SHARP®

LH79532

Product Summary

SHARP reserves the right to make changes in specifications described herein at any time and without notice in order to improve design or reliability. SHARP does not assume any responsibility for the use of any circuitry described; no circuit patent licenses are implied. SHARP assumes no responsibility for damage caused by misuse or improper use of devices.

LIFE SUPPORT POLICY

SHARP components should not be used in medical devices with life support functions, safety equipment (or similar applications where component failure would result in loss of life or physical harm), aerospace equipment, telecommunication equipment (trunk lines) or nuclear power control equipment. Contact a SHARP representative or sales office before using SHARP devices for any applications other than those recommended by SHARP.

LIMITED WARRANTY

SHARP warrants to its Customer that the Products will be free from defects in material and workmanship under normal use and service for a period of sixty (60) days from the date of invoice. Customer's exclusive remedy for breach of this warranty is that SHARP will either (i) repair or replace, at its option, any Product which fails during the warranty period because of such defect (if Customer promptly reported the failure to SHARP in writing) or, (ii) if SHARP is unable to repair or replace, refund the purchase price of the Product upon its return to SHARP. This warranty does not apply to any Product which has been subjected to misuse, abnormal service or handling, or which has been altered or modified in design or construction, or which has been serviced or repaired by anyone other than SHARP. The warranties set forth herein are in lieu of, and exclusive of, all other warranties, express or implied. ALL EXPRESS AND IMPLIED WARRANTIES, INCLUDING THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR USE AND FITNESS FOR A PARTICULAR PURPOSE, ARE SPECIFICALLY EXCLUDED. In no event will SHARP be liable, or in any way responsible, for any incidental or consequential economic or property damage.

The above warranty is also extended to Customers of SHARP authorized distributors with the following exception: reports of failures of Products during the warranty period and return of Products that were purchased from an authorized distributor must be made through the distributor. In case SHARP is unable to repair or replace such Products, refunds will be issued to the distributor in the amount of distributor cost.

LH79532 Universal Microcontroller Product Summary

Revision No. ICG000021-001

Released on November 12, 2002

Table of Contents

Introduction

The LH79532, powered by an ARM7TDMI™, is a complete 'System-On-Chip' with a high level of integration to satisfy a wide range of requirements and expectations**.** The LH79532 combines a 32-bit ARM7TDMI™ RISC, Cache, Local SRAM, a number of essential peripherals such as Direct Memory Access, Serial and Parallel Interfaces, Counter/Timers, Real Time Clock, Watch Dog Timer, Pulse Width Modulators, LCD Controller and an on-chip Phase Lock Loop. Debug is made simple by JTAG support.

This high level of integration lowers overall system costs, reduces development cycle time and accelerates product introduction. The LH79532's fully static design, power management unit, low voltage operation (2.5 V Core, 3.3 V I/O), on-chip PLL, fast interrupt response time, on-chip cache/SRAM, powerful instruction set, and very low power RISC core provide high performance at a low current draw.

The needs of a mobile lifestyle require advanced processing capability in our portable devices. This capability must come with increased performance in the display system and peripherals, and yet demand less power from batteries. The LH79532 is an integrated solution to fit these needs.

Features

- Highly Integrated Single Chip
- High Performance (50 MHz)
  - 32 kHz PLL Driven, or External Clock
- Low Power Modes
  - Active, Standby, Sleep, Stop
- 32-bit ARM7TDMI™ RISC Core
- 4kB on-chip Memory
  - Cache or SRAM Selectable
  - On-chip Cache
    - Unified Instruction / Data
    - Support Byte, Half-words, and Words
    - Write-back/Write-through
    - Write Buffer
- On-chip Programmable PLL (32 kHz)
- Clock and Power Management
- On-chip Programmable Interrupt Controller
  - 8 External Interrupts
  - Programmable
    - Active High / Low, Edge / Level
    - Drive IRQ / FIQ, Enable / Disable
- Two UARTs - 16C550-class
  - Support for Rx, Tx, RTS & CTS
  - 1Mbps Interface
  - 16 locations deep Receive & Transmit FIFO
- Real-Time Clock (RTC)
  - Full Calendar
  - Separate Power & Clock
- Four 16-bit Pulse Width Modulators
- Serial Peripheral Interface (SPI)
  - Support Motorola SPI format
  - Master / Slave
- Programmable Color LCD Controller
  - Up to 1024x768 Resolution
  - 18bit Video Bus
    - Support STN, Color STN, TFT
  - STN: 15 Gray Shade
  - Color STN: Supports 256 Colors Selected from a Palette of 3375 Colors or 3375 Direct Colors.
  - TFT: Supports 256 Colors Selected from a palette of 64K Colors or 64K Direct Colors
- Flexible Programmable Memory Interface
  - Boot from x8 or x16 or x32 Static Memory
  - 26-bit External Address Bus
  - 32 / 16 / 8-bit External Data Bus
  - SRAM / Flash / ROM Controller
    - Eight Banks (64MB Each)
    - External WAIT States Supported
  - SDRAM Controller
    - Two Chip Select Signals (256MByte each)
    - Supports Self & Auto Refresh
- 80 Programmable Parallel I/O Signals
- Four 16-bit Counter / Timer Channels
  - Two Modes of Operation
    - Interrupt on Terminal Count
    - Rate Generator / Square Wave Generator
  - Cascadable
- Hardware Watchdog Timer
  - Programmable Time-out Intervals
  - Protection Mechanism
  - System Reset & Interrupt Options
- Two DMA channel
  - Internal and External Transfer
  - Single and Burst
  - Buffered
  - 8bit, 16bit, 32bit Transfer
- JTAG & Debug Support
- Core & Boundary Scan Chains
- Little & Big Endian
- Package: 176-pin LQFP
- Operating Conditions
  - Core: 2.35-2.75v
  - I/O: 3.0v-3.6v
  - 0°C to 70°C (Commercial)

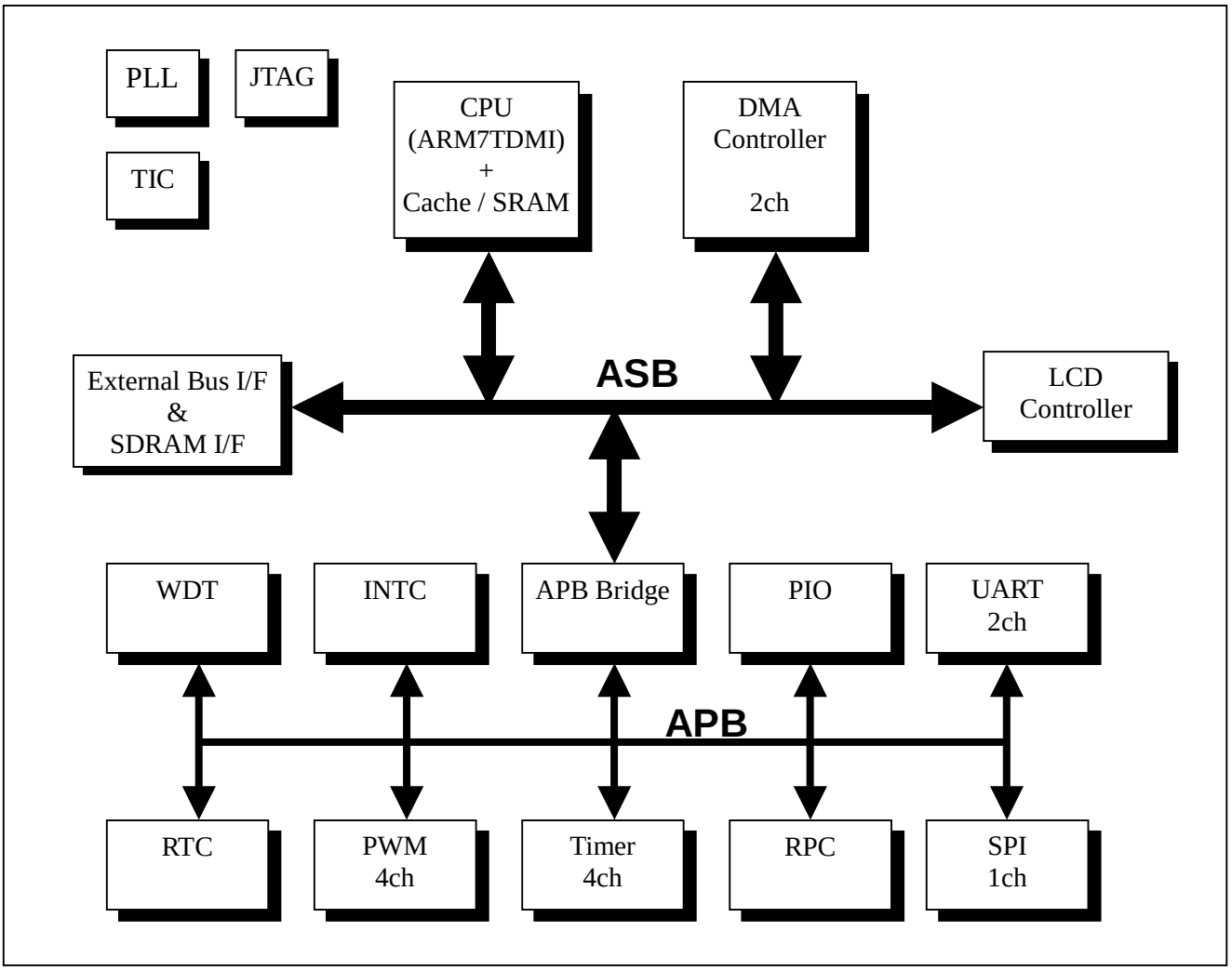


Figure 1. Block Diagram of LH79532

Functional Descriptions

ARM7TDMI™ Core

The LH79532 is built around the ARM7TDMI™ core. The ARM7TDMI™ is comprised of a Thumb-aware ARM7T processor, with Debugging (D), Enhanced Multiplier (M), In-Circuit-Emulation (I), and JTAG-style port to ease the development of application software, operating systems, and hardware.

- Thumb Aware Core
- Supports Powerful Standard 32-bit ARM Architecture/Instructions
- Supports Code Efficient 16-bit Thumb Architecture/Instructions
- Transparent, Real Time Decompression Of Thumb Instructions
- 64-bit Enhanced Multiplier with Accumulator
- Excellent Code Density
- Fully Static Design for Power Sensitive Applications
- Low Power Consumption
- High Performance
- Fast Interrupt Response with Minimal Context Switching
- Big Endian or Little Endian Mode
- Built-in Debug and ICE Support
- JTAG-style Port Based on the IEEE Standard 1149.1-1990

NOTE: Refer to ARM ARM7TDMI™ Data Sheet for additional information, including: Programmer's model, ARM instruction set, Thumb instruction set, instruction cycle operation, JTAG and debug interface.

LH79532 have a restriction of ARM7TDMI debug interface. If you set the hardware breakpoint using ICEBreaker to the cachable region in cache enabled state, it is possible not to break. To avoid this, use the software breakpoint.

Memory Interface Architecture

The LH79532 provides the following data-path-management resources on chip:

- 32-bit high-bandwidth internal data busses
- 4 kB of zero-wait-state instruction/data cache, configurable as write-back or write-through
- 4 kB of zero-wait-state scratch-pad (Local SRAM) (Shared resource with instruction/data cache)
- An 8-word posted-write buffer that accumulates and forwards outgoing data for slow memory
- An 8 Chip Select External Bus Controller for ROM/SRAM
- A Synchronous DRAM controller and 16-bit interface to a common program and graphics memory
- An External Bus Interface with 26 address and 32 bi-directional data pins
- A LCD Controller with build-in DMA
- A 2-channel general purpose DMA controller

Memory Map

All system resources accessible by the LH79532 are memory mapped. These include external resources (e.g. ROM, PROM, SRAM, SDRAM, External Peripherals) and internal resources (system configuration registers, peripheral configuration registers, and Local SRAM). Allocation of address ranges to each physical resource is accomplished by programming a number of segment start and size registers, allowing wide flexibility in partitioning the memory space. These configurable partitions will be called ‘segments’ throughout this document. They are programmed by means of several segment registers contained within each of the three system busses comprising the LH79532.

The broadest partitioning of memory space is its subdivision into four ‘regions’. The address range of each of these regions is fixed, according to the two highest-order of the 32 address bits. These regions define the broad type of resource being addressed. Some regions can only contain external SDRAM. Others can only contain external devices connected to the External Bus Interface. One region of address space is reserved for accessing the system configuration registers themselves, as well as many of the peripheral control registers. See Figure 2.

The lowest level partitioning of memory space is defined by programming up to eight separate memory segments, plus a default segment. Of these, up to 8 can be used to describe the address range and configure the features of different External Bus Interface peripherals; up to two can be used to describe the address range and features of external SDRAM devices; one (the default segment) is used after system reboot to immediately access boot ROM; and one may describe the Local SRAM. The segment’s segment-descriptor registers carry information about how to communicate with corresponding external devices, such as number of system clock cycles, type of SDRAM, etc., and access protection information such as cache-ability by the CPU, or queuing through the write-buffers. The storage of such information is distributed through several blocks within the LH79532, and thus must be programmed into each of the affected resource’s segment and descriptor registers before the corresponding resource can be used.

The following sections describe how each resource category must be programmed for proper resource-access configuration and address-range selection. In some cases, the same segment’s address range must be defined consistently for several of the system resources, so the same address mapping should be programmed into each of the affected peripheral’s segment registers. In other cases, the segment’s address range is fixed by hardware, so that access to a common resource may depend upon the perspective of the requesting bus master.

Memory is partitioned into segments, each of which associates a type of peripheral to a designated address-range and a set of controls over its access permissions and configuration:

- 1 segment activates System-level peripherals (e.g. memory map registers, cache configuration)
- 1 segment activates Local SRAM
- 8 segments define ROM/SRAM configurations
- 2 segments distinguish between SDRAM configurations
- 1 segment activates User-level peripherals such as UART and I/O ports

Memory map lookup tables are used by the bus decoders for block selection, privilege information, and configuration information such as bus latency. Their principal function is to decode addresses to select the appropriate peripheral destinations for routing data. To program these lookup-tables, they appear as 32-bit registers that are permanently mapped into the peripherals region of memory. Mapping information must be made available to each bus controller.

Bus Operation

The cache and DMA controllers perform the majority of their data transfers in 4-word sequential bursts within a single memory segment, maximizing throughput. A high hit rate to cache and/or Local SRAM provides good processor throughput with little interference to graphics updates or peripheral DMA. Code and Data memory can reside in either SRAM or SDRAM.

The following types of transactions require data traffic to traverse more than one of the busses. Depending on the system and application, code should be optimized to keep these transactions as infrequent as possible, since their interaction reduces the achievable system throughput:

1. Cache misses causing line-fills
2. Load/stores with the cache disabled
3. Stores while the cache is in write-through mode
4. Transfers between the processor and peripherals
5. DMA into Local SRAM
6. DMA into SDRAM

System Priorities

The LH79532 provides two independent operating busses. This provides for a large degree of concurrency among tasks, such as CPU/Cache transactions DMA transactions, and LCD refreshes. However, the programmer must take care to issue any inter-dependent tasks in the proper sequence. A busy high-priority bus master can monopolize control of its bus, obstructing lower-priority requestors. When programming concurrent tasks (such as DMA), care must be taken not to make a high-priority task dependent on completion of a competing, lower-priority task, as this could hang the system. SDRAM maintenance (for example, refresh) can affect latency of any SDRAM access. Refer to the SDRAM section for more details. Setting the ARBPR (Arbitration Priority Control) bit to '0' in the Reset and Power Controller's Arbitration register sets the priorities as follows:

Master Device / Access	ASB Bus	Peripheral Bus
LCDC Refreshes	1	-
DMA Channel 0	2	1
CPU	3	2
DMA Channel 1	4	3

NOTES:

- Priority (1 ⇨ Highest, 4 ⇨ Lowest)
- Default setting for ARBPR bit. ('0')

Alternatively, setting the ARBPR (Arbitration Priority Control) bit to '1' in the RPC's Arbitration register can change the priorities for DMA channel 1:

Master Device / Access	ASB Bus	Peripheral Bus
LCDC Refreshes	1	-
DMA Channel 0	2	1
CPU	4	3
DMA Channel 1	3	2

Memory Region

The system memory map of the LH79532 has two views, based on RPC_MAP (the Reset Power Control Memory Map; See Reset and Power Controller Section):

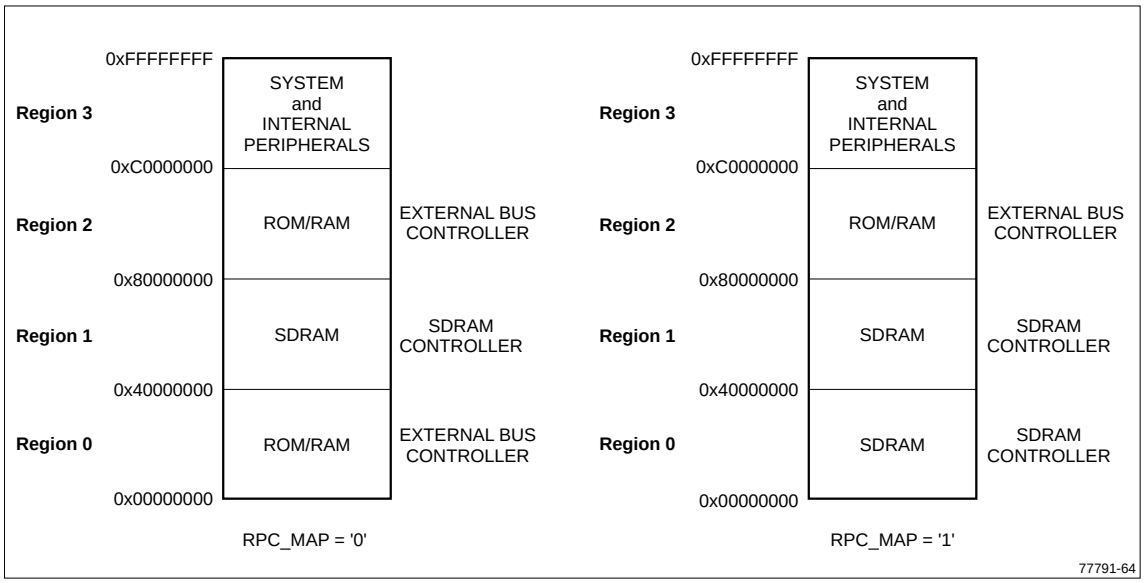


Figure 2. Memory Map

NOTES:

1. RPC_MAP is initialized to ‘0’ upon reset.
2. Local RAM can be mapped only in regions 0,1,and 2. See the Cache section for more details.
3. Right after reset or when RPC_MAP = ‘0’, ROM/RAM is mapped to two regions:
   0x00000000-0x3FFFFFFF
   0x80000000-0xBFFFFFFF
   - This means the same physical ROM/RAM memory can be accessed from both locations.
4. Programming RPC_MAP to ‘1’ will extend the SDRAM map to lower memory. This will map SDRAM to:
   0x00000000-0x3FFFFFFF
   0x40000000-0x7FFFFFFF
   - This means that the user can only access ROM/RAM from 0x80000000-0xBFFFFFFF.
5. If two segments have overlapping areas then the lower numbered segment is activated (Segment_0 has higher priority than Segment_1).

Supervisor Access to Internal Memory-Mapped Objects

Table 1 and Table 2 show the addresses, Remap functions, and Class of the various devices within the LH79532.

