SHARP®

LH79531

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.

LH79531 Universal Microcontroller Product Summary

Revision No. ICG000016-001

Released on November 12, 2002

Table of Contents

Introduction

The LH79531, 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 LH79531 combines a 32-bit ARM7TDMI™ RISC, color LCD controller, Cache, Local SRAM, a number of essential peripherals such as Direct Memory Access, Serial and Parallel Interfaces, Infrared support, Counter/Timers, Real Time Clock, Watch Dog Timer, Pulse Width Modulators, 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 LH79531'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 LH79531 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
- Dual Bus Architecture
  - SDRAM 32-bit Data Bus
  - SRAM 16-bit Data Bus
- 8 kB On-Chip Memory
  - 8 kB Cache / 0 kB SRAM
  - 4 kB Cache / 4 kB SRAM
  - 0 kB Cache / 8 kB SRAM
  - On-Chip Cache
    - Unified Instruction / Data
    - Supports Bytes, Half-words, Words
    - Write-back / Write-through
    - Write Buffer
- On-chip Programmable PLL and oscillator
- Clock and Power Management
- Programmable Color LCD Controller
  - Up to 1024 x 768 Resolution
  - 18-bit Video Bus
  - Direct Interface to HR-TFT panels
  - Supports STN, Color STN, HR-TFT, TFT, DMTN
  - Supports 15 Gray Shades
  - TFT: Supports 256 Colors Selected from a Palette of 64 k Colors or 64 k Direct Colors
  - Color STN: Supports 256 Colors Selected from a palette of 3375 Colors or 3375 Direct Colors
- On-chip Programmable Interrupt Controller
  - Six External Interrupts
- Three UARTs - 16C550-like
  - Support for Rx, Tx, RTS & CTS
- Universal IR Communication Controller
  - IrDA-1.0 (115.2 kbps)
  - IrDA-1.1 (4 Mbps)
  - DASK IR Interface
- Real-Time Clock (RTC)
  - Full Calendar
  - Separate Power & Clock
- Two DMA channels
  - Internal and External Transfers
  - Single & Burst
  - Buffered
  - 8-bit, 16-bit, and 32-bit Transfers
- Four 16-bit Pulse Width Modulators
- Synchronous Serial Interface (Compatible with SPI, µWire, TI Synchronous Serial Standard)
- USB Device (Compatible with USB1.1 Standard)
- Flexible Programmable Memory Interface
  - SRAM / Flash / ROM Controller
    - 26-bit External Address Bus
    - 16/8-bit External Data Bus
    - Eight Segments (64 MB Each)
  - SDRAM Controller
    - 15-bit External Address Bus
    - 32/16-bit External Data Bus
    - Two Segments (256 Mbytes each)
    - Supports Self & Auto Refresh
- 74 Programmable Parallel I/O Signals
- Three 16-bit Counter / Timer Channels
  - Interrupt on Terminal Count
  - Rate Generator / Square Wave Generator
  - Cascadeable (32-bit or 48-bit)
- Hardware Watchdog Timer
  - Programmable Time-out Intervals
  - Protection Mechanism
  - System Reset & Interrupt Options
- JTAG & Debug Support
- Core & Boundary Scan Chains
- Little & Big Endian
- Package: 216-pin TQFP
- Operating Conditions
  - Core: 2.35V - 2.75 V
  - I/O: 3.0 V - 3.6 V
  - 0°C to 70°C (Commercial)

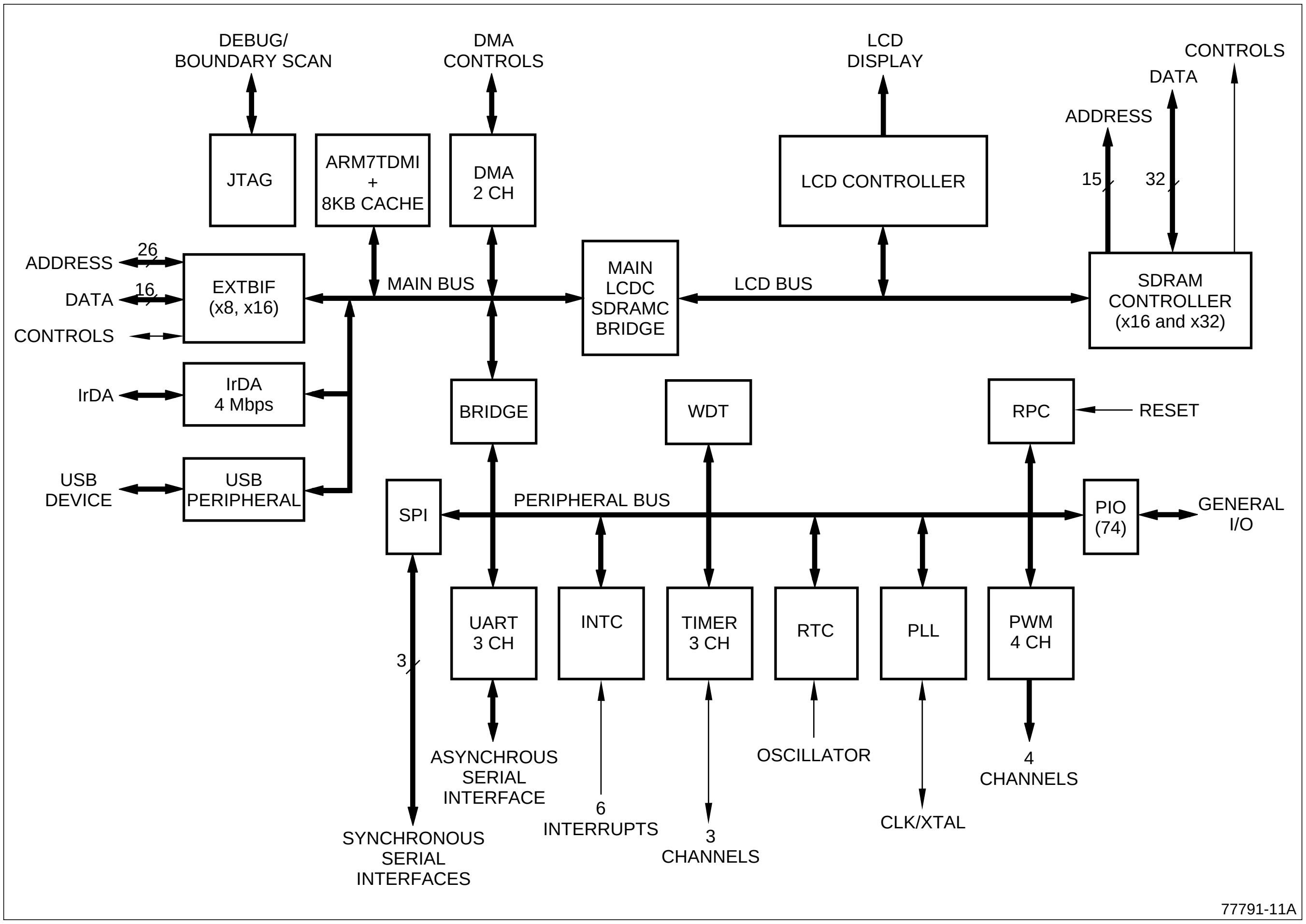


Figure 1. Block Diagram of LH79531

Application Flexibility

The LH79531's flexibility simplifies the creation of innovative products. In Figure 2, the LH79531 forms the basis of an advanced Internet appliance. Its many available functions allow the use of a touch screen, plus its low power consumption makes it an excellent choice for portable applications.

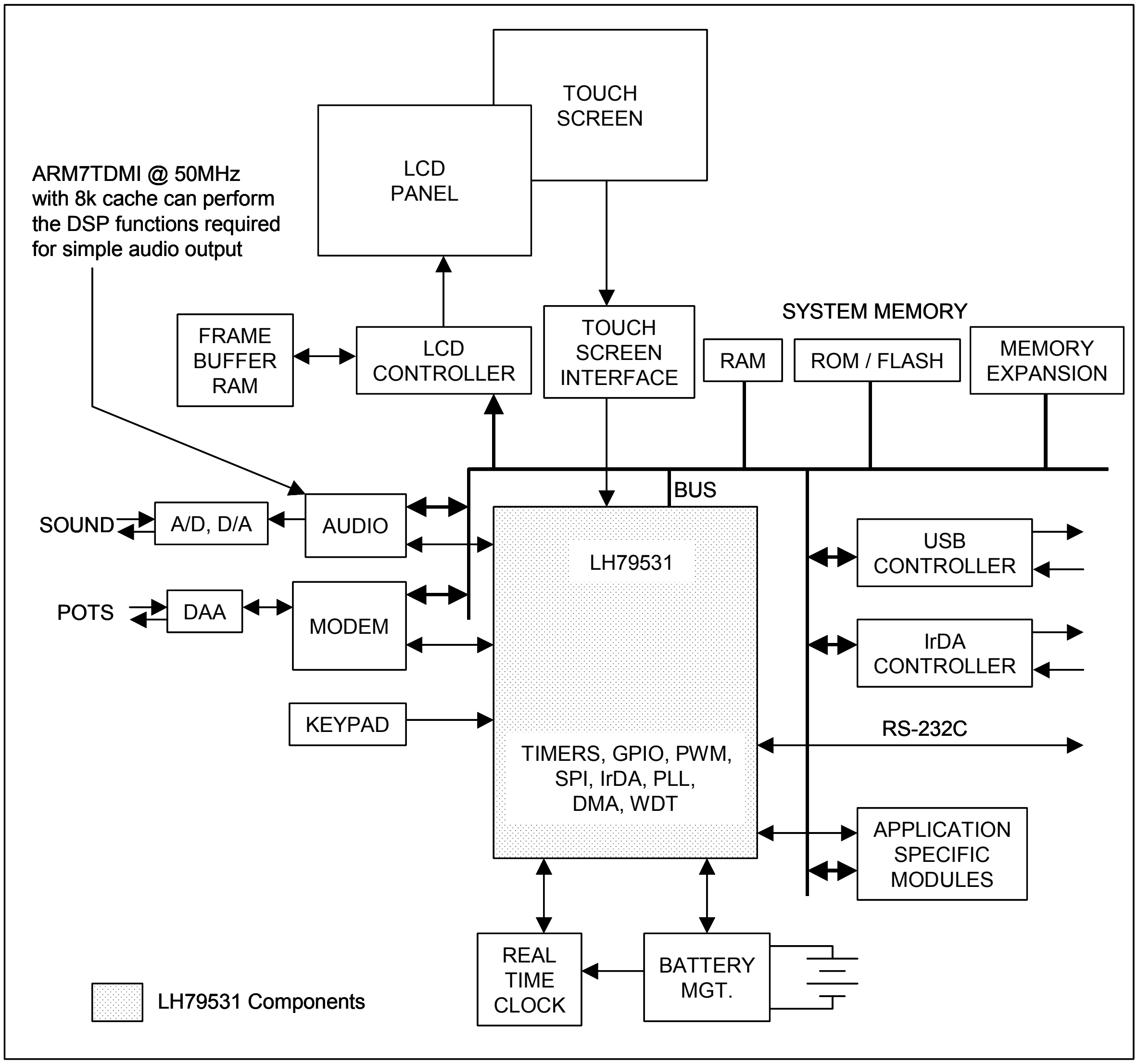


Figure 2. Sample Application Using the LH79531

Functional Descriptions

ARM7TDMI™ Core

The LH79531 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.

LH79531 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 LH79531 provides the following data-path-management resources on chip:

- Three 32-bit high-bandwidth internal data busses
- Up to 8 kB of zero-wait-state instruction/data cache, configurable as write-back or write-through
- Up to 8 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 32-bit interface to a common program and graphics memory
- An External Bus Interface with 26 address and 16 bi-directional data pins
- A high-resolution LCD controller with built-in DMA and HR-TFT timing logic
- A 2-channel general purpose DMA controller

Memory Map

All system resources accessible by the LH79531 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 segments 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 LH79531.

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 3.

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 LH79531, 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.

The three separate busses provide bi-directional address/control and data transport between system resources, allowing transactions to be originated concurrently by the ARM7TDMI™, DMA controller, and LCD controller. 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

The bus decoders for block selection, privilege information, and configuration information such as bus latency use memory map lookup tables. Their principal function is to decode addresses to select 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

Separating the LCD bus optimizes latency and bandwidth between the LCD Controller and external SDRAM (graphics memory region) with little disruption of local processor and peripheral activity. The LCD, cache, and DMA controllers perform the majority of their data transfers in 4-word sequential bursts within a single memory segment, maximizing throughput. This supports high pixel–rate bandwidth to the LCD since the ARM7TDMI, cache, and Local SRAM have their own bus. A high hit rate to cache and/or Local SRAM provides good processor throughput with little interference to graphics updates or peripheral DMA.

Code, Data and Graphics memory can reside in either SRAM or SDRAM. For high performance, graphics memory should reside in SDRAM, and Code/Data in SRAM.

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 LH79531 provides three 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	LCD Bus	Main Bus	Peripheral Bus
LCDC	1	1	-
DMA Channel 0	2	2	1
CPU	3	3	2
DMA Channel 1	4	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	LCD Bus	Main Bus	Peripheral Bus
LCDC	1	1	-
DMA Channel 0	2	2	1
CPU	4	4	3
DMA Channel 1	3	3	2

The LH79531 is optimized to have the LCD frame buffer in SDRAM. The system can also support graphics memory in SRAM (in addition to Code and Data) or Code and Data in SDRAM (in addition to Graphics memory). The presence of the bridges allows the busses to operate independently.

