PIC64GX1000 64-bit Microprocessor

Introduction

The PIC64GX1000 RISC-V MPU is a highly power-efficient 64-bit Linux® capable processor, innovative, mid-range, embedded compute platform based on the RISC-V ISA.

The RISC-V CPU micro-architecture implementation is a simple, 5-stage single issue, in-order pipeline that does not suffer from the Meltdown and Spectre exploits found in common out-of-order machines. It has five RISC-V cores which are coherent with the memory subsystem allowing a versatile mix of deterministic real-time systems and Linux in a single, multi-core processor cluster. With Secure Boot built-in, innovative Linux and Real Time modes, a large flexible L2 memory subsystem, and a rich set of embedded peripherals. RISC-V MPU provides embedded developers new choices in secure, power-efficient, embedded compute platforms.

Features

The PIC64GX1000 RISC-V MPU supports the following features:

- Quad 625 MHz RV64GC RISC-V application cores (U54)
  - Physical Memory Protection (PMP) Unit
  - Memory Management Unit
  - L1 memory subsystem with single-error correct, double-error detect (SECDED)
    - 32 KB 8-way instruction cache or optional 28 KB tightly integrated memory
    - 32 KB 8-way data cache
- Single 625 MHz 64-bit RV64IMAC monitor processor core (E51)
  - 16 KB memory subsystem with SECDED configurable as 2-way L1 instruction cache or as an instruction tightly integrated memory
  - 8 KB data tightly integrated memory
  - PMP unit
- Flexible 2 MB L2 memory subsystem with SECDED configurable as:
  - 16-way set associative L2 cache
  - Loosely Integrated Memory (LIM) mode for deterministic access
  - Coherent Scratchpad Memory mode for shared messages across cores
- Integrated 36-bit DDR4/LPDDR4 memory controller with SECDED
  - DDR4 at 1.6 Gbps with a 8 Gb address reach
- Cache coherent CPU bus matrix
- AMBA I/O switch with QoS and memory protection
- Integrated 128 Kbytes embedded non-volatile memory (eNVM) for boot
- Boot options
  - Microchip secure boot
  - User defined, PUF-protected secure boot
  - Boot directly from 128 KB eNVM

© 2024 Microchip Technology Inc. and its subsidiaries

- Platform interrupt controller
  - 48 interrupt sources, with seven priority levels per core
- Debug
  - Ten hardware triggers per CPU (triggers can be configured as a breakpoint or a watchpoint)
  - Instruction trace on all CPUs
  - Performance counters
  - Runtime-configurable AXI bus monitors
    - Monitor AXI commands to DDR
    - Monitor an AXI port going into or out of the AMBA I/O AXI switch
- Processor I/O
  - Two GigE MACs
  - A USB 2.0 OTG
  - MMC 5.1 SD/SDIO
  - Two CAN 2.0 A and B
  - Execute in place Quad SPI flash controller
  - Five multi-mode UARTs
  - Two SPI, two I^2C
  - HDMI® 1.4
  - MIPI® CSI-2
  - RTC, GPIO
  - Five watchdog timers
  - Timers
  - Integrated x4 PCIe® Gen 2
- 1.0 V and 1.05 V operating modes
- Security Features
  - Use cryptoprocessor – Athena F5200 TeraFire Crypto Processor (1x), 200 MHz
  - Cryptography Research Incorporated (CRI)-patented differential power analysis (DPA) bitstream protection
  - Integrated dual physically unclonable function (PUF)
  - 56 Kbytes of secure, non-volatile memory (sNVM)
  - Built-in tamper detectors and countermeasures
  - Digest integrity check for sNVM, and eNVM
- Junction Temperature (TJ) Range – Industrial : -40°C to +100°C
- Packages
  - FCSG325 (11 mm x 11 mm)
  - FCVG484 (13 mm x 13 mm)

© 2024 Microchip Technology Inc. and its subsidiaries

Table of Contents

© 2024 Microchip Technology Inc. and its subsidiaries

© 2024 Microchip Technology Inc. and its subsidiaries

© 2024 Microchip Technology Inc. and its subsidiaries

1. Configuration Summary

The following tables list the configuration summary.

Table 1-1. PIC64GX1000-V/FCS and PIC64GX1000-C/FCS

Feature	PIC64GX1000-V/FCS and PIC64GX1000-C/FCS
Package	FCSG325 (11mmx11mm)
Pin pitch	0.5mm
External memory support	DDR4/LPDDR4
DDR bus width	16 bit
1GigE MACs	2
PCIe Gen 2	1 root port x1
USB 2.0	1 on-the-go port
SPI	2
I^2C	2
MMUARTS	5 multi-mode UARTs
CAN	2 CAN2.0 A and B
MMC	1 5.1 SD/SDIO
SPI Flash controller	1 quad SPI
Security	DPA protection, dual PUF, tamper detection and countermeasures
Application processing	4x 625MHZ RV64GC RISC-V with memory management and protection
Monitor processor	1x 625MHz RV64IMAC RISC-V with 16KB protected memory
Boot options	Secure boot, user defined secure boot, eNVM direct

Table 1-2. PIC64GX1000-V/FCV and PIC64GX1000-C/FCV

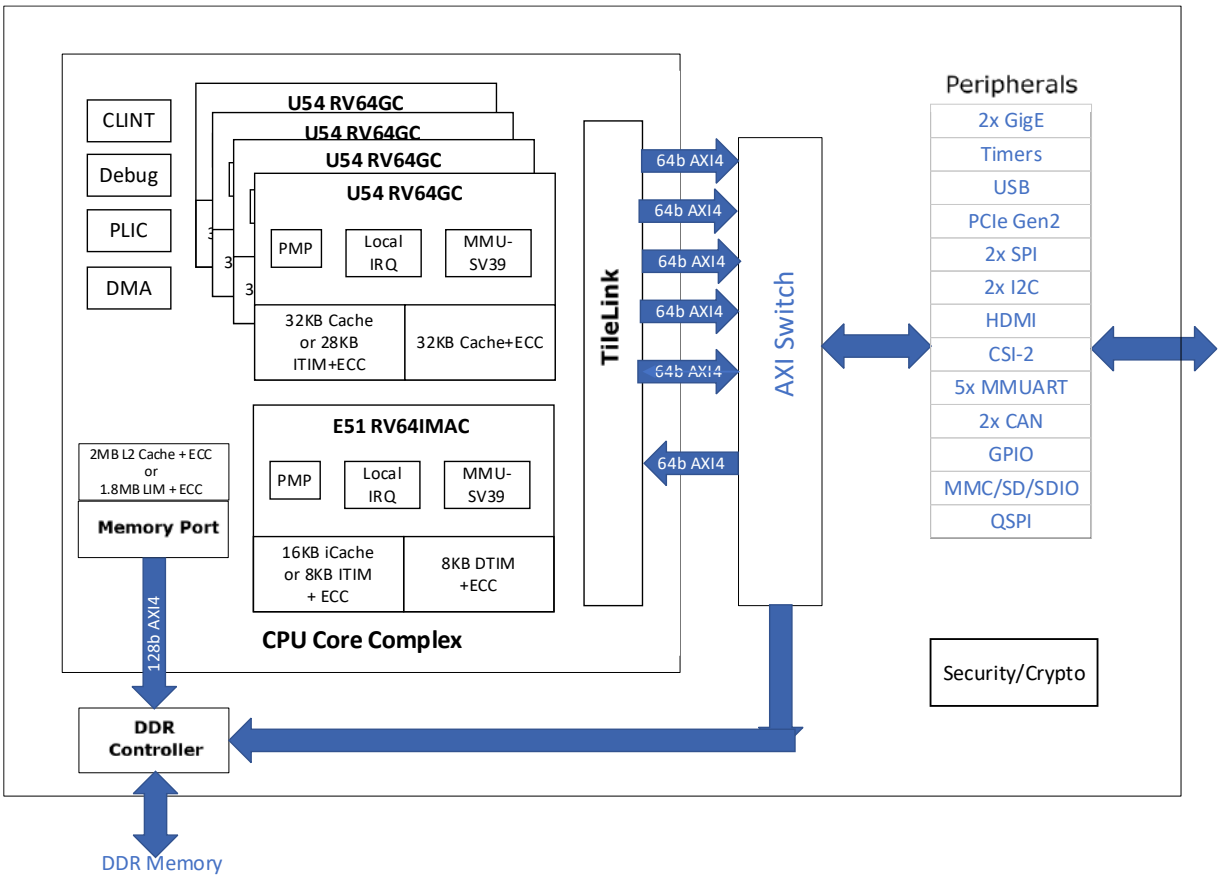
Feature	PIC64GX1000-V/FCV and PIC64GX1000-C/FCV
Package	FCVG484 (19mmx19mm)
Pin pitch	0.8mm
External memory support	DDR4/LPDDR4
DDR bus width	32 bit
1GigE MACs	2
PCIe Gen 2	1 root port x4
USB 2.0	1 on-the-go port
SPI	2
I^2C	2
MMUARTS	5 multi-mode UARTs
CAN	2 CAN2.0 A and B
MMC	1 5.1 SD/SDIO
SPI Flash controller	1 quad SPI
Security	DPA protection, dual PUF, tamper detection and countermeasures
Application processing	4x 625MHZ RV64GC RISC-V with memory management and protection
Monitor processor	1x 625 MHz RV64IMAC RISC-V with 16KB protected memory
Boot options	Secure boot, user defined secure boot, eNVM direct

© 2024 Microchip Technology Inc. and its subsidiaries

2. Block Diagram

The following figure shows the detailed block diagram of the device.

Figure 2-1. Detailed Block Diagram



© 2024 Microchip Technology Inc. and its subsidiaries

3. Absolute Maximum Ratings

The following table lists the absolute maximum ratings for PIC64GX1000 devices.

Table 3-1. Absolute Maximum Rating[3]

Parameter	Symbol	Min	Max	Unit
MPU core power supply	V_{DD}	–0.5	1.13	V
PCIe Tx and Rx lanes supply	V_{DDA}	–0.5	1.13	V
MPU core and MPU PLL high voltage supply	V_{DD25}	–0.5	2.7	V
PCIe PLL high-voltage supply	V_{DDA25}	–0.5	2.7	V
PCIe reference clock supply	$V_{DD\_XCVR\_CLK}$	–0.5	3.6	V
Global V_{REF} for transceiver reference clocks	$XCVR_{VREF}$	–0.5	3.6	V
DC supply for JTAG	V_{DDI3}	–0.5	3.6	V
GPIO I/O DC I/O Supply	V_{DDIx}	–0.5	3.6	V
DC supply for SGMII I/O	V_{DDI5}	–0.5	3.6	V
DC supply for DDR I/O	V_{DDI6}	–0.5	3.6	V
GPIO auxiliary power supply for I/O bank x[2]	VDD_{AUXx}	–0.5	3.6	V
Maximum DC input voltage on GPIO	V_{IN}	–0.5	3.8	V
PCIe receiver absolute input voltage	PCIe V_{IN}	–0.5	1.26	V
PCIe reference clock absolute input voltage	PCIe $REFCLKV_{IN}$	–0.5	3.6	V
Storage temperature(ambient)[1]	T_{STG}	–65	150	°C
Junction temperature[1]	T_J	–55	135	°C
Maximum soldering temperature RoHS	$T_{SOLROHS}$	—	260	°C

Note:

1. Absolute maximum ratings are stress ratings only; functional operation of the device at these or any other conditions beyond those listed under the recommended operating conditions specified in 4. Recommended Operating Conditions is not implied. Stresses beyond those listed in the following table might cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

© 2024 Microchip Technology Inc. and its subsidiaries

4. Recommended Operating Conditions

The following table lists the recommended operating conditions.

Table 4-1. Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Unit	Condition
MPU core supply at 1.0V mode[1, 6, 8]	V_{DD}	0.97	1.00	1.03	V	—
MPU core supply at 1.05V mode[1,6, 8]	V_{DD}	1.02	1.05	1.08	V	—
PCIe Tx and Rx lanes supply (1.0V mode)[6, 7, 9]	V_{DDA}	0.97	1.00	1.03	V	When all lane rates are 10.3125 Gbps or less.[1]
PCIe Tx and Rx lanes supply (1.05V mode)[6, 9]	V_{DDA}	1.02	1.05	1.08	V	Must when any lane rate is greater than 10.3125 Gbps. Lane rates 10.3125 Gbps or less may also be powered in 1.05V mode.[1]
MPU core and MPU PLL high voltage supply[6]	V_{DD25}	2.425	2.50	2.575	V	—
PCIe PLL high voltage supply[6]	V_{DDA25}	2.425	2.50	2.575	V	—
PCIe reference clock supply[6, 7]	$V_{DD\_PCIe\_CLK}$	3.135	3.3	3.465	V	3.3V nominal
		2.375	2.5	2.625	V	2.5V nominal
Global V_{REF} for PCIe reference clocks[3, 9]	$PCIe_{VREF}$	Ground	—	VDD_XCVR_ CLK	V	—
GPIO DC I/O supply[6]	V_{DDIx}	1.14	Various	3.465	V	Allowed nominal options: 1.2V, 1.5V, 1.8V, 2.5V, and 3.3V[2, 4, 5]
DC supply for JTAG	V_{DDI3}	1.71	Various	3.465	V	Allowed nominal options: 1.8V, 2.5V, and 3.3V
Peripheral DC I/O Supply	V_{DDIx}	1.14	Various	3.465	V	Allowed nominal options: 1.2V, 1.5V, 1.8V, 2.5V, and 3.3V[2, 4, 5]
DC supply for SGMII	V_{DDI5}	2.375	Various	3.465	V	Allowed nominal options: 2.5V, and 3.3V
		1.14	1.2	1.26	V	For DDR4
DC supply for DDR	V_{DDI6}	1.06	1.1	1.17	V	For LPDDR4
GPIO auxiliary supply[6]	V_{DDAUXx}	3.135	3.3	3.465	V	For I/O bank x with V_{DDIx} = 3.3V nominal[2, 4, 5]
		2.375	2.5	2.625	V	For I/O bank x with V_{DDIx} = 2.5V nominal or lower[2, 4, 5]
Extended commercial temperature range	T_J	0	—	100	°C	—
Industrial temperature range	T_J	–40	—	100	°C	—
Automotive T2 temperature range	T_J	–40	—	125	°C	—

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued						
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Unit**	**Condition**
Military temperature range	T_J	–55	—	125	°C	—

1. V_{DD} and V_{DDA} can independently operate at 1.0V or 1.05V nominal. These supplies are not dynamically adjustable.
2. If V_{DDIx} is 2.5V nominal or 3.3V nominal, V_{DDAUXx} must be connected to the V_{DDIx} supply. If V_{DDIx} is <2.5V nominal, V_{DDAUXx} must be powered at 2.5V nominal.
3. $PCIe_{VREF}$ globally sets the reference voltage of PCIe's single-ended reference clock input buffers. It is typically near $V_{DD\_PCIe\_CLK}/2$ but is allowed in the specified range.
4. The recommended power supply tolerances include DC offset of the supply plus any power supply ripple over the customer design frequencies of interest, as measured at the device package pins. An example for a valid power supply that meets the recommendations for the V_{DD} supply is 1.0V ±10 mV or 1.05V ±10 mV for DC offset with an additional power supply ripple of ±20 mV for a total of 1.0V ±30 mV or 1.05V ±30 mV.
5. Both V_{DDA} and $V_{DD\_PCIe\_CLK}$ supplies must be powered when any of the transceivers are used. $V_{DD\_PCIe\_CLK}$ must power on within the I/O calibration time (as specified for the device in MPLAB®). V_{DDA} and $V_{DD\_PCIe\_CLK}$ must both then remain powered during operation. If V_{DDA} needs to be powered down, $V_{DD\_PCIe\_CLK}$ must also be powered down. There is no required sequence for powering up or down V_{DDA} and $V_{DD\_PCIe\_CLK}$.
6. V_{DDI5} must be powered on when V_{DD} is powered on. V_{DDI5} must be powered on when V_{DD} is powered on. V_{DDI5} must power on within the I/O calibration time (as specified for the device in MPLAB). V_{DDI5} and V_{DD} must both remain powered during operation. If V_{DD} needs to be powered down, then V_{DDI5} must also be powered down. There is no required sequence for powering up or down of V_{DD} and V_{DDI5}.
7. Both V_{DDA} and $PCIe_{VREF}$ supplies must be powered when any of the PCIe ports are being used. $PCIe_{VREF}$ must be powered on when V_{DDA} is powered on. $PCIe_{VREF}$ must power on within the I/O calibration time (as specified for the device in MPLAB). V_{DDA} and $PCIe_{VREF}$ must both then remain powered during operation. If V_{DDA} needs to be powered down, $PCIe_{VREF}$ must also be powered down. There is no required sequence for powering up or down V_{DDA} and $PCIe_{VREF}$.

© 2024 Microchip Technology Inc. and its subsidiaries

5. CPU Core Complex

5.1 E51 RISC-V Monitor Core

The following table describes the features of E51.

Table 5-1. E51 RISC-V Monitor Core Features

Feature	Description
ISA	RV64IMAC
iCache/ITIM	16 KB 2-way set-associative/8 KB ITIM
DTIM	8 KB
ECC Support	Single-Error Correction and Double-Error Detection (SECDED) on iCache and DTIM.
Modes	Machine Mode, User Mode

Typically, in a system, the E51 is used to execute the following:

- Bootloader to boot the operating system on U54 cores
- Bare-metal user applications
- Monitoring user applications on U54 cores

Note: Load-Reserved and Store-Conditional atomic instructions (lr, sc) are not supported on the E51 processor core.

5.1.1 Instruction Fetch Unit

The instruction fetch unit consists of a 2-way set-associative 16 KB instruction cache that supports

64-byte cache line size with an access latency of one clock cycle. The instruction cache is asynchronous with the data cache. Writes to memory can be synchronized with the instruction fetch stream using the `FENCE.I` instruction.

5.1.2 Execution Pipeline

The E51 execution unit is a single-issue, in-order core with 5-stage execution pipeline. The pipeline comprises following five stages:

1. Instruction fetch
2. Instruction decode and register fetch
3. Execution
4. Data memory access
5. Register write back.

The pipeline has a peak execution rate of one instruction per clock cycle.

5.1.3 ITIM

The 16 KB iCache can be partially reconfigured into 8 KB ITIM. The 8 KB ITIM address range is listed in Table 11-1. ITIM is allocated in quantities of cache blocks, so it is not necessary to use the entire 8 KB as ITIM. Based on the requirement, part of the iCache can be configured as 2-way set associative and part of the cache can be configured as ITIM.

© 2024 Microchip Technology Inc. and its subsidiaries

5.1.4 DTIM

E51 includes an 8 KB DTIM, the address range of the DTIM is listed in Table 11-1. The DTIM has an access latency of two clock cycles for full words and three clock cycles for smaller words. Misaligned accesses are not supported in hardware and result in a trap.

5.1.5 Hardware Performance Monitor

The CSRs described in the following table implement the hardware performance monitoring scheme.

Table 5-2. Hardware Performance Monitoring CSRs

CSR	Function
mcycle	Holds a count of the number of clock cycles executed by a Hart since some arbitrary time in the past. The arbitrary time is the time since power-up.
minstret	Holds a count of the number of instructions retired by a Hart since some arbitrary time in the past. The arbitrary time is the time since power-up.
mhpmevent3 and mhpmevent4	Event Selectors: Selects the events as described in Table 5-3, and increments the corresponding mhpmcounter3 and mhpmcounter4 counters. The event selector register mhpmevent3 and mhpmevent4 are partitioned into two fields: event class and event mask as shown in Table 5-3. The lower 8 bits select an event class, and the upper bits form a mask of events in that class. The counter increments if the event corresponding to any set mask bit occurs. For example, if mhpmevent3 is set to 0x4200, mhpmcounter3 increments when either a load instruction or a conditional branch instruction retires. **Note:** In-flight and recently retired instructions may or may not be reflected when reading or writing the performance counters or writing the event selectors.
mhpmcounter3 and mhpmcounter4	40-bit event counters

Table 5-3. mhpmeventx Register

Event Class	mhpmeventx[8:18] Bit Field Description Events
mhpmeventx[7:0] = 0: Instruction Commit Events	8: Exception taken 9: Integer load instruction retired 10: Integer store instruction retired 11: Atomic memory operation retired 12: System instruction retired 13: Integer arithmetic instruction retired 14: Conditional branch retired 15: JAL instruction retired 16: JALR instruction retired 17: Integer multiplication instruction retired 18: Integer division instruction retired
mhpmeventx[7:0] = 1: Micro-architectural Events	8: Load-use interlock 9: Long-latency interlock 10: CSR read interlock 11: Instruction cache/ITIM busy 12: Data cache/DTIM busy 13: Branch direction misprediction 14: Branch/jump target misprediction 15: Pipeline flush from CSR write 16: Pipeline flush from other event 17: Integer multiplication interlock

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued	
Event Class	**mhpmeventx[8:18] Bit Field Description Events**
mhpmeventx[7:0] = 2: Memory System Events	8: Instruction cache miss 9: Memory-mapped I/O access 10: Data cache write back 11: Instruction TLB miss 12: Data TLB miss **Note:** Only L1 cache performance monitoring is supported.

5.1.6 ECC

By default, the E51 iCache and DTIM implement SECDED for ECC. The granularity at which this protection is applied (the codeword) is 32-bit (with an ECC overhead of 7 bits per codeword). The ECC feature of L1 cache is handled internally, user control is not supported.

When a single-bit error is detected in the L1 iCache, the error is corrected automatically, and the cache line is flushed and written back to the next level of memory hierarchy. When a single-bit error is detected in the L1 DTIM, the error is corrected automatically and written back to L1 DTIM.

5.1.6.1 ECC Reporting

ECC events are reported by the Bus Error Unit (BEU) block for a given core. The BEU can be configured to generate interrupts either globally via the Platform-Level Interrupt Controller (PLIC) or locally to the specific Hart where the ECC event occurred. When BEU interrupts are enabled, software can be used to monitor and count ECC events.

To detect uncorrectable ECC errors in the L1 cache memories, interrupts must be enabled in the BEU. The BEU must be configured to generate a local interrupt to halt the execution of a Hart when an uncorrectable instruction is detected. For more information about configuring ECC reporting, see 5.11. Bus Error Unit.

5.2 U54 RISC-V Application Cores

The following table describes the features of the U54 application cores.

Table 5-4. U54 RISC-V Application Cores Features

Feature	Description
ISA	RV64GC[1]
iCache/ITIM	32 KB 8-way set-associative/28 KB ITIM
dCache	32 KB 8-way set-associative
ECC Support	ECC on iCache, ITIM, and dCache
MMU	40-bit MMU compliant with Sv39
Modes	Machine mode, Supervisor mode, and User mode

Note:

1. In RV64GC, “G” = “IMAFD”.

Typically, in a system, the U54 cores are used to execute any of the following:

- Bare-metal user applications
- Operating systems

Note: Load-Reserved and Store-Conditional atomic instructions (lr, sc) are supported on U54 processor cores.

© 2024 Microchip Technology Inc. and its subsidiaries

5.2.1 Instruction Fetch Unit

The instruction fetch unit consists of an 8-way set-associative 32 KB iCache/28 KB ITIM that supports 64-byte cache line size with an access latency of one clock cycle. The U54s implement the standard Compressed (C) extension of the RISC-V architecture which allows 16-bit RISC-V instructions.

5.2.2 Execution Pipeline

The U54 execution unit is a single-issue, in-order core with 5-stage execution pipeline. The pipeline comprises following five stages:

1. Instruction fetch
2. Instruction decode and register fetch
3. Execution
4. Data memory access
5. Register write back.

The pipeline has a peak execution rate of one instruction per clock cycle, and is fully bypassed so that most instructions have a one-cycle result latency.

Most CSR writes result in a pipeline flush with a five-cycle latency.

5.2.3 Instruction Cache

The iCache memory consists of a dedicated 32 KB 8-way set-associative, Virtually Indexed Physically Tagged (VIPT) instruction cache memory with a line size of 64 bytes. The access latency of any block in the iCache is one clock cycle. iCache is not coherent with the platform memory system. Writes to iCache must be synchronized with the instruction fetch stream by executing the `FENCE.I` instruction.

A cache line fill triggers a burst access outside the CPU Core Complex. The U54 processor core caches instructions from executable addresses, with the exception of ITIM. See 5.3. CPU Memory Map for all executable address regions, which are denoted by the attribute X. Trying to execute an instruction from a non-executable address results in a trap.

5.2.4 ITIM

iCache can be partially configured as ITIM, which occupies a 28 KB of address range in 5.3. CPU Memory Map. ITIM provides high-performance, predictable instruction delivery. Fetching an instruction from ITIM is as fast as an iCache hit, without any cache misses. ITIM can hold data and instructions. Load and store operations to ITIM are not as efficient as load and store operations to E51 DTIM.

The iCache can be configured as ITIM for any ways in units of cache lines (64 B bytes). A single iCache way must remain as instruction cache. ITIM is allocated simply by storing to it. A store to the nth byte of the ITIM memory map reallocates the first (n + 1) bytes of iCache as ITIM, rounded up to the next cache line.

ITIM can be deallocated by storing zero to the first byte after the ITIM region, that is 28 KB after the base address of ITIM as indicated in 5.3. CPU Memory Map. The deallocated ITIM space is automatically returned to iCache.

Software must clear the contents of ITIM after allocating it. It is unpredictable whether ITIM contents are preserved between deallocation and allocation.

5.2.5 Data Cache

The U54 dCache has an 8-way set-associative 32 KB write-back, VIPT data cache memory with a line size of 64 bytes. Access latency is two clock cycles for words and double-words, and three clock cycles for smaller quantities. Misaligned accesses are not supported in hardware and result in a

© 2024 Microchip Technology Inc. and its subsidiaries

trap. dCache is kept coherent with a directory-based cache coherence manager, which resides in the L2 cache.

Stores are pipelined and committed on cycles where the data memory system is otherwise idle. Loads to addresses currently in the store pipeline result in a five-cycle latency.

5.2.6 Atomic Memory Operations

The U54 core supports the RISC-V standard Atomic (A) extension on regions of the Memory Map denoted by the attribute A in 5.3. CPU Memory Map. Atomic memory operations to regions that do not support them generate an access exception precisely at the core.

The load-reserved and store-conditional instructions are only supported on cached regions, hence generate an access exception on DTIM and other uncached memory regions.

See The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Version 2.1 for more information on the instructions added by this extension.

5.2.7 Floating Point Unit

The U54 FPU provides full hardware support for the IEEE 754-2008 floating-point standard for 32-bit single-precision and 64-bit double-precision arithmetic. The FPU includes a fully pipelined fused-multiply-add unit and an iterative divide and square-root unit, magnitude comparators, and float-to-integer conversion units, all with full hardware support for subnormals and all IEEE default values

5.2.8 MMU

The U54 has support for virtual memory using a Memory Management Unit (MMU). The MMU supports the Bare and Sv39 modes as described in The RISC-V Instruction Set Manual, Volume II: Privileged Architecture, Version 1.10.

The U54 MMU has a 39-bit virtual address space mapped to a 48-bit physical address space. A hardware page-table walker refills the address translation caches. Both instruction and data address translation caches are fully associative, and have 32 entries. The MMU supports 2 MB megapages and 1 GB gigapages to reduce translation overheads for large contiguous regions of virtual and physical address space.

U54 cores do not automatically set the Accessed (A) and Dirty (D) bits in a Sv39 PTE. The U54 MMU raises a page fault exception for a read to a page with PTE.A=0 or a write to a page with PTE.D=0.

5.2.9 ECC

By default, the iCache, ITIM, and dCache implement SECDED for ECC. ECC is applied at the 32-bit codeword level, with an ECC overhead of 7 bits per codeword. The ECC feature of L1 cache is handled internally, user control is not supported.

When a single-bit error is detected in the ITIM, the error is corrected automatically and written back to the SRAM. When a single-bit error is detected in the L1 instruction cache, the error is corrected automatically and the cache line is flushed. When a single-bit error is detected in the L1 data cache, the data cache automatically implements the following sequence of operations:

1. Corrects the error.
2. Invalidates the cache line.
3. Writes the line back to the next level of the memory hierarchy.

The ECC reporting scheme is same as described in 7.1.6.1 ECC Reporting.

5.2.10 Hardware Performance Monitor

The scheme is the same as described in 7.1.5. Hardware Performance Monitor.