Table 1. Main Bus Memory Mapping

Address	REMAP = '0'	REMAP = '1'	Class
0x40000000	SDRAM	SDRAM	
0x80000000	Ext. ROM/RAM	Ext. ROM/RAM	
0xC0000000	-	-	
0xFFFF0000	-	-	
0xFFFF0400	-	-	
0xFFFF0800	DMA	DMA	Int.
0xFFFF2000	LCDC	LCDC	Int.
0xFFFF2400	-	-	
0xFFFF2800	-	-	
0xFFFFA000	MainASB	MainASB	Sys.
0xFFFFA400	EXTBIF	EXTBIF	Sys.
0xFFFFAC00	Cache Ctrl.	Cache Ctrl.	Sys.
0xFFFFB000	-	-	Sys.
0xFFFFC000	SDRAMC	SDRAMC	Sys.
0xFFFFC800	-	-	
0xFFFFE400	-	-	

NOTE: - : Reserved.

Table 2. Peripheral Bus Memory Mapping

Address	MACRO	Device Class
0xFFFF4000	UART0	Internal
0xFFFF4400	UART1	Internal
0xFFFF4800	-	Internal
0xFFFF4C00	PIO	Internal
0xFFFF5000	PWM	Internal
0xFFFF5400	SPI	Internal
0xFFFF5800	Counter/Timer	Internal
0xFFFF5C00	-	
0xFFFF7000	RTC	System
0xFFFF7400	INTC	System
0xFFFF7800	RPC	System
0xFFFF7C00	PLL	System
0xFFFF8000	WDT	System
0xFFFF8400	-	-
0xFFFF8800	-	

NOTE: - : Reserved.

Locked Instruction Sequences

The ARM7 instructions SWP and SWPB are supported by the LH79532 for its own use. Users should restrict external-semaphore SWP targets to non-bufferable memory space to assure memory coherency. External bus cycles, however, may not be atomic if the system includes multiple competing bus masters sharing the external bus interface.

Bus Error Handling

The bus decoders are responsible for flagging memory protection faults (during memory map lookup) or attempts to access unassigned areas of memory. The selected slaves are responsible for flagging all other types of access faults, such as configuration and addressing problems. All bus errors generate a BERROR interrupt.

Memory Segmentation

CPU Memory Protection Unit

The system supports eight programmable memory segments (0 to 7), and a default segment. The user must define these segments consistently for all bus decoders. See the EBI(External Bus Interface) and SDRAM blocks for descriptions of these registers. When an address is outside the area specified in at least one of these segments, the interface will respond with a SDRAM bus error.

The start (SegxStart) and size (SegxSize) registers determine the boundaries of the segment. (Where 'x' describes one of the numerical registers.) The configuration (SegxCfg) registers contain information about the system/user write/read privileges, cache-ability, and buffer-ability for each of the eight segments (0 to 7) and the default segment. The default segment is used after Reset to access boot ROM. It has its own configuration register SEGDefCfg. (See Default Segment Configuration Register, and RAM_SEG_REG Configuration.)

Register Summary

Table 3. CPU Memory Protection Unit Registers (Base Address: 0xFFFFAC00)

Offset from Base	Register	Name	Access	Size	Reset Value
0x00	Seg0Start	Segment 0 START	R/W	22[1]	0x000000
0x04	Seg1Start	Segment 1 START	R/W	22[1]	0x000000
0x08	Seg2Start	Segment 2 START	R/W	22[1]	0x000000
0x0C	Seg3Start	Segment 3 START	R/W	22[1]	0x000000
0x10	Seg4Start	Segment 4 START	R/W	22[1]	0x000000
0x14	Seg5Start	Segment 5 START	R/W	22[1]	0x000000
0x18	Seg6Start	Segment 6 START	R/W	22[1]	0x000000
0x1C	Seg7Start	Segment 7 START	R/W	22[1]	0x000000
0x20	Seg0Size	Segment 0 SIZE	R/W	5	0x00
0x24	Seg1Size	Segment 1 SIZE	R/W	5	0x00
0x28	Seg2Size	Segment 2 SIZE	R/W	5	0x00
0x2C	Seg3Size	Segment 3 SIZE	R/W	5	0x00
0x30	Seg4Size	Segment 4 SIZE	R/W	5	0x00
0x34	Seg5Size	Segment 5 SIZE	R/W	5	0x00
0x38	Seg6Size	Segment 6 SIZE	R/W	5	0x00
0x3C	Seg7Size	Segment 7 SIZE	R/W	5	0x00
0x40	Seg0Cfg[2]	Segment 0 Configuration	R/W	6	0x3C
0x44	Seg1Cfg[2]	Segment 1 Configuration	R/W	6	0x3C
0x48	Seg2Cfg[2]	Segment 2 Configuration	R/W	6	0x3C
0x4C	Seg3Cfg[2]	Segment 3 Configuration	R/W	6	0x3C
0x50	Seg4Cfg[2]	Segment 4 Configuration	R/W	6	0x3C
0x54	Seg5Cfg[2]	Segment 5 Configuration	R/W	6	0x3C
0x58	Seg6Cfg[2]	Segment 6 Configuration	R/W	6	0x3C
0x5C	Seg7Cfg[2]	Segment 7 Configuration	R/W	6	0x3C
0x60	SegDefCfg[2]	Default Configuration	R/W	6	0x3C

NOTES:

1. These registers must be written with a 32-bit value; the addresses are left-justified.
2. It is important that these registers be programmed compatibly with those in the SDRAM and External Bus I/F controllers. Those controllers also contain configuration registers that must be programmed compatibly with these registers.

All segments default on reset to R/W privileges (Supervisor), R/W privileges (User), non-bufferable, non-cacheable.

Cache and Local SRAM Description

The 4 kB combined instruction and data cache can be configured as 4 kB cache, or 4 kB SRAM. When configured in a cache mode, the cache is 4-way set-associative. The cache is designed to supply the CPU with data at a higher average bandwidth and shorter latency than main memory, while greatly reducing the traffic load on the main and external busses. The cache write-coherency policy can be configured as either write-back (also known as copy-back) or write-through. In both mode, the cache controller must evict a line (four words). It uses a modified, least-recently-used (LRU) algorithm to select the victim line. In write-through mode, data from write-hits are written into the cache and to the write buffer, to be forwarded to external memory.

Local SRAM

4kbytes of on-chip data SRAM is shared between cache and local SRAM. 4kbyte of fast (no wait state) local scratchpad-memory can be made available. Its size is 4kbyte, organized according to the SM bits of the CacheControlReg. The data contents of Local SRAM are undefined after reset. This memory uses the cache data-memory array according to the SM bits in the CacheControlReg. The user must not change the Local SRAM configuration while live data resides in it, nor while live data resides in the region of external memory where Local SRAM was or will be overlaid.

The SRAMTag register contains the start address of the local SRAM area.

The system/user privileges for the local SRAM are under the control of the Segment Configuration Register associated with the address range of the local SRAM, or the default segment configuration register if no SegxCfg is associated with the local SRAM address.

Cache Configurations

4kbytes of on-chip data SRAM is shared between cache and local SRAM. Two possible operating modes give the system designer a high degree of flexibility. The three modes are shown in Table 4.

Table 4. Memory Combinations

Mode	Cache	SRAM
Cache Mode	4 kbyte	0
SRAM Mode	0	4kbyte

Register Summary

Table 5. Local SRAM Configuration Registers

Address	Register	Name	Access	Size	Reset Value
0xFFFFAC84	SRAMTag	Local SRAM Start Address Register	R/W	20	0xFFFE0000
0xFFFFAC88	SRAMWait	Local SRAM Wait Register for Main Bus	R/W	1	0x00000000
0xFFFFA000	MAINASBCTRL	Local SRAM Start Address Register for Main Bus	R/W	32	0x00000000

Table 6. Cache Contorol Register

Address	Register	Name	Access	Size	Reset Value
0xFFFFAC80	CacheControlReg	Cache Control Register	R/W	10	*

External Bus Interface (EBI)

The External Bus Interface (EBI) supports standard x8, x16 and x32 SRAM, ROM, Flash, SDRAM and memory-mapped peripherals on a bi-directional data bus.

Features

- Flexible programmable memory interface with SDRAM controller.
- Supports SRAM, Normal or Page Mode Flash, ROM and SDRAM
- 26-bit External Address Bus (When in SDRAM access, 15-bits are valid)
- 32-bit External Data Bus (x8, x16 and x32)
- nWE0, nWE1 ,nWE2, nWE3 and nRE support for Byte/Halfword/Word Writes and Reads
- Eight Banks (64MB) selected through dedicated Chip Enables (nCE7: nCE0)
- Programmable Address Setup time to nCE
- Programmable nCE Setup and Hold times to nWE & nRE
- Programmable memory cycle and bus turnaround times (Inserts up to 32 internal Wait states and up to 8 Idle states)
- Selectable boot-up from x8 or x16 Static Memory
- Endian Control
- Supports external WAIT requests
- Programmable power-down state of data bus (Three-state or Output)
- Programmable SDRAM Controller
  - Programmable address multiplexing (64 Mb, 128 Mb, 256 Mb)
  - 32 bit, 16 bit or 8 bit external data bus access
  - 32 bit, 16 bit or 8 bit internal data bus access
  - Big or Little Endian
  - 1 or 2 block, 2 or 4 banks in one block
  - Compatible with 100 MHz and 133 MHz SDRAMs
  - Programmable timing of CAS latency, Refresh rate, t_{RC}, t_{RAS}, t_{RP}, t_{RCD}, t_{XSR}
  - Full page burst access
  - Page Mode accesses with up to 8 open pages
  - Supports Self refresh and Auto refresh
  - Programmable Refresh Rate

Page Mode Support

The External Bus Interface supports 4- and 8- word page memory devices**.** The first cycle of an access has non-sequential (N-cycle) timing**.** Subsequent sequential accesses have sequential (S-cycle) timing until a page boundary is reached, at which time a N-cycle timing is resumed**.** This page mode function only affects the Read cycle.

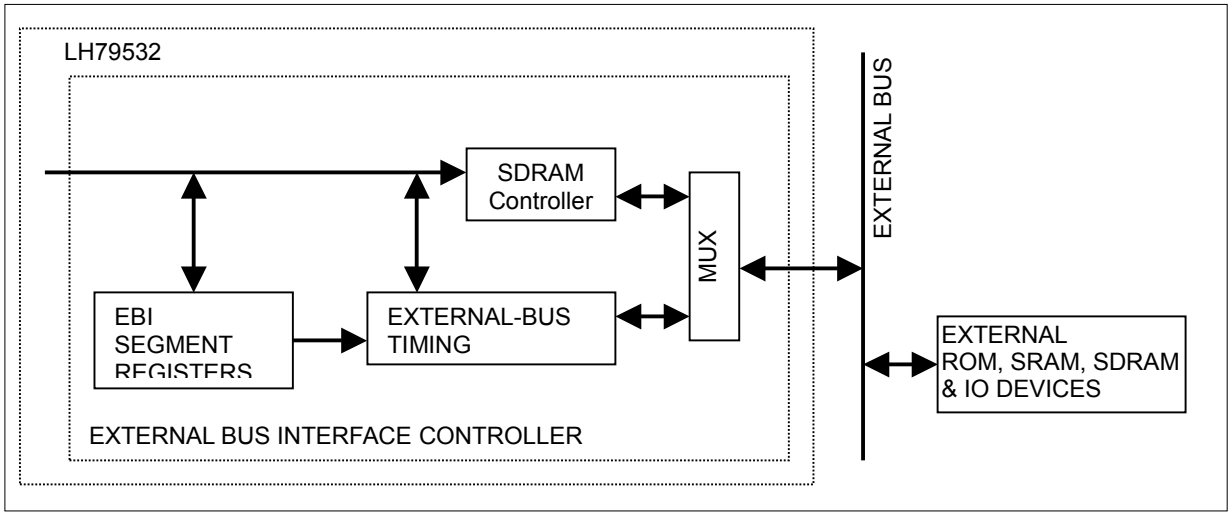


Figure 3. External Bus Interface Block Diagram

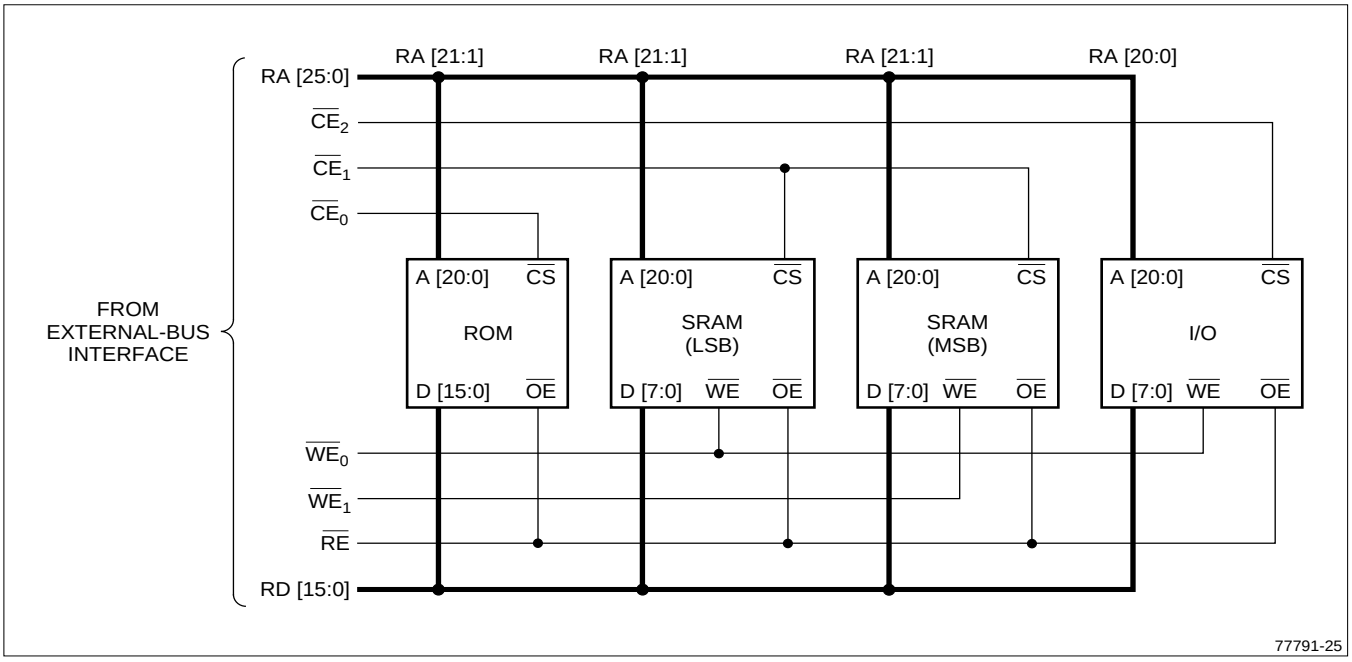


Figure 4. Example of External Bus Connections

External Bus Interface Memory Map Registers

The user must define these segments consistently for all bus decoders. See the CPU Memory Protection Unit and SDRAM Memory Map sections for descriptions of the additional registers.

The External Bus Interface segment registers must be programmed with start and size values that define identical areas as the segments defined in the CPU Memory Protection Unit. Note that the start values and size values must be consistent. (The External Bus Interface does not support 0, 1, 2, and 4 k segment sizes.)

Register Summary

Table 7. EBI Segment Map Registers (Base address: 0xFFFFA400)

Offset from Base	Register	Access	Reset Value	Bit Position												
				31	30	29:13	12	11	10	9	8	7	6	5	4	3:0
0x00	RAM_SEG_REG0	R/W	0x8000000D	E	-	<START>	-	-	-	-	-	-	-	-	-	<SIZE>
0x04	RAM_SEG_REG1	R/W	0x0	E	-	<START>	-	-	-	-	-	-	-	-	-	<SIZE>
0x08	RAM_SEG_REG2	R/W	0x0	E	-	<START>	-	-	-	-	-	-	-	-	-	<SIZE>
0x0C	RAM_SEG_REG3	R/W	0x0	E	-	<START>	-	-	-	-	-	-	-	-	-	<SIZE>
0x10	RAM_SEG_REG4	R/W	0x0	E	-	<START>	-	-	-	-	-	-	-	-	-	<SIZE>
0x14	RAM_SEG_REG5	R/W	0x0	E	-	<START>	-	-	-	-	-	-	-	-	-	<SIZE>
0x18	RAM_SEG_REG6	R/W	0x0	E	-	<START>	-	-	-	-	-	-	-	-	-	<SIZE>
0x1C	RAM_SEG_REG7	R/W	0x0	E	-	<START>	-	-	-	-	-	-	-	-	-	<SIZE>

Table 8. EBI Control Registers (Base Address: 0xFFFFA400)

Offset from Base	Register	Access	Reset Value	Bit Position												
				31:24	23	22:19	18	17	16	15	14	13	12	11:10	9:5	4:0
0x20	RAM_CTL_REG0	R/W	0	-	MType	IDLE	RS	RH	WS	WH	AS	AH	P	S*	N-speed	S-speed
0x24	RAM_CTL_REG1	R/W	0	-	MType	IDLE	RS	RH	WS	WH	AS	AH	P	S	N-speed	S-speed
0x28	RAM_CTL_REG2	R/W	0	-	MType	IDLE	RS	RH	WS	WH	AS	AH	P	S	N-speed	S-speed
0x2C	RAM_CTL_REG3	R/W	0	-	MType	IDLE	RS	RH	WS	WH	AS	AH	P	S	N-speed	S-speed
0x30	RAM_CTL_REG4	R/W	0	-	MType	IDLE	RS	RH	WS	WH	AS	AH	P	S	N-speed	S-speed
0x34	RAM_CTL_REG5	R/W	0	-	MType	IDLE	RS	RH	WS	WH	AS	AH	P	S	N-speed	S-speed
0x38	RAM_CTL_REG6	R/W	0	-	MType	IDLE	RS	RH	WS	WH	AS	AH	P	S	N-speed	S-speed
0x3C	RAM_CTL_REG7	R/W	0	-	MType	IDLE	RS	RH	WS	WH	AS	AH	P	S	N-speed	S-speed
0x40	EXTBIFCtrl	R/W	0	-	-	-	-	-	-	-	-	-	-	-	-	Z

NOTE: *S-bit (bit 11:10): Reset value for RAM_CTL_REG0 is the same as the value of BOOT [1:0] signals.
If N-speed = S-Speed, then Page Mode is disabled.