Memory Region

The system memory map of the LH79531 has two views, based on RPC_MAP (the Reset Power Control Memory Map; See Reset and Power Controller Section):

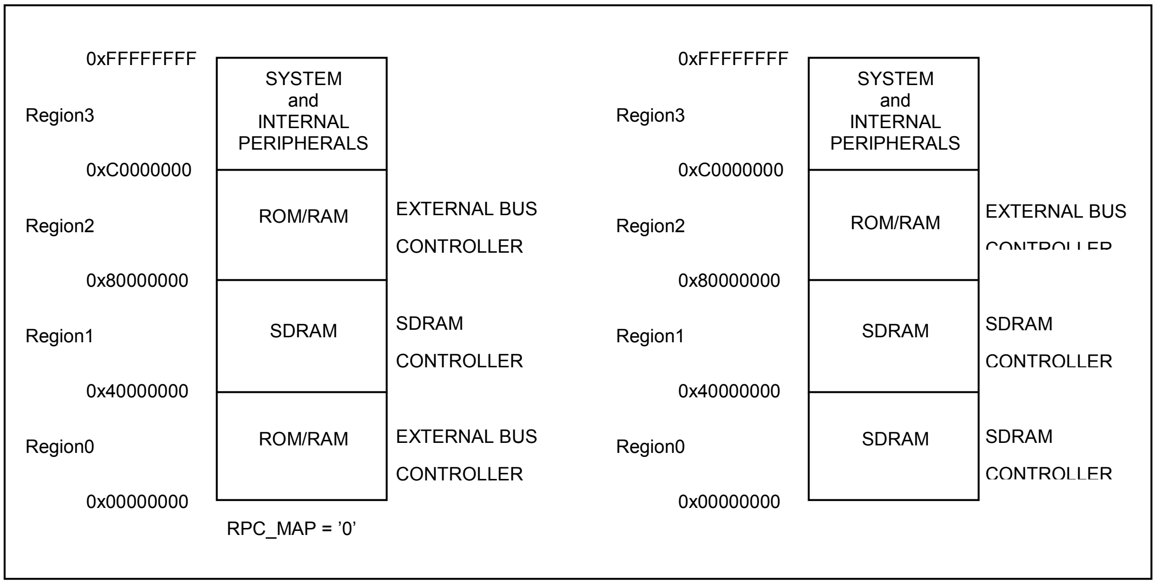


Figure 3. Memory Map

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, Table 2, and Table 3 show the addresses, Remap functions, and Class of the various devices within the LH79531.

Table 1. Main Bus Memory Mapping

Address	REMAP = '0'	REMAP = '1'	Class
0x80000000	Ext. ROM/RAM	Ext. ROM/RAM	
0xC0000000	-	-	
0xFFFF0000	IrDA	IrDA	Int.
0xFFFF0400	USB	USB	Int.
0xFFFF0800	DMA	DMA	Int.
0xFFFFA000	MainASB	MainASB	Sys.
0xFFFFA400	EXTBIF	EXTBIF	Sys.
0xFFFFAC00	Cache Ctrl	Cache Ctrl	Sys.
0xFFFFB000	-	-	
0xFFFFE400	-	-	

NOTE: -: Reserved.

Table 2. LCD Bus Memory Mapping

Address	REMAP = '0'	REMAP = '1'	Class
0x40000000	SDRAM	SDRAM	
0xC0000000	-	-	
0xFFFF2000	LCDC	LCDC	Int.
0xFFFF2400	-	-	
0xFFFF2800	LCDMUX	LCDMUX	Int
0xFFFFC000	SDRAMC	SDRAMC	Sys.
0xFFFFC400	LCDASB	LCDASB	Sys

NOTE: -: Reserved.

Table 3. Peripheral Bus Memory Mapping

Address	MACRO	Device Class
0xFFFF4000	UART0	Internal
0xFFFF4400	UART1	Internal
0xFFFF4800	UART2	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 LH79531 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 4. 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 8 kB combined instruction and data cache can be configured as 8 kB cache, 4 kB cache and 4 kB SRAM, or 8 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, LCD, and external busses. The cache write-coherency policy can be configured as either write-back (also known as copy-back) or write-through. Whenever the cache fills in write-back 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

8 k bytes of on-chip data SRAM is shared between cache and local SRAM. Up to 8 kB of fast (no wait state) local scratchpad-memory can be made available. Its size is either 4 kB or 8 kB, organized according to the CM bits of the CacheControlReg. The data contents of Local SRAM are undefined after reset. This memory uses the cache data-memory array, so the total amount of Cache + Local SRAM is exactly 8 kB, partitioned either as all cache, all Local SRAM, or 4 kB cache and 4 kB Local SRAM according to the CM 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

8 k bytes of on-chip data SRAM is shared between cache and local SRAM. Three possible operating modes give the system designer a high degree of flexibility. The three modes are shown in Table 5.

Table 5. Memory Combinations

Mode	Cache	SRAM
Cache Mode	8 kB	0
SRAM Mode	0	8 kB
Split Mode	4 kB	4 kB

Register Summary

Table 6. Local SRAM Configuration Registers

Address	Register	Name	Access	Size	Reset Value
0xFFFFAC84	SRAMTag	Local SRAM Start Address Register for CPU	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
0xFFFFC400	LCDASBCTRL	Local SRAM Start Address Register for LCD Bus	R/W	32	0x00000000

Table 7. Cache Contorol Register

Address	Register	Name	Access	Size	Reset Value
0xFFFFAC80	CacheControlReg	Cache Control Register	R/W	11	*

External Bus Interface (EBI)

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

Features

- Flexible programmable memory interface
  - Supports SRAM, Normal or Page Mode Flash and ROM
  - 26-bit External Address Bus
  - 16-bit External Data Bus (x8 & x16)
  - nWE0, nWE1 and nRE support for Byte/Halfword 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)

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 an N-cycle timing is resumed**.** This page mode function only affects the Read cycle.

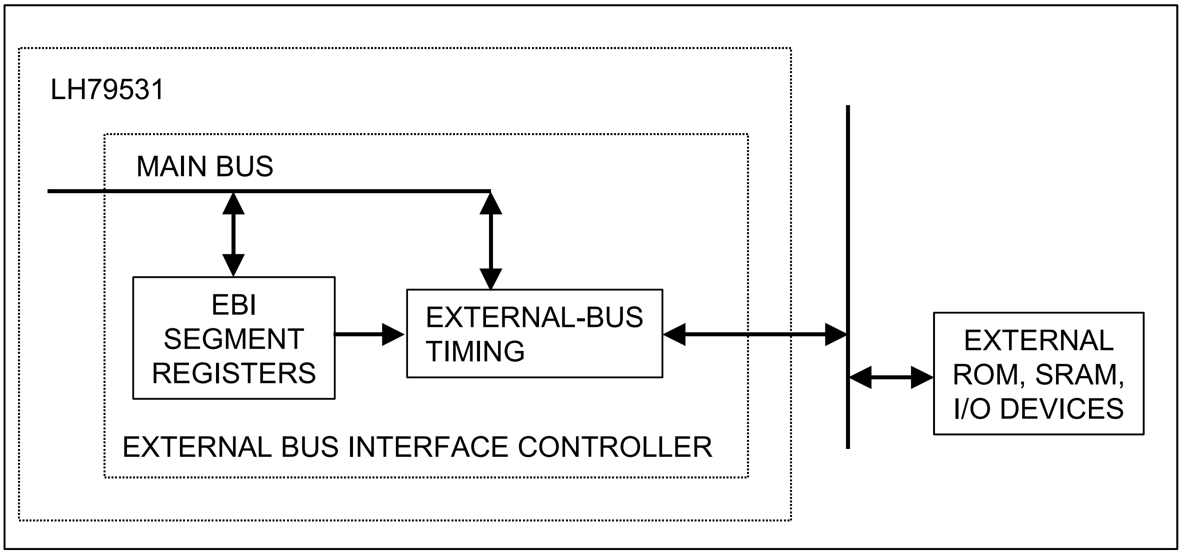


Figure 4. External Bus Interface Block Diagram

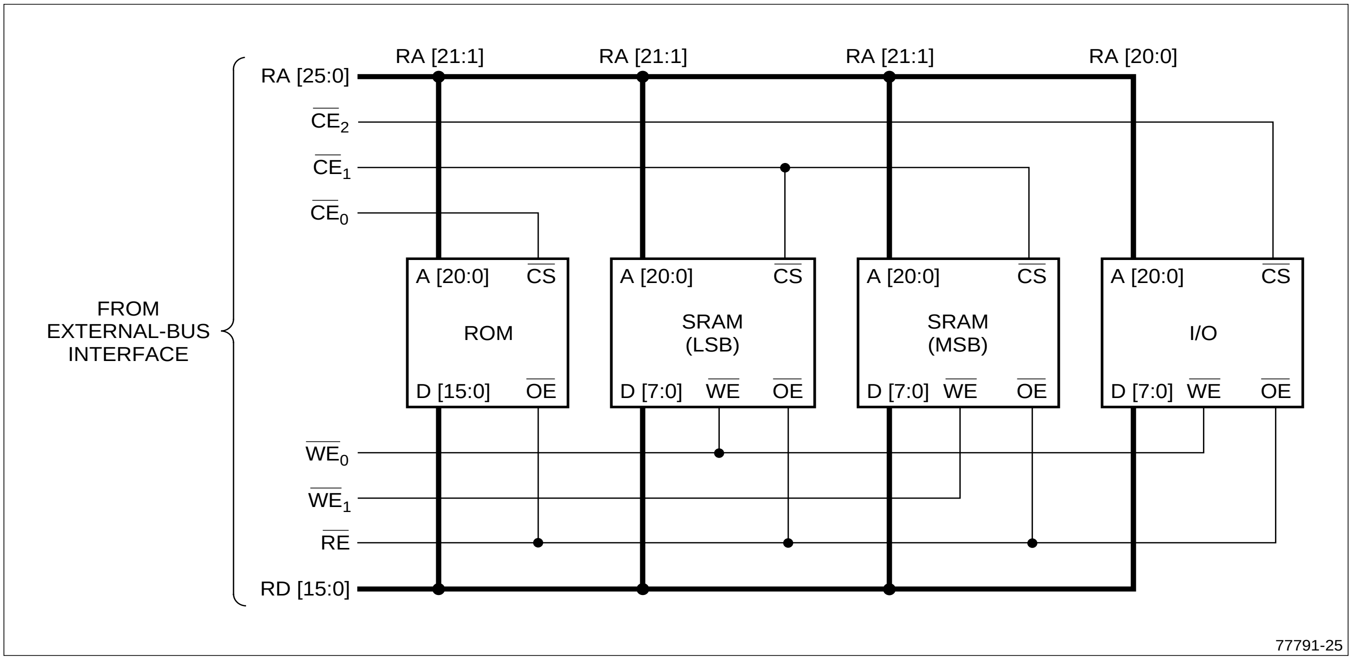


Figure 5. Example of External Bus Connections

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

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

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

Synchronous Dynamic RAM Controller (SDRAMC)

Features

- 15 bit external address bus (256 MB SDRAM support)
- Programmable address multiplexing (64 MB, 128 MB, 256 MB)
- 32 bit or 16 bit external data bus
- 32 bit or 16 bit or 8 bit access
- Big or Little Endian
- 1 or 2 block, 2 or 4 banks in one block (A13 and A12 are used as bank select signal)
- 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

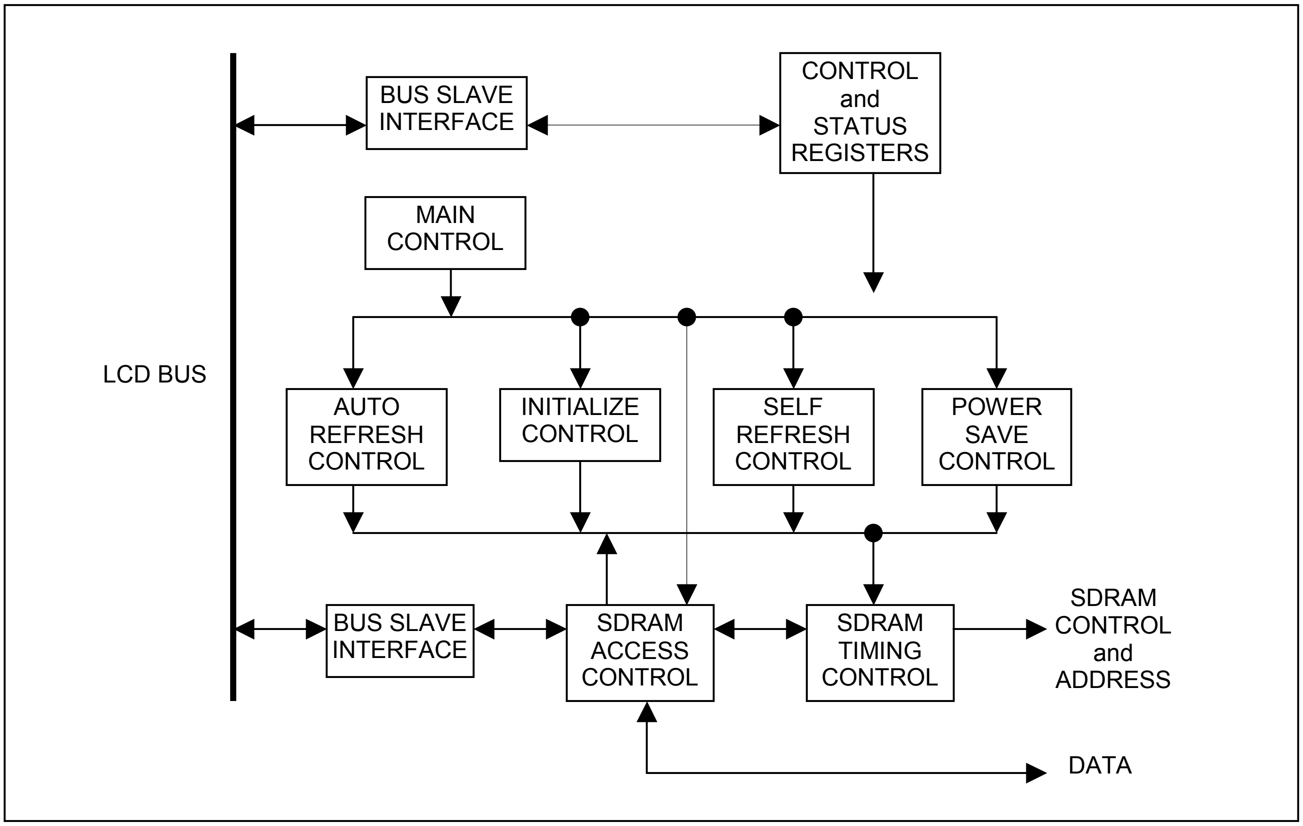


Figure 6. SDRAM Control Block Diagram

Register Summary

Table 10. 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		-	-	BAW	EBW
0x008	SDRAM_SEG1_REG	R/W	32	0x000000C0	E	-	Start Address (17 bits)																	-	-	-	-	-	-		-		Size			
0x00C	SDRAM_SDR1_REG	R/W	10	0x00000003	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CAW		-	-	RAW		-	-	BAW	EBW
0x010	SDRAM_CONTROL	R/W	5	0x00000000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DOZ	PME	PSE	SRE	SCE
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

