CIRRUS LOGIC®

EP7209

Preliminary Data Sheet

FEATURES

- **Audio decoder system-on-chip**
  - — Allows for support of multiple audio decompression algorithms
  - — Supports MPEG 1, 2, & 2.5 layer 3 audio decoding, including ISO compliant MPEG 1 & 2 layer 3 support for all standard sample rates and bit rates
  - — Supports bit streams with adaptive bit rates
  - — DAI (Digital Audio Interface) providing glueless interface to low power DACs, ADCs, and Codecs
- **Ultra low power consumption for MP3 playback**
  - — 87 mW (typical) for 44.1 kHz samples/sec, 128 kbits/second
  - — 50 mW for 22.05 kHz samples/sec, 64 kbits/second
  - — <1 mW in Standby State
- **ARM720T processor**
  - — ARM7TDMI CPU
  - — 8 k-bytes of four-way set-associative cache
  - — MMU with 64-entry TLB (transition look-aside buffer)
  - — Write Buffer
  - — Windows® CE enabled
  - — Thumb code support enabled

(cont.)

Ultra-Low-Power Audio Decoder System-on-Chip

OVERVIEW

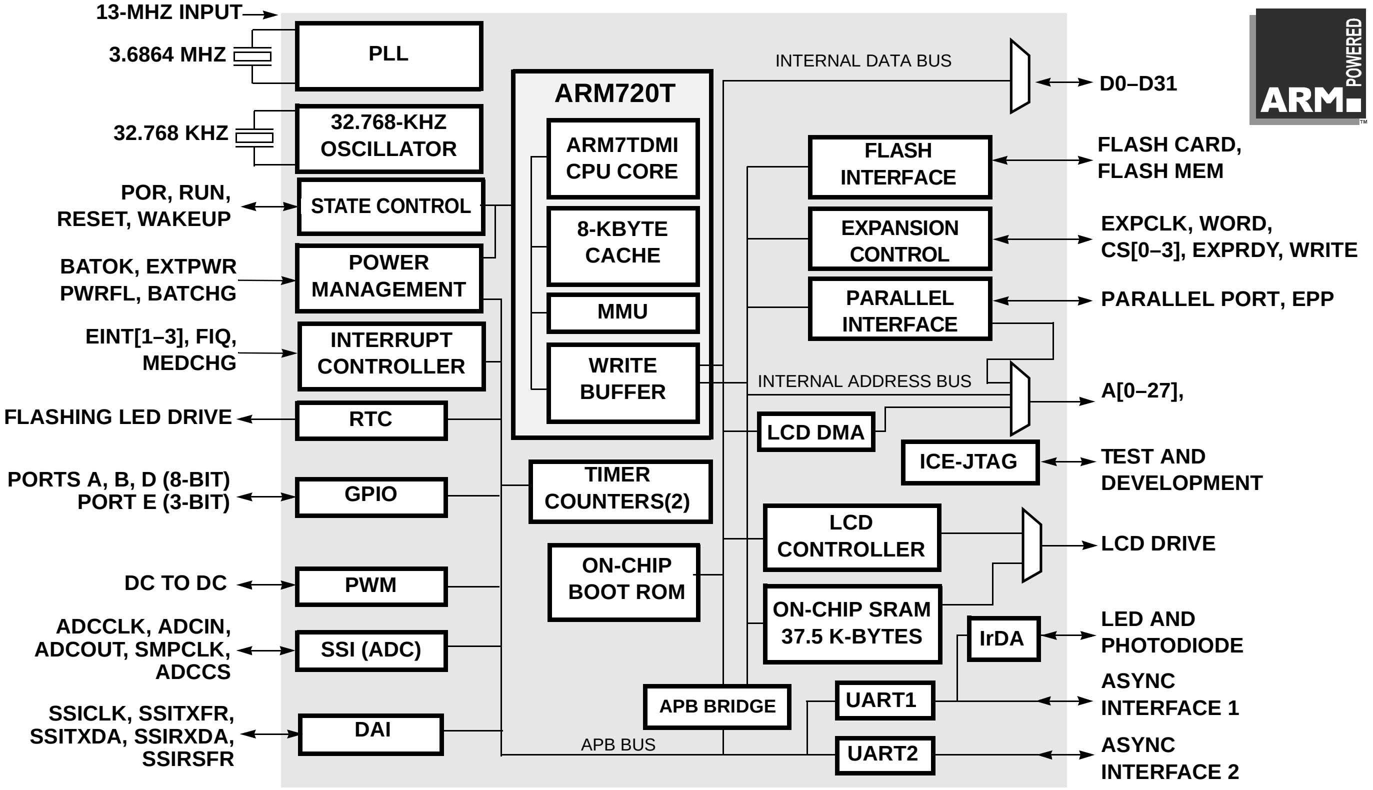
The EP7209 is a complete integrated system on a chip for enabling personal digital audio solutions. It is designed specifically for implementing audio processing algorithms in power sensitive applications. The core-logic functionality of the device is built around an ARM720T embedded processor.

The EP7209 also includes a 32-bit Y2K-compliant Real-Time Clock (RTC) and comparator.

The EP7209 also includes a comprehensive set of integrated peripherals such as an LCD display con-

(cont.)

Functional Block Diagram



FEATURES *(cont.)*

- **Dynamically programmable clock speeds of 18, 36, 49, and 74 MHz at 2.5 V**
- **Performance matching 100 MHz Intel® Pentium-based PC**
- **OEM Customization**
  - — Integrated ARM720T RISC processor
  - — Up to 25 MHz of CPU processing power available for custom features such as software EQ or tone control, volume control, spectrum analyzer, random play order, etc.
  - — Allows for control of digital voice recorder function
- **LCD controller**
  - — Interfaces directly to a single-scan panel monochrome LCD
  - — Panel width is programmable from 32 to 128 pixels
  - — Panel length up to 64 pixels
  - — Bits per pixel of 1, 2, or 4 bits
- **ROM/SRAM/FLASH memory control**
  - — Decodes up to 6 separate memory segments of up to 256 Mbytes each
  - — Each segment can be configured as 8, 16, or 32 bits wide and supports page-mode access
  - — Programmable access time for conventional ROM/SRAM/FLASH memory
- **Supports Removable FLASH card interface**
  - — Enables connection to removable FLASH card for addition of expansion FLASH memory modules
- **Parallel Interface**
  - — Allows for bi-directional connection to PC parallel port using an enhanced parallel port (EPP) interface.
- **37.5 k-bytes of on-chip SRAM for fast program execution and/or as a frame buffer**
- **On-chip Boot ROM for manufacturing support**
- **Integrated DAI interface**
  - — Connects directly to a Crystal® audio DAC
- **27-bits of general-purpose I/O**
  - — Three 8-bit and one 3-bit GPIO port
  - — Supports scanning keyboard matrix
- **SIR (up to 115.2 kbps) infrared encoder/decoder**
  - — IrDA (Infrared Data Association) SIR protocol encoder
- **DC-to-DC converter interface (PWM)**
  - — Provides two 96-kHz clock outputs with programmable duty ratio (from 1-in-16 to 15-in-16) that can be used to drive a DC to DC converter
- **208-pin LQFP or 256-ball PBGA packages**
- **Full JTAG boundary scan and Embedded ICE support**

OVERVIEW *(cont.)*

troller, an audio DAC interface, and a FLASH memory interface. Using the EP7209, a portable audio decoder solution can be built with the addition of an LCD display, an audio DAC, a FLASH memory subsystem, and a small number of additional low cost components.

The EP7209 uses its powerful 32-bit RISC processing engine to implement audio decompression algorithms in software. The nature of the on-board RISC processor and the availability of efficient C-compilers and other software development tools ensures that a wide range of audio decompression algorithms can easily be ported to and run on the EP7209.

The EP7209 uses external memory for storing application code. The use of external memory to support software audio decompression algorithms ensures that the audio decompression system solution can be tailored to the requirements of the application. Software can be placed in a low cost mask ROM for price sensitive applications, or can be placed in external FLASH memory to enable upgradeable systems. The EP7209's 8 kbyte on-board cache and programmable wait state generator ensure that a wide range of memory options can be utilized.

The EP7209 implements a full ISO-compliant MPEG 1, 2, & 2.5 layer 3 audio decompression engine in less than 50% of its available processing capability. This leaves significant processing power available for product differentiation.

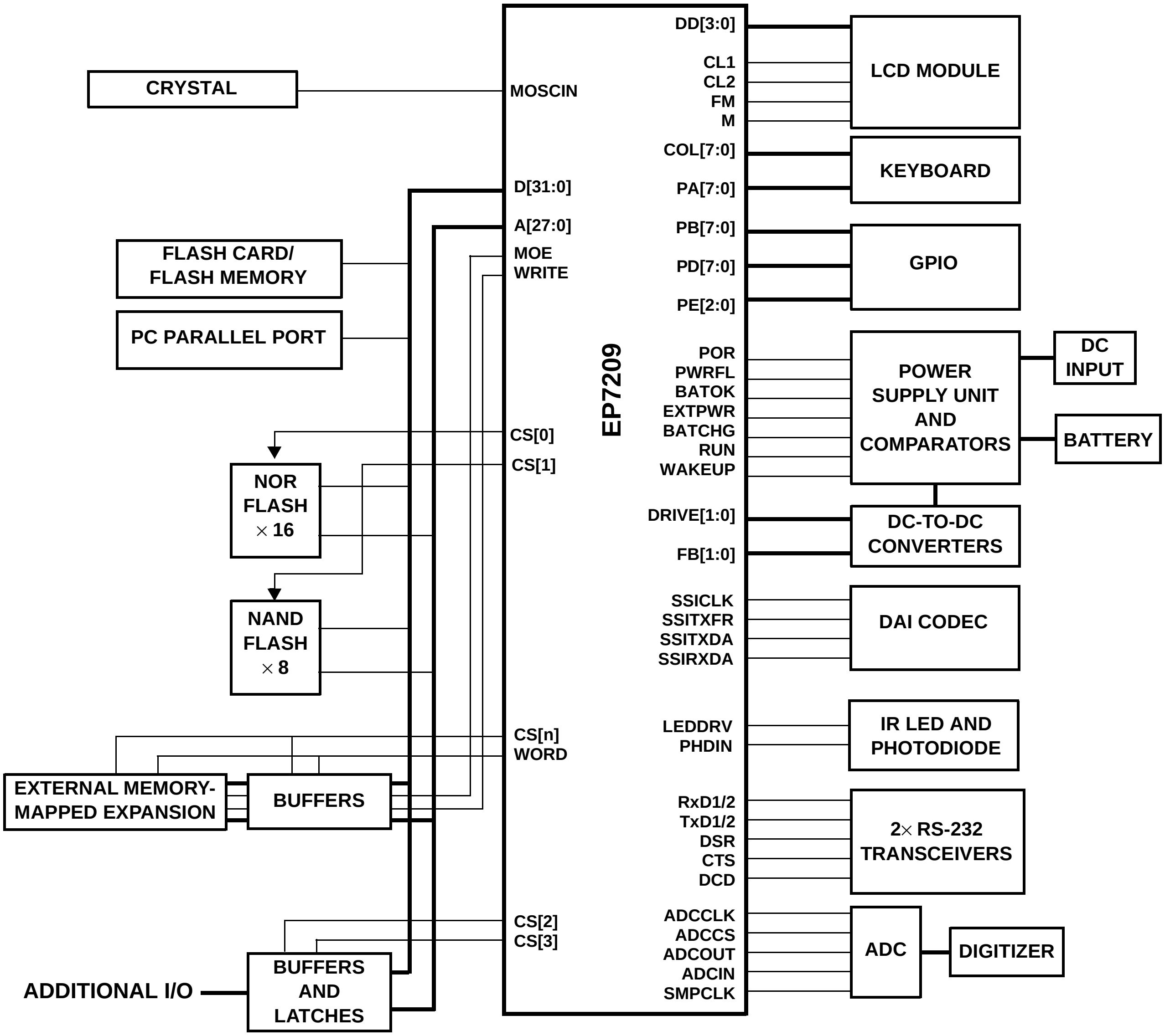
MPEG 1, 2, & 2.5 Layer 3 Object Code Library

Cirrus Logic provides an object code library for enabling MPEG 1, 2, & 2.5 layer 3 audio decompression. This library supports the MPEG 1 sample rates of 48 k, 44.1 k and 32 k bits per second; the MPEG 2 sample rates of 24 k, 22.05 k and 16 k bits per second; and the MPEG 2.5 sample rates of 12 k, 11.025 k and 8 k bits per second. In addition to all standard fixed compressed data rates, the MPEG layer 3 object code library also supports decompression of variable bit-rate data streams.

OVERVIEW *(cont.)*

Power Management

The EP7209 is designed for ultra-low-power operation. Its core operates at only 2.5V, while its I/O has an operation range of 2.5V-3.3V. Through careful design, Cirrus Logic has achieved extremely low power consumption with the EP7209. This is achieved by using a combination of dynamically adjustable core clock frequencies, low power states utilized during periods of inactivity, and fully static design principles. For example, when decompressing MPEG 1 layer 3 music data with sample rates of 44.1 kHz and 128 kbits/sec, the EP7209 consumes less than 87 mW. At sampling frequencies of 22.05 kHz and 64 kbits/sec, power consumption falls to 50 mW.



An EP7209–Based System

OVERVIEW (cont.)

Audio Data Memory Interfaces

The EP7209 connects directly to both on-system FLASH memory and to removable FLASH memory-cards. The generality of the external interface on the EP7209 allows for the use of a wide variety of additional memory types for compressed audio data storage.

LCD Display interface

The EP7209 interfaces directly to a single-scan panel monochrome LCD display. For portable digital audio player applications that require LCD displays, a 4 kbyte display buffer will enable display sizes of up to 640 pixels wide by 240 pixels high by 4 bits deep.

Data Download

The EP7209 includes support for bi-directional connection to a PC through an enhanced parallel port (EPP) interface. This interface enables downloading of compressed music data from a PC to an EP7209-based portable digital audio player.

The EP7209 can also be connected to industry standard USB slave devices through an external interface. The power of the EP7209 coupled with the 36 MHz external data bus ensures that the EP7209 can support rapid transfer of compressed audio data over a USB interface.

The EP7209 also includes a built-in 115.2 kbps IrDA SIR protocol encoder/decoder that can be used to drive an infrared communication interface to download the data.

Digital Audio Interface

The EP7209 integrates an DAI interface to enable a direct connection to many low cost, low power, high quality audio converters. In particular, the DAI interface can be used to drive the Crystal CS4343 low power audio DAC or the Crystal CS4250 low power audio codec. Both the CS4343 and the CS4250 provide digital bass and treble boost, digital volume control and compressor-limiter functions.

Packaging

The EP7209 is available in a 208-pin LQFP package and a 256-ball PBGA package.

System Design

As shown in the system block diagram, simply adding FLASH memory, an LCD display, an audio DAC, a code ROM, and some discrete components completes a low power digital audio player system.

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/

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. CONVENTIONS

This section presents acronyms, abbreviations, units of measurement, and conventions used in this data book.

1.1 Acronyms and Abbreviations

The following table lists abbreviations and acronyms used in this data book.

Acronym/ Abbreviation	Definition
AC	alternating current
A/D	analog-to-digital
ADC	analong-to-digital converter
codec	coder/decoder
CMOS	complementary metal oxide semiconductor
CPU	central processing unit
D/A	digital-to-analog
DC	direct current
DMA	direct-memory access
EPB	embedded peripheral bus
FCS	frame check sequence
FIFO	first in/first out
GPIO	general purpose I/O
ICT	in circuit test
IR	infrared
IrDA	Infrared Data Association
JTAG	Joint Test Action Group
LCD	liquid crystal display
LED	light-emitting diode
LSB	least significant bit

Acronym/ Abbreviation	Definition
MIPS	millions of instructions per second
LQFP	low profile quad flat pack
MMU	memory mangement unit
MSB	most significant bit
PBGA	plastic ball grid array
PCB	printed circuit board
PDA	personal digital assistant
PIA	peripheral interface adapter
PLL	phase locked loop
PSU	power supply unit
p/u	pull-up resistor
RAM	random access memory
RISC	reduced instruction set computer
ROM	read-only memory
RTC	realtime clock
SIR	slow (9600–115.2 kbps) infrared
SRAM	static random access memory
SSI	synchronous serial interface
TAP	test access port
TLB	translation look aside buffer
UART	universal asynchronous receiver transmitter

1.2 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

Symbol	Unit of Measure
μF	microfarad
μW	microwatt
μs	microsecond (1,000 nanoseconds)
mA	milliampere
mW	milliwatt
ms	millisecond (1,000 microseconds)
ns	nanosecond
V	volt
W	watt

1.3 General Conventions

Hexadecimal numbers are presented with all letters in uppercase and a lowercase *h* appended. For example, 14h and 03CAh are hexadecimal numbers. Binary numbers are enclosed in single quotation marks when in text (for example, ‘11’ designates a binary number). Numbers not indicated by an *h* 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]**).

The use of ‘tbd’ indicates values that are ‘to be determined’, ‘n/a’ designates ‘not available’, and ‘n/c’ indicates a pin that is a ‘no connect’.

1.4 Pin Description Conventions

Abbreviations used for signal directions in **Section 2** are listed in the following table:

Abbreviation	Direction
I	Input
O	Output
I/O	Input or Output

2. PIN INFORMATION

2.1 Pin Diagrams

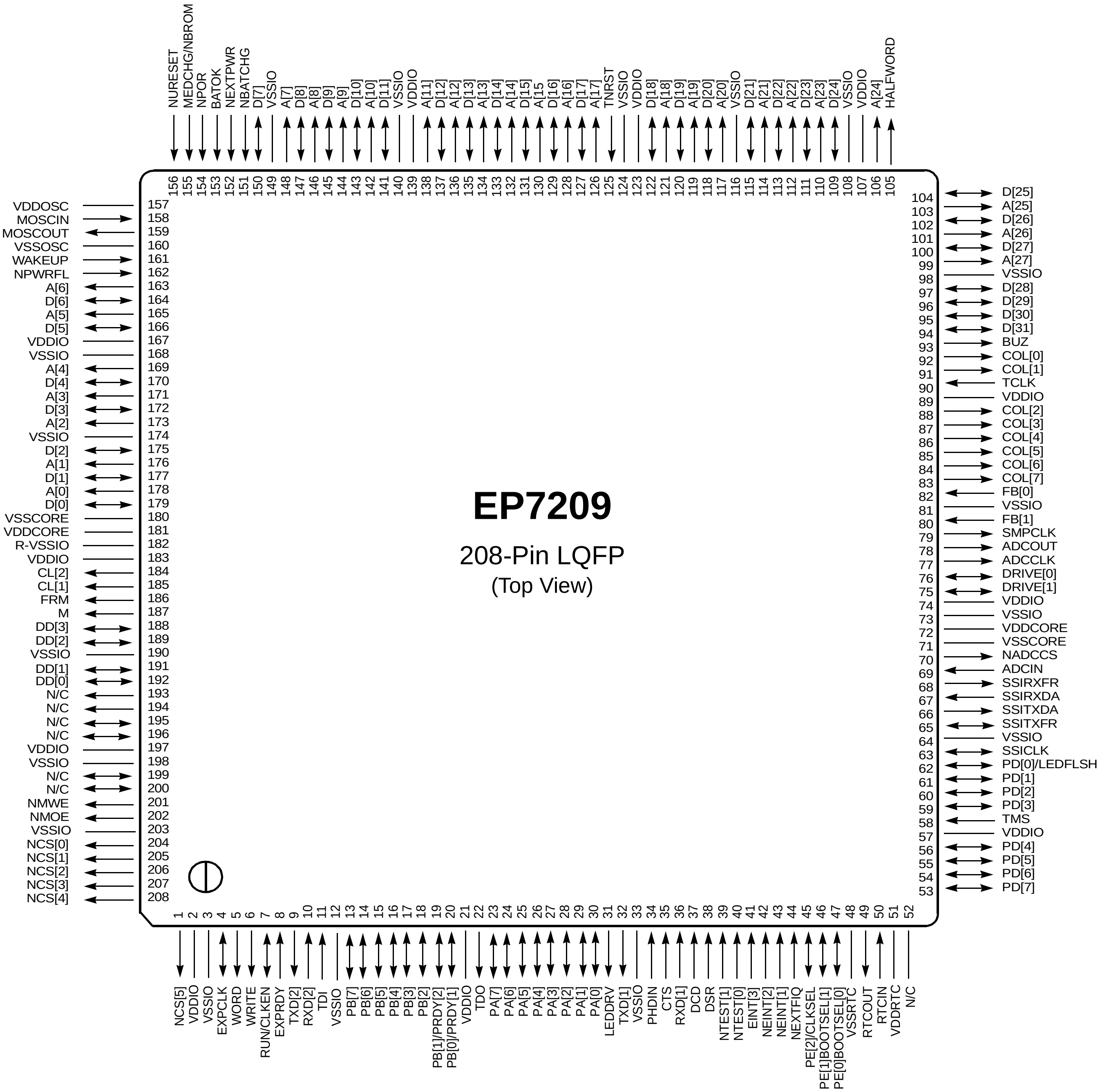
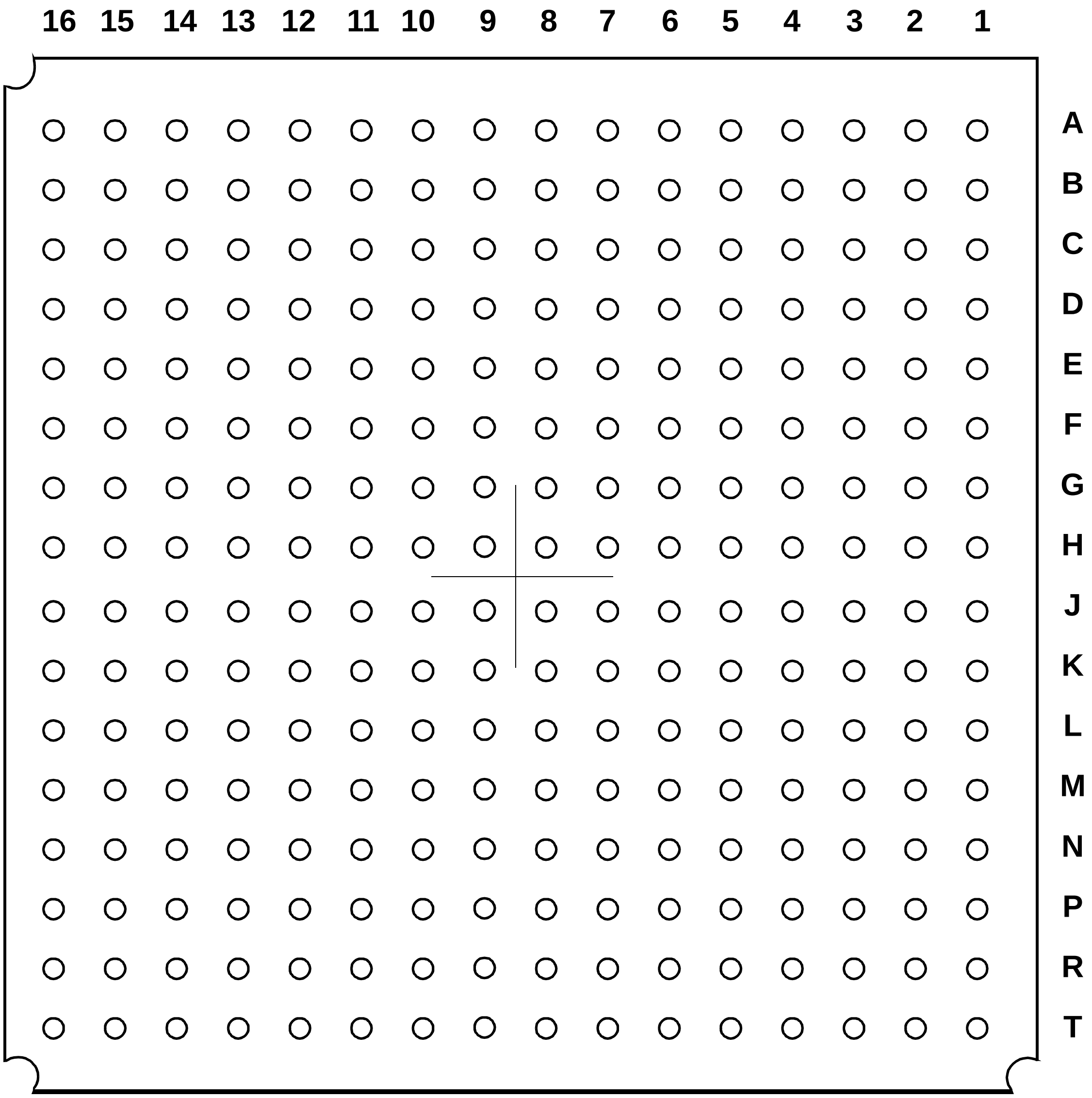


Figure 2-1. 208-Pin LQFP (Low Profile Quad Flat Pack) Pin Diagram



256-Ball PBGA
(Bottom View)

(Call Factory for Availability)

Figure 2-2. 256-Ball Plastic Ball Grid Array Diagram

2.2 Pin Descriptions

This table describes the function of all the external signals to the EP7209. Note that all output signals are tri-stateable to enable the Hi-Z test modes to be supported.