SDRAM Controller

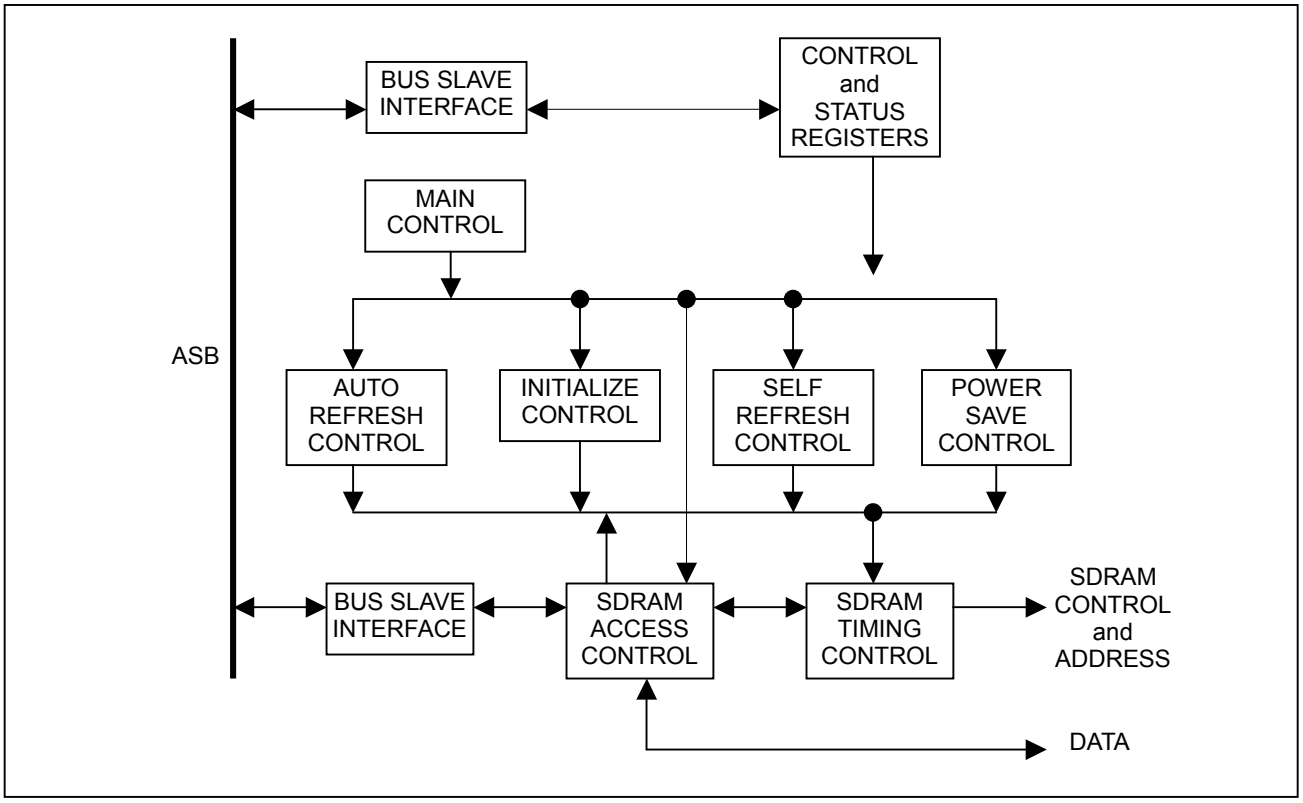


Figure 5. SDRAM Control Block Diagram

Register Summary

Table 9. SDRAMC Register map (Base address: 0xFFFFC000)

Offset from Base	Register	Access	Size	Reset Value	Bit Position																																	
					31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	SDRAM_SEG0_REG	R/W	32	0x000000C0	E	-	Start Address (17 bits)														-	-	-	-	-	-	-	-	-	-	Size						
0x004	SDRAM_SDR0_REG	R/W	10	0x00000003	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CAW		-	-	RAW		-	EBW1	BAW	EBW0
0x008	SDRAM_SEG1_REG	R/W	32	0x000000C0	E	-	Start Address (17 bits)														-	-	-	-	-	-	-	-	-	-	Size						
0x00C	SDRAM_SDR1_REG	R/W	10	0x00000003	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CAW		-	-	RAW		-	EBW1	BAW	EBW0
0x010	SDRAM_CONTROL	R/W	4	0x00000000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PME	PSE	SREE	SCEE
0x014	SDRAM_STATUS	R	3	0x00000000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PSS	SRS	SCS
0x018	SDRAM_TIMING0	R/W	10	0x00000000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RCC		-	RIF						
0x01C	SDRAM_TIMING1	R/W	32	0x00000000	-	-	-	-	t_{RC}				-	-	-	-	-	-	t_{RCD}		-	-	-	-	-	-	t_{RAS}				-	-	-	-	-	t_{RP}		
0x020	SDRAM_TIMING2	R/W	32	0x00000000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	t_{WR}	-	-	-	-	-	-	CAL				-	-	-	-	t_{XSR}			

Address Mapping

Following Table show the supported configurations of SDRAMs and the relationship between external SDRAM multiplexed address bus and internal address bus.

Table 10. Address Multiplexing (32 bit External Data Bus Mode)

Total Size	Chip Size	Configuration				Addr-MUX	Bits	A14	A13	A12	A11	A10	A9	A8	A [7:0]
		Words	Bits	Banks	Chips										
8MB	16M	1M	8	2	4	Row	11			22		21	20	19	18:11
						Col	9			22		AP		10	9:2
4MB	16M	0.5M	16	2	2	Row	11			21		20	19	18	17:10
						Col	8			21		AP			9:2
8MB	64M	0.5M	32	4	1	Row	11		22	21		20	19	18	17:10
						Col	8		22	21		AP			9:2
16MB	64M	1M	16	4	2	Row	12		23	22	21	20	19	18	17:10
						Col	8		23	22		AP			9:2
32MB	64M	2M	8	4	4	Row	12		24	23	22	21	20	19	18:11
	128M	2M	16	4	2	Col	9		24	23		AP		10	9:2
64MB	64M	4M	4	4	8	Row	12		25	24	23	22	21	20	19:12
	128M	4M	8	4	4	Col	10		25	24		AP	11	10	9:2
128MB	128M	8M	4	4	8	Row	12		26	25	24	23	22	21	20:13
						Col	11		26	25	12	AP	11	10	9:2
64MB	256M	4M	16	4	2	Row	13	23	25	24	22	21	20	19	18:11
						Col	9		25	24		AP		10	9:2
128MB	256M	8M	8	4	4	Row	13	24	26	25	23	22	21	20	19:12
						Col	10		26	25		AP	11	10	9:2
256MB	256M	16M	4	4	8	Row	13	25	27	26	24	23	22	21	20:13
						Col	11		27	26	12	AP	11	10	9:2

Table 11. Address Multiplexing (16 bit External Data Bus Mode)

Total Size	Chip Size	Configuration				Addr-MUX	Bits	A14	A13	A12	A11	A10	A9	A8	A [7:1]	A0
		Words	Bits	Banks	Chips											
4MB	16M	1M	8	2	2	Row	11			21		20	19	18	17:11	10
						Col	9			21		AP		9	8:2	HA
2MB	16M	0.5M	16	2	1	Row	11			20		19	18	17	16:10	9
						Col	8			20		AP			8:2	HA
8MB	64M	1M	16	4	1	Row	12		22	21	20	19	18	17	16:10	9
						Col	8		22	21		AP			8:2	HA
16MB	64M	2M	8	4	2	Row	12		23	22	21	20	19	18	17:11	10
	128M	2M	16	4	1	Col	9		23	22		AP		9	8:2	HA
32MB	64M	4M	4	4	4	Row	12		24	23	22	21	20	19	18:12	11
	128M	4M	8	4	2	Col	10		24	23		AP	10	9	8:2	HA
64MB	128M	8M	4	4	4	Row	12		25	24	23	22	21	20	19:13	12
						Col	11		25	24	11	AP	10	9	8:2	HA
32MB	256M	4M	16	4	1	Row	13	22	24	23	21	20	19	18	17:11	10
						Col	9		24	23		AP		9	8:2	HA
64MB	256M	8M	8	4	2	Row	13	23	25	24	22	21	20	19	18:12	11
						Col	10		25	24		AP	10	9	8:2	HA
128MB	256M	16M	4	4	4	Row	13	24	26	25	23	22	21	20	19:13	12
						Col	11		26	25	11	AP	10	9	8:2	HA

Table 12. Address Multiplexing (8 bit External Data Bus Mode)

Total Size	Chip Size	Configuration				Addr-MUX	Bits	A14	A13	A12	A11	A10	A9	A8	A [7:2]	A1	A0
		Words	Bits	Banks	Chips												
2MB	16M	1M	8	2	1	Row	11			20		19	18	17	16:11	10	9
						Col	9			20		AP		8	7:2	HA	BA
8MB	64M	2M	8	4	1	Row	12		22	21	20	19	18	17	16:11	10	9
						Col	9		22	21		AP		8	7:2	HA	BA
16MB	64M	4M	4	4	2	Row	12		23	22	21	20	19	18	17:12	11	10
	128M	4M	8	4	1	Col	10		23	22		AP	9	8	7:2	HA	BA
32MB	128M	8M	4	4	2	Row	12		24	23	22	21	20	19	18:13	12	11
						Col	11		24	23	10	AP	9	8	7:2	HA	BA
32MB	256M	8M	8	4	1	Row	13	22	24	23	21	20	19	18	17:12	11	10
						Col	10		24	23		AP	9	8	7:2	HA	BA
64MB	256M	16M	4	4	2	Row	13	23	25	24	22	21	20	19	18:13	12	11
						Col	11		25	24	10	AP	9	8	7:2	HA	BA

NOTES:
1. A0 - A14 refer to the external pins named RA0 - RA14.
2. Figures in the table refer to the Main address bus bit locations.
3. Shaded entries are ‘bank select’.
4. AP (in A10 of ‘Col’ addresses) refers to optional bank select.
5. Null entries are ‘don’t care’, but stable.
6. HA is generated by SDRAM Controller when in word (32-bit) access. Otherwise HA is Main bus address <1>
7. BA is generated by SDRAM Controller when in word (32-bit) and half word (16 bit) access. In byte access BA is Main bus address <0>

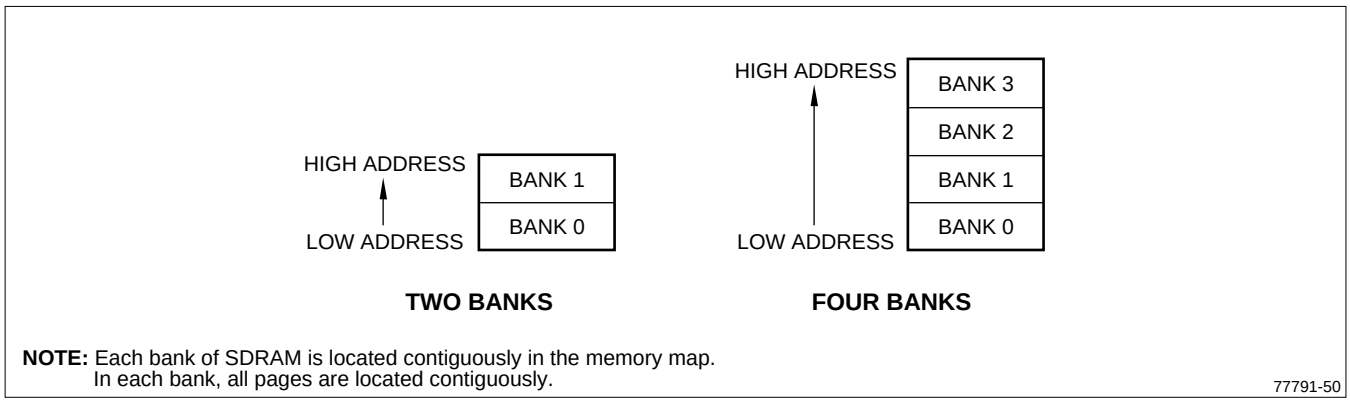


Figure 6. Memory Map

DMA Controller

Overview

The DMA controller (DMAC) in the LH79532 handles data transfers between memory and peripherals; and from memory to memory, without the intervention of the ARM7TDMI™ Core. This controller provides two independent channels (Channel 0 and Channel 1) that operate autonomously (subject to priority conflict resolution).

A DMA transfer consists of multiple DMA cycles. Each DMA cycle is either a single DMA cycle or a burst DMA cycle. A single DMA cycle consists of one read bus cycle followed by a write bus cycle. A burst DMA Cycle consists of four read bus cycles followed by four write bus cycles.

DMA transfers can be hardware or software triggered. A DMA transfer is software triggered by writing a value to the appropriate DMACModeCtrlReg, which disables the hardware triggers and sets the software trigger bit. A DMA transfer is hardware triggered when hardware triggers are enabled in the appropriate DMACModeCtrlReg and the peripheral that is enabled by the appropriate CFG field of the DMACRequestSource requests a transfer.

Each DMA cycle is buffered; the data is read in from the source and stored internally before being written to the destination.

The DMACCounterReg registers control the number of items to be transferred. The size of each item can be 8, 16 or 32 bits controlled by DMACModeCtrlReg. The item size must be the same for the source and the destination.

Transfers can be between external devices or memory; between internal devices or memory; or between an internal or external memory or device. The source address DMACSrcAddrReg registers and the destination address DMACDstAddrReg registers can each either be fixed or can increment with each DMA cycle.

Status for the current DMA transfer is stored in the DMACStatusReg. This status reflects whether a DMA request has been made, a DMA transfer is in progress, or if an error has occurred.

The DMAC can be configured via the DMACControlReg to enable the DMA interrupt to suspend the DMA transfers and to generate interrupts upon the completion of a DMA transfer (‘end of transfer interrupt’) or if the bus controller reports an error during a DMA bus cycle (‘transfer error interrupt’). Regardless of whether the transfer interrupt is masked the DMA will assert the appropriate DMACStatusReg: ERR bit, and END bit, and then stop the current DMA transaction.

The source and destination address registers can be placed in double buffer mode to support hardware DMA requests or re-triggering.

In the event of a conflict for shared resources between the channels, channel 0 has the higher priority. If a channel 1 DMA cycle is suspended in this manner, it will continue when the channel 0 bus cycle terminates.

For more details, and information about other control registers, see ‘Register Descriptions’ in this section.

Register Summary

Table 13. DMA Control Registers (Base Address: 0xFFFF0800)

Offset from Base	Register	Access	Size	Reset Value	Description
0x00	DMACModeCtrlRegCh0	R/W	9	0	Controls the triggers, transfer mode, width and type, bus lock and release for channel 0
0x04	DMACSrcCtrlRegCh0	R/W	1	0	Controls whether the source address for channel 0 increments or is frozen.
0x08	DMACDstCtrlRegCh0	R/W	1	0	Controls whether the destination address for channel 0 increments or is frozen.
0x0C	DMACStatusRegCh0	R	4	0	Reflects the status of DMA transfers on channel 0
0x10	DMACSrcAddrRegCh0	R/W	32	0	Current source address for channel 0 (See also buffering modes under DMACControlReg)
0x14	DMACDstAddrRegCh0	R/W	32	0	Current destination address for channel 0 (See also buffering modes under DMACControlReg)
0x18	DMACCounterRegCh0	R/W	24	0	The number of items that remain to be moved in the transfer currently ongoing in Channel 0 (See also buffering modes under DMACControlReg)
0x20	DMACModeCtrlRegCh1	R/W	9	0	Controls the triggers, transfer mode, width and type, bus lock and release for channel 1
0x24	DMACSrcCtrlRegCh1	R/W	1	0	Controls whether the source address for channel 1 increments or is frozen.
0x28	DMACDstCtrlRegCh1	R/W	1	0	Controls whether the destination address for channel 1 increments or is frozen.
0x2C	DMACStatusRegCh1	R	4	0	Reflects the status of DMA transfers on channel 1
0x30	DMACSrcAddrRegCh1	R/W	32	0	Current source address for channel 1 (See also buffering modes under DMACControlReg)
0x34	DMACDstAddrRegCh1	R/W	32	0	Current destination address for channel 1 (See also buffering modes under DMACControlReg)
0x38	DMACCounterRegCh1	R/W	24	0	The number of items that remain to be moved in the transfer currently ongoing in Channel 1 (See also buffering modes under DMACControlReg)
0x40	DMACControlReg	R/W	11	0	Controls interrupt and buffer modes and flags for both channels
0x80	DMACRequestSource	R/W	10	0	Enables and controls the polarity of hardware DMA REQ signals.
0x84	DMACIntSourceCh0	R/W	28	0	Enables individual interrupt sources to suspend an active DMA transfer in channel 0
0x88	DMACIntSourceCh1	R/W	28	0	Enables individual interrupt sources to suspend an active DMA transfer in channel 1
0x8C	DMACAckSelect	R/W	4	0	DMA Acknowledge select control for each peripheral
0x90	DMACExtReqCtl	R/W	8	0	DMA External Request control

Real Time Clock (RTC)

Overview

The RTC is a real time clock with a programmable timer and calendar. The clock input comes from either the recommended internal 32.768 kHz crystal oscillator or an external 32.768 kHz clock source. The RTC features a digital clock with an alarm function as well as an auto-calendar function. Other functions include:

- Input clock conversion to a real time clock.
- Settable timer, alarm and calendar.
- Current time and date output.
- Control of the RTC operation modes using RCSR (RTC control status register).
- Backup Power Operation

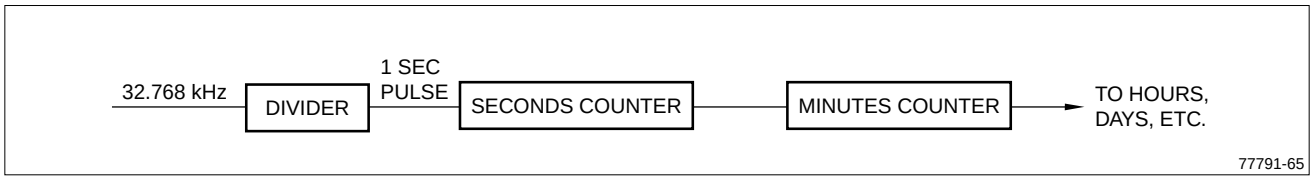


Figure 7. RTC Divider Block Diagram

Functional Description

The RTC has 8 8-bit registers to store and count real time for second, minutes, hour, day, week, month, year and century. The auto-calendar function operates from 1st January 1901 00:00:00 to 31st December 2099 23:59:59. It is important to note that the RTC is connected to the same power supply as the PLL, and is not reset by nRESETi.

Register Summary

Table 14. RTC Registers (Base Address: 0xFFFF7000)

Offset from Base	Register	Value	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00	Second	00 - 59	-	40^{s}	20^{s}	10^{s}	8^{s}	4^{s}	2^{s}	1^{s}
0x04	Minute	00 - 59	-	40^{m}	20^{m}	10^{m}	8^{m}	4^{m}	2^{m}	1^{m}
0x08	Hour	00 - 31 80 - A3	'0' 12^{h} '1' 24^{h}	-	AM/PM 20^{h}	10^{h}	8^{h}	4^{h}	2^{h}	1^{h}
0x0C	Day M	01 - 31	-	-	20^{d}	10^{d}	8^{d}	4^{d}	2^{d}	1^{d}
0x10	Day W	00 - 06	-	-	-	-	-	4	2	1
0x14	Month	01 - 12	-	-	-	10^{m}	8^{m}	4^{m}	2^{m}	1^{m}
0x18	Year	00 - 99	80^{y}	40^{y}	20^{y}	10^{y}	8^{y}	4^{y}	2^{y}	1^{y}
0x1C	Century	00 - 01	-	-	-	-	-	-	-	Century
0x20	MinD	00 - 59	-	40^{m}	20^{m}	10^{m}	8^{m}	4^{m}	2^{m}	1^{m}
0x24	HourD	00 - 23	-	-	AM/PM 20^{h}	10^{h}	8^{h}	4^{h}	2^{h}	1^{h}
0x28	Check	-	*	*	*	*	*	*	CLKSEL	CLKEN
0x2C	RCSR	-	EALM	-	ALM*	I4	I3	MOD	CHLD	ADJ

NOTES: - : Unused bit.
* : Reserved. Read-only bit.
All the address and the values of the above table are in hexadecimal.

Reset and Power Controller (RPC)

Overview

The Reset and Power Controller (RPC) is used to control the CPU and peripheral clocks, system reset and power. The system clock is chosen from the external clock (XCLKIN) or the PLL clock generated from the crystal output.