Table 11 and Table 12 show the supported configurations of SDRAMs and the relationship between external SDRAM multiplexed address bus and internal address bus.

Table 11. Address Multiplexing (32 bit External Data Bus Mode)

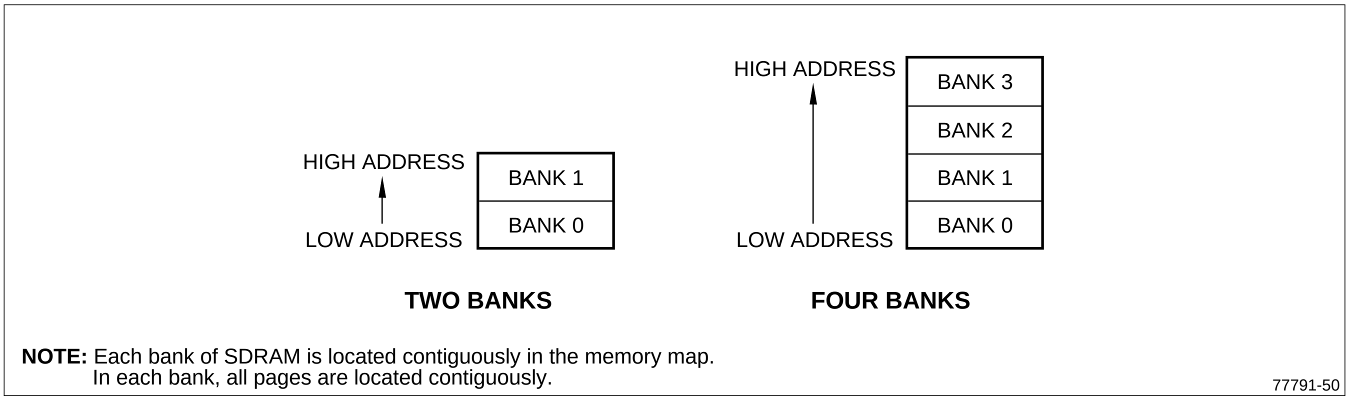
Total Size	Chip Size	Configuration				Addr-MUX	Bits	A14	A13	A12	A11	A10	A9	A8	A [7:0]
		Words	Bits	Banks	Chips										
8MB	16M	1M	8	2	4	Row	11			22		21	20	19	18:11
						Col	9			22		AP		10	9:2
4MB	16M	0.5M	16	2	2	Row	11			21		20	19	18	17:10
						Col	8			21		AP			9:2
8MB	64M	0.5M	32	4	1	Row	11		22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

NOTES:

1. A0 - A13 refer to the external pins named sdra0 - sdra13.
2. A14 refers to the multiplexed function pin named sdra14.
3. Figures in the table refer to the Main address bus bit locations.
4. Shaded entries are ‘bank select’.
5. AP (in A10 of ‘Col’ addresses) refers to optional bank select.
6. Null entries are ‘don’t care’, but stable.
7. HA is generated by SDRAM Controller when in word (32-bit) access. Otherwise HA is Main bus address <1>.



NOTE: Each bank of SDRAM is located contiguously in the memory map.
In each bank, all pages are located contiguously.

Figure 7. Memory Map

DMA Controller

Overview

The DMA controller (DMAC) in the LH79531 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	27	0	Enables individual interrupt sources to suspend an active DMA transfer in channel 0
0x88	DMACIntSourceCh1	R/W	27	0	Enables individual interrupt sources to suspend an active DMA transfer in channel 1
0x8C	DMACAckSelect	R/W	8	0	DMA Acknowledge select control for each peripheral

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

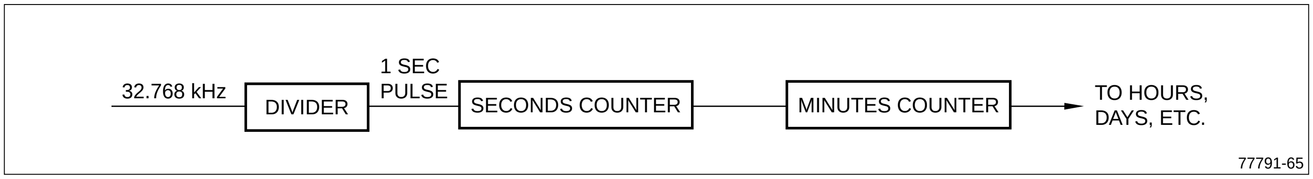


Figure 8. RTC Divider Block Diagram

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	Name	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)
- USB 48 MHz clock, internally/externally selectable
- USB/FIR clock with a selectable source from either:
  - System clock
  - External clock (USB_FIR_CLK: 48 MHz/ 24 MHz/ 16 MHz)
- 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, UART2	SYSCLK	On/Off	On/Off	Off	Off	Off
	UCLK[1]	On	On	On	On	On
CT0, CT1, CT2	XCLK	On/Off	On/Off	On/Off	Off	Off
LCDC	SYSCLK or MCLK	On/Off	On/Off	Off	Off	Off
IrDA, USB	SYSCLK	On/Off	On/Off	Off	Off	Off
	USB_FIR_CLK[1]	On	On	On	On	On
SPI	SYSCLK	On/Off	On/Off	Off	Off	Off
PWM	XCLK	On/Off	On/Off	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	0x0
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
0x2C	ASBBridgeCtrl	ASB Bridge Control Register	R/W	1	0x0

PLL Controller

Overview

Figure 9 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. 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.

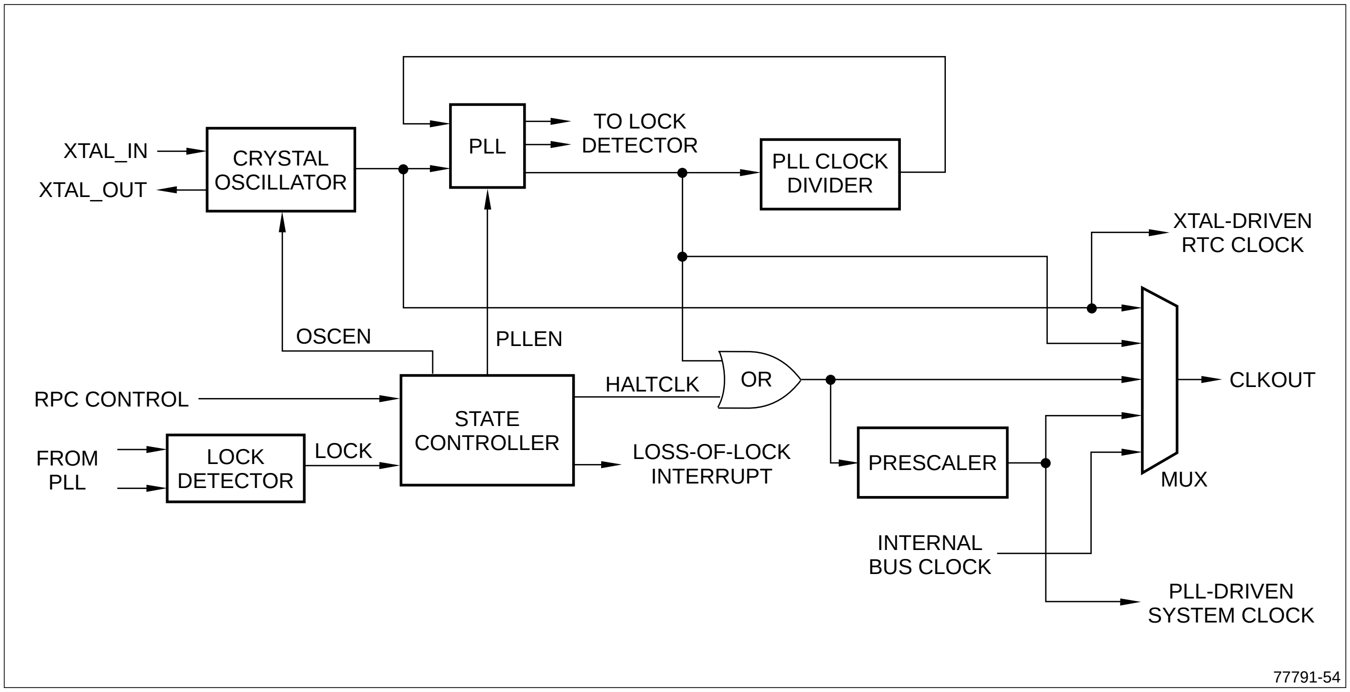


Figure 9. PLL Controller block diagram

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
- DMTN Gray-scale format:
  - 8 levels of gray scale
  - For monochrome Transflective DMTN-LCD Panel
- 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
- HR-TFT Active Color format:
  - Direct Interface to HR-TFT
  - 16 bits per pixel driving an 18-bit-wide panel bus
  - Normal or Reverse Scan in horizontal or vertical directions
  - For High-Reflectivity TFT Panel

Overview

The LH79531 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 a dedicated 32-bit internal LCD bus directly from the SDRAM 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, including a direct interface to HR-TFT. 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 LCD Bus Slave interface. The parameters for controlling DMTN and HR-TFT panel formats are accessed through the Peripheral Bus interface. Once programmed, the LCD Bus Master (LCD-DMA interface) automatically fetches graphics data from the designated frame buffer region of external memory. These activities are synchronized to the LH79531 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.

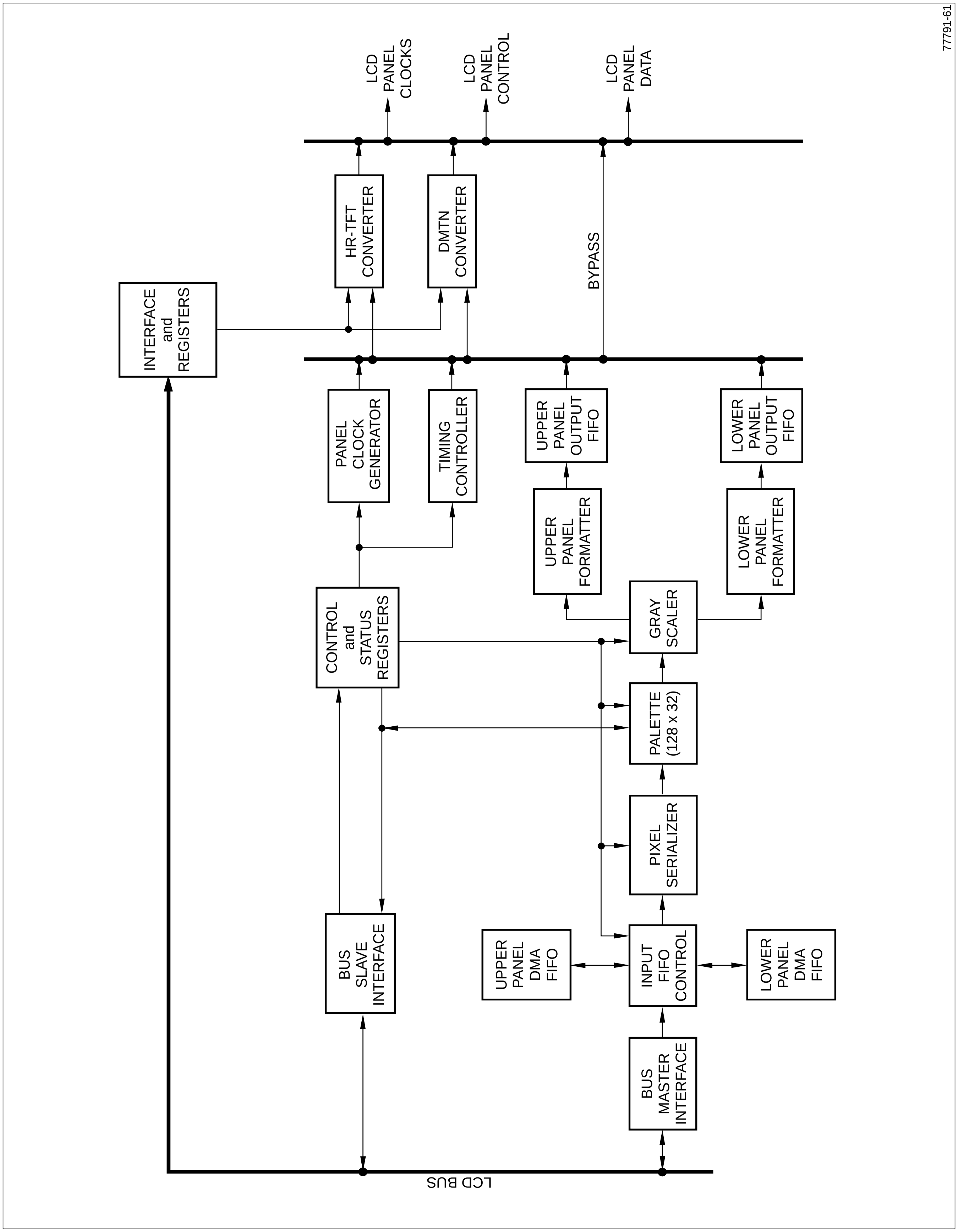


Figure 10. LCD Controller Block Diagram

Register Summary

Table 18 and Table 19 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 LCD bus interface. Registers that control the DMTN and HR-TFT formats are offset from a different base address; they are physically accessed through the Peripheral 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				