© 2024 Microchip Technology Inc. and its subsidiaries

5.3 CPU Memory Map

The overall physical memory map of the CPU Core Complex is shown in Section 13- MPU Memory Map. The CPU Core Complex is configured with a 38-bit physical address space.

5.4 Physical Memory Protection

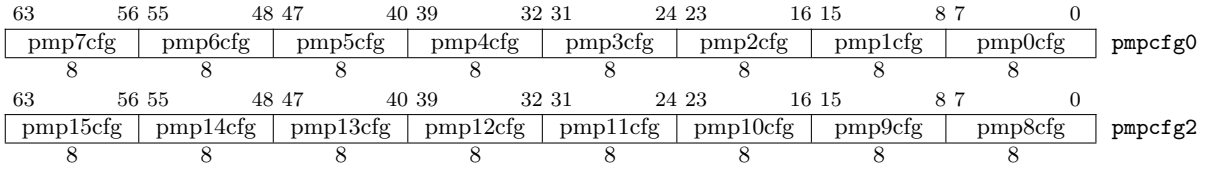
Exclusive access to memory regions for a processor core (Hart) can be enabled by configuring its PMP registers. Each Hart supports a Physical Memory Protection (PMP) unit with 16 PMP regions. The PMP unit in each processor core includes the following control and status registers (CSRs) to enable the PMP:

- PMP Configuration Register (pmpcfg)—Used for setting privileges (R, W, and X) for each PMP region.
- PMP Address Register (pmpaddr)—Used for setting the address range for each PMP region.

5.4.1 PMP Configuration Register (pmpcfg)

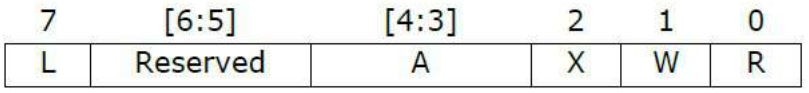
The pmpcfg0 and pmpcfg2 registers support eight PMP regions, each as shown in the following figure. These two registers hold the configurations for the 16 PMP regions. Each PMP region is referred as pmpicfg. In pmpicfg, i ranges from 0 to 15 (pmp0cfg, pmp1cfg ... pmp15cfg). PIC64GX supports RV64. For RV64, pmpcfg1 and pmpcfg3 are not used.

Figure 5-1. RV64 PMP Configuration CSR Layout



The following figure shows the layout of a pmpicfg register. The R, W, and X bits, when set, indicate that the PMP entry permits read, write, and instruction execution, respectively. When one of these bits is cleared, the corresponding access type is denied. The Address-Matching (A) field encodes the Address-Matching mode of the associated PMP address register. The Locking and Privilege mode (L) bit indicates that the PMP entry is locked.

Figure 5-2. PMP Configuration Register Format



The A field in a PMP entry's configuration register encodes the address-matching mode of the associated PMP address register. When A=0, this PMP entry is disabled and matches no addresses. Three address-matching modes are supported—Top of Range (TOR), naturally aligned four-byte regions (NA4), naturally aligned power-of-two regions (NAPOT) as listed in the following table.

Table 5-5. Encoding of A field in PMP Configuration Registers

Address Matching	Name	Description
0	OFF	No region (disabled)
1	TOR	Top of range
2	NA4	Naturally aligned four-byte region
3	NAPOT	Naturally aligned power-of-two region, ≥ 8 bytes

© 2024 Microchip Technology Inc. and its subsidiaries

NAPOT ranges make use of the low-order bits of the associated address register to encode the size of the range, as listed in the following figure.

Table 5-6. NAPOT Range Encoding

pmpaddr (Binary)	pmpcfg.A Value	Match Type and Size
aaaa...aaaa	NA4	4-byte NAPOT range
aaaa...aaa0	NAPOT	8-byte NAPOT range
aaaa...aa01	NAPOT	16-byte NAPOT range
aaaa...a011	NAPOT	32-byte NAPOT range
...	...	...
aa01...1111	NAPOT	2XLEN-byte NAPOT range
a011...1111	NAPOT	2XLEN+1byte NAPOT range
0111...1111	NAPOT	2XLEN+2byte NAPOT range

5.4.1.1 Locking and Privilege Mode

The L bit indicates that the PMP entry is locked, that is, writes to the Configuration register (pmpicfg) and associated address registers (pmpaddr) are ignored. Locked PMP entries can only be unlocked with a system reset. In addition to locking the PMP entry, the L bit indicates whether the R/W/X permissions are enforced on Machine (M) mode accesses. When the L bit is set, these permissions are enforced for all privilege modes. When the L bit is clear, any M-mode access matching the PMP entry succeeds; the R/W/X permissions apply only to Supervisor (S) and User (U) modes.

5.4.2 PMP Address Register (pmpaddr)

The PMP address registers are CSRs named from pmpaddr0 to pmpaddr15. Each PMP address register encodes the bits [55:2] of a 56-bit physical address as shown in the following figure.

Figure 5-3. RV64 PMP Address Register Format

Note: Bits [1:0] of PMP address region are not considered because minimum granularity is four bytes.

For more information about the RISC-V physical memory protection, see The RISC-V Instruction Set Manual, Volume II: Privileged Architecture, Version 1.10.

5.5 L2 Cache

The shared 2 MB L2 cache is divided into four address-interleaved banks to improve performance. Each bank is 512 KB in size, and is a 16-way set-associative cache. The L2 also supports runtime reconfiguration between cache and scratchpad RAM.

5.6 L2 Cache Controller

The L2 cache controller offers extensive flexibility as it allows for several features in addition to the Level 2 cache functionality such as memory-mapped access to L2 cache RAM for disabled cache ways, scratchpad functionality, way masking and locking, and ECC support with error tracking statistics, error injection, and interrupt signaling capabilities.

Note: L2 cache controller supports single-bit ECC via ECC registers. Dual-bit ECC is implemented by default and is not visible to the user.

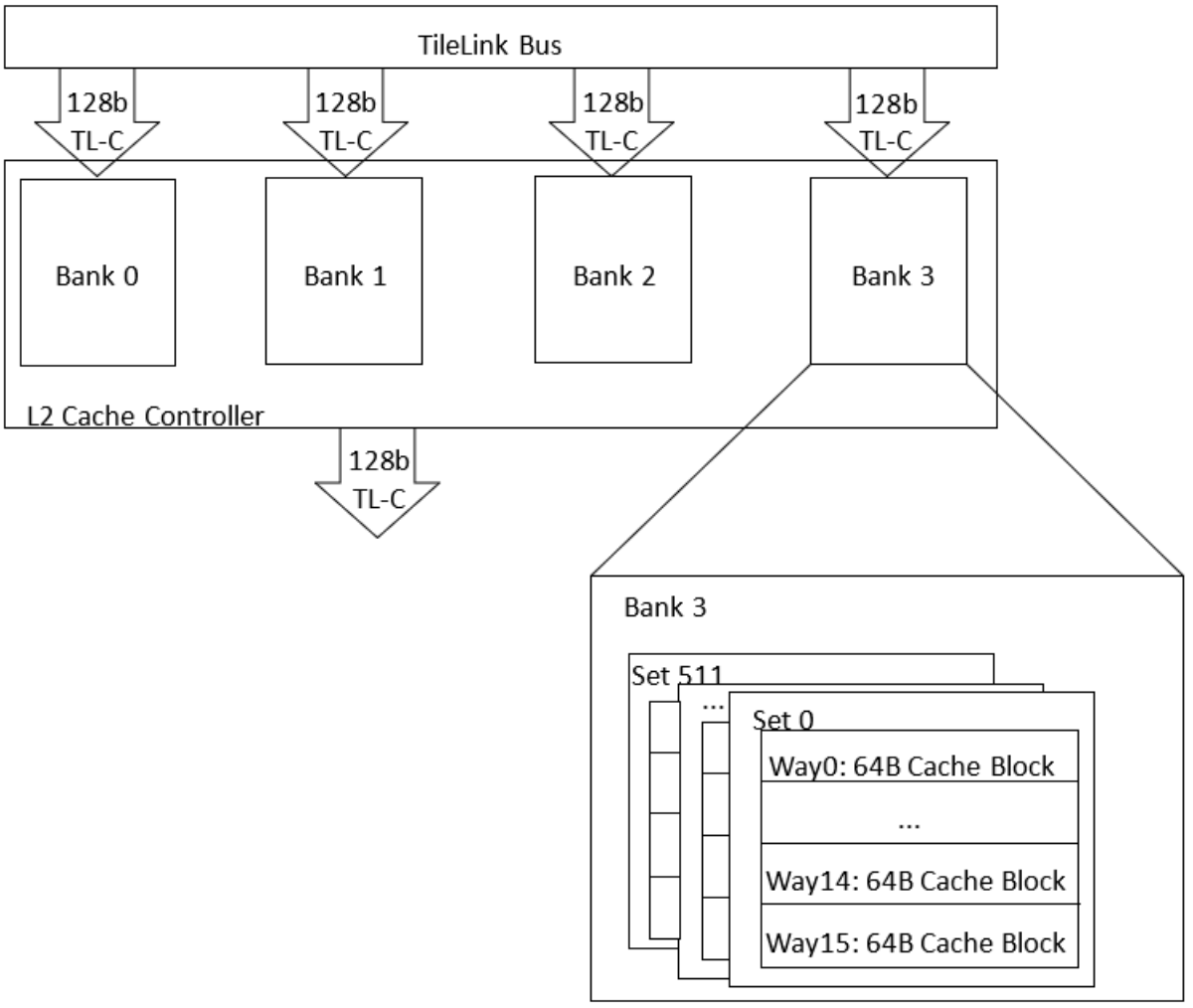
© 2024 Microchip Technology Inc. and its subsidiaries

5.6.1 Functional Description

The L2 cache controller is configured into four banks, each bank contains 512 sets of 16 ways and each way contains a 64-byte block. This subdivision into banks facilitates increased available bandwidth between CPU masters and the L2 cache as each bank has its own 128-bit TL-C (TileLink Cached) inner port. Hence, multiple requests to different banks may proceed in parallel.

The outer port of the L2 cache controller is a 128-bit TL-C port shared among all banks and connected to a DDR controller. The overall organization of the L2 cache controller is shown in the following figure.

Figure 5-4. L2 Cache Controller



5.6.1.1 Way Enable and the L2 LIM

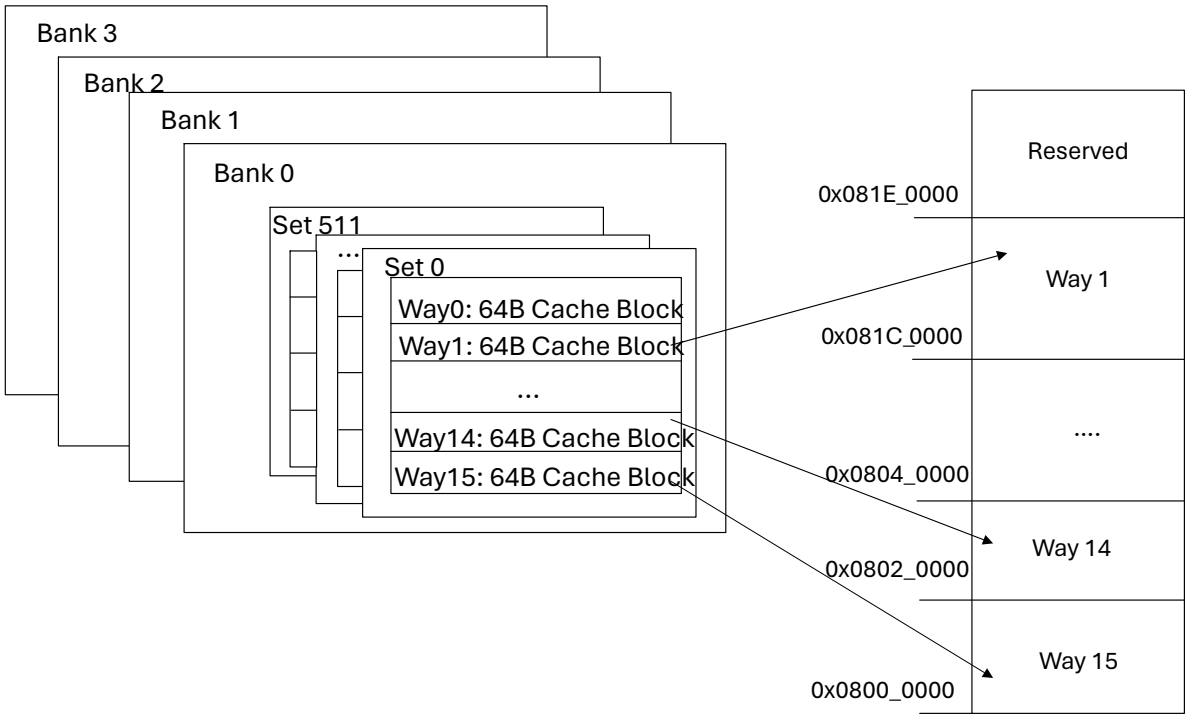
Similar to ITIM, L2 cache can be configured as LIM, or as a cache which is controlled by the L2 cache controller to contain a copy of any cacheable address.

When cache ways are disabled, they are addressable in the L2-LIM address space in 11. Memory Map. Fetching instructions or data from the L2-LIM provides deterministic behavior equivalent to an L2 cache hit, with no possibility of a cache miss. Accesses to L2-LIM are always given priority over cache way accesses which target the same L2 cache bank.

© 2024 Microchip Technology Inc. and its subsidiaries

After reset, all ways are disabled, except way0. Cache ways can be enabled by writing to the WayEnable register described in 5.6.3.2. Way Enable Register (WayEnable). Once a cache way is enabled, it cannot be disabled unless the Core Complex is reset. The highest numbered L2 cache way is mapped to the lowest L2-LIM address space, and way 1 occupies the highest L2-LIM address range. When L2 cache ways are enabled, the size of the L2-LIM address space shrinks. The mapping of L2 cache ways to L2-LIM address space is shown in the following figure.

Figure 5-5. Mapping of L2 Cache Ways to L2-LIM Addresses



5.6.1.2 Way Masking and Locking

The L2 cache controller controls the amount of cache allocated to a CPU master using the `WayMaskX` register described in 5.6.3.13. Way Mask Registers (WayMaskX). `WayMaskX` registers only affect allocations and reads can still occur to ways which are masked. To lock down specific cache ways, mask them in all `WayMaskX` registers. In this scenario, all masters will be able to read data in the locked cache ways but not be able to evict.

5.6.1.3 L2 Cache Power Control

Shutdown controls are provided for the 2 MB L2 cache memory with configuration support for either 512 KB, 1 MB, or 1,512 KB of L2 cache. This enables less static power consumption. The following 4-bit control register is provided for shutting down L2 cache blocks.

Table 5-7. L2 Cache Power Down

Register	Bits	Description
L2_SHUTDOWN_CR (0x174)	[3:0]	Configured to shutdown L2 cache blocks of Bank 0 to 3.

The above 4-bit control register powers down L2 cache blocks as per the physical RAM construction represented in the following table. Each bank contains 512 KB, constructed from thirty two 2048x64 RAMs (cc_ram_x), where the size of each RAM is 16 KB.

Note: Actual RAM width is 72 bits as an additional 8 ECC bits are used per 64-bit word.

© 2024 Microchip Technology Inc. and its subsidiaries

Table 5-8. L2 RAM Shutdown

	L2_SHUTDOWN_CR[3]	L2_SHUTDOWN_CR[2]	L2_SHUTDOWN_CR[1]	L2_SHUTDOWN_CR [0]
	cc_ram_24	cc_ram_16	cc_ram_8	cc_ram_0
Bank 0	cc_ram_25	cc_ram_17	cc_ram_9	cc_ram_1
	cc_ram_26	cc_ram_18	cc_ram_10	cc_ram_2
	cc_ram_27	cc_ram_19	cc_ram_11	cc_ram_3
	cc_ram_28	cc_ram_20	cc_ram_12	cc_ram_4
	cc_ram_29	cc_ram_21	cc_ram_13	cc_ram_5
	cc_ram_30	cc_ram_22	cc_ram_14	cc_ram_6
	cc_ram_31	cc_ram_23	cc_ram_15	cc_ram_7
	cc_ram_24	cc_ram_16	cc_ram_8	cc_ram_0
Bank 1	cc_ram_25	cc_ram_17	cc_ram_9	cc_ram_1
	cc_ram_26	cc_ram_18	cc_ram_10	cc_ram_2
	cc_ram_27	cc_ram_19	cc_ram_11	cc_ram_3
	cc_ram_28	cc_ram_20	cc_ram_12	cc_ram_4
	cc_ram_29	cc_ram_21	cc_ram_13	cc_ram_5
	cc_ram_30	cc_ram_22	cc_ram_14	cc_ram_6
	cc_ram_31	cc_ram_23	cc_ram_15	cc_ram_7
	cc_ram_24	cc_ram_16	cc_ram_8	cc_ram_0
Bank 2	cc_ram_25	cc_ram_17	cc_ram_9	cc_ram_1
	cc_ram_26	cc_ram_18	cc_ram_10	cc_ram_2
	cc_ram_27	cc_ram_19	cc_ram_11	cc_ram_3
	cc_ram_28	cc_ram_20	cc_ram_12	cc_ram_4
	cc_ram_29	cc_ram_21	cc_ram_13	cc_ram_5
	cc_ram_30	cc_ram_22	cc_ram_14	cc_ram_6
	cc_ram_31	cc_ram_23	cc_ram_15	cc_ram_7
	cc_ram_24	cc_ram_16	cc_ram_8	cc_ram_0
Bank 3	cc_ram_25	cc_ram_17	cc_ram_9	cc_ram_1
	cc_ram_26	cc_ram_18	cc_ram_10	cc_ram_2
	cc_ram_27	cc_ram_19	cc_ram_11	cc_ram_3
	cc_ram_28	cc_ram_20	cc_ram_12	cc_ram_4
	cc_ram_29	cc_ram_21	cc_ram_13	cc_ram_5
	cc_ram_30	cc_ram_22	cc_ram_14	cc_ram_6
	cc_ram_31	cc_ram_23	cc_ram_15	cc_ram_7

5.6.1.4 Scratchpad

The L2 cache controller has a dedicated scratchpad address region which allows for allocation into the cache using an address range which is not memory backed. This address region is denoted as the L2 Zero Device in 11. Memory Map. Writes to the scratchpad region will allocate into cache ways which are enabled and not masked. Care must be taken with the scratchpad, as there is no memory backing this address space. Cache evictions from addresses in the scratchpad results in data loss.

The main advantage of the L2 scratchpad over the L2-LIM is that it is a cacheable region allowing for data stored to the scratchpad to also be cached in a master's L1 data cache resulting in faster access.

© 2024 Microchip Technology Inc. and its subsidiaries

The recommended procedure for using the L2 Scratchpad is as follows:

1. Use the WayEnable register to enable the desired cache ways.
2. Designate a single master which will be allocated into the scratchpad. For this procedure, designate the master as Master S. All other masters (CPU and non-CPU) will be denoted as Masters X.
3. Masters X: write to the WayMaskX register to mask all ways which are to be used for the scratchpad. This will prevent Masters X from evicting cache lines in the designated scratchpad ways.
4. Master S: write to the WayMaskX register to mask all ways except the ways which are to be used for the scratchpad. At this point Master S should only be able to allocate into the cache ways meant to be used as a scratchpad.
5. Master S: write scratchpad data into the L2 Scratchpad address range (L2 Zero Device).
6. Master S: Repeat steps 4 and 5 for each way to be used as scratchpad.
7. Master S: Use the WayMaskX register to mask the scratchpad ways for Master S so that it cannot evict cache lines from the designated scratchpad ways.

At this point, the scratchpad ways should contain the scratchpad data, with all masters able to read, write, and execute from this address space, and no masters able to evict the scratchpad contents.

5.6.1.5 L2 ECC

The L2 cache controller supports ECC for Single-Error Correction and Double-Error Detection (SECDED). The cache controller also supports ECC for meta-data information (index and tag information) and can perform SECDED. The single-bit error injection is available for the user to control. Dual-bit error injection is handled internally without user control.

Whenever a correctable error is detected, the caches immediately repair the corrupted bit an write it back to SRAM. This corrective procedure is completely invisible to application software. However, to support diagnostics, the cache records the address of the most recently corrected meta-data and data errors. Whenever a new error is corrected, a counter is incremented and an interrupt is raised. There are independent addresses, counters, and interrupts for correctable meta-data and data errors.

DirError, DirFail, DataError, and DataFail signals are used to indicate that an L2 meta-data, data, or un-correctable L2 data error has occurred respectively. These signals are connected to the PLIC as described in 10.6.2. Interrupt Sources and are cleared upon reading their respective count registers.

5.6.2 Register Map

Table 5-9. L2 Cache Controller Register Map

Offset	Width	Attributes	Register Name	Notes
0x000	4B	RO	Config	Information on the configuration of the L2 cache
0x008	1B	RW	WayEnable	Way enable register

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued				
Offset	**Width**	**Attributes**	**Register Name**	**Notes**
0x040	4B	RW	ECCInjectError	ECC error injection register
0x100	8B	RO	ECCDirFixAddr	Address of most recently corrected metadata error
0x108	4B	RO	ECCDirFixCount	Count of corrected metadata errors
0x120	8B	RO	ECCDirFailAddr	Address of most recent uncorrectable metadata error
0x128	8B	RO	ECCDirFailCount	Count of uncorrectable metadata errors
0x140	8B	RO	ECCDataFixAddr	Address of most recently corrected data error
0x148	4B	RO	ECCDataFixCount	Count of corrected data errors
0x160	8B	RO	ECCDataFailAddr	Address of most recent uncorrectable data error
0x168	4B	RO	ECCDataFailCount	Count of uncorrectable data errors
0x200	8B	WO	Flush64	Flush cache block, 64-bit address
0x240	4B	WO	Flush32	Flush cache block, 32-bit address
0x800	8B	RW	Master 0 way mask register	DMA
0x808	8B	RW	Master 1 way mask register	AXI4_front_port ID#0
0x810	8B	RW	Master 2 way mask register	AXI4_front_port ID#1
0x818	8B	RW	Master 3 way mask register	AXI4_front_port ID#2
0x820	8B	RW	Master 4 way mask register	AXI4_front_port ID#3 Hart 0 dCache MMIO
0x828	8B	RW	Master 5 way mask register	Hart 0 iCache
0x830	8B	RW	Master 6 way mask register	Hart 1 dCache
0x838	8B	RW	Master 7 way mask register	Hart 1 iCache
0x840	8B	RW	Master 8 way mask register	Hart 2 dCache
0x848	8B	RW	Master 9 way mask register	Hart 2 iCache
0x850	8B	RW	Master 10 way mask register	Hart 3 dCache
0x858	8B	RW	Master 11 way mask register	Hart 3 iCache
0x860	8B	RW	Master 12 way mask register	Hart 4 dCache
0x868	8B	RW	Master 13 way mask register	Hart 4 ICache
0x870	8B	RW	Master 14 way mask register	

5.6.3 Register Description

This section describes registers of the L2 cache controller referenced in 5.6.2. Register Map.

5.6.3.1 Cache Register Configuration (Config)

The `Config` register can be used to programmatically determine information regarding the cache.

Table 5-10. Cache Configuration Register (Config)

Register Offset		**0x000**		
Bits	**Field Name**	**Attributes**	**Reset**	**Description**
[7:0]	Banks	RO	4	Return the number of banks in the cache
[15:8]	Ways	RO	16	Return the total number of enabled ways in the cache
[23:16]	Sets	RO	9	Return the Base-2 logarithm of the number of sets in a cache bank
[31:24]	Bytes	RO	6	Return the Base-2 logarithm of the number of bytes in a cache blocks

5.6.3.2 Way Enable Register (WayEnable)

The WayEnable register determines which ways of the L2 cache controller are enabled as cache. Cache ways which are not enabled, are mapped into the L2-LIM as described in 11. Memory Map.

© 2024 Microchip Technology Inc. and its subsidiaries

This register is initialized to 0 on reset and may only be increased. This means that, out of Reset, only a single L2 cache way is enabled as one cache way must always remain enabled. Once a cache way is enabled, the only way to map it back into the L2-LIM address space is by a Reset.

Table 5-11. Way Enable Register (WayEnable)

Register Offset		0x008		
Bits	Field Name	Attributes	Reset	Description
[7:0]	Way Enable	RW	0	Way indexes less than or equal to this register value may be used by the cache
[63:8]	Reserved	RW	—	—

5.6.3.3 ECC Error Injection Register (ECCInjectError)

The `ECCInjectError` register can be used to insert an ECC error into either the backing data or meta-data SRAM. This function can be used to test error correction logic, measurement, and recovery.

The ECC Error injection system works only during writes, which means that the stored data and ECC bits are modified on a write. ECC error is not injected or detected until a write occurs. Hence, a read will complete without ECC errors being detected if a write is not carried out after enabling the ECC error injection register.

Table 5-12. ECC Error Injection Register (ECCInjectError)

Register Offset		0x040		
Bits	Field Name	Attributes	Reset	Description
[7:0]	Bit Position	RW	0	Specifies a bit position to toggle, within an SRAM. The width is SRAM width depends on the micro architecture, but is typically 72 bits for data SRAMs and ≈ 24 bits for Directory SRAM.
[15:8]	Reserved	RW		—
16	Target	RW	0	Setting this bit means the error injection will target the metadata SRAMs. Otherwise, the error injection targets the data SRAMs.
[31:17]	Reserved	RW	—	—

5.6.3.4 ECC Directory Fix Address (ECCDirFixAddr)

The `ECCDirFixAddr` register is a Read-Only register which contains the address of the most recently corrected metadata error. This field only supplies the portions of the address which correspond to the affected set and bank, because all ways are corrected together.

5.6.3.5 ECC Directory Fix Count (ECCDirFixCount)

The ECCDirFixCount register is a Read Only register which contains the number of corrected L2 metadata errors. Reading this register clears the DirError interrupt signal described in 5.6.1.5. L2 ECC.

5.6.3.6 ECC Directory Fail Address (ECCDirFailAddr)

The `ECCDirFailAddr` register is a Read-Only register which contains the address of the most recent uncorrected L2 metadata error.

5.6.3.7 ECC Directory Fail Count (ECCDirFailCount)

The `ECCDirFailCount` register is a Read-Only register which contains the number of uncorrected L2 metadata errors.

5.6.3.8 ECC Data Fix Address (ECCDataFixAddr)

The `ECCDataFixAddr` register is a Read-Only register which contains the address of the most recently corrected L2 data error.

© 2024 Microchip Technology Inc. and its subsidiaries

5.6.3.9 ECC Data Fix Count (ECCDataFixCount)

The ECCDataFixCount register is a Read Only register which contains the number of corrected data errors. Reading this register clears the DataError interrupt signal described in 5.6.1.5. L2 ECC.

5.6.3.10 ECC Data Fail Address (ECCDataFailAddr)

The `ECCDataFailAddr` register is a Read-Only register which contains the address of the most recent uncorrected L2 data error.

5.6.3.11 ECC Data Fail Count (ECCDataFailCount)

The `ECCDataFailCount` register is a Read-Only register which contains the number of uncorrected data errors. Reading this register clears the DataFail interrupt signal described in 5.6.1.5. L2 ECC.

5.6.3.12 Cache Flush Registers

The L2 cache controller provides two registers which can be used for flushing specific cache blocks. Flush64 is a 64-bit write only register that will flush the cache block containing the address written.

Flush32 is a 32-bit write only register that will flush a cache block containing the written address left shifted by 4 bytes. In both registers, all bits must be written in a single access for the flush to take effect.

5.6.3.13 Way Mask Registers (WayMaskX)

The WayMaskX register allows a master connected to the L2 cache controller to specify which L2 cache ways can be evicted by master 'X' as specified in the WayMaskX register. Masters can still access memory cached in masked ways. At least one cache way must be enabled. It is recommended to set/clear bits in this register using atomic operations.