Features

The RPC provides the following power modes:
- Active
- Standby
- Sleep
- Stop
- Stop2

The RPC generates the following signals:
- Reset output (nRESETO)
- System clock with a selectable source from either:
- External clock (XCLKIN)
- Clock generated by PLL controller
- UART clock with a selectable source from either:
- System clock
- External clock (UCLK)
- Internal bus clock
- Divided system clock (×1, ×1/2, ×1/4, ×1/8)

Power Mode Operation

The RPC manages the system power. Table 15 summarizes the state of operation for the various peripherals.

Table 15. Power Mode

Function	Clock Source	Power Mode				
		Active	Standby	Sleep	Stop	Stop2[6]
ARM7TDMI™, Cache, WDT	SYSCLK	On	Halt	Halt	Halt	Halt
SDRAM Controller, DMA, PIO, EBI	SYSCLK	On	On	Off	Off	Off
SDRAM Self-Refresh	—	On/Off	On/Off	On/Off	On/Off	On/Off
RTC	XTLCLK	On/Off	On/Off	On/Off	On/Off	Off
	RTC_CLKIN[1]	On	On	On	On	On
UART0, UART1	SYSCLK	On/Off	On/Off	Off	Off	Off
	UCLK[1]	On	On	On	On	On
CT0, CT1, CT2, CT3	XCLK	On/Off	On/Off	On/Off	Off	Off
LCDC	SYSCLK or MCLK	On/Off	On/Off	Off	Off	Off
SPI	SYSCLK	On/Off	On/Off	Off	Off	Off
PWM	XCLK	On/Off	On/Off	On/Off	Off	Off
PLL[5]	XTLCLK	On	On	On	Off	Off
Crystal Oscillator	—	On	On	On	On	Off
Running Clocks		SYSCLK[3]	SYSCLK[3]			
		XCLK[2]	XCLK[2]	XCLK[2]		
		XTLCLK[4]	XTLCLK[4]	XTLCLK[4]	XTLCLK[4]	

NOTES:
[1] Disabling these modules' clocks will lower the total system power without disabling the peripheral.
[2] XCLK is either PLLCLK (PLL-driven clock) when CLKSEL = '0' or XCLKIN (External clock input) when CLKSEL = '1'.
[3] SYSCLK is the system clock. It is the same as XCLK but is halted in sleep mode.
[4] XTLCLK is the crystal oscillator clock.
[5] The PLL is turned off while XCLKIN is selected.
[6] Stop2 mode can only be selected if PLLCLK is selected.

Register Summary

The base address of the RPC is 0xFFFF7800. The address of any particular register from this base address is detailed in Table 16.

Table 16. RPC Registers (Base Address: 0xFFFF7800)

Offset from Base	Regisger	Name	Access	Size	Reset Value
0x00	PowerMode	Power Mode Register	W	3	0x0
0x04	Identification	ID Register	R	16	ID code
0x08	MemoryMap	Memory Map Control Register	R/W	1	0x0
0x0C	ResetStatus	Reset Status Register	R	2	0x1
0x10	ResetStatusClear	Reset Status Clear Register	W	2	—
0x14	CPUClockCtrl	CPU Clock Control Register	R/W	2	0x0
0x18	MacroClockCtrl	Macro Clock Control Register	R/W	16	0x0000
0x1C	MacroClockSel	Macro Clock Select Register	R/W	5	0x00
0x20	SoftReset	Soft Reset Register	R/W	1	0x0
0x24	Arbitration	Arbitration Register	R/W	2	0x0
0x28	DecodeMode	Decode Mode Register	R/W	1	0x0
0x34	Ir2MCkSel	IrDA 2MHz Clock Select Register	R/W	8	0x0

PLL Controller

Overview

Figure 8 shows a block diagram of the PLL controller circuit and interfacing to the PLL and crystal oscillator. The PLL controller disables the PLL for different power-down modes and controls the varying of the PLL frequency. It also controls the disabling of the on-chip crystal oscillator. The PLL controller also provides for an orderly start-up until the crystal oscillator stabilizes and the PLL acquires lock. If it is desired to change the operating frequency during normal operation, the PLL controller will ensure a smooth glitch-free transition between the old and new frequencies.

The PLL controller can alter the PLL-based system clock frequency in two ways. The first method requires reprogramming the PLL clock divider with a new value. This causes all clocks to halt until the PLL resynchronizes to the new frequency. With this method, the PLL frequency can be 2 to 4096 times the crystal oscillator frequency in integer increments with the restriction that the PLL frequency must be kept within its limits of operation. The second method for altering the PLL-based system clock frequency is to reprogram the prescaler with a new value. There is no interruption of clock activity with this method. The prescaler can divide the PLL clock by 2, 4, 8, 16, or 32.

The PLL controller provides a 'Loss of Lock' interrupt to the interrupt controller. This interrupt is asserted when the PLL has lost lock for reasons other than startup, halt modes, or a frequency change. It is deasserted when lock is reacquired.
The PLL controller contains a 'warm-up' counter that prevents the crystal oscillator output from being fed to the PLL until the crystal oscillator becomes stable following reactivation. The counter is driven by the crystal oscillator output and is 16 bits wide. On power-up, it defaults to its maximum count of 65535 crystal oscillator cycles but may be reprogrammed to a smaller value via the PLL_WUC register. Reprogramming the count to zero turns off the warm-up counter.
The warm-up counter is in effect after exiting STOP2 mode. On power up, it is expected that nRESETi is to be held asserted until the crystal oscillator stabilizes. The PLL controller keeps the PLL disabled while nRESETi remains asserted.

On power-up, the PLL multiplier frequency (PLL_FREQ) is set to 0x400 (1024) and the PLL Prescaler (PLL_PSR) is set to 0b100000 (32). This yields a power-up frequency of 1.0496 MHz.

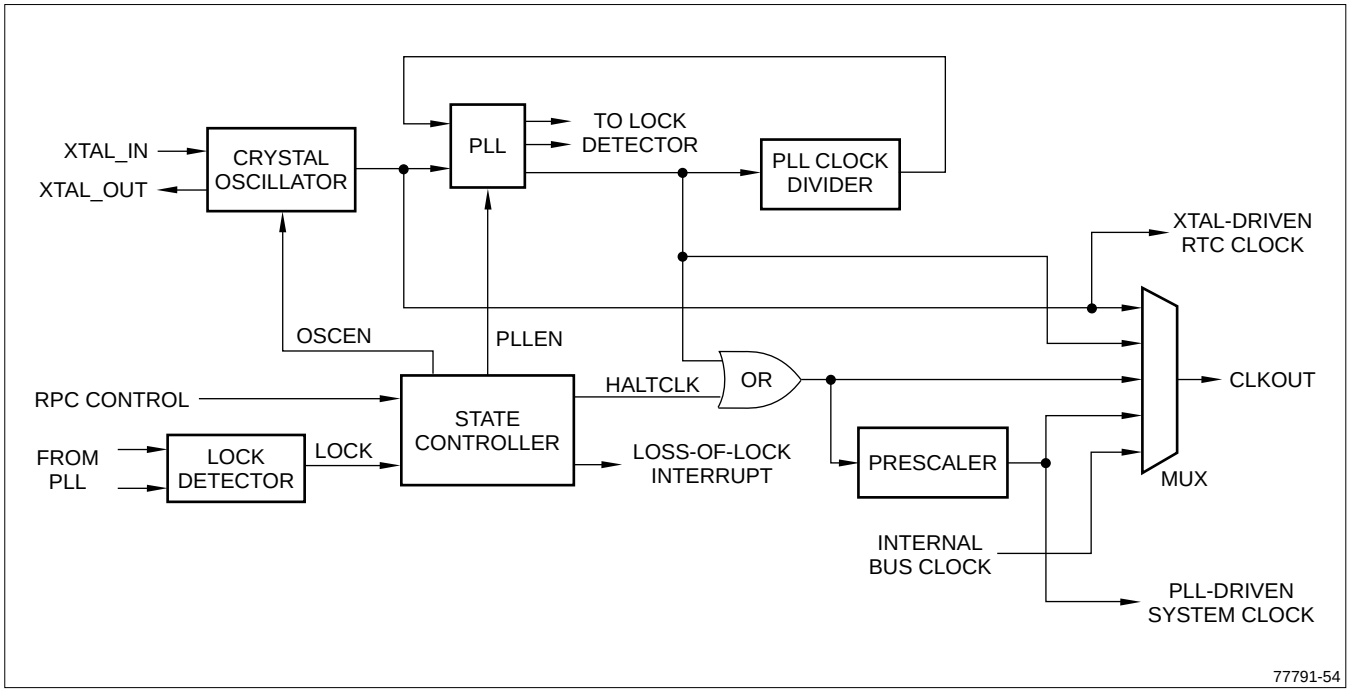


Figure 8. PLL Controller block diagram

Register Summary

Table 17. PLL Registers (Base Address: 0xFFFF7C00)

Offset from Base	Register	Name	Access	Size	Reset Value
0x08	PLL_CTL	PLL Control Register	R/W	10	0b0000100011 if CLKSEL pin is '0' 0b0001100010 if CLKSEL pin is '1'
0x0C	PLL_FREQ	PLL Frequency Divide Count Register	R/W	12	0x400
0x10	PLL_PSR	PLL Prescale Count Register	R/W	6	0b100000
0x20	PLL_WUC	PLL Warm-Up Count Register	R/W	16	0xFFFF

LCD Controller

Features

- 500 Hz Maximum Frame Rate
- Internal or External Clock Source
- 256 x 16-bit Palette RAM
- Memory Interface
  - 32-bit wide DMA access to external frame buffer
  - Double Buffering to support animation
  - Dual Programmable 16 x 32-bit FIFO DMA buffers
  - Little Endian, Big Endian and WinCE pixel data formats
- Interrupt event generation
- LCD Interface
  - Programmable resolution up to 1,024 x 768 pixels
  - Programmable sync timing
  - Single or Dual Scan panels
  - Produces frame and line syncs, pixel-clock, control, enable and AC-bias signals
  - 18-bit wide Panel Data Bus
  - Power Down Modes and LCD Power Sequencing
- STN Monochrome / Gray format:
  - 15 shades of synthesized gray scale
  - 1, 2, or 4 bits per pixel
  - 4- or 8-bit wide Panel Data Bus
- STN Passive Color format:
  - 256 Palletized Colors or 3,375 Direct Colors
  - 4-, 8-, or 16-bit Panel Data Bus
- TFT Active Color format:
  - Color palette accepts 1, 2, 4, or 8 bits per pixel providing up to 256 out of 64K colors
  - 18-bit Panel Data Bus
  - 16 bits per pixel un-palletized for over 64K Direct Colors

Overview

The LH79532 Color Liquid Crystal Display Controller translates a bit-mapped graphics image 'frame' residing in memory into the specialized sequences required to run active, passive, color, or monochrome LCD panel drivers. Image data to periodically refresh the display is automatically downloaded from graphics memory through the EBI port, achieving high display bandwidth for good resolution and refresh rates.

Two programmable FIFOs buffer the incoming pixel-data stream for single or dual-panel displays. A built-in 256 x 16 bit Palette RAM is also provided. This can enhance storage density and bandwidth by mapping 16 bit gray-scale or RGB panel data (5 x 5 x 5 bits per color plus 1 intensity bit) from 8 or fewer bits of pixel data.

For Thin Film Transistor (TFT) color displays, the pixel data can be directly displayed, providing up to 64 k direct simultaneous colors. Or when used to address the palette, up to 256 out of 64 k palletized colors can be selected. For Super Twisted Nematic (STN) displays, the algorithmic gray-scale pattern generator can synthesize up to 15 gray shades, or 15 saturation levels per color component for up to 3,375 hues.

Timings and configurations can be programmed to interface to a variety of LCD panels. Control signals are generated for pixel clocking, horizontal synchronization (line-sync and line-end) and vertical (frame) sync pulses, AC bias (for STN panels), and to enable data capture and LCD panel power control. For software synchronization, interrupts can be generated at specific base-address update opportunities, vertical frame regions, DMA FIFO underflows, and bus errors.

LCD and Peripheral Bus Interfaces provide the CPU with read/write access for setting up or querying the mode-control, timing, and status registers. The parameters for controlling STN and basic TFT panels are accessed through the ASB Bus Slave interface. Once programmed, the ASB Bus Master (LCD-DMA interface) automatically fetches graphics data from the designated frame buffer region of external memory. These activities are synchronized to the LH79532 system clock. After translating this data stream to the desired panel format, panel-timing signals are generated and panel data is streamed through a dedicated 18-bit panel-data bus to cyclically refresh the display. These output signals are synchronized either to the external MCLK input, or to a gated version of SYSCLK controlled by the RPC block. FIFO buffers are provided at the inputs and the outputs of this pipeline for a wide range of clock rates.

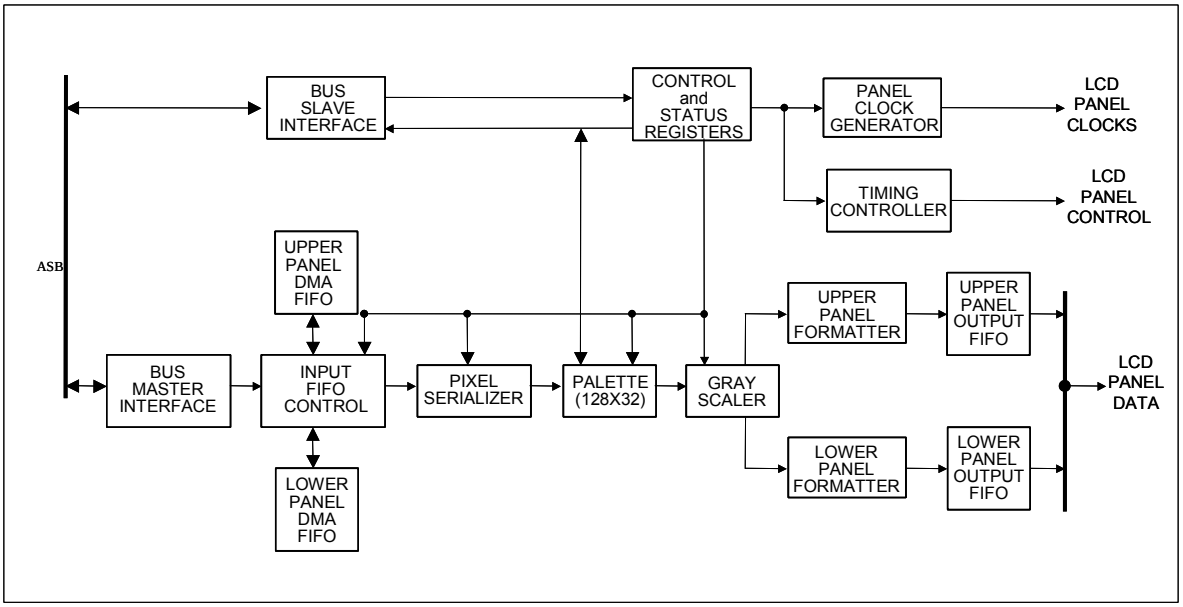


Figure 9. LCD Controller Block Diagram

Register Summary

Following Table summarize the registers available for programming the LCD Controller to interface with any of the supported types of LCD panels. Registers that control the basic functions including STN and TFT formats are all located at sequential offsets from the same base address, and are physically accessed through the ASB bus interface.

Table 18. STN and TFT Panel Format Registers (Base Address: 0xFFFF2000)

Offset from Base	Register	Access	Size	Reset	Description	Bit Position																															
						31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	LCDTiming0	R/W	32	0	Horiz. axis panel control	HBP: Horiz Back Porch								HFP: Horiz Front Porch								HSP: HSync Pulse Width								PPL: Pixels / Line						-	-
0x004	LCDTiming1	R/W	32	0	Vert. axis panel control	VBP: Vert Back Porch								VFP: Vert Front Porch								VSP: VSync Pulse Width						LPP: Lines Per Panel									
0x008	LCDTiming2	R/W	27	0	Clk & Signal polarity control BCD: Bypass pixel clk divider IPC: Invert Panel Clock IVS: Invert Vert. Sync IOE: Invert Output Enable IHS: Invert Horiz. Sync CKS: Ext.CLCDCLKSEL Mux	-	-	-	-	-	BCD	CPL: Clocks/Line										-	IOE	IPC	IHS	IVS	ACB: AC-bias freq					CKS	PCD: Pixel Clk Divisor				
0x00C	LCDTiming3	R/W	17	0	Line end control LEE: LCD Line End Enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LEE	-	-	-	-	-	-	-	-	-	LED: Line-End Delay						
0x010	LCDUPBASE	R/W	32	0	Upper Panel frame base addr.	LCDUPBASE: Upper panel base address																														-	-
0x014	LCDLPBASE	R/W	32	0	Lower Panel frame base addr.	LCDLPBASE: Lower panel base address																														-	-
0x018	LCDINTR-Enable	R/W	5	0	Interrupt Enable mask ME: Master Error Int En VC: Vert. Compare Int En NBU: Next Base Update Int. Enable FUFE: FIFO Underflow Int. Enable	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ME	VC	NBU	FUFE	-
0x01C	LCDControl	R/W	16	0	LCD panel mode control WM: FIFO Water Mark VCI: Vert. Comp Int. trigger event PWR: Enables LCD power BEPO: Big-Endian Pixel Order BEBO: Big-Endian Byte Order BGR: swap RGB colors > BGR DUAL: Dual-Panel LCD MON8: 8-bit monochrome TFT: bypasses grayscaler BW: Monochrome STN-LCD BPP: Bits/Pixel (1,2,4,8,16) EN: Enables LCD Controller	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	WM	-	-	VCI		PWR	BEPO	BEBO	BGR	DUAL	MON8	TFT	BW	BPP			EN
0x020	LCDStatus	R/Clr	5	0	Raw Interrupt Status Flags →Write '1' to clear flag MBE: AMBA bus error VCOM: Vert comp Interrupt LNBU: Next Adr Base Updated FUF: FIFO Underflow	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MBE	VCOM	LNBU	FUF	-
0x024	LCDInterrupt	R	5	0	Final masked Interrupt values MBE: AMBA bus error VCOM: Vert comp Interrupt LNBU: Next Adr Base Updated FUF: FIFO Underflow	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MBE	VCOM	LNBU	FUF	-
0x028	LCDUPCURR	R	32	x	LCD Upper Panel current addr.	LCDUPCURR: Approx Current Upper Panel DMA Address																															
0x02C	LCDLPCURR	R	32	x	LCD Lower Panel current addr.	LCDLPCURR: Approx Current Lower Panel DMA Address																															
0x030-0x1FC		-	-	-	Reserved	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0x200 -0x3FC	LCDPalette	R/W	32	x	256 x 16b color palette I: Optional Intensity adjust	I	Blue					Green					Red					I	Blue					Green					Red				

Pixel Data Storage Format

Pixel data is stored as 32-bit words in the graphics region of memory. Each word is fetched as needed from the memory in ascending sequential order, and stored temporarily in 32-bit FIFO buffers awaiting serialization. Memory page boundaries are managed by the built-in DMA controller. The pixel data can be packed into memory words in one of six available formats based on byte-order, pixel-order, and panel-number.

