of Left Channel Receive FIFO data Write — Top of Left Channel Transmit FIFO data
16-31	Reserved

Table 60. DAI Data Register 1

3.16.5 DAIDR2 DAI Data Register 2 (address 0x8000.20C0)

31-21	20-16	15	14-0
Reserved	FIFO Channel Select	FIFOEN	Reserved

DAIDR2 is a 32-bit register that utilizes 21 bits and is used to enable and disable the FIFOs for the left and right channels of the DAI data stream. The left channel FIFO is enabled by writing 0x000D.8000 and disabled by writing 0x000D.0000. The right channel FIFO is enabled by writing 0x0011.8000 and disabled by writing 0x0011.0000. After writing a value to this register, wait until the

FIFO operation complete bit (FIFO) is set in the DAI status register before writing another value to this register.

Bit	Description
0-14	Reserved
15	**FIFOEN:** FIFO Transmit Bit 0 — Disable Transmit 1 — Enable Transmit
16-20	FIFO CHANNEL SELECT: 01101b — Left channel select 10001b — Right channel select
21-31	Reserved

Table 61. DAI Data Register 2

3.16.6 DAISR DAI *Status (address 0x8000.2100)*

The DAI Status register (DAISR) contains bits which signal FIFO overrun and underrun errors and FIFO service requests. Each of these conditions signal an interrupt request to the interrupt controller. The status register also flags when transmit FIFOs are not full, when the receive FIFOs are not empty, when a FIFO operation is complete, and when the right channel or left channel portion of the CODEC is enabled (no interrupt generated).

Bits which cause an interrupt signal the interrupt request as long as the bit is set. Once the bit is cleared, the interrupt is cleared. Read / write bits are called status bits, read-only bits are called flags. Status bits are referred to as “sticky” (once set by hardware, they must be cleared by software). Writing a one to a sticky status bit clears it, while writing a zero has no effect. Read-only flags are set and cleared by hardware, and writes have no effect. Additionally, some bits which cause interrupts have corresponding mask bits in the control register and are indicated in the section headings below. Note that the user has the ability to mask all DAI interrupts by clearing the DAI bit within the interrupt controller mask register INTMR3.

31-13	12	11	10	9	8	7
Reserved	FIFO	LCNE	LCNF	RCNE	RCNF	RCCELCRO
6	**5**	**4**	**3**	**2**	**1**	**0**
RCNFLCTU	LCRORCRO	LCTURCTU	LCRS	LCTS	LCRSRCRS	LCTSRCTS

Bit	Description
0	**RCTS:** Right Channel Transmit FIFO Service Request Flag (read-only) 0 — Right Channel Transmit FIFO is more than half full (five or more entries filled) or DAI disabled 1 — Right Channel Transmit FIFO is half full or less (four or fewer entries filled) and DAI operation is enabled, interrupt request signaled if not masked (if RCTM = 1)

Table 62. DAI Control, Data and Status Register Locations

Bit	Description
1	**RCRS:** Right Channel Receive FIFO Service Request (read-only) 0 — Right Channel Receive FIFO is less than half full (five or fewer entries filled) or DAI disabled 1 — Right Channel Receive FIFO is half full or more (six or more entries filled) and DAI operation is enabled, interrupt request signaled if not masked (if RCRM = 1)
2	**LCTS:** Left Channel Transmit FIFO Service Request Flag (read-only) 0 — Left Channel Transmit FIFO is more than half full or less (four or fewer entries filled) or DAI disabled. 1 — Left Channel Transmit FIFO is half full or less (four or fewer entries filled) and DAI operation is enabled, interrupt request signaled if not masked (if LCTM = 1)
3	**LCRS:** 0 — Left Channel Receive FIFO is less than half full (five or fewer entries filled) or DAI disabled. 1 — Left Channel Receive FIFO is half full or more (six or more entries filled) and DAI operation is enabled, interrupt request signalled if not masked (if LCRM = 1)
4	Right Channel Transmit FIFO Underrun 0 — Right Channel Transmit FIFO has not experienced an underrun 1 — Right Channel Transmit logic attempted to fetch data from transmit FIFO while it was empty, request interrupt
5	**RCRO:** Right Channel Receive FIFO Overrun 0 — Right Channel Receive FIFO has not experienced an overrun 1 — Right Channel Receive logic attempted to place data into receive FIFO while it was full, request interrupt
6	**LCTU:** Left Channel Transmit FIFO Underrun 0 — Left Channel Transmit FIFO has not experienced an underrun 1 — Left Channel Transmit logic attempted to fetch data from transmit FIFO while it was empty, request interrupt
7	**LCRO:** Left Channel Receive FIFO Overrun 0 — Left Channel Receive FIFO has not experienced an overrun 1 — Left Channel Receive logic attempted to place data into receive FIFO while it was full, request interrupt
8	**RCNF:** Right Channel Transmit FIFO Not Full (read-only) 0 — Right Channel Transmit FIFO is full 1 — Right Channel Transmit FIFO is not full
9	**RCNE:** Right Channel Receive FIFO Not Empty (read-only) 0 — Right Channel Receive FIFO is empty 1 — Right Channel Receive FIFO is not empty
10	**LCNF:** LCNETelecom Transmit FIFO Not Full (read-only) 0 — Left Channel Transmit FIFO is full 1 — Left Channel Transmit FIFO is not full

Table 62. DAI Control, Data and Status Register Locations (Continued)

Bit	Description
11	**LCNE:** Left Channel Receive FIFO Not Empty (read-only) 0 — Left Channel Receive FIFO is empty 1 — Left Channel Receive FIFO is not empty
12	**FIFO:** FIFO Operation Completed (read-only)0 — A FIFO Operation has not completed since the last time this bit was cleared1 — THe FIFO Operation was completed
13	Reserved
14	Reserved
15	Reserved
16-31	Reserved

Table 62. DAI Control, Data and Status Register Locations (Continued)

3.16.6.1 Right Channel Transmit FIFO Service Request Flag (RCTS)

The Right Channel Transmit FIFO Service Request Flag (RCTS) is a read-only bit which is set when the Right Channel Transmit FIFO is nearly empty and requires service to prevent an underrun. RCTS is set any time the Right Channel Transmit FIFO has four or fewer entries of valid data (half full or less), and is cleared when it has five or more entries of valid data. When the RCTS bit is set, an interrupt request is made unless the Right Channel Transmit FIFO interrupt request mask (RCTM) bit is cleared. After the CPU fills the FIFO such that four or more locations are filled within the Right Channel Transmit FIFO, the RCTS flag (and the service request and / or interrupt) is automatically cleared.

3.16.6.2 Right Channel Receive FIFO Service Request Flag (RCRS)

The Right Channel Receive FIFO Service Request Flag (RCRS) is a read-only bit which is set when the Right Channel Receive FIFO is nearly filled and requires service to prevent an overrun. RCRS is set any time the Right Channel Receive FIFO has six or more entries of valid data (half full or more), and cleared when it has five or fewer (less than half full) entries of data. When the RCRS bit is set, an interrupt request is made unless the Right Channel Receive FIFO interrupt request mask (RCRM) bit is cleared. After six or more entries are removed from the receive FIFO, the LCRS flag (and the service request and / or interrupt) is automatically cleared.

3.16.6.3 Left Channel Transmit FIFO Service Request Flag (LCTS)

The Left Channel Transmit FIFO Service Request Flag (LCTS) is a read-only bit which is set when the Left Channel Transmit FIFO is nearly empty and requires service to prevent an underrun. LCTS is set any time the Left Channel Transmit FIFO has four or fewer entries of valid data (half full or less). It is cleared when it has five or more entries of valid data. When the LCTS bit is set, an interrupt request is made unless the Left Channel Transmit FIFO interrupt request mask (LCTM) bit is cleared. After the CPU fills the FIFO such that four or more locations are filled within the Left Channel Transmit FIFO, the LCTS flag (and the service request and / or interrupt) is automatically cleared.

3.16.6.4 Left Channel Receive FIFO Service Request Flag (LCRS)

The Left Channel Receive FIFO Service Request Flag (LCRS) is a read-only bit which is set when the Left Channel Receive FIFO is nearly filled and requires service to prevent an overrun. LCRS is set any time the Left Channel Receive FIFO has six or more entries of valid data (half full or more), and cleared when it has five or fewer (less than half full) entries of data. When the LCRS bit is set, an interrupt request is made unless the Left Channel Receive FIFO interrupt request mask (LCRM) bit is cleared. After six or more entries are removed from the receive FIFO, the LCRS flag (and the ser-

vice request and / or interrupt) is automatically cleared.

3.16.6.5 Right Channel Transmit FIFO Underrun Status (RCTU)

This is set when the Right Channel Transmit logic attempts to fetch data from the FIFO after it has been emptied. When an underrun occurs, the Right Channel Transmit logic continuously transmits the last valid right channel value which was transmitted before the underrun. Once data is placed in the FIFO and it is transferred down to the bottom, the Right Channel Transmit logic uses the new value within the FIFO for transmission. When the RCTU bit is set, an interrupt request is made.

3.16.6.6 Right Channel Receive FIFO Overrun Status (RCRO)

This is set when the right channel receive logic attempts to place data into the Right Channel Receive FIFO after it has been completely filled. Each time a new piece of data is received, the set signal to the RCRO status bit is asserted, and the newly received data is discarded. This process is repeated for each new sample received until at least one empty FIFO entry exists. When this bit is set, an interrupt request is made.

3.16.6.7 Left Channel Transmit FIFO Underrun Status (LCTU)

The Left Channel Transmit FIFO Underrun Status Bit (LCTU) is set when the Left Channel Transmit logic attempts to fetch data from the FIFO after it has been completely emptied. When an underrun occurs, the Left Channel Transmit logic continuously transmits the last valid left channel value which was transmitted before the underrun occurred. Once data is placed in the FIFO and it is transferred down to the bottom, the Left Channel Transmit logic uses the new value within the FIFO for transmission. When the LCTU bit is set, an interrupt request is made.

3.16.6.8 Left Channel Receive FIFO Overrun Status (LCRO)

The Left Channel Receive FIFO Overrun Status Bit (LCRO) is set when the Left Channel Receive logic places data into the Left Channel Receive FIFO after it has been completely filled. Each time a new piece of data is received, the set signal to the LCRO status bit is asserted, and the newly received sample is discarded. This process is repeated for each new piece of data received until at least one empty FIFO entry exists. When the LCRO bit is set, an interrupt request is made.

3.16.6.9 Right Channel Transmit FIFO Not Full Flag (RCNF)

The Right Channel Transmit FIFO Not Full Flag (RCNF) is a read-only bit which is set whenever the Right Channel Transmit FIFO contains one or more entries which do not contain valid data and is cleared when the FIFO is completely full. This bit can be polled when using programmed I/O to fill the Right Channel Transmit FIFO. This bit does not request an interrupt.

3.16.6.10 Right Channel Receive FIFO Not Empty Flag (RCNE)

The Right Channel Receive FIFO Not Empty Flag (RCNELCNF) is a read-only bit which is set when ever the Right Channel Receive FIFO contains one or more entries of valid data and is cleared when it no longer contains any valid data. This bit can be polled when using programmed I/O to remove remaining data from the receive FIFO. This bit does not request an interrupt.

3.16.6.11 Left Channel Transmit FIFO Not Full Flag (LCNF)

This is a read-only bit which is set when ever the Left Channel Transmit FIFO contains one or more entries which do not contain valid data. It is cleared when the FIFO is full. This bit can be polled when using programmed I/O to fill the Left Channel Transmit FIFO. This bit does not request an interrupt.

3.16.6.12 Left Channel Receive FIFO Not Empty Flag (LCNE)

This is a read-only bit set when the Left Channel Receive FIFO contains one or more entries of valid

data and is cleared when it no longer contains any valid data. This bit can be polled when using programmed I/O to remove remaining data from the receive FIFO. This bit does not request an interrupt.

3.16.6.13 FIFO Operation Completed Flag (FIFO)

The FIFO Operation Completed (FIFO) Flag is set after the FIFO operation requested by writing to DAIDR2 as completed.

FIFO is automatically cleared when DAIDR2 is read or written. This bit does not request an interrupt.

3.16.7 DAI64Fs Control (address 0x8000.2600)

The DAI now includes a divider network for the frequency of the clock source. The CS89712 provides for both 128 and 64 times the sample frequency (128 fs and 64 fs) to better support the various MP3 sample rates. There are two clocks to choose from: 73.728 MHz PLL clock as well as the 11.2896 Mhz external clock. The purpose is support more devices that use the 64 fs rate. Both clocks are fixed rate clocks so the divider network (AUDIV) is required.

31-15	14-8	7-6	5	4	3	2	1	0
	AUDDIV		LOOPBACK		MCLK256EN	AUDCLKSRC	AUDIOCLKEN	I2SFS64

Bit	Description
0	**I2SF64: 0** => 128 fs **1**=>64 fs If high, SYSCON3 bit 9 must be low. The converse is also true.
1	**AUDCLKEN:** Enable audio clock generator
2	**AUDCLKSRC:** ClockSource **0**=>73.728MHz**(PLL)** **1**=>11.2896MHz**(ExtClock)**
3	**MCLK256EN:** Selects MCLK (256 fs) or the BUZZ pin
4	**Reserved**
5	**LOOPBACK:** Test mode. Digital data normally output to DAC loops internally.
6-7	**Reserved**
8-14	**AUDIV:** Frequency divisor for sample frequency and bit clock using either the external clock or the PLL clock for the audio clock generator
15-31	**Reserved**

Table 63. DAI64Fs Control Register

Clock Source (MHz)	Sample Frequency (KHz)	128 fs Audio Bit Clock (MHz)	64 fs Audio Bit Clock (MHz)	128 fs Divisor (AUDDIV)	64 fs Divisor (AUDDIV)
73.728	8	1.0240	0.5120	36	72
11.2896	11.025	1.4112	0.7056	8	16
73.728	16	1.5360	0.7680	18	36
11.2896	22.050	2.8224	1.4112	4	8
73.728	24	3.0720	1.5360	12	24
73.728	32	4.0960	2.0480	9	18
11.2896	44.1	5.6448	2.8224	2	4
73.728	48	6.1440	3.0720	6	12

Table 64. Clock Source for 64 fs and 128 fs

3.17 Ethernet Bus Interface Registers

3.17.1 Master Interrupt Enable (address offset 0022h)

Address 0023h	Address 0022h
00h	Interrupt number assignment: 0000 0000b = Enable 0000 01xxb = Disable

Reset value is: XXXX XXXX XXXX X100 - Interrupts disabled

3.17.2 EEPROM Command (address offset 0040h)

7:0	F:B	A	9	8
ADD7 to ADD0	Reserved	Reserved	OB1	OB0

This register is used to control the reading, writing and erasing of the EEPROM. See Section 2.24, *"Programming the EEPROM"*.

Bit	Description
7:0	**ADD7-ADD0:** Address of the EEPROM word being accessed.
9:8	**OB1,OB0:** Indicates the Opcode of the command being executed. See Table 26.
A	**Reserved:** Reserved and must be written as 0.
F:B	**Reserved:** Reserved and must be written as 0.

Table 65. EEPROM Command Bits

Reset value is: XXXX XXXX XXXX XXXX

3.17.3 EEPROM Data (address offset 0042h)

Address 0043h	Address 0042h
Most significant byte of the EEPROM data.	Least significant byte of the EEPROM data.

This register contains the word being written to, or read from, the EEPROM. See Section 2.24, *"Programming the EEPROM"*.

Reset value is: XXXX XXXX XXXX XXXX

3.17.4 Receive Frame Byte Counter (address offset 0050h)

Address 0051h	Address 0050h
Most significant byte of the byte count.	Least significant byte of the byte count.

This register contains the count of the total number bytes received in the current received frame. This count continuously increments as more bytes in this frame are received. See Section 2.32, *"Basic Receive Operation"*.

Reset value is: XXXX XXXX XXXX XXXX

3.18 Ethernet Port Status/Control Registers

3.18.1 Interrupt Status Queue Register (ISQ, address 120h)

7:6	5:0	F:8
RegContent	RegNum	RegContent

The Interrupt Status Queue Register is used to provide interrupt information. Whenever an event occurs that triggers an enabled interrupt, the Ethernet Port sets the appropriate bit(s) in one of five registers, maps the contents of that register to the ISQ register, and drives an IRQ pin high. Three of the registers mapped to ISQ are event registers:

RxEvent, TxEvent, and BufEvent. The other two registers are counter-overflow reports: RxMISS and TxCOL. In addition, ISQ is located at offset address 0008h. See Section 2.31, *"Managing Interrupts & Status Queue"*.

Bit	Description
5:0	**RegNum:** The lower six bits describe which register (4, 8, C, 10 or 12) is contained in the ISQ.
7:6 F:8	**RegContent:** The upper ten bits contain the register data contents.

Table 66. Interrupt Status Queue

Reset value is: 0000 0000 0000 0000

3.18.2 Receiver Configuration Register (RxCFG, address offset 102h)

7	6	5:0	F	E	D
RSVD	Skip_1	000011		ExtradataiE	RuntiE

C	B	A	9	8
CRCerroriE	BufferCRC	RSVD	RSVD	RxOKiE

RxCFG determines what frame types will cause interrupts.