Table 19. DMTN and HR-TFT Panel Format Registers (Base Address: 0xFFFF2800)

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	Setup	R/W	0x000C	16	LCD Interface Peripheral Setup CR: Conversion mode (00=Bypass, 01=HR-TFT, 10=DMTN, 11=Reserved) HRVE: Horiz Reverse Scan VRVE: Vert Reverse Scan	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PPL: Pixels per line									VRVE	HRVE	CR	
0x004	Control	R/W	0x00	8	LCD Interface Periph Control SPSEN: asserts SPSEN output CLSEN: asserts CLSEN output UBLEN: asserts UBLEN output DISP: Asserts DISP output EN0-3: General purpose output-Enables	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EN3	EN2	EN1	EN0	DISP	UBLEN	CLSEN	SPSEN
0x008	Timing1	R/W	0x0000	16	HR-TFT Timing register 1 LPDel: Hsync-LP time delay REVDel: Hsync-REV delay PSDel/CLSDel: Hsync-PS/CLS Delay	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PSDel / CLSDel				REV Del				LPDel			
0x00C	Timing2	R/W	0x0000	16	HR-TFT Timing register 2 PSDel2/CLSdel2: SPL-CLS and SPR-PS time delay SPLValue: SPL and SPR delays During vert front & back porches	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SPLValue							PSDel2/CLSDel2								

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 20, Table 21, and Table 22:

Table 20. 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 21. 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 22. 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 the DMTN/HR-TFT SETUP register.
3. Program the DMTN/HR-TFT Timing1 and Timing2 registers.
4. Program all other LCD Controller registers.
5. Enable the LCD Controller (LCDControl bit EN = ’1’).

Table 23. LCD Panel Output Signal Multiplexing

Panel Type		TFT		STN		DMTN		HR-TFT	
TFT Bit of LCD Control Register		1		0		1		1	
CR [1:0] Bits of Setup Register		00 (Bypass Mode)				10		01	
Pin No.	Pin Name	Function	Driver	Function	Driver	Function	Driver	Function	Driver
157	frm_clk	Vsync	output	FRAME	output	Vsync	output	SPS / SPSEN	3-state
158	vbl_enb	EN1	output	EN1	output	EN1	output	SPR / SPREN	3-state
159	line_clk	Hsync	output	LINEsync	output	Hsync	output	LP	output
160	dspl_enb	DISP	output	DISP	output	DISP	output	REV	output
161	shift_clk	LCDCLK	output	LCDCLK	output	LCDCLK	output	LCDCLK	output
164	vdd_enb	POWER ENABLE	output	POWER ENABLE	output	POWER ENABLE	output	POWER ENABLE	output
165	vee_enb	EN0	output	EN0	output	EN0	output	LBR	output
166	enab	DATA ENABLE	output	ACBiasDrive	output	DATA ENABLE	output	SPL / SPLEN	3-state
169[1]	lcd_ps	EN3	output	EN3	output	EN3	output	PS	output
170[1]	lcd_ubl	LINE END CLLE	output	LINE END CLLE	output	LINE END CLLE	output	UBL / UBLEN	3-state
172[1]	lcd_cls	EN2	output	EN2	output	EN2	output	CLS / CLSEN	3-state

NOTE: [1] Signals on pins 169-172 are function-3. All others are function-1.

Table 24. LCD Panel Output Data Multiplexing

Ext. Pin	4bit Mono		8bit Mono		8bit Color		18bit Color TFT and HR-TFT	12bit Gray DMTN
	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]	1st msb D32
vd10	-	MLSTN[1]	-	MLSTN[5]	-	CLSTN[5]	GREEN[3]	1st data D31
vd9	-	MLSTN[2]	-	MLSTN[6]	-	CLSTN[6]	GREEN[2]	1st lsb D30
vd8	-	MLSTN[3]	-	MLSTN[7]	-	CLSTN[7]	GREEN[1]	2nd msb D22
vd7	-	-	MUSTN[0]	MUSTN[0]	CUSTN[0]	CUSTN[0]	GREEN[0]	2nd data D21
vd6	-	-	MUSTN[1]	MUSTN[1]	CUSTN[1]	CUSTN[1]	Intensity bit	2nd lsb D20
vd5	-	-	MUSTN[2]	MUSTN[2]	CUSTN[2]	CUSTN[2]	RED[4]	3rd msb D12
vd4	-	-	MUSTN[3]	MUSTN[3]	CUSTN[3]	CUSTN[3]	RED[3]	3rd data D11
vd3	MUSTN[0]	MUSTN[0]	MUSTN[4]	MUSTN[4]	CUSTN[4]	CUSTN[4]	RED[2]	3rd lsb D10
vd2	MUSTN[1]	MUSTN[1]	MUSTN[5]	MUSTN[5]	CUSTN[5]	CUSTN[5]	RED[1]	4th msb D02
vd1	MUSTN[2]	MUSTN[2]	MUSTN[6]	MUSTN[6]	CUSTN[6]	CUSTN[6]	RED[0]	4th data D01
vd0	MUSTN[3]	MUSTN[3]	MUSTN[7]	MUSTN[7]	CUSTN[7]	CUSTN[7]	Intensity bit	4th lsb D00

NOTE: -: Reserved.

USB Device

Features

The USB Device provides modulation/demodulation and control circuitry for Universal Serial Bus (USB) communication devices at a full speed of 12 Mbps. The USB controller is fully compatible with USB1.1.

The USB controller allows the following communication:

Table 25. USB Device Features

ITEM	Description
Transmission rate	Full speed (12 Mbps)
Endpoints	4:1 control transfer 2 bulk transfers (In and Out) 1 interrupt transfer
Host function	Not Supported
Hub function	Not Supported

Endpoint 0 is defined as a control transfer endpoint and therefore it supports bi-directional data transfers. Its FIFOs are eight bytes deep. Endpoints 1 and 2 are both defined as bulk transfer endpoints with endpoint 1 configured to send data and endpoint 2 configured to receive data. Their FIFOs are 64 bytes deep. Endpoint 3 is defined as an interrupt transfer endpoint and therefore supports the reception of data only. Its FIFO is eight bytes deep.

The USB device provides an interrupt to the CPU. It will be triggered under the following conditions:

- A STALL handshake issued by the USB controller to the USB host for any of the four endpoints.
- A ‘device request’ received by the USB controller from the USB host.
- An SOF (Start Of Frame) received by the USB controller from the USB host.
- A ‘suspend request’ received by the USB controller from the USB host.
- A ‘reset request’ received by the USB controller from the USB host.
- A ‘resume from suspend’ received by the USB controller from the USB host.
- A ‘less than maximum-size’ packet received by the USB host.

Register Summary

The base address for USB Device registers is 0xFFFF0400**.** The address, initial value, access, and number of bits for each register are outlined in Table 26:

Table 26. USB Module Register Map (Base Address: 0xFFFF0400)

Offset from Base	Register	Name	Access	Size	Reset Value
0x00	USBINTRDA	Interrupt read register bank A	R	8	0x00
	USBINTRSTA	Interrupt write register bank A	W	8	0x00
0x04	USBINTRDB	Interrupt read register bank B	R	8	0x00
	USBINTRSTB	Interrupt write register bank B	W	8	0x00
0x08	USBFRML	Frame number register – Low	R	8	0x00
	USBPRMDATA	Parameter register	W	8	0x00
0x0C	USBFRMH	Frame number register – High	R	3	0x0
	USBEPT0I	IN FIFO 0 data write register	W	8	Undefined
0x10	USBREQDATA	Device request data register	R	8	Undefined
	USBEPT1I	IN FIFO 1 data write register	W	8	Undefined
0x14	USBEPT3I	IN FIFO 3 data write register	W	8	Undefined
0x18	USBEPTSTR	Endpoint status read register	R	4	0x0
	USBEPTSTW	Endpoint status write register	W	8	0x00
0x1C	USBCTRL1	Control register 1	R/W	8	0x00
0x20	USBCTRL2	Control register 2	R/W	8	0x00
0x24	USBFIFOST	FIFO status register	R	8	0x31
	USBEPT0RES	FIFO reset register [7:2] / IN FIFO 0 response register [1:0]	W	8	0x02
0x28	USBEPT0O	OUT FIFO 0 data read register	R	8	Undefined
	USBEPT1RES	IN FIFO 1 response register	W	2	0x02
0x2C	USBEPT2O	OUT FIFO 2 data read register	R	8	Undefined
	USBEPT3RES	IN FIFO 3 response register	W	2	0x02
0x40	USBWNUM	USB wait count register	R/W	5	0x00
0x44	USBSRST	USB software reset register	R/W	1	0x1

Synchronous Serial Interface

(Compatible with SPI, µWire, TI Synchronous Serial Standard)

The Synchronous Serial Port controller provides a serial communications interface that is compatible with several common synchronous protocols; including Serial Peripheral Interface (SPI), National µwire, and TI Synchronous Serial Standards. This allows the LH79531 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
- Programmable clock bit rate and prescaler
- Separate transmit and receive FIFOs
  - 16 bits wide
  - 8 locations deep
- Programmable choice of interface operation:
  - Motorola SPI
  - National µwire
  - TI synchronous serial
- 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 SSI controller is located at a base address of 0xFFFF5400.

Table 27. SSI 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	7	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 LH79531 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 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 LH79531 has three UARTs that can be individually addressed. The registers are shown in Table 28.

Register Summary

Table 28. 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	3	0x0
0x018	UART0FLG	UART0 Flag Register	R/W	6	0b010xxx
0x01C	UART0MCR	UART0 MODEM Control Register	R/W	2	0x00
0x020	UART0INTMSK	UART0 Interrupt Mask Register	R/W	4	0x0
0x024	UART0STAT	UART0 Interrupt Raw Status Register	R	4	Undefined
0x028	UART0INTR	UART0 Interrupt Status Register	R	4	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	3	0x0
0x418	UART1FLG	UART1 Flag Register	R/W	6	0b010xxx
0x41C	UART1MCR	UART1 MODEM Control Register	R/W	2	0x00
0x420	UART1INTMSK	UART1 Interrupt Mask Register	R/W	4	0x0
0x424	UART1STAT	UART1 Interrupt Raw Status Register	R	4	Undefined
0x428	UART1INTR	UART1 Interrupt Status Register	R	4	Undefined
0x800	UART2DR	UART2 Data Register	R/W	8	Undefined
0x804	UART2RXSTAT	UART2 Receive Status Register	R	3	Undefined
0x804	UART2MSEOI	UART2 Clear MODEM status change interrupt	W	3	Undefined
0x808	H_U2BRLCR	UART2 High byte Baud Rate/Line Control Register	R/W	7	0x00
0x80C	M_U2BRLCR	UART2 Middle byte Baud Rate/Line Control Register	R/W	8	0x00
0x810	L_U2BRLCR	UART2 Lower byte Baud Rate/Line Control Register	R/W	8	0x00
0x814	UART2CON	UART2 Control Register	R/W	3	0x0
0x818	UART2FLG	UART2 Flag Register	R/W	6	0b010xxx
0x81C	UART2MCR	UART2 MODEM Control Register	R/W	2	0x00
0x820	UART2INTMSK	UART2 Interrupt Mask Register	R/W	4	0x0
0x824	UART2STAT	UART2 Interrupt Raw Status Register	R	4	Undefined
0x828	UART2INTR	UART2 Interrupt Status Register	R	4	Undefined

NOTE: Because there are 3 UARTs to be addressed, Register abbreviations take the form of:
UART(n) register where n = 0, 1, or 2.
‘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