For dual-panel mode, the input FIFO is configured as two 16-word by 32-bit buffers corresponding to the upper and lower panel halves, into which alternating 32-bit words are placed. Pixel data for alternating panels is word-interleaved into the graphics memory. For single-panel applications, the FIFO buffers are concatenated into a single 32-word deep buffer, relaxing the demands on memory latency. The byte-order determines the significance of the byte-packing sequence: Little-endian Byte order (LB) signifies that the low-order (least-significant) byte has been placed at bit positions [7:0], up to the high-order (most significant) byte placed at bit positions [31:24]. WinCE byte order is also LB. Big-endian Byte order (BB) signifies that the sequence is reversed: the high-order (most-significant) byte has been stored at bit positions [7:0] of the data word, down to the low-order (least-significant) byte stored at bit positions [31:24]. The pixel-order determines the significance of the pixel-packing sequence within each byte. Little-endian Pixel order (LP) signifies that the low-order (least-significant) pixel has been stored at the lowest bit position (the bit-0 end) while the high-order (most-significant) pixel is at the high (bit-7) end of the byte. Big-endian Pixel order (BP) is the reverse: the high-order (most-significant) pixel is stored at the bit-0 end of the byte, while the low-order (least-significant) pixel is stored at the bit-7 end of the byte. Three combinations of word packing sequences are supported, as illustrated in Table 19, Table 20, and Table 21:

Table 19. Little-endian Byte, Little-endian Pixel packing sequence (LBLP)

Bits / Pixel	Bit Position																													
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2	p15		p14		p13		p12		p11		p10		p9		p8		p7		p6		p5		p4		p3		p2		p1		p0	
4	pix-7(3:0)				pix-6(3:0)				pix-5(3:0)				pix-4(3:0)				pix-3(3:0)				pix-2(3:0)				pix-1(3:0)				pix-0(3:0)	
8	pixel-3(7:0)								pixel-2(7:0)								pixel-1(7:0)								pixel-0(7:0)					
16	pixel-1(15:0)																pixel-0(15:0)													

Table 20. Big-endian Byte, Big-endian Pixel packing sequence (BBBP)

Bits / Pixel	Bit Position																													
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	p0		p1		p2		p3		p4		p5		p6		p7		p8		p9		p10		p11		p12		p13		p14		p15	
4	pix-0(3:0)				pix-1(3:0)				pix-2(3:0)				pix-3(3:0)				pix-4(3:0)				pix-5(3:0)				pix-6(3:0)				pix-7(3:0)	
8	pixel-0(7:0)								pixel-1(7:0)								pixel-2(7:0)								pixel-3(7:0)					
16	pixel-0(15:0)																pixel-1(15:0)													

Table 21. Little-endian Byte, Big-endian Pixel packing sequence (LBBP)

Bits / Pixel	Bit Position																													
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	24	25	26	27	28	29	30	31	16	17	18	19	20	21	22	23	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7
2	p12		p13		p14		p15		p8		p9		p10		p11		p4		p5		p6		p7		p0		p1		p2		p3	
4	pix-6(3:0)				pix-7(3:0)				pix-4(3:0)				pix-5(3:0)				pix-2(3:0)				pix-3(3:0)				pix-0(3:0)				pix-1(3:0)	
8	pixel-3(7:0)								pixel-2(7:0)								pixel-1(7:0)								pixel-0(7:0)					
16	pixel-1(15:0)																pixel-0(15:0)													

NOTES: When setting up the LCD Controller, registers must be programmed in the following sequence:

1. Disable the LCD Controller (LCDControl bit EN = ‘0’).
2. Program all other LCD Controller registers.
3. Enable the LCD Controller (LCDControl bit EN = ’1’).

Table 22. LCD Panel Output Signal Multiplexing

Panel Type	TFT		STN	
TFT Bit of LCD Control Register	1		0	
Pin Name	Function	Driver	Function	Driver
CLPOWER	POWER ENABLE	output	POWER ENABLE	output
CLFP	Vsync	output	FRAME	output
CLLP	Hsync	output	LINEsync	output
CLAC	DATA ENABLE	output	ACBiasDrive	output
CLCP	LCDCLK	output	LCDCLK	output
CLLE	LINE END CLLE	output	LINE END CLLE	output

Table 23. LCD Panel Output Data Multiplexing

Ext. Pin	4bit Mono		8bit Mono		8bit Color		TFT
	Single Panel STN	Dual Panel STN	Single Panel STN	Dual Panel STN	Single Panel STN	Dual Panel STN	
VD17	-	-	-	-	-	-	BLUE[4]
VD16	-	-	-	-	-	-	BLUE[3]
VD15	-	-	-	MLSTN[0]	-	CLSTN[0]	BLUE[2]
VD14	-	-	-	MLSTN[1]	-	CLSTN[1]	BLUE[1]
VD13	-	-	-	MLSTN[2]	-	CLSTN[2]	BLUE[0]
VD12	-	-	-	MLSTN[3]	-	CLSTN[3]	Intensity bit
VD11	-	MLSTN[0]	-	MLSTN[4]	-	CLSTN[4]	GREEN[4]
VD10	-	MLSTN[1]	-	MLSTN[5]	-	CLSTN[5]	GREEN[3]
VD9	-	MLSTN[2]	-	MLSTN[6]	-	CLSTN[6]	GREEN[2]
VD8	-	MLSTN[3]	-	MLSTN[7]	-	CLSTN[7]	GREEN[1]
VD7	-	-	MUSTN[0]	MUSTN[0]	CUSTN[0]	CUSTN[0]	GREEN[0]
VD6	-	-	MUSTN[1]	MUSTN[1]	CUSTN[1]	CUSTN[1]	Intensity bit
VD5	-	-	MUSTN[2]	MUSTN[2]	CUSTN[2]	CUSTN[2]	RED[4]
VD4	-	-	MUSTN[3]	MUSTN[3]	CUSTN[3]	CUSTN[3]	RED[3]
VD3	MUSTN[0]	MUSTN[0]	MUSTN[4]	MUSTN[4]	CUSTN[4]	CUSTN[4]	RED[2]
VD2	MUSTN[1]	MUSTN[1]	MUSTN[5]	MUSTN[5]	CUSTN[5]	CUSTN[5]	RED[1]
VD1	MUSTN[2]	MUSTN[2]	MUSTN[6]	MUSTN[6]	CUSTN[6]	CUSTN[6]	RED[0]
VD0	MUSTN[3]	MUSTN[3]	MUSTN[7]	MUSTN[7]	CUSTN[7]	CUSTN[7]	Intensity bit

NOTE: - : Reserved.

Synchronous Serial Interface (Compatible with SPI)

The Synchronous Serial Port controller provides a serial communications interface that is compatible with Motorola Serial Peripheral Interface (SPI) protocols. This allows the LH79532 to communicate with other compatible external devices. The controller provides both Master and Slave capabilities. See the ARM PrimeCell™ Synchronous Serial Port Master and Slave document for a full General Description and Block Diagram.

Features

- Master and Slave Operation
- Compatible with Motorola Serial Peripheral Interface
- Programmable clock bit rate and prescaler
- Separate transmit and receive FIFOs
- 16 bits wide
- 8 locations deep
- Support LSB and MSB first
- Programmable data frame from 4 to 16 bits
- Independent masking of transmit FIFO, receive FIFO and receive overrun interrupts
- Internal loopback test mode available

Register Summary

The SPI controller is located at a base address of 0xFFFF5400.

Table 24. SPI Registers (Base Address: 0xFFFF5400)

Offset from Base	Register	Name	Access	Size	Reset Value
0x00	SSPCR0	SSP Control Register 0	R/W	16	0x0000
0x04	SSPCR1	SSP Control Register 1	R/W	8	0x0000
0x08	SSPDR	SSP Data Register	R/W	16	Undefined
0x0C	SSPSR	SSP Status Register	R	5	0x00
0x10	SSPCPSR	SSP Clock Prescale Register	R/W	8	0x00
0x14	SSPIIR	SSP Interrupt Identification Register	R	3	0x00
0x14	SSPICR	SSP Interrupt Clear Register	W	16	Undefined

Universal Asynchronous Receiver / Transmitter (UART)

Features

The UARTs in the LH79532 are similar in function to a standard 16C550 device. These Universal Asynchronous Receiver/Transmitters support bit rates of up to 115.2 kbps and contain two 16-byte FIFOs for receive and transmit. MODEM control input signals, RI, RTS, DCD, DSR, DSR, DTR and CTS, are supported. The UART operation and bit rate values are controlled by the bit rate and line control register (UBRLCR).
Each UART can generate four interrupts: 'Receive' is asserted when the receive FIFO becomes half full (eight bytes in the FIFO are filled) or if the receive FIFO is non-empty for longer than three-character-length-time with no more characters received. 'Transmit' is asserted if the transmit FIFO buffer reaches half empty. 'MODEM' status is asserted if any of the MODEM status bits changes. 'UART_disabled' is asserted when a start bit is detected on the receive line and the UART is disabled. If a framing, overrun or parity error occurs during reception, the appropriate error bit is set, and is stored in the FIFO. The FIFOs can also be programmed to be only one byte deep, like a conventional UART with double buffering.

The LH79532 has two UARTs that can be individually addressed. The registers are shown in Table 25.

Register Summary

Table 25. UART Registers (Base Address: 0xFFFF4000)

Offset from Base	Register	Name	Access	Size	Reset Value
0x000	UART0DR	UART0 Data Register	R/W	8	Undefined
0x004	UART0RXSTAT	UART0 Receive Status Register	R	3	Undefined
0x004	UART0MSEOI	UART0 Clear MODEM status change interrupt	W	3	Undefined
0x008	H_U0BRLCR	UART0 High byte Baud Rate/Line Control Register	R/W	7	0x00
0x00C	M_U0BRLCR	UART0 Middle byte Baud Rate/Line Control Register	R/W	8	0x00
0x010	L_U0BRLCR	UART0 Lower byte Baud Rate/Line Control Register	R/W	8	0x00
0x014	UART0CON	UART0 Control Register	R/W	8	0b0xxx0000
0x018	UART0FLG	UART0 Flag Register	R/W	7	0b010xxx
0x01C	UART0MCR	UART0 MODEM Control Register	R/W	2	0x00
0x020	UART0INTMSK	UART0 Interrupt Mask Register	R/W	5	0x0
0x024	UART0STAT	UART0 Interrupt Raw Status Register	R	5	Undefined
0x028	UART0INTR	UART0 Interrupt Status Register	R	5	Undefined
0x400	UART1DR	UART1 Data Register	R/W	8	Undefined
0x404	UART1RXSTAT	UART1 Receive Status Register	R	3	Undefined
0x404	UART1UMSEOI	UART1 Clear MODEM status change interrupt	W	3	Undefined
0x408	H_U1BRLCR	UART1 High byte Baud Rate/Line Control Register	R/W	7	0x00
0x40C	M_U1BRLCR	UART1 Middle byte Baud Rate/Line Control Register	R/W	8	0x00
0x410	L_U1BRLCR	UART1 Lower byte Baud Rate/Line Control Register	R/W	8	0x00
0x414	UART1CON	UART1 Control Register	R/W	8	0b0xxx0000
0x418	UART1FLG	UART1 Flag Register	R/W	7	0b010xxx
0x41C	UART1MCR	UART1 MODEM Control Register	R/W	2	0x00
0x420	UART1INTMSK	UART1 Interrupt Mask Register	R/W	5	0x0
0x424	UART1STAT	UART1 Interrupt Raw Status Register	R	5	Undefined
0x428	UART1INTR	UART1 Interrupt Status Register	R	5	Undefined

NOTE: Because there are 2 UARTs to be addressed, Register abbreviations take the form of:
UART(n) register where n = 0 or 1.
'x' = undefined.

Pulse Width Modulators (PWM)

Features

- Four channels
- Synchronous or Asynchronous operation
- Start PWM with on-chip Counter/Timer0 or external input
- Sleep Mode to save power
- Programmable Pulse Width (Duty Cycle), Interval (Frequency), and Polarity
- Frequency range DC to 18.75 MHz
- Up to 16-bit Resolution
- Double Buffered to allow dynamic programming while PWM is running
- Stops or updates Duty Cycle, Frequency, and Polarity at end of a PWM Cycle
- Smooth output: no glitches or errors when updated statically or dynamically
- Allows chaining of PWMs:
- PWM0→PWM1→PWM2→PWM3
- PWM0→PWM1→PWM2
- PWM0→PWM1

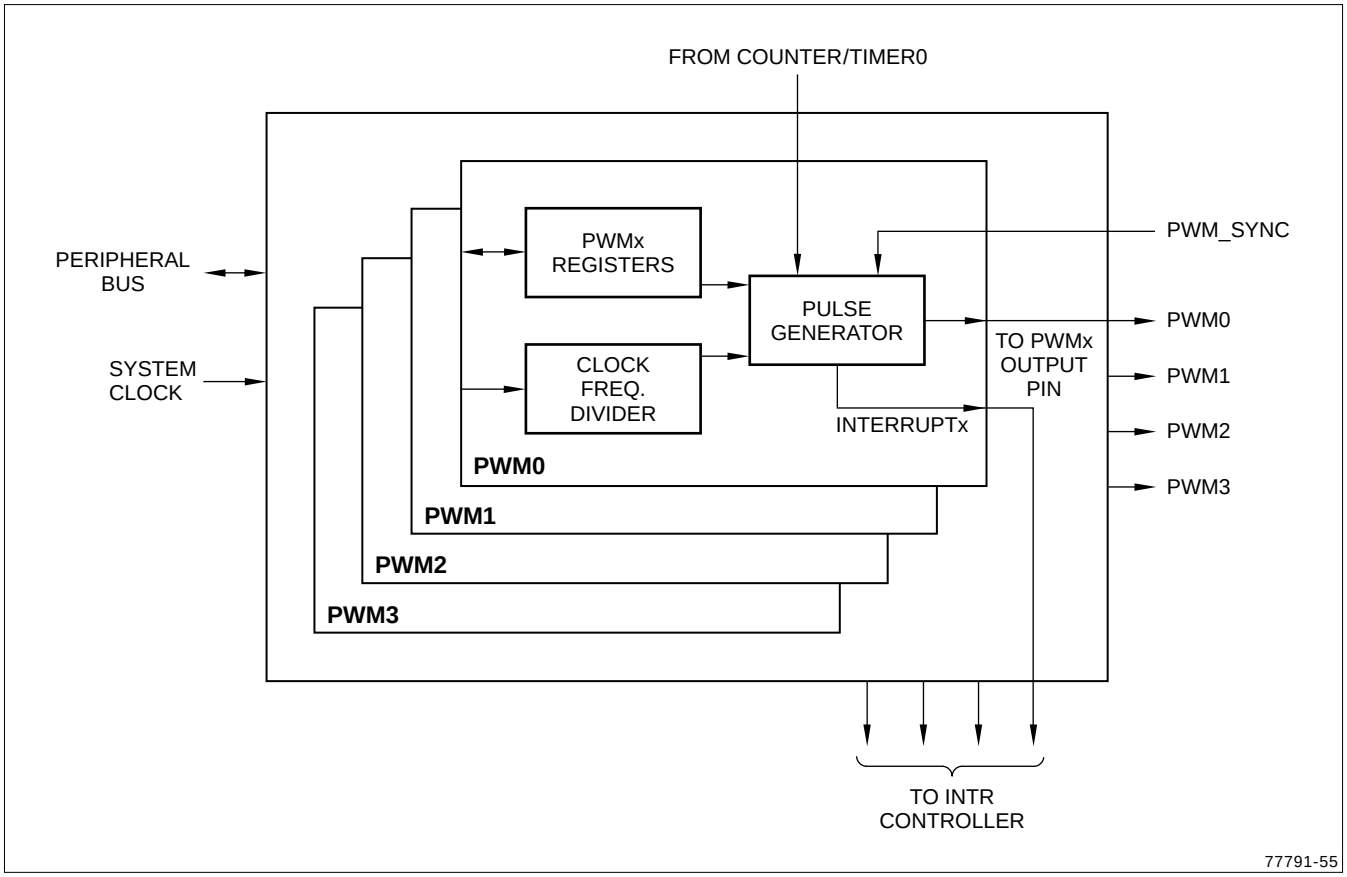


Figure 10. PWM Block Diagram

Overview

Pulse Width Modulation is a method of communicating information to an external device by means of a constant-amplitude square-wave whose period and duty cycle are programmable. Figure 11 shows an example of a PWM output signal.

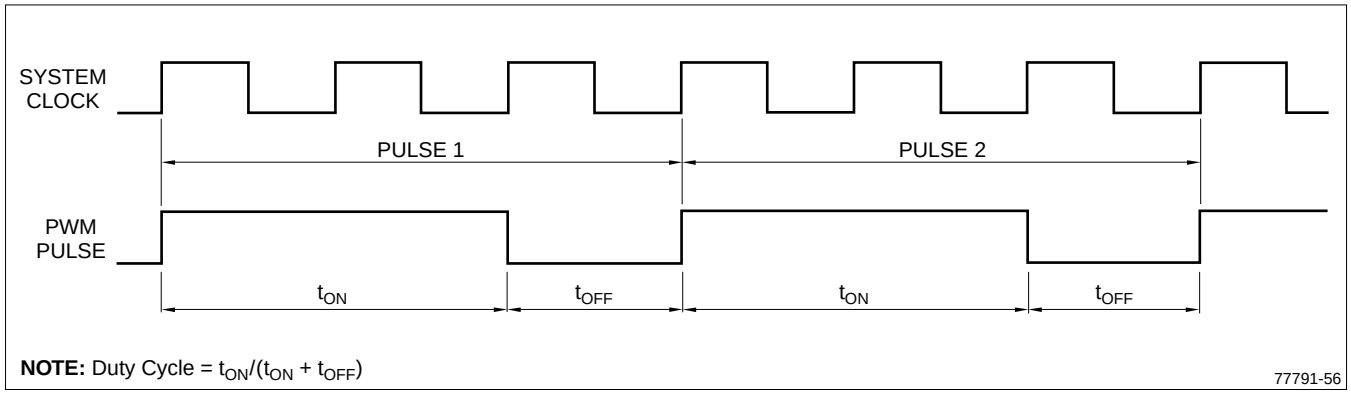


Figure 11. PWM Duty Cycles

The duty cycle is expressed as the duration of t_{ON} over the sum of t_{ON} and t_{OFF}. If t_{ON} is equal to t_{OFF}, the signal has a 50% duty cycle. The input clock PWM_in_CLK is a gated version of the system clock (controlled by the Reset and Power Controller) divided by a programmable factor from 1 to 255.

Each PWM channel also generates a positive edge triggered interrupt to the interrupt controller. The LOW to HIGH transition of the interrupt output of each channel signals the end of cycle of that channel. The width of the interrupt signal depends on the settings of the TC and DIV.

All the four channels of Pulse Width Modulator (PWM) are identical, except that the selection between Counter/Timer0 and external input to start the PWM during synchronous mode can only be set through the register of PWM0. PWMx (where x = 0, 1, 2, 3) has 16-bit resolution. Each PWM can function in either normal mode or synchronous mode.

During reset, all the PWMs will enter halt mode and output 0s.

Normal Mode

A PWMx channel operates in normal mode when its PWMx_SYNC register is reset to ‘0’. Each PWM channel can be independently enabled and started by writing a ‘1’ to register PWMx_ENB. All PWM channels are set to this mode upon reset. A PWM can be stopped by writing a ‘0’ to PWMx_ENB, which will then stop the PWM at the end of its current cycle.

Synchronization Mode

A PWM channel operates in synchronous mode when both its PWMx_ENB and PWMx_SYNC register bits are set to ‘1’. In this mode, the PWM will be started by a rising edge on the PWM_SYNC input.

There are two signal sources for the synchronization:

- Counter/Timer0 output
- External signal input PWM_SYNC

Register Summary

The base address for PWM registers is 0xFFFF5000.