Table 5-13. Way MaskX Register (WayMaskX)

Register Offset	0x800 + (8 x Master ID)			
Bits	**Field Name**	**Attributes**	**Reset**	**Description**
0	Way0 Mask	RW	1	Clearing this bit masks L2 Cache Way 0
1	Way1 Mask	RW	1	Clearing this bit masks L2 Cache Way 1
...				
15	Way15 Mask	RW	1	Clearing this bit masks L2 Cache Way 15
[63:16]	Reserved	RW	1	—

Front Port Way Masks

The CPU Core Complex front port passes through an AXI to TileLink interface. This interface maps incoming transactions to the four internal TileLink IDs, which are referred to in the above WayMaskX table. These IDs are not related to the incoming AXI transaction IDs. The allocation of the TileLink IDs is dependent on the number of outstanding AXI transactions, the arrival rate relative to the transaction completion cycle, and previous events. It is not possible to predict which internal ID will be allocated to each AXI transaction and therefore, which set of way masks will apply to that AXI transaction. Hence, it is recommended that all four front port way masks are configured identically. See 5.6.2. Register Map for front port WayMaskX registers.

5.7 Branch Prediction

Branch prediction is supported by all the processor cores. The branch prediction block includes the following components:

- A 28-entry branch target buffer (BTB), for predicting the target of taken branches.
- A 512-entry branch history table (BHT), for predicting the direction of conditional branches.
- A 6-entry return address stack (RAS), for predicting the target of procedure returns.

The branch prediction incurs a one-cycle latency, such that correctly predicted control-flow instructions result in no penalty. Mispredicted control-flow instructions incur a three-cycle latency.

© 2024 Microchip Technology Inc. and its subsidiaries

The Branch Prediction Mode (bpm) M-mode CSR at 0x7C0 is used to customize the current branch prediction behavior for predictable execution time. The following table lists the `bpm` CSR.

Table 5-14. Branch Prediction Mode (bpm) CSR

Branch Prediction Mode (0x7C0)			
Bits	**Field Name**	**Attribute**	**Description**
0	bdp	WARL	Branch Direction Prediction. Determines the value returned by the BHT component of the branch prediction system. • A zero value indicates the dynamic direction prediction • a non-zero value indicates the static-taken direction prediction. The BTB is cleared on any write to bdp, and the RAS is unaffected by writes to bdp.
[63:1]	Reserved	RO	—

5.8 External Bus Interfaces

The following six AMBA AXI4 compliant external ports enable the CPU Core Complex to access main memory and peripherals (see 2. Block Diagram).

- AXI 128 to DDR Controller
- D0 (Datapath0)
- D1 (Datapath1)
- F0 (FIFO0)
- F1 (FIFO1)
- NC (Non-Cached)

To enable non-CPU masters access to the CPU Core Complex, there is an AMBA AXI4 compliant master bus port (S8 on the AXI Switch).

5.9 DMA Engine

The DMA Engine supports the following:

- Independent concurrent DMA transfers using four DMA channels
- Generation of PLIC interrupts on various conditions during DMA execution

The memory-mapped control registers of the DMA engine can be accessed over the TileLink slave interface. This interface enables the software to initiate DMA transfers. The DMA engine also includes a master port which goes into the TileLink bus. This interface enables the DMA engine to independently transfer data between slave devices and main memory, or to rapidly copy data between two locations in the main memory.

The DMA engine includes four independent DMA channels capable of operating in parallel to enable multiple concurrent transfers. Each channel supports an independent set of control registers and two interrupts which are described in the next sections.

The DMA engine supports two interrupts per channel to signal a transfer completion or a transfer error. The channel's interrupts are configured using its Control register described in the next section. The mapping of the CPU Core Complex DMA interrupt signals to the PLIC is described in 10.6. Platform Level Interrupt Controller.

5.9.1 DMA Memory Map

The DMA engine contains an independent set of registers for each channel. Each channel's registers start at the offset 0x1000 so that the base address for any DMA channel is:

DMA Base Address + (0x1000 × Channel ID). The register map of a DMA channel is described in the following table.

© 2024 Microchip Technology Inc. and its subsidiaries

Table 5-15. DMA Register Map

DMA Memory Map per channel				
Channel Base Address			DMA Controller Base Address + (0x1000 × Channel ID)	
Offset	Width	Attributes	Register Name	Description
0x000	4B	RW	Control	Channel control register
0x004	4B	RW	NextConfig	Next transfer type
0x008	8B	RW	NextBytes	Number of bytes to move
0x010	8B	RW	NextDestination	Destination start address
0x018	8B	RW	NextSource	Source start address
0x104	4B	R	ExecConfig	Active transfer type
0x108	8B	R	ExecBytes	Number of bytes remaining
0x110	8B	R	ExecDestination	Destination current address
0x118	8B	R	ExecSource	Source current address

The following sections describe the Control and Status registers of a channel.

5.9.2 Control Register

The Control register stores the current status of the channel. It can be used to claim a DMA channel, initiate a transfer, enable interrupts, and to check for the completion of a transfer. The following table defines the bit fields of the Control register.

Table 5-16. Control Register (Control)

Register Offset		0x000 + (0x1000 × Channel ID)		
Bits	Field Name	Attributes	Reset	Description
0	claim	RW	0	Indicates that the channel is in use. Setting this bit clears all of the channel's Next registers (NextConfig, NextBytes, NextDestination, and NextSource). This bit can only be cleared when run (CR bit 0) is low.
1	run	RW	0	Setting this bit starts a DMA transfer by copying the Next registers into their Exec counterparts
[13:2]	Reserved	—	0	—
14	doneIE	RW	0	Setting this bit will trigger the channel's Done interrupt once a transfer is complete
15	errorIE	RW	0	Setting this bit will trigger the channel's Error interrupt upon receiving a bus error
[28:16]	Reserved	—	0	—
29	Reserved	—	0	—
30	done	RW	0	Indicates that a transfer has completed since the channel was claimed
31	error	RW	0	Indicates that a transfer error has occurred since the channel was claimed

5.9.3 Channel Next Configuration Register (NextConfig)

The read-write NextConfig register holds the transfer request type. The wsize and rsize fields are used to determine the size and alignment of individual DMA transactions as a single DMA transfer may require multiple transactions. There is an upper bound of 64B on a transaction size (read and write).

© 2024 Microchip Technology Inc. and its subsidiaries

Note: The DMA engine supports the transfer of only a single contiguous block at a time. Supports byte-aligned source and destination size (rsize and wsize) because the granularity is at the byte level in terms of only the base 2 Logarithm (1 byte, 8 byte, 32 byte).

These fields are WARL (Write-Any Read-Legal), so the actual size used can be determined by reading the field after writing the requested size. The DMA can be programmed to automatically repeat a transfer by setting the repeat bit field. If this bit is set, once the transfer completes, the Next registers are automatically copied to the Exec registers and a new transfer is initiated. The Control.run bit remains set during "repeated" transactions so that the channel cannot be claimed.

To stop repeating transfers, a master can monitor the channel's Done interrupt and lower the repeat bit accordingly.

Table 5-17. Channel Next Configuration Register

Register Offset		0x004 + (0x1000 × Channel ID)		
Bits	**Field Name**	**Attributes**	**Reset**	**Description**
[1:0]	Reserved	—	—	—
2	repeat	RW	0	If set, the Exec registers are reloaded from the Next registers once a transfer is complete. The repeat bit must be cleared by software for the sequence to stop.
3	order	RW	0	Enforces strict ordering by only allowing one of each transfer type in-flight at a time.
[23:4]	Reserved	—	—	—
[27:24]	wsize	WARL	0	Base 2 Logarithm of DMA transaction sizes. Example: 0 is 1 byte, 3 is 8 bytes, 5 is 32 bytes
[31:28]	rsize	WARL	0	Base 2 Logarithm of DMA transaction sizes. Example: 0 is 1 byte, 3 is 8 bytes, 5 is 32 bytes

5.9.4 Channel Next Bytes Register (NextBytes)

The read-write NextBytes register holds the number of bytes to be transferred by the channel. The NextConfig.xsize fields are used to determine the size of the individual transactions which will be used to transfer the number of bytes specified in this register. The NextBytes register is a WARL register with a maximum count that can be much smaller than the physical address size of the machine.

5.9.5 Channel Next Destination Register (NextDestination)

The read-write NextDestination register holds the physical address of the destination for the transfer.

5.9.6 Channel Next Source Address (NextSource)

The read-write NextSource register holds the physical address of the source data for the transfer.

5.9.7 Channel Exec Registers

Each DMA channel contains a set of Exec registers which hold the information about the currently executing transfer. These registers are Read-Only and initialized when the Control.run bit is set.

Upon initialization, all of the Next registers are copied into the Exec registers and a transfer begins. The status of the transfer can be monitored by reading the following Exec registers.

- ExecBytes: Indicates the number of bytes remaining in a transfer.
- ExecSource: Indicates the current source address.
- ExecDestination: Indicates the current destination address.

The base addresses of the above registers are listed in Table 5-15

© 2024 Microchip Technology Inc. and its subsidiaries

.

5.10 Write Combining Buffer (WCB)

WCB combines multiple consecutive writes to a given address range into a TileLink burst to increase the efficiency of Write transactions. Read transactions are bypassed by WCB. WCB accesses the 256 MB of non-cached DDR region via system port 4 AXI-NC as shown in the following table.

Table 5-18. WCB Address Range

WCB Address Range		
Base Address	Top	Port
0xD000_0000	0xDFFF_FFFF	System Port 4 (AXI4-NC)
0x18_0000_0000	0x1B_FFFF_FFFF	System Port 4 (AXI4-NC)

WCB manages its internal buffers efficiently based on the incoming Write/Read transaction addresses. The key properties of WCB are as follows:

- The WCB supports all single byte, multi-byte, and word writes (any single beat writes)
- Multi-beat transactions bypass WCB
- If all internal buffers are in use, and a write to a different base address occurs, the WCB may insert idle cycles while it empties a buffer

A buffer in WCB is also emptied under the following conditions:

- All bytes in the buffer have been written
- The buffer is not written for idle cycles
- A write to WCB address range followed by a read of the same address will cause a buffer to flush. The read is not allowed to pass through the WCB until the write has completed.
- A write from a different master that matches a buffer's base address
- A write from the same master to an already written byte(s) in the buffer

5.10.1 Idle Configuration Register

The Idle Configuration register specifies the number of idle cycles before a buffer is automatically emptied. WCB can be configured to be idle for up to 255 cycles.

When idle is set to 0, WCB is disabled and writes to the WCB address range bypass WCB.

Table 5-19. Idle Configuration Register

Idle Configuration Register (idle)				
Register Offset		0		
Bits	Field Name	Attributes	Reset	Description
[7:0]	idle	RW	16	Number of idle cycles before flushing a buffer. Setting to 0 disables WCB and all buffers are emptied.
[31:8]	Reserved	RW	X	—

5.11 Bus Error Unit

There is a Bus Error Unit (BEU) for each processor core. The address range of BEU 0, BEU 1, BEU 2, BEU 3, and BEU 4 is given in 11. Memory Map. BEUs record erroneous events in L1 instruction and data caches, and report them using the global and local interrupts. Each BEU can be configured to generate interrupts on L1 correctable and uncorrectable memory errors, including TileLink bus errors.

5.11.1 BEU Register Map

The register map of a BEU is listed in the following table.

© 2024 Microchip Technology Inc. and its subsidiaries

Table 5-20. BEU Register Map

Offset	Width	Attributes	Register Name	Description
0x000	1B	RW	cause	Cause of error event based on mhpmevent register (see Table 5-21).
0x008	1B	RW	value	Physical address of the error event
0x010	1B	RW	enable	Event enable mask
0x018	1B	RW	plic_interrupt	Platform level interrupt enable mask
0x020	1B	RW	accrued	Accrued event mask
0x028	1B	RW	local_interrupt	Local interrupt enable mask

5.11.2 Functional Description

The following table lists the mhpmevent[7:0] register bit fields which correspond to BEU events that can be reported.

Table 5-21. mhpmevent[7:0]

Cause	Meaning
0	No Error
1	Reserved
2	Instruction cache or ITIM correctable ECC error
3	ITIM uncorrectable error
4	Reserved
5	Load or store TileLink bus error
6	Data cache correctable ECC error
7	Data cache uncorrectable ECC error

When one of the events listed in the preceding table occurs, the BEU can record information about that event and can generate a global or local interrupt to the Hart. The enable register (Table 5-20) contains a mask of the events that can be recorded by the BEU. Each bit in the enable register corresponds to an event in Table 5-20. For example, if enable[3] is set, the BEU records uncorrectable ITIM errors.

The cause register indicates the event recorded most recently by the BEU. For example, a value of 3 indicates an uncorrectable ITIM error. The cause value 0 is reserved to indicate no error. The cause register is only written for events enabled in the enable register. The cause register is written when its current value is 0; that is, if multiple events occur, only the first one is latched, until software clears the cause register.

The value register holds the physical address that caused the event, or 0 if the address is unknown. The BEU writes to the value register whenever it writes the cause register. For example, when an event is enabled in the enable register and when the cause register contains 0.

The accrued register indicates all the events that occurred since the register was cleared by the software. Its format is the same as the enable register. The BEU sets bits in the accrued register whether or not they are enabled in the enable register.

The plic_interrupt register indicates the accrued events for which an interrupt must be generated via the PLIC. An interrupt is generated when any bit is set in accrued and plic_interrupt register. For example, when accrued and plic_interrupt is not 0.

The local_interrupt register indicates the accrued events for which an interrupt must be generated directly to the Hart. An interrupt is generated when any bit is set in both accrued and local_interrupt registers. For example, when accrued and local_interrupt is not 0.

© 2024 Microchip Technology Inc. and its subsidiaries

The interrupt cause is 128; it does not have a bit in the `mie` CSR, so it is always enabled; nor does it have a bit in the `mideleg` CSR, so it cannot be delegated to a mode less privileged than M-mode.

© 2024 Microchip Technology Inc. and its subsidiaries

6. Peripheral Descriptions

6.1 I²C

Philips Inter-Integrated Circuit (I²C) is a two-wire serial bus interface that provides data transfer between many devices. PIC64GX contains two identical I²C peripherals in the microprocessor subsystem (I2C_0 and I2C_1), that provide a mechanism for serial communication between the PIC64GX device and the external I²C compliant devices.

PIC64GX I²C peripherals support the following protocols:

- I²C protocol v2.1 specification
- SMBus protocol v2.0 specification
- PMBus protocol v1.1 specification

6.1.1 Features

I²C peripherals support the following features:

- Master and Slave modes
- 7-bit addressing format and data transfers up to 100 Kbit/s in Standard mode and up to 400 Kbit/s in Fast mode
- Multi-master collision detection and arbitration
- Own slave address and general call address detection
- Second slave address detection
- System management bus (SMBus) time-out and real-time idle condition counters
- Optional SMBus signals, SMBSUS_N, and SMBALERT_N, which are controlled through the APB interface
- Input glitch or spike filters

The I²C peripherals are connected to the AMBA interconnect through the advanced peripheral bus (APB) interfaces.

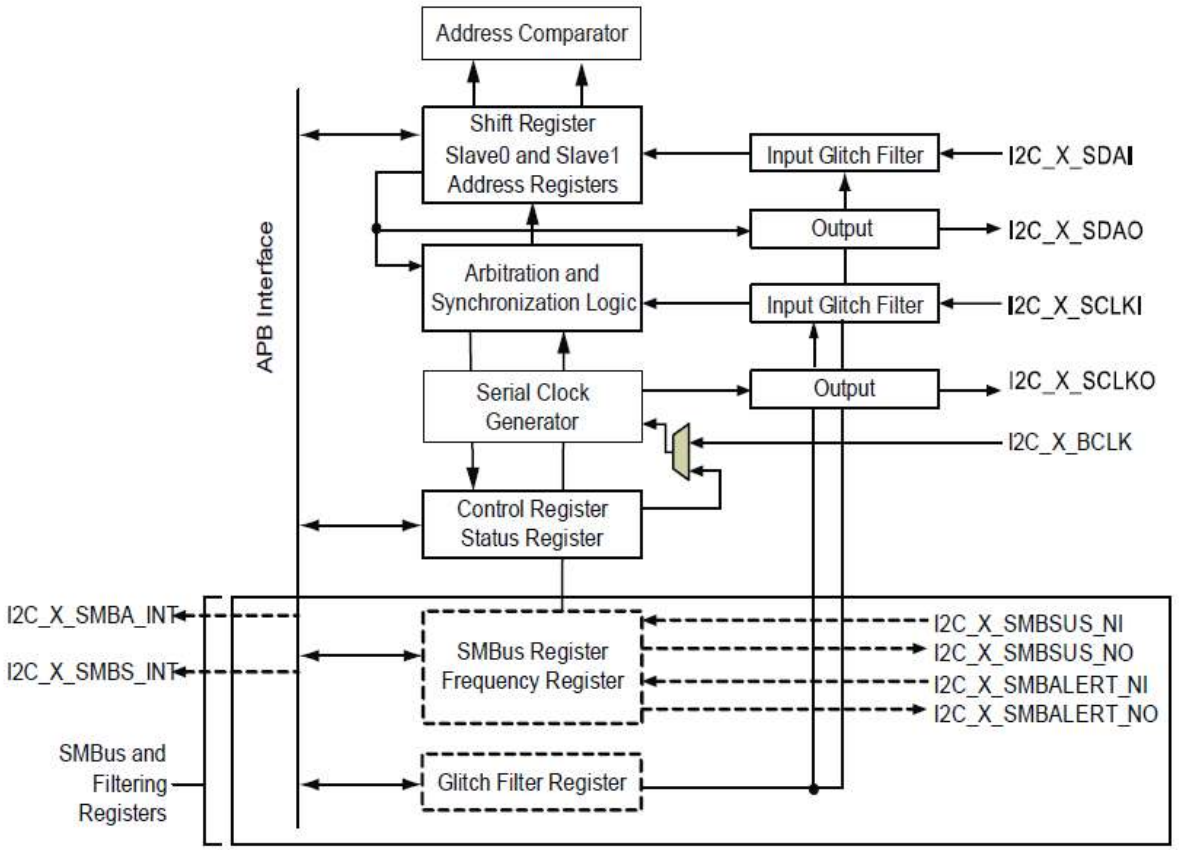
6.1.2 Functional Description

The I²C peripherals consist mainly of the following components. See the following figure.

- Input Glitch Filter
- Arbitration and Synchronization Logic
- Address Comparator

© 2024 Microchip Technology Inc. and its subsidiaries

Figure 6-1. I²C Block Diagram



6.1.3 Input Glitch Filter

The I²C Fast mode (400 Kbit/s) specification states that glitches 50 ns or less should be filtered out of the incoming clock and data lines. The input glitch filter performs this function by filtering glitches on incoming clock and data signals. Glitches shorter than the glitch filter length are filtered out. The glitch filter length is defined in terms of APB interface clock cycles and configurable from 3 to 21 APB interface clock cycles. Input signals are synchronized with the internal APB interface clock.

6.1.4 Arbitration and Synchronization

In Master mode, the arbitration logic monitors the data line. If any other device on the bus drives the data line Low, the I²C peripheral immediately changes from Master-Transmitter mode to Slave-Receiver mode. The synchronization logic synchronizes the serial clock generator block with the transmitted clock pulses coming from another master device.

The arbitration and synchronization logic implements the time-out requirements as per the SMBus specification version 2.0.

6.1.5 Address Comparator

When a master transmits a slave address on the bus, the address comparator checks the 7-bit slave address with its own slave address. If the transmitted slave address does not match, the address comparator compares the first received byte with the general call address (0x00). If the address matches, the STATUS Register is updated. The general call address is used to address each device connected to the I²C bus.

© 2024 Microchip Technology Inc. and its subsidiaries

6.1.6 Serial Clock Generator

In Master mode, the serial clock generator generates the serial clock line (SCL). The clock generator is switched OFF when I²C is in Slave mode.

I²C uses APB clock to generate the serial clock. See the `MPU_I2C_init()` function in the MPU I2C Driver . This driver provides access to the I²C : I2C0_CTRL control register, which configures I²C serial clock using the provided divider value to generate the serial clock from the APB clock.

6.1.7 Register Maps and Descriptions

See the configuration and register maps in the ZIP document on the PIC64GX1000 product page www.microchip.com/PIC64GX1000.

6.1.8 Switching Characteristics

The following table describes I²C switching characteristics.

Table 6-1. I²C Switching Characteristics

Parameter	Symbol	Min	Typ	Max	Units	Conditions
Input low voltage	V_{IL}	–0.3	—	0.8	V	See Single-Ended I/O Standards for more information. I/O standard used for illustration: MPUIO bank–LVTTL 8 mA low drive.
Input high voltage	V_{IH}	2	—	3.45	V	See Single-Ended I/O Standards for more information. I/O standard used for illustration: MPUIO bank–LVTTL 8 mA low drive.
Hysteresis of Schmitt Trigger inputs for V_{DDI} > 2V	V_{hys}	—	0.05 x V_{DDI}	—	V	—
Output low voltage (open drain) at 3 mA sink current for V_{DDI} > 2V	V_{OL}	—	—	0.4	V	See Single-Ended I/O Standards for more information. I/O standard used for illustration: MPUIO bank–LVTTL 8 mA low drive.
Rise time for input clock and data	t_r	—	—	1000	ns	Standard mode
		—	—	300	ns	Fast mode
Output fall time from V_{ihmin} to V_{ilmax}	t_{fo}	—	21	—	ns	C_{LOAD} = 400 pF
		—	6	—	ns	C_{LOAD} = 400 pF
SCL clock frequency	F_{I2C}	—	—	400	KHz	Fast mode
		—	—	100	KHz	Standard mode
Input capacitance	C_i	—	—	10	pF	V_{IN} = 0, f = 1.0 MHz
Capacitive load for each bus line	C_b	—	—	400	pF	—
I2C MPU input clock	PCLK	156.25	—	156.25	MHz	—
Low period of SCL clock	T_{I2CL}	1	—	—	PCLK cycles	—
High period of SCL clock	T_{I2CH}	1	—	—	PCLK cycles	—
Setup time for a repeated START condition	$T_{su(start)}$	1	—	—	PCLK cycles	—

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

Parameter	Symbol	Min	Typ	Max	Units	Conditions
Hold time for a repeated START condition (after this period, the first clock pulse is generated)	$T_{h(start)}$	1	—	—	PCLK cycles	—
Data setup time	$T_{su(data)}$	1	—	—	PCLK cycles	—
Data input hold time	$T_{h(data)}$	1	—	—	PCLK cycles	—
Data output delay time	$T_{od(data)}$	1	—	—	PCLK cycles	—
Setup time for STOP condition	$T_{su(stop)}$	1	—	—	PCLK cycles	—
Bus free time between a STOP and START condition	T_{buf}	1	—	—	PCLK cycles	—

6.2 USB OTG

Universal serial bus (USB) is an industry standard that defines cables, connectors, and serial communication protocol used in a bus for connection, communication, and power supply between electronic devices. The PIC64GX device contains a USB On-The-Go (OTG) controller as part of the microprocessor subsystem (MPU). USB OTG controller provides a mechanism for the USB communication between the PIC64GX device and external USB host/USB device/USB OTG protocol compliant devices.

6.2.1 USB Features

USB OTG controller supports the following features:

- Operates as a USB host in a point-to-point or multi-point communication with other USB devices
- Operates as a USB peripheral with other USB hosts
- Compliant with the USB 2.0 standard and includes OTG supplement
- Supports USB 2.0 speeds:
  - High speed (480 Mbps)
  - Full speed (12 Mbps)
- Supports session request protocol (SRP) and host negotiation protocol (HNP)
- Supports suspend and resume signaling
- Supports multi-point capabilities
- Supports four direct memory access (DMA) channels for data transfers
- Supports high bandwidth isochronous (ISO) pipe enabled endpoints
- Hardware selectable option for 8-bit/4-bit Low Pin Count Interface (LPI)
- Supports ULPI hardware interface to external USB physical layer (PHY)
- Soft connect/disconnect
- Configurable for up to five transmit endpoints (TX EP) and up to five receive endpoints (RX EP), including control endpoint (EP0)
- Offers dynamic allocation of endpoints, to maximize the number of devices supported
- Internal memory of 8 KB with support for dynamic allocation to each endpoint
- Performs all USB 2.0 transaction scheduling in hardware

© 2024 Microchip Technology Inc. and its subsidiaries

- Supports link power management
- SECDED protection on the internal USB memory with the following features:
  - Generates interrupts on 1-bit or 2-bit errors; these interrupts can be masked
  - Corrects 1-bit errors
  - Counts the number of 1-bit and 2-bit errors

For more information on USB 2.0 and OTG protocol specifications, see the following web pages:

- www.usb.org/developers/docs/
- www.usb.org/developers/onthego/

The USB OTG controller can function as an AHB master for DMA data transfers and as an AHB slave for configuring the USB OTG controller from the masters processor.

The USB OTG controller can function as one of the following:

- A high speed or a full speed peripheral USB device attached to a conventional USB host (such as a PC)
- A point-to-point or multi-point USB host
- An OTG device that can dynamically switch roles between the host and the device

In all cases (USB host, USB device, or USB OTG), USB OTG controller supports control, bulk, ISO, and interrupt transactions in all three modes

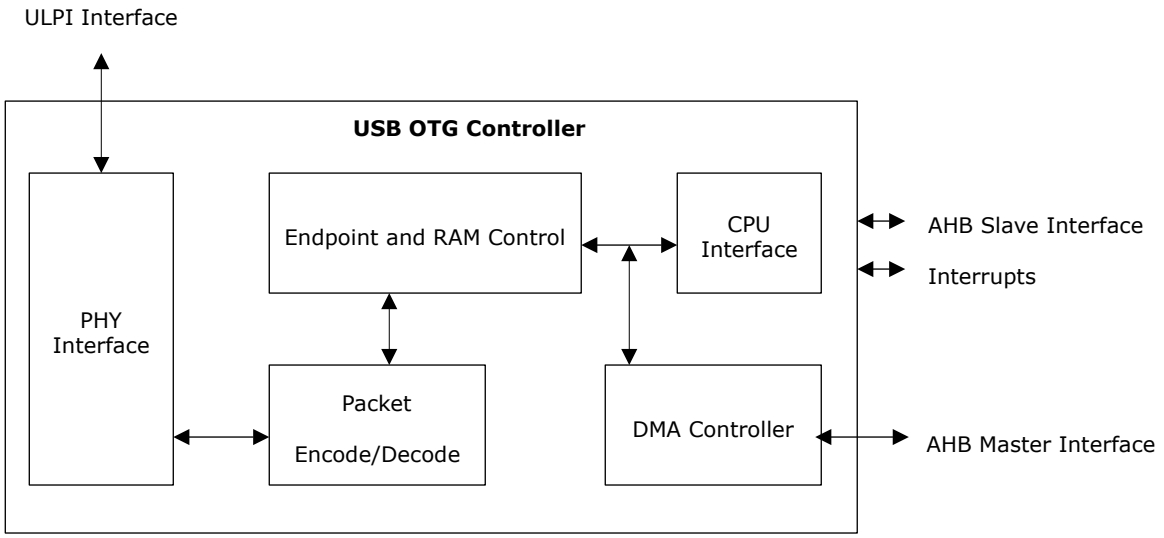
6.2.2 Functional Description

The following block diagram highlights the main blocks in the USB OTG controller. The USB OTG controller is interfaced through the AMBA interconnect in the MPU. The USB OTG controller provides an ULPI interface to connect to the external PHY. Following are the main component blocks in the USB OTG controller:

- AHB Master and Slave Interfaces
- CPU Interface
- Endpoints (EP) Control Logic and RAM Control Logic
- Packet Encoding, Decoding, and CRC Block
- PHY Interfaces

© 2024 Microchip Technology Inc. and its subsidiaries

Figure 6-2. USB OTG Controller Block Diagram



6.2.3 AHB Master and Slave Interfaces

The USB OTG controller functions as both AHB master and AHB slave on the AMBA interconnect. The AHB master interface is used by the DMA engine, which is built into the USB OTG controller, for data transfer between memory in the USB OTG controller and the system memory. The AHB slave interface is used by other masters to configure registers in the USB OTG controller.

6.2.4 CPU Interface

USB OTG controller send interrupts to the processor using the CPU interface. The USB OTG controller send interrupts for the following events:

- When packets are transmitted or received
- When the USB OTG controller enters Suspend mode
- When USB OTG controller resumes from Suspend mode

The CPU interface block contains the common configuration registers and the interrupt control logic for configuring the OTG controller.