Bit	Description
5:0	**000011:** These bits provide an internal address used by the CS89712 to identify this as register 3, the Receiver Configuration Register.
6	**Skip_1:** When set, this bit causes the last committed received frame to be deleted from the receive buffer. To skip another frame, this bit must be set again. This bit is not to be used if RxDMAonly (Bit 9) is set. Skip_1 is an Act-Once bit. See Section 2.32.4, *"Buffering Held Receive Frames"*.
8	**RxOKiE:** When set, there is an RxOK Interrupt if a frame is received without errors. RxOK interrupt is not generated when DMA mode is used for frame reception.
A	**RSVD:** Reserved - must be a "0" when writing to this register.
B	**BufferCRC:** When set, the received CRC is included with the data stored in the receive-frame buffer, and the four CRC bytes are included in the receive-frame length (Ethernet Port offset address 0402h). When clear, neither the receive buffer nor receive length include the CRC.
C	**CRCerroriE:** When set, there is a CRCerror Interrupt if a frame is received with a bad CRC.
D	**RuntiE:** When set, ther is a Runt Interrupt if a frame is received that is shorter than 64 bytes. The CS89712 always discards any frame that is shorter than 8 bytes.
E	**ExtradataiE:** When set, there is an Extradata Interrupt if a frame is received that is longer than 1518 bytes. The operation of this bit is independent of the received packet integrity (good or bad CRC).

Table 67. Receiver Configuration

After reset, if no EEPROM is found by the CS89712, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 2.24, *"Programming the EEPROM"*.

Reset value is: 0000 0000 0000 0011

3.18.3 Receiver Event Register, (RxEvent, address offset 124h)

7	6	5:0	F	E	D
Dribblebits	IAHash	000100		Extradata	Runt

C	B	A	9	8
CRCerror	Broadcast	Individual Adr	Hashed	RxOK

Alternate meaning if bits 8 and 9 are both set (see Section 2.32.7, *"Receive Ethernet Port Locations"* for exception regarding Broadcast frames).

7	6	5:0	F:A	9	8
Dribblebits	IAHash	000100	Hash Table Index (see Section 2.32.7, *"Receive Ethernet Port Locations"*)	Hashed = 1	RxOK = 1

RxEvent reports the status of the current received frame.

Bit	Description
5:0	**000100:** These bits identify this as register 4, the Receiver Event Register. When reading this register, these bits will be 000100, where the LSB corresponds to Bit 0.
6	**IAHash:** If the received frame's Destination Address is accepted by the hash filter, then this bit is set if, and only if IAHashA (Register 5, RxCTL, Bit 6) is set, and Hashed (Bit 9) is set. See Section 2.32.7, *"Receive Ethernet Port Locations"*.
7	**Dribblebits:** If set, the received frame had from one to seven bits after the last received full byte. An "Alignment Error" occurs when Dribblebits and CRCerror (Bit C) are both set.
8	**RxOK:** If set, the received frame had a good CRC and valid length (i.e., there is not a CRC error, Runt error, or Extradata error). When RxOK is set, then the length of the received frame is contained at Ethernet Port offset address 0402h. If RxOKiE (Register 3, RxCFG, Bit 8) is set, there is an interrupt.
9	**Hashed:** If set, the received frame had a Destination Address that was accepted by the hash filter. If Hashed and RxOK (Bit 8) are set, Bits F through A of RxEvent become the Hash Table Index for this frame [See Section 2.32.7, *"Receive Ethernet Port Locations"* for an exception regarding broadcast frames!].If Hashed and RxOK are not both set, then Bits F through A are individual event bits as defined below.
A	**IndividualAdr:** If the received frame had a Destination Address which matched the Individual Address found at Ethernet Port offset address 0158h, then this bit is set if, and only if, RxOK (Bit 8) is set and IndividualA (Register 5, RxCTL, Bit A) is set.
B	**Broadcast:** If the received frame had a Broadcast Address (FFFF FFFF FFFFh) as the Destination Address, then this bit is set if, and only if, RxOK is set and BroadcastA (RxCTL register bit B) is set.
C	**CRCerror:** If set, the received frame had a bad CRC. If CRCerroriE (Register 3, RxCFG, Bit C) is set, there is an interrupt.
D	**Runt:** If set, the received frame was shorter than 64 bytes. If RuntiE (Register 3, RxCFG, Bit D) is set, there is an interrupt.
E	**Extradata:** If set, the received frame was longer than 1518 bytes. All bytes beyond 1518 are discarded. If ExtradataiE (Register 3, RxCFG, Bit E) is set, there is an interrupt.

Table 68. Receiver Event

Reset value is: 0000 0000 0000 0100

Notes: 1. All RxEvent bits are cleared upon readout. The software is responsible for processing all event bits.

2. RxStatus register (Ethernet Port offset address 0400h) is the same as the RxEvent register except RxStatus is not cleared when RxEvent is read. See Section 2.32, *"Basic Receive Operation"*. The value in the RxEvent register is undefined when RxDMAOnly bit (Bit 9, Register 3, RxCFG) is set.

3.18.4 Receiver Control Register (RxCTL, address offset 104h)

7	6	5:0	F	E	D
PromiscuousA	IAHashA	000101		ExtradataA	RuntA

C	B	A	9	8
CRCerrorA	BroadcastA	IndividualA	MulticastA	RxOKA

RxCTL has two functions: Bits 8, C, D, and E define what types of frames to accept. Bits 6, 7, 9, A, and B configure the Destination Address filter. See Section 2.32.7, *"Receive Ethernet Port Locations"*.

Bit	Description
5:0	**000101:** These bits provide an internal address used by the CS89712 to identify this as register 5, the Receiver Control Register. For a received frame to be accepted, the Destination Address of that frame must pass the filter criteria found in Bits 6, 7, 9, A, and B (see Section 2.32.7, *"Receive Ethernet Port Locations"*).
6	**IAHashA:** When set, receive frames are accepted when the Destination Address is an Individual Address that passes the hash filter.
7	**PromiscuousA:** Frames with any address are accepted when this bit is set.
8	**RxOKA:** When set, the CS89712 accepts frames with correct CRC and valid length (valid length is: 64 bytes <= length <= 1518 bytes).
9	**MulticastA:** When set, receive frames are accepted if the Destination Address is an Multicast Address that passes the hash filter.
A	**IndividualA:** When set, receive frames are accepted if the Destination Address matches the Individual Address found at Ethernet Port offset address 0158h to 015Dh.
B	**BroadcastA:** When set, receive frames are accepted if the Destination Address is FFFF FFFF FFFFh.
C	**CRCerrorA:** When set, receive frames that pass the Destination Address filter, but have a bad CRC, are accepted. When clear, frames with bad CRC are discarded. See Note 1.
D	**RuntA:**When set, receive frames that are smaller than 64 bytes, and that pass the Destination Address filter are accepted. When clear, received frames less that 64 bytes in length are discarded. The CS89712 discards any frame that is less than 8 bytes. See Note 1.
E	**ExtradataA:** When set, receive frames longer than 1518 bytes and that pass the Destination Address filter are accepted. The CS89712 accepts only the first 1518 bytes and ignores the rest. When clear, frames longer than 1518 bytes are discarded. See Note 1.

Table 69. Receiver Control

After reset, if no EEPROM is found by the CS89712, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 2.32.7, *"Receive Ethernet Port Locations"*.

Reset value is: 0000 0000 0000 0101

Notes: 1. Typically, when bits CRCerrorA, RuntA and ExtradataA are cleared (meaning bad frames are being discarded), then the corresponding bits CRCerroriE, RuntiE and ExtradataiE should be set in register 3 (Receiver Configuration register) to allow the device driver to keep track of discarded frames.

3.18.5 Transmit Configuration Register (TxCFG, address offset 106h)

7:6	5:0	F:C	B	A	9	8
RSVD	000111	16colliE	AnycolliE	JabberiE	Out-of-window	TxOKiE

Each bit in TxCFG is an interrupt enable. When set, the interrupt is enabled as described below. When clear, there is no interrupt.

Bit	Description
5:0	**000111:** These bits provide an internal address used by the CS89712 to identify this as register 7, the Transmit Configuration Register.
8	**TxOKiE:** When set, an interrupt is generated if a packet is completely transmitted.
9	**Out-of-windowiE:** When set, an interrupt is generated if a late collision occurs (a late collision is a collision which occurs after the first 512 bit times). When this occurs, the CS89712 forces a bad CRC and terminates the transmission.
A	**JabberiE:** When set, an interrupt is generated if a transmission is longer than approximately 26 ms.
B	**AnycolliE:** When set, if one or more collisions occur during the transmission of a packet, an interrupt occurs at the end of the transmission
F:C	**16colliE**: If the CS89712 encounters 16 normal collisions while attempting to transmit a particular packet, the CS89712 stops attempting to transmit that packet. When this bit is set, there is an interrupt upon detecting the 16th collision.
7:6	**RSVD:** Reserved. must be a “0” when writing to this register.

Table 70. Transmit Configuration

After reset, if no EEPROM is found by the CS89712, then the register has the following initial state. If an EEPROM is found, then the register’s initial value may be set by the EEPROM. See Section 2.24, *“Programming the EEPROM”*.

Reset value is: 0000 0000 0000 0111

Note: Bit 8 (TxOKiE) and Bit B (AnycolliE) are interrupts for normal transmit operation. Bits 6, 7, 9, A, and Fare interrupts for abnormal transmit operation.

3.18.6 Transmitter Event Register (TxEvent, address offset 128h)

7:6	5:0	F	E:B	A	9	8
RSVD	001000	16coll	Number-of-Tx-collisions	Jabber	Out-of-window	TxOK

TxEvent gives the event status of the last packet transmitted.

Bit	Description
5:0	**001000:** These bits provide an internal address used by the CS89712 to identify this as register 8, the Transmitter Event Register.
7:6	**RSVD:** Reserved. must be a “0” when writing to this register.
8	**TxOK:** This bit is set if the last packet was completely transmitted (Jabber (Bit A), out-of-window-collision (Bit 9), and 16Coll (Bit F) must all be clear). If TxOKiE (Register 7, TxCFG, Bit 8) is set, there is an interrupt.
9	**Out-of-Window:** This bit is set if a collision occurs more than 512 bit times after the first bit of the preamble. When this occurs, the CS89712 forces a bad CRC and terminates the transmission. If Out-of-windowiE (Register 7, TxCFG, Bit 9) is set, there is an interrupt
A	**Jabber:** If the last transmission is longer than 26 msec, then the packet output is terminated by the jabber logic and this bit is set. If JabberiE (Register 7, TxCFG, Bit A) is set, there is an interrupt.
E:B	**#-of-TX-collisions:** These bits give the number of transmit collisions that occurred on the last transmitted packet. Bit B is the LSB. If AnycolliE (Register 7, TxCFG, Bit B) is set, there is an interrupt when any collision occurs.
F	**16coll:** This bit is set if the CS89712 encounters 16 normal collisions while attempting to transmit a particular packet. When this happens, the CS89712 stops further attempts to send that packet. If 16colliE (Register 7, TxCFG, Bit F) is set, there is an interrupt.

Table 71. Transmitter Event

Reset value is: 0000 0000 0000 1000

Notes: 1.In any event register, like TxEvent, all bits are cleared upon readout. The software is responsible for processing all event bits.
2.TxOK (Bit 8) and the Number-of-Tx-Collisions (Bits E-B) are used in normal packet transmission.All other bits (6, 7, 9, A, and F) give the status of abnormal transmit operation.

3.18.7 Transmit Command Status register (TxCMD, address offset 108h)

7:6	5:0	F	E	D
TxStart	001001			TxPadDis

C	B	A	9	8
InhibitCRC			Onecoll	Force

This register contains the latest transmit command which describes how the next packet should be sent. The command must be written to Ethernet Port offset 0144h in order to initiate a transmission. The software can read the command from this register (offset 0108h). See Section 2.34, *"Transmit Operation"*.

Bit	Description
5:0	**001001:** These bits provide an internal address used by the CS89712 to identify this as register 9, the Transmit Command Register. When reading this register, these bits will be 001001, where the LSB corresponds to Bit 0.
7:6	**TxStart:** This pair of bits determines how many bytes are transferred to the CS89712 before the MAC starts the packet transmit process. Bit 7 Bit 6 0 0 Start transmission after 5 bytes are in the CS89712 0 1 Start transmission after 381 bytes are in the CS89712 1 0 Start transmission after 1021 bytes are in the CS89712 1 1 Start transmission after the entire frame is in the CS89712
8	**Force:** When set in conjunction with a new transmit command, any transmit frames waiting in the transmit buffer are deleted. If a previous packet has started transmission, that packet is terminated within 64 bit times with a bad CRC.
9	**Onecoll:** When this bit is set, any transmission will be terminated after only one collision. When clear, the CS89712 allows up to 16 normal collisions before terminating the transmission.
C	**InhibitCRC:** When set, the CRC is not appended to the transmission.
D	**TxPadDis:** When TxPadDis is clear, if the software gives a transmit length less than 60 bytes and InhibitCRC is set, then the CS89712 pads to 60 bytes. If the software gives a transmit length less than 60 bytes and InhibitCRC is clear, then the CS89712 pads to 60 bytes and appends the CRC. When TxPadDis is set, the CS89712 allows the transmission of runt frames (a frame less than 64 bytes). If InhibitCRC is clear, the CS89712 appends the CRC. If InhibitCRC is set, the CS89712 does not append the CRC

Table 72. Transmit Command Status

After reset, if no EEPROM is found by the CS89712, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 2.24, *"Programming the EEPROM"*.

Regster value is: 0000 0000 0000 1001

Note: The CS89712 does not transmit a frame if TxLength < 3

3.18.8 Buffer Configuration Register (BufCFG, address offset 10Ah)

7	6	5:0	F	E	D
RSVD	SWint-X	001011	RxDestiE		Miss OvfloiE

C	B	A	9	8
TxCol OvfloiE	Rx128iE	RxMissiE	TxUnder runtiE	Rdy4TxiE

Each bit in BufCFG enables an interrupt. The corresponding interrupt is enabled when set, and disabled when clear.

Bit	Description
5:0	**001011:** Identify this as register B, the Buffer Configuration Register.
6	**SWint-X:** When set, there is an interrupt requested by the software. The Ethernet port provides the interrupt, and sets the SWint (Register C, BufEvent, Bit 6) bit. The Ethernet port acts upon this command at once. SWint-X is an Act-Once bit. To generate another interrupt, rewrite a "1" to this bit.
7	**RSVD:** Reserved; must be a “0” when writing to this register.
8	**Rdy4TxiE:** When set, there is an interrupt when the CS89712 is ready to accept a frame from the software for transmission. (See Section 2.34, *“Transmit Operation”* for a description of the transmit bid process.)
9	**TxUnderruniE:** When set, there is an interrupt if the CS89712 runs out of data before it reaches the end of the frame (called a transmit underrun). When this happens, event bit TXUnderrun (Register C, BufEvent, Bit 9) is set and the CS89712 makes no further attempts to transmit that frame. If the software still wants to transmit that particular frame, the software must go through the transmit request process again.
A	**RxMissiE:** When set, there is an interrupt if one or more received frames is lost due to slow movement of receive data out of the receive buffer (called a receive miss). When this happens, the RxMiss bit (Register C, BufEvent, Bit A) is set.
B	**Rx128iE:** When set, there is an interrupt after the first 128 bytes of a frame have been received. This allows software to examine the Destination Address, Source Address, Length, Sequence Number, and other information before the entire frame is received. This interrupt should not be used with DMA. Thus, if either AutoRxDMA (Register 3, RxCFG, Bit A) or RxDM-Aonly (Register 3, RxCFG, Bit 9) is set, the Rx128iE bit must be clear.
C	**TxColOvfiE:** If set, there is an interrupt when the TxCOL counter increments from 1FFh to 200h. (The TxCOL counter (Register 18) is incremented whenever the CS89712 sees that the RXD+/RXD- pins (10BASE-T) go active while a packet is being transmitted.)
D	**MissOvfloiE:** If MissOvfloiE is set, there is an interrupt when the RxMISS counter increments from 1FFh to 200h. (A receive miss is said to have occurred if packets are lost due to slow movement of receive data out of the receive buffers. When this happens, the RxMiss bit (Register C, BufEvent, Bit A) is set, and the RxMISS counter (Register 10) is incremented.)
F	**RxDestiE:** When set, there is an interrupt when a receive frame passes the Destination Address filter criteria defined in the RxCTL register (Register 5). This bit provides an early indication of an incoming frame. It is earlier than Rx128 (Register C, BufEvent, Bit B). If RxDestiE is set, the BufEvent could be RxDest or Rx128. After 128 bytes are received, the BufEvent changes from RxDest to Rx128.

Table 73. Buffer Configuration

After reset, if no EEPROM is found by the CS89712, then the register has the following initial state after reset. If an EEPROM is found, then the register’s initial value may be set by the EEPROM. See Section 2.24, *“Programming the EEPROM”*.

Reset value is: 0000 0000 0000 1011

3.18.9 Buffer Event Register (BufEvent, address offset 12Ch)

7	6	5:0	F	E	D
RSVD	SWint	001100	RxDest		

C	B	A	9	8
	Rx128	RxMiss	TxUnder run	Rdy4Tx

BufEvent gives the status of the transmit and receive buffers.