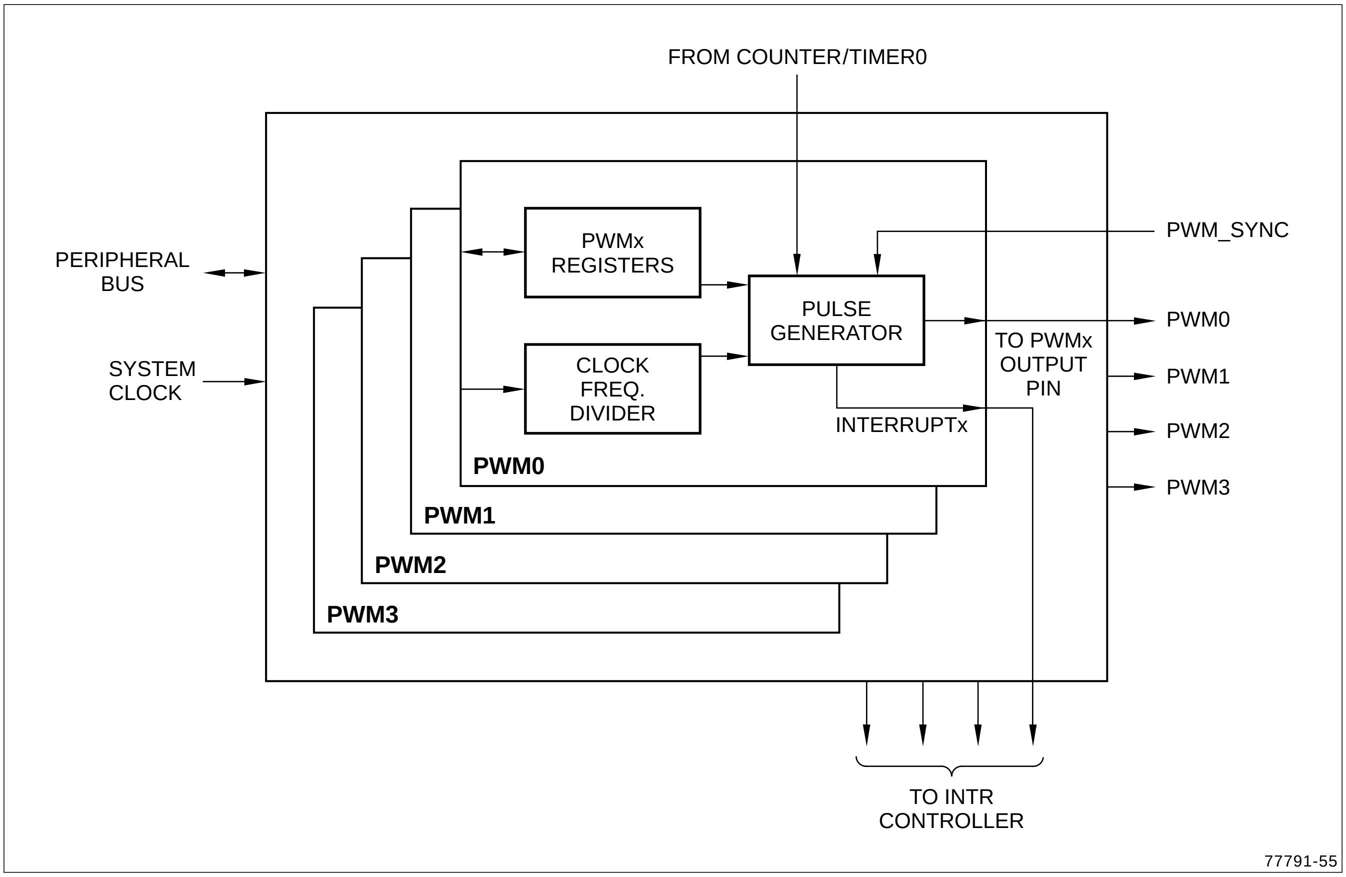
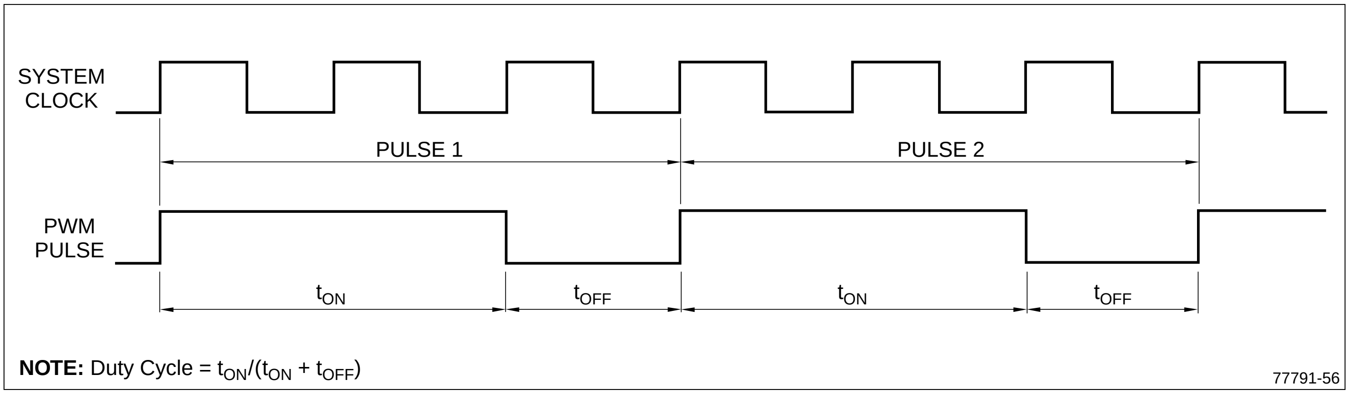


Figure 11. PWM Block Diagram

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 12 shows an example of a PWM output signal.



NOTE: Duty Cycle = $t_{ON}/(t_{ON} + t_{OFF})$

Figure 12. 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 PWMx 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 29. 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

Infrared Controller

The LH79531 supports three IR interface standards.

- SHARP DASK 9600 bps – 57.6 Kbps
- IrDA SIR 9600 bps – 115.2 Kbps
- IrDA SIR 1.1 – High Speed Extension 4.0 Mbps (IrDA 4M)

Overview

The LH79531 IR controller supports SHARP DASK, IrDA SIR and IrDA 4M modes of operation and contains a separate modem for each; however only one can be active at a time (CR9, CR10). The core of the IR controller is based on the UIRCC building block with several interfaces.

IrDA 4M is supported by a stand alone IR Interface. The IR Interface block contains all the necessary circuitry including the parallel/serial data converter, address recognition (CR8 and CR3), CRC calculation (CR3) and FIFOs. When in IrDA 4M mode, the IrDA controller can use the DMA block for service by using the hardware trigger of the DMA block. Single and burst DMA cycles are supported. If DMA is not used, the processor can poll SR0 to determine if the IrDA requires data to be fetched or provided.

IrDA SIR and SHARP DASK are supported via UART2. The IR Controller provides only the modulation and demodulation of the signal. The parallel-to-serial and serial-to-parallel conversion of the transmitted and received data, respectively, is carried out by UART2 as in standard (wired) serial communication. DMA and interrupt capabilities are also provided by UART2. Serial data is also sent through UART2. Serial data and clock are received.

Optional control of the IR I/O signals, including SIP/SIR transmission is available (CR9, CR11, CR15).

A general-purpose 16-bit timer is provided within the IR Interface block that can be used by software for protocol timing functions. The timer does not affect the operation of the IR communication (CR11, CR2, CR0, SR12, and SR13). Note that the timer will assert the IrDA interrupt when it transitions from 0x0000 to 0xFFFF, if enabled.

The IrDA controller may interrupt the operation of the LH79531 depending upon the setting of the interrupt mask bits (CR2, CR15). The detailed bit definitions for the SR3 register outline the types of errors that the IrDA controller can detect. Note that there is only one IrDA interrupt. The processor must poll the appropriate SR registers following the interrupt to determine its cause. IrDA interrupts are prioritized as follows:

Interrupt due to timer > Interrupt due to receiving > Interrupt due to transmission

In addition to the LH79531 hardware reset, the IrDA Controller supports several types of local software reset (CR0). UIRCC reset performs the same as the hardware reset on the IrDA registers without affecting the rest of the LH79531. The transmit and reset registers reset a number of portions of the controller as shown in the Register Bit Definition Summary.

All data, status or control transactions with the IrDA Interface block should be 16 bits wide. If an odd number of bytes are to be transferred, they must be in the most significant byte of the 16-bit transfer.

IrDA Interface Registers

Table 30. IrDA Interface Registers (Base Address: 0xFFFF0000)

Offset from Base	Register	Name	Access	Size	Reset Value
0x080	IrDAWCNT	IR Wait Count Register	R/W	3	0x00
0x084	IrDAMCLKSEL	IR MCLK Frequency Select Register	R/W	2	0x0

UIRCC-Type Core Registers

The UIRCC core contains 8-bit registers with a 16-bit bus interface. All access to this block can be made via 16-bit access. Therefore, two 8-bit registers are accessed simultaneously. For backward compatibility, the registers are described as pairs of 8-bit registers. For example, the 8-bit registers CR5 and CR4 are written simultaneously through the 16-bit register CR5_4.

There are 16 x 8 bit control registers (CR0 – CR15) and 16 x 8 bit status registers (SR0 – SR15) mapped onto the same address space. The control registers are writing only and the status registers are read only. They are accessed as shown in Table 31.
Note that some of the registers may also be written to or read a byte at a time through the alternate addresses shown in Table 31.

Table 31. Mapping of Control Registers for 16-bit or 8-bit Access (Base Address: 0xFFFF0000)

Offset from Base (16-bit Register)	16-Bit Write Function	16-Bit Read Function	Offset from Base (8-bit Register)	8-bit Write	8-bit Read
0x40	Write to register CR1_0	Read of register SR1_0	0x00	CR0	SR0
			0x04	CR1	SR1
0x48	Write to register CR3_2	Read of register SR3_2	0x08	CR2	SR2
			0x0C	CR3	SR3
0x50	Write to register CR5_4	Read of register SR5_4	0x10	-	-
			0x14	-	-
0x58	Write to register CR7_6	Read of register SR7_6	0x18	-	-
			0x1C	-	-
0x60	Write to register CR9_8	Read of register SR9_8	0x20	CR8	SR8
			0x24	CR9	SR9
0x68	Write to register CR11_10	Read of register SR11_10	0x28	CR10	SR10
			0x2C	CR11	SR11
0x70	Write to register CR13_12	Read of register SR13_12	0x30	-	-
			0x34	-	-
0x78	Write to register CR15_14	Read of resister SR15_14	0x38	CR14	-
			0x3C	CR15	SR15

NOTE: -: Not available for 8-bit access. Use 16-bit registers.

Counter / Timer

Three general purposes Counter/Timers (CT) are included in the LH79531. Each Counter/Timer is 16 bits wide and can be used as a periodic timer, frequency generator, etc. Of the three, only CT2 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- or 48-bit counter. This is controlled by the Carry bit in the control register. The 32 bit counter can be made from CT0 and CT1. Cascading the 3 counters makes the 48 bit counter. The cascaded counters must have the same configuration.
After reset, the operations which change the prescale bits of CT0/CT1/CT2 to different value or select external clock for CT2 may cause illegal operation for the cascaded 48 bit counter.

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 32. 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

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 LH79531 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 LH79531) 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 WDCTRL. 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 41.95 seconds with a system clock of 60 MHz.

When the reset option is secected, the time-out causes an 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,

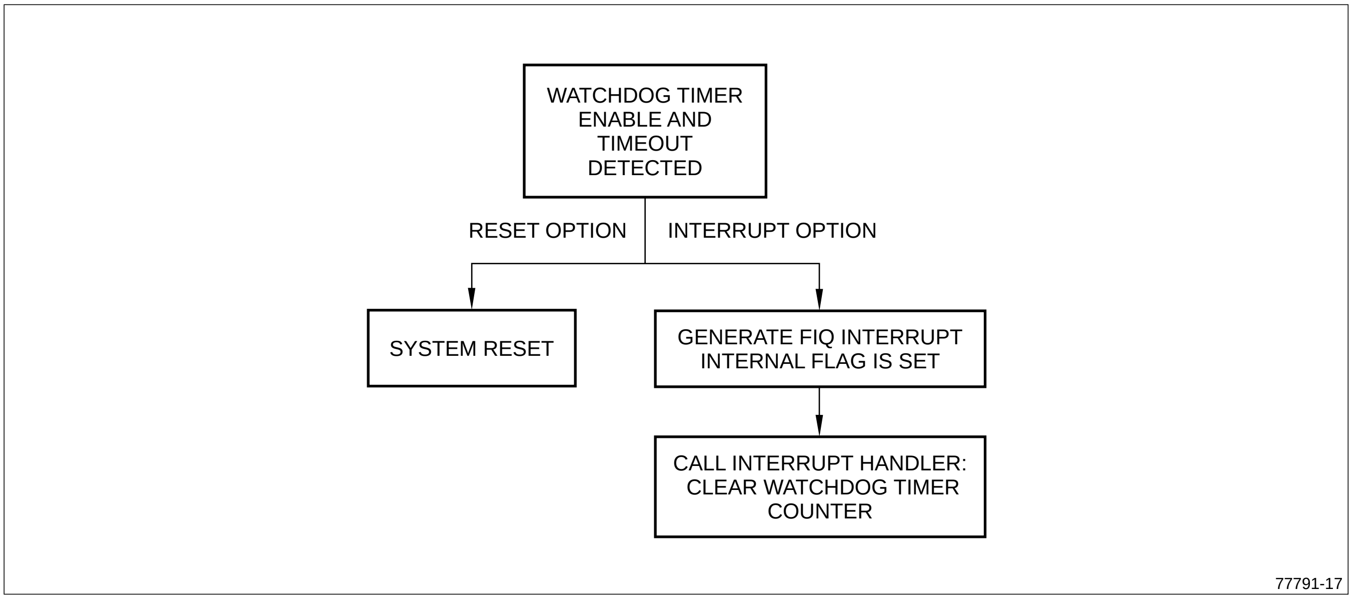

WATCHDOG TIMER ENABLE AND TIMEOUT DETECTED
RESET OPTION
INTERRUPT OPTION
SYSTEM RESET
GENERATE FIQ INTERRUPT INTERNAL FLAG IS SET
CALL INTERRUPT HANDLER: CLEAR WATCHDOG TIMER COUNTER
77791-17

Figure 13. 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 33. 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	WDCTRL	R/W	0x00	7	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 74-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).