6.2.5 Endpoints Control Logic and RAM Control Logic

These two blocks constitute buffer management for the data buffers in Host mode and in Device mode. This block manages endpoint buffers and their properties, called pipes, which are defined by control, bulk, interrupt, and ISO data transfers. Data buffers in Device mode (endpoints) and in Host mode are supported by the SECDED block, which automatically takes care of single-bit error correction and dual-bit error detection. This SECDED block maintains the counters for the number of single-bit corrections made and the number of detections of dual-bit errors. The SECDED block is provided with the interrupt generation logic. If enabled, this block generates the corresponding interrupts to the processor.

6.2.6 Packet Encoding, Decoding, and CRC Block

This block generates the CRC for packets to be transmitted and checks the CRC on received packets. This block generates the headers for the packets to be transmitted and decodes the headers on received packets. There is a 16-bit CRC for the data packets and a 5-bit CRC for control and status packets.

© 2024 Microchip Technology Inc. and its subsidiaries

6.2.7 PHY Interfaces

The USB OTG controller supports Universal Low Pin Count Interface (ULPI) at the link side. For a ULPI interface, the I/Os are routed through the MPU onto multi-standard I/Os.

6.2.8 Register Maps and Descriptions

See the configuration and register maps in the ZIP document on the PIC64GX1000 product page www.microchip.com/PIC64GX1000.

6.2.9 Switching Characteristics

The following table describes the USB. Test conditions are LVCMOS33, slow slew rate, 8 mA drive strength, 15 pF loads, and 60 MHz device clock frequency.

Table 6-2. USB

Parameter	Symbol	Min	Max	Units
Input setup to ULPI clocks, all inputs	TULPIDCK	4.5	—	ns
Input hold to ULPI clocks, all inputs	TULPICKD	0.0	—	ns
ULPI clock to poutput valid, all outputs	TULPICKO	2.0	8.5	ns
ULPI device clock frequency	FULPICLK	60	60	MHz

6.3 eNVM Controller

The PIC64GX devices include one embedded non-volatile memory (eNVM) block size of 128 KB. The eNVM controller interfaces the eNVM block to the AMBA interconnect.

6.3.1 Features

eNVM supports the following features:

- SECDED protected
- High Data Retention Time
- 32-bit data input and 64-bit data output

6.3.2 Functional Description

The eNVM controller implements an AHB interface to the eNVM R and C interfaces. The C-Bus (32-bit) is used for programming operations and the R-Bus (64-bit) for read operations.

The eNVM controller operates at the AHB clock, and generates a slower clock for the eNVM whose maximum clock rate is 26.3 MHz. This is achieved by creating a clock pulse that is a multiple of the master clock that supports an NVM access time of up to 80 ns.

To minimize clock synchronization latency, the AHB controller only generates an eNVM clock when it needs access or the eNVM requests a clock. This allows the AHB controller to send the address to the eNVM as soon as it is ready as it can restart the clock at any AHB clock cycle.

6.3.3 Data Retention Time

The following table shows the retention time of the eNVM with respect to the junction temperature.

Table 6-3. Data Retention Time

Junction Temperature	Data Retention	Write Cycles
110 °C	10 years	10000
125 °C	4 years	1000

© 2024 Microchip Technology Inc. and its subsidiaries

6.3.4 eNVM Access Time Speed

Table 6-4. eNVM Access Timing

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Plain text programming	—	7.0	7.2	7.9	ms	—
Authenticated text programming	—	7.2	7.4	9.4	ms	—
Authenticated and encrypted text programming	—	7.2	7.4	9.4	ms	—
Authentication R/W first access from power-up overhead	TPUF_OVHD	10	13	111	ms	From T_{FAB_READY}
Plain text read	—	8	8.5	9	µs	—
Authenticated text read	—	113	114.5	119	µs	—
Authenticated and decrypted text read	—	159	161	167	µs	—

Notes:

- Page size = 256 bytes (non-authenticated), 236 bytes (authenticated).
- Only page reads and writes allowed.
- T_{PUF_OVHD} is an additional time that occurs on the first R/W, after cold or warm boot, to sNVM using authenticated or authenticated and encrypted text.

6.3.5 R-Bus Access

The AHB controller interfaces the 32-bit AHB bus to the 64-bit R (Read) interface on the eNVM. The controller always reads 64 bits from the eNVM and stores the data in case there is a subsequent read requests data from the same 64-bit location.

When an AHB read request is made, the controller checks whether the data for the requested address is held in the buffer and returns the data.

6.3.6 C-Bus Access

The AHB controller simply maps the AHB read/write operations directly to the C-Bus signals. The controller stalls write operations until the eNVM indicates that it is ready (c_grant asserted) and then asserts HREADY, this releases the MPU Core Complex Processor while the eNVM completes any required operations. If a second operation is requested, it is stalled until the eNVM re-asserts the c_grant signal.

6.3.7 eNVM Address and Segments

The eNVM consists of four segments mapped into a contiguous 128 KB address space as listed in the following table. The C-Bus provides eNVM configuration, read/write capability. The R-Bus allows reading of the eNVM over AHB.

Table 6-5. eNVM Segments and Addresses

Bus	Size		Access	Offset	Description	Address
C-Bus	512 Bytes		RW	0x00000000	Configuration	0x20200000
R-Bus	128 K Bytes	8K	RO	0x00000000	Sector 2	0x20220000
		56K	RO	0x00002000	Sector 0	0x20222000
		56K	RO	0x00010000	Sector 1	0x20230000
		8K	RO	0x0001E000	Sector 3	0x2023E000

6.3.8 eNVM Access Capabilities

The eNVM is an optional boot ROM for the MPU. For the MPU boot process, eNVM is used to store a baremetal application or a Zero Stage Boot Loader (ZSBL). The eNVM programming is executed

© 2024 Microchip Technology Inc. and its subsidiaries

during the device programming through JTAG. The eNVM read access is available to the System Controller to support MPU boot. The MPU CPU Core Complex can read eNVM through the R-Bus. However, CPU Core Complex eNVM write is not supported.

6.3.9 Register Map

See the configuration and register maps in the ZIP document on the PIC64GX1000 product page www.microchip.com/PIC64GX1000.

6.4 SPI

Serial peripheral interface (SPI) is a synchronous serial data protocol that enables the microprocessor and peripheral devices to communicate with each other. The SPI controller is an APB slave in the PIC64GX device that provides a serial interface compliant with the Motorola SPI, Texas Instruments synchronous serial, and National Semiconductor MICROWIRE™ formats. In addition, SPI supports interfacing with large SPI Flash and EEPROM devices and a hardware-based slave protocol engine. PIC64GX contains two identical SPI controllers SPI_0 and SPI_1 in the microprocessor subsystem.

6.4.1 Features

SPI peripherals support the following features:

- Master and Slave modes
- Configurable Slave Select operation
- Configurable clock polarity
- Separate transmit (Tx) and receive (Rx) FIFOs to reduce interrupt service loading

6.4.2 SPI Functional Description

The SPI controller supports Master and Slave modes of an operation.

- In Master mode, the SPI generates SPI_X_CLK, selects a slave using SPI_X_SS, transmits the data on SPI_X_DO, and receives the data on SPI_X_DI.
- In Slave mode, the SPI is selected by SPI_X_SS. The SPI receives a clock on SPI_X_CLK and incoming data on SPI_X_DI.

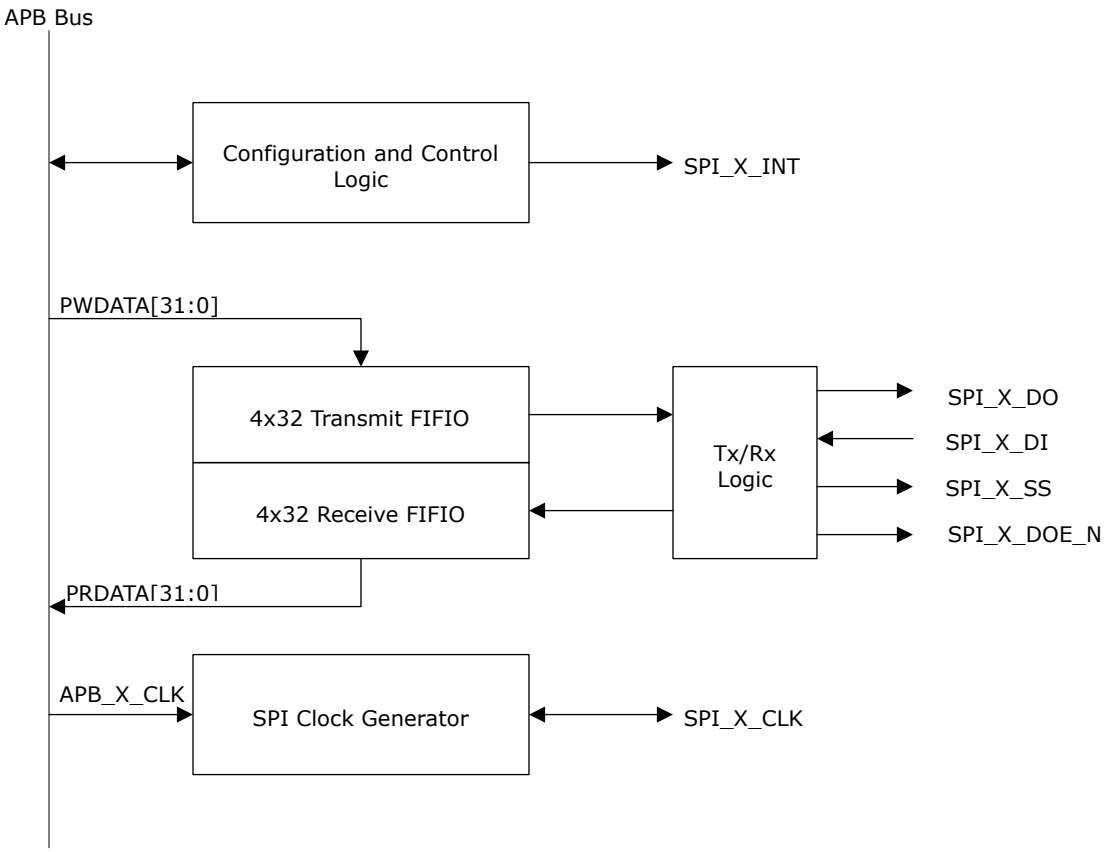
The SPI peripherals consist mainly of the following components as shown in Figure 6-3.

- Transmit and receive FIFOs
- Configuration and control logic
- SPI clock generator

The following figure shows the SPI controller block diagram.

© 2024 Microchip Technology Inc. and its subsidiaries

Figure 6-3. SPI Controller Block Diagram



Notes:

- The SPI_X_DO, SPI_X_DI, SPI_X_SS, and SPI_X_CLK signals are available to the MPU.
- SPI_X_DOE_N is accessible through the SPI control register.
- SPI_X_INT is sent to the MPU Core Complex.

Note: X is used as a place holder for 0 or 1 in the register and signal descriptions. It indicates SPI _0 (on the APB_0 bus) or SPI_1 (on the APB_1 bus).

6.4.3 Transmit and Receive FIFOs

The SPI controller embeds two 4 × 32 (depth × width) FIFOs for receive and transmit, as shown in Figure 6-3. These FIFOs are accessible through RX data and TX data registers. Writing to the TX data register causes the data to be written to the transmit FIFO. This is emptied by the transmit logic. Similarly, reading from the RX data register causes the data to be read from the receive FIFO.

6.4.4 Configuration and Control Logic

The SPI peripheral can be configured for Master or Slave mode by using the Mode bit of the SPI CONTROL register. This type of data transfer protocol can be configured by using the TRANSFPRTL bit of the SPI CONTROL register. The control logic monitors the number of data frames to be sent or received and enables the interrupts when the data frame transmission or reception is completed. During data frames transmission or reception, if a transmit under-run error or receive overflow error is detected, the STATUS Register is updated.

© 2024 Microchip Technology Inc. and its subsidiaries

6.4.5 SPI Clock Generator

In Master mode, the SPI clock generator generates the serial programmable clock from the APB clock.

6.4.6 Register Map

See the configuration and register maps in the ZIP document on the PIC64GX1000 product page www.microchip.com/PIC64GX1000.

6.4.7 Switching Characteristics

The following tables describe the SPI Initiator and Target mode switching characteristics. The test conditions are configured to the LVCMOS 3.3V I/O standard with a 12 mA drive strength, fast slew rate, and a 30 pF load.

Table 6-6. SPI Initiator Mode Switching Characteristics

Parameter	Symbol	Min	Max	Units	Condition
SCK frequency	sp1i_MPU	—	20	MHz	CPU at 667 MHz (– 1), 5 ns Flash access time
SCK minimum pulse width high	sp2i_MPU	SCK_period/2	—	ns	—
SCK minimum pulse width low	sp3i_MPU	SCK_period/2	—	ns	—
Rise and fall	sp4i_MPU	—	—	ns	
times	sp5i_MPU				
SCK to SDO	sp6i_MPU	–6.0	7.0	ns	—
SDI setup time to SCK	sp7i_MPU	10.0	—	ns	—
SDI hold time from SCK	sp8i_MPU	0.0	—	ns	—

The following timing diagram specifies the meaning of the various SPI timing parameters in Initiator and Target modes.

© 2024 Microchip Technology Inc. and its subsidiaries

Figure 6-4. SPI Timing Diagram for a Single Frame Transfer in Motorola Mode (SPH=1)

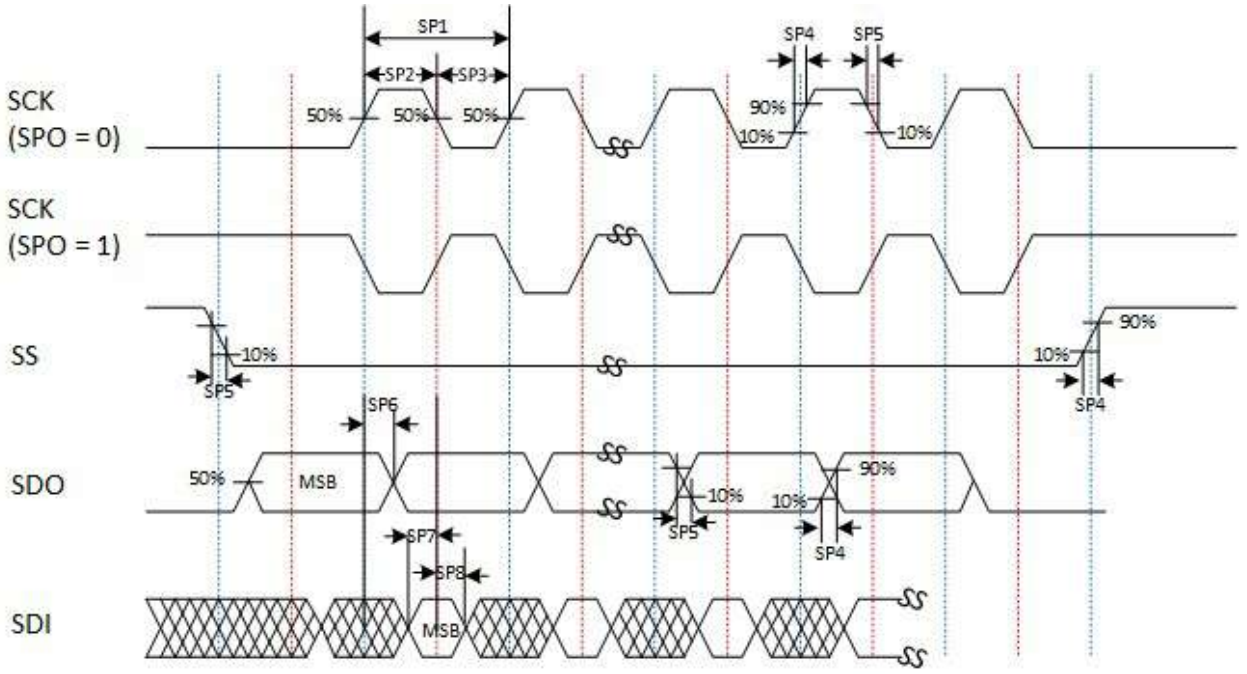


Table 6-7. SPI Target Mode Switching Characteristics

Parameter	Symbol	Min	Max	Units	Condition
SCK frequency	sp1t_MPU	—	50	MHz	External initiator with 6.0 ns SCK to data out
SCK minimum pulse width high	sp2t_MPU	SCK_period/2	—	ns	—
SCK minimum pulse width low	sp3t_MPU	SCK_period/2	—	ns	—
Rise and fall time	sp4t_MPU	—	—	ns	
	sp5t_MPU				IBIS models[1]
SCK to SDO	sp6t_MPU	3.0	13.0	ns	—
SDI setup to SCK	sp7t_MPU	4.0	—	ns	—
SDI hold to SCK	sp8t_MPU	3.0	—	ns	—

6.5 MMUART

Multi-mode universal asynchronous/synchronous receiver/transmitter (MMUART) performs serial-to-parallel conversion on data originating from modems or other serial devices, and performs parallel-to-serial conversion on data from the MPU Core Complex processor to these devices. The PIC64GX1000 contains five identical MMUART peripherals in the microprocessor subsystem (MMUART_0, MMUART_1, MMUART_2, MMUART_3, and MMUART_4).

6.5.1 Features

MMUART supports the following features:

- Asynchronous and synchronous operations
- Full programmable serial interface characteristics
  - Data width is programmable to 5, 6, 7, or 8 bits

© 2024 Microchip Technology Inc. and its subsidiaries

  - Even, odd, or no-parity bit generation/detection
  - 1, 1½, and 2 stop bit generation
- 9-bit address flag capability used for multi-drop addressing topologies
- Separate transmit (Tx) and receive (Rx) FIFOs to reduce processor interrupt service loading
- Single-wire Half-Duplex mode in which Tx pad can be used for bidirectional data transfer
- Local Interconnect Network (LIN) header detection and auto-baud rate calculation
- Communication with ISO 7816 smart cards
- Fractional baud rate capability
- Return to Zero Inverted (RZI) mod/demod blocks that allow infrared data association (IrDA) and serial infrared (SIR) communications
- The MSb or the LSb is the first bit while sending or receiving data

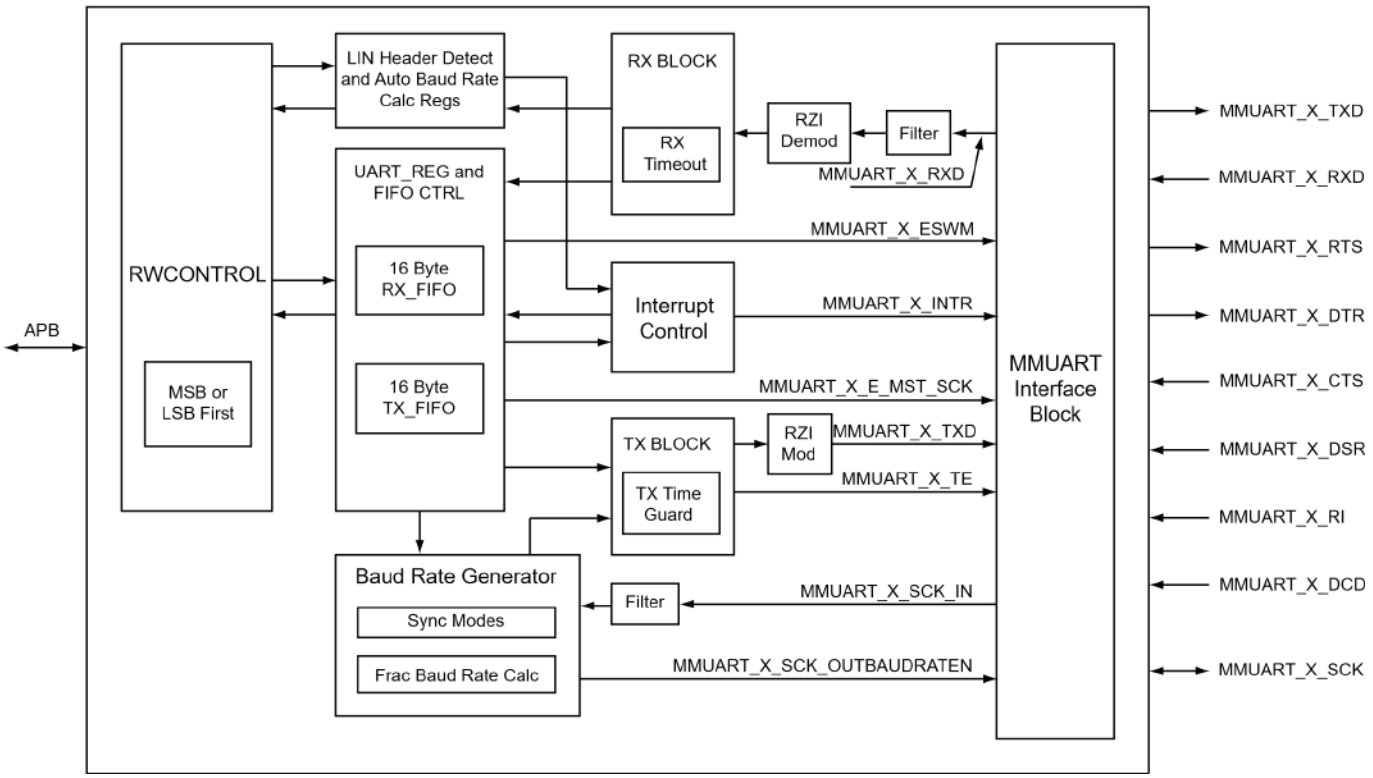
6.5.2 Functional Description

The functional block diagram of MMUART is shown in the following diagram. The main components of MMUART include Transmit and Receive FIFOs (TX FIFO and RX FIFO), Baud Rate Generator (BRG), input filters, LIN Header Detection and Auto Baud Rate Calculation block, RZI modulator and demodulator, and interrupt controller.

While transmitting data, the parallel data is written to TX FIFO of the MMUART to transmit in serial form. While receiving data to RX FIFO, the MMUART transforms the serial input data into parallel form to facilitate reading by the processor.

The Baud Rate Generator contains free-running counters and utilizes the asynchronous and synchronous baud rate generation circuits. The input filters in MMUART suppress the noise and spikes of incoming clock signals and serial input data based on the filter length. The RZI modulation/demodulation blocks are intended to allow for IrDA serial infrared (SIR) communications.

Figure 6-5. MMUART Block Diagram



© 2024 Microchip Technology Inc. and its subsidiaries

6.5.3 Register Map

For the base addresses and register descriptions of MMUART_0, MMUART_1, and MMUART_2, see the configuration and register maps in the ZIP document on the PIC64GX1000 product page www.microchip.com/PIC64GX1000.

The following table describes MMUART.

Table 6-8. MMUART

Parameter	Symbol	Min	Max	Units
Transmit baud rate	$BAUD_{TXMAX}$	—	6.25	Mbps
Receive baud rate	$BAUD_{RXMAX}$	—	6.25	Mbps
UART reference clock frequency	$F_{UART\_REF\_CLK}$	156.25	156.25	MHz

6.6 Quad SPI with XIP

Quad Serial Peripheral Interface (QSPI) is a synchronous serial data protocol that enables the microprocessor and peripheral devices to communicate with each other. The QSPI controller is an AHB slave in the MPU that provides a serial interface compliant with the Motorola SPI format. QSPI with execute in place (XIP) support allows a processor to directly boot rather than moving the SPI content to SRAM before execution.

6.6.1 Features

Quad SPI supports the following features:

- Master only operation with SPI data-rate
  - Programmable SPI clock—HCLK/2, HCLK/4, or HCLK/6
  - Maximum data-rate is HCLK/2
- FIFOs
  - Transmit and Receive FIFO
  - 16-byte transmit FIFO depth
  - 32-byte receive FIFO depth
  - AHB interface transfers up to four bytes at a time
- SPI Protocol
  - Master operation
  - Motorola SPI supported
  - Slave Select operation in idle cycles configurable
  - Extended SPI operation (1, 2, and 4-bit)
  - QSPI operation (4-bit operation)
  - BSPI operation (2-bit operation)
  - Execute in place (XIP)
  - Three or four-byte SPI address.
- Frame Size
  - 8-bit frames directly
  - Back-to-back frame operation supports greater than 8-bit frames
  - Up to 4 GB Transfer (2 × 32 bytes)
- Processor overhead reduction
  - SPI Flash command/data packets with automatic data generation and discard function

© 2024 Microchip Technology Inc. and its subsidiaries

- Direct Mode
  - Allows a CPU to directly control the SPI interface pins

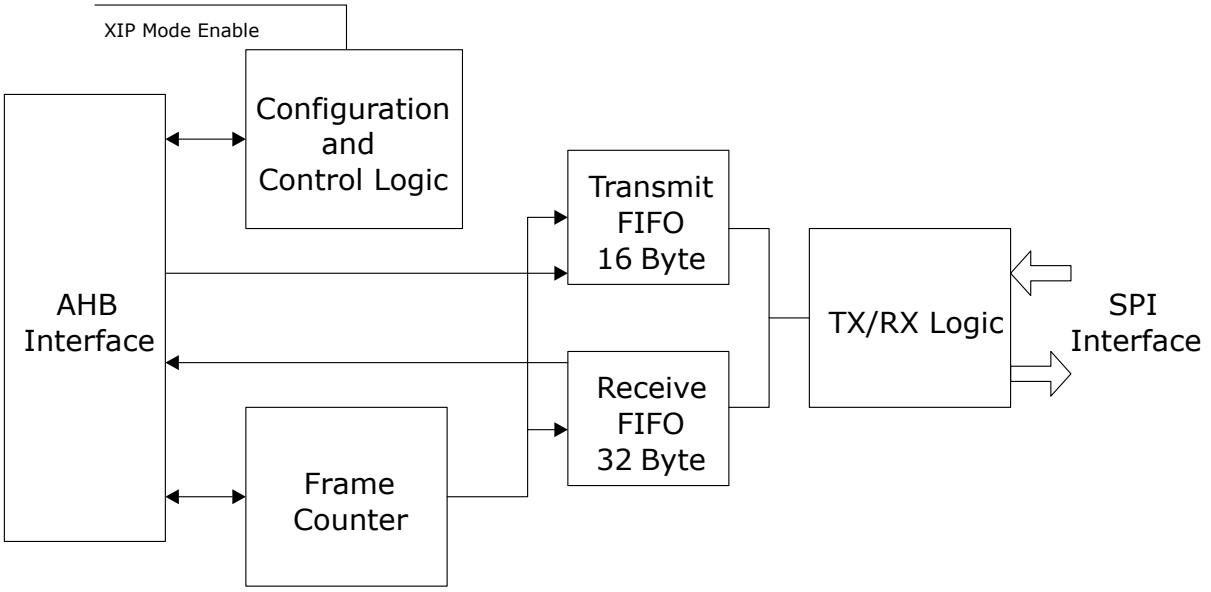
6.6.2 Functional Description

The QSPI controller supports only Master mode operation. The Master mode operation runs directly off the controller clock (HCLK) and supports SPI transfer rates at the HCLK/2 frequency and slower.

The SPI peripherals consist mainly of the following components.

- Transmit and receive FIFOs
- Configuration and control logic

Figure 6-6. QSPI Controller Block Diagram



6.6.3 Transmit and Receive FIFOs

The QSPI controller embeds two FIFOs for receive and transmit, as shown in Figure 6-6. These FIFOs are accessible through ReceiveData and TransmitData registers. Writing to the TransmitData register causes the data to be written to the transmit FIFO. This is emptied by the transmit logic. Similarly, reading from the ReceiveData register causes the data to be read from the receive FIFO.

6.6.4 Configuration and Control Logic

The SPI peripheral is configured for master-only operation. The type of data transfer protocol can be configured by using the QSPIMODE0 and QSPIMODE21 bits of the CONTROL register. The control logic monitors the number of data frames to be sent or received and enables the XIP mode when the data frame transmission or reception is completed. During data frames transmission/reception, if a transmit under-run error or receive overflow error is detected, the STATUS register is updated.

6.6.5 XIP Operation

Execute in place (XIP) allows a processor to directly boot from the QSPI device rather than moving the SPI content to SRAM before execution. A system configuration bit (XIP bit in CONTROL register) is used to set the controller in XIP mode.

When QSPI is in XIP mode, all AHB reads simply return the 32-bit data value associated with the requested address. Each access to the QSPI device requires a 3-byte or 4-byte address transfers, a

© 2024 Microchip Technology Inc. and its subsidiaries

3-byte IDLE period and 4-byte data transfer. Assuming the SPI clock is ¼ of the AHB clock, then this requires approximately 80 clock cycles per 32-bit read cycle. In XIP mode, data is returned directly to the AHB bus in response to an AHB read, data is not read from the FIFOs. The QSPI device stays in XIP mode as long as the Xb bit is zero.