2.2.1 External Signal Functions

Function	Signal Name	Signal	Description
Data bus	D[0–31]	I/O	32-bit system data bus for ROM/SRAM/Flash, and memory mapped I/O expansion
Address bus	A[0–19]	O	20 bits of system byte address during ROM/SRAM/Flash and expansion cycles
		O	– Whenever the EP7209 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 power-down from draining current. Also, the internal peripheral’s signals get set to their Reset State**.**
Memory and Expansion Interface	NMOE	O	ROM/SRAM/Flash, and expansion output enable
	NMWE	O	ROM/SRAM/Flash, and expansion write enable
	NCS[0–3]	O	Expansion channel I/O strobes; active low SRAM-like chip selects for expansion
	NCS[4–5]	O	Expansion channel I/O strobes; active low CS for expansion or for CL-PS6700 select
	EXPRDY	I/O	Expansion channel ready; external expansion devices drive this low to extend the bus cycle
	WRITE	O	Transfer direction, low during reads, high during writes from the EP7209
	WORD	O	Word access enable; driven high during word-wide cycles, low during byte-wide cycles
	HALFWORD	O	Half-Word access flag; driven high to denote upper half-word accesses
	EXPCLK	I/O	Expansion clock rate is the same as the CPU clock for 13 MHz and 18 MHz. It runs at 36.864 MHz for 36,49 and 74 MHz modes; in 13 MHz mode this pin is used as the clock input
Interrupts	NMEDCHG/ BROM	I	Media changed input; active low, deglitched — it is 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	I	External active low fast interrupt request input
	EINT3	I	External active high interrupt request input
	NEINT[1–2]	I	Two general purpose, active low interrupt inputs

Function	Signal Name	Signal	Description
Power Management	NPWRFL	I	Power fail input; active low deglitched input to force system into the Standby State
	BATOK	I	Main battery OK input; falling edge generates a FIQ, a low level in the Standby State inhibits system start up; deglitched input
	NEXTPWR	I	External power sense; must be driven low if the system is powered by an external source
	NBATCHG	I	New battery sense; driven low if battery voltage falls below the "no-battery" threshold; it is a deglitched input
State Control	NPOR	I	Power-on reset input; active low input completely resets the entire system; must be held active for at least two clock cycles to be detected cleanly
	RUN/CLKEN		This pin is programmed to either output the RUN signal or the CLKEN signal. The CLKENSL bit is used to configure this pin. When RUN is selected, the pin will be high when the system is active or idle, low while in the Standby State. When CLKEN is selected, the pin will only be driven low when in the Standby State.
	WAKEUP	I	Wake up deglitched input signal; rising edge forces system into the Operating State; active after a power-on reset
	NURESET	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.
I^{2}S, Codec or SSI2 Interface (See Note)	SSICLK	I/O	I^{2}S/Codec/SSI2 clock signal
	SSITXFR	I/O	I^{2}S/Codec/SSI2 serial data output frame/synchronization pulse output
	SSITXDA	O	I^{2}S/Codec/SSI2 serial data output
	SSIRXDA	I	I^{2}S/Codec/SSI2 serial data input
	SSIRXFR	I/O	SSI2 serial data input frame/synchronization pulse I^{2}S external clock input
ADC Interface (SSI1)	ADCCLK	O	Serial clock output
	NADCCS	O	Chip select for ADC interface
	ADCOUT	O	Serial data output
	ADCIN	I	Serial data input
	SMPCLK	O	Sample clock output

Function	Signal Name	Signal	Description
IrDA and RS232 Interfaces	LEDDRV	O	Infrared LED drive output (UART1)
	PHDIN	I	Photo diode input (UART1)
	TXD[1–2]	O	RS232 UART1 and 2 TX outputs
	RXD[1–2]	I	RS232 UART1 and 2 RX inputs
	DSR	I	RS232 DSR input
	DCD	I	RS232 DCD input
	CTS	I	RS232 CTS input
LCD	DD[0–3]	I/O	LCD serial display data; pins can be used on power up to read the ID of some LCD modules
	CL1	O	LCD line clock
	CL2	O	LCD pixel clock
	FRM	O	LCD frame synchronization pulse output
	M	O	LCD AC bias drive
Keyboard & Buzzer drive LED Flasher	COL[0–7]	O	Keyboard column drives
	BUZ	O	Buzzer drive output
	PD[0]/ LEDFLSH	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	I/O	Port B I/O. All eight Port B bits can be used as GPIOs.
	PB[1]/PRDY2 PB[2–7]		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]/ BOOTSEL0	I/O	Port E I/O (3 bits only). Can be used as general purpose I/O during normal operation.
	PE[1]/ BOOTSEL1	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 EP7209 will use to read from the boot code storage device (e.g., external 8-bit-wide Flash bank).
	PE[2]/ CLKSEL	I/O	During power-on reset, PE[2] is latched by the rising edge of NPOR to select the clock mode of operation (i.e., either the PLL or external 13 MHz clock mode).

Function	Signal Name	Signal	Description
PWM Drives	DRIVE[0–1]	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.
	FB[0–1]	I	PWM feedback inputs
Boundary Scan	TDI	I	JTAG data in
	TDO	O	JTAG data out
	TMS	I	JTAG mode select
	TCLK	I	JTAG clock
	TNRST	I	JTAG async reset
Test	NTEST[0–1]	I	Test mode select inputs. These pins are used in conjunction with the power-on latched state of NURESET.
Oscillators	MOSCIN MOSCOUT	I O	Main 3.6864 MHz oscillator for 18.432MHz–73.728 MHz PLL
	RTCIN RTCOUT	I O	Realtime clock 32.768 kHz oscillator

NOTE: See table below for pin assignment and direction following pin multiplexing.

.

Table 2-1. SSI/Codec/I^{2}S Pin Multiplexing

SSI2	Codec	I^{2}S	Direction	Strength
SSICCLK	PCMCLK	SCLK	I/O	1
SSITXFR	PCMSYNC	LRCK	I/O	1
SSITXDA	PCMOUT	SDOUT	Output	1
SSIRXDA	PCMIN	SDIN	Input	
SSIRXFR	p/u*	MCLK	I/O	1

* p/u = use an ~10 k pull-up

The selection between SSI2 and the codec is controlled by the state of the SERSEL bit in SYSCON2 (**Section 5.2.2 SYSCON2 System Control Register 2**). The choice between the SSI2, codec, and the I^{2}S is controlled by the I2SSEL bit in SYSCON3 (**Section 5.2.3 SYSCON3 System Control Register 3**).

NOTE: All deglitched inputs are via the 16.384 kHz clock. Therefore, the input signal must be active for at least ~61 µs to be detected cleanly.

The following output pins are implemented as bi-directional pins to enable the output side of the pad to be monitored and hence provide more accurate control of timing or duration:

RUN The **RUN** pin is looped back in to skew the address and data bus from each other.

Drive 0 and **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.

2.3 256-Ball PBGA Ball Listing

Table 2-2. 256-Ball PBGA Ball Listing

Ball Location	Name	Type
A1	VDDIO	Pad power
A2	NCS[4]	O
A3	NCS[1]	O
A4	NCAS[0]	O
A5	NCAS[3]	O
A6	DD[1]	O
A7	M	O
A8	VDDIO	Pad power
A9	D[0]	I/O
A10	D[2]	I/O
A11	A[3]	O
A12	VDDIO	Pad power
A13	A[6]	O
A14	MOSCOUT	O
A15	VDDOSC	Oscillator power
A16	VSSIO	Pad ground
B1	NCS[5]	O
B2	VDDIO	Pad power
B3	NCS[3]	O
B4	NMOE	O
B5	VDDIO	Pad power
B7	DD[2]	O
B8	CL[1]	O
B9	VDDCORE	Core power
B10	D[1]	I/O
B11	A[2]	O
B12	A[4]	O
B13	A[5]	O
B14	WAKEUP	I
B15	VDDIO	Pad power
B16	NURESET	I
C1	VDDIO	Pad power
C2	EXPCLK	I
C3	VSSIO	Pad ground
C4	VDDIO	Pad power
C5	VSSIO	Pad ground
C6	VSSIO	Pad ground
C7	VSSIO	Pad ground
C8	VDDIO	Pad power
C9	VSSIO	Pad ground
C10	VSSIO	Pad ground
C11	VSSIO	Pad ground
C12	VDDIO	Pad power
C13	VSSIO	Pad ground
C14	VSSIO	Pad ground
C15	NPOR	I
C16	NEXTPWR	I
D1	WRITE	O
D2	EXPRDY	I
D3	VSSIO	Pad ground
D4	VDDIO	Pad power
D5	NCS[2]	O
D6	NMWE	O
D8	CL[2]	O
D9	VSSCORE	Core ground
D10	D[4]	I/O
D11	NPWRFL	I
D12	MOSCIN	I
D13	VDDIO	Pad power

Table 2-2. *(cont.)*256-Ball PBGA Ball Listing

Ball Location	Name	Type
D14	VSSIO	Pad ground
D15	D[7]	I/O
D16	D[8]	I/O
E1	RXD[2]	I
E2	PB[7]	I
E3	TDI	I
E4	WORD	O
E5	VSSIO	Pad ground
E6	NCS[0]	O
E7	NCAS[2]	O
E8	FRM	O
E9	A[0]	O
E10	D[5]	I/O
E11	VSSOSC	Oscillator ground
E12	VSSIO	Pad ground
E13	NMEDCHG/NBROM	I
E14	VDDIO	Pad power
E15	D[9]	I/O
E16	D[10]	I/O
F1	PB[5]	I
F2	PB[3]	I
F3	VSSIO	Pad ground
F4	TXD[2]	O
F5	RUN/CLKEN	O
F6	VSSIO	Pad ground
F7	NCAS[1]	O
F8	DD[3]	O
F9	A[1]	O
F10	D[6]	I/O

Table 2-2. *(cont.)*256-Ball PBGA Ball Listing

Ball Location	Name	Type
F11	VSSRTC	RTC ground
F12	BATOK	I
F13	NBATCHG	I
F14	VSSIO	Pad ground
F15	D[11]	I/O
F16	VDDIO	Pad power
G1	PB[1]/PRDY[2]	I
G2	VDDIO	Pad power
G3	TDO	O
G4	PB[4]	I
G5	PB[6]	I
G6	VSSCORE	Core ground
G7	VSSRTC	RTC ground
G8	DD[0]	O
G9	D[3]	I/O
G10	VSSRTC	RTC ground
G11	A[7]	O
G12	A[8]	O
G13	A[9]	O
G14	VSSIO	Pad ground
G15	D[12]	I/O
G16	D[13]	I/O
H1	PA[7]	I
H2	PA[5]	I
H3	VSSIO	Pad ground
H4	PA[4]	I
H5	PA[6]	I
H6	PB[0]/PRDY[1]	I
H7	PB[2]	I
H8	VSSRTC	RTC ground

Table 2-2. *(cont.)***256-Ball PBGA Ball Listing**

Ball Location	Name	Type
H9	VSSRTC	RTC ground
H10	A[10]	O
H11	A[11]	O
H12	A[12]	O
H13	A[13]	O
H14	VSSIO	Pad ground
H15	D[14]	I/O
H16	D[15]	I/O
J1	PA[3]	I
J2	PA[1]	I
J3	VSSIO	Pad ground
J4	PA[2]	I
J5	PA[0]	I
J6	TXD[1]	O
J7	CTS	I
J8	VSSRTC	RTC ground
J9	VSSRTC	RTC ground
J10	A[17]/DRA[10]	O
J11	A[16]/DRA[11]	O
J12	A[15]/DRA[12]	O
J13	A[14]	O
J14	TNRST	I
J15	D[16]	I/O
J16	D[17]	I/O
K1	LEDDRV	O
K2	PHDIN	I
K3	VSSIO	Pad ground
K4	DCD	I
K5	NTEST[1]	I
K6	EINT[3]	I

Table 2-2. *(cont.)***256-Ball PBGA Ball Listing**

Ball Location	Name	Type
K7	VSSRTC	RTC ground
K8	ADCIN	I
K9	COL[4]	O
K10	TCLK	I
K11	D[20]	I/O
K12	D[19]	I/O
K13	D[18]	I/O
K14	VSSIO	Pad ground
K15	VDDIO	Pad power
K16	VDDIO	Pad power
L1	RXD[1]	I
L2	DSR	I
L3	VDDIO	Pad power
L4	NEINT[1]	I
L5	PE[2]/CLKSEL	I
L6	VSSRTC	RTC ground
L7	PD[0]/LEDFLSH	I/O
L8	VSSCORE	Core ground
L9	COL[6]	O
L10	D[31]	I/O
L11	VSSRTC	RTC ground
L12	A[22]/DRA[5]	O
L13	A[21]/DRA[6]	O
L14	VSSIO	Pad ground
L15	A[18]/DRA[9]	O
L16	A[19]/DRA[8]	O
M1	NTEST[0]	I
M2	NEINT[2]	I
M3	VDDIO	Pad power
M4	PE[0]/BOOTSEL[0]	I

Table 2-2. *(cont.)***256-Ball PBGA Ball Listing**

Ball Location	Name	Type
M5	TMS	I
M6	VDDIO	Pad power
M7	SSITXFR	I/O
M8	DRIVE[1]	I/O
M9	FB[0]	I
M10	COL[0]	O
M11	D[27]	I/O
M12	VSSIO	Pad ground
M13	A[23]/DRA[4]	O
M14	VDDIO	Pad power
M15	A[20]/DRA[7]	O
M16	D[21]	I/O
N1	NEXTFIQ	I
N2	PE[1]/BOOTSEL[1]	I
N3	VSSIO	Pad ground
N4	VDDIO	Pad power
N5	PD[5]	I/O
N6	PD[2]	I/O
N7	SSIRXDA	I/O
N8	ADCCLK	O
N9	SMPCLK	O
N10	COL[2]	O
N11	D[29]	I/O
N12	D[26]	I/O
N13	HALFWORD	O
N14	VSSIO	Pad ground
N15	D[22]	I/O
N16	D[23]	I/O
P1	VSSRTC	32 K oscillator ground

Table 2-2. *(cont.)***256-Ball PBGA Ball Listing**

Ball Location	Name	Type
P2	RTCOUT	O
P3	VSSIO	Pad ground
P4	VSSIO	Pad ground
P5	VDDIO	Pad power
P6	VSSIO	Pad ground
P7	VSSIO	Pad ground
P8	VDDIO	Pad power
P9	VSSIO	Pad ground
P10	VDDIO	Pad power
P11	VSSIO	Pad ground
P12	VSSIO	Pad ground
P13	VDDIO	Pad power
P14	VSSIO	Pad ground
P15	D[24]	I/O
P16	VDDIO	Pad power
R1	RTCIN	O
R2	VDDIO	Pad power
R3	PD[4]	I/O
R4	PD[1]	I/O
R5	SSITXDA	O
R6	NADCCS	O
R7	VDDIO	Pad power
R8	ADCOUT	O
R9	COL[7]/PTOUT	O
R10	COL[3]	O
R11	COL[1]	O
R12	D[30]	I/O
R13	A[27]/DRA[0]	O
R14	A[25]/DRA[2]	O
R15	VDDIO	Pad power

Table 2-2. *(cont.)***256-Ball PBGA Ball Listing**

Ball Location	Name	Type
R16	A[24]/DRA[3]	O
T1	VDDRTC	32 K oscillator power
T2	PD[7]	I/O
T3	PD[6]	I/O
T4	PD[3]	I/O
T5	SSICLK	I/O
T6	SSIRXFR	–
T7	VDDCORE	Core power
T8	DRIVE[0]	I/O
T9	FB[1]	I
T10	COL[5]	O
T11	VDDIO	Pad power
T12	BUZ	O
T13	D[28]	I/O
T14	A[26]/DRA[1]	O
T15	D[25]	I/O
T16	VDDIO	Pad power

2.3.1 PBGA Ground Connections

Table 2-3 lists the balls on the PBGA package that must be connected to ground. These extra leads are not used on the PBGA package.

Table 2-3. PBGA Balls to Connect to Ground (V_{SS})

A1 A16	E3 E5 E12 E14	J3 J8 J9 J14	N3 N4 N14
B2 B15	F3 F6 F11 F14	K3 K7 K14	P3–P14
C3–C14	G7 G10 G14	L3 L6 L11 L14	R2 R15
D3 D4 D13 D14	H3 H8 H9 H14	M3 M12 M14	T16

2.4 208-Pin LQFP Numeric Pin Listing

Table 2-4. 208-Pin LQFP Numeric Pin Listing

Pin No.	Signal	Type	Strength	Reset State
1	NCS[5]	Out	1	Low
2	VDDIO	Pad Pwr		
3	VSSIO	Pad Gnd		
4	EXPCLK	I/O	1	
5	WORD	Out	1	Low
6	WRITE	Out	1	Low
7	RUN/CLK EN	I/O	1	Low
8	EXPRDY	I/O	1	
9	TXD2	Out	1	High
10	RXD2	In		
11	TDI	In	with p/u*	
12	VSSIO	Pad Gnd		
13	PB[7]	I/O	1	Input
14	PB[6]	I/O	1	Input
15	PB[5]	I/O	1	Input
16	PB[4]	I/O	1	Input
17	PB[3]	I/O	1	Input
18	PB[2]	I/O	1	Input
19	PB[1]/ PRDY2	I/O	1	Input
20	PB[0]/ PRDY1	I/O	1	Input
21	VDDIO	Pad Pwr		
22	TDO	Out	1	Tristate
23	PA[7]	I/O	1	Input
24	PA[6]	I/O	1	Input
25	PA[5]	I/O	1	Input
26	PA[4]	I/O	1	Input
27	PA[3]	I/O	1	Input
28	PA[2]	I/O	1	Input
29	PA[1]	I/O	1	Input
30	PA[0]	I/O	1	Input
31	LEDDRV	Out	1	Low
32	TXD1	Out	1	High
33	VSSIO	Pad Gnd	1	High
34	PHDIN	In		
35	CTS	In		
36	RXD1	In		
37	DCD	In		
38	DSR	In		

Table 2-4. *(cont.)* **208-Pin LQFP Numeric Pin**

Pin No.	Signal	Type	Strength	Reset State
39	NTEST1	In	With p/u*	High
40	NTEST0	In	With p/u*	High
41	EINT3	In		
42	NEINT2	In		
43	NEINT1	In		
44	NEXTFIQ	In		
45	PE[2]/ CLKSEL	I/O	1	Input
46	PE[1]/ BOOTSEL1	I/O	1	Input
47	PE[0]/ BOOTSEL0	I/O	1	Input
48	VSSRTC	VSSRTC32 K Osc Gnd		
49	RTCOUT	32 K Osc		X
50	RTCIN	32 K Osc		X
51	VSSRTC	32 K Osc power		
52	nc		—	
53	PD[7]	I/O	1	Low
54	PD[6]	I/O	1	Low
55	PD[5]	I/O	1	Low
56	PD[4]	I/O	1	Low
57	VDDIO	Pad Pwr		—
58	TMS	In	with p/u*	—
59	PD[3]	I/O	1	Low
60	PD[2]	I/O	1	Low
61	PD[1]	I/O	1	Low

Table 2-4. *(cont.)* **208-Pin LQFP Numeric Pin**

Pin No.	Signal	Type	Strength	Reset State
62	PD[0]/ LEDFLSH	I/O	1	Low
63	SSI/CLK	I/O	1	Input
64	VSSIO	Pad Gnd		—
65	SSITXFR	I/O	1	Low
66	SSITXDA	Out	1	Low
67	SSIRXDA	In		
68	SSIRXFR	I/O		Input
69	ADCIN	In		
70	NADCCS	Out	1	High
71	VSSCORE	Core Gnd		
72	VDDCORE	Core Pwr		
73	VSSIO	Pad Gnd		
74	VDDIO	Pad Pwr		
75	DRIVE[1]	I/O	1	High/ Low
76	DRIVE[0]	I/O	1	High/ Low
77	ADCCLK	Out	1	Low
78	ADCOUT	Out	1	
79	SMPCLK	Out	1	
80	FB[1]	In		
81	VSSIO	Pad Gnd	1	
82	FB0	In	1	
83	COL7	Out	1	High
84	COL6	Out	1	High
85	COL5	Out	1	High
86	COL4	Out	1	High

Table 2-4. *(cont.)*208-Pin LQFP Numeric Pin

Pin No.	Signal	Type	Strength	Reset State
87	COL3	Out	1	High
88	COL2	Out	1	High
89	VDDIO	Pad Pwr		
90	TCLK	In		
91	COL[1]	Out	1	High
92	COL[0]	Out	1	High
93	BUZ	Out	1	Low
94	D[31]	I/O	1	Low
95	D[30]	I/O	1	Low
96	D[29]	I/O	1	Low
97	D[28]	I/O	1	Low
98	VSSIO	Pad Gnd		
99	A[27]/DR A[0]	Out	2	Low
100	D[27]	I/O	1	Low
101	A[26]/DR A[1]	Out	2	Low
102	D[26]	I/O	1	Low
103	A[25]/DR A[2]	Out	2	Low
104	D[25]	I/O	1	Low
105	HALFWORD	Out	1	Low
106	A[24]/DR A[3]	Out	1	Low
107	VDDIO	Pad Pwr		—
108	VSSIO	Pad Gnd		—
109	D[24]	I/O	1	Low
110	A[23]/DR A[4]	Out	1	Low

Table 2-4. *(cont.)*208-Pin LQFP Numeric Pin

Pin No.	Signal	Type	Strength	Reset State
111	D[23]	I/O	1	Low
112	A[22]/DR A[5]	Out	1	Low
113	D[22]	I/O	1	Low
114	A[21]/DR A[6]	Out	1	Low
115	D[21]	I/O	1	Low
116	VSSIO	Pad Gnd		—
117	A[20]/DR A[7]	Out	1	Low
118	D[20]	I/O	1	Low
119	A[19]/DR A[8]	Out	1	Low
120	D[19]	I/O	1	Low
121	A[18]/DR A[9]	Out	1	Low
122	D[18]	I/O	1	Low
123	VDDIO	Pad Pwr		—
124	VSSIO	Pad Gnd		—
125	TNRST	In		—
126	A[17]/DR A[10]	Out	1	Low
127	D[17]	I/O	1	Low
128	A[16]/DR A[11]	Out	1	Low
129	D[16]	I/O	1	Low
130	A[15]/DR A[12]	Out	1	Low
131	D[15]	I/O	1	Low
132	A[14]	Out	1	Low
133	D[14]	I/O	1	Low
134	A[13]	Out	1	Low

Table 2-4. *(cont.)*208-Pin LQFP Numeric Pin

Pin No.	Signal	Type	Strength	Reset State
135	D[13]	I/O	1	Low
136	A[12]	Out	1	Low
137	D[12]	I/O	1	Low
138	A[11]	Out	1	Low
139	VDDIO	Pad Pwr		
140	VSSIO	Pad Gnd		
141	D[11]	I/O	1	Low
142	A[10]	Out	1	Low
143	D[10]	I/O	1	Low
144	A[9]	Out	1	Low
145	D[9]	I/O	1	Low
146	A[8]	Out	1	Low
147	D[8]	I/O	1	Low
148	A[7]	Out	1	Low
149	VSSIO	Pad Gnd		
150	D[7]	I/O	1	Low
151	NBATCHG	In		
152	NEXTPWR	In		
153	BATOK	In		
154	NPOR	In	Schmitt	
155	NMEDCHG/BROM	In		
156	NURESET	In	Schmitt	
157	VDDOSC	VDDOSCVSSOSCOsc Pwr		—
158	MOSCIN	3M6864 Osc		X

Table 2-4. *(cont.)*208-Pin LQFP Numeric Pin

Pin No.	Signal	Type	Strength	Reset State
159	MOSCOUT	3M6864 Osc		X
160	VSSOSC	VSSOSCVSSCOREOsc Gnd		
161	WAKEUP	In	Schmitt	Input
162	NPWRFL	In		Input
163	A[6]	Out	1	Low
164	D[6]	I/O	1	Low
165	A[5]	Out	1	Low
166	D[5]	I/O	1	Low
167	VDDIO	Pad Pwr		
168	VSSIO	Pad Gnd		
169	A[4]	Out	1	Low
170	D[4]	I/O	1	Low
171	A[3]	Out	2	Low
172	D[3]	I/O	1	Low
173	A[2]	Out	2	Low
174	VSSIO	Pad Gnd		
175	D[2]	I/O	1	Low
176	A[1]	Out	1	Low
177	D[1]	I/O	1	Low
178	A[0]	Out	1	Low
179	D[0]	I/O	1	Low
180	VSSCORE	Core Gnd		
181	VDDCORE	VDDCOREVDDRTCCore Pwr		

Table 2-4. *(cont.)***208-Pin LQFP Numeric Pin**

Pin No.	Signal	Type	Strength	Reset State
182	VSSIO	Pad Gnd		
183	VDDIO	Pad Pwr		
184	CL2	Out	1	Low
185	CL[1]	Out	1	Low
186	FRM	Out	1	Low
187	M	Out	1	Low
188	DD[3]	I/O	1	Low
189	DD[2]	I/O	1	Low
190	VSSIO	VSSIO VDDIO Pad Gnd		
191	DD[1]	I/O	1	Low
192	DD[0]	I/O	1	Low
193	NC/NRAS [1]	Out	1	High
194	NC/NRAS [0]	Out	1	High
195	NC/NCAS [3]	I/O	2	High

Table 2-4. *(cont.)***208-Pin LQFP Numeric Pin**

Pin No.	Signal	Type	Strength	Reset State
196	NC/NCAS [2]	I/O	2	High
197	VDDIO	Pad Pwr		
198	VSSIO	Pad Gnd		
199	NC/NCAS [1]	I/O	2	High
200	NC/NCAS [0]	I/O	2	High
201	NMWE	Out	1	High
202	NMOE	Out	1	High
203	VSSIO	Pad Gnd		—
204	NCS[0]	Out	1	High
205	NCS[1]	Out	1	High
206	NCS[2]	Out	1	High
207	NCS[3]	Out	1	High
208	NCS[4]	Out	1	High

NOTE: ‘With p/u’ means with internal pull-up on the pin.