Bit	Description
5:0	**001100:** These bits provide an internal address used by the CS89712 to identify this as register C, the Buffer Event Register.
6	**SWint:** If set, there has been a software initiated interrupt. This bit is used in conjunction with the SWint-X bit (Register B, BufCFG, Bit 6).
7	**RSVD:** Reserved; must be a “0” when writing to this register.
8	**Rdy4Tx:** If set, the CS89712 is ready to accept a frame from the software for transmission. If Rdy4TxiE (Register B, BufCFG, Bit 8) is set, there is an interrupt. (See Section 2.34, *“Transmit Operation”* for a description of the transmit bid process.)
9	**TxUnderrun:** This bit is set if CS89712 runs out of data before it reaches the end of the frame (called a transmit underrun). If TxUnderruniE (Register B, BufCFG, Bit 9) is set, there is an interrupt.
A	**RxMiss:** If set, one or more receive frames have been lost due to slow movement of data out of the receive buffers. If RxMissiE (Register B, BufCFG, Bit A) is set, there is an interrupt.
B	**Rx128:** This bit is set after the first 128 bytes of an incoming frame have been received. This bit will allow the software the option of preprocessing frame data before the entire frame is received. If Rx128iE (Register B, BufCFG, Bit B) is set, there is an interrupt.
F	**RxDest:** When set, this bit shows that a receive frame has passed the Destination Address Filter criteria as defined in the RxCTL register (Register 5). This bit is useful as an early indication of an incoming frame. It will be earlier than Rx128 (Register C, BufEvent, Bit B). If RxDestiE (Register B, BufCFG, Bit F) is set, there is an interrupt.

Table 74. Buffer Event

Reset value is: 0000 0000 0000 1100

Notes: With any event register, like BufEvent, all bits are cleared upon readout. The software is responsible for processing all event bits.

3.18.10 Receiver Miss Counter Register (RxMISS, address offset 130h)

7:6	5:0	F:8
MissCount	010000	MissCount

The RxMISS counter (Bits 6 through F) records the number of receive frames that are lost (missed) due to the lack of available buffer space. If the MissOvfloiE bit (Register B, BufCFG, Bit D) is set, there is an interrupt when RxMISS increments from 1FFh to 200h. This interrupt provides the software with an early warning that the RxMISS counter should be read before it reaches 3FFh and starts over (by interrupting at 200h, the software has an additional 512 counts before RxMISS actually overflows). The RxMISS counter is cleared when read.

Bit	Description
5:0	**010000:** These bits provide an internal address used by the CS89712 to identify this as register 10, the Receiver Miss Counter. When reading this register, these bits will be 010000, where the LSB corresponds to Bit 0.
7:6 F:8	**MissCount:** The upper ten bits contain the number of missed frames.

Table 75. Receiver Miss Counter

Register's value is: 0000 0000 0001 0000

3.18.11 Transmit Collision Counter (TxCOL, address offset 132h)

7:6	5:0	F:8
ColCount	010010	ColCount

The TxCOL counter (Bits 6 through F) is incremented whenever the 10BASE-T Receive Pair (RXD+ / RXD-) becomes active while a packet is being transmitted. If the TxColOvfiE bit (Register B, BufCFG, Bit C) is set, there is an interrupt when TxCOL increments from 1FFh to 200h. This interrupt provides the software with an early warning that the TxCOL counter should be read before it reaches 3FFh and starts over (by interrupting at 200h, the software has an additional 512 counts before TxCOL actually overflows). The TxCOL counter is cleared when read.

Bit	Description
5:0	**010010:** These bits provide an internal address used by the CS89712 to identify this as register 12, the Transmit Collision Counter. When reading this register, these bits will be 010010, where the LSB corresponds to Bit 0.
7:6 F:8	**ColCount:** The upper ten bits contain the number of collisions.

Table 76. Transmit Collision Counter

Reset value is: 0000 0000 0001 0010

3.18.12 Line Control Register (LineCTL, address offset 112h)

7	6	5:0	F	E
SerTxOn	SerRxON	010011		LoRx Squelch

D	C	B	A	9:8
2-part DefDis	PolarityDis	Mod BackoffE		RSVD

LineCTL determines the configuration of the MAC engine and physical interface.

Bit	Description
5:0	**010011:** These bits provide an internal address used by the CS89712 to identify this as register 13, the Line Control Register.
6	**SerRxON:** When set, the receiver is enabled. When clear, no incoming packets pass through the receiver. If SerRxON is cleared while a packet is being received, reception is completed and no subsequent receive packets are allowed until SerRxON is set again.
7	**SerTxON:** When set, the transmitter is enabled. When clear, no transmissions are allowed. If SerTxON is cleared while a packet is being transmitted, transmission is completed and no subsequent packets are transmitted until SerTxON is set again.
B	**ModBackoffE:** When clear, the ISO/IEC standard backoff algorithm is used (see Section 2.26, *"Media Access Control Engine"*). When set, the Modified Backoff algorithm is used. (The Modified Backoff algorithm extends the backoff delay after each of the first three Tx collisions.)
C	**PolarityDis:** The 10BASE-T receiver automatically determines the polarity of the received signal at the RXD+/RXD- input (see Section 2.28, *"10BASE-T Transceiver"*). When this bit is clear, the polarity is corrected, if necessary. When set, no effort is made to correct the polarity. This bit is independent of the PolarityOK bit (Register 14, LineST, Bit C), which reports whether the polarity is normal or reversed.
D	**2-partDefDis:** Before a transmission can begin, the 89712 follows a deferral procedure. With the 2-partDefDis bit clear, the CS89712 uses the standard two-part deferral as defined in ISO/IEC 8802-3 paragraph 4.2.3.2.1. With the 2-partDefDis bit set, the two-part deferral is disabled.
E	**LoRxSquelch:** When clear, the 10BASE-T receiver squelch thresholds are set to levels defined by the ISO/IEC 8802-3 specification. When set, the thresholds are reduced by approximately 6 dB. This is useful for operating with "quiet" cables that are longer than 100 meters.
9:8	**RSVD:** Reserved; must be a "0" when writing to this register.

Table 77. Line Control

After reset, if no EEPROM is found by the CS89712, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 2.24, *"Programming the EEPROM"*.

Reset value is: 0000 0000 0001 0011

3.18.13 Line Status Register (LineST, address offset 134h)

7	6	5:0	F	E
LinkOK		010100		CRS
C	**B**	**A**	**9**	**8**
PolarityOK			10BT	RSVD

LineST reports the status of the Ethernet physical interface.

Bit	Description
5:0	**010100:** These bits provide an internal address used by the CS89712 to identify this as register 14, the Line Status Register. When reading this register, these bits will be 010100, where the LSB corresponds to Bit 0.
7	**LinkOK:** If set, the 10BASE-T link has not failed. When clear, the link has failed, either because the CS89712 has just come out of reset, or because the receiver has not detected any activity (link pulses or received packets) for at least 50 ms.
8	**RSVD:** Reserved; must be a “0” when writing to this register.
9	**10BT:** If set, the CS89712 is using the 10BASE-T interface.
C	**PolarityOK:** If set, the polarity of the 10BASE-T receive signal (at the RXD+ / RXD- inputs) is correct. If clear, the polarity is reversed. If PolarityDis (Register 13, LineCTL, Bit C) is clear, the polarity is automatically corrected, if needed. The PolarityOK status bit shows the true state of the incoming polarity independent of the PolarityDis control bit. Thus, if PolarityDis is clear and PolarityOK is clear, then the receive polarity is inverted, and corrected.
E	**CRS:** This bit tells the software the status of an incoming frame. If CRS is set, a frame is currently being received. CRS remains asserted until the end of frame (EOF). At EOF, CRS goes inactive in about 1.3 to 2.3 bit times after the last low-to-high transition of the recovered data.

Table 78. Line Status

Reset value is: 0000 0000 0001 0100

3.18.14 Self Control Register (SelfCTL, address offset 114h)

7	6	5:0	F	E	D
	RESET	010101	HCB1	HCB0	RESERVED

C	B	A	9	8
HC0E		RESERVED	RESERVED	SW Suspend

SelfCTL controls the operation of the LED outputs and the lower-power modes.

Bit	Description
5:0	**010101:** These bits provide an internal address used by the CS89712 to identify this as register 15, the Self Control Register.
6	**RESET:** When set, a chip-wide reset is initiated immediately. RESET is an Act-Once bit. This bit is cleared as a result of the reset.
8	**SWSuspend:** When set, the CS89712 Ethernet port enters the software initiated Suspend mode. Upon entering this mode, there is a partial reset. All Ethernet registers and circuits are reset. There is no transmit nor receive activity in this mode. To come out of software Suspend, the software issues an Write within the Ethernet port's assigned memory space.
C	**HC0E:** The $\overline{\text{LINKLED}}$ or $\overline{\text{HC0}}$ output pin is selected with this control bit. When HC0E is clear, the output pin is $\overline{\text{LINKLED}}$. When HC0E is set, the output pin is $\overline{\text{HC0}}$ and the HCB0 bit (Bit E) controls the pin.
E	**HCB0:** When $\overline{\text{HC0E}}$ (Bit C) is set, this bit controls the $\overline{\text{HC0}}$ pin. If HCB0 is set, $\overline{\text{HC0}}$ is low. If HCB0 is clear, $\overline{\text{HC0}}$ is high. HC0 may drive an LED or a logic gate. When HC0E (Bit C) is clear, this control bit is ignored.
F	**HCB1:** When $\overline{\text{HC1E}}$ (Bit D) is set, this bit controls the $\overline{\text{HC1}}$ pin. If HCB1 is set, $\overline{\text{HC1}}$ is low. If HCB1 is clear, $\overline{\text{HC1}}$ is high. $\overline{\text{HC1}}$ may drive an LED or a logic gate. When HC1E (Bit D) is clear, this control bit is ignored.

Table 79. Self Control

After reset, if no EEPROM is found by the CS89712, then the register has the following initial state. If an EEPROM is found, then the register's initial value may be set by the EEPROM. See Section 2.24, *"Programming the EEPROM"*.

Reset value is: 0000 0000 0001 0101

3.18.15 Self Status Register (SelfST, address offset 136h)

7	6	5:0	F	E	D
INITD	3.3V Active	010110			

C	B	A	9	8
EEsize	RSVD	EEPROM OK	EEPROM present	SIBUSY

SelfST reports the status of the EEPROM interface and the initialization process.

Bit	Description
5:0	**010110:** These bits provide an internal address used by the CS89712 to identify this as register 16, the Self Status Register. When reading this register, these bits will be 010110, where the LSB corresponds to Bit 0.
6	**3,3VActive:** If the CS89712 is operating on a 3.3 V supply, this bit is set. If the CS89712 is operating on a 5V supply, this bit is clear.
7	**INITD:** If set, the CS89712 initialization, including read-in of the EEPROM, is complete.
8	**SIBUSY:** If set, the EECS output pin is high indicating that the EEPROM is currently being read or programmed. The software must not write to Ethernet Port offset address 0040h nor 0042h until SIBUSY is clear.
9	**EEPROMpresent:** If the EEDataIn pin is low after reset, there is no EEPROM present, and the EEPROMpresent bit is clear. If the EEDataIn pin is high after reset, the CS89712 "assumes" that an EEPROM is present, and this bit is set.
A	**EEPROMOK:** If set, the checksum of the EEPROM readout was OK.
B	**RSVD:** Reserved; must be a “0” when writing to this register.
C	**EEsize:** This bit shows the size of the attached EEPROM and is valid only if the EEPROMpresent bit (Bit 9) and EEPROMOK bit (Bit A) are both set. If clear, the EEPROM size is either 128 words ('C56 or 'CS56) or 256 words (C66 or 'CS66). If set, the EEPROM size is 64 words ('C46 or 'CS46).

Table 80. Self Status

Reset value is: 0000 0000 0001 0110

3.18.16 Ethernet IRQ Control Register (BusCTL, address offset 116h)

7	6	5:0	F	E	D:8
	RSVD	010111	EnableIRQ		RSVD

Bit	Description
5:0	**010111:** These bits provide an internal address used by the CS89712 to identify this as register 17, the Bus Control Register.
6 D:8	**RSVD:** Reserved; must be a “0” when writing to this register.
F	**EnableIRQ:** When set, the CS89712 will generate an interrupt in response to an interrupt event (Section 2.31, *“Managing Interrupts & Status Queue”*). When cleared, the CS89712 will not generate any interrupts.

Table 81. Ethernet IRQ Control

After reset, if no EEPROM is found by the CS89712, then the register has the following initial state. If an EEPROM is found, then the register’s initial value may be set by the EEPROM. See Section 2.24, *“Programming the EEPROM”*.

Reset value is: 0000 0000 0001 0111

3.18.17 TX Bid Status Register (BusST, address offset 138h)

7	6	5:0	F	E	D
TxBidErr		011000			

C	B	A	9	8
				Rdy4Tx NOW

BusST describes the status of the current transmit operation.

Bit	Description
5:0	**011000:** These bits provide an internal address used by the CS89712 to identify this as register 18, the Bus Status Register. When reading this register, these bits will be 011000, where the LSB corresponds to Bit 0.
7	**TxBidErr:** If set, the software has commanded the Ethernet port to transmit a frame that the Ethernet port will not send. Frames that the Ethernet port will not send are: 1) Any frame greater than 1514 bytes, provided that InhibitCRC (Register 9, TxCMD, Bit C) is clear. 2) Any frame greater than 1518 bytes. Note that this bit is not set when transmit frames are too short.
8	**Rdy4TxNOW:** Rdy4TxNOW signals the software that the Ethernet port is ready to accept a frame from the software for transmission. This bit is similar to Rdy4Tx (Register C, BufEvent, Bit 8) except that there is no interrupt associated with Rdy4TxNOW. The software can poll the CS89712 and check Rdy4TxNOW to determine if the CS89712 is ready for transmit. (See Section 2.34, *"Transmit Operation"* for a description of the transmit bid process.)

Table 82. TX Bid Status

Reset value is: 0000 0000 0001 1000

3.18.18 Test Control Register (TestCTL, address offset 0118h)

7	6	5:0	F	E	D
DisableLT		011001		FDX	

C	B	A	9	8
	Disable Backoff	RSVD	ENDEC loop	

TestCTL controls the diagnostic test modes of the CS89712.

Bit	Description
	011001: These bits provide an internal address used by the CS89712 to identify this as register 19, the Test Control Register.
	DisableLT: When set, the 10BASE-T interface allows packet transmission and reception regardless of the link status. DisableLT is used in conjunction with the LinkOK (Register 14, LineST, Bit 7) as follows:

LinkOK	DisableLT	
0	0	No packet transmission for reception allowed. Transmitter sends link impulses.
0	1	DisableLT overrides LinkOK to allow packet transmission and reception. Transmitter does not send link pulses.
1	N/A	Disable has no meaning if LinkOK = 1.

	ENDECloop: When set, the CS89712 enters internal loopback mode where the internal Manchester encoder output is connected to the decoder input. The 10BASE-T are disabled. When clear, the CS89712 is configured for normal operation
	Disable Backoff: When set, the backoff algorithm is disabled. The CS89712 transmitter looks only for completion of the inter packet gap before starting transmission. When clear, the back-off algorithm is used.
	FDX: When set, 10BASE-T full duplex mode is enabled and CRS (Register 14, LineST, Bit E) is ignored. This bit must be set when performing loopback tests on the 10BASE-T port. When clear, the CS89712 is configured for standard half-duplex 10BASE-T operation.
	RSVD: Reserved; must be a “0” when writing to this register.

Table 83. Test Control

At reset, if no EEPROM is found by the CS89712, then the register has the following initial state. If an EEPROM is found, then the register’s initial value may be set by the EEPROM. See Section 2.24, *“Programming the EEPROM”*.

Reset value is: 0000 0000 0001 1001

3.19 Initiate Transmit Registers

3.19.1 Transmit Command Request Register (TxCMD, address offset 144h)

7:6	5:0	F	E	D
TxStart	001001			TxPadDis

C	B	A	9	8
InhibitCRC			Onecoll	Force

The word written to offset 0144h tells the CS89712 how the next packet should be transmitted. **Note that this Ethernet Port location is write-only, and the written word can be read from Ethernet address offset 0108h.** The

CS89712 does not transmit a frame if TxLength is less than 3. See Section 2.34, *"Transmit Operation"*.

Bit	Description
	001001: These bits provide an internal address used to identify this as register 9. When reading this register, these bits will be 001001, where the LSB corresponds to Bit 0.
	TxStart: This pair of bits determines how many bytes are transferred to the CS89712 before the MAC starts the packet transmit process. **Bit 7** / **Bit 6** 0 / 0 — Start transmission after 5 bytes are in the CS89712 0 / 1 — Start transmission after 381 bytes are in the CS89712 1 / 0 — Start transmission after 1021 bytes are in the CS89712 1 / 1 — Start transmission after the entire frame is in the CS89712
	Force: When set in conjunction with a new transmit command, any transmit frames waiting in the transmit buffer are deleted. If a previous packet has started transmission, that packet is terminated within 64 bit times with a bad CRC.
	Onecoll: When this bit is set, any transmission will be terminated after only one collision. When clear, the CS89712 allows up to 16 normal collisions before terminating the transmission.
	InhibitCRC: When set, the CRC is not appended to the transmission.
	TxPadDis: When TxPadDis is clear, if the software gives a transmit length less than 60 bytes and InhibitCRC is set, then the Ethernet port pads to 60 bytes. If the software gives a transmit length less than 60 bytes and InhibitCRC is clear, then the CS89712 pads to 60 bytes and appends the CRC. When TxPadDis is set, the CS89712 allows the transmission of runt frames (a frame less than 64 bytes). If InhibitCRC is clear, the CS89712 appends the CRC. If InhibitCRC is set, the CS89712 does not append the CRC.

Table 84. Transmit Command

Since this register is write-only, its initial state after reset is undefined.

3.19.2 Transmit Length (TxLength, address offset 146h)

Address 0147h	Address 0146h
Most-significant byte of Transmit Frame Length	Least-significant byte of Transmit Frame Length

This register is used in conjunction with the TxCMD register. When a transmission is initiated via a command in TxCMD, the length of the transmitted frame is written into this register. The length of the transmitted frame may be modified by the configuration of the TxPadDis and InhibitCRC bits in the TxCMD register. See Table 30, and Section 2.34, *"Transmit Operation"*. TxLength must be >3 and < 1519.