In XIP mode, AHB write cycles access the core registers allowing the values to change, although the registers cannot be read when in XIP mode.

In the application, the XIP mode is not enabled at Reset as the CPUs are initially booted by system controller and the boot code can initialize the normal QSPI configuration registers.

To exit XIP mode, the firmware should clear the XIP bit in the CONTROL register, at this time it should not be executing from the QSPI device. When this bit is written to zero, the QSPI core returns to Normal mode and the reads access the core registers.

6.6.6 Register Map

See the configuration and register maps in the ZIP document on the PIC64GX1000 product page www.microchip.com/PIC64GX1000.

6.6.7 QSPI Switching Characteristics

The following table describes the QSPI switching characteristics.

Table 6-9. QSPI Switching Characteristics

Parameter	Symbol	Min	Max	Unit	Condition
SCK frequency	sp1_qspi	—	39	MHz	CLKRATE=2, CPU at 625 MHz (–STD)
			41.625	MHz	CLKRATE=2, CPU at 667 MHz (–1)
SCK minimum pulse width high	sp2_qspi	SCK_period/2	—	ns	—
SCK minimum pulse width low	sp3_qspi	SCK_period/2	—	ns	—
Rise and fall times	sp4_qspi sp5_qspi	—	—	ns	—
SCK to SDO	sp6_qspi	-1.0	4.0	ns	—
SDI setup time to SCK	sp7_qspi	10.0 - (SCK_period/ (2*CLKRATE))	—	ns	—
SDI hold time from SCK	sp8_qspi	SCK_period / (2*CLKRATE)	—	—	—

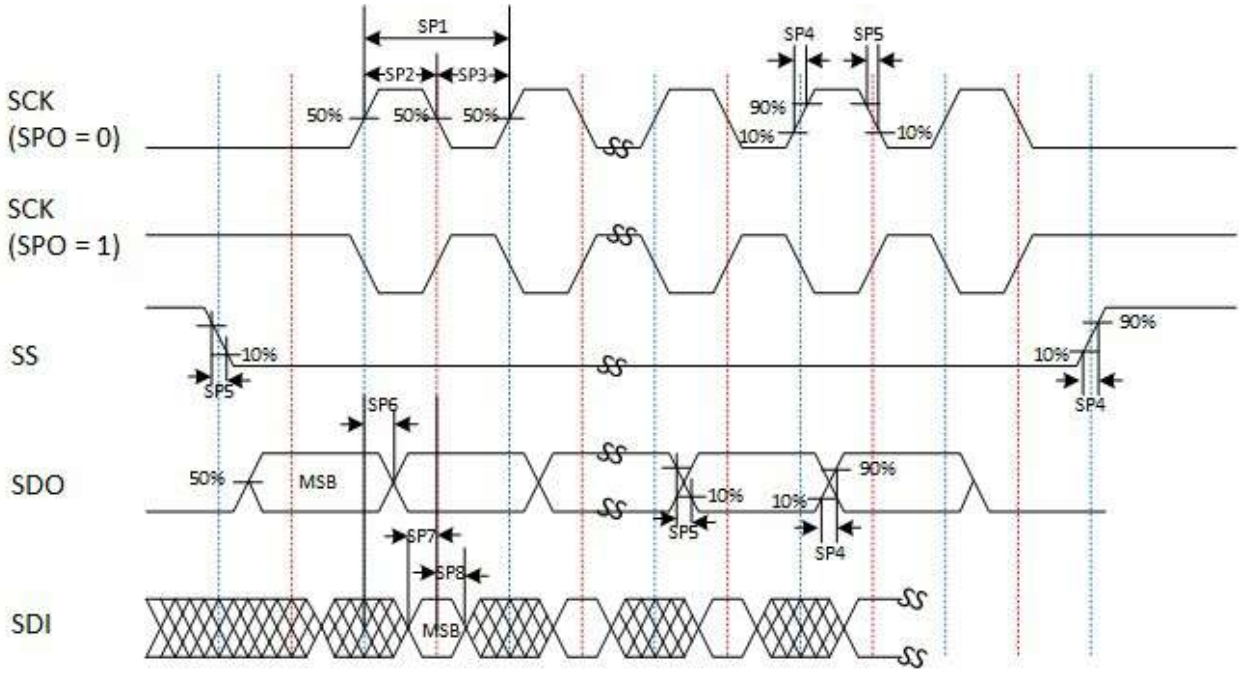
Note: CLKRATE refers to the value of the corresponding field in the appropriate QSPI register.

For specific rise/fall times, board design considerations, and detailed output buffer resistances, use the corresponding IBIS models located online at Microchip.com.

The following timing diagram specifies the meaning of the various QSPI timing parameters.

© 2024 Microchip Technology Inc. and its subsidiaries

Figure 6-7. QSPI Timing for a Single Frame Transfer



6.7 CAN

The PIC64GX1000 MPU contains an integrated control area network (CAN) peripheral. It is an APB slave on the AMBA interconnect. The MPU Core Complex configures the CAN controller through the APB slave interface.

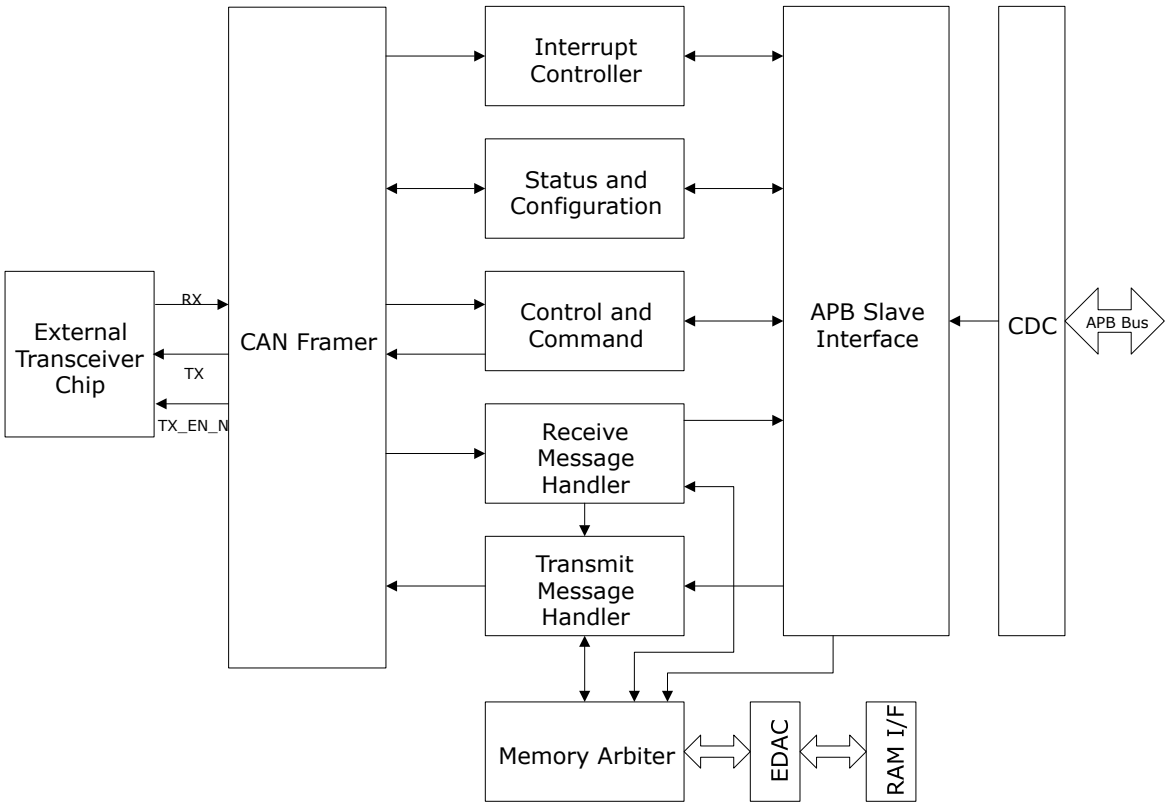
The CAN controller in the PIC64GX1000 supports the concept of mailboxes and contains 32 receive buffers. Each buffer has its own message filter and 32 transmit buffers with prioritized arbitration scheme. For optimal support of HLP such as DeviceNet, the message filter also covers the first two data bytes of the message payload. A block diagram of the CAN controller is shown in the following figure. Transmit and receive message buffers are SECDED through the error detection and correction (EDAC) controller.

To remove the requirement of APB clock in multiples of 8 MHz, a separate MPU CAN clock is provided and added from the APB bus. The MPU uses toggle synchronizers and there is no restriction on the APB clock relative to the CAN clock.

The CAN clock is derived from PLL output. The CAN clock frequency is based on the PLL clock frequency. The supported frequencies in MHz are 8, 16, 24, 32, 40, 48, 56, 64, 72, and 80.

© 2024 Microchip Technology Inc. and its subsidiaries

Figure 6-8. CAN Controller Block Diagram



6.7.1 Features

CAN controller supports the following features:

Compliance

- Full CAN 2.0B compliant
- Conforms to ISO 11898-1
- Maximum baud rate of 1 Mbps with 8 MHz CAN clock

APB

- APB 3.0 compliant
- APB interface has clock-domain-crossing to CAN logic, allowing APB to operate at any frequency.

Receive Path

- 32 receive (Rx) buffers
- Each buffer has its own message filter
- Message filter covers: ID, IDE, remote transmission request (RTR), data byte 1, and data byte 2
- Message buffers can be linked together to build a bigger message array
- Automatic RTR response handler with optional generation of RTR interrupt

Transmit Path

© 2024 Microchip Technology Inc. and its subsidiaries

- 32 transmit (Tx) message holding registers with programmable priority arbitration
- Message abort command
- Single-shot transmission (SST); no automatic retransmission upon error or arbitration loss

System Bus Interface

- AMBA 3 APB Interface
- Full synchronous zero wait-states interface
- Status and configuration interface

Programmable Interrupt Controller

- Local interrupt controller covering message and CAN error sources

Test and Debugging Support

- Listen Only mode
- Internal Loopback mode
- External Loopback mode
- SRAM Test mode
- Error Capture register
- Provides option to either: show current bit position within CAN message
- Provides option to either: show bit position and type of last captured CAN error

SRAM Based Message Buffers

- Single port, synchronous memory based
- 100% synchronous design

6.7.2 EDAC

An internal 256 x 32 RAM in the CAN controller is protected with EDAC. EDAC configurations and error counters related to the CAN are maintained in MPU system registers.

After power-up, the internal SRAM is not initialized and any READ to the memory location results in an ECC error if EDAC is enabled. To initialize the SRAM, you can put the CAN controller into SRAM Test mode, initialize the SRAM, and enable the EDAC. If SECDED is enabled, it is recommended that the CAN controller must be put into SRAM Test mode and the RAM initialized with user defined known data before operation so that a future read or an uninitialized address does not trigger a SECDED error.

6.7.3 Reset

The CAN controller resets on power-up and is held in Reset until enabled in the SOFT_RESET_CR register. The CAN controller can be reset by writing to CAN0 or CAN1 of the SOFT_RESET_CR register. The SOFT_RESET_CR register is located in the MPU_top_sysreg block.

6.7.4 Functional Descriptions

6.7.4.1 CAN Controller Interface Signals

The external interface signals connecting the PIC64GX device to an off-chip CAN transceiver are listed in the following table.

Table 6-10. CAN BUS Interface

Signal Name	Direction	Description
canclk	Input	CAN Clock.

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued		
Signal Name	**Direction**	**Description**
RX	Input	CAN bus receive signal. This signal connects to the receiver bus of the external transceiver.
TX	Output	CAN bus transmit signal. This signal connects to the external transceiver.
TX_EN_N	Output	External driver enable control signal. This signal is used to enable or disable an external CAN transceiver. TX_EN_N is asserted when the CAN controller is stopped or if the CAN state is bus-off (shut down completely). The CAN transmit enable TX_EN_N signal provided through the I/O MUX to the I/O pads are active-low and the CAN transmit enable provided to the MPU is active-high.

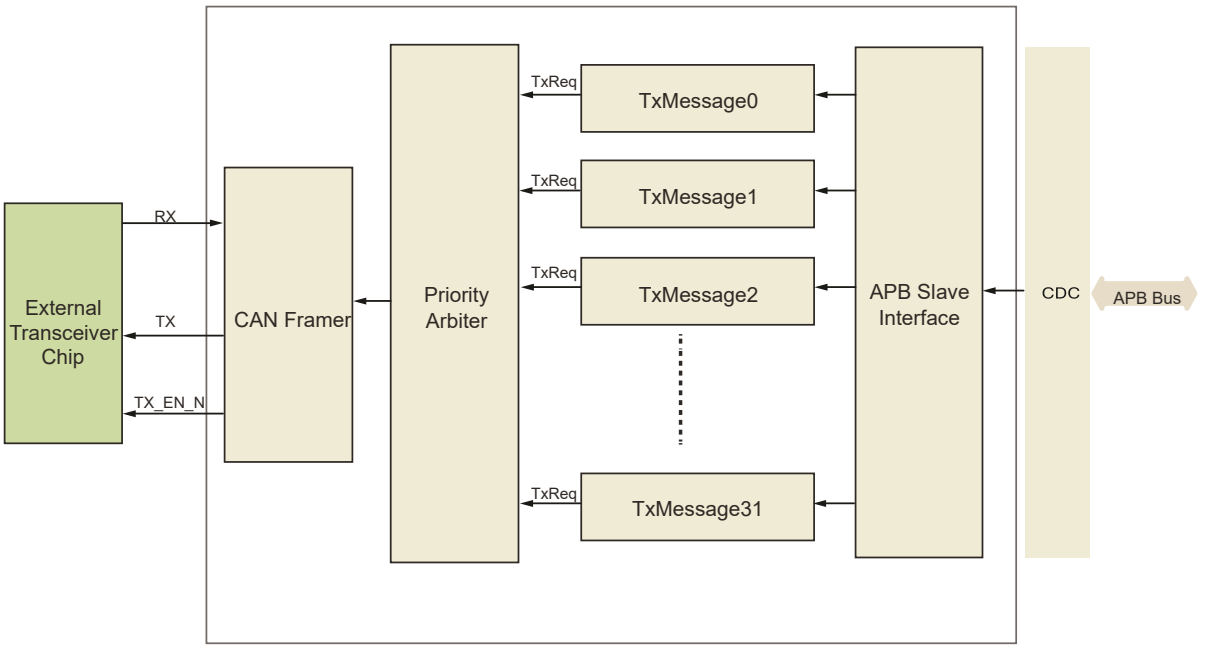
When enabled, CAN ports are configured to connect to multi-standard I/Os by default. CAN signals can also be configured to interface with the MPU general purpose inputs/outputs (GPIOs).

Note: The I/Os allocated to the CAN instance are shared with other MPU peripherals. These I/Os are available as MPU GPIOs when the CAN controller is disabled.

6.7.4.2 Transmit Procedures

The CAN controller provides 32 transmit message holding buffers. An internal priority arbiter selects the message according to the chosen arbitration scheme. Upon transmission of a message or message arbitration loss, the priority arbiter re-evaluates the message priority of the next message. The following figure gives an overall view of the transmit message buffers.

Figure 6-9. Transmit Message Buffers



Two types of message priority arbitration are supported. The type of arbitration is selected using the CAN_CONFIG configuration register. Following are the arbitration types:

- Round Robin: Buffers are served in a defined order: 0-1-2... 31-0-1... A particular buffer is only selected if its TxReq flag is set. This scheme guarantees that all buffers receive the same probability to send a message.
- Fixed Priority: Buffer 0 has the highest priority. This way it is possible to designate buffer 0 as the buffer for error messages and it is guaranteed that they are sent first.

© 2024 Microchip Technology Inc. and its subsidiaries

Note: RTR message requests are served before transmit message buffers are handled. For example, RTRreq0, RTRreq31, TxMessage0, TxMessage1, and TxMessage31.

Procedure for Sending a Message

1. Write message into an empty transmit message holding buffer. An empty buffer is indicated by the TxReq (Bit 0 of TX_MSG#_CTRL_CMD register) that is equal to zero.
2. Request transmission by setting the respective TxReq flag to 1.
3. The TxReq flag remains set as long as the message transmit request is pending. The content of the message buffer must not be changed while the TxReq flag is set.
4. The internal message priority arbiter selects the message according to the chosen arbitration scheme.
5. Once the message is transmitted, the TxReq flag is set to zero and the TX_MSG (Bit 11 of the INT_STATUS register) interrupt status bit is asserted.

Remove a Message from a Transmit Holding Register

A message can be removed from the transmit holding buffer by asserting the TxAbort (Bit 1 if TX_MSG#_CTRL_CMD register) flag. The content of a particular transmit message buffer can be removed by setting TxAbort to 1 to request message removal. This flag remains set as long as the message abort request is pending. It is cleared when either the message wins arbitration (TX_MSG interrupt active) or the message is removed (TX_MSG interrupt inactive).

Single-Shot Transmission

Single-shot transmission (SST) mode is used in systems where the re-transmission of a CAN message due to an arbitration loss or a bus error must be prevented. An SST request is set by asserting TxReq and TxAbort at the same time. Upon a successful message transmission, both flags are cleared.

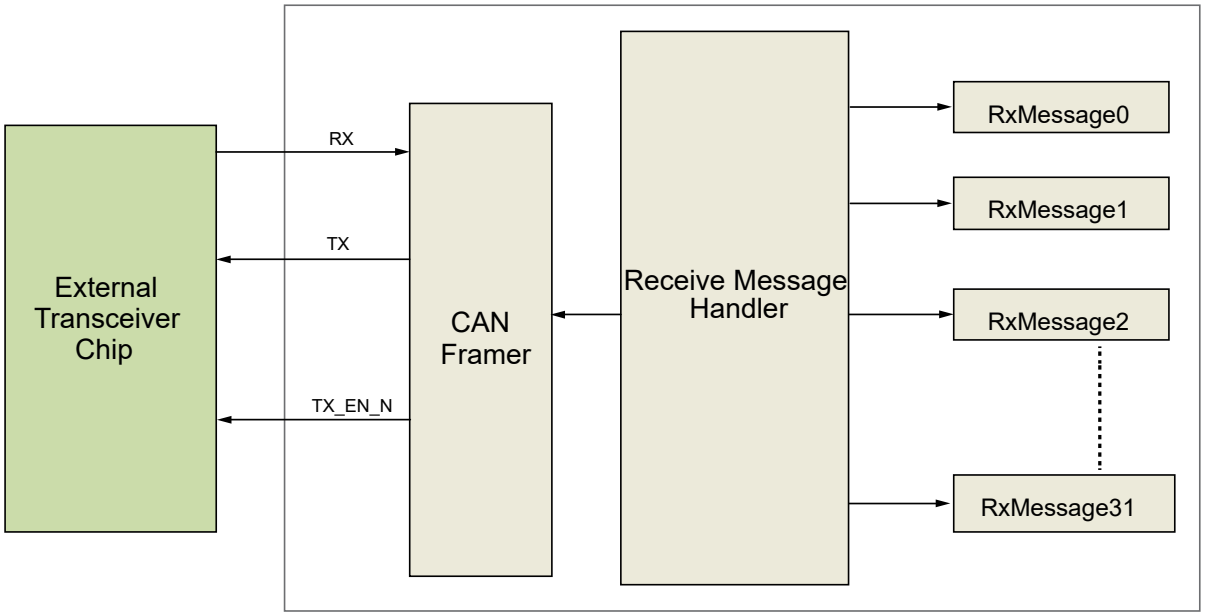
If an arbitration loss or if a bus error happens during the transmission, the TxReq and TxAbort flags are cleared when the message is removed or when the message wins arbitration. At the same time, the SST_FAILURE interrupt is asserted.

6.7.4.3 Receive Procedures

The CAN controller provides 32 individual receive message buffers. Each one has its own message filter mask. Automatic reply to RTR messages is supported. If a message is accepted in a receive buffer, its MsgAv flag is set. The message remains valid as long as the MsgAv flag is set. The master CPU has to reset the MsgAv flag to enable receipt of a new message. The following figure shows the overall block diagram of the receive message buffers.

© 2024 Microchip Technology Inc. and its subsidiaries

Figure 6-10. Receive Message Buffers



Received Message Processing

After a new message is received, the receive message handler searches all receive buffers, starting from the receive message0 until it finds a valid buffer. A valid buffer is indicated by:

- Receive buffer is enabled (indicated by RxBufferEbl = 1)
- Acceptance filter of receive buffer matches incoming message

If the receive message handler finds a valid buffer that is empty, then the message is stored and the MsgAv flag of this buffer is set to 1. If the RxIntEbl flag is set, then the RX_MSG flag of the interrupt controller is asserted.

If the receive buffer already contains a message indicated by MsgAv = 1 and the link flag is not set, then the RX_MSG_LOSS interrupt flag is asserted. Refer to Receive Buffer Linking.

If an incoming message has its RTR flag set and the RTR reply flag of the matching buffer is set, then the message is not stored but an RTR auto-reply request is issued. Refer to RTR Auto-Reply and the RX_MSG0_CTRL_CMD register for more details.

Note: In case of an Extended frame, the received message ID is stored in [31:3] bits of RX ID (RX_MSGn_ID) register. In case of a Standard frame, the message ID is stored in [31:21] bits of RX ID (RX_MSGn_ID) register. Both message identifier (Standard frame and Extended frame) is stored at different bit position of RX ID (RX_MSGn_ID) register.

Acceptance Filter

Each receive buffer has its own acceptance filter that is used to filter incoming messages. An acceptance filter consists of acceptance mask register (AMR) and acceptance code register (ACR) pair. The AMR defines which bits of the incoming CAN message match the corresponding ACR bits.

The following message fields are covered:

- ID
- IDE

© 2024 Microchip Technology Inc. and its subsidiaries

- RTR
- Data byte 1 and data byte 2

Note:
Some CAN HLPs, such as Smart Distributed System (SDS) or DeviceNet, carry additional protocol related information in the first or first and second data bytes that are used for message acceptance and selection. Having the capability to filter these fields provides a more efficient implementation of the protocol stack running on the processor.

The AMR register defines whether the incoming bit is checked against the ACR register. The incoming bit is checked against the respective ACR when the AMR register is 0. The message is not accepted when the incoming bit does not match the respective ACR flag. When the AMR register is 1, the incoming bit is a “don't care”.

RTR Auto-Reply

The CAN controller supports automatic answering of RTR message requests. All 32 receive buffers support this feature. If an RTR message is accepted in a receive buffer where the RTRreply flag is set, then this buffer automatically replies to this message with the content of this receive buffer. The RTRreply pending flag is set when the RTR message request is received. It is cleared when the message is sent or when the message buffer is disabled. To abort a pending RTRreply message, use the `RTRabort` command.

If the RTR auto-reply option is selected, the RTR sent (RTRS) flag is asserted when the RTR auto-reply message is successfully sent. It is cleared by writing “1” to it.

An RTR message interrupt is generated, if the MsgAv_RTRS flag and RxIntEbl are set. This interrupt is cleared by clearing the RTRS flag.

Receive Buffer Linking

Several receive buffers can be linked together to form a receive buffer array which acts almost like a receive FIFO. For a set of receive buffers to be linked together, the following conditions must be met:

- All buffers of the same array must have the same message filter setting (AMR and ACR are identical)
- The last buffer of an array may not have its link flag set

When a receive buffer already contains a message (MsgAv = 1) and a new message arrives for this buffer, this message is discarded (RX_MSG_LOSS Interrupt). To avoid this situation, several receive buffers can be linked together. When the CAN controller receives a new message, the receive message handler searches for a valid receive buffer. If one is found that is already full (MsgAv = 1) and the link flag is set (LF = 1); the search for a valid receive buffer continues. If no other buffer is found, the RX_MSG_LOSS interrupt is set and the message is discarded.

It is possible to build several message arrays. Each of these arrays must use the same AMR and ACR.

Note: The receive buffer locations do not need to be contiguous.

6.7.5 CAN Register Maps and Descriptions

See the configuration and register maps in the ZIP document on the PIC64GX1000 product page www.microchip.com/PIC64GX1000.

6.7.6 CAN Reference Clock

The following table describes CAN. The reference clock should be fixed to achieve the 1 Mbps bus rate.

© 2024 Microchip Technology Inc. and its subsidiaries

Table 6-11. CAN Reference Clock

Parameter	Symbol	Min	Typ	Max	Units
Receive pulse width	$T_{PWCANRX}$	0.8	1.0	1.2	µs
Transmit pulse width	$T_{PWCANTX}$	0.8	1.0	1.2	µs
CAN reference clock frequency	F_{CANCKL}	80	—	80	MHz

6.8 eMMC

The following tables describe the eMMC. The test conditions for eMMC Standard mode use an 8 mA drive strength, fast slew rate, and a 30 pF load (I/O voltage of 3V/1.8V/1.2V). For eMMC High-Speed mode, the test conditions use a 12 mA drive strength, fast slew rate, and 30 pF load (I/O voltage of 3V/1.8V/1.2V). For other eMMC modes, the test conditions use a 12 mA drive strength, fast slew rate, and 15 pF load (I/O voltage of 1.8V/1.2V).

Table 6-12. eMMC Standard Interface

Parameter	Symbol	Min	Typ	Max	Units
eMM clock duty cycle	$T_{DCEMMCHSCLK}$	45	—	55	%
Clock to output delay, all outputs	$T_{EMMCHSCKO}$	–2.0	—	33	ns
Input setup time, all inputs	$T_{EMMCHSDCK}$	2.0	—	—	ns
Input hold time, all inputs	$T_{EMMCHSCKD}$	2.0	—	—	ns
eMMC clock frequency	$F_{EMMCHSCLK}$	25	—	25	MHz

Table 6-13. eMMC High-Speed SDR Interface

Parameter	Symbol	Min	Typ	Max	Units
eMMC high-speed SDR clock duty cycle	$T_{DCEMMCHSCLK}$	45	—	55	%
Clock to output delay, all outputs[1]	$T_{EMMCHSCKO}$	3.2	—	16.8	ns
Input valid data window[2]	$T_{EMMCDIVW}$	0.4	—	—	UI
eMMC high-speed SDR clock frequency	$T_{EMMCHSCLK}$	50	—	50	MHz

Clock to output delay valid after completion of tuning/training.

Input valid data window valid after completion of tuning/training.

Table 6-14. eMMC High-Speed DDR Interface

Parameter	Symbol	Min	Typ	Max	Units
eMMC high-speed DDR clock duty cycle	$T_{DCEMMCDDRCLK}$	45	—	55	%
Data clock to output delay	$T_{EMMCDDRSCKO1}$	2.7	—	7.3	ns
Input valid data window[1]	$T_{EMMCDDRIVW}$	3.5	—	—	ns
Command clock to output delay	$T_{EMMCDDRSCKO2}$	3.2	—	16.0	ns
Command input setup time	$T_{EMMCDDRSCK2}$	3.9	—	—	ns
Command input hold time	$T_{EMMCDDRSCKD2}$	2.5	—	—	ns
eMMC high-speed DDR clock frequency	$T_{EMMCDDRCLK}$	50	—	50	MHz

Input valid data window valid after completion of tuning/training.

© 2024 Microchip Technology Inc. and its subsidiaries

Table 6-15. eMMC HS200 Interface

Parameter	Symbol	Min	Typ	Max	Units
eMMC HS200 clock duty cycle	$T_{DCEMMCHS200CLK}$	40	—	60	%
Clock to output delay, all outputs[1]	$T_{EMMCHS200CKO}$	1.0	—	3.4	ns
Input valid data window[2]	$T_{EMMCSDR1IVW}$	0.4	—	—	UI
eMMC HS200 clock frequency	$T_{EMMCHS200CLK}$	200	—	200	MHz

Clock to output delay valid after completion of tuning/training.

Input valid data window valid after completion of tuning/training.

Table 6-16. eMMC HS400 Interface

Parameter	Symbol	Min	Typ	Max	Units
eMMC HS400 clock duty cycle	$T_{DCEMMCHS400CLK}$	40	—	60	%
Clock to output delay, all outputs[1]	$T_{DCEMMCHS400CKO}$	1.0	—	3.4	ns
Input valid data window[2]	$T_{DCEMMCSDRIVW}$	0.4	—	—	UI
eMMC HS400 clock frequency	$T_{EMMCHS400CLK}$	200	—	200	MHz

Clock to output delay valid after completion of tuning/training.