NOTE: Not all macro functions are available at the same time.

Features

- Five PIO ports(PIOA-E)
  - PIOA (8-bit)
  - PIOB (15-bit)
  - PIOC (27-bit)
  - PIOD (18-bit)
  - PIOE (6-bit)
  - PIOF (10-bit) - For Peripheral Multiplex control functions only.
- PIO control registers allow multiplexing of pins for different functions.

Register Summary

A summary of the PIO registers are shown in Table 34.

Table 34. PIO Register (Base Address: 0xFFFF4C00)

Offset from Base	Register	Name	Access	Size	Reset Value
0x00	PIOAData	PA Data Register	R/W	8	0x00000000
0x04	PIOACtrl	PA Control Register	R/W	8	0x00000000
0x08	PIOADataSet	PA Data Set Register	W	8	0x00000000
0x0C	PIOADataReset	PA Data Reset Register	W	8	0x00000000
0x10	PIOAMuxControl	PA Mux Control Register	R/W	1	0x00000000
0x20	PIOBData	PB Data Register	R/W	15	0x00000000
0x24	PIOBCtrl	PB Control Register	R/W	15	0x00000000
0x28	PIOBDataSet	PB Data Set Register	W	15	0x00000000
0x2C	PIOBDataReset	PB Data Reset Register	W	15	0x00000000
0x30	PIOBMuxControl1	PB Mux Control Reg.1	R/W	15	0x00000000
0x34	PIOBMuxControl2	PB Mux Control Reg.2	R/W	14	0x00000000
0x40	PIOCData	PC Data Register	R/W	27	0x00000000
0x44	PIOCCtrl	PC Control Register	R/W	27	0x00000000
0x48	PIOCDataSet	PC Data Set Register	W	27	0x00000000
0x4C	PIOCDataReset	PC Data Reset Register	W	27	0x00000000
0x50	PIOCMuxControl1	PC Mux Control Reg.1	R/W	27	0x00000000
0x60	PIODData	PD Data Register	R/W	18	0x00000000
0x64	PIODCtrl	PD Control Register	R/W	18	0x00000000
0x68	PIODDataSet	PD Data Set Register	W	18	0x00000000
0x6C	PIODDataReset	PD Data Reset Register	W	18	0x00000000
0x70	PIODMuxControl1	PD Mux Control Reg.1	R/W	3	0x00000000
0x80	PIOEData	PE Data Register	R/W	6	0x00000000
0x84	PIOECtrl	PE Control Register	R/W	6	0x00000000
0x88	PIOEDataSet	PE Data Set Register	W	6	0x00000000
0x8C	PIOEDataReset	PE Data Reset Register	W	6	0x00000000
0x90	PIOEMuxControl	PE Mux Control Register	R/W	6	0x00000000
0xB0	PIOFMuxControl1	PF Mux Control Register 1	R/W	10	0x00000000
0xB4	PIOFMuxControl2	PF Mux Control Register 2	R/W	8	0x00000000

Multiplexed I/O

There are 5 ‘banks’ of PIO, or Programmable I/O. These are labeled PIOA through PIOE. The map for the multiplexed I/O is shown below. Note that the MUX1 and MUX2 columns refer to the PIOxMuxControlReg1 and PIOxMuxControlReg2 of the appropriate bank of registers.

Multiplexed Functions												
Bit Index	**Pin**	**Func1**	**Macro**	**Dir**	**Func2**	**Macro**	**Dir**	**MUX1**	**Func3**	**Macro**	**Dir**	**MUX2**
PIOA, BIGEND = ‘0’												
0	205	RD[8]	SRAM	B	PIO[0]	PIO	B	0				
1	204	RD[9]	SRAM	B	PIO[1]	PIO	B	0				
2	203	RD[10]	SRAM	B	PIO[2]	PIO	B	0				
3	202	RD[11]	SRAM	B	PIO[3]	PIO	B	0				
4	200	RD[12]	SRAM	B	PIO[4]	PIO	B	0				
5	199	RD[13]	SRAM	B	PIO[5]	PIO	B	0				
6	198	RD[14]	EBI	B	PIO[6]	PIO	B	0				
7	197	RD[15]	EBI	B	PIO[7]	PIO	B	0				
PIOA, BIGEND = ‘1’												
0	215	RD[0]	EBI	B	PIO[0]	PIO	B	0				
1	214	RD[1]	EBI	B	PIO[1]	PIO	B	0				
2	213	RD[2]	EBI	B	PIO[2]	PIO	B	0				
3	212	RD[3]	EBI	B	PIO[3]	PIO	B	0				
4	210	RD[4]	EBI	B	PIO[4]	PIO	B	0				
5	209	RD[5]	EBI	B	PIO[5]	PIO	B	0				
6	208	RD[6]	EBI	B	PIO[6]	PIO	B	0				
7	207	RD[7]	EBI	B	PIO[7]	PIO	B	0				
PIOB												
0	193	PIO[8]	PIO	B	SPI_OUT	SPI	O	0	IrRXA	IrDA	I	0
1	192	PIO[9]	PIO	B	SPI_IN	SPI	I	1	UCLK	UART	I	1
2	190	PIO[10]	PIO	B	INTR0	INT	I	2	nRIO	UART	I	2
3	189	PIO[11]	PIO	B	INTR1	INT	I	3	ALM	RTC	O	3
4	188	PIO[12]	PIO	B	INTR2	INT	I	4	PWM_SYNC	PWM	I	4
5	187	PIO[13]	PIO	B	INTR3	INT	I	5	USB_FIR_CLK	USB	I	5
6	178	PIO[14]	PIO	B	DEOT0	DMA	O	6	USB_VMO	USB	O	6
7	177	PIO[15]	PIO	B	nDACK[0]	DMA	O	7	USB_VPO	USB	O	7
8	175	PIO[16]	PIO	B	DREQ0	DMA	I	8	USB_VM	USB	I	8
9	174	PIO[17]	PIO	B	DEOT1	DMA	O	9	USB_VP	USB	I	9
10	173	PIO[18]	PIO	B	nDACK[1]	DMA	O	10	USB_REV	USB	I	10
11	172	PIO[19]	PIO	B	DREQ1	DMA	I	11	LCD_CLS	LCDC	O	11
12	170	PIO[20]	PIO	B	nWAIT	EXT	I	12	LCD_UBL	LCDC	O	12
13	169	PIO[21]	PIO	B	CLKOUT	RPC	O	13	LCD_PS	LCDC	O	13
14	168	PIO[22]	PIO	B	XCLKEN	RPC	O	14				
PIOC												
0	166	ENAB	LCDC	O	PIO[23]	PIO	B	0				
1	165	VEE_ENB	LCDC	O	PIO[24]	PIO	B	1				
2	164	VDD_ENB	LCDC	O	PIO[25]	PIO	B	2				
3	163	MCLK	LCDC	I	PIO[26]	PIO	B	3				
4	161	SHIFT_CLK	LCDC	O	PIO[27]	PIO	B	4				
5	160	DSPL_ENB	LCDC	O	PIO[28]	PIO	B	5				
6	159	LINE_CLK	LCDC	O	PIO[29]	PIO	B	6				
7	158	VBL_ENB	LCDC	O	PIO[30]	PIO	B	7				
8	157	FRM_CLK	LCDC	O	PIO[31]	PIO	B	8				
9	155	VD[0]	LCDC	O	PIO[32]	PIO	B	9				
10	154	VD[1]	LCDC	O	PIO[33]	PIO	B	10				
11	153	VD[2]	LCDC	O	PIO[34]	PIO	B	11				

Multiplexed Functions												
Bit Index	**Pin**	**Func1**	**Macro**	**Dir**	**Func2**	**Macro**	**Dir**	**MUX1**	**Func3**	**Macro**	**Dir**	**MUX2**
12	152	VD[3]	LCDC	O	PIO[35]	PIO	B	12				
13	150	VD[4]	LCDC	O	PIO[36]	PIO	B	13				
14	149	VD[5]	LCDC	O	PIO[37]	PIO	B	14				
15	148	VD[6]	LCDC	O	PIO[38]	PIO	B	15				
16	147	VD[7]	LCDC	O	PIO[39]	PIO	B	16				
17	146	VD[8]	LCDC	O	PIO[40]	PIO	B	17				
18	144	VD[9]	LCDC	O	PIO[41]	PIO	B	18				
19	143	VD[10]	LCDC	O	PIO[42]	PIO	B	19				
20	142	VD[11]	LCDC	O	PIO[43]	PIO	B	20				
21	141	VD[12]	LCDC	O	PIO[44]	PIO	B	21				
22	140	VD[13]	LCDC	O	PIO[45]	PIO	B	22				
23	138	VD[14]	LCDC	O	PIO[46]	PIO	B	23				
24	137	VD[15]	LCDC	O	PIO[47]	PIO	B	24				
25	136	VD[16]	LCDC	O	PIO[48]	PIO	B	25				
26	135	VD[17]	LCDC	O	PIO[49]	PIO	B	26				
PIOD												
0	102	DQM[2]	SDRAM	O	PIO[50]	PIO	B	0				
1	101	DQM[3]	SDRAM	O	PIO[51]	PIO	B	1				
2	73	SDD[16]	SDRAM	B	PIO[52]	PIO	B	2				
3	72	SDD[17]	SDRAM	B	PIO[53]	PIO	B	2				
4	71	SDD[18]	SDRAM	B	PIO[54]	PIO	B	2				
5	69	SDD[19]	SDRAM	B	PIO[55]	PIO	B	2				
6	68	SDD[20]	SDRAM	B	PIO[56]	PIO	B	2				
7	67	SDD[21]	SDRAM	B	PIO[57]	PIO	B	2				
8	66	SDD[22]	SDRAM	B	PIO[58]	PIO	B	2				
9	65	SDD[23]	SDRAM	B	PIO[59]	PIO	B	2				
10	63	SDD[24]	SDRAM	B	PIO[60]	PIO	B	2				
11	62	SDD[25]	SDRAM	B	PIO[61]	PIO	B	2				
12	61	SDD[26]	SDRAM	B	PIO[62]	PIO	B	2				
13	60	SDD[27]	SDRAM	B	PIO[63]	PIO	B	2				
14	59	SDD[28]	SDRAM	B	PIO[64]	PIO	B	2				
15	57	SDD[29]	SDRAM	B	PIO[65]	PIO	B	2				
16	56	SDD[30]	SDRAM	B	PIO[66]	PIO	B	2				
17	55	SDD[31]	SDRAM	B	PIO[67]	PIO	B	2				
PIOE												
0	15	RA[20]	EBI	O	PIO[68]	PIO	B	0				
1	14	RA[21]	EBI	O	PIO[69]	PIO	B	1				
2	12	RA[22]	EBI	O	PIO[70]	PIO	B	2				
3	11	RA[23]	EBI	O	PIO[71]	PIO	B	3				
4	10	RA[24]	EBI	O	PIO[72]	PIO	B	4				
5	9	RA[25]	EBI	O	PIO[73]	PIO	B	5				
PIOF												
0	195	URXD2	UART2	I	SPI_CLK	SPI	B	0	IrRX2	IrDA	I	0
1	194	UTXD2	UART2	O	SPI_ENB	SPI	B	1	IrTXA	IrDA	O	1
2	185	URXD1	UART1	I	PWM0	PWM	O	2	INTR4	INTC	I	2
3	184	UTXD1	UART1	O	PWM1	PWM	O	3	INTR5	INTC	I	3
4	183	nURST0	UART0	O	PWM2	PWM	O	4	CTOUT0	CT0	O	4
5	182	nUCTS0	UART0	I	PWM3	PWM	O	5	CTCLK	CT2	I	5
6	180	URXD0	UART0	I	CTOUT1	CT1	O	6	USB_SUSPND	USB	O	6
7	179	UTXD0	UART0	O	CTOUT2	CT2	O	7	USB_OEB	USB	O	7
8	114	nCE[6]	EBI	O	SDRA[14]	SDRAM	O	8				
9	113	nCE[7]	EBI	O	nDCS[1]	SDRAM	O	9				

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

- 26 interrupt sources
  - 6 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 35.