Since this register is write-only, its initial state after reset is undefined.

3.20 Address Filter Registers

3.20.1 Logical Address Filter (hash table, address offset 0150h)

Address 0157h	Address 0156h	Address 0155h	Address 0154h	Address 0153h	Address 0152h	Address 0151h	Address 0150h
Most-significant byte of hash filter.							Least-significant byte of hash filter.

The CS89712 hashing decoder circuitry compares its output with one bit of the Logical Address Filter Register. If the decoder output and the Logical Address Filter bit match, the frame passes the hash filter and the Hashed bit

(Register 4, RxEvent, Bit 9) is set. See Section 2.32.7, *"Receive Ethernet Port Locations"*.

Reset value is: 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

3.20.2 Individual Address (IEEE or MAC address, address offset 0158h)

Address 0015Dh	Address 015Ch	Address 015Bh	Address 015Ah	Address 0159h	Address 0158h
Octet 5 of IA					Octet 0 of IA

For example: if the MAC address is "00242010ABCD" then load register address 0158h with 00h, 0159h with 24h, 015Ah with 21h, etc.

The value of this register must be loaded from external storage, for example, from the EEPROM. See Section 2.24, *"Programming the EEPROM"*. If the CS89712 is not able to load the IA from the EEPROM, then after a reset this register is undefined, and the driver must write an address to this register.

4. TEST & DEBUG MODES

The CS89712 supports a number of hardware activated test and debug modes.

4.1 Entering test modes

Test modes are activated by the pin combinations shown in Table 85. All latched signals will only alter test modes while NPOR is low, their state is latched on the rising edge of NPOR. This allows these signals to be used normally during various test modes.

Within each test mode, a selection of pins is used as multiplexed outputs or inputs to provide / monitor the test signals unique to that mode.

4.1.1 Oscillator and PLL Bypass Mode

This mode is selected by nTEST[0] = 1 and nTEST[1] = 0.

In this mode, all the internal oscillators and PLL are disabled, and the appropriate crystal oscillator pins become the direct external oscillator inputs bypassing the oscillator and PLL. MOSCIN must be driven by a 36.864 MHz clock source and RTCIN by a 32.768 kHz source.

4.1.2 Oscillator and PLL Test Mode

This mode is selected by nTEST[0] = 0, nTEST[1] = 1, Latched nURESET = 0

This test mode will enable the main oscillator and will output various buffered clock and test signals derived from the main oscillator, PLL, and 32-kHz oscillator. All internal logic in the CS89712 will be static and isolated from the oscillators, with the exception of the 6-bit ripple counter used to generate 576 kHz and the Real-Time Clock divide chain. Port A is used to drive the inputs of the PLL directly, and the various clock and PLL outputs are monitored on the COL pins. Table 86 defines the CS89712 signal pins used in this test mode. This mode is only intended to allow test of the oscillators and PLL. Note that these inputs are inverted before being passed to the PLL to ensure that the default state of the port (all zero) maps onto the correct default state of the PLL (TSEL = 1, XTALON = 1, PLLON = 1, D0 = 0, D1 = 1, PLLBP = 0). This state will produce the correct frequencies as shown in Table 86. Any other combinations are for testing the oscillator and PLL and should not be used in-circuit.

Test Mode	Latched nMEDCHG	Latched PE[0]	Latched PE[1]	Latched nURESET	nTEST[0]	nTEST[1]
Normal operation (32-bit boot)	1	0	0	X	1	1
Normal operation (8-bit boot)	1	1	0	X	1	1
Normal operation (16-bit boot)	1	0	1	X	1	1
Alternative test ROM boot	0	X	X	X	1	1
Oscillator / PLL bypass	X	X	X	X	1	0
Oscillator / PLL test mode	X	X	X	0	0	1
ICE Mode	X	X	X	1	0	0
System test (all HiZ)	X	X	X	0	0	0

Table 85. CS89712 Hardware Test Modes

Signal	I/O	Pin	Function
TSEL *	I	PA5	PLL test mode
XTLON *	I	PA4	Enable to oscillator circuit
PLLON *	I	PA3	Enable to PLL circuit
PLLBP	I	PA0	Bypasses PLL
RTCCLK	O	COL0	Output of RTC oscillator
CLK1	O	COL1	1 Hz clock from RTC divider chain
OSC36	O	COL2	36 MHz divided PLL main clock
CLK576K	O	COL4	576 KHz divided from above
VREF	O	COL6	Test clock output for PLL

Table 86. Oscillator and PLL Test Mode Signals

4.1.3 Debug / ICE Test Mode

This mode is selected by nTEST0 = 0, nTEST1 = 0, Latched nURESET = 1.

Selection of this mode enables the debug mode of the ARM720T. By default, this is disabled which saves approximately 3% on power.

4.1.4 Hi-Z (System) Test Mode

This mode selected by nTEST0 = 0, nTEST1 = 0, Latched nURESET = 0.

This test mode asynchronously disables all output buffers on the CS89712. This has the effect of removing the CS89712 from the PCB so that other devices on the PCB can be in-circuit tested. The internal state of the CS89712 is not altered directly by this test mode.

4.1.5 Software Selectable Test Functionality

When bit 11 of the SYSCON register is set high, internal peripheral bus register accesses are output on the main address and data buses as though they were external accesses to the address space addressed by nCS[5]. Hence, nCS[5] takes on a dual role, it will be active as the strobe for internal accesses and for any accesses to the standard address range for nCS[5]. Additionally, in this mode, the internal signals shown in Table 87 are multiplexed out of the device on port pins.

Signal	I/O	Pin	Function
CLK	O	PE0	Waited clock to CPU
nFIQ	O	PE1	nFIQ interrupt to CPU
nIRQ	O	PE2	nIRQ interrupt to CPU

Table 87. Software Selectable Test Functionality

This test is not intended to be used when LCD DMA accesses are enabled. This is due to the fact that it is possible to have internal peripheral bus activity simultaneously with a DMA transfer. This would cause bus contention to occur on the external bus.

The “Waited clock to CPU” is an internally ANDed source that generates the actual CPU clock. Thus, it is possible to know exactly when the CPU is being clocked by viewing this pin. The signals nFIQ and nIRQ are the two output signals from the internal interrupt controller. They are input directly into the ARM720T processor.

4.1.6 Loopback/Collision Diagnostic Tests

Internal and external Loopback and Collision tests can be used to verify the CS89712's Ethernet port functionality. Internal tests allow the major digital functions to be tested, independent of the analog functions. During these tests, the Manchester encoder is connected to the decoder. All digital circuits are operational, and the transmitter and receiver are disabled. External test modes allow the complete chip to be tested without connecting it directly to an Ethernet network.

4.1.7 Loopback Tests

During Loopback tests, the internal Carrier Sense (CRS) signal, used to detect collisions, is ignored, allowing packet reception during transmission.

4.1.8 10BASE-T Loopback/Collision Tests

10BASE-T Loopback and Collision Tests are controlled by two bits in the Test Control register: FDX (Register 19, TestCTL, Bit E) and ENDECloop (Register 19, TestCTL, Bit 9). Table 88 describes these tests.

4.2 In-Circuit Emulation

4.2.1 Introduction

EmbeddedICE™ is an extension to the architecture of the ARM family of processors, and provides the ability to debug cores that are deeply embedded into systems. It consists of three parts:

1) A set of extensions to the ARM core
2) The EmbeddedICE macrocell, which provides external access to the extensions

Test Mode	FDX	ENDECloop	DisableLT	Description of Test
10BASE-T Internal Loopback	1	1	1	Transmit a frame and verify that the frame is received without error.
10BASE-T Internal Collision	0	1	1	Transmit frames and verify that collisions are detected and that the internal counters function properly. After 16 collisions, verify that 16coll (Register 8, TxEvent, Bit F) is set.
10BASE-T External Loopback	1	0	0	Connect TXD+ to RXD+ and TXD- to RXD-. Transmit a frame and verify that the frame is received without error.
10BASE-T External Collision	0	0	0	Connect TXD+ to RXD+ and TXD- to RXD-. Transmit frames and verify that collisions are detected and that internal counters function properly. After 16 collisions, verify that 16coll (Register 8, TxEvent, Bit F) is set.

Table 88. 10BASE-T Loopback and Collision Tests

3) The EmbeddedICE interface, which provides communication between the host computer and the EmbeddedICE macrocell

The EmbeddedICE macrocell is programmed, in a serial fashion, through the TAP controller on the ARM via the JTAG interface. The EmbeddedICE macrocell is by default disabled to minimize power usage, and must be enabled at boot-up to support this functionality.

4.2.2 Functionality

The ICEBreaker module consists of two real-time watchpoint units together with a control and status register. One or both of the units can be programmed to halt the execution of the instructions by the ARM processor. Execution is halted when either a match occurs between the values programmed into the ICEBreaker and the values currently appearing on the address bus, data bus, and the various control signals. Any bit can be masked to remove it from the comparison. Either unit can be programmed as a watchpoint (monitoring data accesses) or a breakpoint (monitoring instruction fetches).

Using one of these watchpoint units, an unlimited number of software breakpoints (in RAM) can be supported by substitution of the actual code.

Note: The EXTERN[1:0] signals from the ICEBreaker module are not wired out in this device. This mechanism is used to allow watchpoints to be dependent on an external event. This behavior can be emulated in software via the ICEBreaker control registers.

A more detailed description is available in the ARM Software Development Toolkit User Guide and Reference Manual. The ICEBreaker module and its registers are fully described in the "*ARM7TDMI*" Data Sheet.

5. MECHANICAL INFORMATION

5.1 256-PBGA PIN DIAGRAM

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
TXDP	TXDN	N/C	N/C	RXDP	RXDN	RES	VssA	XTAL 1	XTAL 2	EECS	EE DOUT	Vss I/O	DD 3	SD CS0	CS 0	A
VddA	VssA	Vss I/O	VssA	VddA	VssA	VddA	LINK LED	LAN LED	VssA	EESK	EE DIN	N/C	DD 2	SDQM 3	CS 1	B
D 3	D 4	D 5	Vss I/O	D 6	Vdd I/O	N/C	N/C	Vss I/O	N/C	N/C	Vdd I/O	CL 1	DD 1	SDQM 2	CS 3	C
MOE/ SDCAS	Vdd I/O	Vss I/O	D 2	N/C	A 19	D 0	N/C	D 15	D 13	N/C	Vdd CORE	FRM	DD 0	SD CKE	CS 4	D
A 16	A 15	D 1	A 18	A 6	A 17	MWE/ SDWE	D 12	Vss I/O	D 10	Vdd I/O	CL 2	M	SD CS1	SD CLK	EXP CLK	E
A 14	A 13	Vss I/O	Vss I/O	A 4	A 5	WAKE UP	PWR FL	Vdd I/O	D 11	Vss CORE	Vss I/O	Vdd I/O	Vss I/O	CS 5	WORD	F
A 12	VDD I/O	A 11	A 3	ETH ENBL	Vss I/O	Vdd OSC	MOSC IN	Vss OSC	Vss I/O	Vss I/O	TXD 2	TDI	EXP RDY	Vdd I/O	WRITE	G
A 10	A 9	A 7	ETH ENBL	uRST	Vss I/O	Vss I/O	MOSC OUT	PA 4	PA 6	PB 6	PB 5	PB 1	RXD 2	Vss I/O	RUN	H
N/C	A 2	A 1	D 7	nMED CHG	D 8	Vss I/O	TEST 1	PE 2	LED DRV	Vdd I/O	PA 3	TDO	PB 4	PB0/ PRDY1	PB 7	J
N/C	A 0	N/C	nBAT CHG	A 8	Vdd I/O	SSI TXDA	SSI TXFR	PD 1	PD 2	TEST 0	PA 1	PHD IN	PA 5	PB 2	PB 3	K
Vdd I/O	Vdd I/O	BAT OK	VSS I/O	D 9	D 14	Vss I/O	DRV 0	SSI RXFR	DRV 1	PD 3	PE 0	TXD 1	CTS	PA 2	PA 7	L
nPOR	nTRST	Vss I/O	nEXT PWR	D 17	COL 6	Vdd I/O	FB 0	ADC CLK	Vss I/O	Vdd I/O	PE 1	RXD 1	Vss I/O	DSR	PA 0	M
D 18	D 21	D 16	D 19	Vdd I/O	COL 1	COL 0	COL 2	COL 7	SMP CLK	ADC CS	PD 5	nEXT FIQ	N/C	nEINT 1	DCD	N
A 20	A 23	D 20	D 22	A 22	D 28	Vss I/O	D 31	COL 3	FB 1	Vdd I/O	SSI RXDA	PD6/ SDQM0	Vdd RTC	Vss RTC	nEINT 2	P
D 23	D 24	A 21	Vss I/O	Vdd I/O	Vss I/O	A 27	D 30	BUZ	COL 4	Vdd I/O	ADC IN	PD 0	PD 4	RTC OUT	RTC IN	R
A24	HALF WORD	D 25	A 25	D 26	A 26	D 27	D 29	TCLK	COL 5	ADC OUT	Vdd I/O	Vss I/O	SSI CLK	TMS	PD7/ SDQM1	T

Table 89. 256-Ball PBGA Ball Listing (bottom view)

5.2 External Signal Functions

Function	Signal Name		Signal	Description
Data bus	D[0-31]		I/O	32-bit system data bus for memory, DRAM, and I/O interface
Address bus	A[0-14]		O	15 bits of system byte address during memory and expansion cycles
	A[13-27] DRA[0-14]			DRA[0-14] is multiplexed with A[13-27], offering additional power savings since the lightest loading is expected on the high order ROM address lines. Whenever the CS89712 is in the Standby State, the external address and data buses are driven low. The RUN signal is used internally to force these buses to be driven low. This is done to prevent peripherals that are powered-down from draining current. Also, the internal peripheral's signals get set to their Reset State.
Memory Interface	BA[0-1]/A[13/14]		I/O	SDRAM bank select pins.
	SDCS[0-1]	A2, E3	O	SDRAM chip selects.
	SDCLK	E2	O	SDRAM clock.
	SDCKE	D2	O	SDRAM clock enable.
	SDQM[2-3]	C2, B2		SDRAM byte masks. SDQM0-1 are muxed with Port D data pins.
	nMOE/nSDCAS	D16	O	Memory output enable/SDRAM CAS control signal
	nMWE/nSDWE	E10	O	Memory write enable/SDRAM write enable control signal
	nCS[0-1, 3]	A1, B1, C1	O	Chip select; active low, SRAM-like chip selects for expansion
	Ethernet Enable / nCS2	G12, H13	O	Chip select; active low, indicates either Ethernet port or CS2 expansion range activity. Pins G12 and H13 must be tied together.
	nCS[4-5]	D1, F2	O	Chip select; active low, CS for expansion or for CL-PS6700 select
	EXPRDY	G3	I	Expansion port ready; external expansion devices drive this low to extend the bus cycle. This is used to insert wait states for an external bus cycle.
	WRITE	G1	O	Write strobe, low during reads, high during writes from the CS89712
	WORD/ HALFWORD	F1 T15	O	To do write accesses of different sizes Word and Half-Word must be externally decoded. The encoding of these signals is as follows:

Access Size	Word	Half-Word
Word	1	0
Half-Word	*	1
Byte	0	0

	EXPCLK	E1	I/O	Expansion clock rate is the same as the CPU clock for 18 MHz. It runs at 36.864 MHz for 36,49 and 74 MHz modes.
Interrupts	nMEDCHG/ nBROM	J12	I	Media changed input; active low, deglitched. Used as a general purpose FIQ interrupt during normal operation. It is also used on power up to configure the processor to either boot from the internal Boot ROM, or from external memory. When low, the chip will boot from the internal Boot ROM.
	nEXTFIQ	N4	I	External active low fast interrupt request input
	nEINT[1:2]	N2, P1	I	Two general purpose, active low interrupt inputs

Table 90. External Signal Functions

Function	Signal Name	Signal		Description
Power Management	nPWRFL	F9	I	Power fail input; active low, deglitched input to force system into the Standby State (Note 1)
	BATOK	L14	I	Main battery OK input; falling edge generates a FIQ, a low level in the Standby State inhibits system start up; deglitched input (Note 2)
	nEXTPWR	M13	I	External power sense; must be driven low if the system is powered by an external source
	nBATCHG	K13	I	New battery sense; driven low if battery voltage falls below the "no-battery" threshold; it is a deglitched input (Note 2)
State Control	nPOR	M16	I	Power-on reset input. This signal is not deglitched. When active it completely resets the entire system, including all the RTC registers. Upon power-up, the signal must be held active low for a minimum of 100 µsec after V_{DD} has settled. During normal operation, nPOR needs to be held low for at least one clock cycle of the selected clock speed (i.e., when running at 74 MHz, the pulse width of nPOR needs to be > 14 nsec). Note that nURESET, TEST(0), TEST(1), PE(0), PE(1), PE(2), DRIVE(0), DRIVE(1), DD(0), DD(1), DD(2), and DD(3) are all latched on the rising edge of nPOR.
	RUN	H1	O	This pin is the RUN signal. The pin will be high when the system is active or idle, low while in the Standby State. See Table 91.
	WAKEUP	F10	I	Wake up is a deglitched input signal. It must be held high for at least 125 µsec to guarantee its detection. Once detected it forces the system into the Operating State from the Standby State. It is only active when the system is in the Standby State. This pin is ignored when the system is in the Idle or Operating State. It is used to wakeup the system after first power-up, or after software has forced the system into the Standby State. WAKEUP will be ignored for up to two seconds after nPOR goes HIGH. Therefore, the external WAKEUP logic must be designed to allow it to rise and stay HIGH for at least 125 usec, two seconds after nPOR goes HIGH. (Note 2)
	nURESET	H12	I	User reset input; active low deglitched input from user reset button. This pin is also latched upon the rising edge of nPOR and read along with the input pins nTEST[0-1] to force the device into special test modes. nURESET does not reset the RTC. (Note 2)
DAI, Codec or SSI2 Interface (See Table 22) .	SSICLK	T3	I/O	DAI/Codec/SSI2 clock signal
	SSITXFR	K9	I/O	DAI/Codec/SSI2 serial data output frame/synchronization pulse output
	SSITXDA	K10	O	DAI/Codec/SSI2 serial data output
	SSIRXDA	P5	I	DAI/Codec/SSI2 serial data input
	SSIRXFR	L8	I/O	SSI2 serial data input frame/synchronization pulse DAI external clock input
ADC Interface (SSI1)	ADCCLK	M8	O	Serial clock output
	nADCCS	N6	O	Chip select for ADC interface
	ADCOUT	T6	O	Serial data output
	ADCIN	R5	I	Serial data input
	SMPCLK	N7	O	Sample clock output