Table 26. PWM Registers (Base Address: 0xFFFF5000)

Offset from Base	Register	Name	Access	Size	Reset Value
0x000	PWM0_TC	PWM0 Terminal Count Register	R/W	16	0x00
0x004	PWM0_DC	PWM0 Duty Cycle Register	R/W	16	0x00
0x008	PWM0_ENB	PWM0 Enable Register	R/W	1	0x00
0x00c	PWM0_DIV	PWM0 Clock Divider Register	R/W	8	0x01
0x010	PWM0_SYNC	PWM0 Synchronous Register	R/W	2	0x00
0x014	PWM0_INV	PWM0 Invert Register	R/W	1	0x00
0x100	PWM1_TC	PWM1 Terminal Count Register	R/W	16	0x00
0x104	PWM1_DC	PWM1 Duty Cycle Register	R/W	16	0x00
0x108	PWM1_ENB	PWM1 Enable Register	R/W	1	0x00
0x10c	PWM1_DIV	PWM1 Clock Divider Register	R/W	8	0x01
0x110	PWM1_SYNC	PWM1 Synchronous Register	R/W	1	0x00
0x114	PWM1_INV	PWM1 Invert Register	R/W	1	0x00
0x200	PWM2_TC	PWM2 Terminal Count Register	R/W	16	0x00
0x204	PWM2_DC	PWM2 Duty Cycle Register	R/W	16	0x00
0x208	PWM2_ENB	PWM2 Enable Register	R/W	1	0x00
0x20c	PWM2_DIV	PWM2 Clock Divider Register	R/W	8	0x01
0x210	PWM2_SYNC	PWM2 Synchronous Register	R/W	1	0x00
0x214	PWM2_INV	PWM2 Invert Register	R/W	1	0x00
0x300	PWM3_TC	PWM3 Terminal Count Register	R/W	16	0x00
0x304	PWM3_DC	PWM3 Duty Cycle Register	R/W	16	0x00
0x308	PWM3_ENB	PWM3 Enable Register	R/W	1	0x00
0x30c	PWM3_DIV	PWM3 Clock Divider Register	R/W	8	0x01
0x310	PWM3_SYNC	PWM3 Synchronous Register	R/W	1	0x00
0x314	PWM3_INV	PWM3 Invert Register	R/W	1	0x00

Counter/Timer

Four general purpose Counter/Timers (CT) are included in the LH79532. Each Counter/Timer is 16 bits wide and can be used as a periodic timer, frequency generator, etc. Of the four, only CT3 accepts an external clock.

Counter/Timer Operation

Each Counter/Timer is 16 bits wide with a selectable pre-scale of 2, 6, or 10 bits. The system clock is used to clock the Counter/Timers, and is first divided by 4, 64, or 1024, according to the pre-scale selection.

The counters are loaded by writing to the Load register after which, if enabled, the Counter/Timer will count down to zero. On reaching a count of zero, an interrupt will be generated. The interrupt may be cleared by writing to the Clear register.

After reaching a zero count, if the Counter/Timer is operating in free-running mode, the counter will continue to decrement from its maximum value (0xFFFF). If periodic timer mode is selected, the Counter/Timer will reload from the Load register and continue to decrement. In this mode, the Counter/Timer will effectively generate a periodic interrupt. A bit in the control register selects the mode.

It is possible to cascade the counters and use them as a 32-, 48-, or 64-bit counter. This is controlled by the Carry bit in the control register. The 32 bit counter can be made from CT0 and CT1, CT1 and CT2, or CT2 and CT3. Cascading the 3 counters and 4 counters makes the 48 bit counter and 64 bit counter respectively. The cascaded counters must have the same configuration.

Each counter has an output signal. When a counter load occurs, the output value is high. If the Counter/Timer is operating in free-running mode, the output value goes LOW when the counter reaches zero. If operating in periodic timer mode the output value is toggled when the counter reaches zero.

At any time, the current Counter/Timer value may be read from the Value register.

A bit in the Control register enables the Counter/Timer.

At reset, the counter will be disabled, the interrupt will be cleared and the Load register will be undefined. The mode and pre-scale value will also be undefined.

Register Summary

Table 27. Counter / Timer Registers (Base Address: 0xFFFF5800)

Offset from Base	Register	Name	Access	Size	Reset Value
0x00	CT0Load	Counter/Timer 0 Load Register	R/W	16	Undefined
0x04	CT0Value	Counter/Timer 0 Value Register	R	16	Undefined
0x08	CT0Control	Counter/Timer 0 Control Register	R/W	8	0x00
0x0C	CT0Clear	Counter/Timer 0 Clear Register	W	0	Undefined
0x20	CT1Load	Counter/Timer 1 Load Register	R/W	16	Undefined
0x24	CT1Value	Counter/Timer 1 Value Register	R	16	Undefined
0x28	CT1Control	Counter/Timer 1 Control Register	R/W	8	0x00
0x2C	CT1Clear	Counter/Timer 1 Clear Register	W	0	Undefined
0x40	CT2Load	Counter/Timer 2 Load Register	R/W	16	Undefined
0x44	CT2Value	Counter/Timer 2 Value Register	R	16	Undefined
0x48	CT2Control	Counter/Timer 2 Control Register	R/W	8	0x00
0x4C	CT2Clear	Counter/Timer 2 Clear Register	W	0	Undefined
0x60	CT3Load	Counter/Timer 3 Load Register	R/W	16	Undefined
0x64	CT3Value	Counter/Timer 3 Value Register	R	16	Undefined
0x68	CT3Control	Counter/Timer 3 Control Register	R/W	8	0x00
0x6C	CT3Clear	Counter/Timer 3 Clear Register	W	0	Undefined

Watchdog Timer (WDT)

The Watchdog Timer is a hardware protection against malfunctions. It is a programmable timer to be reset by software at regular intervals. Failure to do so will cause the LH79532 to interrupt or reset.

Features

- Driven by System Clock
- Programmable Timeout Periods: 2^{21}, 2^{22}, 2^{23}, 2^{24}, 2^{25}, 2^{26}, 2^{29}, or 2^{31} clock cycles
- Generates a System Reset (resets LH79532) or a FIQ Interrupt whenever a timeout period is reached
- Software enable, lockout, and counter-reset mechanisms add security against inadvertent writes
- Protection mechanism guards against interrupt-service failure:
  - The first WDT timeout triggers FIQ and asserts nWDFIQ status flag
  - If FIQ service routine fails to clear nWDFIQ, then next WDT timeout triggers a system reset

Overview

The Watchdog Timer consists of a 32-bit counter that is used to cause a selectable time-out interval to detect malfunctions**.** The timer needs to be periodically reset by software**.** Failure to do so will result in a time-out that will cause either an interrupt to be taken, an external reset to occur, or a system reset to be issued**.** The Watchdog Timer is controlled through the control register WDCTLR**.** There are eight selectable time intervals for a timeout: 2^{21}, 2^{22}, 2^{23}, 2^{24}, 2^{25}, 2^{26}, 2^{29}, or 2^{31} clock cycles for time-out periods ranging from 41.94 ms to 41.95seconds with a system clock of 50 MHz.

When the reset option is secected, the time-out causes a system reset immediately. When the interrupt option is selected, the first time-out will cause an FIQ interrupt. After first time out, the WDT counter must clear. If the second time out is detected before clearing the WDT counter, a system reset is occurred,

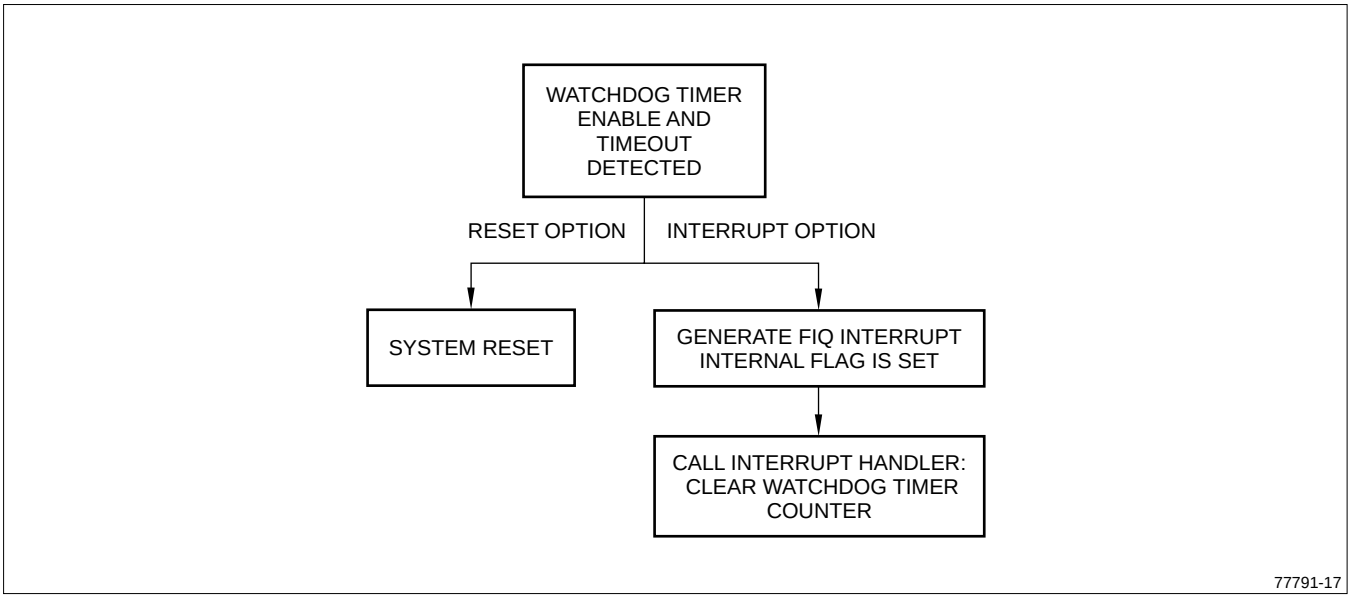


Figure 12. Block Diagram of Watchdog Timer

NOTE: If a second timeout has occurred without clearing the first timeout, a system reset is forced.

Register Summary

Table 28. WDT Registers (Base Address: 0xFFFF8000)

Offset from Base	Register	Access	Reset	Size	Description	Bit Position																											
						31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	WDCTLR	R/W	0x00	8	Watchdog Control Register EN: Enable Watchdog RSP: Timeout Response 00=FIQ,11=SysReset FRZ: Freeze EN bit (set-only) TOP: Time-Out Period 0x0 = 2^{21}, … 0x5 = 2^{26}, 0x6 = 2^{29},0x7 = 2^{31} clock tics																							-	TOP				FRZ	RSP		EN
0x04	WDCNTR	W	0xA5A5	16	Watchdog Counter Reset Reset WDT by writing 0xA5A5	Reserved																1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1

Programmable I/O (PIO)

The Programmable I/O (PIO) is a 32 bit general-purpose input/output port controller. Each bit can be set as an input or output. The PIO port has multiple functions (either as a general purpose I/O or another macro's function pin).

Features

- Four PIO ports (PIOA, PIOB, PIOC and PIOD)
- PIOA (25-bit)
- PIOB (24-bit)
- PIOC (15-bit)
- PIOD (16-bit)
- PIO control registers allow multiplexing of pins for different functions.

Register Summary

A summary of the PIO registers is shown in Table 29.

Table 29. PIO Register (Base Address: 0xFFFF4C00)

Offset from Base	Register	Name	Access	Size	Reset Value
0x00	PIOAData	PA Data Register	R/W	25	0
0x04	PIOACtrl	PA Control Register	R/W	25	0
0x08	PIOADataSet	PA Data Set Register	W	25	0
0x0C	PIOADataReset	PA Data Reset Register	W	25	0
0x10	PIOAMuxControl1	PA Mux Control1 Register	R/W	25	0
0x14	PIOAMuxControl2	PA Mux Control2 Register	R/W	25	0
0x20	PIOBData	PB Data Register	R/W	24	0
0x24	PIOBCtrl	PB Control Register	R/W	24	0
0x28	PIOBDataSet	PB Data Set Register	W	24	0
0x2C	PIOBDataReset	PB Data Reset Register	W	24	0
0x30	PIOBMuxControl1	PB Mux Control1 Register	R/W	24	0
0x34	PIOBMuxControl2	PB Mux Control2 Register	R/W	24	0
0x40	PIOCData	PC Data Register	R/W	15	0
0x44	PIOCCtrl	PC Control Register	R/W	15	0
0x48	PIOCDataSet	PC Data Set Register	W	15	0
0x4C	PIOCDataReset	PC Data Reset Register	W	15	0
0x50	PIOCMuxControl1	PC Mux Control1 Register	R/W	15	0
0x54	PIOCMuxControl2	PC Mux Control2 Register	R/W	15	0
0x60	PIODData	PD Data Register	R/W	16	0
0x64	PIODCtrl	PD Control Register	R/W	16	0
0x68	PIODDataSet	PD Data Set Register	W	16	0
0x6C	PIODDataReset	PD Data Reset Register	W	16	0
0x70	PIODMuxControl	PD Mux Control Register	R/W	16	0

Multiplexed I/O

There are 4 ‘banks’ of PIO, or Programmable I/O. These are labeled PIOA, PIOB, PIOC and PIOD. The map for the multiplexed I/O is shown below. Note that the MUX column refer to the PIOxMuxControl[1:2]Reg of the appropriate bank of registers.

Multiplexed Functions												
Bit Index	**Pin**	**Func1**	**Macro**	**Dir**	**Func2**	**Macro**	**Dir**	**MUX1**	**Func3**	**Macro**	**Dir**	**MUX2**
PIOA												
0	136	PIO[0]	PIO	B	EXTLCDCLK	LCDC	I	0	-	-	-	-
1	135	PIO[1]	PIO	B	CLLE	LCDC	O	1	-	-	-	-
2	133	PIO[2]	PIO	B	CLCP	LCDC	O	2	-	-	-	-
3	132	PIO[3]	PIO	B	CLAC	LCDC	O	3	-	-	-	-
4	131	PIO[4]	PIO	B	CLLP	LCDC	O	4	-	-	-	-
5	129	PIO[5]	PIO	B	CLFP	LCDC	O	5	-	-	-	-
6	128	PIO[6]	PIO	B	CLPOWER	LCDC	O	6	-	-	-	-
7	127	PIO[7]	PIO	B	VD[0]	LCDC	O	7	-	-	-	-
8	126	PIO[8]	PIO	B	VD[1]	LCDC	O	8	-	-	-	-
9	125	PIO[9]	PIO	B	VD[2]	LCDC	O	9	-	-	-	-
10	123	PIO[10]	PIO	B	VD[3]	LCDC	O	10	-	-	-	-
11	122	PIO[11]	PIO	B	VD[4]	LCDC	O	11	-	-	-	-
12	121	PIO[12]	PIO	B	VD[5]	LCDC	O	12	-	-	-	-
13	120	PIO[13]	PIO	B	VD[6]	LCDC	O	13	-	-	-	-
14	119	PIO[14]	PIO	B	VD[7]	LCDC	O	14	-	-	-	-
15	118	PIO[15]	PIO	B	VD[8]	LCDC	O	15	INTR4	INTC	I	15
16	117	PIO[16]	PIO	B	VD[9]	LCDC	O	16	INTR5	INTC	I	16
17	115	PIO[17]	PIO	B	VD[10]	LCDC	O	17	INTR6	INTC	I	17
18	114	PIO[18]	PIO	B	VD[11]	LCDC	O	18	INTR7	INTC	I	18
19	113	PIO[19]	PIO	B	VD[12]	LCDC	O	19	nUDSR0	UART0	I	19
20	112	PIO[20]	PIO	B	VD[13]	LCDC	O	20	nUDTR0	UART0	O	20
21	111	PIO[21]	PIO	B	VD[14]	LCDC	O	21	nUDCD0	UART0	I	21
22	110	PIO[22]	PIO	B	VD[15]	LCDC	O	22	nURI0	UART0	I	22
23	109	PIO[23]	PIO	B	VD[16]	LCDC	O	23	nUCTS0	UART0	I	23
24	108	PIO[24]	PIO	B	VD[17]	LCDC	O	24	nURTS0	UART0	O	24
PIOB												
0	106	PIO[25]	PIO	B	SPI_CLK	SPI	B	0	-	-	-	-
1	105	PIO[26]	PIO	B	SPI_ENB	SPI	B	1	-	-	-	-
2	104	PIO[27]	PIO	B	SPI_OUT	SPI	O	2	-	-	-	-
3	103	PIO[28]	PIO	B	SPI_IN	SPI	I	3	-	-	-	-
4	101	PIO[29]	PIO	B	INTR0	INTC	I	4	-	-	-	-
5	100	PIO[30]	PIO	B	INTR1	INTC	I	5	ALM	RTC	O	5
6	99	PIO[31]	PIO	B	INTR2	INTC	I	6	UCLK	UART	I	6
7	98	PIO[32]	PIO	B	INTR3	INTC	I	7	CTCLK	CT	I	7
8	81	PIO[33]	PIO	B	PWM0	PWM	O	8	CTOUT2	CT	O	8
9	80	PIO[34]	PIO	B	PWM1	PWM	O	9	CTOUT3	CT	O	9
10	79	PIO[35]	PIO	B	DEOT0	DMA	O	10	-	-	-	-
11	78	PIO[36]	PIO	B	nDACK[0]	DMA	O	11	-	-	-	-
12	77	PIO[37]	PIO	B	DREQ0	DMA	I	12	-	-	-	-
13	75	PIO[38]	PIO	B	DEOT1	DMA	O	13	PWM2	PWM	O	13
14	74	PIO[39]	PIO	B	nDACK[1]	DMA	O	14	PWM3	PWM	O	14
15	73	PIO[40]	PIO	B	DREQ1	DMA	I	15	PWM_SYNC	PWM	I	15