Input valid data window valid after completion of tuning/training.

Table 6-17. eMMC HS400 Enhanced Strobe Interface

Parameter	Symbol	Min	Typ	Max	Units
eMMCHS400ES clock duty cycle	$T_{DCEMMCHS400ESCLK}$	40	—	60	%
Clock to output delay, command	$T_{DCEMMCHS400ESCKOCMD}$	0.8	—	3.4	ns
Clock to output delay, data	$T_{DCEMMCHS400ESCKODAT}$	1.0	—	2.2	ns
Input valid data window	$T_{DCEMMCSDRIVW}$	0.4	—	—	UI
eMMCHS400ES clock frequency	$T_{EMMCHS400ESCLK}$	200	—	200	MHz

6.9 SD/SDIO

The following tables describe the SD_SDIO peripheral. The test conditions for SD/SDIO Standard mode (default Speed mode) use an 8 mA drive strength, fast slew rate, and a 30 pF load. For SD/SDIO High-Speed mode, the test conditions use a 12 mA drive strength, fast slew rate, and a 30 pF load. For other SD/SDIO High-Speed modes, the test conditions use a 12 mA drive strength, fast slew rate, and a 15 pF load.

Table 6-18. SD/SDIO Interface DDR50 Mode

Parameter	Symbol	Min	Typ	Max	Units
SD device clock duty cycle	$T_{DCDDRCLK}$	45	—	55	%
Clock to output delay, data	$T_{SDDDRCKO1}$	1.0	6.8	—	ns
Input valid data window	$T_{SDDDRIVW}$	3.5	—	—	ns
Input setup time, command	$T_{SDDDRDCK2}$	4.7	—	—	ns
Input hold time, command	$T_{SDDDRCKD2}$	1.5	—	—	ns
Clock to output delay, command[1]	$T_{SDDDRCKO2}$	1.0	13.8	—	ns
High-speed mode SD device clock frequency	$F_{SDDDRCLK}$	50	—	50	MHz

Note: This timing arc is trained at start-up and will be adjusted to match the board characteristics.

© 2024 Microchip Technology Inc. and its subsidiaries

Table 6-19. SD/SDIO Interface SDR104

Parameter	Symbol	Min	Typ	Max	Units
SD device clock duty cycle	$T_{DCSDHSCLK1}$	40	—	60	%
Clock to output delay, all outputs	$T_{SDSDRCKO1}$	1.0	—	3.2	ns
Input valid data window	$T_{SDSDR1IVW}$	0.5	—	—	UI
SDR104 mode device clock frequency	$F_{SDSDRCLK1}$	200	—	200	MHz

Table 6-20. SD/SDIO Interface SDR50/25

Parameter	Symbol	Min	Typ	Max	Units
SD device clock duty cycle	$T_{DCSDHSCLK2}$	40	—	60	%
Clock to output delay, all outputs	$T_{SDSDRCKO2}$	1.0	—	6.8	ns
Input valid data window	$T_{SDSDR2IVW}$	0.3	—	—	UI
SDR50 mode device clock frequency	$F_{SDSDRCLK2}$	—	—	100	MHz
SDR25 mode device clock frequency	$F_{SDSDRCLK2}$	50	—	50	MHz

Table 6-21. SD/SDIO Interface SDR12

Parameter	Symbol	Min	Typ	Max	Units
SD device clock duty cycle	$T_{DCSDHSCLK3}$	40	—	60	%
Clock to output delay, all outputs	$T_{SDSDRCKO3}$	1.0	—	36.8	ns
Input setup time, all inputs	$T_{SDSDRDCK3}$	10.0	—	—	ns
Input hold time, all inputs	$T_{SDSDRCKD3}$	1.5	—	—	ns
SDR12 mode device clock frequency	$F_{SDSDRCLK3}$	25	—	25	MHz

Table 6-22. SD/SDIO Interface High-Speed Mode

Parameter	Symbol	Min	Typ	Max	Units
SD device clock duty cycle	$T_{DCSDHSCLK}$	47	—	53	%
Clock to output delay, all outputs [1]	$T_{SDHSCKO}$	2.2	13.8	—	ns
Input valid data window	$T_{SDHSDIVW}$	0.4	—	—	UI
High-speed mode SD device clock frequency	$F_{SDHSCLK}$	50	—	50	MHz

Note: This timing arc is trained at start-up and will be adjusted to match the board characteristics.

Table 6-23. SD/SDIO Interface Standard Mode

Parameter	Symbol	Min	Typ	Max	Units
SD device clock duty cycle	$T_{DCSDHSCLK}$	45	—	55	%
Clock to output delay, all outputs	T_{SDSCKO}	–2.0	—	4.9	ns
Input setup time, all inputs	T_{SDSDCK}	2.0	—	—	ns
Input hold time, all inputs	T_{SDSCKD}	2.0	—	—	ns
Clock frequency in Identification mode	$F_{SDIDCLK}$	400	—	400	KHz
Standard SD device clock frequency	F_{SDSCLK}	19	—	19	MHz

© 2024 Microchip Technology Inc. and its subsidiaries

6.10 DDR

Table 6-24. DDR Speed Grades

Memory Standard	Package	DRAM Type	Min[9]	Max	Unit
DDR4	All	Single rank component	1333	1333	Mbps
	All	1 rank DIMM[1, 2]	1333	1333	Mbps
	All	2 rank DIMM[1, 3, 7]	1333	1333	Mbps
LPDDR4	All	Single die package	1333	1333	Mbps
	All	Dual die package[6]	1333	1333	Mbps
DDR3L	All	Single rank component	1067	1067	Mbps
	All	1 rank DIMM[1, 2]	1067	1067	Mbps
	All	2 rank DIMM[1, 3, 7]	1067	1067	Mbps

1. Dual In-line Memory Module (DIMM) includes RDIMM, SODIMM, and UDIMM.
2. Includes: 1 rank 1 slot, dual-die package 2 rank.
3. Includes: 2 rank 1 slot.
4. The JEDEC JESD79-4B standard for DDR4 SDRAM limits the maximum tCK to 1.6 ns. Because of this limitation, Microchip recommends working with your DRAM vendor to verify support for data rates at or less than 1066 Mbps.
5. Byte-mode LPDDR4 devices are not supported.
6. Dual die package includes single die with ECC.
7. Refer to board design guidelines for trace matching requirements.
8. The values in this “STD Min” column correspond to the minimum speed tested.

6.11 Gigabit Ethernet MAC (GEM)

The PIC64GX1000 MPU contains two hardened Gigabit Ethernet MAC IP blocks—GEM_0 and GEM_1 — to enable Ethernet solutions over copper or optical cabling.

GEM_0 and GEM_1 are functionally identical, hence, GEM_0 and GEM_1 are referred to as GEM throughout the document.

GEM supports 10 Mb/s, 100 Mb/s, and 1000 Mb/s (1 Gb/s) speeds. GEM provides a complete range of solutions for implementing IEEE 802.3 standard-compliant Ethernet interfaces for chip-to-chip, board-to-board, and backplane interconnects.

6.11.1 Features

GEM supports the following features.

- IEEE 802.3 compliant
- IEEE 802.1Q TSN features:
  - IEEE 802.1AS
  - IEEE 802.1Qav
  - IEEE 802.1Qbv
  - IEEE 802.1CB frame redundancy and elimination
  - IEEE 802.1Qci receive (ingress) traffic policing
  - IEEE 802.3br frame preemption (or interspersing express traffic)
  - IEEE 802.1Qbb priority-based flow control
  - IEEE 802.1Q VLAN tagging with recognition of incoming VLAN and priority tagged frames

© 2024 Microchip Technology Inc. and its subsidiaries

- DMA support
- TCP/IP offloading capability
- Integrated 1000 BASE-X PCS for SGMII-based applications
- Programmable jumbo frames up to 10,240 bytes
- Frame filtering
- Full and half duplex modes at 10/100M and full duplex at 1 Gbps interface speeds for SGMII.
- Wake-on LAN support

6.11.2 Overview

GEM is accessed by the CPU Core Complex through the AXI Switch using the following interfaces:

- AXI interface—used for data transfers.
- APB interface—used for configuration purpose.

GEM can be configured for SGMII.

The PCS sub-block performs the 8b/10b operation for SGMII. The external PHY registers are configured using management interface (MDIO) of the GEM.

6.11.3 Clocking

GEM requires the following clocks:

- tsu_clk: Clock frequency ranges from 5 MHz to 400 MHz for the timestamp unit. Timestamp accuracy improves with higher frequencies. To support single-step timestamping, tsu_clk frequency must be greater than 1/8th the frequency of tx_clk or rx_clk.
- tx_clk: Clock frequency ranges are: 1.25 MHz, 2.5 MHz, 12.5 MHz, 25 MHz, and 125 MHz for MAC transmit clock used by the MAC transmit block. When using Gigabit mode, the transmit clock must be sourced from a 125 MHz reference clock. Depending on the system architecture, this reference clock may be sourced from an on-chip clock multiplier, generated directly from an off-chip oscillator, or taken from the PHY rx_clk. In the SGMII mode, this clock is sourced from the gtx_clk.
- gtx_clk: 125 MHz PCS transmit clock. In SGMII application, this is recovered from input data and needs to be balanced with the tx_clk.
- rx_clk: Clock frequency ranges are: 1.25 MHz, 2.5 MHz, 12.5 MHz, 25 MHz, 62.5 MHz, and 125 MHz. This clock is used by the MAC receive synchronization blocks.
  The following table lists the required frequencies of the transmit clock.

Table 6-25. GEM Transmit Clock Frequencies

MAC Speed Mode (Mbps)	gtx_clk (MHz)	tx_clk (MHz)
	SGMII	SGMII
10	125	1.25
100	125	12.5
1000	125	125

Table 6-26. GEM Receive Clock Frequencies

MAC Speed Mode (Mbps)	pcs_rx_clk (MHz)	rx_clk (MHz)
	SGMII	SGMII
10	125	1.25
100	125	12.5
1000	125	62.5

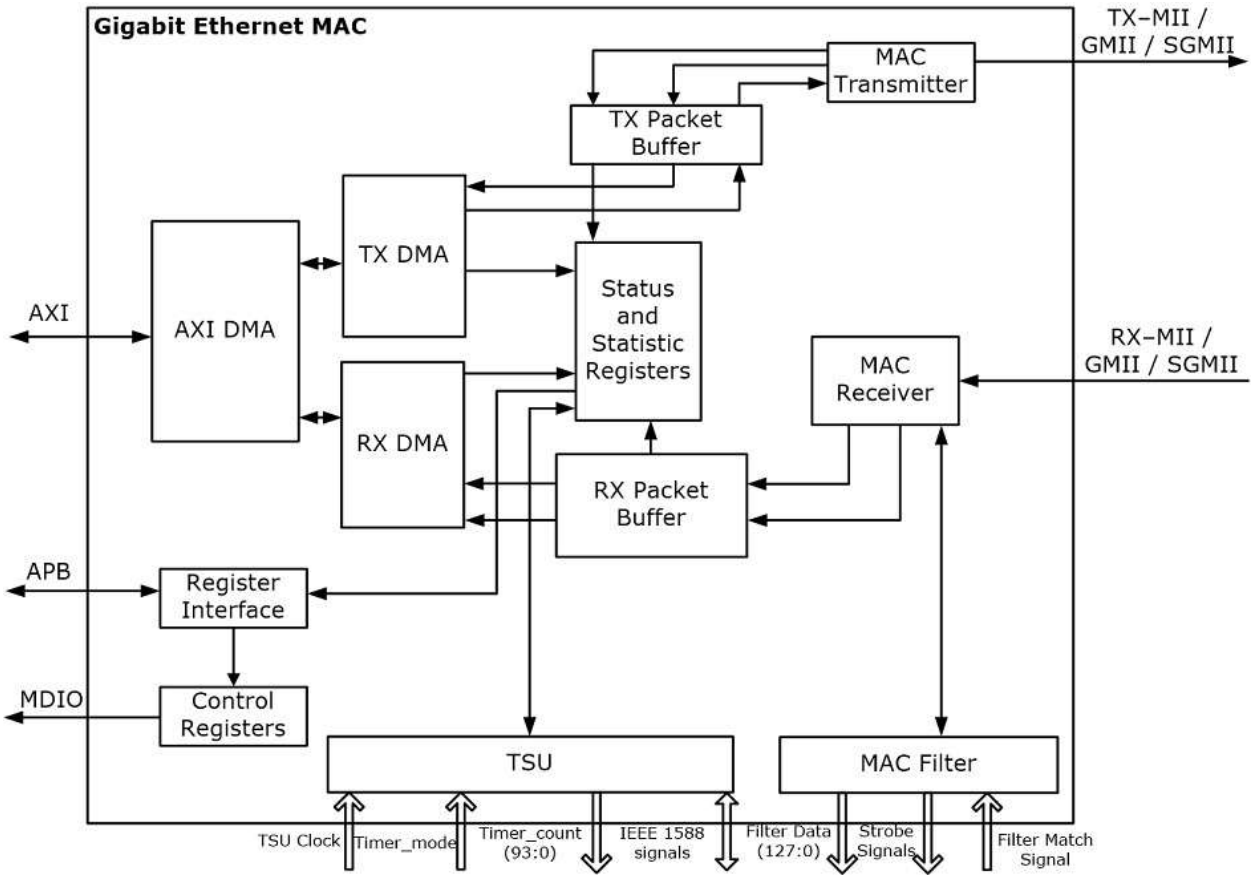
© 2024 Microchip Technology Inc. and its subsidiaries

6.11.4 Functional Description

GEM includes the following functional blocks:

- Integrated 1000BASE-X Physical Coding Sublayer (PCS) for encoding and decoding the data and for Auto Negotiation (AN)
- Time Stamping Unit (TSU) for timer operations
- TSN block to support Timing Sensitive Networking (TSN) features
- High-speed AXI DMA block to transfer data to and from the processor
- Filter block filters out the received frames

Figure 6-11. GEM Block Diagram



Note: PIC64GX supports only SGMII.

6.11.5 MAC Transmitter

The MAC transmitter block retrieves data from the memory using DMA controller, which is connected through the AXI interface. DMA reads the data from memory using the AXI master interface and stores it to the TX packet buffer. Then, the MAC transmitter block retrieves the data from the TX packet buffer and adds preamble and, if necessary, pad and Frame Check Sequence (FCS). The data is transmitted using SGMII.

Both half-duplex and full-duplex Ethernet modes of operation are supported. When operating in Half-Duplex mode, the MAC transmitter block generates data according to the Carrier Sense Multiple Access with Collision Detect (CSMA/CD) protocol. The start of transmission is deferred if Carrier Sense (CS) is active. If collision (col) becomes active during transmission, a jam sequence is

© 2024 Microchip Technology Inc. and its subsidiaries

asserted, and the transmission is re-tried after a random backoff. The CS and col signals have no effect in Full-Duplex mode.

According to the IEEE 802.3 standards, an Ethernet MAC must allow a minimum amount of time before another packet is sent. This pause time between packets is known as Inter-Packet Gap (IPG). The purpose of the IPG is to allow enough time for the receiver to recover the clock and to perform cleanup operations. During this period, IDLE packets will be transmitted. The standard minimum IPG for transmission is 96 bit times. Using GEM, the IPG may be stretched beyond 96 bits depending on the length of the previously transmitted frame. The IPG stretch only works in the Full-Duplex mode.

6.11.5.1 Transmit DMA Buffers

Frames to be transmitted are stored in one or more transmit buffers. The maximum size of a transmit frame is 10240 bytes. The start location for each transmit buffer is stored in AXI memory in a list of transmit buffer descriptors at a location pointed to by the transmit buffer queue pointer. The base address for this queue pointer is set in software using the transmit buffer queue base address register. The upper transmit queue base address register at 0x04c8 is used to set the upper 32 bits of the transmit buffer descriptor queue base address. All the descriptors must be located within a region of memory that does not cross a 4 GB region. The actual 32 bits, chosen for the upper bits, are programmed in the upper receive queue base address register at 0x04c8.

To transmit frames, the buffer descriptors must be initialized by writing an appropriate byte address to bits 31:0 in the first word of each descriptor list entry to indicate the location of the data to be transmitted.

The second word of the transmit buffer descriptor is initialized with control information that indicates the length of the frame, whether or not the MAC is to append CRC and whether the buffer is the last buffer in the frame. It also contains the “used” and “wrap” bits. It is very important that the transmit buffer descriptor list contains at least one entry with its “used” bit set. This is because the transmit DMA can read the buffer descriptor list very fast and will loop round retransmitting data when it encounters the wrap bit. When initializing the descriptor list the user needs to add an additional buffer descriptor with its “used” bit set after the buffer descriptors which describe the data to be transmitted.

The following table lists the transmit buffer descriptor entry.

Table 6-27. Transmit Buffer Descriptor Entry

Bit	Function
Word 0	
31:0	Byte address of the buffer
Word 1	
31	Used – must be zero for GEM to read data to the transmit buffer. GEM sets this to one for the first buffer of a frame once it has been successfully transmitted. Software must clear this bit before the buffer can be used again.
30	Wrap – marks last descriptor in transmit buffer descriptor list. This can be set for any buffer within the frame.
29	Retry limit exceeded, transmit error detected
28	Transmit underrun. Occurs when the start of packet data has been written into the FIFO and either the transmit data could not be fetched in time, or when buffers are exhausted. This is not set when the DMA is configured for packet buffer mode.
27	Transmit frame corruption due to AXI error – set if an error occurs whilst midway through reading transmit frame from the AXI, and RRESP/BRESP errors and buffers exhausted mid frame (if the buffers run out during transmission of a frame then transmission stops, Frame Check Sequence (FCS) shall be bad and tx_er asserted).

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued	
Bit	**Function**
26	Late collision, transmit error detected. Late collisions only force this status bit to be set in gigabit mode.
25:24	Reserved
23	Reserved
22:20	Transmit IP/TCP/UDP checksum generation offload errors: • 000 - No Error • 001 - The Packet was identified as a VLAN type, but the header was not fully complete, or had an error in it • 010 - The Packet was identified as a SNAP type, but the header was not fully complete, or had an error in it • 011 - The Packet was not of an IP type, or the IP packet was invalidly short, or the IP was not of type IPv4/IPv6 • 100 - The Packet was not identified as VLAN, SNAP, or IP • 101 - Non supported packet fragmentation occurred. For IPv4 packets, the IP checksum was generated and inserted • 110 - Packet type detected was not TCP or UDP. TCP/UDP checksum was therefore not generated. For IPv4 packets, the IP checksum was generated and inserted • 111- A premature end of packet was detected and the TCP/UDP checksum could not be generated
19:17	Reserved. Must be set to 3'b000 to disable TSO and UFO
16	No CRC to be appended by MAC. When set, this implies that the data in the buffers already contains a valid CRC and hence no CRC or padding is to be appended to the current frame by the MAC. This control bit must be set for the first buffer in a frame and is ignored for the subsequent buffers of a frame. This bit must be clear when using the transmit IP/TCP/UDP checksum generation offload, otherwise checksum generation and substitution will not occur. **Note:** This bit must also be cleared when TX Partial Store and Forward mode is active.
15	Last buffer. When set, this bit indicates the last buffer in the current frame is reached.
14	Reserved
13:0	Length of the buffer

6.11.6 MAC Receiver

MAC receiver block receives data using SGMII interface and stores the data in the RX packet buffer. Using RX DMA controller, data from the RX packet buffer is read and transferred to the memory using AXI interface.

The MAC receive block checks for valid preamble, FCS, alignment, and length, and presents received frames to the MAC address checking block. Firmware can configure GEM to receive jumbo frames up to 10,240 bytes.

The address checker identifies the following:

- Four source or destination specific 48-bit addresses
- Four different types of ID values
- A 64-bit hash register for matching multi-cast and unicast addresses as required.
- Broadcast address of all ones, copy all frames and act on external address matching signals.

© 2024 Microchip Technology Inc. and its subsidiaries

- Supports offloading of IP, TCP, and UDP checksum calculations (both IPv4 and IPv6 packet types are supported) and can automatically discard frames with a bad checksum. As the MAC supports TSN features, it identifies 802.1CB streams and automatically eliminates duplicate frames. Statistics are provided to report counts of rogue and out-of-order frames, latent errors, and the timer reset events.
- Broadcast address of all ones, copy all frames and act on external address matching signals.

During frame reception, if the frame is too long, a bad frame indication is sent to the DMA controller and the receiver logic does not store that frame in the internal DMA buffer. At the end of frame reception, the receive block indicates to the DMA block whether the frame is good or bad. The DMA block recovers the current receive buffer if the frame is bad.

6.11.6.1 Receive DMA Buffers

Received frames, optionally including FCS, are written to receive buffers in the AXI memory. The receive buffer depth is 16384 bytes.

The start location of each receive buffer is stored as a list of receive buffer descriptors. The receive buffer queue pointer stores the address of each buffer descriptor. The base address for the receive buffer queue pointer is configured in software using the receive buffer queue base address register at 0x04d4 location. This register is used to set the upper 32 bits of the base address of the descriptor. With 64-bit addressing, there is a restriction that all the descriptors must be located within a region of memory that does not cross a 4 GB region, in other words the upper 32 bits of the 64-bit address must be fixed. This is only true of the descriptors and not the packet data which can be anywhere in the 64-bit address space.

Each receive buffer start location is a word address. The start of the first buffer in a frame can be offset by up to three bytes depending on the value written to bits 14 and 15 of the network configuration register.

The following table lists the receive buffer descriptor entry:

Table 6-28. Receive Buffer Descriptor Entry

Bit	Function
Word 0	
31:2	Address [31:2] of beginning of buffer
1	Wrap - marks last descriptor in receive buffer descriptor list.
0	Ownership - needs to be zero for GEM to write data to the receive buffer. GEM sets this to 1 once it has successfully written a frame to memory. Software has to clear this bit before the buffer can be used again.
Word 1	
31	Global all ones broadcast address detected
30	Multicast hash match
29	Unicast hash match
28	External address match. **Note:** If the packet buffer mode and the number of configured specific address filters is greater than four in gem_gxl_defs.v then external address matching is not reported in this bit and instead it is set if there has been a match in the first eight specific address registers. Bit 27 is then used along with bits 26:25 to indicate which register matched.
27	Indicates a specific address register match found, bit 25 and bit 26 indicates which specific address register causes the match. See description of preceding bit 28.

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

Bit	Function
26:25	Specific address register match. Encoded as follows: • 00 - Specific address register 1 match • 01 - Specific address register 2 match • 10 - Specific address register 3 match • 11 - Specific address register 4 match If more than one specific address is matched only one is indicated with priority 4 down to 1.
24	This bit has a different meaning depending on whether RX checksum offloading is enabled. • With RX checksum offloading disabled: (bit 24 clear in Network Configuration) Type ID register match found, bit 22 and bit 23 indicate which type ID register causes the match. • With RX checksum offloading enabled: (bit 24 set in Network Configuration) – 0 - The frame was not SNAP encoded and/or had a VLAN tag with the CFI bit set. – 1 - The frame was SNAP encoded and had either no VLAN tag or a VLAN tag with the CFI bit not set.
23:22	This bit has a different meaning depending on whether RX checksum offloading is enabled. • With RX checksum offloading disabled: (bit 24 clear in Network Configuration) Type ID register match. Encoded as follows: – 00 - Type ID register 1 match – 01 - Type ID register 2 match – 10 - Type ID register 3 match – 11 - Type ID register 4 match If more than one Type ID is matched only one is indicated with priority 4 down to 1. • With RX checksum offloading enabled: (bit 24 set in Network Configuration) – 00 - Neither the IP header checksum nor the TCP/UDP checksum was checked. – 01 - The IP header checksum was checked and was correct. Neither the TCP nor UDP checksum was checked. – 10 - Both the IP header and TCP checksum were checked and were correct. – 11 - Both the IP header and UDP checksum were checked and were correct.
21	VLAN tag detected — type ID of 0x8100. For packets incorporating the stacked VLAN processing feature, this bit will be set if the second VLAN tag received has a type ID of 0x8100.
20	Priority tag detected — type ID of 0x8100 and null VLAN identifier. For packets incorporating the stacked VLAN processing feature, this bit will be set if the second VLAN tag received has a type ID of 0x8100 and a null VLAN identifier.
19:17	When bit 15 (End of frame) and bit 21 (VLAN tag) are set, these bits represent the VLAN priority. When header/data splitting is enabled (via bit 5 of the DMA configuration register, offset 0x10) bit 17 indicates this descriptor is pointing to the last buffer of the header.

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued	
Bit	**Function**
16	This bit has a different meaning depending on the state of bit 13 (report bad FCS in bit 16 of word 1 of the receive buffer descriptor) and bit 5 (header/data splitting) of the DMA Configuration register (offset 0x10). When header/data splitting is enabled and this buffer descriptor (BD) is not the last BD of the frame (as indicated in bit 15 of this BD), this bit will indicate that the BD is pointing to a data buffer containing header bytes. When this BD is the last BD of the frame (as indicated in bit 15 of this BD), and bit 13 of the DMA configuration register is set, this bit represents FCS/CRC error. When this BD is the last BD of the frame (as indicated in bit 15 of this BD), and bit 13 of the DMA configuration register is clear, and the received frame is VLAN tagged, this bit represents the Canonical format indicator (CFI).
15	End of frame - when set, the buffer contains the end of a frame. If end of frame is not set, then the only valid status bit (unless header/data splitting is enabled) is start of frame (bit 14). If header/data splitting is enabled, then bits 16 and 17 are also valid status bits when this bit is not set.
14	Start of frame - when set, the buffer contains the start of a frame. If both bits 15 and 14 are set, the buffer contains a whole frame.
13	This bit has a different meaning depending on whether jumbo frames and ignore FCS mode are enabled. If no mode is enabled, this bit will be zero. • With jumbo frame mode enabled: (bit 3 set in Network Configuration Register) Additional bit for length of frame (bit[13]), that is concatenated with bits[12:0] • With ignore FCS mode enabled and jumbo frames disabled: (bit 26 set in Network Configuration Register and bit 3 clear in Network Configuration Register) This indicates per frame FCS status as follows: – 0 – Frame had good FCS – 1 – Frame had bad FCS, but was copied to memory as ignore FCS enabled
12:0	When header/data splitting enabled (via bit 5 of the DMA configuration register, offset 0x10) and bit 17 is set (last buffer of header), these bits represent the length of the header in bytes. When bit 15 (End of frame) is set, these bits represent the length of the received frame which may or may not include FCS depending on whether FCS discard mode is enabled. • With FCS discard mode disabled: (bit 17 clear in Network Configuration Register) Least significant 12-bits for length of frame including FCS. If jumbo frames are enabled, these 12-bits are concatenated with bit[13] of the descriptor above. • With FCS discard mode enabled: (bit 17 set in Network Configuration Register) Least significant 12-bits for length of frame excluding FCS. If jumbo frames are enabled, these 12-bits are concatenated with bit[13] of the descriptor above.

6.11.7 Register Interface

Control registers drive the MDIO interface, set up DMA activity, start frame transmission, and select modes of operation such as Full-Duplex, Half-Duplex, and 10/100/1000 Mbps operation. The register interface is through the APB interface, which connects to the core complex subsystem.

The Statistics register block contains registers for counting various types of an event associated with transmit and receive operations. These registers, along with the status words stored in the receive buffer list, enable the software to generate Network Management Statistics registers.

© 2024 Microchip Technology Inc. and its subsidiaries

6.11.8 AXI DMA

The built-in DMA controller is attached to the MAC buffer memories to provide a scatter-gather type capability for packet data storage.

DMA uses the AXI interface for data transfer and uses the APB interface for configuration and monitoring DMA descriptors. DMA uses separate transmit and receive buffers as memories to store the frames to be transmitted or received. It uses separate transmit and receive lists of buffer descriptors, with each descriptor describing a buffer area in the memory. This allows the Ethernet packets to be broken and scattered around the system memory.

TX DMA is responsible for the transmit operations and RX DMA is responsible for the receive operations. TX DMA reads the data from memory, which is connected through the AXI interface and stores data to the transmit packet buffers. RX DMA fetches the data from the receive packet buffers and transfers it to the application memory.

Receive buffer depth is programmable within the range of 64 bytes to 16,320 bytes. The start location for each receive buffer is stored in the memory in a list of receive buffer descriptors, at an address location pointed by the receive buffer queue pointer. The base address for the receive buffer queue pointer is configured using the DMA registers.