Table 90. External Signal Functions (Continued)

Function	Signal Name	Signal		Description
IrDA and RS232 Interfaces	LEDDRV	J7	O	Infrared LED drive output (UART1)
	PHDIN	K4	I	Photo diode input (UART1)
	TXD[1-2]	L4, G5	O	RS232 UART1 and 2 TX outputs
	RXD[1-2]	M4, H3	I	RS232 UART1 and 2 RX inputs
	DSR	M2	I	RS232 DSR input
	DCD	N1	I	RS232 DCD input
	CTS	L3	I	RS232 CTS input
LCD	DD[0-3]	D3, C3, B3, A3	I/O	LCD serial display data; pins can be used on power up to read the ID of some LCD modules (See Figure 31).
	CL[1]	C4	O	LCD line clock
	CL[2]	E5	O	LCD pixel clock
	FRM	D4	O	LCD frame synchronization pulse output
	M	E4	O	LCD AC bias drive
Keyboard and Buzzer drive LED Flasher	COL[0-7]		O	Keyboard column drives (SYSCON1)
	BUZ	R8	O	Buzzer drive output (SYSCON1)
	PD[0]/ LEDFLSH	R4	O	LED flasher driver — multiplexed with Port D bit 0. This pin can provide up to 4 mA of drive current.
General Purpose I/O	PA[0:7]		I/O	Port A I/O (bit 6 for boot clock option, bit 7 for CL-PS6700 PRDY input); also used as keyboard row inputs
	PB[0]/PRDY1 PB[1]/PRDY2 PB[2:7]		I/O	Port B I/O. All eight Port B bits can be used as GPIOs. When the PC CARD1 or 2 control bits in the SYSCON2 register are de-asserted, PB[0] and PB[1] are available for GPIO. When asserted, these port bits are used as the PRDY signals for connected CL-PS6700 PC Card Host Adapter devices.
	PD[0:7]		I/O	Port D I/O
	PE[0]/ BOOTSEL[0]	L5	I/O	Port E I/O (3 bits only). Can be used as general purpose I/O during normal operation.
	PE[1]/ BOOTSEL[1]	M5	I/O	During power-on reset, PE[0] and PE[1] are inputs and are latched by the rising edge of nPOR to select the memory width that the CS89712 will use to read from the boot code storage device (i.e., external 8-bit-wide FLASH bank).
	PE[2]	J8	I/O	During power-on reset, PE[2] is latched by the rising edge of nPOR to enable the PLL clocking mode.
PWM Drives	DRIVE[0:1]	L9, L7	I/O	PWM drive outputs. These pins are inputs on power up to determine what polarity the output of the PWM should be when active. Otherwise, these pins are always an output (See Table 91).
	FB[0:1]	M9, P7	I	PWM feedback inputs
Boundary Scan	TDI	G4	I	JTAG data in
	TDO	J4	O	JTAG data out
	TMS	T2	I	JTAG mode select
	TCLK	T8	I	JTAG clock
	nTRST	M15	I	JTAG async reset
Test	nTEST[0:1]	K6, J9	I	Test mode select inputs are used in conjunction with the power-on latched state of nURESET to select the various device test modes.

Table 90. External Signal Functions (Continued)

Function	Signal Name		Signal	Description
Oscillators	MOSCIN MOSCOUT	G9 H9	I O	Main 3.6864 MHz oscillator for 18.432 MHz–73.728 MHz PLL
	RTCIN RTCOUT	R1 R2	I O	Real-Time Clock 32.768 kHz oscillator
Ethernet EEPROM	EECS	A6	O	EEPROM Chip Select - Active-high EEPROM select.
	EESK	B6	O	EEPROM Serial Clock **-** clocks data in/out of the EEPROM.
	EEDataOut	A5	O	EEPROM Data Out **-** used to send data to the EEPROM. Connects to the DI pin on the EEPROM. When $\overline{TEST}$ is low, this pin becomes the output for the Boundary Scan Test.
	EEDataIn	B5	I	EEPROM Data In **-** serial input used to receive data from the EEPROM. Connects to the DO pin on the EEPROM. EEDataIn is also used to sense the presence of the EEPROM. This signal has a weak internal pullup.
10BASE-T Interface	TXD+/TXD-	A16, A15	O	10BASE-T Transmit, Differential Output Pair **-** drives 10 Mb/s Manchester-encoded data to the 10BASE-T transmit pair.
	RXD+/RXD-	A12, A11	I	**10BASE-T Receive, Differential Input Pair -** Differential input pair receives 10 Mb/s Manchester-encoded data from the 10BASE-T receive pair.
General Ethernet Pins	RES	A10	I	**Reference Resistor, Input -** This input should be connected to a 4.99KW ± 1% resistor needed for biasing of internal analog circuits.
	XTAL[1:2]	A8, A7		**Crystal, Input/Output -** A 20 MHz crystal should be connected to these pins. If a crystal is not used, a 20 MHz signal should connect to XTAL1 and XTAL2 should be left open.
	$\overline{LINKLED}$ or $\overline{HC0}$	B9	O	**Link Good LED or software Controlled Output 0, Open Drain Output-** When the HCE0 bit of the Self Control register (Register 15) is clear, this active-low output is low when the CS89712 detects the presence of valid link pulses. When the HC0E bit is set, the software may drive this pin low by setting the HCBO in the Self Control register.
	LANLED	B8	O	**LAN Activity LED, Open Drain Output -** During normal operation, this active-low output goes low for 6 ms whenever there is a receive packet, a transmit packet, or a collision.
No Connects	N/C			No connects should be left as no connects; do not connect to ground

Table 90. External Signal Functions (Continued)

Notes: 1. All deglitched inputs are via the 16.384 kHz clock. Each deglitched signal must be held active for at least two clock periods. Therefore, the input signal must be active for at least ~125 µs to be detected cleanly.
2. The RTC crystal must be populated for the device to function properly.

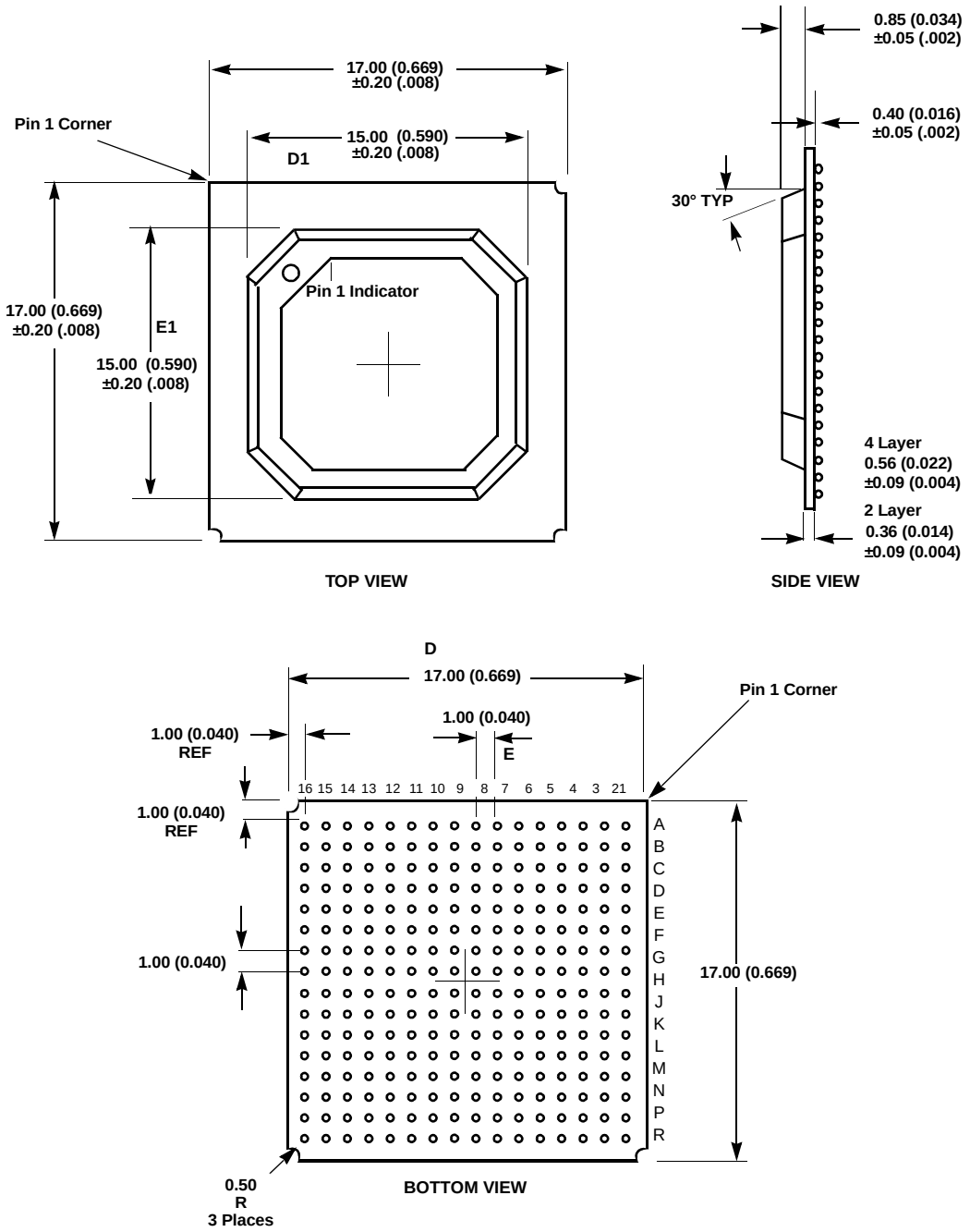
5.2.1 Output Bi-Directional Pins

RUN	The RUN pin is looped back in to skew the address and data bus from each other.
Drive [0-1]	Drive 0 and 1 are looped back in on power up to determine what polarity the output of the PWM should be when active.
DD[3:0]	DD[3:0] are looped back in on power up to enable the reading of the ID of some LCD modules.

Table 91. Output Bi-Directional Pins

Note: Output pins above are bi-directional to enable monitored output to provide accurate control of timing or duration

5.3 256 PBGA Package Dimensions



JEDEC #: MO-151

Ball Diameter: 0.50 mm ± 0.10 mm, package body 17 x 17 mm

Notes: 1. Dimensions are in millimeters (inches), and controlling dimension is millimeter.

2. Before designing with this device, contact Cirrus Logic for the latest package information.

6. ELECTRICAL/THERMAL INFO

6.1 Absolute Maximum Ratings

DC Core, PLL, and RTC Supply Voltage	2.9 V
DC I/O Supply Voltage (Pad Ring)	3.6 V
DC Pad Input Current	±10 mA/pin; ±100 mA cumulative
Storage Temperature, No Power	–40°C to +125°C

RECOMMENDED OPERATING CONDITIONS

DC core, PLL, and RTC Supply Voltage	2.5 V ± 0.2 V
DC I/O Supply Voltage (Pad Ring)	3.3V ± 0.165 V
DC Input / Output Voltage	0V – I/O supply voltage
Operating Temperature	Commercial 0°C to +70°C

6.2 DC Characteristics

All characteristics are specified at V_{DD} = 3.3 volts and V_{SS} = 0 volts over an operating temperature of 0° C to +70° C for all frequencies of operation. The current consumption figures relate to typical conditions at 2.5 V core, 3.3V I/O, 18.432 MHz operation with the PLL enabled.

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
CMOS input high voltage	VIH	1.7		V_{DD} + 0.3	V	V_{DDC} = 2.5 V V_{DDIO} = 3.3 V
CMOS input low voltage	VIL	-0.3		0.8	V	V_{DDC} = 2.5 V V_{DDIO} = 3.3 V
Schmitt trigger negative going threshold	VT-	0.8		1.2 (Typ)	V	
Schmitt trigger hysteresis	Vhst	0.1		0.4	V	VIL to VIH
CMOS output high voltage Output drive 1 Output drive 2	VOH	V_{DD} – 0.2 2.5 2.5			V V V	IOH = 0.1 mA OH = 4 mA OH = 12 mA
CMOS output low voltage Output drive 1 Output drive 2	VOL			0.3 0.5 0.5	V V V	IOL = –0.1 mA OL = –4 mA OL = –12 mA
Input leakage current (Note 1)	IIN			10.0	µA	VIN = V_{DD} or GND
Output tri-state leakage current (Notes 2 and 3)	IOZ	25		100	µA	VOUT = V_{DD} or GND

Notes:
1. The leakage value given assumes that the pin is configured as an input pin but is not currently being driven. An input pin not driven will have a maximum leakage of 1 µA. When the pin is driven, there will be no leakage.
2. Assumes buffer has no pull-up or pull-down resistors.
3. The leakage value given assumes that the pin is configured as an output pin but is not currently being driven. An output pin not driven will have leakage between 25 µA and 100 µA. When the pin is driven, there will be no leakage. Note that this applies to all output pins and all I/O pins configured as outputs.

DC CHARACTERISTICS (Continued)

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Input capacitance	CIN		8		10.0	pF
Output capacitance	COUT	8		10.0	pF	
Transceiver capacitance	CI/O	8		10.0	pF	
Startup current consumption	$IDD_{startup}$			TBD	µA	Initial 100 ms from power up, Cache disabled, 32 kHz oscillator not stable, POR signal at VIL, all other I/O static, VIH = V_{DD} ± 0.1 V, VIL = GND ± 0.1 V
Standby current consumption Core, Osc, RTC @2.5 V, I/O @ 3.3 V	$IDD_{standby}$		TBD	TBD	µA	Just 32 kHz oscillator running, Cache disabled, all other I/O static, VIH = V_{DD} ± 0.1 V, VIL = GND ± 0.1 V
Snooze current consumption	IDD_{snooze}		TBD		mA	Both oscillators running, CPU static, LCD from oc-SRAM.
Idle current consumption	IDD_{idle}		TBD		mA	Both oscillators running, CPU static, Cache disabled, LCD refresh active, VIH = V_{DD} ± 0.1 V, VIL = GND ± 0.1 V

DC CHARACTERISTICS (

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Operating current consumption (74 MHz) Core/Osc/RTC @2.5 V, I/O @ 3.3 V Ethernet active	$IDD_{operating}$		90	TBD	mA	All system active, running typical program, cache disabled, LCD inactive
Operating power consumption (74 MHz)			270	TBD	mW	see above
Ethernet XTAL1 Input Low Voltage	V_{IXH}	-	0.2	-	V	
Ethernet XTAL1 Input High Voltage	V_{IXH}	-	1.5	-	V	
XTAL1 Input Low Current	I_{IXL}	-	-40	-	µA	
XTAL1 Input High Current	I_{IXH}	-	40	-	µA	
Ethernet Port Power Supply Current while Active 3.3V	I_{DD}	-	65	-	mA	
Ethernet Port Software Suspend Mode Current	$I_{DDSWSUS}$	-	1.0	-	mA	
10 BaseT Transmitter Differential Output Voltage (Peak)	V_{OD}	2.2	-	2.8		Secondary side of Transformer
Receiver Normal Squelch Level (Peak)	V_{ISQ}	300	-	525		Primary side of Transformer
Receiver Low Squelch Level (LoRxSquelch bit set)	V_{SQL}	125	-	290		Primary side of Transformer
Standby supply voltage	$V_{DDstandby}$	TBD			V	Minimum standby voltage for state retention and RTC operation only

Notes:
4. Power dissipation values can be derived by multiplying the IDD current by 2.5 V Core/OSC or 3.3V I/O.
5. The RTC of the CS89712 should be brought up at room temperature. The RTC OSC will NOT function properly if it is brought up at –40°C. Once operational, it will continue to operate down to –40°C.
6. A typical design will provide 3.3 V to the I/O supply (i.e., V_{DD}IO), and 2.5 V to the remaining logic. This is to allow the I/O to be compatible with 3.3 V powered external logic (i.e., 3.3 V SDRAMs).
7. Pull-up current = 50 µA typical at V_{DD} = 3.3 volts.

6.3 AC Characteristics

These are specified at V_{DD} = 3.3 volts and V_{SS} = 0 volts over an operating temperature of 0°C to +70°C. The timing values are referenced to 1/2 V_{DD}.