Multiplexed Functions												
Bit Index	**Pin**	**Func1**	**Macro**	**Dir**	**Func2**	**Macro**	**Dir**	**MUX1**	**Func3**	**Macro**	**Dir**	**MUX2**
16	72	PIO[41]	PIO	B	nWAIT	EBI	I	16	-	-	-	-
17	71	PIO[42]	PIO	B	CLKOUT	RPC	O	17	-	-	-	-
18	70	PIO[43]	PIO	B	XCLKEN	RPC	O	18	-	-	-	-
19	95	nRESETO	RPC	O	PIO[44]	PIO	B	19	-	-	-	-
20	87	URXD1	UART1	I	PIO[45]	PIO	B	20	CTOUT0	CT	O	20
21	86	UTXD1	UART1	O	PIO[46]	PIO	B	21	CTOUT1	CT	O	21
22	84	URXD0	UART0	I	PIO[47]	PIO	B	22	-	-	-	-
23	83	UTXD0	UART0	O	PIO[48]	PIO	B	23	-	-	-	-
PIOC												
0	68	DQM[0]	SDRAMC	O	PIO[49]	PIO	B	0	-	-	-	-
1	67	DQM[1]	SDRAMC	O	PIO[50]	PIO	B	1	-	-	-	-
2	66	DQM[2]	SDRAMC	O	PIO[51]	PIO	B	2	-	-	-	-
3	65	DQM[3]	SDRAMC	O	PIO[52]	PIO	B	3	-	-	-	-
4	56	nCE[4]	EBI	O	PIO[53]	PIO	B	4	-	-	-	-
5	55	nCE[5]	EBI	O	PIO[54]	PIO	B	5	-	-	-	-
6	54	nDCS[0]	SDRAMC	O	PIO[55]	PIO	B	6	nCE[6]	EBI	O	6
7	53	nDCS[1]	SDRAMC	O	PIO[56]	PIO	B	7	nCE[7]	EBI	O	7
8	51	nWE[0]	EBI	O	PIO[57]	PIO	B	8	-	-	-	-
9	50	nWE[1]	EBI	O	PIO[58]	PIO	B	9	-	-	-	-
10	43	nWE[2]	EBI	O	PIO[59]	PIO	B	10	-	-	-	-
11	42	nWE[3]	EBI	O	PIO[60]	PIO	B	11	-	-	-	-
12	15	RA[20]	EBI	O	PIO[61]	PIO	B	12	-	-	-	-
13	14	RA[21]	EBI	O	PIO[62]	PIO	B	13	-	-	-	-
14	13	RA[22]	EBI	O	PIO[63]	PIO	B	14	-	-	-	-
PIOD, BIGEND = '0'												
0	156	RD[16]	EBI	O	PIO[64]	PIO	B		-	-	-	-
1	155	RD[17]	EBI	O	PIO[65]	PIO	B		-	-	-	-
2	154	RD[18]	EBI	O	PIO[66]	PIO	B		-	-	-	-
3	153	RD[19]	EBI	O	PIO[67]	PIO	B		-	-	-	-
4	151	RD[20]	EBI	O	PIO[68]	PIO	B		-	-	-	-
5	150	RD[21]	EBI	O	PIO[69]	PIO	B		-	-	-	-
6	149	RD[22]	EBI	O	PIO[70]	PIO	B		-	-	-	-
7	148	RD[23]	EBI	O	PIO[71]	PIO	B		-	-	-	-
8	147	RD[24]	EBI	O	PIO[72]	PIO	B		-	-	-	-
9	145	RD[25]	EBI	O	PIO[73]	PIO	B		-	-	-	-
10	144	RD[26]	EBI	O	PIO[74]	PIO	B		-	-	-	-
11	143	RD[27]	EBI	O	PIO[75]	PIO	B		-	-	-	-
12	142	RD[28]	EBI	O	PIO[76]	PIO	B		-	-	-	-
13	141	RD[29]	EBI	O	PIO[77]	PIO	B		-	-	-	-
14	139	RD[30]	EBI	O	PIO[78]	PIO	B		-	-	-	-
15	138	RD[31]	EBI	O	PIO[79]	PIO	B		-	-	-	-
PIOD, BIGEND = '1'												
0	175	RD[0]	EBI	O	PIO[64]	PIO	B		-	-	-	-
1	174	RD[1]	EBI	O	PIO[65]	PIO	B		-	-	-	-
2	173	RD[2]	EBI	O	PIO[66]	PIO	B		-	-	-	-
3	172	RD[3]	EBI	O	PIO[67]	PIO	B		-	-	-	-
4	171	RD[4]	EBI	O	PIO[68]	PIO	B		-	-	-	-
5	169	RD[5]	EBI	O	PIO[69]	PIO	B		-	-	-	-

Multiplexed Functions												
Bit Index	**Pin**	**Func1**	**Macro**	**Dir**	**Func2**	**Macro**	**Dir**	**MUX1**	**Func3**	**Macro**	**Dir**	**MUX2**
6	168	RD[6]	EBI	O	PIO[70]	PIO	B		-	-	-	-
7	167	RD[7]	EBI	O	PIO[71]	PIO	B		-	-	-	-
8	166	RD[8]	EBI	O	PIO[72]	PIO	B		-	-	-	-
9	165	RD[9]	EBI	O	PIO[73]	PIO	B		-	-	-	-
10	163	RD[10]	EBI	O	PIO[74]	PIO	B		-	-	-	-
11	162	RD[11]	EBI	O	PIO[75]	PIO	B		-	-	-	-
12	161	RD[12]	EBI	O	PIO[76]	PIO	B		-	-	-	-
13	160	RD[13]	EBI	O	PIO[77]	PIO	B		-	-	-	-
14	159	RD[14]	EBI	O	PIO[78]	PIO	B		-	-	-	-
15	157	RD[15]	EBI	O	PIO[79]	PIO	B		-	-	-	-

Interrupt Controller (INTC)

The interrupt controller receives interrupt requests from sources external and internal to the chip. It has two outputs to the CPU. These are nFIQ and nIRQ. External interrupt requests are programmable as level sensitive or edge triggered and active-HIGH or active-LOW. Interrupt requests are selectable to be either an IRQ or FIQ, but the WDT (Watchdog Timer) interrupt is only an FIQ interrupt.

Features

- 28 interrupt sources
  - 8 external interrupts
  - 20 internal interrupts
- Low interrupt latency
- Enable/Disable for each interrupt source
- Active HIGH or LOW, EDGE/LEVEL sensitive (external interrupts only)
- Drive nIRQ or nFIQ

Register Summary

The base address for interrupt controller registers is 0xFFFF7400. The address, initial value, access, and number of bits for each register are given in Table 30.

Table 30. INTC programming Registers (Base Address: 0xFFFF7400)

Offset from Base	Register	Name	Access	Reset Value
0x000	IRQStatus	IRQ Status Register	R	0x0000000
0x004	IRQRawStatus	IRQ Raw Status Register	R	0x0000000
0x008	IRQEnable	IRQ Enable Register	R	0x0000000
0x008	IRQEnableSet	IRQ Enable Set Register	W	Unknown
0x00C	IRQEnableClear	IRQ Enable Clear Register	W	Unknown
0x100	FIQStatus	FIQ Status Register	R	0x0000000
0x104	FIQRawStatus	FIQ Raw Status Register	R	0x0000000
0x108	FIQEnable	FIQ Enable Register	R	0x0000000
0x108	FIQEnableSet	FIQ Enable Set Register	W	Unknown
0x10C	FIQEnableClear	FIQ Enable Clear Register	W	Unknown
0x200	INTConfig0	Configuration Register 0	R/W	0x0000000
0x204	INTConfig1	Configuration Register 1	R/W	0x0000000
0x208	INTClear	Interrupt Clear Register	W	Unknown

Register	Bit Position																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIQStatus, FIQRawStatus FIQEnable, FIQEnableSet, FIQEnableClear	-				CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	-	CH18	-	CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
IRQStatus, IRQRawStatus FIQEnable, FIQEnableSet FIQEnableClear																							-									
INTConfig0 INTConfig1													-																			
INTClear													-				CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	-							

NOTE: -: Reserved

Interrupt Channel Assignments

Interrupt channels are assigned as in Table 31: Interrupt Channel Assignment. The WDT (Watchdog Timer) can drive nFIQ only and cannot be disabled in this controller (If you want to disable the WDT interrupt, set the WDT control register in the WDT).

Table 31. Interrupt Channel Assignment

CH	Interrupt Source	Drive IRQ/FIQ	Interrupt Type (Edge/Level; H/L)
0	Counter/Timer0	Programmable	HIGH Level
1	Counter/Timer1	Programmable	HIGH Level
2	Counter/Timer2	Programmable	HIGH Level
3	Counter/Timer3	Programmable	HIGH Level
4	UART ch0	Programmable	HIGH Level
5	UART ch1	Programmable	HIGH Level
6	DMA ch0	Programmable	HIGH Level
7	DMA ch1	Programmable	HIGH Level
8	PLL Lock Lost	Programmable	Rising Edge
9	WDT	FIQ only	Rising Edge
10	RTC_ALARM	Programmable	Rising Edge
11	RTC_IRQF	Programmable	Rising Edge
12	PWM ch0	Programmable	Rising Edge
13	PWM ch1	Programmable	Rising Edge
14	PWM ch2	Programmable	Rising Edge
15	PWM ch3	Programmable	Rising Edge
16	SPI	Programmable	HIGH Level
17	Reserved	-	-
18	LCDC	Programmable	HIGH Level
19	Reserved	-	-
20	External Interrupt0 (INT0)	Programmable	Programmable
21	External Interrupt1 (INT1)	Programmable	Programmable
22	External Interrupt2 (INT2)	Programmable	Programmable
23	External Interrupt3 (INT3)	Programmable	Programmable
24	External Interrupt4 (INT4)	Programmable	Programmable
25	External Interrupt5 (INT5)	Programmable	Programmable
26	External Interrupt6 (INT6)	Programmable	Programmable
27	External Interrupt7 (INT7)	Programmable	Programmable

Electrical Characteristics

Absolute Maximum Ratings[1]

Parameter	Symbol	Rating	Unit
Supply Voltage	Vddb	-0.3 ~ 4.6	V
	Vddc, Vdd_xtl	-0.3 ~ 3.5	V
Input Voltage[2]	VIN	-0.3 ~ Vddb + 0.3	V
Output Voltage[2]	VOUT	-0.3 ~ Vddb + 0.5	V
Input Voltage[3]	VIN	-0.3 ~ Vdd_xtl + 0.3	V
Output Voltage[3]	VOUT	-0.3 ~ Vdd_xtl + 0.5	V
Storage Temperature	TSTG	-40 ~ +125	°C
Power Dissipation (Package Limit Ta=70°C)	PDPKG	1000	mW

NOTES:
1. These stress ratings are only for transient conditions. Operation at or beyond absolute maximum rating conditions may affect reliability and cause permanent damage to the device.
2. This condition is applied pins except for 89,90,92,93.
3. This condition is applied 89,90,92,93 pins.

Recommended Operating Conditions

Parameter	Symbol	Min.	Max.	Unit
Supply Voltage (Core)	Vddc	2.35	2.75	V
Supply Voltage (PLL)	Vdd_xtl	2.35	2.75	V
Supply Voltage (I/O)	Vddb[1]	3.0	3.6	V
Clock Frequency	FSYSCLK	1	50	MHz
Commercial Operating Temperature	TOPR	0	+70	°C

NOTES:
[1] Vddb ≥ Vddc = Vdd_xtl
Unused input pins should be pulled low or high to their inactive state.

DC Spefications

Over Recommended Operating Conditions
(Ta = 0 °C-70°C, Vddc = 2.35V-2.75V, Vdd_xtl = 2.35V-2.75V, Vddb = 3.0V-3.6V)

Parameter	Symbol	Test Condition	Min.	Typ.	Max.	Unit	NOTES
Input Low Voltage	VIL		0		0.2Vdd	V	8
Input High Voltage	VIH		0.8Vdd		Vdd	V	8
Output Low Voltage	VOL	I_{OL} = 4mA			0.5	V	6
		I_{OL} = 8mA			0.5	V	7
Output High Voltage	VOH	I_{OH} = - 4mA	Vdd -0.5			V	6,9
		I_{OH} = - 8mA	Vdd -0.5			V	7
Input Leakage Current	ILI	V_I = 0 ~ Vddb	-1.0		+1.0	uA	
High Impedance (OFF-State) Output Leakage Current	IOZ	Vin = VIH Vo = 0 ~ Vddb	-1.0		+1.0	uA	
Operating Power (Active Mode)	Iact(Vddc)	Vi = 0 or Vddb-0.2V Output = open f_{OSC}=31kHz, f_{PLL}=50MHz		94		mA	1
	Iact(Vddb)			8			
	Iact(Vdd_xtl)			28			
Operating Power (Standby Mode)	Istb(Vddc)	Vi = 0 or Vddb-0.2V f_{OSC}=31kHz, f_{PLL}=50MHz		74		mA	2
	Istb(Vddb)			8			
	Istb(Vdd_xtl)			28			
Operating Power (Sleep Mode)	Islp(Vddc)	Vi = 0 or Vddb-0.2V f_{OSC}=31kHz, f_{PLL}=50MHz		2		mA	3
	Islp(Vddb)			8			
	Islp(Vdd_xtl)			50		uA	
Operating Power (Stop Mode)	Istp(Vddc)	Vi = 0 or Vddb-0.2V f_{OSC}=31kHz		9		uA	4
	Istp(Vddb)			20			
	Istp(Vdd_xtl)			1			
Operating Power (Stop2 Mode)	Istp2(Vddc)	Vi = 0 or Vddb-0.2V		9		uA	5
	Istp2(Vddb)			1			
	Istp2(Vdd_xtl)			1			

NOTES:
1. Active Mode: Iact = Istb + CPU + Cache
2. Standby Mode: Istb = Islp + LCDC + SDRAMC + DMA + Timer + PWM + Bus Controller
3. Sleep Mode: Islp = Istp + PLL Current
4. Stop Mode: Istp = Leakage Current + Osc current + RTC (1 or 2 gates)
5. Stop2 Mode: Istp2 = Leakage Current + RTC (1 or 2 gates)
6. Pin=(6,70-75,77-81,83,84,86.87,95,98-101,103-106,108-115,117-123,125-129,131,132,135,136)
7. Pin=(9-11,13-15,17-21,23-27,29-33,35-39,41-43,50,51,53-56,58-62,64-68,133,138,139,141-145, 147-151, 153-157,159-163165-169,171-175)
8. Vdd=Vdd_xtl for Pin 92,93, Vddb for other Pins.
9. Vdd=Vddb.

AC Test Conditions

(Ta = 0 °C-70°C, Vddc = 2.35V-2.75V, Vdd_xtl = 2.35V-2.75V, Vddb = 3.0V-3.6V)

Parameter	Rating	Unit
Input Pulse Levels	VIH = 2.4 VIL = 0.6	V
Input Rise and Fall Times	3.0	ns
Input and Output Timing Ref. Levels	VOH = 2.0 VOL = 1.0	V
Output Load*	50	pF

NOTES: *Includes scope and jig capacitance

AC Specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit	NOTE
Trad	Output Delay from XCLKIN ↓ to RA			24	ns	
Trah	Output Hold from XCLKIN ↓ to RA	4			ns	
Tced	Output Delay from XCLKIN ↓ to nCE			23	ns	
Tceh	Output Hold from XCLKIN ↓ to nCE	4			ns	
Tred	Output Delay from XCLKIN ↓ to nRE			22	ns	
Treh	Output Hold from XCLKIN ↓ to nRE	4			ns	
Twed	Output Delay from XCLKIN ↓ to nWE			24	ns	
Tweh	Output Hold from XCLKIN ↓ to nWE	4			ns	
Trdd	Output Delay from XCLKIN ↑ to RD			24	ns	
Trdh	Output Hold from XCLKIN ↑ to RD	4			ns	
Trdis	RD Setup time to XCLKIN ↓	2			ns	
Trdih	RD Hold time to XCLKIN ↓	7			ns	
Tscmd	Output Delay from SDCLK ↓ to SDRAMC signals			8	ns	
Tscmh	Output Hold from SDCLK ↓ to SDRAMC signals	0			ns	
Tsdrd	Output Delay from SDCLK ↓ to SDRA			8	ns	
Tsdrh	Output Hold from SDCLK ↓ to SDRA	0			ns	
Tsdqd	Output Delay from SDCLK ↓ to DQM			8	ns	
Tsdqh	Output Hold from SDCLK ↓ to DQM	0			ns	
Tsdd	Output Delay from SDCLK ↓ to SDD			8	ns	
Tsdh	Output Hold from SDCLK ↓ to SDD	0			ns	
Tsdds	SDD Setup time to SDCLK ↑	16			ns	
Tsddh	SDD Hold time to SDCLK ↑	-4			ns	
Tscked	Output Delay time from SDCLK ↓ to SDCKE	0		8	ns	
Tstcvd	Output Delay from SHIFT_CLK to VD	10			ns	
Tstcvh	Output Hold from SHIFT_CLK to VD	0			ns	
Ttfcvd	Output Delay from SHIFT_CLK to VD			9	ns	
Ttfcvh	Output Hold from SHIFT_CLK to VD	0			ns	
Ttfcld	Output Delay from SHIFT_CLK to LINE_CLK			9	ns	
Tvcon	time from Vdd_xtal on to Vddc on	0			ns	
Tvcoff	time from Vddc off to Vdd_xtal off	0			ns	
Tvbon	time from Vddc on to Vddb on	0			ns	
Tvboff	time from Vddb off to Vddc off	0			ns	
Tbcks	nRESETi Setup time to Vddb on	20			ns	
Tbckh	nRESETi Hold time to Vddb off	20			ns	
Tresiw	nRESETi Pulse Width	3			cycle	1
Tresod	Output Delay from XCLKIN ↓ to nRESETo			20	ns	
Tresoh	Output Hold from nRESETi ↑ to nRESETo	-	64	-	cycle	1
Toscd	Oscillator stabilization time after Power On		500		ms	3
Fosc	Oscillator operation frequency	-	32.768	-	kHz	
Tplock	PLL Lock on time	10			us	
Twud	PLL warming up time	0		65536	cycle	2
Fpll	PLL operation frequency	TBD		50	MHz	

NOTE:

1) Cycle is the system clock.
2) Cycle is the Oscillator cycle. This parameter is programmable by the PLL Controller, and the default value is 65536 cycle.
3) OSC circuit connected outside for the measurement condition is shown as follow

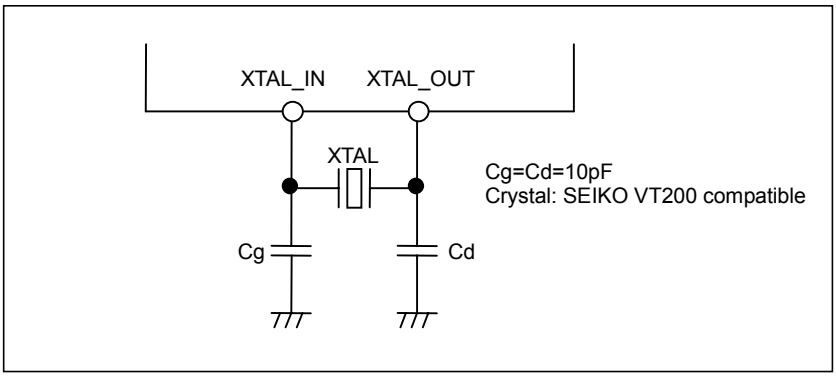


Figure 13. Recommended OSC circuit connection

Pin Descriptions

Pin(S)	Name Function 1	Name Function 2	Name Function 3	Direction	Description
1	XTESTMODE			Input	Test Mode Enable. '1'=Test, '0'=Normal
2	nTRST			Input	JTAG Reset Input. This pin must be pulled down[1] or pulsed low to initialize the internal JTAG Tap Controller and archive normal operation.
3	TCK			Input	JTAG Clock Input. This pin must be pulled up during normal operation.
4	VSS				Ground
5	TDI			Input	JTAG Data Input. This pin must be pulled up during normal operation.
6	TDO			Output	JTAG Data Output
7	TMS			Input	JTAG Mode Select. This pin must be pulled up during normal operation.
8	VDDC				Core Power
9	RA[25]/nRAS			Output	Address Bus of EBI Controller / SDRAM nRAS
10	RA[24]/nCAS			Output	Address Bus of EBI Controller / SDRAM nCAS
11	RA[23]/nDWE			Output	Address Bus of EBI Controller / SDRAM Write Enable. '0' = write, '1' = read
12	VSS				Ground
13	RA[22]			Output	Address Bus of EBI Controller
		PIO[63]		I/O	General Purpose I/O
14	RA[21]			Output	Address Bus of EBI Controller
		PIO[62]		I/O	General Purpose I/O
15	RA[20]			Output	Address Bus of EBI Controller
		PIO[61]		I/O	General Purpose I/O
16	VDDB				I/O Power
17	RA[19]			Output	Address Bus of EBI Controller
18	RA[18]			Output	Address Bus of EBI Controller
19	RA[17]			Output	Address Bus of EBI Controller
20	RA[16]			Output	Address Bus of EBI Controller
21	RA[15]			Output	Address Bus of EBI Controller
22	VSS				Ground
23	RA[14]			Output	Address Bus of EBI / SDRAM Controller
24	RA[13]			Output	Address Bus of EBI / SDRAM Controller
25	RA[12]			Output	Address Bus of EBI / SDRAM Controller
26	RA[11]			Output	Address Bus of EBI / SDRAM Controller
27	RA[10]			Output	Address Bus of EBI / SDRAM Controller
28	VDDB				I/O Power
29	RA[9]			Output	Address Bus of EBI / SDRAM Controller
30	RA[8]			Output	Address Bus of EBI / SDRAM Controller
31	RA[7]			Output	Address Bus of EBI / SDRAM Controller
32	RA[6]			Output	Address Bus of EBI / SDRAM Controller
33	RA[5]			Output	Address Bus of EBI / SDRAM Controller
34	VSS				Ground
35	RA[4]			Output	Address Bus of EBI / SDRAM Controller
36	RA[3]			Output	Address Bus of EBI / SDRAM Controller
37	RA[2]			Output	Address Bus of EBI / SDRAM Controller
38	RA[1]			Output	Address Bus of EBI / SDRAM Controller
39	RA[0]			Output	Address Bus of EBI / SDRAM Controller
40	VDDB				I/O Power
41	nRE			Output	EBI Controller Read Enable Output
42	nWE[3]			Output	EBI Controller Write Enable Output
		PIO[60]		I/O	General Purpose I/O
43	nWE[2]			Output	EBI Controller Write Enable Output
		PIO[59]		I/O	General Purpose I/O
44	VSS				Ground
45	CLKSEL			Input	Selects either internal PLL clock or the External clock as sysclock of the chip '0' = PLL, '1' = external clock.