2.5 JTAG Pin Ordering for 208-Pin LQFP Package

Table 2-5. JTAG Pin Ordering for 208-Pin LQFP Package

Pin No.	Signal	Type	Position
1	NCS[5]	Out	1
4	EXPCLK	I/O	3
5	WORD	Out	6
6	WRITE	Out	8
7	RUN/CLKEN	I/O	10
8	EXPRDY	I/O	13
9	TXD2	Out	14
10	RXD2	In	16
13	PB[7]	I/O	17
14	PB[6]	I/O	20
15	PB[5]	I/O	23
16	PB[4]	I/O	26
17	PB[3]	I/O	29
18	PB[2]	I/O	32
19	PB[1]/ PRDY2	I/O	35
20	PB[0]/ PRDY1	I/O	38
23	PA[7]	I/O	41
24	PA[6]	I/O	44
25	PA[5]	I/O	47
26	PA[4]	I/O	50
27	PA[3]	I/O	53
28	PA[2]	I/O	56
29	PA[1]	I/O	59
30	PA[0]	I/O	62
31	LEDDRV	Out	65
32	TXD1	Out	67

Table 2-5. *(cont.)*JTAG Pin Ordering for 208-Pin LQFP Package

Pin No.	Signal	Type	Position
34	PHDIN	In	69
35	CTS	In	70
36	RXD1	In	71
37	DCD	In	72
38	DSR	In	73
39	NTEST1	In	74
40	NTEST0	In	75
41	EINT3	In	76
42	NEINT2	In	77
43	NEINT1	In	78
44	NEXTFIQ	In	79
45	PE[2]/CLKSEL	I/O	80
46	PE[1]/ BOOTSEL1	I/O	83
47	PE[0]/ BOOTSEL0	I/O	86
53	PD[7]	I/O	89
54	PD[6]	I/O	92
55	PD[5]	I/O	95
56	PD[4]	I/O	98
59	PD[3]	I/O	101
60	PD[2]	I/O	104
61	PD[1]	I/O	107
62	PD[0]/ LEDFLSH	I/O	110
63	SSICCLK	I/O	113
65	SSITXFR	I/O	116
66	SSITXDA	Out	119
67	SSIRXDA	In	121
68	SSIRXFR	Out	122

Table 2-5. *(cont.)*JTAG Pin Ordering for 208-Pin LQFP Package

Pin No.	Signal	Type	Position
68	SSIRXFR	Out	122
69	ADCIN	In	125
70	NADCCS	Out	126
75	DRIVE1	I/O	128
76	DRIVE0	I/O	131
77	ADCCLK	Out	134
78	ADCOUT	Out	136
79	SMPLCK	Out	138
80	FB1	In	140
82	FB0	In	141
83	COL7	Out	142
84	COL6	Out	144
85	COL5	Out	146
86	COL4	Out	148
87	COL3	Out	150
88	COL2	Out	152
91	COL1	Out	154
92	COL0	Out	156
93	BUZ	Out	158
94	D[31]	I/O	160
95	D[30]	I/O	163
96	D[29]	I/O	166
97	D[28]	I/O	169
99	A[27]	Out	172
100	D[27]	I/O	174
101	A[26]	Out	177
102	D[26]	I/O	179
103	A[25]	Out	182
104	D[25]	I/O	184

Table 2-5. *(cont.)*JTAG Pin Ordering for 208-Pin LQFP Package

Pin No.	Signal	Type	Position
105	HALF WORD	Out	187
106	A[24]	Out	189
109	D[24]	I/O	191
110	A[23]	Out	194
111	D[23]	I/O	196
112	A[22]	Out	199
113	D[22]	I/O	201
114	A[21]	Out	204
115	D[21]	I/O	206
117	A[20]	Out	209
118	D[20]	I/O	211
119	A[19]	Out	214
120	D[19]	I/O	216
121	A[18]	Out	219
122	D[18]	I/O	221
126	A[17]	Out	224
127	D[17]	I/O	226
128	A[16]	Out	229
129	D[16]	I/O	231
130	A[15]	Out	234
131	D[15]	I/O	236
132	A[14]	Out	239
133	D[14]	I/O	241
134	A[13]	Out	244
135	D[13]	I/O	246
136	A[12]	Out	249
137	D[12]	I/O	251
138	A[11]	Out	254
141	D[11]	I/O	256

Table 2-5. *(cont.)*JTAG Pin Ordering for 208-Pin LQFP Package

Pin No.	Signal	Type	Position
142	A[10]	Out	259
143	D[10]	I/O	261
144	A[9]	Out	264
145	D[9]	I/O	266
146	A[8]	Out	269
147	D[8]	I/O	271
148	A[7]	Out	274
150	D[7]	I/O	276
151	NBATCHG	In	279
152	NEXTPWR	In	280
153	BATOK	In	281
154	NPOR	In	282
155	NMEDCHG/ BROM	In	283
156	NURESET	In	284
161	WAKEUP	In	285
162	NPWRFL	In	286
163	A[6]	Out	287
164	D[6]	I/O	289
165	A[5]	Out	292
166	D[5]	I/O	294
169	A[4]	Out	297
170	D[4]	I/O	299
171	A[3]	Out	302
172	D[3]	I/O	304

Table 2-5. *(cont.)*JTAG Pin Ordering for 208-Pin LQFP Package

Pin No.	Signal	Type	Position
173	A[2]	Out	307
175	D[2]	I/O	309
176	A[1]	Out	312
177	D[1]	I/O	314
178	A[0]	Out	317
179	D[0]	I/O	319
184	CL2	Out	322
185	CL1	Out	324
186	FRM	Out	326
187	M	Out	328
188	DD[3]	I/O	330
189	DD[2]	I/O	333
191	DD[1]	I/O	336
192	DD[0]	I/O	339
201	NMWE	Out	358
202	NMOE	Out	360
204	NCS[0]	Out	362
205	NCS[1]	Out	364
206	NCS[2]	Out	366
207	NCS[3]	Out	368
208	NCS[4]	Out	370

NOTE:

1)See **Table 2-1. SSI/Codec/I2S Pin Multiplexing** for pin naming/functionality.

2)For each pad, the JTAG connection ordering is input, output, then enable as applicable.

3. FUNCTIONAL DESCRIPTION

The EP7209 device is a single-chip embedded controller designed to be used in low cost and ultra-low-power applications. A hand-held personal organizer and hand-held internet browser are just two of the many potential applications of the device.

The EP7209 operates at 2.5 V and up to 74 MHz using the ARM7TDMI CPU with MMU, a write buffer, and cache.

The EP7209 contains the following functionality:

- ARM720T core processor — giving Thumb and WinCE™ support and dynamically programmable core clock speeds of 18 MHz, 36 MHz, 49 MHz, and 74 MHz
- A digital audio interface to I^2S DACs
- I^2S A/D data flexibility — data width configurable between 8 to 16 bits
- Increased on-chip SRAM size to 38,400 bytes; the entire display memory for a 640 x 240 x 2 bits/pixel can be contained within this on-chip SRAM
- Dedicated LED flasher pin running from the realtime clock (RTC))
- Full JTAG boundary scan and Embedded ICE support

The core-logic functionality is built around the ARM720T processor (i.e., an ARM7TDMI 32-bit RISC CPU with a WinCE compatible MMU, a write buffer, and 8 kbytes of cache). At 74 MHz and with the on-chip four-way set-associative cache, the EP7209 delivers approximately 66 Dhrystone 2.1 MIPS of sustained performance (74 MIPS peak). This is approximately the same as a 100 MHz Pentium-based PC.

EP7209 interfaces:

- Two ROM/SRAM/Flash banks each up to 256 Mbytes
- Up to four expansion devices, each up to 256 Mbytes in size
- An interface to up to two Cirrus Logic CL-PS6700 PC Card controller devices to support two PC Card slots

The expansion devices could be additional ROM/SRAM/Flash as well. In addition, the EP7209 provides 38,400 bytes of on-chip SRAM that can be used for three purposes:

- Critical program storage
- LCD frame buffer
- General purpose data storage

The EP7209 supports a number of serial interface protocols including two high speed (115 kbps) UARTs with RX and TX FIFOs, a I^2S interface with FIFO support, and two additional synchronous serial interfaces. One of the UARTs also supports the IrDA SIR protocol.

The EP7209 also has an on-chip Boot ROM (128 bytes), which is hardware selectable on power-on reset. This Boot ROM initializes UART1, and downloads the application-specific main boot code into the on-chip SRAM. Once download is complete, execution jumps to the start of the on-chip SRAM. This feature could be used in a manufacturing environment to allow the EP7209 to download code into on-board blank Flash. [See **Section 10 Appendix A: Boot Code on page 135**]

The EP7209 design is optimized for low power dissipation, and is fabricated on a fully static 0.25 micron CMOS process. It is available in 256-ball PBGA and 208-pin LQFP packages.

3.1 Main Functional Blocks

The principle functional blocks in the EP7209 are:

- ARM720T processor, which consists of the following functional sub-blocks:
  - ARM7TDMI CPU core (which supports the logic for the Thumb instruction set, core debug, enhanced multiplier, JTAG, and the Embedded ICE)
  - Memory management unit from the ARM700 and ARM710 processors with WinCE support; it also contains a 64-bit translation look aside buffer (TLB)
  - 8 kbytes of unified instruction and data cache, four-way set associative cache controller
  - Write buffer
- 38,400 bytes of SRAM; full address decode is performed.
- Two PC Card slots supported by an interface to one CL-PS6700 per slot
- Interrupt controller
- Expansion and ROM/SRAM/Flash interface giving 4, 5, or 6 x 256 Mbyte expansion segments with independent wait states
- 27 bits of general purpose peripheral I/O — multiplexed to provide additional functionality where necessary
- I^{2}S digital audio interface with 16-byte transmit and receive FIFOs
- On-chip Boot ROM programmed with serial load boot sequence
- Programmable 1-, 2-, or 4-bit-per-pixel LCD controller with 16-level greyscaler
- Programmable frame buffer start address, allowing a system to be built using only external or internal SRAM for memory.
- Two full duplex 16C550 style UARTs with two 16-byte FIFOs
- IrDA SIR Protocol controller capable of speeds up to 115.2 kbps
- Two 16-bit general purpose timer counters
- A 32-bit real time clock and comparator
- Dedicated LED flasher pin driven from the RTC with programmable duty ratio
- Two PWM interfaces

- Advanced system state control and power management
- Two synchronous serial interfaces for Microwire or SPI peripherals such as ADCs, one supporting both the master and slave mode and up to 512 kbps continuous data rate, while the other supports only the master mode with no buffering, and up to 128 kbps
- Full boundary scan — JTAG
- External tracing support for debug
- The main oscillator and phase locked loop (PLL) to generate twice the maximum CPU clock of 73.728 MHz from a 3.6864 MHz crystal, with an alternative external clock input (used in 13 MHz mode)
- A low power 32.768 kHz oscillator

Figure 3-1. EP7209 Block Diagram shows a simplified block diagram of the EP7209. All external memory and peripheral devices are connected to the 32-bit data bus using the external 28-bit address bus and control signals. Bus transfer times can be extended using the **EXPRDY** signal to lengthen bus cycles.

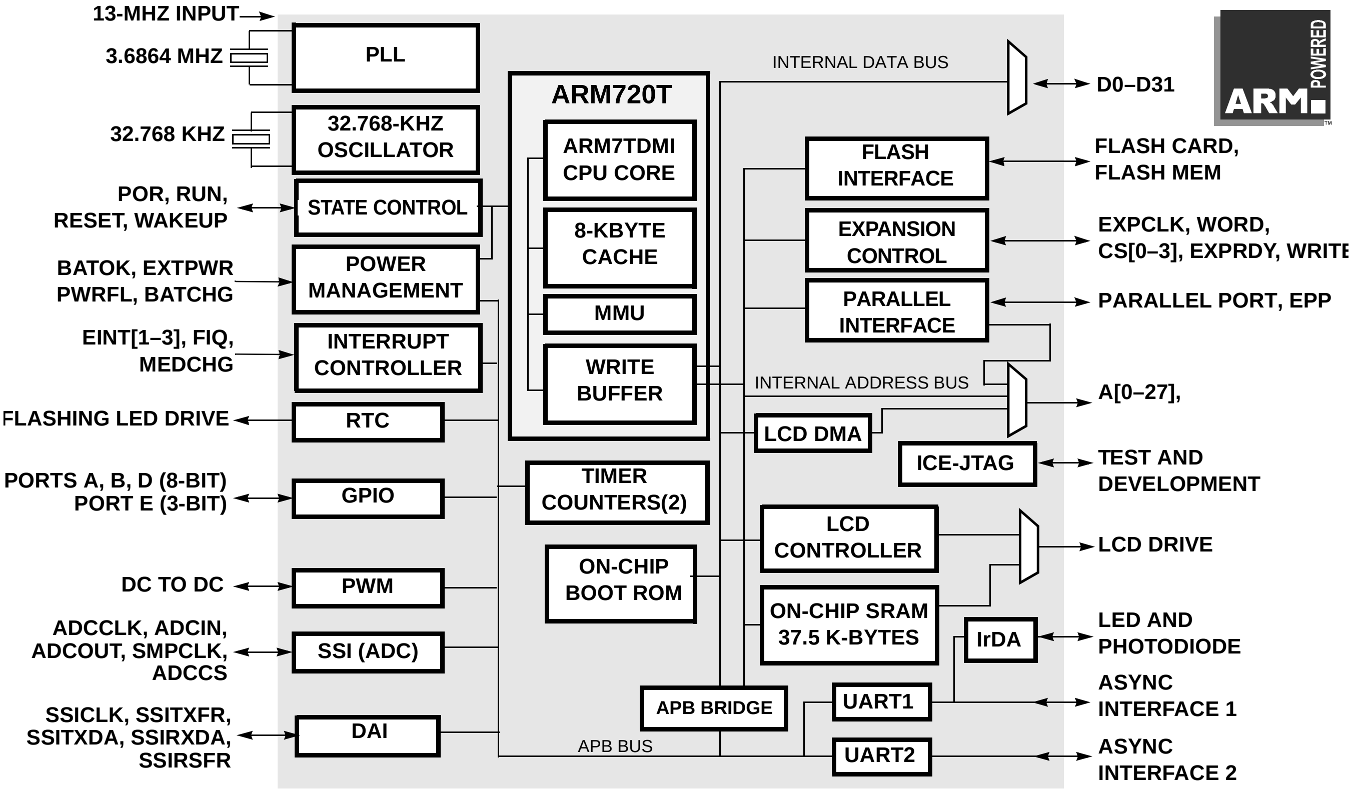


Figure 3-1. EP7209 Block Diagram

3.2 CPU Core

The ARM7TDMI core CPU is a 32-bit RISC processor, which is connected directly to the 8-kbyte unified cache. This cache has 512 lines of 4 words, arranged as a 4-way set associative cache. The cache is directly connected to the ARM7TDMI CPU, and therefore caches the virtual address from the CPU. The MMU translates the virtual address into a physical address, it contains a 64-entry translation look aside buffer (TLB) and is post cache; that is, it only translates external memory references (cache misses) to save power.

The ARM7TDMI CPU, the MMU, the cache, and the write buffer together make up what is called the ARM720T processor. See the *ARM720T Datasheet* for a complete description of the various logic blocks that make up the processor, as well as all internal register information.

3.3 Interrupt Controller

The ARM720T has two interrupt types: interrupt request (IRQ) and fast interrupt request (FIQ). The interrupt controller in the EP7209 controls interrupts from 22 different sources. Seventeen interrupt sources are mapped to the IRQ input and five sources to the FIQ input. FIQs have a higher priority than IRQs. If two interrupts within the same group (IRQ or FIQ) are active, 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 asserted by the interrupt controller. (A description for each bit in this register can be found in **Section 5.3.1 INTSR1 Interrupt Status Register 1 on page 85**.)
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 (e.g., 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 (e.g., external interrupt source) must be held active until its respective service routine starts executing. See **Section 5.13** for more details.

Table 3-1, **Table 3-2** and **Table 3-3** show the names and allocation of interrupts in the EP7209.

Table 3-1. Interrupt Allocation in First Interrupt Register

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	Watchdog expired interrupt
FIQ	3	MCINT	Media changed interrupt
IRQ	4	CSINT	Codec sound interrupt
IRQ	5	EINT1	External interrupt input 1 (NEINT1 pin)
IRQ	6	EINT2	External interrupt input 2 (NEINT2 pin)
IRQ	7	EINT3	External interrupt input 3 (EINT3 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 3-2. Interrupt Allocation in Second 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 3-3. Interrupt Allocation in Third Interrupt Register

Interrupt	Bit in INTMR3 and INTSR3	Name	Comment
FIQ	0	I^2SINT	I^2S interface interrupt

3.3.1 Interrupt Latencies in Different States

3.3.1.1 Operating State

The ARM720T processor checks for a low level on its FIQ/IRQ inputs at the end of each instruction. First, there is a one to two clock cycle synchronization penalty. For the case where the EP7209 is operating at 13 MHz with a 16-bit external memory system, and instruction sequence 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 19.3 µs for the ARM720T processor operating at 13 MHz in this system. For those interrupt inputs which have de-glitching, this figure is increased by the maximum time required to pass through the deglitcher, which is approximately 61 µs (1 cycle of the 16.384 kHz clock derived from the RTC oscillator). This would create an absolute worst case latency of approximately 80 µs. If the ARM720T is run at 36 MHz or greater and/or 32-bit wide external memory, the 19.3 µs value will be reduced.

All the serial data transfer peripherals included in the EP7209 (except for the master-only SSI1) have local buffering to ensure a reasonable interrupt latency response requirement for the OS of 1 ms or less. This assumes that the maximum data rates described in this specification are complied with. If the OS cannot meet this requirement, there will be a risk of data over/underflow occurring.

3.3.1.2 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 20 µs 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 normally the Idle State will have been left because of a pending interrupt, which has to be synchronized by the processor before it can be serviced.

3.3.1.3 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 s to 0.25 s. If the FASTWAKE bit is set, then there will be a latency of between 250 µs to 500 µs. If the system is running from the external clock (at 13 MHz), with the CLKENSL bit in SYSCON2 set to 0, then the latency will also be between 0.125 s and 0.25 s to enable an external oscillator to stabilize. In the case of a 13 MHz system where the clock is not disabled during the Standby State (CLKENSL = 1), then the latency will be the same as described in the Idle State section above.

Whenever the EP7209 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 power-down from draining current. Also, the internal peripheral's signals get set to their Reset State.

Table 3-4 summarizes the three external interrupt sources and the effect they have on interrupt pins.

Table 3-4. External Interrupt Source Latencies

Interrupt Pin	Input State	Operating State Latency	Idle State Latency	Standby State Latency
NEXTFIQ	Not deglitched; must be active for 20 µs to be detected	Worst case latency of 20 µs	Worst case 20 µs: if only single cycle instructions, less than 1 µs	Including PLL/osc. settling time, approx. 0.25 s when FASTWAKE = 0, or approx. 500 µs when FASTWAKE = 1, or = Idle State if in 13 MHz mode with CLENSL set
NEINT1–2	Not deglitched	Worst case latency of 20 µs	As above	As above
EINT3	Not deglitched	Worst case latency of 20 µs	As above	As above
MEDCHG	Deglitched by 16 kHz clock; must be active for at least 80 µs to be detected	Worst case latency of 80 µs	Worst case 80 µs: if only single cycle instructions, 61 µs	As above (note difference if in 13 MHz mode with CLKENSL set)

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 SYSCON2 register description for 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 of 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.

In the case where KBWEN is low and the INTMR2 bit 0 is low, it will only be possible to wakeup the device by using the external **WAKEUP** pin or another enabled interrupt source. The keyboard interrupt capability allows an OS to use either a polled or interrupt-driven keyboard routine, or a combination of both.

NOTE: The keyboard interrupt is NOT deglitched.

3.4 Memory and I/O Expansion Interface

Six separate linear memory or expansion segments are decoded by the EP7209, two of which can be reserved for two PC Card cards, each interfacing to a separate single CL-PS6700 device. Each segment is 256 Mbytes in size. Two additional segments (i.e., in addition to these six) are dedicated to the on-chip SRAM and the on-chip ROM. The on-chip ROM space is fully decoded, and the SRAM space is fully 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 can be 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 4 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 writeable 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, which enables accesses of the form 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 the ROM/SRAM/Flash are by random access without **NCS** being de-asserted between accesses. Again NCS is de-asserted after four consecutive accesses to allow refreshes, etc.

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 38,400 bytes, the on-chip SRAM can be used as the frame buffer.

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

Table 3-5. Boot Options

PE[1]	PE[0]	Boot Block(NCS0)
0	0	32-bit
0	1	8-bit
1	0	16-bit
1	1	Undefined

3.5 EP7209 Boot ROM

The 128 bytes of on-chip Boot ROM contain a 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 10 Appendix A: Boot Code on page 135** 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 EP7209 will boot from an external memory device connected to CS0 (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 power on reset, the EP7209 will be in the Standby State and requires 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 3-6** 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 0x00000000 onwards, and in normal state of operation the Boot ROM contents can be read back from address range 0x70000000.

Table 3-6. Chip Select Address Ranges After Boot From On-Chip Boot ROM

Address Range	Chip Select
0000.0000–0FFF.FFFF	CS7 (Internal only)
1000.0000–1FFF.FFFF	CS6 (Internal only)
2000.0000–2FFF.FFFF	NCS5
3000.0000–3FFF.FFFF	NCS4
4000.0000–4FFF.FFFF	NCS3
5000.0000–5FFF.FFFF	NCS2
6000.0000–6FFF.FFFF	NCS1
7000.0000–7FFF.FFFF	NCS0

3.6 CL-PS6700 PC Card Controller 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** (once 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 EP7209 data bus. Accesses are initiated by a write or read to or 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 3-7. CL-PS6700 Memory Map**.

NOTE: It must be noted that, due to the operating speed of the CL-PS6700, this interface is supported only for processor speeds of 13 and 18 MHz.

Table 3-7. CL-PS6700 Memory Map

Access Type	Addresses for CL-PS6700 Interface 1	Addresses for CL-PS6700 Interface 2
Attribute	0x40000000–0x43FFFFFF	0x50000000– 0x53FFFFFF
I/O	0x44000000–0x47FFFFFF	0x54000000–0x57FFFFFF
Common memory	0x48000000–0x4BFFFFFF	0x58000000–0x5BFFFFFF
CL-PS6700 registers	0x4C000000–0x4FFFFFFF	0x5C000000–0x5FFFFFFF

A complete description of the protocol and AC timing characteristics can be found in the *CL-PS6700 Databook*. 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 of the EP7209'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 3-8. Space Field Decoding** below. The size field is forced to 11 if a word access is required, or 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.

Table 3-8. Space Field Decoding

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

The ‘ptype’ signal on the CL-PS6700s should be connected to the EP7209’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 EP7209 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 EP7209’s CPU to be stalled. The EP7209’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 EP7209 can access the registers in the CL-PS6700, regardless of the state of the **PRDY** signal. If the EP7209 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 EP7209 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 EP7209's CPU will be stalled until the write can 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 EP7209 will re-arbitrate for control of the bus when the **PRDY** signal is reasserted to indicate that the read or write transaction can be completed. The CPU will always be stalled 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 data requested from the card is available in the prefetch buffer (internal to the CL-PS6700).

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 EP7209 continues the cycle to read the data. Otherwise, the **PRDY** signal is de-asserted and the request cycle is stalled. The EP7209 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 EP7209 then arbitrates for the bus and, once the request is granted, the suspended read cycle is resumed. The EP7209 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 EP7209 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 generate an interrupt if a read or write access is timed out (i.e., RD_FAIL or WR_FAIL interrupt). The system software can then determine which access was not successfully completed by reading 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 EP7209 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 EP7209's external interrupts and be used to interrupt the CPU so that the software can service the DMA request under program control.