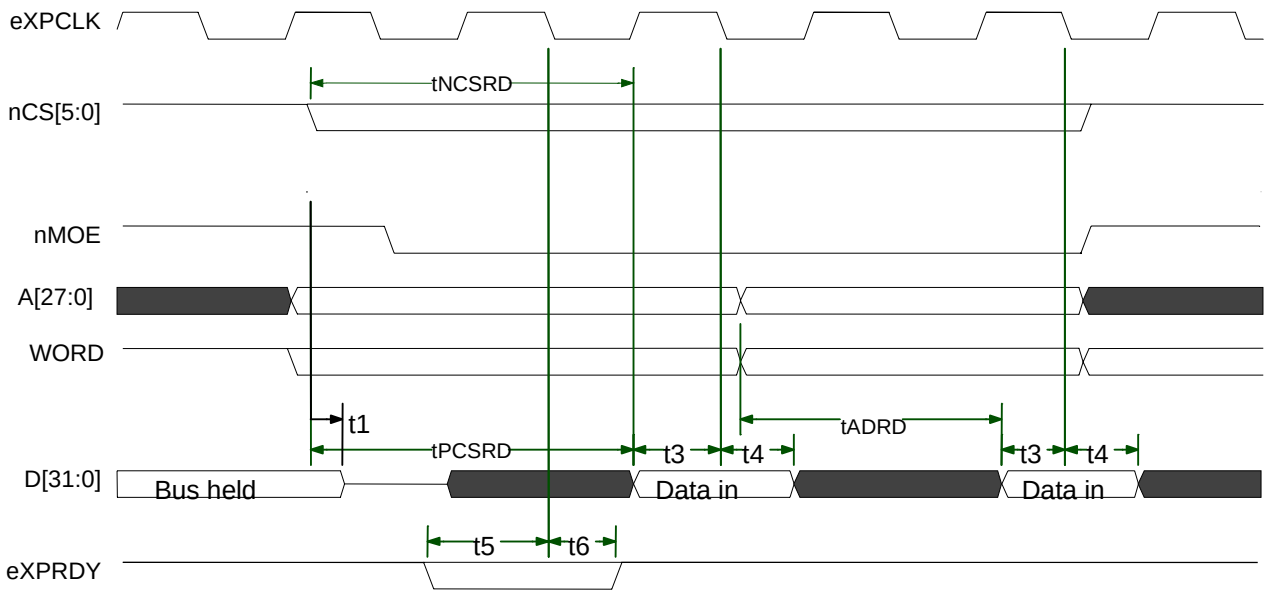
Parameter	Symbol	18/36 MHz		Units
		Min	Max	
Falling CS to data bus Hi-Z	t1	0	25	ns
Address change to valid write data	t2	0	35	ns
DATA in to falling EXPCLK setup time	t3	18	—	ns
DATA in to falling EXPCLK hold time	t4	0	—	ns
EXPRDY to falling EXPCLK setup time	t5	18	—	ns
Falling EXPCLK to EXPRDY hold time	t6	0	50	ns
Rising nMWE to data invalid hold time	t7	5	—	ns
Sequential data valid to falling nMWE setup time	t8	–10	10	ns
Row address to falling nRAS setup time	t9	5	-	ns
Falling nRAS to row address hold time	t10	25	-	ns
Column address to falling nCAS setup time	t11	2	-	ns
Falling nCAS to column address hold time	t12	25	-	ns
Write data valid to falling nCAS setup time	t13	2	-	ns
Write data valid from falling nCAS hold time	t14	50	-	ns
LCD CL2 low time	t15	80	3,475	ns
LCD CL2 high time	t16	80	3,475	ns
LCD falling CL[2] to rising CL[1] delay	t17	0	25	ns
LCD falling CL[1] to rising CL[2]	t18	80	3,475	ns
LCD CL[1] high time	t19	80	3,475	ns
LCD falling CL[1] to falling CL[2]	t20	200	6,950	ns
LCD falling CL[1] to FRM toggle	t21	300	10,425	ns
LCD falling CL[1] to M toggle	t22	–10	20	ns
LCD rising CL[2] to display data change	t23	–10	20	ns
Falling EXPCLK to address valid	t24	—	5	ns
Data valid to falling nMWE for non sequential access only	t25	5	—	ns
SSICLK period (slave mode)	t31	0	512	kHz
SSICLK high	t32	925	1025	ns
SSICLK low	t33	925	1025	ns
SSICLK rise / fall time	t34		7	ns
SSICLK rising to RX and / or TX frame sync	t35		528	ns
SSICLK rising edge to frame sync low	t36		448	ns
SSICLK rising edge to TX data valid	t37		80	ns
SSIRXDA data set-up time	t38	30		ns
SSIRXDA data hold time	t39	40		ns
SSITXFR and / or SSIRXFR period	t40	750		ns

Note: All SDRAM 36 MHz timings are for SDRAM operation.
The values for 36 MHz include 1 wait state, the 18 MHz values have 0 wait states.

TIMING CHARACTERISTICS

Characteristics	Symbol	18 MHz		36 MHz		Units
		Min	Max	Min	Max	
Negative strobe (nCS[0:5]) zero wait state read access time	t_{nCSRD}	TBD		TBD		TBD
Negative strobe (nCS[0:5]) zero wait state write access time	t_{nCSWR}	TBD		TBD		TBD
Sequential expansion burst mode read access time	t_{EXBST}	TBD		TBD		TBD
SDRAM cycle time	t_{RC}	TBD	-	TBD		TBD
Access time from RAS	t_{RAC}	TBD	-	TBD		TBD
RAS precharge time	t_{RP}	TBD	-	TBD		TBD
CAS pulse width	t_{CAS}	TBD	-	TBD		TBD
CAS precharge in page mode	t_{CP}	TBD	-	TBD		TBD
Page mode cycle time	t_{PC}	TBD	-	TBD		TBD
CAS set-up time for auto refresh	t_{CSR}	TBD	-	TBD		TBD
RAS pulse width	t_{RAS}	TBD	-	TBD		TBD

Note: All SDRAM 36 MHz timings are for SDRAM operation.
The values for 36 MHz assume 1 wait state, the 18 MHz values have 0 wait states.



Notes: 1. tnCSRD = 50 ns at 36.864 MHz
70 ns at 18.432 MHz

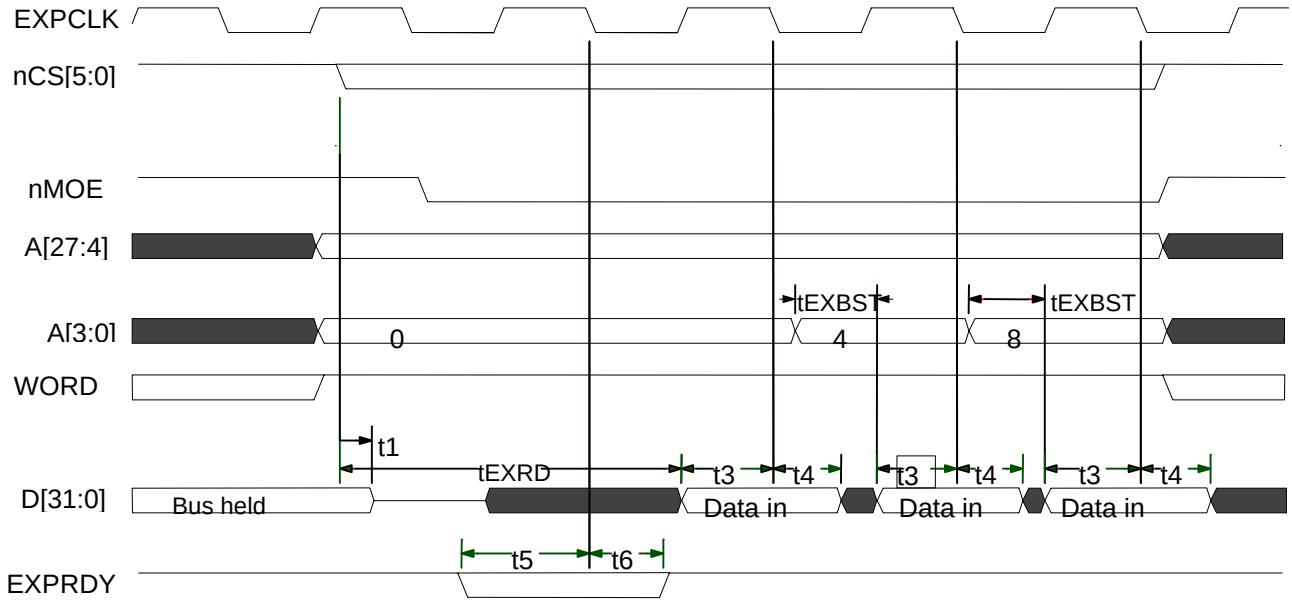
Maximum values for minimum wait states. This time can be extended by integer multiples of the clock period (27 ns at 36 MHz or 54 ns at 18.432 MHz), by either driving EXPRDY low and/or by programming a number of wait states. EXPRDY is sampled on the falling edge of EXPCLK before the data transfer. If low at this point, the transfer is delayed by one clock period where EXPRDY is sampled again. EXPCLK need not be referenced when driving EXPRDY, but is shown for clarity.

2. Consecutive reads with sequential access enabled are identical except that the sequential access wait state field is used to determine the number of wait states, and no idle cycles are inserted between successive non-sequential ROM/expansion cycles. This improves performance so the SQAEN bit should always be set where possible.

3. tnCSRD = tADRD = tPCSRD

4. When the CS89712 device implements consecutive reads(e.g., use of the LDM instruction), regardless of the state of the SQAEN bit, the signals nMOE and nCSx will always remain low through the entire multi-read access. They will not toggle in-between each different address access. In order to have these signals toggle, single access read instructions (e.g., LDR) must be used.

Figure 24. Consecutive Memory Read Cycles with Minimum Wait States

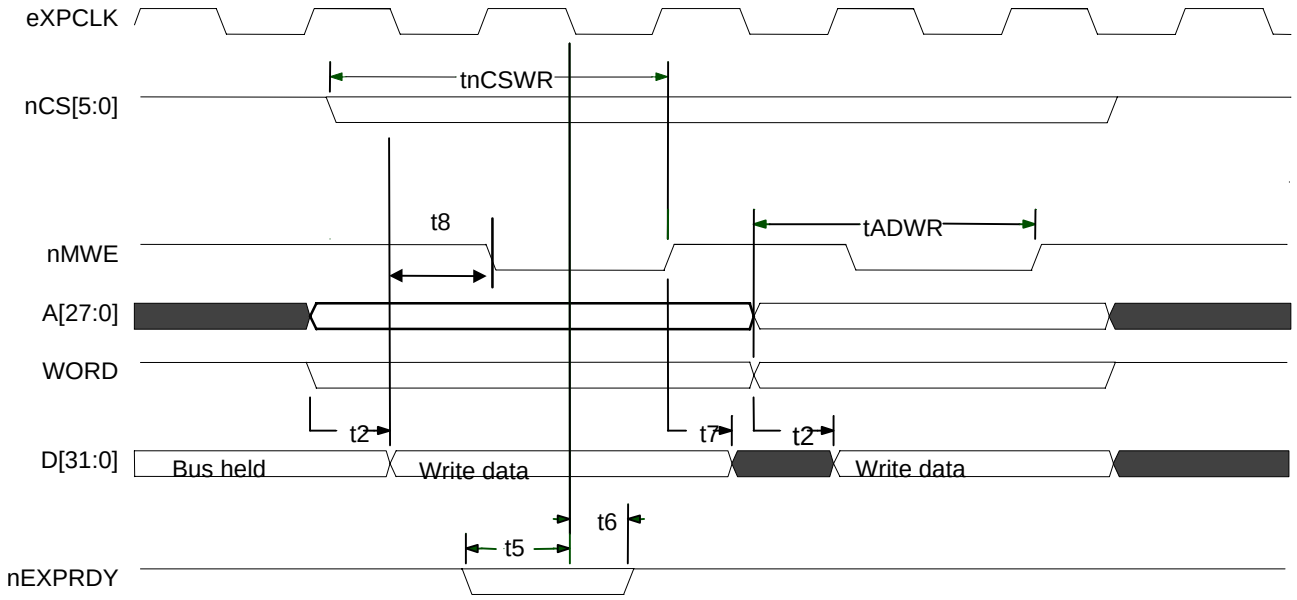


Notes: 1. tEXBST = 35 ns at 36.864 MHz
35 ns at 18.432 MHz
(Value for 36.864 MHz assumes 1 wait state.)

Maximum values for minimum wait states. This time can be extended by integer multiples of the clock period (27 nsec at 36 MHz or 54 nsec at 18.432 MHz), by either driving EXPRDY low and/or by programming a number of wait states. EXPRDY is sampled on the falling edge of EXPCLK before the data transfer. If low at this point, the transfer is delayed by one clock period where EXPRDY is sampled again. EXPCLK need not be referenced when driving EXPRDY, but is shown for clarity.

2. Consecutive reads with sequential access enabled are identical except that the sequential access wait state field is used to determine the number of wait states, and no idle cycles are inserted between successive non-sequential ROM/expansion cycles. This improves performance so the SQAEN bit should always be set where possible.

Figure 25. Sequential Page Mode Read Cycles with Minimum Wait States



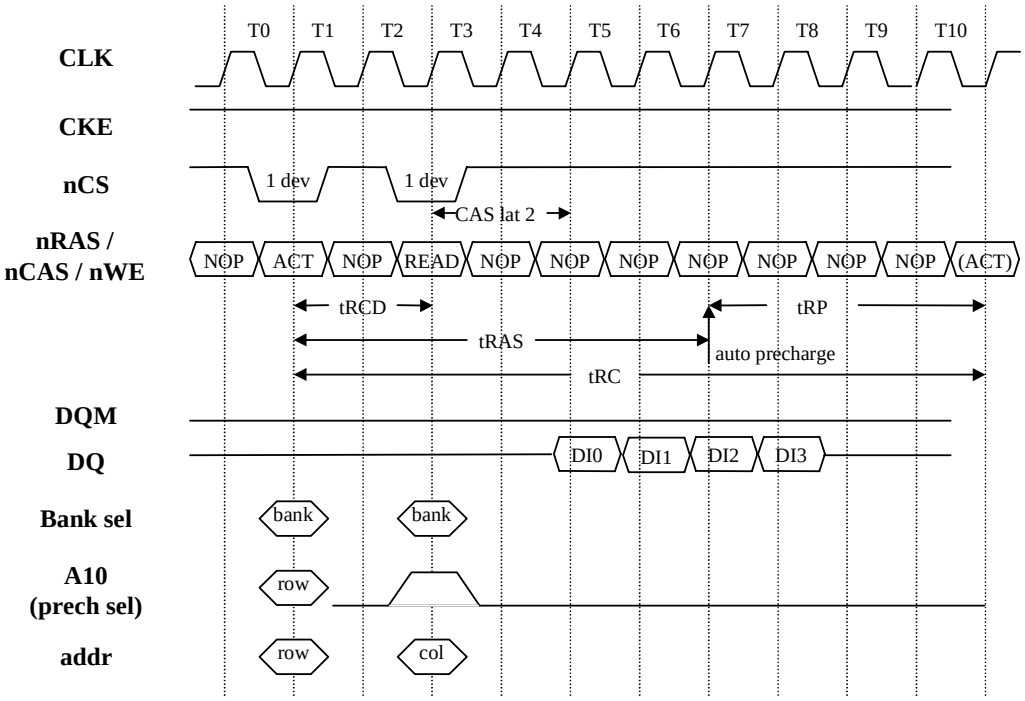
Notes: 1. tnCSWR = 35 nsec at 36.864 MHz, 70 ns at 18.432 MHz

Maximum values for minimum wait states. This time can be extended by integer multiples of the clock period (27 nsec at 36 MHz or 54 nsec at 18.432 MHz), by either driving EXPRDY low and/or by programming a number of wait states. EXPRDY is sampled on the falling edge of EXPCLK before the data transfer. If low at this point, the transfer is delayed by one clock period where EXPRDY is sampled again. EXPCLK need not be referenced when driving EXPRDY, but is shown for clarity.

2. Consecutive reads with sequential access enabled are identical except that the sequential access wait state field is used to determine the number of wait states, and no idle cycles are inserted between successive non-sequential ROM/expansion cycles. This improves performance so the SQAEN bit should always be set where possible.

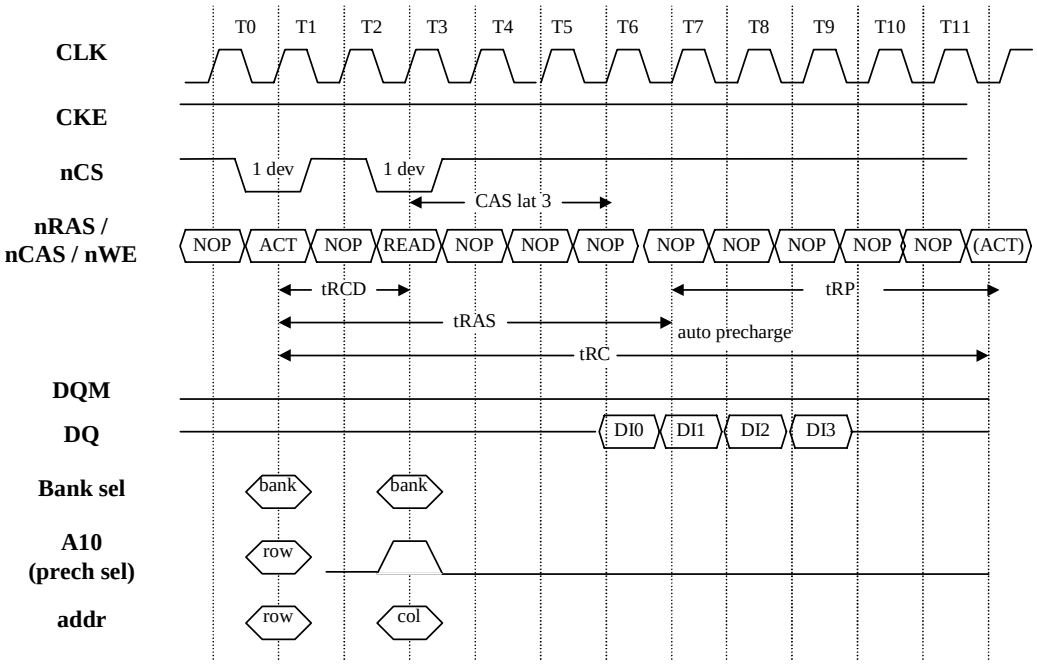
3. Zero wait states for sequential writes is not permitted for memory devices which use nMWE pin, as this cannot be driven with valid timing under zero wait state conditions.