Transmit frames can be in the range of 14 bytes to 10,240 bytes long. As a result, it is possible to transmit jumbo frames. The start location for each transmit buffer is stored in a list of transmit buffer descriptors at a location pointed by the transmit buffer queue pointer. The base address for this queue pointer is configured using the DMA registers.

Following are the features of DMA Controller:

- 64-bit data bus width support
- 64-bit address bus width support
- Support up to 16 outstanding AXI transactions. These transactions can cross multiple frame transfers.
- Ability to store multiple frames in the packet buffer resulting in the maximum line rate
- Supports priority queuing
- Supports TCP/IP advanced offloads to reduce CPU overhead
  AXI read operations are routed to the AXI read channel and all write operations to the write channel. Both read and write channels may operate simultaneously. Arbitration logic is implemented when multiple requests are active on the same channel. For example, when the transmit and receive DMA request for data for transmission and reception of data at the same time, the receive DMA is granted the bus before the transmit DMA. However, most requests are either receive data writes or transmit data reads both of which can operate in parallel and can execute simultaneously.

6.11.9 MAC Filter

The filter block determines which frames are written to the DMA interface. Filtering is performed on received frames based on the state of the external matching pins, the contents of the specific address, type and hash registers, and the frame's destination address, and the field type.

GEM is configured to have four specific address filters. Each filter is configured to contain a MAC address, which is specified to be compared against the Source Address (SA) or Destination Address (DA) of each received frame. There is also a mask field to allow certain bytes of the address that are not to be included in the comparison. If the filtering matches for a specific frame, then it is passed on to the DMA memory. Otherwise, the frame is dropped.

Frames may also be filtered using the Type ID field for matching. There are four types of ID registers in the internal register space, and these may be enabled individually. GEM supports the recognition of specific source or destination addresses. The number of a specific source or destination address filters are configurable and can range from 0 (zero) to 4.

© 2024 Microchip Technology Inc. and its subsidiaries

6.11.10 Time Stamping Unit

TSU implements a timer, which counts the time in seconds and nanoseconds format. This block is supplied with tsu_clk, which ranges from 5 MHz to 400 MHz. The timer is implemented as a 94-bit register as follows.

- The upper 48 bits counts seconds
- The next 30 bits counts nanoseconds
- The lower 16 bits counts sub nanoseconds

Note: sub nanoseconds is a time-interval measurement unit which is shorter than nanoseconds.

The timer increments at each tsu_clk period and an interrupt is generated in the seconds increment. The timer value can be read, written, and adjusted through the APB interface.

There are two modes of operation:

- Timer Adjust Mode
- Increment Mode

6.11.10.1 Timer Adjust Mode

In Timer Adjust mode, the maximum clock frequency is 125 MHz. There are several signals, synchronous to tsu_clk output by the MAC.

In this mode, the timer operation is also controlled from the MPU by input signals called gem_tsu_inc_ctrl [1:0] along with gem_tsu_ms.

When the gem_tsu_inc_ctrl [1:0] is set to:

- 2b'11 – Timer register increments as normal
- 2b'01 – Timer register increments by an additional nanosecond
- 2b'10 – Timer increments by a nanosecond less
- 2b'00:
  - When the gem_tsu_ms is set to: logic 1, the nanoseconds timer register is cleared and the seconds timer register is incremented with each clock cycle.
  - When the gem_tsu_ms is set to: logic 0, the timer register increments as normal, but the timer value is copied to the Sync Strobe register.

The TSU timer count value can be compared to a programmable comparison value. For the comparison, the 48 bits of the seconds value and the upper 22 bits of the nanoseconds value are used. The timer_cmp_val signal is output from the core to indicate when the TSU timer value is equal to the comparison value stored in the timer comparison value registers.

6.11.10.2 Increment Mode

In the Increment mode, the tsu_clk is supplied either from an external reference clock or from the MPU. The maximum clock frequency is 400 MHz. In this mode, the timer signals interfacing the MPU are gated off.

6.11.11 IEEE 1588 Implementation

IEEE 1588 is a standard for precision time synchronization in local area networks. It works with the exchange of special PTP frames. The PTP messages can be transported over IEEE 802.3/Ethernet, over Internet Protocol Version 4 (IPv4) or over Internet Protocol Version 6 (IPv6). GEM detects when the PTP event messages: sync, delay_req, pdelay_req, and pdelay_resp are transmitted and received.

GEM asserts various strobe signals for different PTP event messages.

GEM supports the following functionalities:

- Identifying PTP frames

© 2024 Microchip Technology Inc. and its subsidiaries

- Extracting timestamp information out of received PTP frames
- Inserting timestamp information into received data frames, before passing to buffer memory
- Inserting timestamp information into transmitted data frames
- Allowing control of TSU through MPU

GEM samples the TSU timer value when the TX or RX SOF event of the frame passes the MII/GMII boundary. This event is an existing signal synchronous to MAC TX/RX clock domains. The MAC uses the sampled timestamp to insert the timestamp into transmitted PTP sync frames (if one step sync feature is enabled) or to pass to the register block to capture the timestamp in APB accessible registers, or to pass to the DMA to insert into TX or RX descriptors. For each of these, the SOF event, which is captured in the tx_clk and rx_clk domains respectively, is synchronized to the tsu_clk domain and the resulting signal is used to sample the TSU count value.

There is a difference between IEEE 802.1AS and IEEE 1588. The difference is, IEEE 802.1AS uses the Ethernet multi-cast address 0180C200000E for sync frame recognition whereas IEEE 1588 does not. GEM is designed to recognize sync frames with both 802.1AS and 1588 addresses and so can support both 1588 and 802.1AS frame recognition simultaneously.

6.11.11.1 PTP Probes

There are a number of strobe signals from the GEM to the MPU. These signals indicate the transmission/reception of various PTP frames. The following table lists these signals.

Table 6-29. PTP Strobe Signals

Signal Name	Description
DELAY_REQ_RX	Asserted when the PTP RX delay request is detected.
DELAY_REQ_TX	Asserted when the PTP TX delay request is detected.
PDELAY_REQ_RX	Asserted when the PTP PDELAY RX request is detected.
PDELAY_REQ_TX	Asserted when the PTP PDELAY TX request is detected.
PDELAY_RESP_RX	Asserted when the PTP PDELAY RX response request is detected.
PDELAY_RESP_TX	Asserted when the PTP PDELAY TX response request is detected.
SOF_RX	Asserted on SFD, de-asserted at EOF.
SOF_TX	Asserted on SFD, de-asserted at EOF.
SYNC_FRAME_RX	Asserted when the SYNC_FRAME RX response request is detected.
SYNC_FRAME_TX	Asserted when the SYNC_FRAME TX response request is detected.

6.11.11.2 PTP Probe Usage (GMII)

When GEM is configured in the GMII/MII mode, transmit PTP strobes are synchronous to mac_tx_clk and receive PTP strobes are synchronous to mac_rx_clk. GEM sources these clocks from the MPU.

6.11.11.3 PTP Probe Usage (SGMII)

When GEM is configured in the SGMII mode, the PTP strobes must be considered asynchronous because the Tx and Rx clocks are not available in the MPU. Hence, the strobe signals must be synchronized with a local clock in the MPU before being used.

6.11.12 Time Sensitive Networking

GEM includes the following key TSN functionalities among others:

- IEEE 802.1 Qav Support – Credit based Shaping
- IEEE 802.1 Qbv – Enhancement for Scheduled Traffic
- IEEE 802.1 CB Support
- IEEE 802.1 Qci Receive Traffic Policing

© 2024 Microchip Technology Inc. and its subsidiaries

- IEEE 802.3br Support
  - GEM supports 10 Mb/s, 100 Mb/s, and 1000 Mb/s (1 Gb/s) speeds. GEM provides a complete range of solutions for implementing IEEE 802.3 standard-compliant Ethernet interfaces for chip-to-chip, board-to-board, and backplane interconnects.

A credit-based shaping algorithm is available on the two highest priority active queues and is defined in IEEE 802.1Qav Forwarding and Queuing Enhancements for Time-Sensitive Streams. Traffic shaping is enabled through the register configuration. Queuing can be handled using any of the following methods.

- Fixed priority
- Deficit Weighted Round Robin (DWRR)
- Enhanced transmission selection

Selection of the queuing method is done through register configuration. For information about the internal registers of the GEM, see the configuration and register maps in the ZIP document on the PIC64GX1000 product page www.microchip.com/PIC64GX1000.

6.11.12.1 IEEE 802.1 Qbv – Enhancement for Scheduled Traffic

IEEE 802.1 Qbv is a TSN standard for enhancement for scheduled traffic and specifies time aware queue-draining procedures based on the timing derived from IEEE 802.1 AS. It adds transmission gates to the eight priority queues, which allow low priority queues to be shut down at specific times to allow higher priority queues immediate access to the network at specific times.

GEM supports IEEE 802.1Qbv by allowing time-aware control of individual transmit queues. GEM has the ability to enable and disable transmission on a particular queue on a periodic basis with the ON or OFF cycling, starting at a specified TSU clock time.

6.11.12.2 IEEE 802.1 CB Support

IEEE 802.1CB “Frame Replication and Elimination for Reliability” is one of the Time Sensitive Networking (TSN) standards. Using Frame Replication and Elimination for Reliability (FRER) within a network increases the probability that a given packet is delivered using multi-path paths through the network.

The MAC supports a subset of this standard and provides the capability for stream identification and frame elimination but does not provide support for the replication of frames.

6.11.12.3 IEEE 802.1 Qci Receive Traffic Policing

IEEE 802.11 Qci is a policy mechanism that discards frames in receive (ingress) if they exceed their allocated frame length or flow rate. TSN standards enable provisioning the resources in a network in such a way that high priority traffic is ensured to get through as long as it does not exceed its frame length and flow rate allocation.

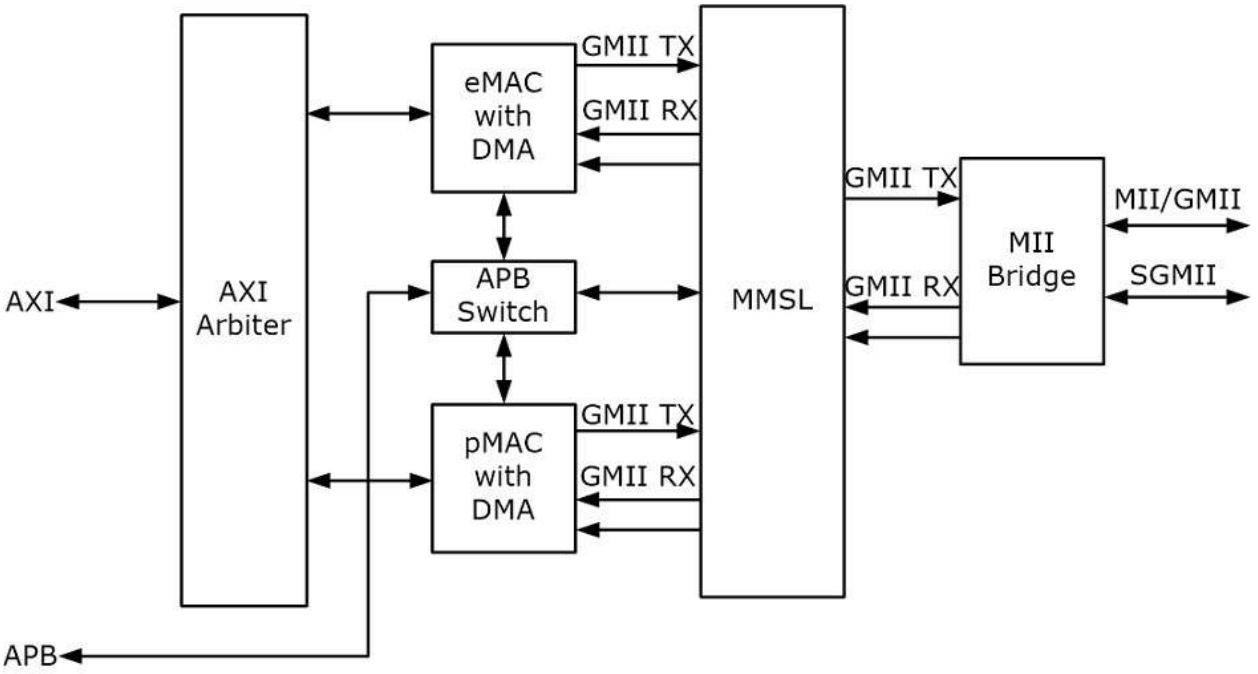
6.11.12.4 IEEE 802.3br Support

All default operations of MAC are done by using PMAC. One more MAC, which is identical to PMAC is used, termed as EMAC, which is used when IEEE 802.3br is configured. IEEE 802.3br Interspersing Express Traffic is one of the TSN standards, which defines a mechanism to allow an express frame to be transmitted with minimum delay at the expense of delaying completion of normal priority frames.

This standard has been implemented by instantiating two separate MAC modules with related DMA, a MAC Merge Sub Layer (MMSL) and an AXI arbiter. One MAC is termed the express or eMAC and the other is a pre-emptable or pMAC. The eMAC is designed to carry time sensitive traffic, which must be delivered within a known time.

© 2024 Microchip Technology Inc. and its subsidiaries

Figure 6-12. IEEE 802.3br Support



Note: PIC64GX supports only SGMII.

6.11.13 PHY Interface

GEM can be configured to support the SGMII PHY. When using SGMII, the PCS block of that GEM is used.

6.11.13.1 Physical Coding Sublayer

A PCS is incorporated for 1000BASE-X operation which includes 8b/10b encoder, decoder, and the Auto Negotiation module. This interface is connected to I/O BANK 5.

6.11.13.2 SGMII

GEM includes the SGMII functional block, which provides the SGMII interface between GEM and Ethernet PHY. The SGMII block provides the following functionalities:

- Clock Domain Recovery (CDR) of received 125 MHz clock
- Serializing or De-serializing
- PLL for synthesis of a 125 MHz transmit clock
  The SGMII block routes the data to the PHY through the dedicated I/O BANK 5.

6.11.13.3 PHY Management Interface

GEM includes an MDIO interface, which can be routed through the MPUIO or the PIC64GX I/Os. The MDIO interface is provided to allow GEM to access the PHY's management registers. This interface is controlled by the PHY management register. Writing to this register causes a PHY management frame to be sent to the PHY over the MDIO interface. PHY management frames are used to either write or read from PHY's control and STATUS registers.

If desired, however, the user can just bring out one management interface (and not use the second) as it is possible to control multiple PHYs through one interface. Management Data Clock (MDC) should not toggle faster than 2.5 MHz (minimum period of 400 ns), as defined by the IEEE 802.3 standard. MDC is generated by dividing processor clock (pclk). A register configuration determines by how much pclk should be divided to produce MDC.

© 2024 Microchip Technology Inc. and its subsidiaries

6.11.14 Address Register Map

GEM is configured using the following internal registers.

Table 6-30. Register Address Map

Address Offset (Hex)	Register Type	Width
MAC Registers or Pre-emptable MAC Registers		
0x0000	Control and STATUS	32
0x0100	Statistics	32
0x01BC	Time Stamp Unit	32
0x0200	Physical Coding Sublayer	32
0x0260	Miscellaneous	32
0x0300	Extended Filter	32
0x0400	Priority Queue and Screening	32
0x0800	Time Sensitive Networking	32
0x0F00	MAC Merge Sublayer	32
eMAC Registers		
0x1000 to 0x1FFF	eMAC	32

6.11.15 Protocol Characteristics

Table 6-31. Serial-GMII Protocol Characteristics (TJ –40 °C–100 °C)

Parameter	Min	Max	Units
Line rate	1250	1250	Mbps
SGMII deterministic transmitter jitter	—	0.175	UI
SGMII total transmitter jitter	—	0.25	UI
SGMII total receiver jitter tolerance	0.25	—	UI
TX to RX frequency offset	—	±300	ppm

Table 6-32. Serial-GMII Protocol Characteristics (TJ –55 °C–125 °C)

Parameter	Min	Max	Units
Line rate	1250	1250	Mbps
SGMII deterministic transmitter jitter	—	0.175	UI
SGMII total jitter	—	0.25	—
SGMII total receiver jitter tolerance	0.25	—	UI
TX to RX frequency offset	—	±200	ppm

6.12 Watchdog Timer

The watchdog timer is an advanced peripheral bus (APB) slave that guards against the system crashes requiring regular service by the processor. The PIC64GX1000 MPU contains five identical watchdog timers in the microprocessor subsystem (watchdog_0, watchdog_1, watchdog_2, watchdog_3, and watchdog_4). Watchdog_0 is associated with the E51 core and is the only one out of the five watchdogs capable of resetting the core complex when it triggers. Each of the other four

© 2024 Microchip Technology Inc. and its subsidiaries

watchdog timers is maintained by a dedicated U54 core and is only capable of interrupting the E51 upon triggering.

6.12.1 Features

The watchdog timer supports the following features:

- A 32-bit timer counts down from a preset value to zero, then performs one of the following user-configurable operations: If the counter is not refreshed, it times out and either causes a system reset or generates an interrupt to the processor.
- The watchdog timer counter is halted when the processor enters the Debug state
- The watchdog timer can be configured to generate a wake-up interrupt when the processor is in Sleep mode

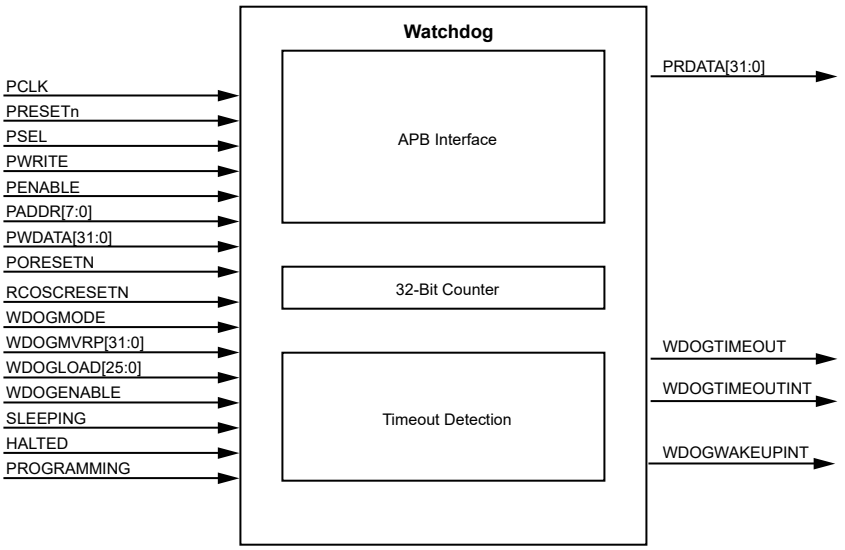
The watchdog timer is connected to the MPU AMBA interconnect through the APB interface.

6.12.2 Functional Description

The watchdog timer consists of the following components (as shown in the following figure):

- APB Interface
- 32-Bit Counter
- Timeout Detection

Figure 6-13. Watchdog Timer Block Diagram



6.12.3 APB Interface

The watchdog timer has an APB interface through which the processor can access various CONTROL and STATUS registers to control and monitor its operation. The APB interface is clocked by the PCLK clock signal.

6.12.4 32-Bit Counter

The operation of the watchdog timer is based on a 32-bit down counter that must be refreshed at regular intervals by the processor. If not refreshed, the counter will time-out. In normal operation, the generation of a Reset or time-out interrupt by the watchdog timer does not occur because the watchdog timer counter is refreshed on a regular basis.

© 2024 Microchip Technology Inc. and its subsidiaries

The MPU watchdog timers are not enabled initially when the MPU comes out of Reset. When the device is powered up, the watchdog timer is enabled with the timeout period set to approximately 10.47 seconds (if VDD = 1.2 V).

6.12.5 Timeout Detection

A control bit in the WDOG_CONTROL register is used to determine whether the watchdog timer generates a Reset or an interrupt if a counter time-out occurs. The default setting is Reset generation on time-out. When interrupt generation is selected, the WDOGTIMEOUTINT output is asserted on time-out and remains asserted until the interrupt is cleared. When Reset generation is selected, the watchdog timer does not directly generate the system Reset signal. Instead, when the counter reaches zero, the watchdog timer generates a pulse on the WDOGTIMEOUT output, and this is routed to the Reset controller to cause it to assert the necessary Reset signals.

Note: Only watchdog_0 can reset the MPU. The other watchdogs can only generate interrupts to the E51 core.

6.12.6 Register Map and Descriptions

See the configuration and register maps in the ZIP document on the PIC64GX1000 product page www.microchip.com/PIC64GX1000.

6.12.7 Clock Input

Table 6-33. Watchdog Timer

Parameter	Symbol	Min	Max	Units
Watchdog timer input clock frequency	F_{WDTCLK}	—	156.25	MHz

6.13 Real-time Counter

The MPU real-time counter (RTC) keeps track of seconds, minutes, hours, days, weeks, and years.

6.13.1 Features

It has two modes of operation:

- Real-time Calendar: Counts seconds, minutes, hours, days, week, months, and years
- Binary Counter: Consecutively counts from 0 to 2^{43} - 1

The RTC is connected to the main MPU AMBA interconnect via an APB interface.

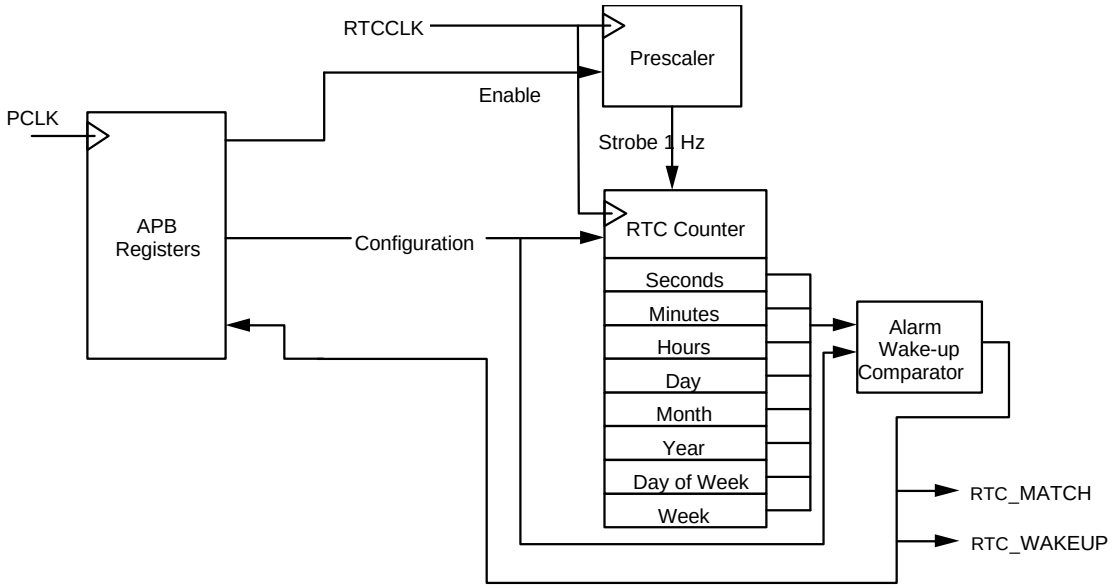
6.13.2 Functional Description

The RTC architecture and its components are as follows:

- Prescaler
- RTC Counter
- Alarm Wake-up Comparator

© 2024 Microchip Technology Inc. and its subsidiaries

Figure 6-14. RTC Block Diagram



6.13.3 Prescaler

The prescaler divides the input frequency to create a time-based strobe (typically 1 Hz) for the calendar counter. The Alarm and Compare Registers, in conjunction with the calendar counter, facilitate time-matched events.

To properly operate in Calendar mode, (Clock mode: 1), the 26-bit prescaler must be programmed to generate a 1 Hz strobe to the RTC. In Binary mode, (Clock mode: 0), the prescaler can be programmed as required in the application.

6.13.4 RTC Counter

The RTC counter keeps track of seconds, minutes, hours, days, weeks, and years when in Calendar mode, and for this purpose it requires a 43-bit counter. When counting in Binary mode, the 43-bit register is treated as a linear up counter.

The following table lists the detail for Calendar mode and Binary mode.

Table 6-34. Calendar Counter Description

Function	Number of Bits	Range		Reset Value	
		Calendar Mode	Binary Mode	Calendar Mode	Binary Mode
Second	6	0-59	0-63	0	0
Minute	6	0-59	0-63	0	0
Hour	5	0-23	0-31	0	0
Day	5	1-31 (auto adjust by month and year)	0-31	1	0
Month	4	1-12	0-15	1	0
Year	8	0-255 Year 2000 to 2255	0-255	0 (year 2000)	0
Weekday	3	1-7	0-7	7	0
Week	6	1-52	0-63	1	0

© 2024 Microchip Technology Inc. and its subsidiaries

Note: The long-term accuracy of the RTC depends on the accuracy of the external reference frequency. For instance, if the external reference frequency is 124.988277868 MHz rather than 125 MHz, the RTC loses approximately 7.4 seconds over 24 hours.

6.13.5 Alarm Wake-up Comparator

The RTC has two modes of operation, selectable through the clock mode bit.

In Calendar mode, the RTC counts seconds, minutes, hours, days, month, years, weekdays, and weeks. In Binary mode, the RTC consecutively counts from 0 all the way to 2^{43} - 1. In both the modes, the alarm event generation logic simply compares the content of the Alarm register with that of the RTC; when they are equal, the RTC_MATCH output is asserted.

6.13.6 Register Map and Descriptions

See the configuration and register maps in the ZIP document on the PIC64GX1000 product page www.microchip.com/PIC64GX1000.

6.14 Timers

The PIC64GX system Timer (hereinafter referred as Timer) consists of two programmable 32-bit decrementing counters that generate interrupts to the processor.

6.14.1 Features

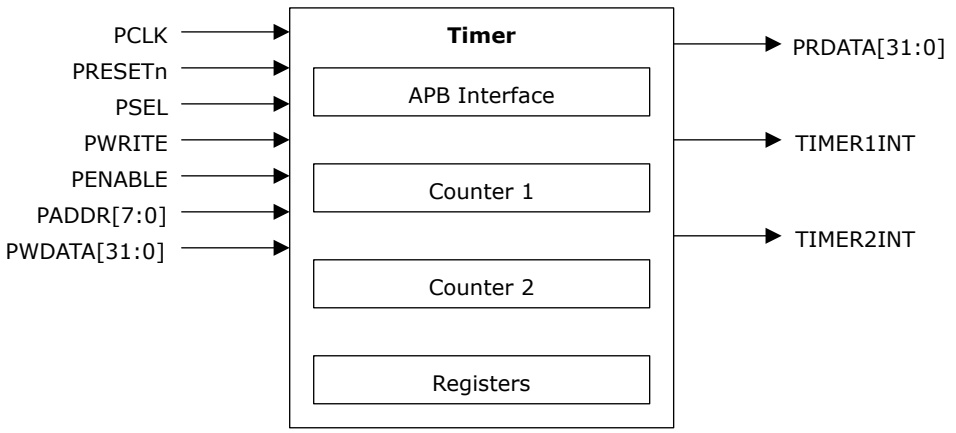
The timer supports the following features:

- Two count modes: One-shot and Periodic
- Decrementing 32-bit counters
- Two 32-bit timers can be concatenated to create a 64-bit timer
- Option to enable or disable the interrupt requests when timer reaches zero
- Controls to start, stop, and reset the timer

6.14.2 Functional Description

The Timer is an APB slave that provides two programmable, interrupt generating, 32-bit decrementing counters, as shown in the following figure. The counters generate the interrupts TIMER1INT and TIMER2INT on reaching zero.

Figure 6-15. Timer Block Diagram



© 2024 Microchip Technology Inc. and its subsidiaries

6.14.3 Register Map and Descriptions

See the configuration and register maps in the ZIP document on the PIC64GX1000 product page www.microchip.com/PIC64GX1000.

6.14.4 Timer Input Clock

Table 6-35. Timers

Parameter	Symbol	Min	Max	Units
Timer input clock frequency	$F_{TIMERCLK}$	—	156.25	MHz

6.15 MIPI CSI-2

This section will be added later

© 2024 Microchip Technology Inc. and its subsidiaries

7. PCIe Root Port

PCI Express (PCIe®) is a scalable, high-bandwidth serial interconnect technology that maintains compatibility with existing PCI systems. Microchip's PIC64GX1000 series MPUs contain fully integrated PCIe root port subsystems with optimized embedded controller blocks which include the physical layer interface of the transceiver for the PCI Express (PIPE) interconnection.