Table 35. INTC programming Registers (Base Address: 0xFFFF7400)

Offset from Base	Register	Name	Access	Size	Reset Value
0x000	IRQStatus	IRQ Status Register	R	26	0x0000000
0x004	IRQRawStatus	IRQ Raw Status Register	R	26	0x0000000
0x008	IRQEnable	IRQ Enable Register	R	26	0x0000000
0x008	IRQEnableSet	IRQ Enable Set Register	W	26	Unknown
0x00C	IRQEnableClear	IRQ Enable Clear Register	W	26	Unknown
0x100	FIQStatus	FIQ Status Register	R	27	0x0000000
0x104	FIQRawStatus	FIQ Raw Status Register	R	27	0x0000000
0x108	FIQEnable	FIQ Enable Register	R	26	0x0000000
0x108	FIQEnableSet	FIQ Enable Set Register	W	26	Unknown
0x10C	FIQEnableClear	FIQ Enable Clear Register	W	26	Unknown
0x200	INTConfig0	Configuration Register 0	R/W	6	0x0000000
0x204	INTConfig1	Configuration Register 1	R/W	6	0x0000000
0x208	INTClear	Interrupt Clear Register	W	27	Unknown

Interrupt Channel Assignments

Interrupt channels are assigned as in Table 36: 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 36. Interrupt Channel Assignment

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

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 107,108,110,111.
3. This condition is applied 107,108,110,111 pins.

Recommended Operating Conditions

Parameter	Symbol	Rating	Unit	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 Specifications

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	Vi = 0 or Vddb-0.2V Output = open f_{OSC} = 31kHz, f_{PLL} = 50MHz		130		mA	1
Operating Power (Standby Mode)	Istb	Vi = 0 or Vddb-0.2V f_{OSC} = 31kHz, f_{PLL} = 50MHz		95		mA	2
Operating Power (Sleep Mode)	Islp	Vi = 0 or Vddb-0.2V f_{OSC} = 31kHz, f_{PLL} = 50MHz		10		mA	3
Operating Power (Stop Mode)	Istp	Vi = 0 or Vddb-0.2V f_{OSC} = 31kHz		30		uA	4
Operating Power (Stop2 Mode)	Istp2	Vi = 0 or Vddb-0.2V		10		uA	5

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, 135-138, 140-144, 146-150, 152-155, 157-160, 163-166, 168-170, 172-175, 177-180, 182-185, 187-190, 192-195, 197-200, 202-205, 207-210, 212-215)
7. Pin=(9-12, 14-17, 19-22, 24-27, 29-32, 34-37, 39-42, 44-46, 48-51, 55-57, 59-63, 65-69, 71-75, 77-80, 82-85, 87-90, 92-96, 98-102, 104-105, 113-114, 116-119, 121-124, 126-129, 131-132, 161)
8. Vdd=Vdd_xtl for Pin 110,111, Vddb for other Pins.
9. Vdd=Vdd_xtl for Pin 112, Vddb for other Pins.

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			17	ns	
Trah	Output Hold from XCLKIN ↓ to RA	5			ns	
Tced	Output Delay from XCLKIN ↓ to nCE			18	ns	
Tceh	Output Hold from XCLKIN ↓ to nCE	5			ns	
Tred	Output Delay from XCLKIN ↓ to nRE			18	ns	
Treh	Output Hold from XCLKIN ↓ to nRE	5			ns	
Twed	Output Delay from XCLKIN ↓ to nWE			18	ns	
Tweh	Output Hold from XCLKIN ↓ to nWE	5			ns	
Trdd	Output Delay from XCLKIN ↑ to RD			26	ns	
Trdh	Output Hold from XCLKIN ↑ to RD	5			ns	
Trdis	RD Setup time to XCLKIN ↓	0			ns	
Trdih	RD Hold time to XCLKIN ↓	7			ns	
Tscmd	Output Delay from SDCLK ↓ to SDRAMC signals			7	ns	
Tscmh	Output Hold from SDCLK ↓ to SDRAMC signals	0			ns	
Tsdrd	Output Delay from SDCLK ↓ to SDRA			7	ns	
Tsdrh	Output Hold from SDCLK ↓ to SDRA	0			ns	
Tsdqd	Output Delay from SDCLK ↓ to DQM			7	ns	
Tsdqh	Output Hold from SDCLK ↓ to DQM	0			ns	
Tsdd	Output Delay from SDCLK ↓ to SDD			7	ns	
Tsdh	Output Hold from SDCLK ↓ to SDD	0			ns	
Tsdds	SDD Setup time to SDCLK ↑	11			ns	
Tsddh	SDD Hold time to SDCLK ↑	-4			ns	
Tscked	Output Delay time from SDCLK ↓ to SDCKE	0		6	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

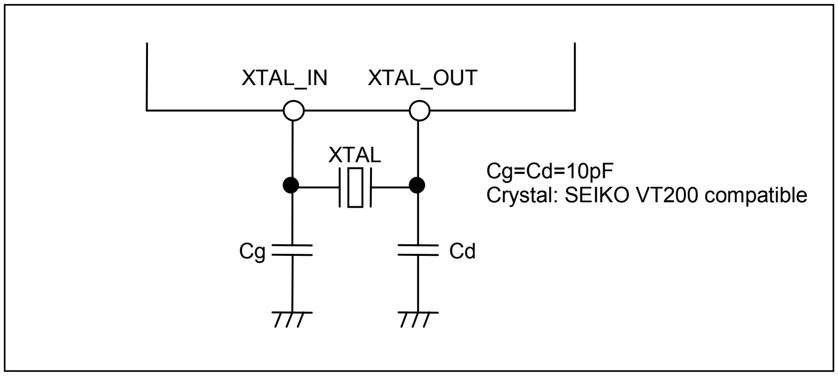


Figure 14. Recommended OSC circuit connection

Pin Descriptions

Table 37. LH79531 Pin Descriptions

Pin(S)	Name Function 1	Name Function 2	Name Function 3	Direction	Description
1	boot			Input	Selects startup mode memory width '0' = X8, '1' = X16
2	tsta			Input	This pin must be drive to LOW.
3	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.
4	tck			Input	JTAG Clock Input. This pin must be pulled up during normal operation.
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	Vddb				I/O Power
9	ra25			Output	Address Bus of EBI Controller.
		pio73		I/O	General Purpose I/O
10	ra24			Output	Address Bus of EBI Controller.
		pio72		I/O	General Purpose I/O
11	ra23			Output	Address Bus of EBI Controller.
		pio71		I/O	General Purpose I/O
12	ra22			Output	Address Bus of EBI Controller.
		pio70		I/O	General Purpose I/O
13	Vss				Ground
14	ra21			Output	Address Bus of EBI Controller.
		pio69		I/O	General Purpose I/O
15	ra20			Output	Address Bus of EBI Controller.
		pio68		I/O	General Purpose I/O
16	ra19			Output	Address Bus of EBI Controller.
17	ra18			Output	Address Bus of EBI Controller.
18	Vddc				Core Power
19	ra17			Output	Address Bus of EBI Controller.
20	ra16			Output	Address Bus of EBI Controller.
21	ra15			Output	Address Bus of EBI Controller.
22	ra14			Output	Address Bus of EBI Controller.
23	Vss				Ground
24	ra13			Output	Address Bus of EBI Controller.
25	ra12			Output	Address Bus of EBI Controller.
26	ra11			Output	Address Bus of EBI Controller.
27	ra10			Output	Address Bus of EBI Controller.
28	Vddb				I/O Power
29	ra9			Output	Address Bus of EBI Controller.
30	ra8			Output	Address Bus of EBI Controller.
31	ra7			Output	Address Bus of EBI Controller.
32	ra6			Output	Address Bus of EBI Controller.
33	Vss				Ground
34	ra5			Output	Address Bus of EBI Controller.
35	ra4			Output	Address Bus of EBI Controller.
36	ra3			Output	Address Bus of EBI Controller.
37	ra2			Output	Address Bus of EBI Controller.
38	Vddb				I/O Power
39	ra1			Output	Address Bus of EBI Controller.
40	ra0			Output	Address Bus of EBI Controller.
41	nwe1			Output	EBI Controller Write Enable Output.
42	nwe0			Output	EBI Controller Write Enable Output.
43	Vss				Ground
44	nre			Output	EBI Controller Read Enable Output.
45	nce5			Output	Chip Selects of EBI Controller.

Pin(S)	Name Function 1	Name Function 2	Name Function 3	Direction	Description
46	nce4			Output	Chip Selects of EBI Controller.
47	Vddc				Core Power
48	nce3			Output	Chip Selects of EBI Controller.
49	nce2			Output	Chip Selects of EBI Controller.
50	nce1			Output	Chip Selects of EBI Controller.
51	nce0			Output	Chip Selects of EBI Controller.
52	Vss				Ground
53	clksel			Input	Selects either internal PLL clock or the External clock as sysclock of the chip ‘0’ = PLL, ‘1’ = external clock
54	bigend			Input	ARM Byte Address Mode Selection. ‘0’ = Little Endian ‘1’ = Big Endian. This signal must not be changed during operation.
55	sdd31			I/O	SDRAM Data Input/Output
		pio67		I/O	General Purpose I/O
56	sdd30			I/O	SDRAM Data Input/Output
		pio66		I/O	General Purpose I/O
57	sdd29			I/O	SDRAM Data Input/Output
		pio65		I/O	General Purpose I/O
58	Vddb				I/O Power
59	sdd28			I/O	SDRAM Data Input/Output
		pio64		I/O	General Purpose I/O
60	sdd27			I/O	SDRAM Data Input/Output
		pio63		I/O	General Purpose I/O
61	sdd26			I/O	SDRAM Data Input/Output
		pio62		I/O	General Purpose I/O
62	sdd25			I/O	SDRAM Data Input/Output
		pio61		I/O	General Purpose I/O
63	sdd24			I/O	SDRAM Data Input/Output
		pio60		I/O	General Purpose I/O
64	Vss				Ground
65	sdd23			I/O	SDRAM Data Input/Output
		pio59		I/O	General Purpose I/O
66	sdd22			I/O	SDRAM Data Input/Output
		pio58		I/O	General Purpose I/O
67	sdd21			I/O	SDRAM Data Input/Output
		pio57		I/O	General Purpose I/O
68	sdd20			I/O	SDRAM Data Input/Output
		pio56		I/O	General Purpose I/O
69	sdd19			I/O	SDRAM Data Input/Output
		pio55		I/O	General Purpose I/O
70	Vddb				I/O Power
71	sdd18			I/O	SDRAM Data Input/Output
		pio54		I/O	General Purpose I/O
72	sdd17			I/O	SDRAM Data Input/Output
		pio53		I/O	General Purpose I/O
73	sdd16			I/O	SDRAM Data Input/Output
		pio52		I/O	General Purpose I/O
74	sdd15			I/O	SDRAM Data Input/Output
75	sdd14			I/O	SDRAM Data Input/Output
76	Vss				Ground
77	sdd13			I/O	SDRAM Data Input/Output
78	sdd12			I/O	SDRAM Data Input/Output
79	sdd11			I/O	SDRAM Data Input/Output
80	sdd10			I/O	SDRAM Data Input/Output
81	Vddc				Core Power
82	sdd9			I/O	SDRAM Data Input/Output

Pin(S)	Name Function 1	Name Function 2	Name Function 3	Direction	Description
83	sdd8			I/O	SDRAM Data Input/Output
84	sdd7			I/O	SDRAM Data Input/Output
85	sdd6			I/O	SDRAM Data Input/Output
86	Vss				Ground
87	sdd5			I/O	SDRAM Data Input/Output
88	sdd4			I/O	SDRAM Data Input/Output
89	sdd3			I/O	SDRAM Data Input/Output
90	sdd2			I/O	SDRAM Data Input/Output
91	Vddb				I/O Power
92	sdd1			I/O	SDRAM Data Input/Output
93	sdd0			I/O	SDRAM Data Input/Output
94	sdclk			Output	SDRAM Clock
95	sdcke			Output	SDRAM Clock Enable. ‘0’=deselected, ‘1’ = selected
96	ndcs0			Output	SDRAM Chip Select. ‘0’=selected, ‘1’ = deselected
97	Vss				Ground
98	ndrwe			Output	SDRAM Write Enable. ‘0’ = write, ‘1’ = read
99	nras			Output	SDRAM RAS
100	ncas			Output	SDRAM CAS
101	dqm3			Output	SDRAM Data Write Mask for handling Byte and Half-Word Transfers
		pio51		I/O	General Purpose I/O
102	dqm2			Output	SDRAM Data Write Mask for handling Byte and Half-Word Transfers
		pio50		I/O	General Purpose I/O
103	Vddb				I/O Power
104	dqm1			Output	SDRAM Data Write Mask for handling Byte and Half-Word Transfers
105	dqm0			Output	SDRAM Data Write Mask for handling Byte and Half-Word Transfers
106	Vdd_xtl				Power Supply for xtal buffer
107	xtal_in			Input	Crystal Clock Input
108	xtal_out			Output	Crystal Clock Output
109	Vss_xtl				Ground Supply for PLL, OSC, & RTC
110	rtc_clkin			Input	RTC Clock Input
111	nreseti			Input	Asynchronous Reset. Active low.
112	nreseto			Output	Reset Output. When low, indicates that the chip has successfully completed reset.
113	nce7			Output	Memory Chip Select
		ndcs1		Output	SDRAM Chip Select
114	nce6			Output	Memory Chip Select
		sdra14		Output	SDRAM Address Bus
115	Vddc				Core Power
116	sdra13			Output	SDRAM Address Bus
117	sdra12			Output	SDRAM Address Bus
118	sdra11			Output	SDRAM Address Bus
119	sdra10			Output	SDRAM Address Bus
120	Vss				Ground
121	sdra9			Output	SDRAM Address Bus
122	sdra8			Output	SDRAM Address Bus
123	sdra7			Output	SDRAM Address Bus
124	sdra6			Output	SDRAM Address Bus
125	Vddb				I/O Power
126	sdra5			Output	SDRAM Address Bus
127	sdra4			Output	SDRAM Address Bus
128	sdra3			Output	SDRAM Address Bus
129	sdra2			Output	SDRAM Address Bus
130	Vss				Ground
131	sdra1			Output	SDRAM Address Bus
132	sdra0			Output	SDRAM Address Bus
133	xclkin			Input	External Clock Input