Figure 26. Consecutive Memory Write Cycles with Minimum Wait States



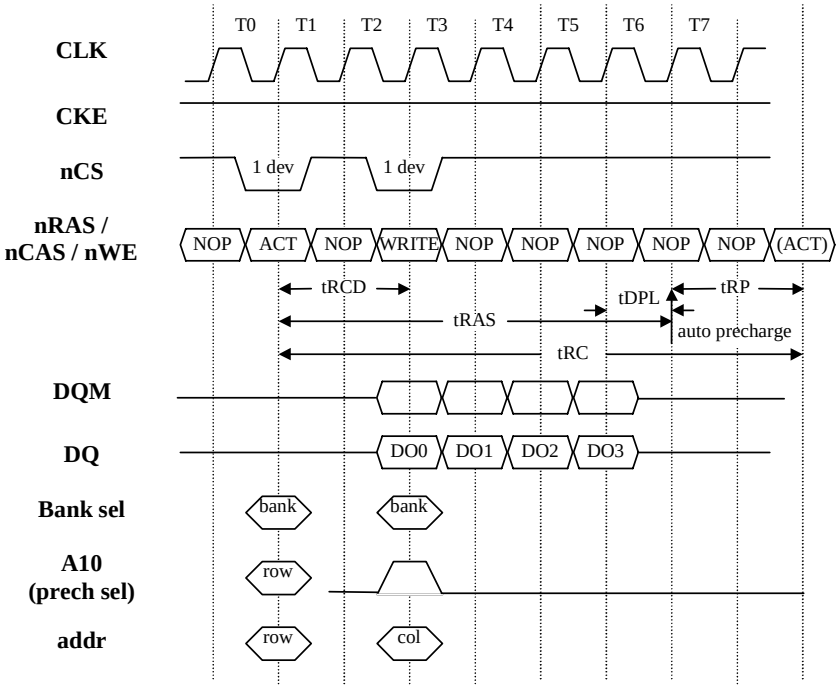
1. tRCD (delay time ACT to READ/WRITE command) = 30 ns or 2 cycles at 36 MHz.
2. tRP (PRE to ACT command period) = 30 ns or 2 cycles at 36 MHz.
3. tRAS (ACT to PRE command period) = 60 ns or 3 cycles at 36 MHz.
4. tRC (ACT to REF/ACT command period [operation]) = 90 ns or 4 cycles at 36 MHz.
5. For SDCAS latency 3, there will be an extra cycle between T4 and T5.
6. For CAS latency 3, there will be an extra cycle between T4 and T5.

Figure 27. SDRAM Read Cycles CAS Latency = 2



Note: tRC1 (REF to REF/ACT command period [refresh]) = 90 ns or 4 cycles at 36 MHz.

Figure 28. SDRAM Read Cycles CAS Latency = 3



Note: tDPL (data in to PRE command period command) = 10 ns or 1 cycle at 36 MHz.

Figure 29. SDRAM Write Cycles

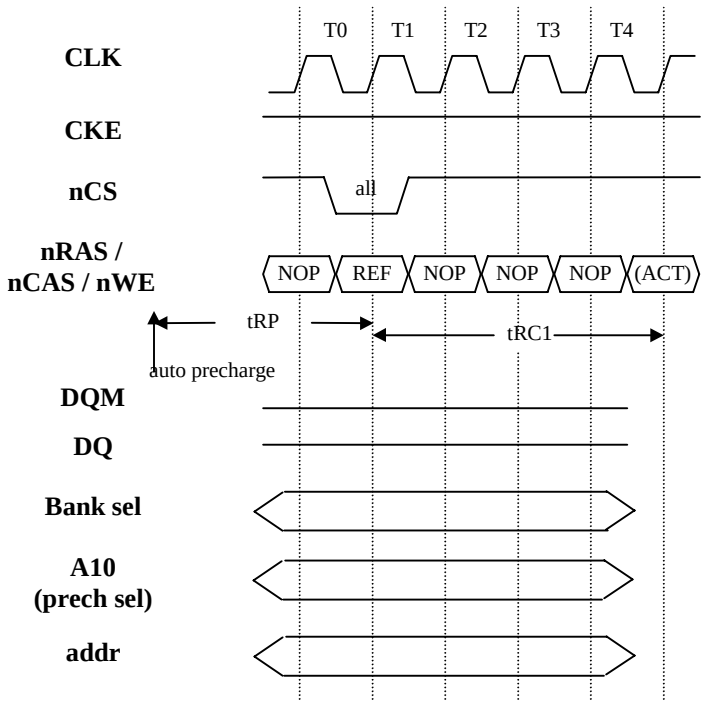
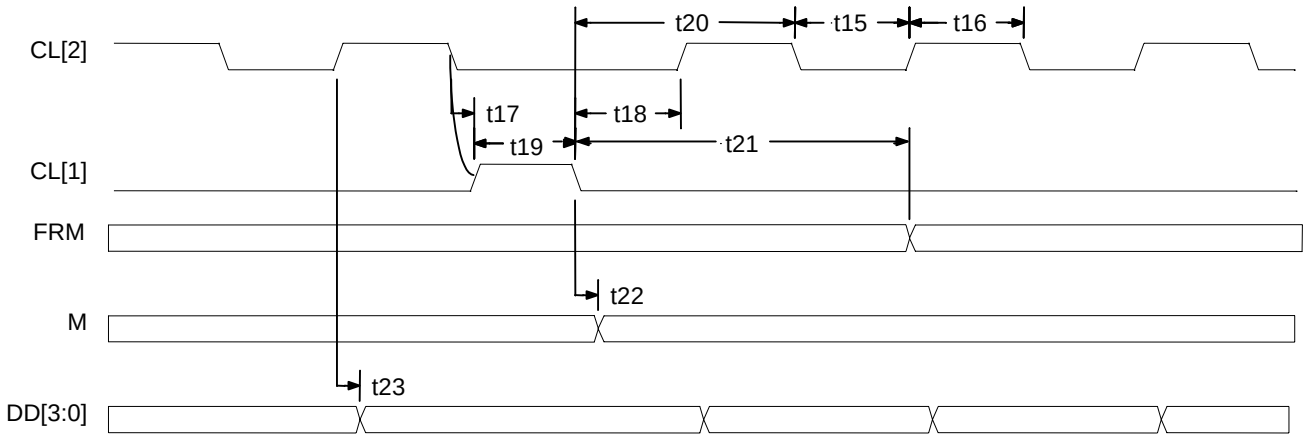


Figure 30. SDRAM Refresh Cycles



Notes: 1. The figure shows the end of a line.
2. If FRM is high during the CL[1] pulse, this marks the first line in the display.
3. CL[2] low time is doubled during the CL[1] high pulse

Figure 31. LCD Controller Timings

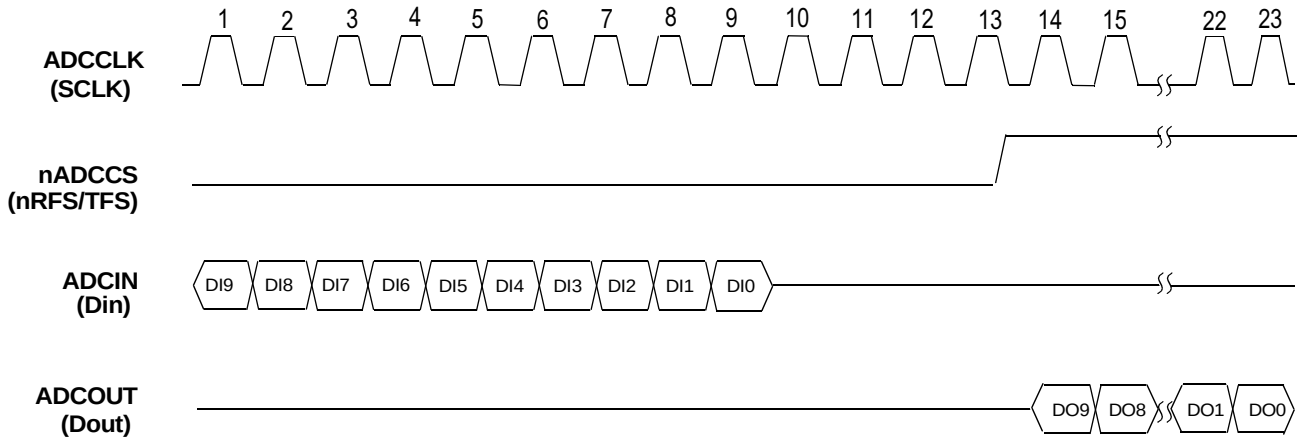


Figure 32. SSI1 Interface for AD7811/2

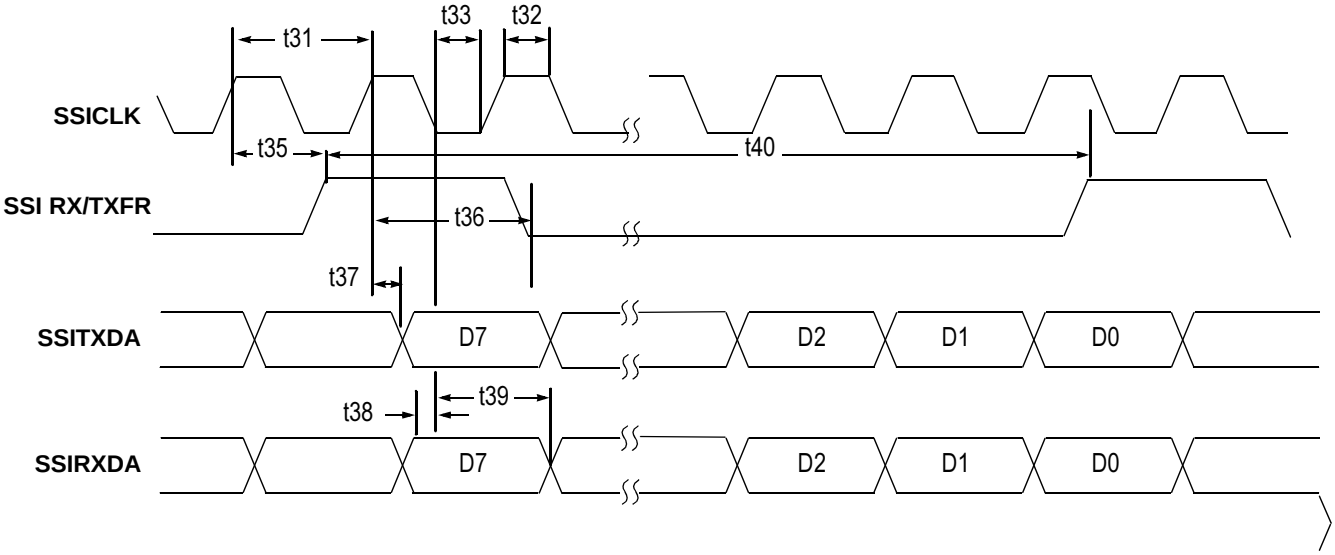


Figure 33. SSI2 Interface Timings

ETHERNET TIMING CHARACTERISTICS

Parameter	Symbol	Min	Typ	Max	Unit
10BASE-T Receive					
Allowable Received Jitter at Bit Cell Center	t_{TRX1}	-	-	±13.5	ns
Allowable Received Jitter at Bit Cell Boundary	t_{TRX2}	-	-	±13.5	ns
Carrier Sense Assertion Delay	t_{TRX3}	-	540	-	ns
Invalid Preamble Bits after Assertion of Carrier Sense	t_{TRX4}	1	-	2	bits
Carrier Sense Deassertion Delay	t_{TRX5}	-	270	-	ns
10BASE-T Link Integrity					
First Transmitted Link Pulse after Last Transmitted Packet	t_{LN1}	8	16	24	ms
Time Between Transmitted Link Pulses	t_{LN2}	8	16	24	ms
Width of Transmitted Link Pulses	t_{LN3}	60	100	200	ns
Minimum Received Link Pulse Separation	t_{LN4}	2	-	7	ms
Maximum Received Link Pulse Separation	t_{LN5}	25	-	150	ms
Last Receive Activity to Link Fail (Link Loss Timer)	t_{LN6}	50	-	150	ms
Ethernet EEPROM					
EESK Setup time relative to EECS	t_{SKS}	100	-	-	ns
EECS Setup time wrt ↑ EESK	t_{CCS}	250	-	-	ns
EEDataOut Setup time wrt ↑ EESK	t_{DIS}	250	-	-	ns
EEDataOut Hold time wrt ↑ EESK	t_{DIH}	500	-	-	ns
EEDataIn Hold time wrt ↑ EESK	t_{DH}	10	-	-	ns
EECS Hold time wrt ↓ EESK	t_{CSH}	100	-	-	ns
Min EECS Low time during programming	t_{CS}	1000	-	-	ns

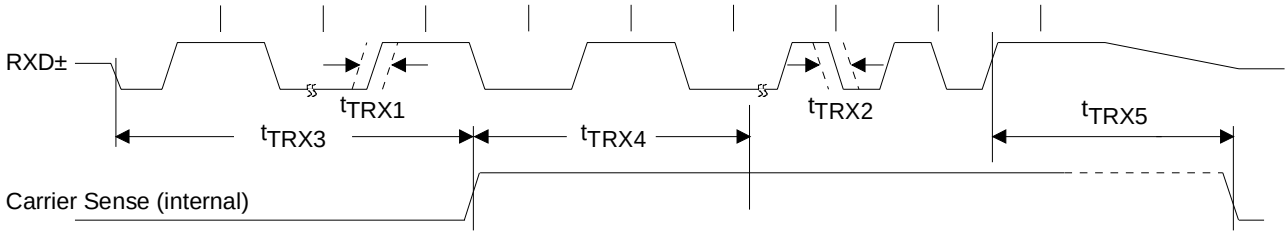


Figure 34. 10BASE-T Receive

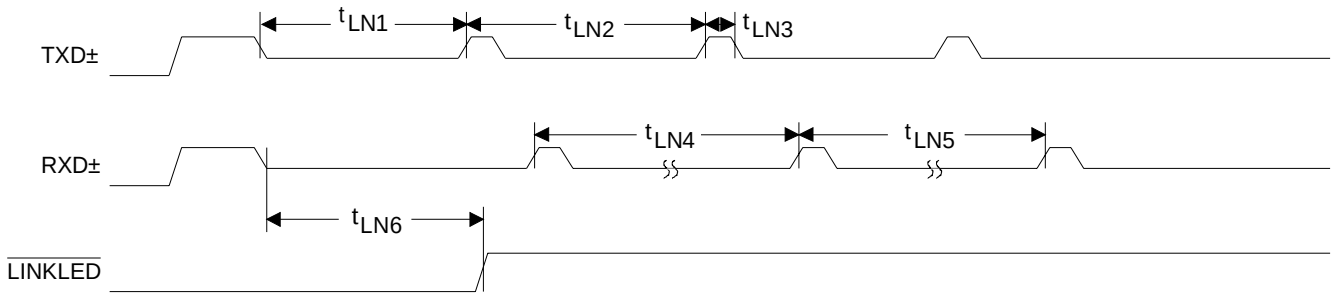


Figure 35. 10BASE-T Link Integrity

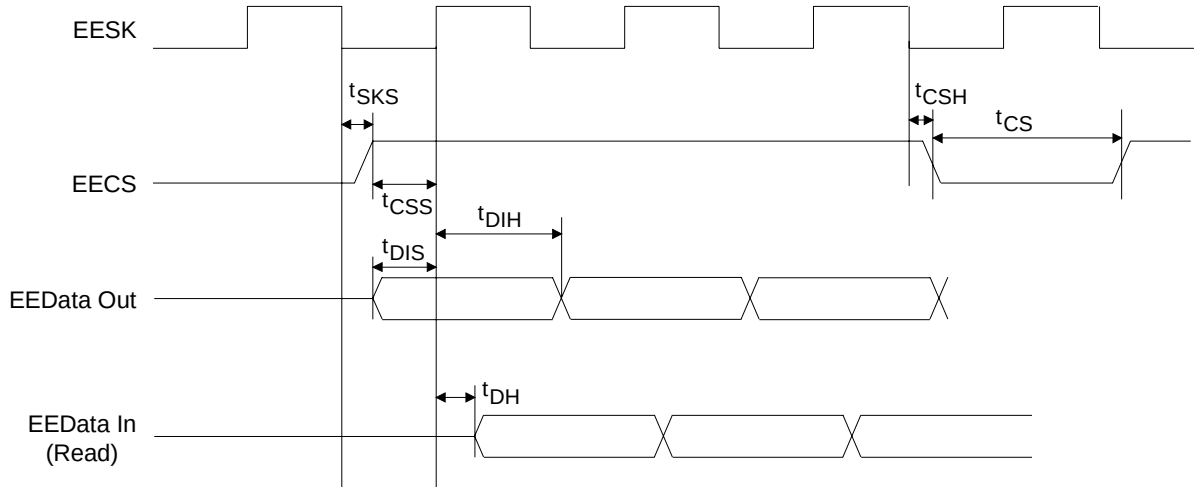
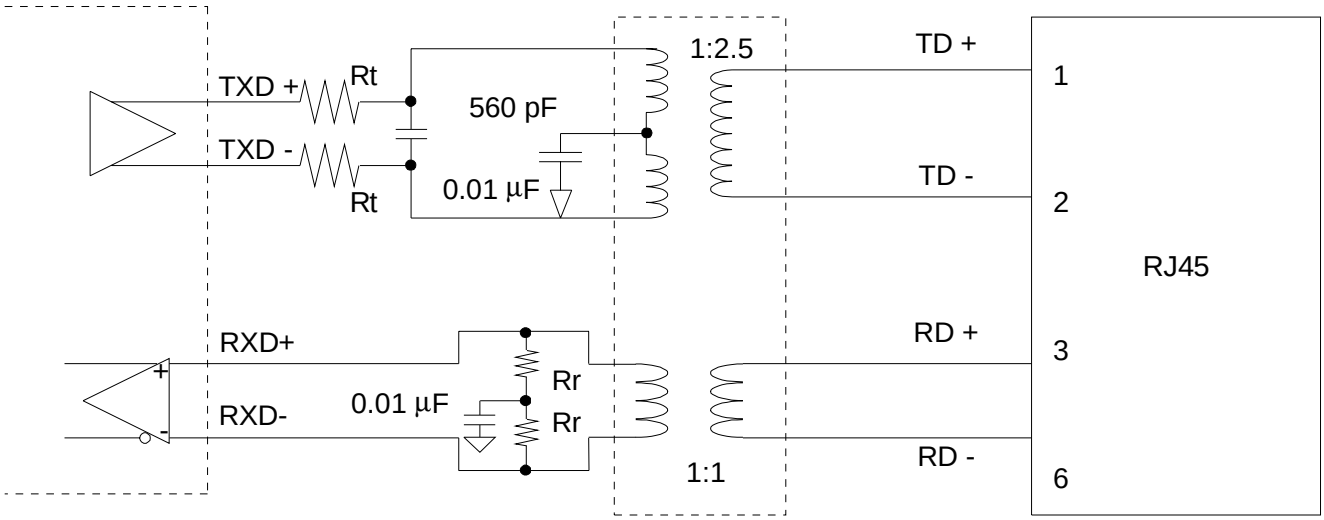