Each of the CL-PS6700 devices can generate an interrupt **PIRQ**. The **PIRQ** output is open drain on the CL-PS6700 devices, so if there are two CL-PS6700 devices they may be wire OR'ed to the same interrupt which can be connected to one of the EP7209'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 EP7209 in 18.432 MHz mode or the external 13 MHz clock. 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 EP7209 has completed its bus cycles.

A GPIO signal from the EP7209 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 EP7209 enters the Standby State. It is essential that the software monitor 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.

3.7 Serial Interfaces

The EP7209 offers the following serial interfaces in addition to the two UARTs.

Table 3-9. Serial Interface Options

Type	Comments	Referred To As	Max. Transfer Speed
SPI/Microwire 1	Master mode only	ADC Interface	128 Kbps
SPI/Microwire 2	Master/slave mode	SSI2 Interface	512 Kbps
I^{2}S Interface	CD quality DACs and ADCs	I^{2}S Interface	9.216 Mbps
Codec Interface	**Only** for use in the PLL clock mode	Codec Interface	64 Kbps

The inputs/outputs of three of the serial synchronous interfaces (I^{2}S, codec, and SSI2) are multiplexed onto a single set of external interface pins. If the I^{2}SSEL bit of SYSCON3 is low, then either SSI2 or the codec interface will be selected to connect to the external pins. When Bit 0 of SYSCON2 (SERSEL) is high, then the codec is connected to the external pins, when low the master/slave SSI2 is connected to these pins. When the I^{2}SSEL bit is set high, the I^{2}S interface is connected to the external pins. On power up, both the I^{2}SSEL and SERSEL bits are reset low, thus the master/slave SSI2 will be connected to these pins (and configured for slave mode operation to avoid external drive clashes).

The following table contains pin definition information for the three multiplexed interfaces.

NOTE: 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.

Table 3-10. Serial-Pin Assignments

Pin No. LQFP	External Pin Name	SSI2 Slave Mode (Internal Name)	SSI2 Master Mode	Codec Internal Name	I^{2}S Internal Name	Strength
63	SSICCLK	SSICCLK = serial bit clock; Input	Output	PCMCLK = Output	SCLK = Output	1
65	SSITXFR	SSKTXFR = TX frame sync; Input	Output	PCMSYNC = Output	LRCK = Output	1
66	SSITXDA	SSITXDA = TX data; Output	Output	PCMOUT = Output	SDOUT = Output	1
67	SSIRXDA	SSIRXDA = RX data; Input	Input	PCMIN = Input	SDIN = Input	
68	SSIRXFR	SSIRXFR = RX frame sync; Input	Output	p/u*	MCLK	1

* p/u = use an 10 k pull-up

3.7.1 Codec Sound Interface

The codec interface allows direct connection of a telephony type codec to the EP7209. It provides all the necessary clocks and timing pulses and 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 wide, one for the receive data, another for the transmit data).

Data is transferred to or from the codec at 64 kbps. 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 ms, with a latency of 1 ms.

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 ms 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 ms 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 the following diagram:

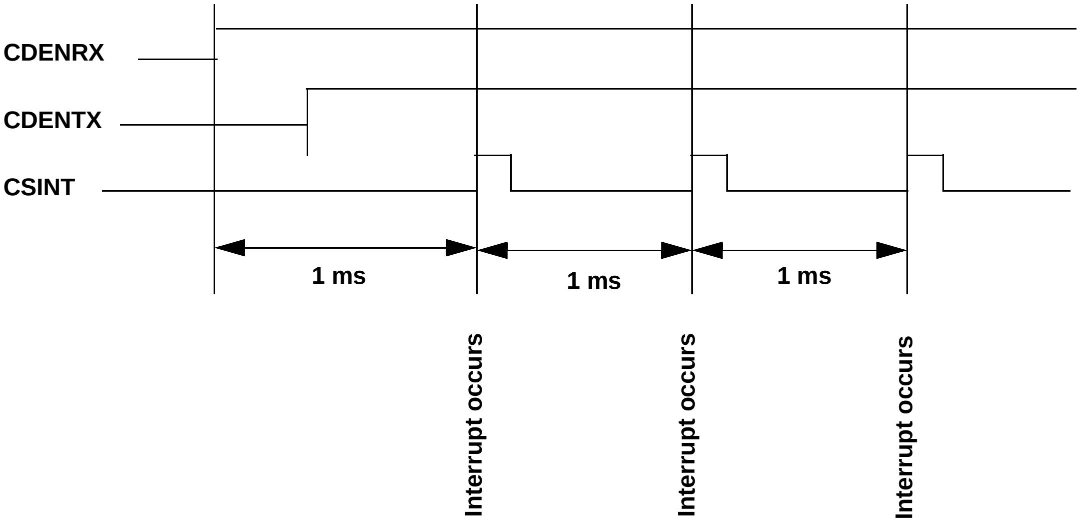


Figure 3-2. Codec Interrupt Timing

3.7.1.1 Codec Interrupt Timing

After the CDENRX and CDENTX enable bits get asserted, the corresponding FIFOs become enabled. When deasserted low, the FIFO status flags (i.e., CRXFE and CTXFF located in the SYSFLAG register) are cleared and the FIFOs appear empty. Additionally, if the CDENTX bit is low, the PCMOUT output is disabled. Asserting either of the two enable bits causes the sync and interrupt generation logic to become active; 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. 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. Input data is sampled on the falling edge of PCMCLK. Data is read from the CODR register.

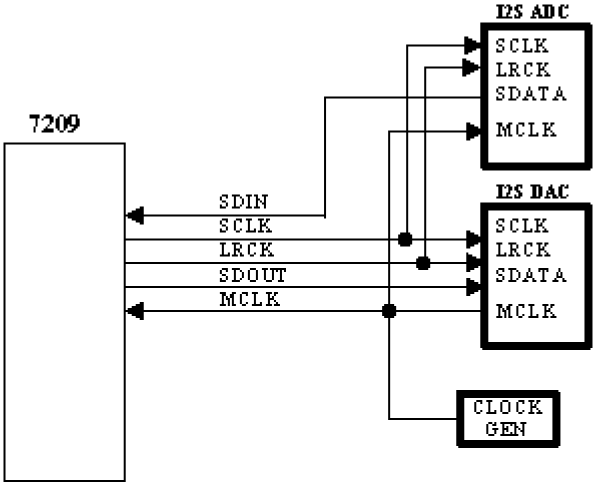
NOTE: After data is transmitted, the speaker amplifier should be turned off to avoid audible noise. This is needed because the EP7209 will continue to transmit data from the FIFO even though it is empty, thus causing noise. This will occur even when receiving.

3.7.2 I^{2}S Interface

The I^{2}S interface provides a high quality digital audio connection to I^{2}S compatible audio devices. The I^{2}S is a digital audio format that is supported by a number of manufacturers.

The I^{2}S 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 either fixed at 9.216 MHz or specified externally by an externally supplied MCLK signal.

The I^{2}S 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.



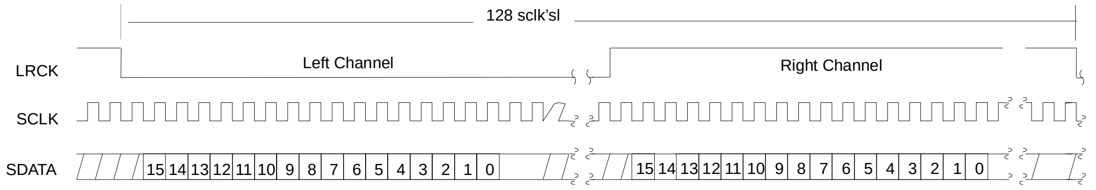
I2S INTERFACE

3.7.2.1 I^{2}S Operation

Following reset, the I^{2}S logic is disabled. To enable the I^{2}S, 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 (during a *I2SSR* register write). Next, the I^{2}S 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 interrrupt service routine which handles the I2S FIFOs. Finally, the FIFOs for each channel must be enabled via writes to I2SDR2. At this point, transmission/reception of data begins on the transmit (**SDOUT**) and receive (**SDIN**) pins. This is synchronously controlled by the 9.216 MHz (6.5 MHz in 13 MHz mode) or external serial clock (**SCLK**) and serial frame (**LRCK**) pins.

3.7.2.2 I^{2}S Frame Format

Each I^{2}S frame is 128 bits long and is comprises one audio sample. Of this 128-bit frame, only 32 bits are actually used for digital audio data. The remaining bits are ignored. The **LRCK** signal is used as a frame synchronization signal. 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 left side of an audio sample. When **LRCK** transitions from low to high the next 16-bits make up the right side of an audio sample.



3.7.2.3 I^{2}S Signals

MCLK – oversampled clock. Used as an input to the EP7209 for generating the I2S 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.
SCLK – bit clock. Used as the bit clock input into the DAC/ADC. This signal is fixed at 128 times the audio sample frequency.
LRCK – frame sync. 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 – digital audio data out. Used for sending playback data to a DAC. This signal is clocked out on the negative going edge of the **SCLK** output.

SDIN – digital audio input. Used for receiving record data from an ADC. This signal is latched by the EP7209 on the positive going edge of **SCLK**.

3.7.3 ADC Interface — Master Mode Only SSI1 (Synchronous Serial Interface)

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 be connected directly to the EP7209.
- 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, which will be slightly different depending whether the device is operating in a 13 MHz mode or a 18.432 MHz–73.728 MHz mode (see **Table 3-11. ADC Interface Operation Frequencies**). 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).

Table 3-11. ADC Interface Operation Frequencies

SYSCON1 Bit 17	SYSCON1 Bit 16	13.0 MHz Operation ADCCLK Frequency (kHz)	18.432–73.728 MHz Operation ADCCLK Frequency (kHz)
0	0	4.2	4
0	1	16.9	16
1	0	67.7	64
1	1	135.4	128

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.

3.7.4 Master/Slave SSI2 (Synchronous Serial Interface 2)

A second SPI/Microwire interface with full master/slave capability is provided by the EP7209. Data rates in slave mode are theoretically up to 512 kbps, full duplex, although continuous operation at this data rate will give an interrupt rate of 2 kHz, which is too fast for many operating systems. This would require a worst case interrupt response time of less than 0.5 ms 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 kbps, the sustained data rate would in fact only be 85.3

kbps (i.e., 1 byte every 750 µs). At this data rate, the required interrupt rate will be greater than the 1 ms, 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, to save power it can be disabled by writing a zero to the SS2TXEN and the SS2RXEN bits also within the system control register 2 (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 mode, is shown in **Figure 3-3. SSI2 Port Directions in Slave and Master Mode**.

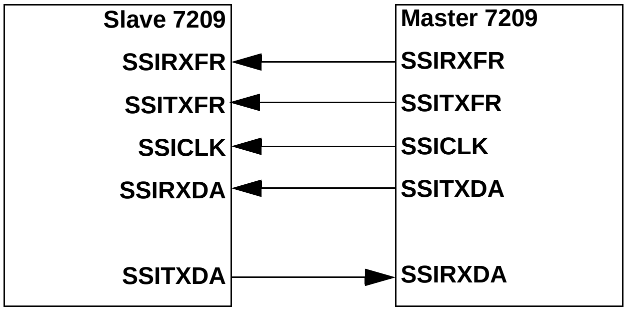


Figure 3-3. SSI2 Port Directions in Slave and Master Mode

Data on the link is sent MSB first and coincides with an appropriate frame sync pulse, of one clock in duration, located one clock prior to the first data bit sent (i.e., 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 kbs. The interface will support continuous transmission at this rate assuming that the OS can respond to the interrupts within 1 ms 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 the **AC Characteristics** section of this document.

3.7.4.1 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, so in the case of a single residual byte remaining at the end of a transmission, it will be necessary for the software 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 3-4. Residual Byte Reading 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)...

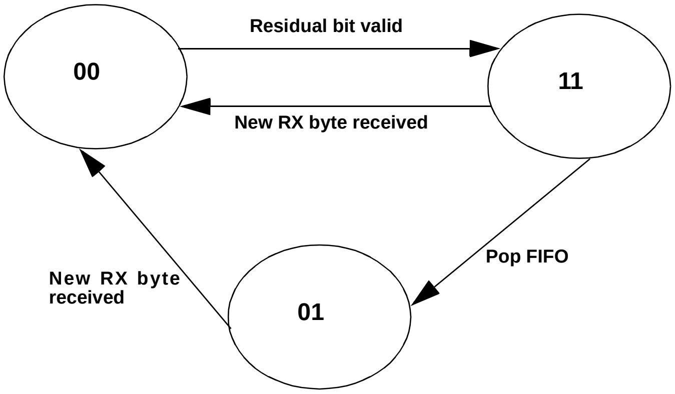


Figure 3-4. Residual Byte Reading

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.

3.7.4.2 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. 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 sufficiently quick.

3.7.4.3 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 in advance of the MSB).

3.7.4.4 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 design does not support the discontinuous clock.

3.7.4.5 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.

3.7.4.6 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.

3.8 LCD Controller with Support for On-Chip Frame Buffer

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 0x0000000 within each chip select. The start address of the LCD video frame buffer is defined in the FBADDR register bits [3:0]. These bits become the most significant nibble of the external address bus. The default start address is 0xC000 0000 (FBADDR = 0xC). A system can be 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 for the software 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.

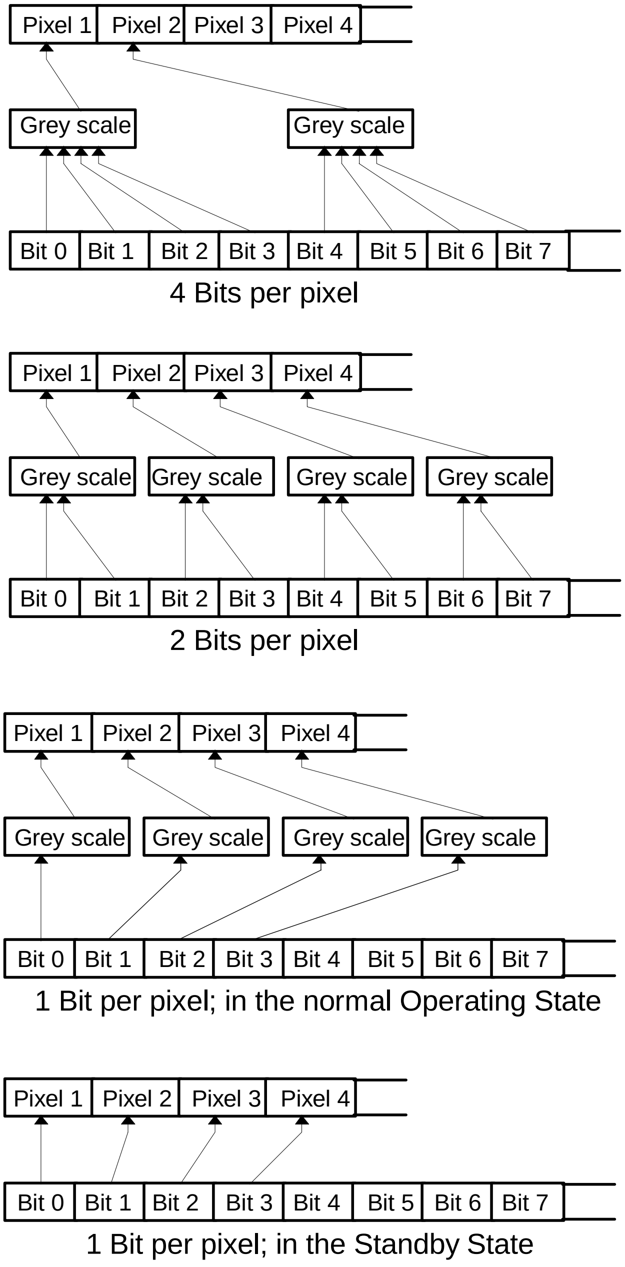
The screen is mapped to the video frame buffer as one contiguous block where each horizontal line of pixels is mapped to a set of consecutive bytes or words in the video RAM. The video frame buffer can be accessed word wide as pixel 0 is mapped to the LSB in the buffer such that the pixels are arranged 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 when operating in 18.432–73.728 MHz mode, or 13 MHz to 203 kHz when operating from a 13 MHz clock. 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 16 grey scale values available. 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, or 29.7 Mbytes/sec at 13 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 approx. 3.2 µs (640 x 240 x 4bpp x 80 Hz/8 bits/byte x 5 words (=20 bytes) = 3.2 µs). 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 higest 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, the worst case latency is almost exactly 3.2 µs, with 13 MHz page mode cycles. With each cycle consuming ~77 ns (i.e., 1/13 MHz), the value of 3.2 µs comes from 42 cycles x 77 ns/cycle = ~3.23 µs. If 16-bit wide, then the worst case latency will double. In this case, the maximum permissible display size will be halved, to approx. 320 x 240 pixels, assuming the same pixel depth and refresh rate has to be maintained. If the frame buffer is to be stored in static memory, then further calculations must be performed. If 18 MHz mode is selected, and 32-bit wide, then the worst case latency will be 2.26 µs (i.e., 42 cycles x 54 ns/cycle). If 36 MHz mode is selected, and 32-bit wide, then the worst case latency drops down to 1.49 µs. 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 µs.

Figure 3-5. Video Buffer Mapping shows the organization of the video map for all combinations of bits per pixel.



4 Bits per pixel

2 Bits per pixel

1 Bit per pixel; in the normal Operating State

1 Bit per pixel; in the Standby State

Figure 3-5. Video Buffer Mapping

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.

3.9 Internal UARTs (Two) and SIR Encoder

The EP7209 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 Kbps 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 EP7209. 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 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 EP7209 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 in to 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 take appropriate action in 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 kbps downwards. The master clock frequency is chosen so that most of the required data rates are obtainable exactly. When operating with a 13.0 MHz external clock source, the baud rates generated will have a slight error, which is less than or equal to 0.75%. The rates obtainable from the 13 MHz clock include 9.6 k, 19.2 k, 38 k, 58 k and 115.2 kbps. See **Section 5.9.2** in the register programming section of this spec for full details of the available bit rates in the 13 MHz mode.

3.10 Timer Counters

Two identical timer counters are integrated into the EP7209. 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 various bits in the system control register. When run from the internal PLL, 512 kHz and 2 kHz rates are provided. When using the 13 MHz external source, the default frequencies will be 541 kHz and 2.115 kHz, respectively. However, only in non-PLL mode, an optional divide by 26 frequency can be generated (thus generating a 500 kHz frequency when using the 13 MHz source). This divider is enabled by setting the OSTB (Operating System Timing Bit) in the SYSCON2 register (Bit 12). When this bit is set high to select the 500 kHz mode, the 500 kHz frequency is routed to the timers instead of the 541 kHz clock. This does not affect the frequencies derived for any of the other internal peripherals.

NOTE: When the EP7209 is in the Standby State, the timer counters are disabled.

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

3.10.1 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.

3.10.2 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.

3.11 Realtime Clock

The EP7209 contains a 32-bit realtime clock (RTC). This can be written to and read from in the same way as the timer counters but it 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 time in the RTC matches a specific time 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.

3.12 Dedicated LED Flasher

The LED flasher feature enables an external pin (PD[0]/LEDFLSH) to be toggled at a programmable rate and duty ratio — with the intention that the external pin is connected to an LED. This module is driven from the RTC's 32.768 kHz oscillator and allows it to work 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, 2/16, 3/16, …, 16/16 of the full cycle. The external pin can provide up to 4 mA of drive current.

3.13 Two PWM Interfaces

Two Pulse Width Modulator (PWM) duty ratio clock outputs are provided by the EP7209. When the device is operating from the internal PLL, this 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[0:1]** 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. 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.

3.14 State Control

The EP7209 supports the following Power Management States: Standby, Idle, and Operating.

In the description below, the **RUN/CLKEN** pin can be used either for the **RUN** functionality, or the **CLKEN** functionality to allow an external oscillator to be disabled in the 13 MHz mode. Either **RUN** or **CLKEN** functionality can be selected according to the state of the **CLKENSL** bit in the SYSCON2 register.

The Standby State equates to the system being switched "off" (i.e., no display, and the main oscillator is shut down). When the 18.432–73.728 MHz mode is selected, the PLL will be shut down. In the 13 MHz mode, if the CLKENSL bit is set low, then the **CLKEN** signal will be forced low and can, if required, be used to disable an external oscillator.

Table 3-12. Peripheral Status in Different Power Management States

Address (W/B)	Operating	Idle	Standby	NPOR RESET	NRESET RESET
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
I^{2}S 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

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 when in the Standby State. This signal can be used externally in the system to power down other system modules.

Whenever the EP7209 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.

When first powered, or reset by the **NPOR** (Not Power On Reset) signal, the EP7209 is forced into the Standby State. This is known as a cold reset, and is the only completely asynchronous reset to the EP7209. When leaving the Standby State after a cold reset, external wake up is the only way for waking 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 check to ensure that they are idle before issuing the write to the Standby State location.

Before entering the Standby State, the software must properly disable the I^{2}S. Failing to do so will result in higher than expected power consumption in the Standby State, as well as unpredictable operation of the I^{2}S. The I^{2}S 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 and either the **NEXTPWR** input pin is low or the **BATOK** input pin is high. This prevents the system from starting when the power supply is inadequate (e.g., 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 EP7209 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 first clocked. This is to allow the clock to the CPU to settle.

During the Standby State, the UARTs are disabled and cannot detect any activity (e.g., 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 EP7209 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 get captured correctly.

If in the Operating State, the Idle State can be entered by writing to a special internal memory location in the EP7209. If an interrupt or wakeup event occurs, the EP7209 will return immediately back to the Operating Stateand execute the next instruction.

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, or a rising edge on the external **WAKEUP** pin. The PLL (in 18.432–73.728 MHz mode) or the external 13 MHz clock source always remains active in the Idle State.

Figure 3-6. State Diagram shows a state diagram for the EP7209..

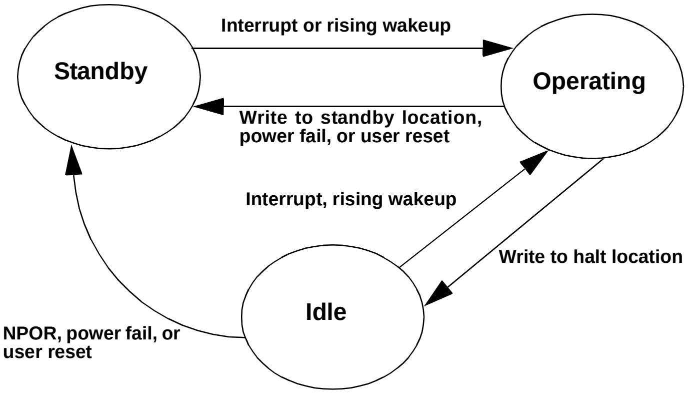


Figure 3-6. State Diagram

3.15 Resets

There are three asynchronous resets to the EP7209, these are **NPOR**, **NPWRFL** and **NURESET**. If any of these is active, a system reset is generated internally. This will clear all internal registers in the EP7209 to zero and the RTC data and match registers, which are only cleared by an active **NPOR)**. This ensures that the system time is preserved through a user reset or power fail condition. Any reset will also reset the CPU and cause it to start execution at the reset vector when the EP7209 returns to the Operating State.

Internal to the EP7209, three different signals are used to reset storage elements. These are **NPOR**, **NSYSRES** and **NSTBY**. **NPOR** is an external signal. **NSTBY** is equivalent to the external **RUN** signal.

NPOR (Not Power On Reset) is the highest priority reset signal. When active (low), it will reset all storage elements in the EP7209. **NPOR** active forces **NSYSRES** and **NSTBY** active. **NPOR** will only be active after the EP7209 is first powered up and not during any other resets. **NPOR** active will clear all flags in the status register apart from the cold reset flag (CLDFLG) bit, which is set.

NSYSRES (Not System Reset) is generated internally to the EP7209 if **NPOR**, **NPWRFL** or **NURESET** are active. It is the second highest priority reset signal, used to asynchronously reset most internal registers in the EP7209. **NSYSRES** active forces **NSTBY** and **RUN** low. **NSYSRES** is used to reset the EP7209 and force it into the Standby State with no co-operation from software. The CPU is also reset.

The **NSTBY** and **RUN** signals are high when the EP7209 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). In general, a system reset will clear all registers and **NSTBY** will disable all peripherals that require a main clock. 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.

When operating from an external 13 MHz oscillator which has become disabled in the Standby State by using the **CLKEN** signal (i.e., with CLKENSL = 0), the oscillator must be stable within 0.125 s from the rising edge of the **CLKEN** signal.

3.16 Clocks

There are two clocking modes for the EP7209. Either an external clock input can be used or the on-chip PLL. The clock source is selected by a strapping option on Port E, Pin 2 (**PE[2]**). If **PE[2]** is high at the rising edge of **NPOR** (i.e., upon power-up), the external clock mode is selected. If **PE[2]** is low, then the on-chip PLL mode is selected.