Pin(S)	Name Function 1	Name Function 2	Name Function 3	Direction	Description
134	Vddb				I/O Power
135	vd17			Output	LCD Video Output
		pio49		I/O	General Purpose I/O
136	vd16			Output	LCD Video Output
		pio48		I/O	General Purpose I/O
137	vd15			Output	LCD Video Output
		pio47		I/O	General Purpose I/O
138	vd14			Output	LCD Video Output
		pio46		I/O	General Purpose I/O
139	Vss				Ground
140	vd13			Output	LCD Video Output
		pio45		I/O	General Purpose I/O
141	vd12			Output	LCD Video Output
		pio44		I/O	General Purpose I/O
142	vd11			Output	LCD Video Output
		pio43		I/O	General Purpose I/O
143	vd10			Output	LCD Video Output
		pio42		I/O	General Purpose I/O
144	vd9			Output	LCD Video Output
		pio41		I/O	General Purpose I/O
145	Vddb				I/O Power
146	vd8			Output	LCD Video Output
		pio40		I/O	General Purpose I/O
147	vd7			Output	LCD Video Output
		pio39		I/O	General Purpose I/O
148	vd6			Output	LCD Video Output
		pio38		I/O	General Purpose I/O
149	vd5			Output	LCD Video Output
		pio37		I/O	General Purpose I/O
150	vd4			Output	LCD Video Output
		pio36		I/O	General Purpose I/O
151	Vss				Ground
152	vd3			Output	LCD Video Output
		pio35		I/O	General Purpose I/O
153	vd2			Output	LCD Video Output
		pio34		I/O	General Purpose I/O
154	vd1			Output	LCD Video Output
		pio33		I/O	General Purpose I/O
155	vd0			Output	LCD Video Output
		pio32		I/O	General Purpose I/O
156	Vddb				I/O Power
157	frm_clk			Output	LCD Frame Clock / Vertical Sync
		pio31		I/O	General Purpose I/O
158	vbl_enb			Output	LCD Backlight Control. ‘0’ = Off, ‘1’ = On
		pio30		I/O	General Purpose I/O
159	line_clk			Output	LCD Line Clock / Horizontal Sync
		pio29		I/O	General Purpose I/O
160	dspl_enb			Output	LCD Display Control. ‘0’ = Off, ‘1’ = On
		pio28		I/O	General Purpose I/O
161	shift_clk			Output	LCD Pixel / Shift Clock
		pio27		I/O	General Purpose I/O
162	Vss				Ground
163	mclk			Input	External LCD Clock Input
		pio26		I/O	General Purpose I/O
164	vdd_enb			Output	LCD VDD Enable. ‘0’ = Off, ‘1’ = On

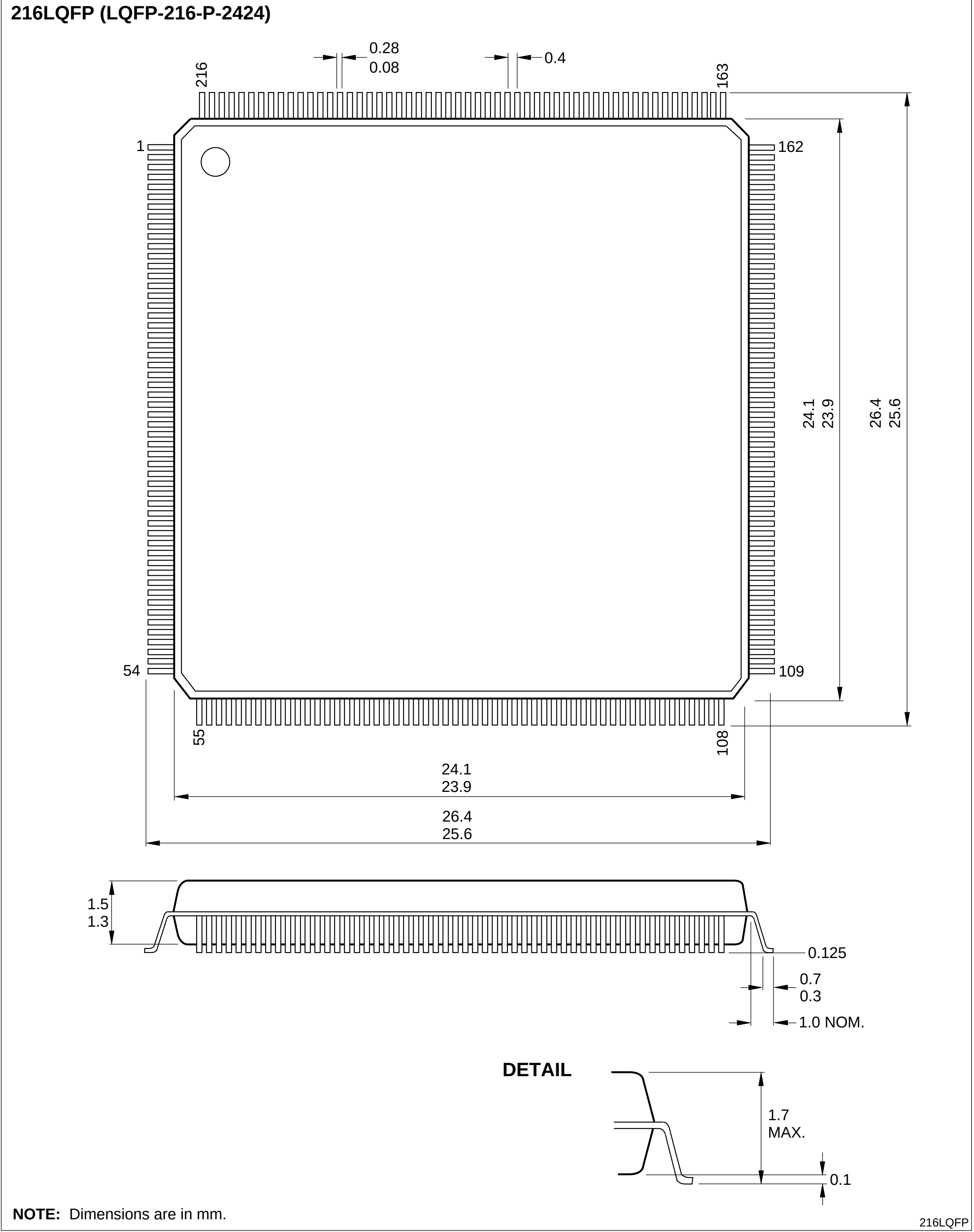
Pin(S)	Name Function 1	Name Function 2	Name Function 3	Direction	Description
		pio25		I/O	General Purpose I/O
165	vee_enb			Output	LCD VEE Enable. - = Off, ‘1’ = On
		pio24		I/O	General Purpose I/O
166	enab			Output	LCD Data Enable. ‘0’ = Disabled, ‘1’ = Enabled
		pio23		I/O	General Purpose I/O
167	Vddc				Core Power
168	pio22			I/O	General Purpose I/O IF CLKSEL = 0
		xclken		Output	This is the only function of this pin when CLKSEL = 1.
169	pio21			I/O	General Purpose I/O
		clkout		Output	System clock output.
			lcd_ps	Output	Power saving signal
170	pio20			I/O	General Purpose I/O
		nwait		Input	External Memory Wait. Facilitates the use of slow memories.
			lcd_ubl	Output	Selection for Vertical Scanning Direction. H = Normal, L = Inverted.
171	Vss				Ground
172	pio19			I/O	General Purpose I/O
		dreq1		Input	DMA Request Channel 1.
			lcd_cls	Output	LCD Clock Signal for Gate driver.
173	pio18			I/O	General Purpose I/O
		ndack1		Output	DMA Acknowledge Channel 1. Active LOW.
			usb_rev	Input	USB receive data
174	pio17			I/O	General Purpose I/O
		deot1		Output	DMA End of Transfer Channel 1.
			usb_vp	Input	USB VP input
175	pio16			I/O	General Purpose I/O
		dreq0		Input	DMA Request Channel 0.
			usb_vm	Input	USB VM input
176	Vddb				I/O Power
177	pio15			I/O	General Purpose I/O
		ndack0		Output	DMA Acknowledge Channel 0. Active LOW.
			usb_vpo	Output	USB VP output
178	pio14			I/O	General Purpose I/O
		deot0		Output	DMA End of transfer Channel 0.
			usb_vmo	Output	USB VM output
179	utxd0			Output	UART 0 Transmit Data
		ctout2		Output	Counter / Timer Output Signal
			usb_oeb	Output	USB output enable(Active low)
180	urxd0			Input	UART 0 Receive Data
		ctout1		Output	Counter / Timer Output Signal
			usb_subpnd	Output	USB suspend(Active high)
181	Vss				Ground
182	nucts0			Input	UART 0 clear to send
		pwm3		Output	Pulse Width Modulator Output Signal
			ctclk	Input	Counter / Timer Clock
183	nurts0			Output	UART 0 request to send
		pwm2		Output	Pulse Width Modulator Output Signal
			ctout0	Output	Counter / Timer Output Signal
184	utxd1			Output	UART 1 Transmit Data
		pwm1		Output	Pulse Width Modulator Output Signal
			intr5	Input	External Interrupt Signal
185	urxd1			Input	UART 1 Receive Data
		pwm0		Output	Pulse Width Modulator Output Signal
			intr4	Input	External Interrupt Signal

Pin(S)	Name Function 1	Name Function 2	Name Function 3	Direction	Description
186	Vddb				I/O Power
187	pio13			I/O	General Purpose I/O
		intr3		Input	External Interrupt Signal
			usb_fir_clk	Input	USB/IrDA Clock
188	pio12			I/O	General Purpose I/O
		intr2		Input	External Interrupt Signal
			pwm_sync	Input	Pulse Width Modulator Sync
189	pio11			I/O	General Purpose I/O
		intr1		Input	External Interrupt Signal
			ALM	Output	Alarm out: RTC
190	pio10			I/O	General Purpose I/O
		intr0		Input	External Interrupt Signal
			nRI0	Input	UART0 Ring Indicator
191	Vss				Ground
192	pio9			I/O	General Purpose I/O
		spi_in		Input	SPI Data Input
			uclk	Input	UART Clock
193	pio8			I/O	General Purpose I/O
		spi_out		Output	SPI Data Output
			irrxa	Input	IR Data Receive (Input)
194	utxd2			Output	UART 2 Transmit Data
		spi_enb		I/O	SPI Enable
			irtxa	Output	IR Transmit Data Output
195	urxd2			Input	UART 2 Receive Data
		spi_clk		I/O	SPI Clock
			irrx2	Input	IR Receive Data (2)
196	Vddc				Core Power
197	rd15			I/O	EBI Controller Data Input / Output Bus
		pio7		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘0’)
198	rd14			I/O	EBI Controller Data Input / Output Bus
		pio6		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘0’)
199	rd13			I/O	EBI Controller Data Input / Output Bus
		pio5		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘0’)
200	rd12			I/O	EBI Controller Data Input / Output Bus
		pio4		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘0’)
201	Vss				Ground
202	rd11			I/O	EBI Controller Data Input / Output Bus
		pio3		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘0’)
203	rd10			I/O	EBI Controller Data Input / Output Bus
		pio2		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘0’)
204	rd9			I/O	EBI Controller Data Input / Output Bus
		pio1		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘0’)
205	rd8			I/O	EBI Controller Data Input / Output Bus
		pio0		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘0’)
206	Vddb				I/O Power
207	rd7			I/O	EBI Controller Data Input / Output Bus – See Multiplexed I/O Tables.
		pio7		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘1’)
208	rd6			I/O	EBI Controller Data Input / Output Bus – See Multiplexed I/O Tables.
		pio6		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘1’)
209	rd5			I/O	EBI Controller Data Input / Output Bus – See Multiplexed I/O Tables.
		pio5		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘1’)
210	rd4			I/O	EBI Controller Data Input / Output Bus – See Multiplexed I/O Tables.
		pio4		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘1’)
211	Vss				Ground
212	rd3			I/O	EBI Controller Data Input / Output Bus – See Multiplexed I/O Tables.

Pin(S)	Name Function 1	Name Function 2	Name Function 3	Direction	Description
		pio3		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘1’)
213	rd2			I/O	EBI Controller Data Input / Output Bus – See Multiplexed I/O Tables.
		pio2		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘1’)
214	rd1			I/O	EBI Controller Data Input / Output Bus – See Multiplexed I/O Tables.
		pio1		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘1’)
215	rd0			I/O	EBI Controller Data Input / Output Bus – See Multiplexed I/O Tables.
		pio0		I/O	General Purpose I/O – See Multiplexed I/O Tables. (BIGEND = ‘1’)
216	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



NOTE: Dimensions are in mm.

NORTH AMERICA

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

EUROPE

SHARP Electronics (Europe) GmbH
Microelectronics Division
Sonninstraße 3
20097 Hamburg, Germany
Phone: (49) 40 2376-2286
Facsimile: (49) 40 2376-2232
http://www.sharpmed.com

ASIA

SHARP Corporation
Integrated Circuits Group
2613-1 Ichinomoto-Cho
Tenri-City, Nara, 632, Japan
Phone: +81-743-65-1321
Facsimile: +81-743-65-1532
http://www.sharp.co.jp