Figure 36. EEPROM

ETHERNET QUARTZ CRYSTAL REQUIREMENTS (If a 20 MHz quartz crystal is used, it must meet the following specifications)

Parameter	Min	Typ	Max	Unit
Parallel Resonant Frequency	-	20	-	MHz
Resonant Frequency Error (C_L = 18 pF)	-50	-	+50	ppm
Resonant Frequency Change Over Operating Temperature	-40	-	+40	ppm
Crystal Capacitance	-	-	18	pF
Motional Crystal Capacitance	-	0.022	-	pF
Series Resistance	-	-	50	Ohm
Shunt Capacitance	-	-	7	pF



Notes:
1. If a center tap transformer is used on the RXD+ and RXD- inputs, replace the pair of Rr resistors with a single 2xRr resistor.
2. The Rt and Rr resistors are ±1% tolerance.
3. The CS89712 Ethernet port supports 100 W unshielded twisted pair cables. The proper values of Rt and Rr, for a given cable impedance, are shown below:

Cable Impedance (Ω)	Rt (Ω)	Rr (Ω)
100	8.2	49.9

Figure 37. 10Base-T Wiring

6.4 I/O Buffer Strength & Characteristics

All I/O buffers on the CS89712 are CMOS threshold input bidirectional buffers except the oscillator and power pads. For signals that are nominally inputs, the output buffer is only enabled during pin test mode. All output buffers are three stated during system (hi-Z) test mode. All buffers have a standard CMOS threshold input stage (apart from the Schmitt-triggered inputs) and CMOS slew-rate-controlled output stages to reduce system noise. Table 92 defines the I/O buffer output characteristics which will apply across the full range of temperature and voltage (i.e., these values are for 3.3 V, +70° C).

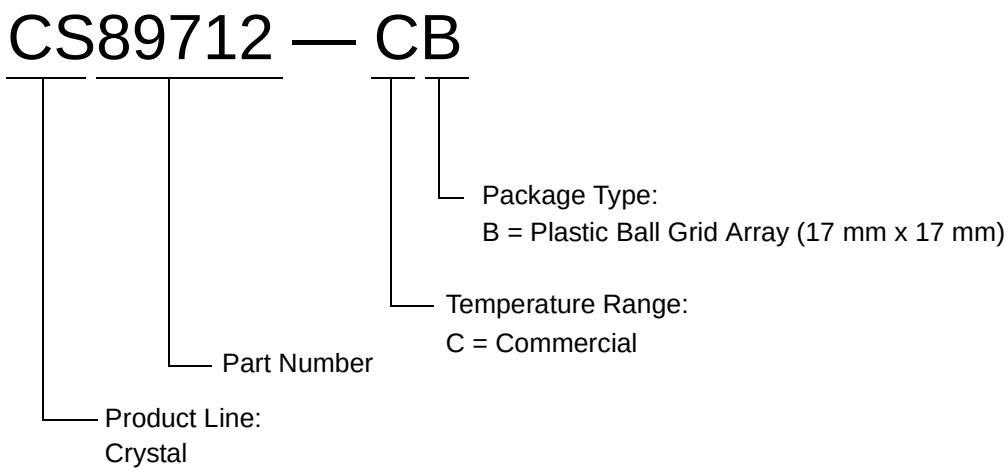
All propagation delays are specified at 50% V_{DD} to 50% V_{DD}, all rise times are specified as 10% V_{DD} to 90% V_{DD} and all fall times are specified as 90% V_{DD} to 10% V_{DD}.

Buffer Type	Drive Current	Propagation Delay (Max)	Rise Time (Max)	Fall Time (Max)	Load
Address outputs	±12 mA	5	6	6	50 pF
Non-address outputs	±4 mA	7	14	14	50 pF

Table 92. I/O Buffer Output Characteristics

7. ORDERING INFORMATION

The order number for the device is:



Note: Contact Cirrus Logic for up-to-date information on revisions. Go to the Cirrus Logic Internet site at *http://cirrus.com/corporate/contacts* to find contact information for your local sales representative.

Appendix A - Terms

Acronyms, abbreviations, units of measurement, and conventions used in this document.

Acronyms and Abbreviations

Table 93 lists abbreviations and acronyms.

Acronym	Definition
A/D	Analog-to-Digital.
ADC	Analog-to-Digital Converter.
BufCFG	Buffer Configuration.
BufEvent	Buffer Event.
BusCTL	Bus Control.
BusST	Bus State.
CODEC	Coder / Decoder.
CRC	Cyclic Redundancy Check.
CS	Carrier Sense.
CSMA/CD	Carrier Sense Multiple Access with Collision Detection.
DA	Destination Address.
D/A	Digital-to-Analog.
ENDEC	Manchester Encoder/Decoder.
EOF	End-of-Frame.
EPB	Embedded Peripheral Bus.
FCS	Frame Check Sequence.
FDX	Full Duplex.
GPIO	General Purpose I/O.
IA	Individual Address.
ICT	In Circuit Test.
IPG	Inter-Packet Gap.
IR	Infrared.
IrDA	Infrared Data Association.
ISQ	Interrupt Status Queue.
JTAG	Joint Test Action Group.
Line CTL	Ethernet Line Control.

Table 93. Acronyms and Abbreviations

Acronym	Definition
LineST	Ethernet Line Status.
LLC	Logical Link Control.
LQFP	Low Profile Quad Flat Pack.
MAC	Media Access Control.
MAU	Medium Attachment Unit.
MIB	Management Information Base.
MMU	Memory Management Unit.
PBGA	Plastic Ball Grid Array.
PDA	Personal Digital Assistant.
PIA	Peripheral Interface Adapter.
PLL	Phase Locked Loop.
PSU	Power Supply Unit.
p/u	Pull-Up Resistor.
RTC	Real-Time Clock.
RxCFG	Receive Configuration.
RxCTL	Receive Control.
RxEvent	Receive Event.
SelfCTL	Self Control.
SA	Source or System Address.
SelfST	Self Status.
SFD	Start-of-Frame Delimiter.
SIR	Slow (9.6 – 115.2 kbps) Infrared.
SNMP	Simple Network Management Protocol.
SOF	Start-of-Frame.
SQE	Signal Quality Error.
SSI	Synchronous Serial Interface.
TAP	Test Access Port.
TestCTL	Test Control.
TLB	Translation Lookaside Buffer.
TxCFG	Transmit Configuration.
TxCMD	Transmit Command.
TxEvent	Transmit Event.
UTP	Unshielded Twisted Pair.

Table 93. Acronyms and Abbreviations (Continued)

Units of Measurement

Symbol	Unit of Measure
°C	degree Celsius
Hz	hertz (cycle per second)
kbits/s	kilobits per second
kbyte	kilobyte (1,024 bytes)
kHz	kilohertz
kΩ	kilohm
Mbps	megabits (1,048,576 bits) per second
Mbyte	megabyte (1,048,576 bytes)
MHz	megahertz (1,000 kilohertz)
µA	microampere
µF	microfarad
µW	microwatt
µs	microsecond (1,000 nanoseconds)
mA	milliampere
mW	milliwatt
ms	millisecond (1,000 microseconds)
ns	nanosecond
V	volt
W	watt

Table 94. Units of Measurement

General Conventions

Hexadecimal numbers are presented with all letters in uppercase and a lowercase “h” appended or with a 0x at the beginning, for example, 0x14 and 03CAh. Binary numbers are enclosed in single quotation marks when in text (for example, ‘11’ designates a binary number). Numbers not indicated by an “h”, 0x or quotation marks are decimal.

Registers are referred to by acronym, as listed in the tables on the previous page, with bits listed in brackets MSB-to-LSB separated by a colon (:) (for example, CODR[7:0]), or LSB-to-MSB separated by a hyphen (for example, CODR[0–2]).

“TBD” indicates values “to be determined,” “n/a” designates “not available”, “n/c” indicates a “no connect. pin” and “dc” means “don’t care”.

Ethernet port Definitions

- **Act-Once bit**

  Causes the Ethernet port to take a certain action once when the bit is set. To cause the action again, the software must rewrite a "1".

- **Committed Receive Frame**

  A receive frame is "committed" after the frame has been buffered by the Ethernet port, and a receive interrupt has been generated.

- **Committed Transmit Frame**

  A transmit frame is "committed" after software has issued a Transmit Command, and the Ethernet port has reserved buffer space and notified the software that it is ready for transmit.

- **Cyclic Redundancy Check**

  The method used to compute the 32-bit frame check sequence.

- **Event or Interrupt Event**

  Refers to something that can trigger an interrupt. Items that are considered "Events" are reported in the three Event registers (RxEvent, TxEvent, or BufEvent) and in two counter-overflow bits (RxMISS and TxCOL).

- **Frame**

  The portion of a packet from the DA to the FCS. This includes the Destination Address (DA), Source Address (SA), Length field, Data field, pad bits (if necessary), and Frame Check Sequence (FCS, also called CRC). "Frame data" refers to all data from the DA to the FCS to be transmitted, or that has been received.

- **Frame Check Sequence**

  The 32-bit field at the end of a frame that contains the cyclic redundancy check result.

- **Individual Address**

  The specific Ethernet address assigned to a device attached to the Ethernet media.

- **Inter-Packet Gap**

  Time interval between packets on the Ethernet. Minimum interval is 9.6 ms.

- **Jabber**

  A condition that results when an Ethernet node transmits longer than between 20 ms and 150 ms.

- **Packet**

  The entire serial string of bits transmitted over an Ethernet network. This includes the preamble, Start-of-Frame Delimiter, Destination Address, Source Address, Length field, Data field, pad bits (if necessary), and Frame Check Sequence (FCS, also called CRC). A packet is a frame plus the Preamble and SFD.

- **Slot Time**

  Time required for an Ethernet Frame to cross a maximum length Ethernet network. One Slot Time equals 512 bit times.

- **Suspend**

  Used to conserve power. When in Suspend mode, the Ethernet port can be awakened only by software command.

- **Transfer**

  Moving frame data across the memory bus from the Ethernet port RAM to system RAM. During receive operations, only frame data are transferred (preamble and SFD are stripped off by the CS89712's MAC engine). The FCS may or may not be transferred, depending on the configuration. Transfers are counted in bytes.

- **Transmit Collision**

  The receive inputs, RXD+/RXD- (10BASE-T) are active while a packet is being transmitted.

- **Transmit Request**

  A Transmit Request is issued by the software to initiate the start of a new packet transmission.

Suffixes Specific to the Ethernet port

These have meaning only at the end of a term:

A	Accept
CMD	Command
CFG	Configure
CTL	Control
Dis	Disable
E	Enable
h	Indicates the number is hexadecimal
iE	Interrupt Enable
ST	Status

Appendix B: Boot Code

```
00000000        uart_boot_base
00000000 E3A0C102       MOV     r12, #HwRegisterBase ; R12 = 0x80000000
00000004
00000004 E3A08201       MOV     r8, #InternalRamBase ; R8 = 0x10000000
00000008 E2889B02       ADD     r9, r8, #ImageSize ; R9 = 0x10000800
0000000C
0000000C        ;;; The remaining code is functionally identical to the 7111 boot code
0000000C
0000000C        ;;; First, initialize HW control of UART
0000000C
0000000C 00000480 Hw_UARTDR1   EQU     0x0480
0000000C
0000000C 000004C0 Hw_UBRLCR1   EQU     0x04c0
0000000C 00000017 Hw_BR9600    EQU     0x00000017  ; 9600 baud divisor = 23
0000000C 0000000B Hw_BR9600_13 EQU     0x0000000b  ; 9600 baud divisor = 11
0000000C 00060000 Hw_WRDLEN8   EQU     0x00060000
0000000C
0000000C E3A00C01       MOV     r0, #Hw_UART1EN ; Enable UART
00000010 E58C0100       STR     r0, [r12, #Hw_SYSCON]
00000014
00000014 E28C1D45       ADD     r1, r12, #Hw_SYSFLG2 ; (was LDR, ADD in 7111 code)
00000018 E5917000       LDR     r7, [r1]    ; R7 = SYSFLG2
0000001C
0000001C E3170040       TST     r7, #Hw_CKMODE
00000020 13A0000B       MOVNE   r0, #Hw_BR9600_13 ; Load 13 MhZ value if bit set
00000024 03A00017       MOVEQ   r0, #Hw_BR9600 ; If not set, load other divisor
00000028 E3800806       ORR     r0, r0, #Hw_WRDLEN8 ; Insert 8-bit character mode
0000002C
0000002C E58C04C0       STR     r0, [r12, #Hw_UBRLCR1]
00000030
00000030 0000003C StartFlag EQU     '<'
00000030 0000003E EndFlag EQU       '>'
00000030
00000030        ;;; Send ready signal
00000030 E3A0003C       MOV     r0, #StartFlag
00000034 E58C0480       STR     r0, [r12, #Hw_UARTDR1]
00000038
00000038        ;;; Receive the data
00000038        ;;; Store bytes at R9 address, stop loop when R8 == R9
00000038        ;;; Leaves R8 set to 0x10000800
00000038
00000038        ;;; Wait for byte to be available
00000038
00000038        uart_ready_loop
00000038 E59C1140       LDR     r1, [r12, #Hw_SYSFLG] ; Spin, if Rx FIFO is empty
0000003C E3110501       TST     r1, #Hw_URXFE1
00000040 1AFFFFFC       BNE     uart_ready_loop
00000044
00000044        ;;; Read the data, store it, and accumulate checksum
00000044 E59C0480       LDR     r0, [r12, #Hw_UARTDR1] ; Read data
00000048 E4C80001       STRB    r0, [r8], #1 ; Save it in memory
```

```
0000004C E1580009      CMP     r8, r9
00000050 BAFFFFF8      BLT     uart_ready_loop ; Do more if end of buffer not reached
00000054
00000054        ;;; All received, send end flag
00000054
00000054 E3A0003E      MOV     r0, #EndFlag
00000058 E5CC0480      STRB    r0, [r12, #Hw_UARTDR1] ; Send reply
0000005C
0000005C
0000005C
0000005C        ;;; Having loaded all the bytes, do the right thing to finish.
0000005C        ;;;
0000005C
0000005C E55807FD      LDRB    r0, [r8, #(3-ImageSize)]
00000060 E35000FF      CMP     r0, #BootImageFlagByte
00000064
00000064
00000064 01A0F00E      MOVEQ   pc, r14     ; Return to caller for secure image
00000068
00000068
00000068
00000068
00000068 E28CAB09      ADD     r10, r12, #WWWWWWWWWW ; R10 = 0x80002400 (also XXXXXX)
0000006C E58AC080      STR     r12, [r10, #(ZZZZZZZZZZ - YYYYYYYYYY)]
00000070 E248FB02      SUB     pc, r8, #ImageSize ; Branch to 0x10000000
00000074
00000074
00000074        ;;; Put a checksum here so this part can be verified, too.
00000074        ;;; Have to pad the tail out to 31 words, then the checksum.
00000074
00000074 0000000000    ALIGN   128, -4     ; Align just before end of 128-byte tail
0000007C        uart_checksum
0000007C 436B74AB      DCD     0x436b74ab
00000080
00000080                ASSERT  (. - start_of_rom) = 640 ; Check that it's in the right place
00000080
00000080                END
```

• Notes •

SMART Analog™