Pin(S)	Name Function 1	Name Function 2	Name Function 3	Direction	Description
46	BIGEND			Input	ARM Byte Address Mode Selection. '0' = Little Endian, '1' = Big Endian. This signal must not be changed during operation.
47	VDDC				Core Power
48	BOOT[1]			Input	Selects startup mode memory width '10' = x32, '01' = x16, '00' = x8
49	BOOT[0]			Input	
50	nWE[1]			Output	EBI Controller Write Enable Output
		PIO[58]		I/O	General Purpose I/O
51	nWE[0]			Output	EBI Controller Write Enable Output
		PIO[57]		I/O	General Purpose I/O
52	VSS				Ground
53	nDCS[1]			Output	SDRAM Chip Select. '0' = selected, '1' = deselected.
		PIO[56]		I/O	General Purpose I/O
			nCE[7]	Output	Memory Chip Select
54	nDCS[0]			Output	SDRAM Chip Select. '0' = selected, '1' = deselected.
		PIO[55]		I/O	General Purpose I/O
			nCE[6]	Output	Memory Chip Select
55	nCE[5]			Output	Memory Chip Select
			PIO[54]	I/O	General Purpose I/O
56	nCE[4]			Output	Memory Chip Select
			PIO[53]	I/O	General Purpose I/O
57	VDDB				I/O Power
58	nCE[3]			Output	Memory Chip Select
59	nCE[2]			Output	Memory Chip Select
60	nCE[1]			Output	Memory Chip Select
61	nCE[0]			Output	Memory Chip Select
62	SDCLK			Output	SDRAM Clock
63	VSS				Ground
64	SDCKE			Output	SDRAM Clock Enable. '0' = deselected, '1' = selected.
65	DQM[3] / nByte[3]			Output	SDRAM Data Write Mask for handling Byte and Half-Word Transfers & EXTBIF nByte[3]
		PIO[52]		I/O	General Purpose I/O
66	DQM[2] / nByte[2]			Output	SDRAM Data Write Mask for handling Byte and Half-Word Transfers & EXTBIF nByte[2]
		PIO[51]		I/O	General Purpose I/O
67	DQM[1] / nByte[1]			Output	SDRAM Data Write Mask for handling Byte and Half-Word Transfers & EXTBIF nByte[1]
		PIO[50]		I/O	General Purpose I/O
68	DQM[0] / nByte[0]			Output	SDRAM Data Write Mask for handling Byte and Half-Word Transfers & EXTBIF nByte[0]
		PIO[49]		I/O	General Purpose I/O
69	VDDB				I/O Power
70	PIO[43]			I/O	General Purpose I/O
		XCLKEN		Output	External Clock Control
71	PIO[42]			I/O	General Purpose I/O
		CLKOUT		Output	System clock output
72	PIO[41]			I/O	General Purpose I/O
		nWAIT		Input	External Memory Wait. Facilitates the use of slow memories.
73	PIO[40]			I/O	General Purpose I/O
		DREQ1		Input	DMA Request Channel 1
			PWM_SYNC	Input	Pulse Width Modulator Sync
74	PIO[39]			I/O	General Purpose I/O
		nDACK1		Output	DMA Acknowledge Channel 1. Active Low.
			PWM3	Output	Pulse Width Modulator Output Signal
75	PIO[38]			I/O	General Purpose I/O

Pin(S)	Name Function 1	Name Function 2	Name Function 3	Direction	Description
		DEOT1		Output	DMA End of Transfer Channel 1
			PWM2	Output	Pulse Width Modulator Output Signal
76	VSS				Ground
77	PIO[37]			I/O	General Purpose I/O
		DREQ0		Input	DMA Request Channel 0
78	PIO[36]			I/O	General Purpose I/O
		nDACK0		Output	DMA Acknowledge Channel 0. Active Low.
79	PIO[35]			I/O	General Purpose I/O
		DEOT0		Output	DMA End of transfer Channel 0
80	PIO[34]			I/O	General Purpose I/O
		PWM1		Output	Pulse Width Modulator Output Signal
			CTOUT3	Output	Counter / Timer Output Signal
81	PIO[33]			I/O	General Purpose I/O
		PWM0		Output	Pulse Width Modulator Output Signal
			CTOUT2	Output	Counter / Timer Output Signal
82	VDDB				I/O Power
83	UTXD0			Output	UART 0 Transmit Data
		PIO[48]		I/O	General Purpose I/O
84	URXD0			Output	UART 0 Receive Data
		PIO[47]		I/O	General Purpose I/O
85	VSS				Ground
86	UTXD1			Output	UART 1 Transmit Data
		PIO[46]		I/O	General Purpose I/O
			CTOUT1	Output	Counter / Timer Output Signal
87	URXD1			Output	UART 1 Receive Data
		PIO[45]		I/O	General Purpose I/O
			CTOUT0	Output	Counter / Timer Output
88	VDD_XTL				Power Supply for Crystal buffer
89	XTAL_IN			Input	Crystal Clock Input
90	XTAL_OUT			Output	Crystal Clock Output
91	VSS_XTL				Ground Supply for PLL, OSC, & RTC
92	RTC_CLKIN			Input	RTC Clock Input
93	nBACKUP				Backup mode Active Low
94	nRESETI			Input	Asynchronous Reset. Active Low
95	nRESETO			Output	Reset Output. When low, indicates that the chip has successfully completed rest.
		PIO[44]		I/O	General Purpose I/O
96	XCLKIN			Input	External Clock Input
97	VDDC				Core Power
98	PIO[32]			I/O	General Purpose I/O
		INTR3		Input	External Interrupt Signal
			CTCLK	Input	Counter / Timer Clock
99	PIO[31]			I/O	General Purpose I/O
		INTR2		Input	External Interrupt Signal
			UCLK	Input	UART Clock
100	PIO[30]			I/O	General Purpose I/O
		INTR1		Input	External Interrupt Signal
			ALM	Output	Alarm out: RTC
101	PIO[29]			I/O	General Purpose I/O
		INTR0		Input	External Interrupt Signal
102	VSS				Ground
103	PIO[28]			I/O	General Purpose I/O
		SPI_IN		Input	SPI Data Input
104	PIO[27]			I/O	General Purpose I/O
		SPI_OUT		Output	SPI Data Output
105	PIO[26]			I/O	General Purpose I/O

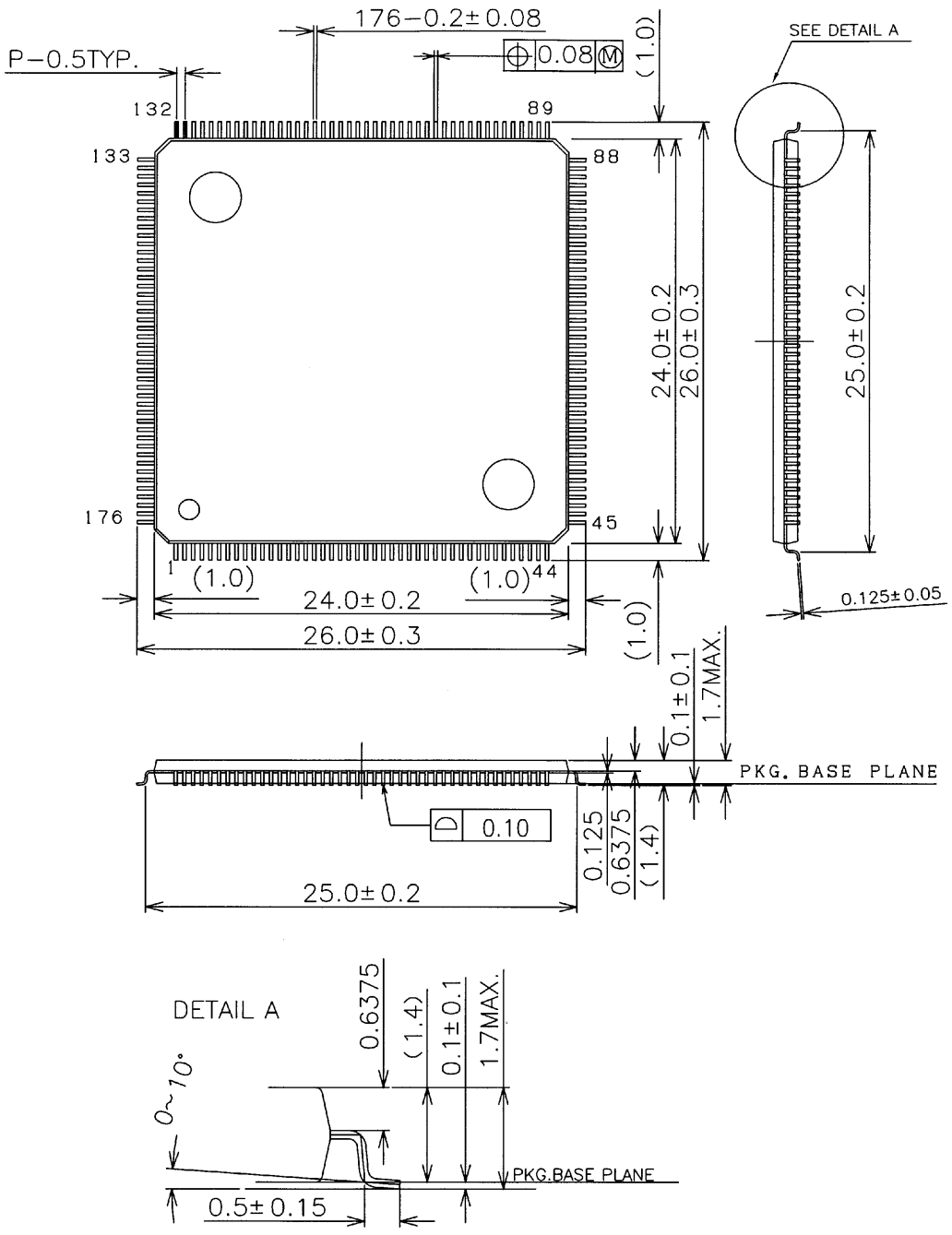
Pin(S)	Name Function 1	Name Function 2	Name Function 3	Direction	Description
		SPI_ENB		I/O	SPI Enable
106	PIO[25]			I/O	General Purpose I/O
		SPI_CLK		I/O	SPI Clock
107	VDDB				I/O Power
108	PIO[24]			I/O	General Purpose I/O
		VD[17]		Output	LCD Video Output
			nURTS0	Output	UART 0 request to send
109	PIO[23]			I/O	General Purpose I/O
		VD[16]		Output	LCD Video Output
			nUCTS0	Input	UART 0 clear to send
110	PIO[22]			I/O	General Purpose I/O
		VD[15]		Output	LCD Video Output
			nURI0	Input	UART 0 Ringer Indicator
111	PIO[21]			I/O	General Purpose I/O
		VD[14]		Output	LCD Video Output
			nUDCD0	Input	UART 0 Carrier Detect
112	PIO[20]			I/O	General Purpose I/O
		VD[13]		Output	LCD Video Output
			nUDTR0	Output	UART 0 Data Terminal Ready
113	PIO[19]			I/O	General Purpose I/O
		VD[12]		Output	LCD Video Output
			nUDSR0	Input	UART 0 Data Set Ready
114	PIO[18]			I/O	General Purpose I/O
		VD[11]		Output	LCD Video Output
			INTR7	Input	External Interrupt Signal
115	PIO[17]			I/O	General Purpose I/O
		VD[10]		Output	LCD Video Output
			INTR6	Input	External Interrupt Signal
116	VSS				Ground
117	PIO[16]			I/O	General Purpose I/O
		VD[9]		Output	LCD Video Output
			INTR5	Input	External Interrupt Signal
118	PIO[15]			I/O	General Purpose I/O
		VD[8]		Output	LCD Video Output
			INTR4	Input	External Interrupt Signal
119	PIO[14]			I/O	General Purpose I/O
		VD[7]		Output	LCD Video Output
120	PIO[13]			I/O	General Purpose I/O
		VD[6]		Output	LCD Video Output
121	PIO[12]			I/O	General Purpose I/O
		VD[5]		Output	LCD Video Output
122	PIO[11]			I/O	General Purpose I/O
		VD[4]		Output	LCD Video Output
123	PIO[10]			I/O	General Purpose I/O
		VD[3]		Output	LCD Video Output
124	VDDB				I/O Power
125	PIO[9]			I/O	General Purpose I/O
		VD[2]		Output	LCD Video Output
126	PIO[8]			I/O	General Purpose I/O
		VD[1]		Output	LCD Video Output
127	PIO[7]			I/O	General Purpose I/O
		VD[0]		Output	LCD Video Output
128	PIO[6]			I/O	General Purpose I/O
		CLPOWER		Output	LCD VDD Enable. ‘0’ = Off, ‘1’ = On.
129	PIO[5]			I/O	General Purpose I/O

Pin(S)	Name Function 1	Name Function 2	Name Function 3	Direction	Description
		CLFP		Output	LCD Frame Clock / Vertical Sync
130	VSS				Ground
131	PIO[4]			I/O	General Purpose I/O
		CLLP		Output	LCD Line Clock / Horizontal Sync
132	PIO[3]			I/O	General Purpose I/O
		CLAC		Output	LCD AC Bias Control
133	PIO[2]			I/O	General Purpose I/O
		CLCP		Output	LCD Pixel / Shift Clock
134	VDDC				Core Power
135	PIO[1]			I/O	General Purpose I/O
		CLLE		Output	LCD Line End
136	PIO[0]			I/O	General Purpose I/O
		EXTLCDCLK		Input	External LCD Clock Input
137	VSS				Ground
138	RD[31]			I/O	EBI / SDRAM Data Input / Output
		PIO[79]		I/O	General Purpose I/O
139	RD[30]			I/O	EBI / SDRAM Data Input / Output
		PIO[78]		I/O	General Purpose I/O
140	VDDB				I/O Power
141	RD[29]			I/O	EBI / SDRAM Data Input / Output
		PIO[77]		I/O	General Purpose I/O
142	RD[28]			I/O	EBI / SDRAM Data Input / Output
		PIO[76]		I/O	General Purpose I/O
143	RD[27]			I/O	EBI / SDRAM Data Input / Output
		PIO[75]		I/O	General Purpose I/O
144	RD[26]			I/O	EBI / SDRAM Data Input / Output
		PIO[74]		I/O	General Purpose I/O
145	RD[25]			I/O	EBI / SDRAM Data Input / Output
		PIO[73]		I/O	General Purpose I/O
146	VSS				Ground
147	RD[24]			I/O	EBI / SDRAM Data Input / Output
		PIO[72]		I/O	General Purpose I/O
148	RD[23]			I/O	EBI / SDRAM Data Input / Output
		PIO[71]		I/O	General Purpose I/O
149	RD[22]			I/O	EBI / SDRAM Data Input / Output
		PIO[70]		I/O	General Purpose I/O
150	RD[21]			I/O	EBI / SDRAM Data Input / Output
		PIO[69]		I/O	General Purpose I/O
151	RD[20]			I/O	EBI / SDRAM Data Input / Output
		PIO[68]		I/O	General Purpose I/O
152	VDDB				I/O Power
153	RD[19]			I/O	EBI / SDRAM Data Input / Output
		PIO[67]		I/O	General Purpose I/O
154	RD[18]			I/O	EBI / SDRAM Data Input / Output
		PIO[66]		I/O	General Purpose I/O
155	RD[17]			I/O	EBI / SDRAM Data Input / Output
		PIO[65]		I/O	General Purpose I/O
156	RD[16]			I/O	EBI / SDRAM Data Input / Output
		PIO[64]		I/O	General Purpose I/O
157	RD[15]			I/O	EBI / SDRAM Data Input / Output
158	VSS				Ground
159	RD[14]			I/O	EBI / SDRAM Data Input / Output
160	RD[13]			I/O	EBI / SDRAM Data Input / Output
161	RD[12]			I/O	EBI / SDRAM Data Input / Output
162	RD[11]			I/O	EBI / SDRAM Data Input / Output

Pin(S)	Name Function 1	Name Function 2	Name Function 3	Direction	Description
163	RD[10]			I/O	EBI / SDRAM Data Input / Output
164	VDDB				I/O Power
165	RD[9]			I/O	EBI / SDRAM Data Input / Output
166	RD[8]			I/O	EBI / SDRAM Data Input / Output
167	RD[7]			I/O	EBI / SDRAM Data Input / Output
168	RD[6]			I/O	EBI / SDRAM Data Input / Output
169	RD[5]			I/O	EBI / SDRAM Data Input / Output
170	VSS				Ground
171	RD[4]			I/O	EBI / SDRAM Data Input / Output
172	RD[3]			I/O	EBI / SDRAM Data Input / Output
173	RD[2]			I/O	EBI / SDRAM Data Input / Output
174	RD[1]			I/O	EBI / SDRAM Data Input / Output
175	RD[0]			I/O	EBI / SDRAM Data Input / Output
176	VDDB				I/O Power

NOTE: Some JTAG ICE or JTAG Controller may not permit the pulled down nTRST signal. In such case the power on reset is recommended.

Mechanical Dimensions



Name	LQFP176-P-2424
Unit	mm
Lead Finish	TIN-LEAD PLATING
Lead Material	Cu

NOTE: Plastic body dimensions do not include burr of resin.

SHARP

NORTH AMERICA

SHARP Microelectronics
of the Americas (SMA)
5700 NW Pacific Rim Blvd.
Camas, WA 98607,
U.S.A.
Tel.: +1-360-834-2500
Fax: +1-360-834-8903
http://www.sharpsma.com

EUROPE

SHARP Microelectronics
Europe (SME)
Sonninstrasse 3
D-20097 Hamburg
Germany
Tel.: +49-40-2376-2286
Fax: +49-40-2376-2232
http://www.sharpsme.com

ASIA

SHARP Corporation
Integrated Circuits Group
2613-1 Ichinomoto-Cho
Tenri, Nara, 632-8567,
Japan
Tel.: +81-743-65-1321
Fax: +81-743-65-1532
http://www.sharp-world.com