The architecture for the EP7209 device contains several separate sections of logic. Some of them are the ARM720T processor, the peripherals, and the address/data bus sections. The peripherals section has its own dedicated bus structure called the Embedded Peripheral Bus (EPB). From the ARM720T's point of view, the peripherals are strobed devices. Each unique peripheral device does have its own dedicated clock source supplied to it according to its clock frequency requirements. However, when the EP7209 is in the PLL mode, the actual frequencies will be different than when in the external clock mode. See each peripheral device section for more details. The section below describes the clocking for both the ARM720T and address/data bus sections.

3.16.1 On-Chip PLL

The ARM720T clock can be programmed to run at 18.432 MHz, 36.864 MHz, 49.152 MHz or 73.728 MHz. The PLL frequency is fixed at twice the highest possible CPU clock frequency (147.456 MHz). The PLL uses an external 3.6864 MHz crystal. On power-up, if the device has been strapped to use the on-chip PLL, then the device will start up with both the ARM720T and address/data buses configured to run at 18.432 MHz by default. After power-up, the clock frequency can be changed by software. If the clock frequency is changed to be 36 MHz, both the ARM720T and the address/data buses will be clocked at 36 MHz as well. 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 individual dedicated peripheral clock sources. In other words, all the peripheral clocks are fixed, regardless of the PLL mode clock speed selected for the ARM720T.

When the PLL mode is selected, the **EXPCLK** pin becomes an output. This output signal can then be used to supply a synchronized clock source to external devices (e.g., a CL-PS6700 PC Card controller device). When in the PLL mode, **PE[2]** can be used as a GPIO after power-up.

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

3.16.2 External Clock Input (13 MHz)

An external 13 MHz crystal oscillator can be used to drive all of the EP7209. On power-up, if the device has been strapped to use the external clock, the ARM720T and the address/data buses both get clocked at 13 MHz. The fixed clock sources to the various peripherals will have different frequencies than in the PLL mode. In this configuration, the PLL will not be used at all. The clock signal should be connected to the **EXPCLK** pin of the EP7209. In this mode, **EXPCLK** becomes an input.

If the CLKENSL bit is set low, then pin **CLKEN/RUN** (an output pin) provides the **CLKEN** signal that can be used to start and stop the external 13 MHz clock source (i.e., the oscillator) used to supply the clock for the EP7209 and the PWM. In this case, when **CLKEN** is active (high), it can be used to enable the external 13 MHz clock source, and when low, it can be used to disable it.

When the Standby State is entered in 13 MHz mode, the 13 MHz source is disabled at the **EXPCLK** input pin pad until the Standby State is exited. Therefore, the external clock source does not have to be disabled to be able to disable the clocks internally.

When in the 13 MHz mode, if the CLKENSL bit is set high, then the Standby State is exited immediately on a wakeup event or an enabled interrupt occurring. If CLKENSL is set low, then the EP7209 will set **CLKEN** high after an interrupt or wakeup event occurs, then wait for between 0.125 s and 0.25 s to allow an external oscillator time to stabilize before the clock is enabled through to the CPU. Only a non-masked interrupt (e.g., realtime clock match), Port A event (corresponding to a keypress), or **WAKEUP** signal asserted will wakeup the EP7209 out of the Standby State. The timing is shown below for the case where CLKENSL is low:

NOTE: When operating at 13 MHz, the CLKCTL[1:0] bits should not be changed from their default value of ‘00’.

CLKEN timing while entering STDBY mode

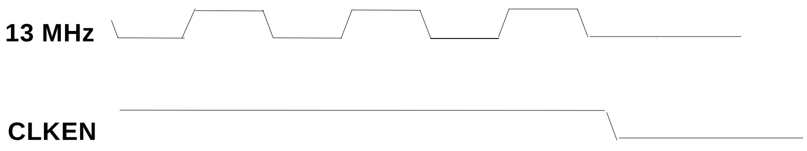


Figure 3-7. CLKEN Timing Entering the Standby State

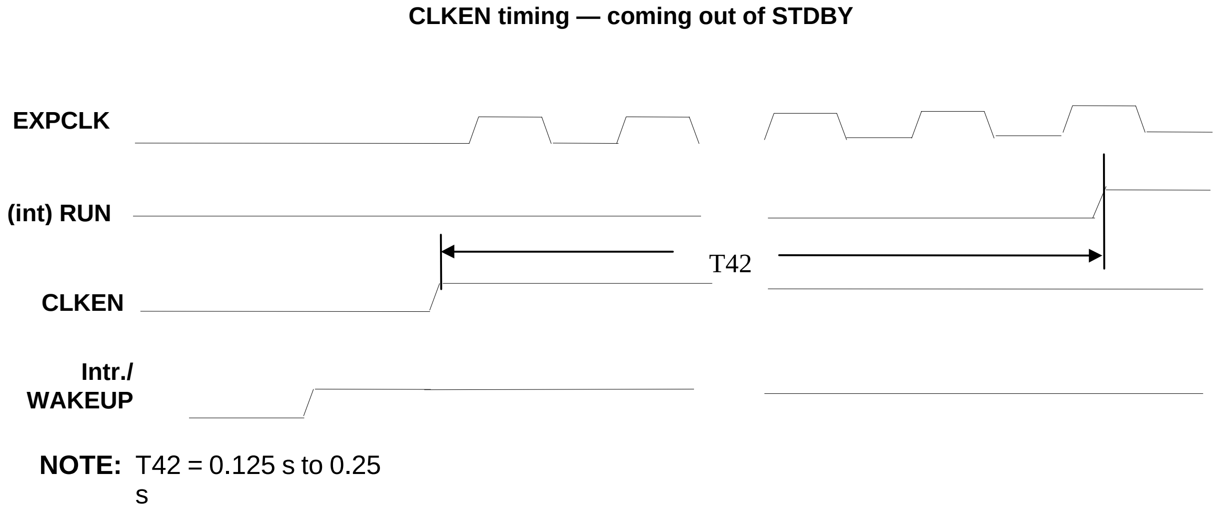

CLKEN timing — coming out of STDBY
EXPCLK
(int) RUN
CLKEN
T42
Intr./
WAKEUP
NOTE: T42 = 0.125 s to 0.25 s

Figure 3-8. CLKEN Timing Leaving the Standby State

3.17 Dynamic Clock Switching When in the PLL Clocking Mode

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 occurs, the state controller switches from the current to the new clock frequency as soon as possible without causing a glitch on the clock signals. The glitch-free 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 to the new clock without a glitch on the clocks.

3.18 Endianness

The EP7209 uses a Little Endian configuration for internal registers. However, it is possible to connect the device to a Big Endian external memory system. The bigend bit in the ARM720T control register sets whether the EP7209 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 data lines 7 through 0 (D[7:0]) in this scheme. 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 data lines 31 through 24 (D[31:24]). Load and store are the only instructions affected by the Endianness.

The following tables demonstrate the behavior of the EP7209 in Big and Little Endian mode, including the effect of performing non-aligned word accesses. The register definition section of this specification defines the behavior of the internal EP7209 registers in the Big Endian mode in more detail.

Table 3-13. Effect of Endianness on Read Operations

Address (W/B)	Data in Memory	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 (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 3-14. Effect of Endianness on Write Operations

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 (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.

3.19 Maximum EP7209-Based System

A maximum configured system using the EP7209 is shown in **Figure 3-9. A Maximum EP7209 Based System**. This system assumes the 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.

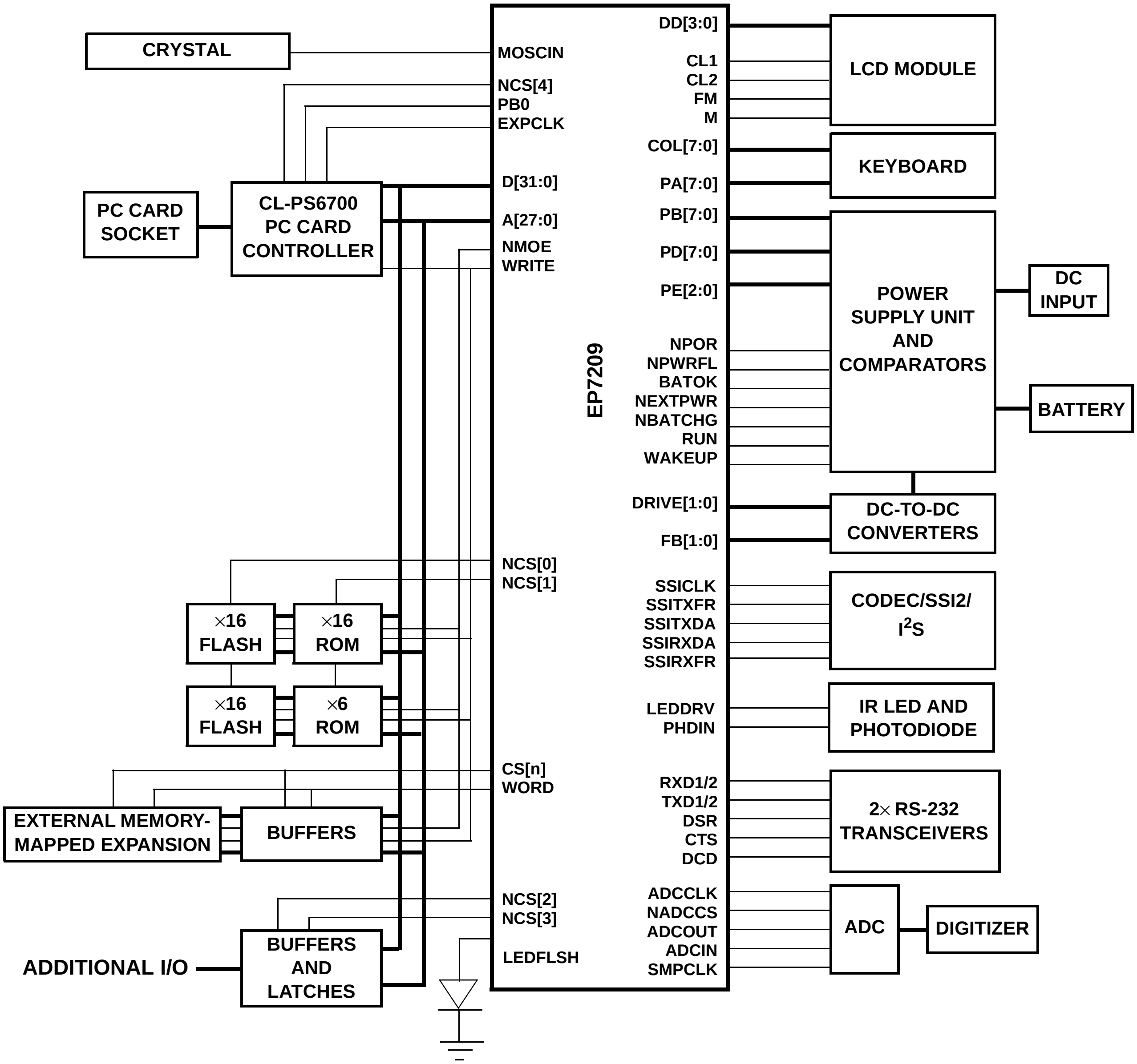


Figure 3-9. A Maximum EP7209 Based System

NOTE: A system can only use one of the following peripheral interfaces at any given time: SSI2, codec, or I^{2}S.

3.20 Boundary Scan

IEEE 1149.1 compliant JTAG is provided with the EP7209. (It should be noted that a full system reset, BNRES[0] = ’0’, must be applied to the device initially.)

Table 3-15. Instructions Supported in JTAG Mode

Instruction	Code
BYPASS	1111
IDCODE	1110
SAMPLE/PRELOAD	0011
EXTEST	0000
SCAN_N	0010

The INTEST function will not be supported for the EP7209.

Additional user-defined instructions exist, but these are not relevant to board-level testing. For further information please refer to the **ARM DDI 0087E** *ARM720T Datasheet* **Section 8.**

As there are additional scan-chains within the ARM720T processor, it is necessary to include a scan-chain select function — shown as SCAN_N in **Table 3-15. Instructions Supported in JTAG Mode**. To select a particular scan chain, this function must be input to the TAP controller, followed by the 4-bit scan-chain identification code. The identification code for the boundary scan chain is 0011.

Note that it is only necessary to issue the SCAN_N instruction if the device is already in the JTAG mode. The boundary scan chain is selected as the default on test-logic reset and any of the system resets.

The contents of the device ID-register for the EP7209 are shown below:

Version				Part number																Manufacturer ID											
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

This is equivalent to 0F0F0F0F Hex. Note this is the ID-code for the ARM720T processor.

3.21 In-Circuit Emulation

3.21.1 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, to provide access the extensions from the outside world
3) The EmbeddedICE interface to provide communication from the EmbeddedICE macrocell and the host computer

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.

3.21.2 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 Datasheet,* **Chapters 8 and 9**.

4. MEMORY MAP

The lower 2 GByte of the address space is allocated to ROM/SRAM/Flash and expansion space. The 2 GByte less 8K for internal registers is not accessible in the EP7209. The MMU in the EP7209 should be programmed to generate an abort exception for access to this area.

Internal peripherals are addressed through a set of internal memory locations from hex address 8000.0000 to 8000.3FFF. These are known as the internal registers in the EP7209.

Figure 4-1 shows how the 4-Gbyte address range of the ARM720T processor (as configured within this chip) is mapped in the EP7209. The memory map shown assumes that two CL-PS6700 PC Card controllers are connected. If this functionality is not required, then the **NCS4** and **NCS5** memory is available as general ROM/SRAM/Flash/Expansion space. The Boot ROM is not fully decoded (i.e., the Boot code will repeat within the 256-Mbyte space from 0x70000000 to 0x80000000). The SRAM is fully decoded up to a maximum size of 128 kbytes. Access to any location above this range will be wrapped to within the range.

Table 4-1. EP7209 Memory Map

Address	Contents	Size
0xF000.0000	Reserved	256 Mbytes
0xE000.0000	Reserved	256 Mbytes
0xD000.0000	Unused	256 Mbytes
0xC000.0000	Unused	256 Mbytes
0x8000.4000	Unused	~1 Gbyte
0x8000.2000 0x8000.0000	Internal registers (new) Internal registers (old)	8 kbytes 8 kbytes
0x7000.0000	Boot ROM (NCS7)	128 bytes
0x6000.0000	SRAM (NCS6)	37.5 kbytes
0x5000.0000	PCMCIA-1 (NCS5)	4 x 64 Mbytes
0x4000.0000	PCMCIA-0 (NCS4)	4 x 64 Mbytes
0x3000.0000	Expansion (CS3)	256 Mbytes
0x2000.0000	Expansion (CS2)	256 Mbytes
0x1000.0000	ROM Bank 1 (CS1)	256 Mbytes
0x0000.0000	ROM Bank 0 (CS0)	256 Mbytes

5. REGISTER DESCRIPTIONS

5.1 Internal Registers

Table 5-1. Internal I/O Memory Locations shows all internal registers in the EP7209, assuming the CPU is configured for operation with a Little Endian memory system. **Table 5-2.Port Byte Addresses in Big Endian Mode** shows the differences that occur for byte-wide access to Ports A, B, and D with the CPU configured to operate in the Big Endian mode. 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. Note that the internal registers have been split into two groups – the “old” and the “new”. The old ones are the same as that used in CL-PS7111 and are there for compatibility. The new registers are for accessing the additional functionality of the I^{2}S interface and the LED flasher.

There is no effect on the register addresses for word accesses. Bits A0 and A1 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 A0 and A1 are not decoded, so that byte reads will return the whole register contents onto the EP7209’s internal bus, from where the appropriate byte (according to the Endianness) will be read by the CPU. 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 word accesses.

An 8K segment of memory in the range 0x8000.0000 to 0x8000.3FFF is reserved for internal use in the EP7209. Accesses in this range will not cause any external bus activity unless debug mode is enabled. 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 A0 –A5 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 for backward compatibility reasons, 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 A0 and A1. All addresses are in hexadecimal notation.

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 registers 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 (e.g., RTC data registers, SYSFLG register (lower 6-bits only), the TC1 and 2 data registers, port registers, interrupt status registers) should be read twice and compared to ensure that a stable value has been read back.

Table 5-1. CL-PS7111-Compatible

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	SYSFLG	0	RD	32	System status flags register 1
0x8000.0180	MEMCFG1	0	RW	32	Expansion and ROM memory configuration register 1
0x8000.01C0	MEMCFG2	0	RW	32	Expansion and ROM memory configuration register 2
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 data to TC1
0x8000.0340	TC2D	0	RW	16	Read/Write data to TC2
0x8000.0380	RTCDR	—	RW	32	Realtime clock data register
0x8000.03C0	RTCMR	—	RW	32	Realtime 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	8W/11R	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	PALMSW	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

Table 5-1. CL-PS7111-Compatible *(cont.)*

Address	Name	Default	RD/WR	Size	Comments
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
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 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	C	RW	4	LCD frame buffer start address
0x8000.1100	SYSCON2	0	RW	16	System control register 2
0x8000.1140	SYSFLG2	0	RD	16	System status register 2
0x8000.1240	INTSR2	0	RD	24	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	8W/11R	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 savings state.
0x8000.1840– 0x8000.1FFF	Reserved	—			Write will have no effect, read is undefined

Table 5-2. Internal I/O Memory Locations (EP7209 Only)

Address	Name	Default	RD/WR	Size	Comments
0x8000.2000	I^{2}SCR	—	RW	32	I^{2}S Control Register
0x8000.2040	I^{2}SDR0	0	RW	32	I^{2}S Data Register0
0x8000.2080	I^{2}SDR1	0	RW	32	I^{2}S Data Register1
0x8000.20C0	I^{2}SDR2	0	RW	32	I^{2}S Data Register2
0x8000.2100	I^{2}SSR	—	RD	32	I^{2}S Status Register
0x8000.2200	SYSCON3	0	RW	8	System control register 3
0x8000.2240	INTSR3	0	RD	8	Interrupt status register 3
0x8000.2280	INTMR3	0	RW	8	Interrupt mask register 3
0x8000.22C0	Ledflsh	0	RW	7	LED Flash control Register
0x8000.2300– 0xBFFF.FFFF	Reserved				This area contains test registers used during manufacturing test. Writes to this area should never be attempted during normal operation as this may cause unexpected behavior. Reads will be undefined.

Table 5-3. Port Byte Addresses in Big Endian Mode

Address	Name	Default	RD/WR	Size	Comments
0x8000.0003	PADR	0	RW	8	Port A data register
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

All internal registers in the EP7209 are reset (cleared to zero) by a system reset (i.e., **NPOR**, **NRESET**, or **NPWRFL** signals becoming active), and the realtime clock data register (RTCDR) and match register (RTCMR), which are only reset by **NPOR** becoming active. This ensures that the system time preserved through a user reset or power fail condition.

5.1.1 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.

5.1.2 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.

5.1.3 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.

5.1.4 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.

5.1.5 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.

5.1.6 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.

5.1.7 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.

5.1.8 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, clearing bit sets the pin to input. All bits are cleared by a system reset so that Port E is input by default.

5.2 SYSTEM Control Registers

5.2.1 SYSCON1 The System Control Register 1

ADDRESS: 0x8000.0100

7	*6*	*5*	*4*	*3:0*
TC2S	TC2M	TC1S	TC1M	Keyboard scan

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*
SIREN	CDENRX	CDENTX	LCDEN	DBGEN	BZMOD	BZTOG	UART1EN

23	*22*	*21*	*20*	*19*	*18*	*17:16*
			IRTXM	WAKEDIS	EXCKEN	ADCKSEL

The system control register is a 21-bit read/write register which controls all the general configuration of the EP7209, 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 below.

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

Bit	Description
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 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 for more details.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>16:17</td><td><b>ADCKSEL</b>: Microwire / SPI peripheral clock speed select. This two bit field selects the frequency of the ADC sample clock; this 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. The frequencies obtained in 13.0 MHz mode can be found in <b>Table 3-15. ADC Interface Operation Frequencies</b>..
<table>
<tr><th>ADCKSEL</th><th>ADC Sample Frequency
(kHz) — SMPCLK</th><th>ADC Clock Frequency
(kHz) — ADCCLK</th></tr>
<tr><td>00</td><td>8</td><td>4</td></tr>
<tr><td>01</td><td>32</td><td>16</td></tr>
<tr><td>10</td><td>128</td><td>64</td></tr>
<tr><td>11</td><td>256</td><td>128</td></tr>
</table></td></tr>
<tr><td>11</td><td><b>DBGEN</b>: Setting this bit will enable the debug mode. In this mode, all internal accesses are output as if they were reads or writes to expansion memory addressed by <b>NCS5</b>. <b>NCS5</b> 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 enable individual accesses to be distinguished. For example, in debug mode:
NCS5 = NCS5 or internal I/O strobe
PE0 = CLK
PE1 = NIRQ
PE2 = NFIQ</td></tr>
<tr><td>12</td><td><b>LCDEN</b>: LCD enable bit. Setting this bit enables the LCD controller.</td></tr>
<tr><td>13</td><td><b>CDENTX</b>: Codec interface enable TX bit. Setting this bit enables the codec interface for data transmission to an external codec device.</td></tr>
<tr><td>14</td><td><b>CDENRX</b>: Codec interface enable RX bit. Setting this bit enables the codec interface for data reception from an external codec device.
<b>NOTE:</b> Both CDENRX and CDENTX need to be enabled/disabled in TANDEM.</td></tr>
<tr><td>15</td><td><b>SIREN</b>: HP SIR protocol encoding enable bit. This bit will have no effect if the UART is not enabled.</td></tr>
<tr><td>18</td><td><b>EXCKEN</b>: External expansion clock enable. If this bit is set, the <b>EXPCLK</b> is enabled continuously; it is the same speed and phase as the CPU clock and will free run all the time the main oscillator is running if this bit is set. This bit should not be left set all the time for power consumption reasons. If the system enters the Standby State, the <b>EXPCLK</b> will become undefined. If this bit is clear, <b>EXPCLK</b> will be active during memory cycles to expansion slots that have external wait state generation enabled only.</td></tr>
<tr><td>19</td><td><b>WAKEDIS</b>: Setting this bit disables waking up from the Standby State, via the wakeup input.</td></tr>
<tr><td>20</td><td><b>IRTXM</b>: IrDA TX mode bit. This bit controls the IrDA encoding strategy. Clearing this bit means 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 µsw regardless of the selected bit rate). Setting this bit will use less power, but will probably reduce transmission distances.</td></tr>
</table>

5.2.2 SYSCON2 System Control Register 2

ADDRESS: 0x8000.1100

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*
SS2RXEN	PC CARD2	PC CARD1	SS2TXEN	KBWEN	DRAMSZ	KBD6	SERSEL

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*
Reserved	BUZFREQ	CLKENSL	OSTB	Reserved	Reserved	SS2MAEN	UART2EN

This register is an extension of SYSCON1, containing additional control for the EP7209, for compatibility with CL-PS7111. 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 MCPSEL bit of SYSCON3 is set, then it overrides the state of the SERSEL bit, and thus the external pins are connected to the I^{2}S 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.
3	**KBWEN**: When the KBWEN bit is high, the EP7209 will get woken 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 receive side is low.
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.

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

Bit	Description
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.
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.
12	**OSTB**: This bit (operating system timing bit) is for use only with the 13 MHz clock source mode. Normally it will be set low, however when set high it will cause a 500 kHz clock to be generated for the timers instead of the 541 kHz which would normally be available. The divider to generate this frequency is not clocked when this bit is 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 fixed frequency clock is output (500 Hz when running from the PLL, 528 Hz in the 13 MHz external clock mode). See the BZMOD and the BZTOG bits for more details.

5.2.3 SYSCON3 System Control Register 3

ADDRESS: 0x8000.2200

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Must be one	FASTWAKE

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*
VERSN[2]R eserved	VERSN[1] Reserved	VERSN[0] Reserved	ADCCKNSEN	MCPSEL	CLKCTL1	CLKCTL0	ADCCON

This register is an extension of SYSCON1 and SYSCON2, containing additional control for the EP7209. The bits of this third system control register are defined below.

Bit	Description
8	**FASTWAKE**: When set, the device will wake from the Standby State within one to two cycles of a 4 kHz clock. This bit is cleared at power up, and thus the device first starts using the default one to two cycles of the 8 Hz clock.
5:7	**VERSN[2:0]**: Additional read-only version bits — will read ‘000’.
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.
3	**I^{2}SPSEL**: When set selects the I^{2}S Interface. This defaults to either the SSI (i.e., I^{2}SSEL bit is low).
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	ASB and APB 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 5-4. Values of the Wait State Field at 13 and 18MHz** and **Table 5-5. Values of the Wait State Field at 36 MHz**. When operating at 13 MHz, the CLKCTL[1:0] bits should not be changed from the default value of ‘00’. Under no circumstances should the CLKCTL bits be changed using a buffered write.

5.2.4 SYSFLG — The System Status Flags Register

ADDRESS: 0x8000.0140

7:4	3	2	1	0
DID	WUON	WUDR	DCDET	MCDR

15	14	13	12	11	10	9	8
CLDFLG	PFFLG	RSTFLG	NBFLG	UBUSY1	DCD	DSR	CTS

23	22	21:16
UTXFF1	URXFE1	RTCDIV

31:30	29	28	27	26	25	24
VERID	ID	BOOTBIT1	BOOTBIT0	SSIBUSY	CTXFF	CRXFE

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 SYSFLG are defined in the following table.

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. These bits identify the LCD display panel fitted.
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.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>14</td><td><b>PFFLG</b>: 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.</td></tr>
<tr><td>15</td><td><b>CLDFLG</b>: Cold start flag. This bit will be set if the EP7209 has been reset with a power on reset, it is cleared by writing to the STFCLR location.</td></tr>
<tr><td>16:21</td><td><b>RTCDIV</b>: 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.</td></tr>
<tr><td>22</td><td><b>URXFE1</b>: 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.</td></tr>
<tr><td>23</td><td><b>UTXFF1</b>: 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.</td></tr>
<tr><td>24</td><td><b>CRXFE</b>: Codec RX FIFO empty bit. This will be set if the 16-byte codec RX FIFO is empty.</td></tr>
<tr><td>25</td><td><b>CTXFF</b>: Codec TX FIFO full bit. This will be set if the 16-byte codec TX FIFO is full.</td></tr>
<tr><td>26</td><td><b>SSIBUSY</b>: 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.</td></tr>
<tr><td>27</td><td><b>BOOTBIT0–1</b>: These bits indicate the default (power-on reset) bus width of the ROM interface. See the memory configuration register for more details on the ROM interface bus width. The state of these bits reflect the state of Port E bits 0–1 during power on reset, as shown in the table below.
<table>
<tr><th>PE1 (Bootbit1)</th><th>PE0 (Bootbit0)</th><th>Boot option</th></tr>
<tr><td>0</td><td>0</td><td>32-bit</td></tr>
<tr><td>0</td><td>1</td><td>8-bit</td></tr>
<tr><td>1</td><td>0</td><td>16-bit</td></tr>
<tr><td>1</td><td>1</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>28</td><td><b>ID</b>: Will always read ‘1’ for the EP7209 device.</td></tr>
<tr><td>30:31</td><td><b>VERID</b>: Version ID bits. These 2 bits determine the version id for the EP7209. Will read ‘10’ for the initial version.</td></tr>
</table>

5.2.5 SYSFLG2 System Status Register 2

ADDRESS: 0x8000.1140

7	6	5	4	3	2	1	0
Reserved	CKMODE	SS2TXUF	SS2TXFF	SS2RXFE	RESFRM	RESVAL	SS2RXOF

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	Reserved	UBUSY2	Reserved	Reserved	Reserved

23	22	21:16
UTXFF2	URXFE2	Reserved

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 below.

Bit	Description
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.
3	**SS2RXFE**: Master/slave SSI2 RX FIFO empty bit. This will be set if the 16 x 16 RX FIFO is empty.
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.
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 EP7209 is reset.
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.
6	**CKMODE**: This bit reflects the status of the CLKSEL (Port E Bit 2) input, latched during **NPOR**. When low, the PLL is running and the chip is operating in 18.432–73.728 MHz mode. When high the chip is operating from an external 13 MHz clock.

5.3 Interrupt Registers

5.3.1 INTSR1 Interrupt Status Register 1

ADDRESS: 0x8000.0240

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*
EINT3	EINT2	EINT1	CSINT	MCINT	WEINT	BLINT	EXTFIQ

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*
SSEOTI	UMSINT	URXINT1	UTXINT1	TINT	RTCMI	TC2OI	TC1OI

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 EP7209. Each bit is set if the appropriate interrupt is active. The interrupt assignment is given below.

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 de-glitched with a 16 kHz clock, so it will only generate an interrupt if it is active for longer than 61 µs. 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 the Standby State.
2	**WEINT**: 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 NOTE: WEINT is disabled during the Standby State. Watch dog timer tick rate is 64 Hz (in 13 MHz and 73.728–18.432 MHz modes). **NOTE:** Watchdog timer is turned off during the Standby State.
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 de-glitched with a 16 kHz clock so it will only generate an interrupt if it is active for longer than 61 µs. 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.

Bit	Description
7	**EINT3**: External interrupt input 3. This interrupt will be active if the **EINT3** input is active (high) 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.
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 realtime 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 6 4Hz clock is derived from the 15-stage ripple counter that divides the 32.768 kHz oscillator input down to 1 Hz for the realtime clock. This interrupt is cleared by writing to the TEOI location. NOTE: TINT is disabled/turned off during the Standby State.
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.

5.3.2 INTMR1 Interrupt Mask Register 1

ADDRESS: 0x8000.0280

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*
EINT3	EINT2	EINT1	CSINT	MCINT	WEINT	BLINT	EXTFIQ

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*
SSEOTI	UMSINT	URXINT	UTXINT	TINT	RTCMI	TC2OI	TC1OI

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 EP7209. 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 the INTSR1 register for individual bit details

5.3.3 INTSR2 Interrupt Status Register 2

ADDRESS: 0x8000.1240

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Reserved	Reserved	Reserved	Reserved	Reserved	SS2TX	SS2RX	KBDINT

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*
Reserved	Reserved	URXINT2	UTXINT2	Reserved	Reserved	Reserved	Reserved

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 EP7209. Each bit is set if the appropriate interrupt is active. The interrupt assignment is given below.

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 receive 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.

Bit	Description
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. It is cleared by reading all the data from the RX FIFO.

5.3.4 INTMR2 Interrupt Mask Register 2

ADDRESS: 0x8000.1280

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	SS2TX	SS2RX	KBDINT

15	14	13	12	11	10	9	8
Reserved	Reserved	URXINT2	UTXINT2	Reserved	Reserved	Reserved	Reserved

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.

5.3.5 INTSR3 Interrupt Status Register 3

ADDRESS: 0x8000.2240

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	I2SINT

This register is an extension of INTSR1 and INTSR2 containing status bits for the new features of the EP7209. Each bit is set if the appropriate interrupt is active. The interrupt assignment is given below.

Bit	Description
0	**MCPINT**: I^{2}S interface interrupt. The cause must be determined by reading the I^{2}S status register. It is mapped to the FIQ interrupt on the ARM720T processor

5.3.6 INTMR3 Interrupt Mask Register 3

ADDRESS: 0x8000.2280

7	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	I^{2}SINT

This register is an extension of INTMR1 and INTMR2, containing interrupt mask bits for the new features of the EP7209. Please refer to INTSR3 for individual bit details.

5.4 Memory Configuration Registers

5.4.1 MEMCFG1 Memory Configuration Register 1

ADDRESS: 0x8000.0180

31 24	*23 16*	*15 8*	*7 0*
NCS3 configuration	NCS2 configuration	NCS1 configuration	NCS0 configuration

Expansion and ROM space is selected by one of eight chip selects. One of the chip selects (CS6) is used internally for the on-chip SRAM, and the configuration is hardwired for 32-bit wide, minimum wait state operation. CS7 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 CS6 or CS7 will be ignored. Two of the chip selects (**NCS4** and **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 **NCS4** and **NCS5** 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-width fields contained in two 32-bit registers, MEMCFG1 and MEMCFG2. All bits in these registers are cleared by a system reset (except for the CS6 and CS7 configurations).

The memory configuration register 1 is a 32-bit read/write register which sets the configuration of the four expansion and ROM selects **NCS0–NCS3**. Each select is configured with a 1-byte field starting with expansion select 0.

5.4.2 MEMCFG2 Memory Configuration Register 2

ADDRESS: 0x8000.01C0

31 *24*	*23* *16*	*15* *8*	*7* *0*
(Boot ROM)	(Local SRAM)	NCS5 configuration	NCS4 configuration

The memory configuration register 2 is a 32-bit read/write register which sets the configuration of the two expansion and ROM selects **NCS4–NCS5**. 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 **NCS0–3**, and **NCS4–5** 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.

7	*6*	*5* *2*	*1* *0*
CLKENB	SQAEN	Wait States Field	Bus width

Table 5-4.Values of the Bus Width Field 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 SYSFLG 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 EP7209 during power-on reset. The state of **PE1** and **PE0** determine whether the EP7209 is going to boot from either 32-bit wide, 16-bit wide or 8-bit wide ROMs.

Table 5-4. Values of the Bus Width Field

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

Table 5-4. Values of the Bus Width Field *(cont.)*

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

The table below show the values for the wait states for random and sequential wait states at 13 and 18 MHz bus rates.

Table 5-5. Values of the Wait State Field at 13 MHz and 18 MHz

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

At 36 MHz bus rate, the encoding becomes more complex:

Table 5-6. Values of the Wait State Field at 36 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

Table 5-6. Values of the Wait State Field at 36 MHz *(cont.)*

Bit 3	Bit 2	Bit 1	Bit 0	Wait States Random	Wait States Sequential
1	1	1	1	1	0

The previous table preserves compatibility with the previous devices, while allowing the previously unused bit combinations to specify more variations of random and sequential wait states.

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. This bit will only affect **EXPCLK** when the PLL is being used (i.e., in 73.728–18.432MHz mode). When operating in 13 MHz mode, the **EXPCLK** pin is an input so cannot be affected by this register bit. To save power internally, it should always be set to zero when operating in 13 MHz mode.

See the **AC Electrical Specification** section for more detail on bus timing.

The memory area decoded by CS6 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 **NCS4** and **NCS5** 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.

5.4.3 DRFPR DRAM Refresh Period Register

ADDRESS: 0x8000.0200

7	*6:0*
RFSHEN	RFDIV

The DRAM refresh period register is an 8-bit read/write register which enables refreshes and selects the refresh period used by the DRAM controller for its periodic CAS before RAS refresh. The value in the DRAM refresh period register is only cleared by a power-on reset (i.e., its state is maintained during a power fail or user reset).

Bit	Description
7	**RFSHEN**: DRAM Refresh enable. Setting this bit enables periodic refresh cycles to be generated by the EP7209 at a rate set by the RFDIV field. Setting this bit also enables self refresh mode when the EP7209 is in the Standby State.
0:6	**RFDIV**: This 7-bit field sets the DRAM refresh rate. The refresh period is derived from a 128 kHz clock and is given by the formula: Frequency (kHz) = 128 / (RFDIV + 1) i.e., RFDIV = (128 / Refresh frequency (kHz)) – 1 This equation is valid for both 13 MHz and 18.432–73.728 Mhz modes. The equation for frequency gives the refresh rate for the DRAM. The maximum refresh frequency is 64 kHz: the minimum is 1 kHz. The RFDIV field should not be programmed with zero as this will result in no refresh cycles being initiated. These values are valid for both 13 MHz and 18.432–73.728 MHz modes of operation.

5.5 Timer/Counter Registers

5.5.1 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.

5.5.2 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.

5.5.3 RTCDR Realtime 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**.

5.5.4 RTCMR Realtime Clock Match Register

ADDRESS: 0x8000.03C0

The Realtime 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**.

5.6 LEDFLSH Register

ADDRESS: 0x8000.22C0

The LEDFLSH register is as follows:

<table>
<tr><td>6</td><td>5</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Enable</td><td colspan="2">Duty ratio</td><td colspan="2">Flash rate</td></tr>
</table>

The output is enabled whenever LEDFLSH[6] = 1. When enabled, Port D’s Bit 0 Direction Register needs to be configured as an output pin and the bit cleared to ‘0’. 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:

Table 5-7. LED Flash Rates

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

Table 5-8. LED Duty Ratio

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

5.7 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).

Bit	Description
0:3	**Drive 0 from battery**: This 4-bit field controls the 'on' time for the drive0 PWM pump while the system is powered from batteries. 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, or 101.6 kHz when operating from the 13 MHz source.
4:7	**Drive 0 from AC**: This 4-bit field controls the 'on' time for the drive0 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, or 101.6 kHz when operating from the 13 MHz source. **NOTE:** The EP7209 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, 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, or 101.6 kHz when operating from the 13 MHz source.

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 5-9**.

Table 5-9. Sense of PWM control lines

Initial State of Driven During Power on Reset	Sense of Driven	Polarity of Bias Voltage
Low	Active high	+ve
High	Active low	-ve

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. They are read upon power-up. When **FB[0]** is high, the PWM is disabled. The same applies to **FB[1]**.

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 disable when the drive ratio fields are zeroed.

5.8 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 when 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 of the byte transfer rate and the state of the FIFOs can be read in the system flags register. The net data transfer rate to/from the codec device is 8 kBps, giving an interrupt rate of 1 kHz.

5.9 UART Registers

5.9.1 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. Data read from this register is popped from the 16 byte data RX FIFO, if the FIFO is enabled. If not it is read from a one byte buffer register containing the last byte received by the UART. Data received and error status is automatically pushed onto the RX FIFO if it is enabled. 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.

5.9.2 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. In 13 MHz mode, the clock frequency fed to the UART is 1.8571 MHz. In this mode, zero is a legal divisor value, and will generate the maximum possible bit rate. The tables below show the bit rates available for both 18.432 MHz and 13 MHz operation.

Table 5-10. UART Bit Rates Running from the PLL Clock

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

Table 5-11. UART Bit Rates Running from an External 13.0 MHz Clock

Divisor Value	Bit Rate at 13 Mhz	Error on 13 Mhz Value
0	116071	0.75%
1	58036	0.75%
2	38690	0.75%
5	19345	0.75%
7	14509	0.75%
11	9673	0.75%
47	2418	0.42%
96	1196	0.28%
1054	110.02	0.18%

Bit	Description
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.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>15</td><td><b>XSTOP</b>: Extra stop bit. Setting this bit will cause the UART to transmit two stop bits after each data byte, clearing it will transmit one stop bit after each data byte.</td></tr>
<tr><td>16</td><td><b>FIFOEN</b>: Set to enable FIFO buffering of RX and TX data. Clear to disable the FIFO (i.e., set its depth to one byte).</td></tr>
<tr><td>17:18</td><td><b>WRDLEN</b>: This two bit field selects the word length according to the table below.
<table>
<tr><th>WRDLEN</th><th>Word Length</th></tr>
<tr><td>00</td><td>5 bits</td></tr>
<tr><td>01</td><td>6 bits</td></tr>
<tr><td>10</td><td>7 bits</td></tr>
<tr><td>11</td><td>8 bits</td></tr>
</table>
</td></tr>
</table>

5.10 LCD Registers

5.10.1 LCDCON — The LCD Control Register

ADDRESS: 0x8000.02C0

31	*30*	*29:25*	*24:19*	*18:13*	*12:0*
GSMD	GSEN	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.

The LCDCON register should only be reprogrammed when 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 e.g., for a 640 x 240 LCD and 4-bits per pixel, the size of the video buffer = 640 x 240 x 4 = 614400 bits Video buffer size field = (614400 / 128) – 1 = 4799 or 0x12BF hex. 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 = (No. pixels in line / 16) – 1 e.g., 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).

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) When the EP7209 is operating at 13 MHz, pixel rate is given by the formula: Pixel rate (MHz) = 13 / (Pixel prescale + 1) The pixel prescale value can be expressed in terms of the LCD size by the formula: When the EP7209 is operating @ 18.432 MHz: Pixel prescale = (36864000 / (Refresh Rate *Total pixels in display)) – 1 When the EP7209 is operating @ 13 MHz: Pixel prescale = (13000000 / (Refresh Rate * 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 * 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 CL2 low pulse time is doubled after every CL1 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 (**CL1**). The m signal will toggle after n+1 counts of the line clock (**CL1**) 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	**GSEN**: Grey scale enable bit. Setting this bit enables grey scale output to the LCD. When this bit is cleared each bit in the video map directly corresponds to a pixel in the display.
31	**GSMD**: Grey scale mode bit. Clearing this bit sets the controller to 2-bits per pixel (4 grey scales), setting it sets it to 4 bits per pixel (16 grey scales). This bit has no effect if GSEN is cleared.

5.10.2 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*
Grey scale value for pixel value 7	Grey scale value for pixel value 6	Grey scale value for pixel value 5	Grey scale value for pixel value 4	Grey scale value for pixel value 3	Grey scale value for pixel value 2	Grey scale value for pixel value 1	Grey scale 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 grey scale level. The 64-bit register is made up of 16 x 4-bit nibbles, each nibble defines the grey scale 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 (D15–D0 in the least significant word).

5.10.3 PALMSW Most Significant Word — LCD Palette Register

ADDRESS: 0x8000.0540

See PALLSW description.

31–28	*27–24*	*23–20*	*19–16*	*15–12*	*11–8*	*7–4*	*3–0*
Grey scale value for pixel value 15	Grey scale value for pixel value 14	Grey scale value for pixel value 13	Grey scale value for pixel value 12	Grey scale value for pixel value 11	Grey scale value for pixel value 10	Grey scale value for pixel value 9	Grey scale value for pixel value 8

The actual physical color and pixel duty ratio for the grey scale values are shown in the table above. Note that colors 8–15 are the inverse of colors 7–0 respectively. This means that colors 7 and 8 are identical. The steps in the grey scale are non-linear, but have been chosen to give a close approximation to perceived linear grey scales. The is due to the eye being more sensitive to changes in grey level close to 50% grey.

Table 5-12. Grey Scale Value to Color Mapping

Grey Scale 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 %

Table 5-12. Grey Scale Value to Color Mapping *(cont.)*

Grey Scale Value	Duty Cycle	% Pixels Lit	% Step Change
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 %	

5.10.4 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).

5.11 SSI Register

5.11.1 SYNCIO Synchronous Serial ADC Interface Data Register

ADDRESS: a0x8000.0500

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[31:16] 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. In order 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 as defined in the ADC Configuration Extension section of the SYNCIO register.

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

<table>
<tr><th><em>31</em></th><th><em>16</em></th></tr>
<tr><td colspan="2">ADC Configuration Extension</td></tr>
</table>

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).

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.
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]).

5.12 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 (e.g., a new battery was installed). Any value may be written to this location.

5.13 ‘End Of Interrupt’ Locations

These locations 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.

5.13.1 BLEOI Battery Low End of Interrupt

ADDRESS: 0x8000.0600

A write to this location will clear the interrupt generated by a low battery (falling edge of **BATOK** with **NEXTPWR** high).

5.13.2 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.

5.13.3 TEOI Tick End of Interrupt Location

ADDRESS: 0x8000.0680

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

5.13.4 TC1EOI TC1 End of Interrupt Location

ADDRESS: 0x8000.06C0

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

5.13.5 TC2EOI TC2 End of Interrupt Location

ADDRESS: 0x8000.0700

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

5.13.6 RTCEOI RTC Match End of Interrupt

ADDRESS: 0x8000.0740

A write to this location will clear the RTC match interrupt

5.13.7 UMSEOI UART1 Modem Status Changed End of Interrupt

ADDRESS: 0x8000.0780

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

5.13.8 COEOI Codec End of Interrupt Location

ADDRESS: 0x8000.07C0

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

5.13.9 KBDEOI Keyboard End of Interrupt Location

ADDRESS: 0x8000.1700

A write to this location clears the KBDINT keyboard interrupt.

5.13.10 SRXEOF End of Interrupt Location

ADDRESS: 0x8000.1600

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

5.14 State Control Registers

5.14.1 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. It will automatically switch the DRAMs into self refresh if the RFSHEN bit is set in the DRAM refresh period register. All transitions to the Standby State are synchronized with DRAM cycles. A write to this location while there is an active interrupt will have no effect.

NOTE: The following restrictions apply when operating with a self-refresh DRAM

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) After exiting from the Standby State, the first instruction that gets executed must be fetched from non-DRAM.

NOTE: If the EP7209 is attempting to get into the 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.

5.14.2 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.

5.15 SS2 Registers

5.15.1 SS2DR Synchronous Serial Interface 2 Data Register

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; the low 16 bits contain 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.

5.15.2 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.

5.16 I^{2}S Register Definitions

There are five registers within the I^{2}S Interface, one control register, three data registers, and one status register. The control register is used to mask or unmask interrupt requests to service the I^{2}S'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.

5.16.1 I^{2}S Control Register

ADDRESS: 0x8000.2000

The I^{2}S control register (MCCR) contains eight different bit fields that control various functions within the I^{2}S interface.

5.16.1.1 I^{2}S Enable (I^{2}SEN)

The I^{2}S enable (I^{2}SEN) bit is used to enable and disable all I^{2}S operation.

When the I^{2}S is disabled, all of its clocks are powered down to minimize power consumption. Note that I^{2}SEN is the only control bit within the I^{2}S interface that is reset to a known state. It is cleared to zero to ensure the I^{2}S timing is disabled following a reset of the device.

When the I^{2}S 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 I^{2}SEN bit is set, the I^{2}S interface operates continuously, transmitting and receiving 128 bit data frames. When the I^{2}SEN bit is cleared, the I^{2}S interface is disabled immediately, causing the current frame which is being transmitted to be terminated. Clearing I^{2}SEN resets the I^{2}S's interface FIFOs. However I^{2}S data register 3, the control and the status registers are not reset. The user must ensure these registers are properly reconfigured before re-enabling the I^{2}S interface.

5.16.1.2 I^{2}S Interrupt Generation

The I^{2}S 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 I^{2}S 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 I^{2}S to determine which source(s) caused the interrupt. It is possible to prevent any I^{2}S sources causing an interrupt by masking the I^{2}S interrupt in the interrupt controller register.

5.16.1.3 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 I^{2}S 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.

5.16.1.4 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 I^{2}S 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.

5.16.1.5 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 I^{2}S 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; it only blocks the generation of the interrupt request.

5.16.1.6 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 I^{2}S 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; it only blocks the generation of the interrupt request.

5.16.1.7 Loop Back Mode (LBM)

The loop back mode (LBM) bit is used to enable and disable the ability of the I^{2}S's transmit and receive logic to communicate. When LBM = 0, the I^{2}S operates normally. The transmit and receive data paths are independent and communicate via their respective pins. When LBM = 1, the output of the serial shifter (MSB) is directly connected to the input of the serial shifter (LSB) internally and control of the **SDOUT**, **SDIN**, **SCLK**, and **LRCK** pins are given to the peripheral pin control (PPC) unit.

The following figure shows the bit locations corresponding to the ten different control bit fields within the I^{2}S control register. Note that the I^{2}SEN bit is the only control bit which is reset to a known state to ensure the I^{2}S is disabled following a reset of the device. The reset state of all other control bits is unknown and must be initialized before enabling the I^{2}S. Writes to reserved bits are ignored and reads return zeros.

Address: 0x 8000 2000 **I2S Control Register: MCCR** **Read/Write**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved								LBM	RCRM	RCTM	LCRM	LCTM	Res.	ECS	I2SEN
Reset	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved															
Reset	0	?	?	?	?	?	?	?	0	?	?	?	?	?	?	?

Table 5-13. I^{2}S Control Register

Bit	Name	Description
15–0	—	Reserved Must be set to 0x0404
7	—	Reserved
15	—	Reserved
16	I2SEN	I^{2}S Interface Enable 0 — I^{2}S operation disabled, control of the SDIN, SDOUT, SCLKLRCK, and LRCK pins given to the SSI2/codec/I^{2}S pin mulitiplexing logic to assign I/O pins 60-64 to another block. 1 — I^{2}S operation enabled Note that by default, the SSI/CODEC have precedence over the I^{2}S interface in regard to the use of the I/O pins. Nevertheless, when Bit 3 (MCPSEL) of register SYSCON3 is set to 1, then the above mentioned I^{2}S ports are connected to I/O pins 60–64.
17	ECS	External Clock Select selects external MCLK when = 1.
18	—	Reserved Must 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).

Table 5-13. I^{2}S Control Register *(cont.)*

Bit	Name	Description
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	LBM	Loop Back Mode 0 — Normal serial port operation enabled 1 — Output of serial shifter is connected to input of serial shifter internally and control of SDIN, SDOUT, SCLK, and LRCK pins is given to the PPC unit.
31–24	—	Reserved

5.16.2 I^{2}S Data Registers

The I^{2}S contains three data registers: I^{2}SDR0 addresses the top entry of the right channel transmit FIFO and bottom entry of the right channel receive FIFO; I^{2}SDR1 addresses the top and bottom entry of the left channel transmit and receive FIFOs, respectively; and I^{2}SDR2 is used to perform enable and disable the I^{2}S FIFOs.

5.16.2.1 I^{2}S Data Register 0

ADDRESS: 0x8000.2040

When I^{2}S Data Register 0 (I^{2}SDR0) is read, the bottom entry of the right channel receive FIFO is accessed. As data is removed by the I^{2}S'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 I^{2}SDR0, which accesses the bottom entry of the right channel FIFO. After I^{2}SDR0 is read, the bottom entry is invalidated, and all remaining values within the FIFO automatically transfer down one location.

When I^{2}SDR0 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, is loaded into the correct position within the 64-bit transmit serial shifter, then serially shifted out onto the SDOUT pin.

The following figure shows I^{2}SDR0. Note that the transmit and receive right channel FIFOs are cleared when the device is reset, or by writing a zero to I^{2}SEN (I^{2}S disabled). Also, note that writes to reserved bits are ignored and reads return zeros.

Address: 0x 8000 2008 **I^{2}S Data Register 0: MCDR0** **Read/Write**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Bottom of Right Channel Receive FIFO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Read Access

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Top of Right Channel Transmit FIFO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Write Access

Figure 5-1. I^{2}S Data Register 0: I^{2}SDR0

Table 5-14. I^{2}S Data Register 0

Bit	Name	Description
15–0	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
31–16	—	Reserved

5.16.2.2 I^{2}S Data Register 1

ADDRESS: 0x8000.2080

When I^{2}S Data Register 1 (I^{2}SDR1) is read, the bottom entry of the left channel receive FIFO is accessed. As data is removed by the I^{2}S'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 I^{2}SDR1, which accesses the bottom entry of the left channel FIFO. After I^{2}SDR1 is read, the bottom entry is invalidated, and all remaining values within the FIFO automatically transfer down one location.

When I^{2}SDR1 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.

The following figure shows I^{2}SDR1. Note that the transmit and receive left channel FIFOs are cleared when the device is reset, or by writing a zero to I^{2}SEN (I^{2}S disabled). Also, note that writes to reserved bits are ignored and reads return zeros.

Address: 0x 8000 200C **I^{2}S Data Register 1: I^{2}SDR1** **Read/Write**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Bottom of Left Channel Receive FIFO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Read Access

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Top of Left Channel Transmit FIFO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Write Access

Figure 5-2. I^{2}S Data Register 1: MCDR1

Table 5-15. I^{2}S Data Register 1

Bit	Name	Description
15–0	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
31–16	—	Reserved

5.16.2.3 I^{2}S Data Register 2

ADDRESS: 0x8000.20C0

I2SDR2 contains 21 bits and is used to enable and disable the FIFOs for the left and right channels of the I2S data stream. The left channel FIFO is enabled by writing 0x000D.C000 and disabled by writing 0x000D.0000. The right channel FIFO is enabled by writing 0x0011.C000 and disabled by writingg 0x0011.0000. After writing a value to this register, wait until the FIFO operation complete bit (FIFO) is set in the I2S status register before writing another value to this register.

5.16.3 I^{2}S Status Register

ADDRESS: 0x8000.2100

The I^{2}S Status Register (I^{2}SSR) 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, 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 I^{2}S interrupts by clearing the I^{2}S bit within the interrupt controller mask register INTMR3.

5.16.3.1 Right Channel Transmit FIFO Service Request Flag (RCTS) (read-only, maskable interrupt)

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.

5.16.3.2 Right Channel Receive FIFO Service Request Flag (RCRS) (read-only, maskable interrupt)

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.

5.16.3.3 Left Channel Transmit FIFO Service Request Flag (LCTS) (read-only, maskable interrupt)

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), and 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.

5.16.3.4 Left Channel Receive FIFO Service Request Flag (LCRS) (read-only, maskable interrupt)

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 service request and/or interrupt) is automatically cleared.

5.16.3.5 Right Channel Transmit FIFO Underrun Status (RCTU) (read/write, non-maskable interrupt)

The right channel transmit FIFO underrun status bit (RCTU) is set when the right channel transmit logic attempts to fetch data from the FIFO after it has been completely emptied. When an underrun occurs, the right channel transmit logic continuously transmits the last valid right 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 right channel transmit logic uses the new value within the FIFO for transmission. When the RCTU bit is set, an interrupt request is made.

5.16.3.6 Right Channel Receive FIFO Overrun Status (RCRO) (read/write, non-maskable interrupt)

The right channel receive FIFO overrun status bit (RCRO) 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 the RCRO bit is set, an interrupt request is made.

5.16.3.7 Left Channel Transmit FIFO Underrun Status (LCTU) (read/write, non-maskable interrupt)

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.

5.16.3.8 Left Channel Receive FIFO Overrun Status (LCRO) (read/write, non-maskable interrupt)

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.

5.16.3.9 Right Channel Transmit FIFO Not Full Flag (RCNF) (read-only, non-interruptible)

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.

5.16.3.10 Right Channel Receive FIFO Not Empty Flag (RCNE) (read-only, non-interruptible)

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.

5.16.3.11 Left Channel Transmit FIFO Not Full Flag (LCNF) (read-only, non-interruptible)

The left channel transmit FIFO not full flag (LCNF) 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 completely 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.

5.16.3.12 Left Channel Receive FIFO Not Empty Flag (LCNE) (read-only, non-interruptible)

The left channel receive FIFO not empty flag (LCNE) is a read-only bit which is set when ever 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.

5.16.3.13 FIFO Operation Completed Flag (FIFO) (read-only, non-interruptible)

The FIFO operation completed (FIFO) flag is set after the FIFO operation requested by writing to I^{2}SDR2 as completed.

FIFO is automatically cleared when I^{2}SDR2 is read or written. This bit does not request an interrupt.

Address: 0x 8000 2018 I^{2}S Status Register: MCSR Read/Write & Read-Only

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Res	Res	Res.	FIFO	LCNE	LCNF	RCNE	RCNF	RCCELCRO	RCN-FLCTU	LCRORCRO	LCTURCTU	LCRS	LCTS	LCRSRCRS	LCTSRCTS
Reset	0	0	0	0	0	1	0	1	?	?	?	?	0	0	0	0

Figure 5-3. I2SEP7209 Status Register: MCSR

Table 5-16. I^{2}S Control, Data and Status Register Locations

Bit	Name	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 I^{2}S disabled 1 — right channel transmit FIFO is half full or less (four or fewer entries filled) and I^{2}S operation is enabled, interrupt request signaled if not masked (if RCTM = 1)
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 I^{2}S disabled 1 — right channel receive FIFO is half full or more (six or more entries filled) and I^{2}S 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 I^{2}S disabled. 1 — Left Channel transmit FIFO is half full or less (four or fewer entries filled) and I^{2}S 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 I^{2}S disabled. 1 — Left Channel receive FIFO is half full or more (six or more entries filled) and I^{2}S operation is enabled, interrupt request signalled if not masked (if LCRM = 1)
4	RCTU	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

Table 5-16. I^2S Control, Data and Status Register Locations *(cont.)*

Bit	Name	Description
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
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 cleared 1 — THe FIFO Operation was completed
13	—	Reserved
14	—	Reserved
15	—	Reserved
31-16	—	Reserved

6. ELECTRICAL SPECIFICATIONS

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 cummulative
Storage Temperature, No Power	–40°C to +125°C

6.2 Recommended Operating Conditions

DC core, PLL, and RTC Supply Voltage	2.5 V ± 0.2 V
DC I/O Supply Voltage (Pad Ring)	2.3V - 3.6V
DC Input/Output Voltage	O–I/O supply voltage
Operating Temperature	0°C to +70°C

6.3 DC Characteristics

All characteristics are specified at V_{DD} = 2.5 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, 18.432 MHz operation with the PLL switched ‘on’.

Table 6-1. DC Characteristics

Symbol	Parameter	Min	Max	Unit	Conditions
VIH	CMOS input high voltage	1.7	V_{DD} + 0.3	V	V_{DD} = 2.5 V
VIL	CMOS input low voltage	-0.3	0.8	V	V_{DD} = 2.5 V
VT+	Schmitt trigger positive going threshold	1.6 (Typ)	2.0	V	
VT-	Schmitt trigger negative going threshold	0.8	1.2 (Typ)	V	
Vhst	Schmitt trigger hysteresis	0.1	0.4	V	VIL to VIH
VOH	CMOS output high voltageOutput drive 1 Output drive 2	V_{DD} – 0.2 2.5 2.5		V V V	IOH = 0.1 mA OH = 4 mA OH = 12 mA
VOL	CMOS output low voltageOutput drive 1 Output drive 2		0.3 0.5 0.5	V V V	IOL = –0.1 mA OL = –4 mA OL = –12 mA

Table 6-1. DC Characteristics *(cont.)*

Symbol	Parameter	Min	Max	Unit	Conditions
IIN	Input leakage current		1	µA	VIN = V_{DD} or GND
IOZ	Output tri-state leakage current*	25	100	µA	VOUT = V_{DD} or GND
CIN	Input capacitance	8	10	pF	
COUT	Output capacitance	8	10	pF	
CI/O	Transceiver capacitance	8	10	pF	
$IDD_{startup}$	Startup current consumption			µA	Initial 100 ms from power up, 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
$IDD_{standby}$	Standby current consumption		4	µA	Just 32 kHz oscillator running, all other I/O static, VIH = V_{DD} ± 0.1 V, VIL = GND ± 0.1 V
IDD_{idle}	Idle current consumption			mA	Both oscillators running, CPU static, LCD refresh active, VIH = V_{DD} ± 0.1 V, VIL = GND ± 0.1 V
	At 13 MHz		4.2		
	At 18 MHz		6		
	At 36 MHz		12		
$IDD_{operating}$	Operating current consumption			mA	All system active, running typical program
	At 13 MHz				
	At 18 MHz		14		
	At 36 MHz		20		
	At 49 MHz		40		
	At 74 MHz		50		
			68		
$V_{DDstandby}$	Standby supply voltage	TBD		V	Minimum standby voltage for state retention and RTC operation only

NOTE: All power dissipation values can be derived from taking the particular IDD current and multiplying by 2.5 V.

The RTC of the EP7209 should be brought up at room temperature. This is required because 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.

A typical design will provide 3.3 V to the I/O supply (i.e., VDDIO), 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 (e.g., 3.3 V DRAMs).

Pull-up current = 50 µA typical at V_{DD} = 3.3 volts.

6.4 AC Characteristics

All characteristics are specified at V_{DD} = 2.3 to 2.7 volts and V_{SS} = 0 volts over an operating temperature of 0°C to +70°C. Those characteristics marked with a # will be significantly different for 13 MHz mode because the **EXPCLK** is provided as an input rather than generated internally. These timings are estimated at present. The timing values are referenced to 1/2 V_{DD}.

Table 6-2. AC Timing Characteristics

Symbol	Parameter	13 MHz		18/36 MHz		Units
		Min	Max	Min	Max	
T1	Falling CS to data bus Hi-Z	0	35	0	25	ns
T2	Address change to valid write data	0	45	0	35	ns
T3	DATA in to falling EXPCLK setup time	0 #	—	18	—	ns
T4	DATA in to falling EXPCLK hold time	10 #	—	0	—	ns
T5	EXPRDY to falling EXPCLK setup time	0 #	—	18	—	ns
T6	Falling EXPCLK to EXPRDY hold time	10 #	50	0	50	ns
T7	Rising NMWE to data invalid hold time	10	—	5	—	ns
T8	Sequential data valid to falling NMWE setup time	–10	10	–10	10	ns
T15	LCD CL2 low time	80	3,475	80	3,475	ns
T16	LCD CL2 high time	80	3,475	80	3,475	ns
T17	LCD falling CL2 to rising CL1 delay	0	25	0	25	ns
T18	LCD falling CL1 to rising CL2	80	3,475	80	3,475	ns
T19	LCD CL1 high time	80	3,475	80	3,475	ns
T20	LCD falling CL1 to falling CL2	200	6,950	200	6,950	ns
T21	LCD falling CL1 to FRM toggle	300	10,425	300	10,425	ns
T22	LCD falling CL1 to M toggle	–10	20	–10	20	ns
T23	LCD rising CL2 to display data change	–10	20	–10	20	ns
T24	Falling EXPCLK to address valid	—	33 #	—	5	ns
T25	Data valid to falling NMWE for non sequential access only	5	—	5	—	ns
T31	SSICLK period (slave mode)	0	512	0	512	kHz
T32	SSICLK high	925	1025	925	1025	ns
T33	SSICLK low	925	1025	925	1025	ns
T34	SSICLK rise/fall time		7		7	ns
T35	SSICLK rising to RX and/or TX frame sync		528		528	ns

Table 6-2. AC Timing Characteristics *(cont.)*

Symbol	Parameter	13 MHz		18/36 MHz		Units
		Min	Max	Min	Max	
T36	SSICLK rising edge to frame sync low		448		448	ns
T37	SSICLK rising edge to TX data valid		80		80	ns
T38	SSIRXDA data set-up time	30		30		ns
T39	SSIRXDA data hold time	40		40		ns
T40	SSITXFR and/or SSIRXFR period	750		750		ns

Table 6-3. Timing Characteristics

Symbol	Characteristics	13 MHz		18 MHz		36 MHz		Units
		Min	Max	Min	Max	Min	Max	
t_{NCSRD}	Negative strobe (NCS[0–5]) zero wait state read access time	120		70		35		ns
t_{NCSWR}	Negative strobe (NCS[0–5]) zero wait state write access time	120		70		35		ns
t_{EXBST}	Sequential expansion burst mode read access time	55		35		8		ns

Consecutive Expansion Read Cycles with Minimum Wait

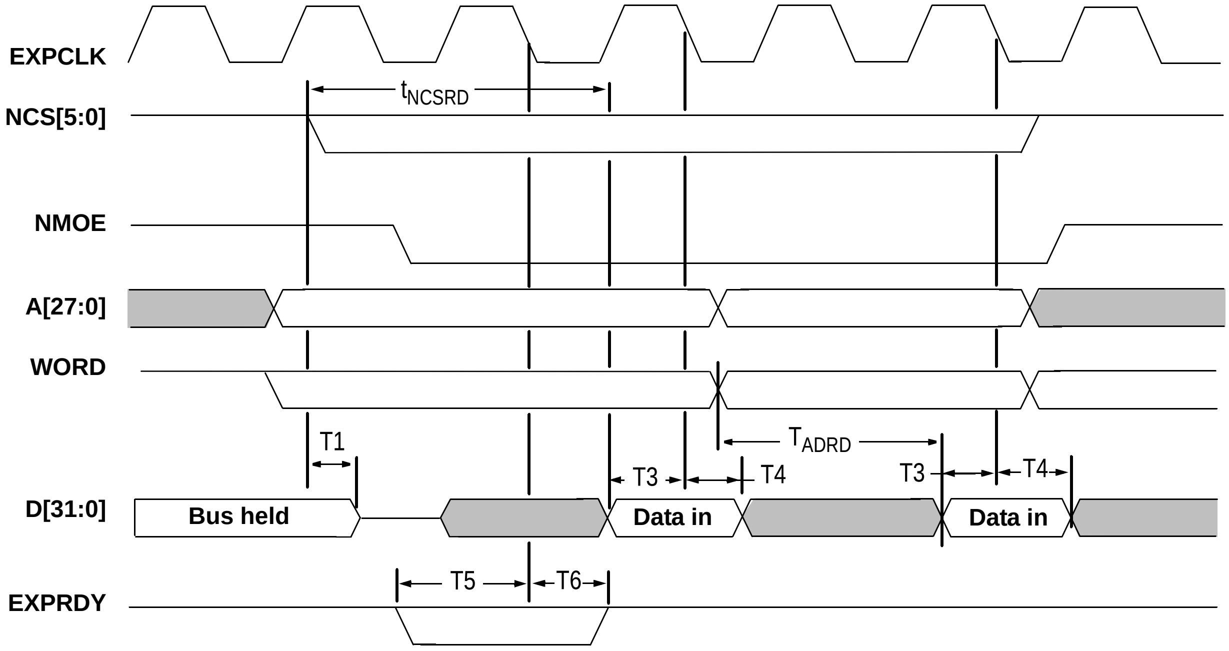


Figure 6-1. Expansion and ROM Timing

NOTES: 1) TNCSRD = 35 ns at 36.864 MHz
70 ns at 18.432 MHz
120 ns at 13.0 MHz

Maximum values for minimum wait states. This time can be extended by integer multiples of the clock period (27 ns at 36 MHz, 54 ns at 18.432 MHz, and 77 ns at 13 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.

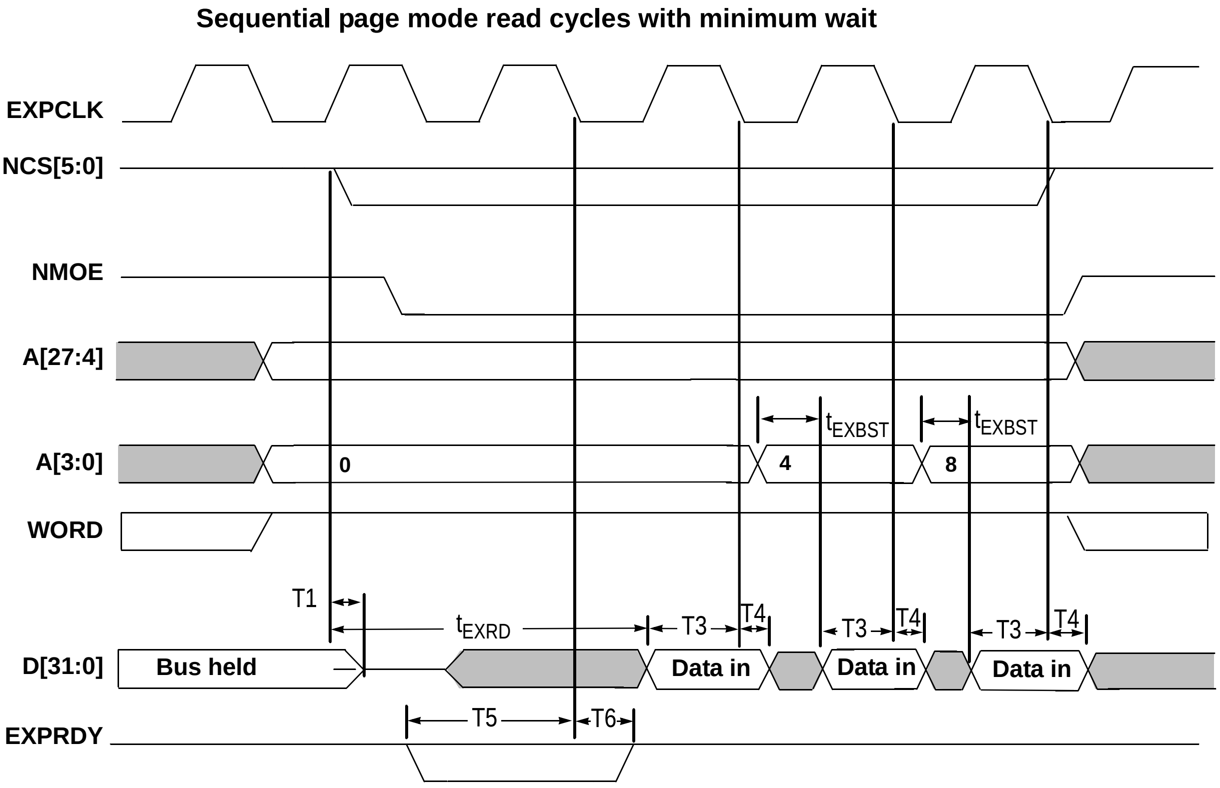


Figure 6-2. Expansion and ROM Sequential Read Timings

NOTES:

1) t_{EXBST} = 35 ns at 36.864 MHz
70 ns at 18.432 MHz
120 ns at 13.0 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 ns at 36 MHz, 54 ns at 18.432 MHz and 77 ns at 13 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.

Consecutive expansion write cycles with minimum wait states

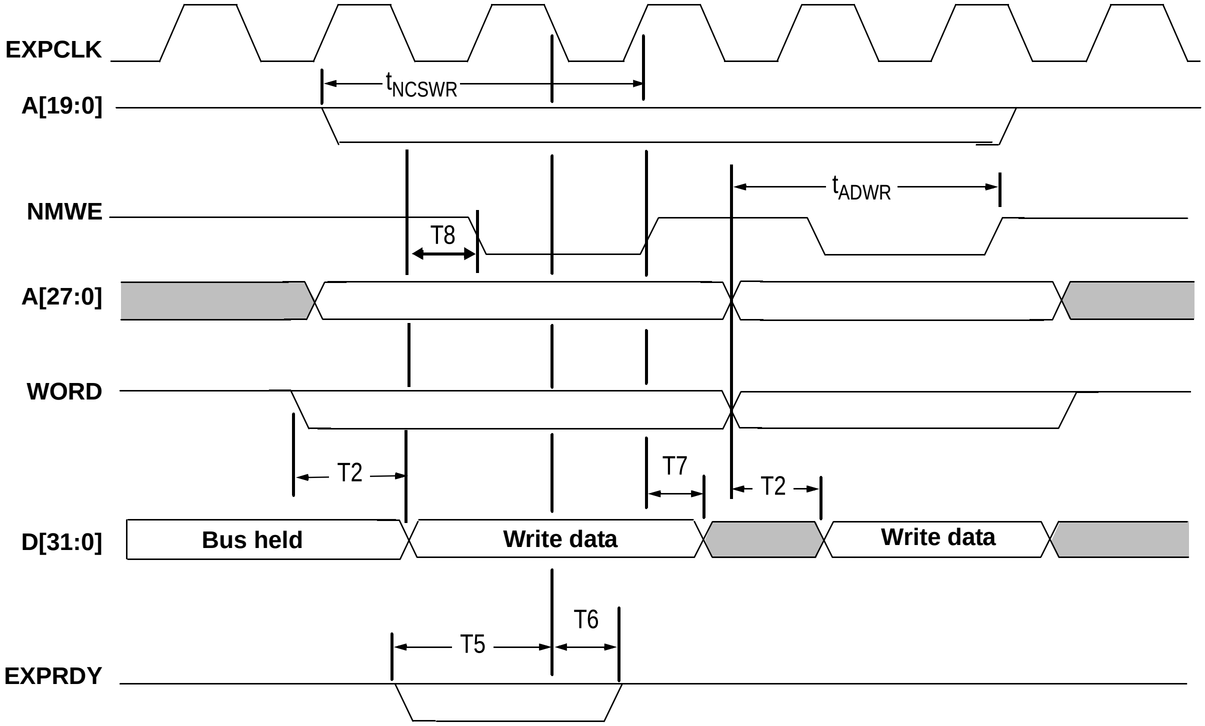


Figure 6-3. Expansion and ROM Write Timings

NOTES:

1) t_{EXWR} = 35 ns at 36.864 MHz
70 ns at 18.432 MHz
120 ns at 13.0 MHz

Maximum values for minimum wait states. This time can be extended by integer multiples of the clock period (27 ns at 36 MHz, 54 ns at 18.432 MHz, and 77 ns at 13 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.

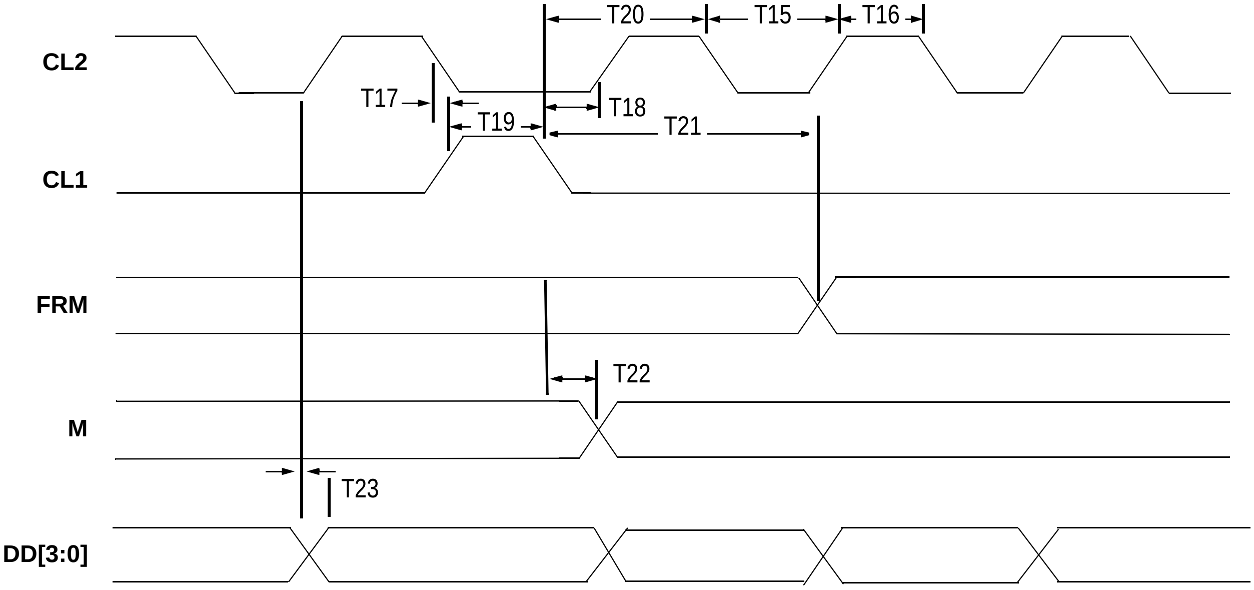


Figure 6-4. LCD Controller Timings

NOTES:
1) The figure shows the end of a line.
2) If FRM is high during the CL1 pulse, this marks the first line in the display.
3) CL2 low time is doubled during the CL1 high pulse

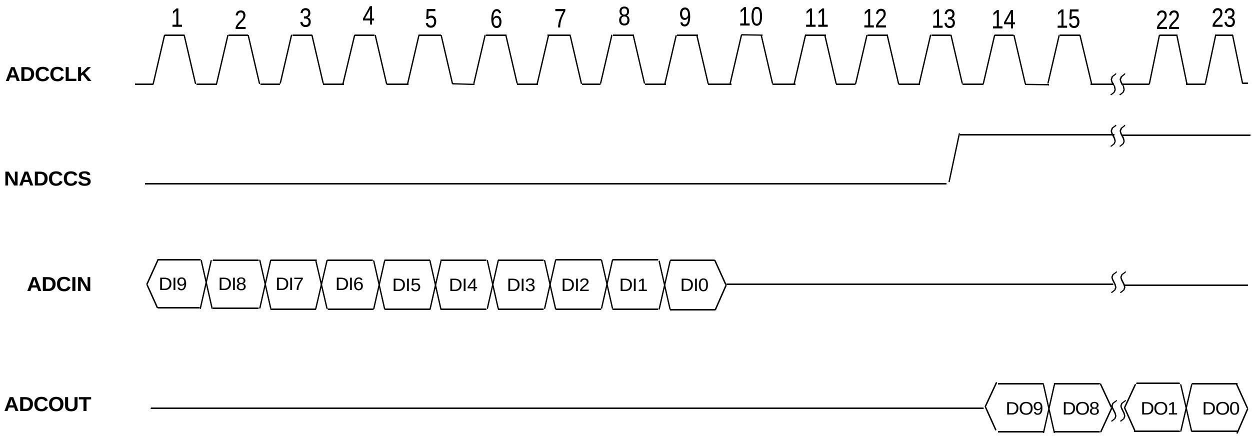


Figure 6-5. SSI Interface for AD7811/2

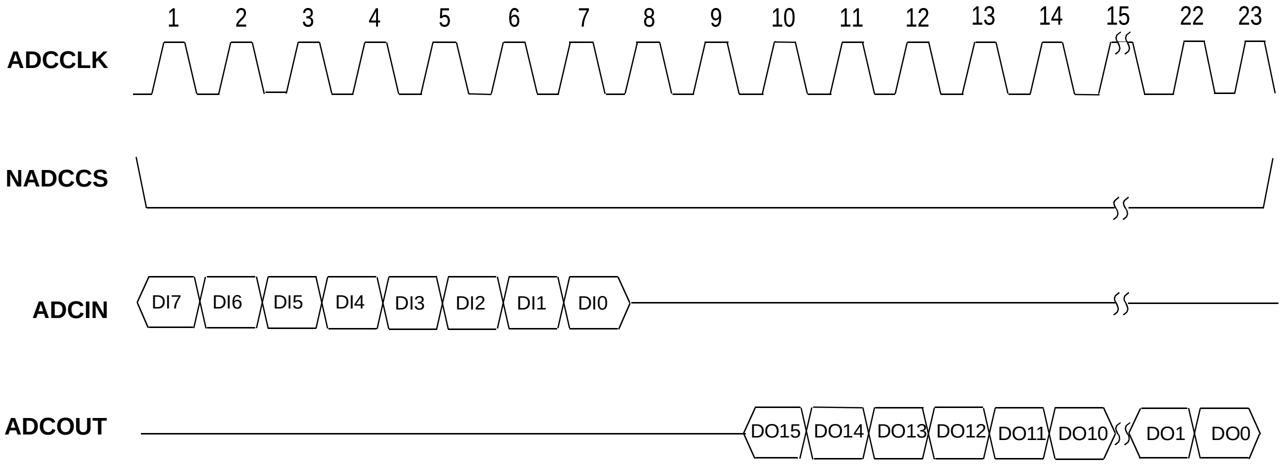


Figure 6-6. SSI Timing Interface for MAX148/9

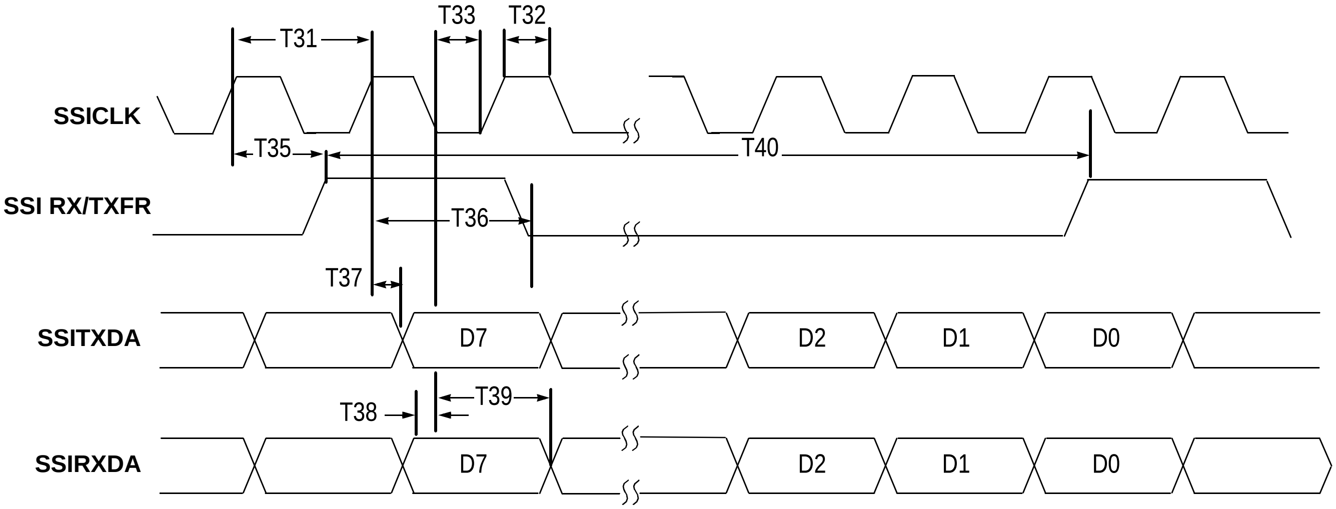


Figure 6-7. SSI2 Interface Timings

6.5 I/O Buffer Characteristics

All I/O buffers on the EP7209 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 tristated 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 6-4. I/O Buffer Output Characteristics** 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).

Table 6-4. I/O Buffer Output Characteristics

Buffer Type	Drive Current	Propagation Delay (Max)	Rise Time (Max)	Fall Time (Max)	Load
I/O strength 1	±4 mA	7	14	14	50 pF
I/O strength 2	±12 mA	5	6	6	50 pF

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}.

7. TEST MODES

The EP7209 supports a number of hardware activated test modes, these are activated by the pin combinations shown in the **Table 7-1. EP7209 Hardware Test Modes**. 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.

Table 7-1. EP7209 Hardware Test Modes

Test Mode	Latched NMEDCHG	Latched PE0	Latched PE1	Latched NURESET	NTEST0	NTEST1
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

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.

7.1 Oscillator and PLL Bypass Mode

This mode is selected by **NTEST0** = 1**, NTEST1** = 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.

7.2 Oscillator and PLL Test Mode

This mode is selected by **NTEST0** = 0**, NTEST1** = 1, Latched **NURESET** = 0

This test mode will enable the main oscillator and it will output various buffered clock and test signals derived from the main oscillator, PLL, and 32-kHz oscillator. All internal logic in the EP7209 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 7-2.**

Oscillator and PLL Test Mode Signals defines the EP7209 signal pins used in this test mode. This mode is only intended to allow test of the oscillators and PLL**.**

Table 7-2. Oscillator and PLL Test Mode Signals

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

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 7-2. Oscillator and PLL Test Mode Signals**. Any other combinations are for testing the oscillator and PLL and should not be used in-circuit.

7.3 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.

7.4 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 EP7209. This has the effect of removing the EP7209 from the PCB so that other devices on the PCB can be in-circuit tested. The internal state of the EP7209is not altered directly by this test mode.

7.5 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 CS5. Hence, CS5 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 CS5. Additionally, in this mode, the internal signals shown in **Table 7-3. Software Selectable Test Functionality** are multiplexed out of the device on port pins.

Table 7-3. Software Selectable Test Functionality

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

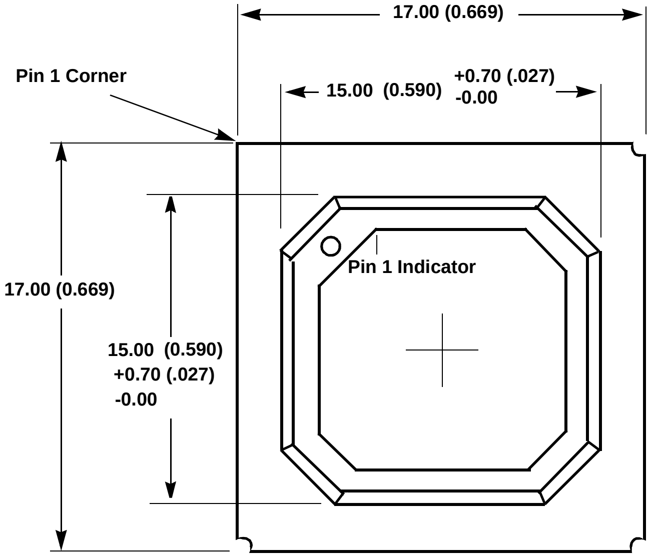
This test is not intended to be used when LCD DMA accesses are enabled. This is due to the fact that it is possible to have internal peripheral bus activity simultaneously with a DMA transfer. This would cause bus contention to occur on the external bus.

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.

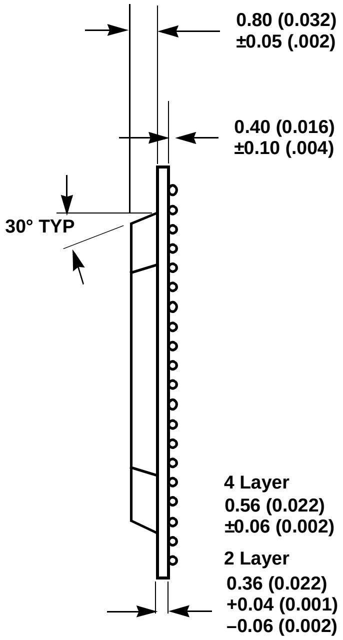
The signals **NFIQ** and **NIRQ** are the two output signals from the internal interrupt controller. They are input directly into the ARM720T processor.

8. PACKAGE SPECIFICATIONS

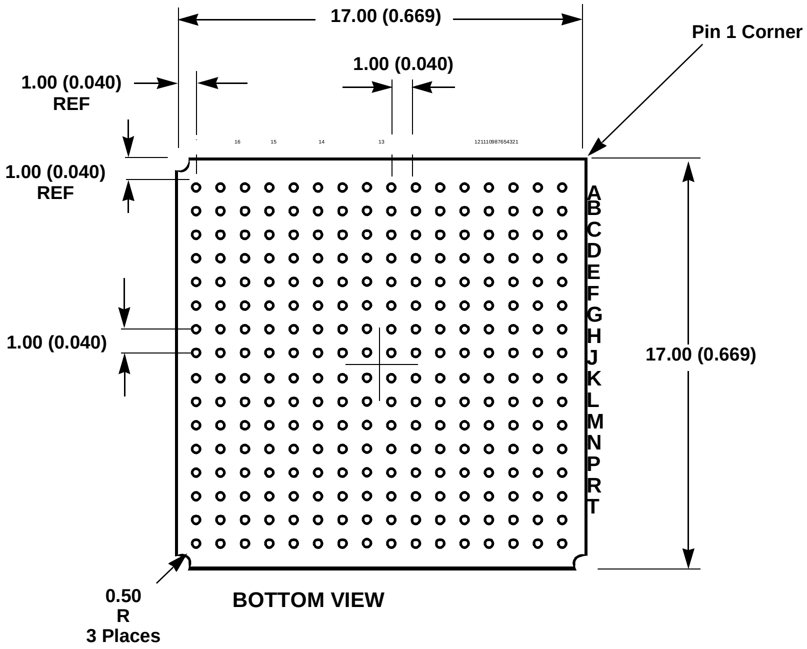
8.1 EP7209 256-Ball PBGA (17 × 17 × 1.53-mm Body) Dimensions



TOP VIEW



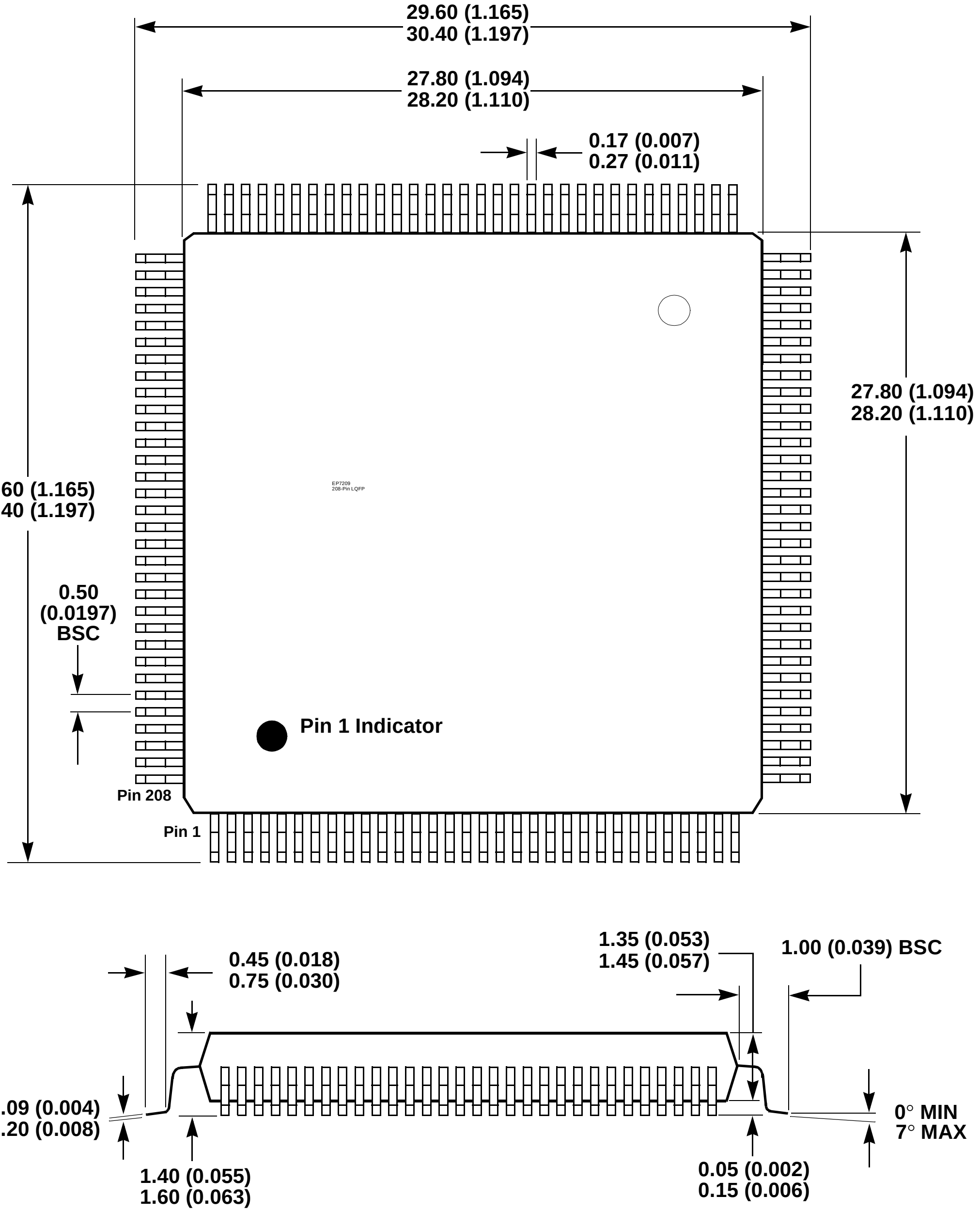
SIDE VIEW



BOTTOM VIEW

(Call Factory for Availability)

8.2 208-Pin LQFP Package Outline Drawing



NOTES:

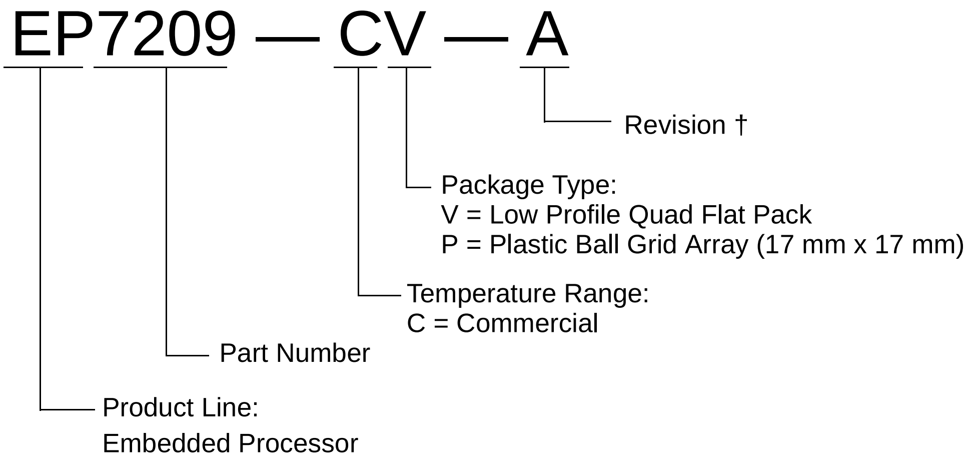
1) Dimensions are in millimeters (inches), and controlling dimension is millimeter.

2) Drawing above does not reflect exact package pin count.

3) Before beginning any new design with this device, please contact Cirrus Logic for the latest package information.

9. ORDERING INFORMATION

The order number for the device is:



EP7209 — CV — A

Revision †

Package Type:
V = Low Profile Quad Flat Pack
P = Plastic Ball Grid Array (17 mm x 17 mm)

Temperature Range:
C = Commercial

Part Number

Product Line:
Embedded Processor

NOTE: †Contact Cirrus Logic for up-to-date information on revisions.

10. APPENDIX A: BOOT CODE

```
;(C) Copyright 1995-1996 , Cirrus Logic, Inc. All Rights Reserved.
;
    TTL     CL-EP7209 Sample program
;       version 1.0 (initial version) ;ks
;       boot from uart1



    AREA    |C$$code|,CODE,READONLY


    ENTRY

;       System constants
;
HwBaseAddress                   EQU     0x80000000
;
;
HwControl                       EQU     0x00000100
HwControl2                      EQU     0x00001100
HwControlUartEnable             EQU     0x00000100
;
HwStatus                        EQU     0x00000140
HwStatus2                       EQU     0x000001140
HwStatusUartRxFifoEmpty         EQU     0x00400000
;
HwUartData                      EQU     0x00000480
HwUartData2                     EQU     0x00001480
HwUartDataFrameErr              EQU     0x0100
HwUartDataParityErr             EQU     0x0200
HwUartDataOverrunErr            EQU     0x0400
HwUartControl                   EQU     0x000004C0
HwUartControl2                  EQU     0x000014C0
HwUartControlRate               EQU     0x00000FFF
HwUartControlRate115200         EQU     0x001
HwUartControlRate76800          EQU     0x002
HwUartControlRate57600          EQU     0x003
```

```
HwUartControlRate38400          EQU     0x005
HwUartControlRate19200          EQU     0x00B
HwUartControlRate14400          EQU     0x00F
HwUartControlRate9600           EQU     0x017
HwUartControlRate4800           EQU     0x02F
HwUartControlRate2400           EQU     0x05F
HwUartControlRate1200           EQU     0x0BF
HwUartControlRate600            EQU     0x17F
HwUartControlRate300            EQU     0x2FF
HwUartControlRate150            EQU     0x5FF
HwUartControlRate110            EQU     0x82E
HwUartControlRate115200_13      EQU     0x000
HwUartControlRate57600_13       EQU     0x001
HwUartControlRate38400_13       EQU     0x002
HwUartControlRate19200_13       EQU     0x005
HwUartControlRate14400_13       EQU     0x007
HwUartControlRate9600_13        EQU     0x00b
HwUartControlRate4800_13        EQU     0x017
HwUartControlRate2400_13        EQU     0x02F
HwUartControlRate1200_13        EQU     0x060
HwUartControlRate600_13         EQU     0x0c0
HwUartControlRate300_13         EQU     0x182
HwUartControlRate150_13         EQU     0x305
HwUartControlRate110_13         EQU     0x41E
HwUartControlBreak              EQU     0x00001000
HwUartControlParityEnable       EQU     0x00002000
HwUartControlPartiyEvenOrOdd    EQU     0x00004000
HwUartControlTwoStopBits        EQU     0x00008000
HwUartControlFifoEnable         EQU     0x00010000
HwUartControlDataLength         EQU     0x00060000
HwUartControlDataLength5        EQU     0x00000000
HwUartControlDataLength6        EQU     0x00020000
HwUartControlDataLength7        EQU     0x00040000
HwUartControlDataLength8        EQU     0x00060000
;
;       9600baud, 8bits/ch no parity, 1 stop bit
UartValue       EQU     HwUartControlRate9600+HwUartControlDataLength8
UartValue_13    EQU     HwUartControlRate9600_13+HwUartControlDataLength8
BufferAddress   EQU     0x10000000      ;start address sram snooze buffer
```

```
codeexeaddr     EQU     0x10000000      ;
count           EQU     0x00000800      ;2k bytes
startflag       EQU     '<'
endflag         EQU     '>'
CLKMOD          EQU     0x40            ;clock mode 1 = 13 MHz

;       ARM Processor constants

ArmIrqDisable           EQU     0x00000080
ArmFiqDisable           EQU     0x00000040

; 26bit mode is not supported
;
ArmUserMode             EQU     0x10
ArmFIQMode              EQU     0x11
ArmIRQMode              EQU     0x12
ArmSVCMode              EQU     0x13
ArmAbortMode            EQU     0x17
ArmUndefMode            EQU     0x1B
ArmMaskMode             EQU     0x1F
;
ArmMmuCP                CP      0xF
;
ArmMmuId                CN      0x00
;
ArmMmuControl           CN      0x01
ArmMmuControlMmuEnable          EQU     0x00000001
ArmMmuControlAlignFaultEnable   EQU     0x00000002
ArmMmuControlCacheEnable        EQU     0x00000004
ArmMmuControlWriteBufferEnable  EQU     0x00000008
ArmMmuControl32BitCodeEnable    EQU     0x00000010
ArmMmuControl32BitDataEnable    EQU     0x00000020
ArmMmuControlMandatory          EQU     0x00000040
ArmMmuControlBigEndianEnable    EQU     0x00000080
ArmMmuControlSystemEnable       EQU     0x00000100
ArmMmuControlRomEnable          EQU     0x00000200
;
ArmMmuPageTableBase     CN      0x02
;
```

```
ArmMmuDomainAccess      CN      0x03
;
ArmMmuFlushTlb          CN      0x05
;
ArmMmuPurgeTlb          CN      0x06
;
ArmMmuFlushIdc          CN      0x07

;InitialMmuConfig      EQU                        ArmMmuControl32BitCodeEnable
+ArmMmuControl32BitDataEnable                          +ArmMmuControlMandatory
+ArmMmuControlBigEndianEnable
InitialMmuConfig           EQU                    ArmMmuControl32BitCodeEnable
+ArmMmuControl32BitDataEnable +ArmMmuControlMandatory   ;leave as little endian
11/6/96 ks

;===============================================================================
======
;       REAL CODE START

;       set little endian, 32bit code, 32bit data by writing to CP15's control
register

    LDR     r0, =InitialMmuConfig
    MCR     ArmMmuCP, 0, r0, ArmMmuControl, c0

;       set the cpu to SVC32 mode

    MRS     r0, CPSR                ;read psr
    BIC     r0, r0, #ArmMaskMode    ;remove the mode bits
    ORR     r0, r0, #ArmSVCMode     ;set to supervisor 32 bit mode
    MSR     CPSR, r0                ; Now set the CPU into the new mode
;

;       initialize HW control
UARTEnable
    LDR     r12,=HwBaseAddress
    MOV     r0,#HwControlUartEnable        ;Enable UART
    STR     r0,[r12,#HwControl]
    LDR     r1,=HwStatus2
    ADD     r1,r1,r12
    LDR     r2,[r1]                 ;read system flag2
    TST     r2,#CLKMOD
    LDREQ   r0,=UartValue           ;load 18 mhz value if bit not set
    LDRNE   r0,=UartValue_13        ;load 13 mhz value if bit set

    STR     r0,[r12,#HwUartControl]         ;initialise Uart

;       Send ready signal
    LDR     r0,=startflag
    STRB    r0,[r12,#HwUartData]            ; send ready

;       receive the data
    LDR     r3,=count
    LDR     r2,=BufferAddress

01
;       wait for byte available
    LDR     r1,[r12,#HwStatus]               ; spin, if Rx FIFO is empty
    TST     r1,#HwStatusUartRxFifoEmpty
    BNE     %b01

;       read the data ,store it and accumulate checksum
    LDRB    r0,[r12,#HwUartData]            ; read data
    STRB    r0,[r2],#1                      ; save it in memory
    SUBS    r3,r3,#1                        ; decrement count
    BNE     %b01                            ; do more if count has not expired

;       all received, send end flag
    LDR     r0,=endflag
    STRB    r0,[r12,#HwUartData]            ; send reply
    LDR     r15,=codeexeaddr                ;jump to execution address

    LTORG

    END
```

11. INDEX