This section describes the PCIe subsystem available in the PIC64GX1000 family of devices.

The PIC64GX1000 devices include one embedded PCIe subsystem (PCIESS) blocks.

7.1 PCIe Subsystem Overview

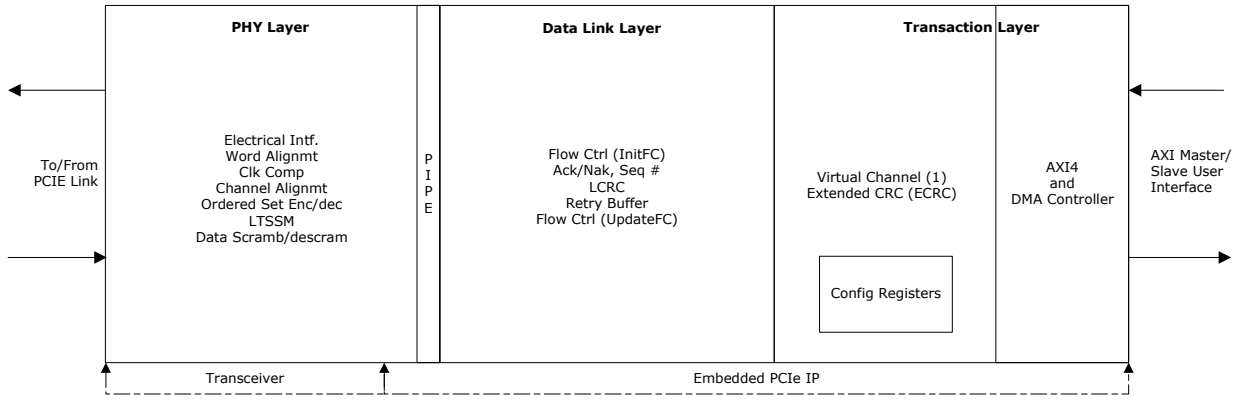
The following table provides an overview of the PCIe subsystem.

Table 7-1. PCIe Subsystem Components

Component	PIC64GX1000-(V or C)/FCS	PIC64GX1000-(V or C)/FCV
Physical layer of XCVR	✓	✓
PCIe Data link layer (DL) and Transaction layer (TL))	✓	✓
Bridge Layer	✓	✓
AXI4 Layer	✓	✓
Number of PCIe controllers	1	1
Lane Configuration	x1	X1, x2, x4

The PCIESS supports AMBA AXI4 Initiator/target user interface functionality between the AXI4 and PCIe systems.

Figure 7-1. PCIe Functional Layers of PCIESS



The PCIESS is compliant with the following standards:

- PCI Express Base Specification Revision 3.0
- PCI Express Card Electromechanical (CEM) Specification Revision 2.0
- PCI Industrial Computer Manufacturers Group (PICMG) 3.4
- PCI Bus Power Management Interface Specification Revision 1.2
- AMBA AXI Protocol Specification, Version 2.0 – ARM, March 2010

© 2024 Microchip Technology Inc. and its subsidiaries

7.1.1 Features

The PCIESS supports the following features:

- PCIE 2.0 compliant with 2.5 and 5.0 Gbps line speeds
- x1, x2, and x4 Root Port with support for x1, x2, and x4 end points[1]
- Root port support for up to two 32-bit or one 64-bit BAR
- Two fully-independent Direct Memory Access (DMA) engines with Scatter-Gather DMA (SGDMA) support
- One virtual channel (VC)
- Single-function capability
- Maximum payload size (MPS) of up to 256 bytes
- Advanced error reporting (AER) support
- Lane reversal/polarity inversion
- Legacy PCI power management
- Native active-state power management L0 and L1 support
- Power management event (PME) message
- Latency tolerance reporting (LTR)
- End-to-end data integrity

Note:

1. Only x1 for the FCS device

7.1.2 Functional Descriptions

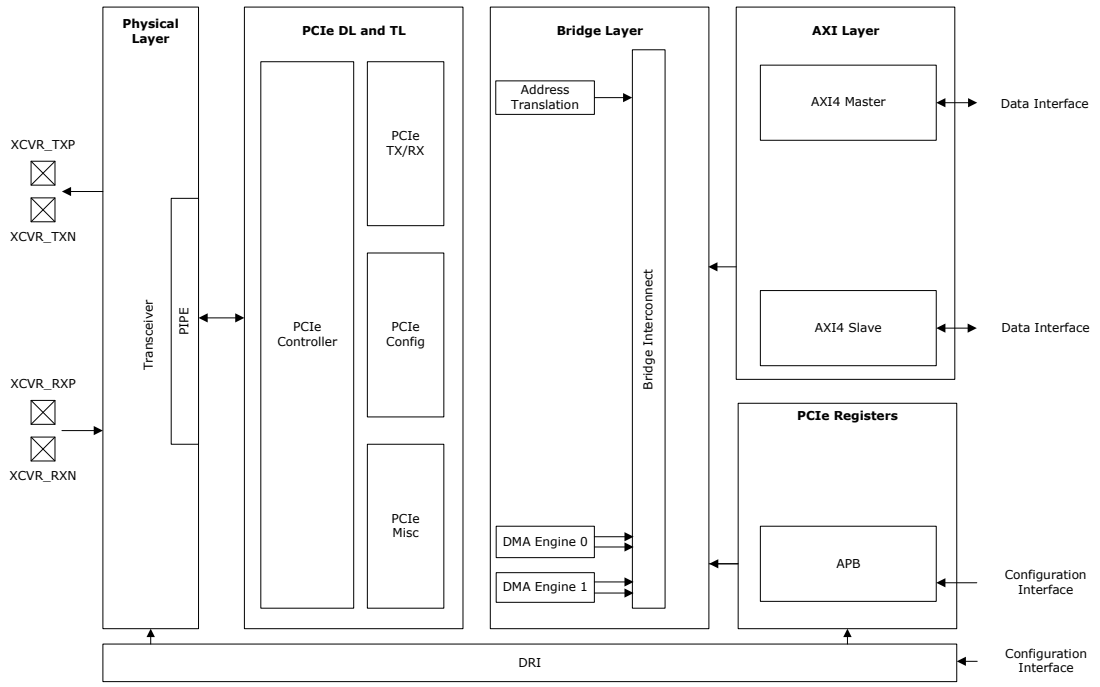
The functional details of the PCIe subsystem are described in this chapter.

As shown in the following figure, the PCIESS is composed of four sub-modules:

- Physical layer
- PCIe DL and TL
- Bridge layer
- AXI layer
- Configuration interface (DRI and APB)

© 2024 Microchip Technology Inc. and its subsidiaries

Figure 7-2. Embedded PCIESS Architecture



7.2 Physical Layer Interface

A PCIe lane consists of a pair of differential transmit signals and a pair of differential receive signals. The lanes are organized, as listed in the following table.

Table 7-2. Lane Configuration

x1	x2	x4
PCIe1 (Lane2)	PCIe1 (Lane2, Lane3)	PCIe1 (Lane 0, Lane1, Lane2, and Lane3)

Note: FCS package only supports x1 .

7.2.1 Physical Layer Functions

The physical layer is an electrical physical media attachment (PMA) required to connect to a PCIe system, and supports the following features:

- PCI Express 2.0 electrical compliance
- 2.5 and 5.0 Gbps common-mode logic (CML) electrical interface
- Signal integrity programmability including differential output voltage, transmitter de-emphasis, and receiver-side continuous-time linear equalization (CTLE)
- ±300 ppm clock-tolerance compensation
- Serializer and de-serializer
- 8b/10b symbol encoding/decoding
- Symbol alignment
- Framing and application of symbols to lanes
- Lane to lane Tx de-skew

© 2024 Microchip Technology Inc. and its subsidiaries

7.2.1.1 Receiver

The transceiver input includes all the features required to build a PCIe interface, such as input level sensitivity, signal detection, and termination.

7.2.1.2 Transmitter

The transceiver output includes features such as output swing, termination, and de-emphasis.

> **Important:** If the receiver detection circuitry within the transmitter is bypassed, the REFCLK is available, the TX PLL is locked and the Link Training and Status State Machine (LTSSM) moves to the "polling compliance" state (that is, when 50 Ω is terminated to the transmitter) the link training will remain unaffected when an active receiver is connected.

This causes the optional key-sight protocol test card (PTC) LOOPBACK_THROUGH_CONFIG test to fail. To pass the test, set the PMA_RXPLL_FLOCK_SEL bit to 0 when LTSSM is in "polling compliance" state and set to 1 in other states.

7.2.1.2.1 Reference Clock

For PCIe applications, a differential 125 MHz reference clock with a ±300 ppm tolerance is used by the transceiver transmit PLL and CDR PLL to generate the required 125 MHz output clock (depending on the lane speed settings), which is passed to the embedded PCIESS.

The transceiver reference clock inputs accept LVDS/CML/HCSL input clock signals according to PCIe specifications. Proper termination is included as required by the specification.

According to PCIe specifications, upstream and downstream PCIe devices must transmit data at a rate within 600 ppm for each other at all times. This specification allows a reference clock with a ±300 ppm tolerance. To ensure that the minimum clock period is not violated, the PCIESS uses a spread-spectrum technique that does not permit modulation above the nominal frequency.

The data rate can be modulated from 0% to 0.5% of the nominal data rate frequency at a modulation rate ranging from 30 kHz to 33 kHz. Along with the ±300 ppm tolerance limit, both ports require the same bit-rate clock when the data is modulated using spread-spectrum clocking (SSC).

The PIC64GX1000 devices support the following clocking topologies defined by the PCIe specifications: common Refclk and separate Refclk.

- The Common Refclk is the most widely supported clocking method in open systems where the root port or root complex provides a clock to the endpoint. An advantage of this clocking architecture is that it supports SSC, which reduces electromagnetic interference (EMI).
- The Separate Refclk uses two independent clock sources: one each for the root and the endpoint. The clock sources must maintain ± 300 ppm frequency accuracy and cannot use SSC.

7.2.1.2.2 Low-Power States

The PCIESS supports PCIe low-power operation states known as L0s and L1 states.

L0s (Autonomous electrical idle): This state reduces power during short intervals of idle. Devices must transition to L0s independently on each direction of the link.

L1 (Directed electrical idle): This state reduces power when the downstream port directs the upstream ports. This state saves power in two ways:

- Shutting down the transceiver circuitry and associated PLL.
- Significantly decreasing the number of internal core transitions.

© 2024 Microchip Technology Inc. and its subsidiaries

Important: PCIe Link Training and Status State Machine (LTSSM) does not support the L2/P2 power management link state.

- Software-driven L2/P2 entry commands issued by the PIC64GX1000 PCIESS Root Port to the downstream endpoints are not supported to the down-stream end-points. As a Root Port, this causes the link to be completely disrupted and only recoverable by re-initializing the link with side-band PERSTn (fundamental reset) or a power cycle.

Important: The following are the debug recommendations when LTSSM does not reach L0.

- A third-party logic analyzer for PCI Express records the traffic on the physical link and decodes traffic. A third-party logic analyzer can show the two-way traffic at different levels for different requirements. For high-level diagnostics, the analyzer shows the LTSSM flows for devices on both side of the link side-by-side. This display can help you see the link training handshake behavior and identify where the traffic is stuck. You can also verify the contents of packets displayed on traffic analyzer.

7.2.1.3 Reference Clock

For PCIe applications, a differential 125 MHz reference clock with a ±300 ppm tolerance is used by the transceiver transmit PLL and CDR PLL to generate the required 125 MHz output clock (depending on the lane speed settings), which is passed to the embedded PCIESS.

The transceiver reference clock inputs accept LVDS/CML/HCSL input clock signals according to PCIe specifications. Proper termination is included as required by the specification.

According to PCIe specifications, upstream and downstream PCIe devices must transmit data at a rate within 600 ppm for each other at all times. This specification allows a reference clock with a ±300 ppm tolerance. To ensure that the minimum clock period is not violated, the PCIESS uses a spread-spectrum technique that does not permit modulation above the nominal frequency.

The data rate can be modulated from 0% to 0.5% of the nominal data rate frequency at a modulation rate ranging from 30 kHz to 33 kHz. Along with the ±300 ppm tolerance limit, both ports require the same bit-rate clock when the data is modulated using spread-spectrum clocking (SSC).

The PIC64GX1000 devices support the following clocking topologies defined by the PCIe specifications: common Refclk and separate Refclk.

- The Common Refclk is the most widely supported clocking method in open systems where the root port or root complex provides a clock to the endpoint. An advantage of this clocking architecture is that it supports SSC, which reduces electromagnetic interference (EMI).
- The Separate Refclk uses two independent clock sources: one each for the root and the endpoint. The clock sources must maintain ± 300 ppm frequency accuracy and cannot use SSC.

7.2.1.4 Low-Power States

The PCIESS supports PCIe low-power operation states known as L0s and L1 states.

L0s (Autonomous electrical idle): This state reduces power during short intervals of idle. Devices must transition to L0s independently on each direction of the link.

L1 (Directed electrical idle): This state reduces power when the downstream port directs the upstream ports. This state saves power in two ways:

- Shutting down the transceiver circuitry and associated PLL.
- Significantly decreasing the number of internal core transitions.

© 2024 Microchip Technology Inc. and its subsidiaries

Important: PCIe Link Training and Status State Machine (LTSSM) does not support the L2/P2 power management link state.

- Software-driven L2/P2 entry commands issued by the PIC64GX1000 PCIESS Root Port to the downstream endpoints are not supported to the down-stream end-points. As a Root Port, this causes the link to be completely disrupted and only recoverable by re-initializing the link with side-band PERSTn (fundamental reset) or a power cycle.

Important: The following are the debug recommendations when LTSSM does not reach L0.

- A third-party logic analyzer for PCI Express records the traffic on the physical link and decodes traffic. A third-party logic analyzer can show the two-way traffic at different levels for different requirements. For high-level diagnostics, the analyzer shows the LTSSM flows for devices on both side of the link side-by-side. This display can help you see the link training handshake behavior and identify where the traffic is stuck. You can also verify the contents of packets displayed on traffic analyzer.

7.3 PCIe Configuration Space

The following tables list the layout of the PCI Express configuration space and provides the mapping for each register in the space.

Table 7-3. PCI Configuration Space

Offset	Description
0x00 to 0x03C	Type0 (endpoint) or Type1 (Root port/Bridge/Switch) Standard PCI configuration header
0x040 to 0x07C	Reserved
0x080 to 0x0B8	PCI Express Capability
0x0BC to 0x0CC	Reserved
0x0DC	Reserved
0x0E0 to 0x0F4	MSI Capability
0x0F8 to 0x0FC	PCI Power Management Capability

Table 7-4. PCI Express Capability Structure

Byte Offset	Bit Number			
	31:24	23:16	15:8	7:0
0x080	Capability Register		Next capability Pointer	Capability ID
0x084	Device capabilities			
0x088	Device status		Device control	
0x08C	Link capabilities			
0x090	Link status		Link control	
0x094	Slot capabilities			
0x098	Slot status		Slot control	
0x09C	Root capabilities		Root control	
0x0A0	Root Status			
0x0A4	Device capabilities 2			

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued				
Byte Offset	**Bit Number**			
0x0A8	Device status 2		Device control 2	
0x0AC	Link capabilities 2			
0x0B0	Link status 2		Link control 2	
0x0B4	Slot capabilities 2			
0x0B8	Slot status 2		Slot control 2	

Table 7-5. MSI Capability Structure

Byte Offset	Bit Number			
	31:24	23:16	15:8	7:0
0x0E0	Message Control		Next pointer	Capability ID
0x0E4	Message Address			
0x0E8	Message Upper Address			
0x0EC			Message Data	

Table 7-6. Power Management Capability Structure

Byte Offset	Bit Number			
	31:24	23:16	15:8	7:0
0x0F8	Power management capabilities		Next item pointer	Capability ID
0x0FC	Data	PMCSR_BSE bridge support extensions	Power management control and status registers	

Table 7-7. PCI Express Extended Configuration Space

Offset	Description
0x100 to 0x104	Vendor-specific capability with VSECID = 1556h; RevID = 1h
0x108 to 0x10C	Latency Tolerance Reporting capability
0x200 to 0x234	Advanced Error Reporting capability

Table 7-8. Vendor Specific Extended Capability Structure

Byte Offset	Bit Number			
	31:24	23:16	15:8	7:0
0x100	Vendor-Specific Extended Capability Header			
0x104	Vendor-Specific Header			

Table 7-9. Latency Tolerance Reporting Capability Structure

Byte Offset	Bit Number			
	31:24	23:16	15:8	7:0
0x108	PCI Express Extended Capability Header			
0x10C	Max No-Snoop Latency Register		Max Snoop Latency Register	

Table 7-10. Advanced Error Reporting Capability Structure

Byte Offset	Bit Number			
	31:24	23:16	15:8	7:0
0x200	PCI Express Enhanced Capability Header			
0x204	Uncorrectable Error Status Register			

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued			
Byte Offset	**Bit Number**		
0x208	Uncorrectable Error Mask Register		
0x20C	Uncorrectable Error Severity Register		
0x210	Correctable Error Status Register		
0x214	Correctable Error Mask Register		
0x218	Advanced Error Capabilities and Control Register		
0x21C	Header Log Register		
0x22C	Root Error Command		
0x230	Root Error Status		
0x234	Error Source Identification Register		

7.4 Board Design Recommendations

This chapter discusses board-level implementation details of a PCIe design using the PIC64GX family of devices. Optimal performance requires understanding the functionality of the device pins and properly addressing issues such as device interfacing, protocol specifications, and signal integrity.

Various specifications from PCI-SIG apply depending on the form factor of the design. This chapter focuses on a subset of these specifications centered on chip-to-chip and add-in card implementations.

For more information, see the PCI Express Base Specification, Revision 2.0members.pcisig.com/wg/PCI-SIG/document/8246and PCI Express Card Electromechanical Specification (CEM) Revision 2.0 from PCI-SIG.

7.4.1 AC-Coupling

PCIe electrical signals require a 75 - 200 nF AC-coupling capacitor between the transmitter and receiver. All transmitters are AC-coupled, either within the media or within the transmitting component itself. If located within the media, the AC-coupling capacitors are placed close to the transmitter. The AC-coupling capacitor is used in conjunction with internal termination for PCIe link detection.

7.4.2 Lane Reversal

The PCIESS supports lane reversal when required, allowing the PCIe physical I/O to be reversed with the block's logical lanes for a more flexible PCB layout. Lane reversal functionality is incorporated into the PCIESS to be layout-agnostic with respect to lane ordering and lane polarity. Using lane reversal eases routing congestion on the PCB, leading to a cleaner interface between the PCIe host and the endpoint or root port.

7.4.3 Polarity Inversion

The transceiver block with PCIESS supports differential polarity inversion. Receiver polarity is automatically detected by the PCIESS during link training, as defined in the PCIe specification. The differential data received by the transceiver RX are reversed if RXP and RXN differential traces are swapped on the PCB, accidentally. The transceiver RX inversion allows within the PCIESS to offset the reversed polarity of a serial differential pair.

7.4.4 PCIe Power-Up

The PCIe specification provides timing requirements for power-up. The PCIe connector specification specifies that the fundamental reset (PERST_N) be de-asserted at a minimum of 100 ms from the point of stable power. The PCIe PERST_N signal release time (known as PCIe timing parameter TPVPERL) of 100 ms is used for the PCIe card electro-mechanical specification for add-in cards.

© 2024 Microchip Technology Inc. and its subsidiaries

The semi-autonomous nature of the PCIESS in a device allows the device to quickly move from power-up to link detect. The transceiver initially terminates to 50 kΩ for hot-swap protection but quickly returns to 100 Ω termination so that link detection operates within the PCIe specifications. When the device is detected by the root, it proceeds to the polling state of the LTSSM. The link then cycles through the remaining LTSSM states. In cases where the root point and the endpoint power-up separately, the PERST_N signal must be used to handshake the link startups.

PERST_N is a system-level requirement for PCIe system as defined by the PCIe specification. The PERST_N pin resets the PL, TL domains, but it does not reach the AXI, bridge logic, or bridge map registers in the PCIESS core within the device.

Board Design Recommendations

7.4.5 PCIe Edge Connector

PCIe is a point-to-point serial differential low-voltage interconnect supporting up to four channels. Each lane consists of two pairs of differential signals: transmit pair, receive pair, XCVR_x_TXy_P/N, and XCVR_x_RXy_P/N. Each signal has a 2.5 GHz embedded clock.

7.5 PCI-SIG TxPLL Electrical Compliance Test

Due to variations in board topologies, it might be necessary to adjust the Tx amplitude characteristics to ensure compliance with electrical requirements of the Peripheral Component Interconnect Special Interest Group (PCI-SIG). The following fields, of the SER_DRV_CTRL_M# register, are available to provide this control:

- TXDRVTRIM_FS_#P#B_M#
- TXDRVTRIM_HS_0DB_M#

Where:

- #P#B indicates the requested amount of de-emphasis
- M# indicates the requested TxSwing value

These register fields consist of two 3-bit controls:

- The most significant 3 bits control the cursor amplitude. Increasing this value by 1 (for example, from 4 to 5) increases the amplitude and height of the inner eye by 20 mV, approximately. Decreasing this value by 1 has the opposite effect.
- The least significant 3 bits control the post-cursor amplitude. Increasing this value by 1 (for example, from 4 to 5) increases the amplitude by approximately 20 mV and decreases the height of the inner eye by the same amount. Decreasing the value by 1 has the opposite effect.

To pass the PCISIG's Tx PLL bandwidth electrical compliance test on the ICICLE kit, the following settings were used:

- TxMARGIN was set to 0x0 and TXSWING was set to 0x1
- Modified the register field TXDRVTRIM_FS_3P5DB_M0 from 0x0A to 0x00. This lowered the total amplitude by 75 mV.

Modified the register field TXDRVTRIM_FS_6P0DB_M0 from 0x24 to 0x13.

© 2024 Microchip Technology Inc. and its subsidiaries

8. Power

The following are recommendations for powering the PIC64GX1000 family MPUs.

Table 8-1. Required Supply Voltages

Voltage	Current	PIC64GX Input	Microchip Component
1.0V	7A	—	MIC22705YML-TR
3.3V	7A	VDDAUX1, VDDAUX2, VDDI1, VDDI2, VDDI3, VDDI5	MIC22705YML-TR
1.2V	4A	DDR4, VDDI6	MIC22405YML-TR
2.5V	5A	PLL, VDDA25, VDD25, VDD_XCVR_CLK, VDDAUX4	MIC69502WR
1.8V	4A	VDD18, VDDI4	MIC22405YML-TR
0.6V	—	VREF_DDR4, VTT_DDR4	MIC5166YML-TR
—	—	—	—

© 2024 Microchip Technology Inc. and its subsidiaries

9. Safety and Security Features

The PIC64GX1000 MPU is designed to address the high-reliability requirements of safety-critical systems in industrial, aviation, military, and communication applications with the following features:

- Secure Non-Volatile Memory (sNVM)
- Secure boot functionality
- Physical Memory Protection per hart
- Memory Protection Unit for peripherals
- Device Level Anti-Tamper Features
- User Crypto-processor
- ECC on RAMs

9.1 Secure Non-Volatile Memory (sNVM)

The sNVM block is a user non-volatile flash memory that can be programmed independently. Each device has 56 Kbytes of sNVM. The sNVM is organized into 224 pages, each page is 256 bytes in size. Three pages are reserved for administrative purposes, leaving 221 pages available for user data. Individual pages in the sNVM can be designated as write-protected (ROM).

The sNVM can be written using system service at run time. The data written to the sNVM can be protected by a device unique intrinsic PUF secret key (SMK) using AES-256 in the synthetic initialization vector (SIV) mode.

The data may be stored in any of the following formats (listed in the ascending order of access time) in sNVM:

- Non-authenticated plaintext
- Authenticated plaintext
- Authenticated ciphertext

Non-authenticated plaintext provides the fastest access time and authenticated ciphertext is the slowest but provides the highest level of security. For authenticated plaintext or ciphertext, a user provided user sNVM key (USK) is used for authentication during read. When the user data is stored in non-authenticated format, 252 bytes of storage per page is available for user data. When the user data is stored in authenticated format, 236 bytes of storage per page is available for user data. If the data is programmed using authentication, the USK key used at the time of programming must be provided while retrieving the data using system service call.

9.2 Secure Boot

The MPU comes with two secure boot options to securely boot the application processors. For the default secure boot method, the system controller copies the Microchip secure boot loader from its private, secure memory area and load it into the 8 KB DTIM of the E51 monitor core. After that, the reset is released to the application CPUs and then the secure boot code starts execution. The default secure boot loader performs a signature check on the 128 KB eNVM, then run a hash on the eNVM image. If no errors are reported, the code jumps to the user application stored in the eNVM. If errors are reported, the system controller activates a tamper alarm that asserts a signal to the MPU. Users can then decide on a plan of action.

The second secure boot method allows users to place their own boot code in the secure non-volatile memory (sNVM) area of the chip. The sNVM is a 56 KB nonvolatile memory that can be protected by the built-in Physically Unclonable Function (PUF), that is, the unique PUF ID can serve as an initialization vector for an AES encrypt/decrypt operation performed by the side-channel resistant system controller co-processor. On power-up, the system controller copies the user code from

© 2024 Microchip Technology Inc. and its subsidiaries

sNVM and write it to the E51 monitor core DTIM. From there, user custom secure boot loader starts executing.

9.3 Physical Memory Protection

Each hart in the MPU includes a physical memory protection (PMP) unit compliant with the RISC-V Instruction Set Manual, Volume II: Privileged Architecture, Version 1.10. The PMP unit can be used to set memory access privileges (read, write, execute) for specified memory regions. Each PMP supports 16 regions with a minimum region size of 4 bytes. It is permitted to have overlapping regions. PMP unit can be used to restrict access to memory and isolate processes from each other. The PMP allows for region locking whereby once a region is locked, further writes to the configuration and address registers are ignored. Locked PMP entries may only be unlocked with a system reset.

PMPs are configured by the HSS on system boot based on the user configuration.

9.4 Memory Protection Unit

Random access to memory regions by any non-hart initiator can corrupt the memory and the overall system. To avoid random access to memory, the MPU includes a built-in memory protection unit for each non-hart master. The GEM0, GEM1, eMMC, USB, SCB, Crypto Co-Processor initiator blocks interface through the memory protection unit. The memory protection unit can be used to create access regions in memories for a particular peripheral and define privileges to those access regions. At reset, all memory regions are restricted.

The memory protection unit monitor transactions on the AXI bus and only legal accesses are permitted. Illegal transactions result in an AXI fault.

The MPU is configured by the HSS on system boot based on the user configuration.

9.5 Device-Level Anti-Tamper Features

The PIC64GX1000 includes a number of built-in tamper detection and response capabilities that can be used to enhance the security of the device. These countermeasures are intended to address various types of attacks that include non-invasive, semi-invasive, and invasive attacks.

The anti-tamper system present in device includes voltage, frequency, and temperature monitors. When a tamper condition is detected, an interrupt is generated to the harts.

The devices also incorporate DPA countermeasures for all built-in design security protocols to protect the secret keys from discovery using side-channel analysis.

9.6 User Crypto-processor

The PIC64GX1000 includes a dedicated crypto co-processor (referred to as the user crypto-processor) for data security applications. The user crypto-processor is an Athena TeraFire EXP-F5200B cryptography microprocessor. It provides complete support for the Commercial National Security Algorithm (CNSA) suite and beyond, and also includes side-channel analysis (SCA) resistant cryptographic countermeasures. These countermeasures provide strong resistance against SCA attacks such as SPA and DPA.

The user crypto-processor supports numerous cryptographic algorithms, including the following:

- AES with 128-bit, 192-bit, and 256-bit key sizes in ECB, CBC, CFB, OFB, CTR, and GCM modes
- AES key wrap and unwrap
- SHA1, SHA2-224, SHA2-256, SHA2-384, and SHA2-512
- AES-CMAC and AES-GMAC
- HMAC-SHA
- True random number generation (non-deterministic random bit generator plus NIST SP800-90A deterministic random bit generator)

© 2024 Microchip Technology Inc. and its subsidiaries

- RSA, DSA, and modular exponentiation (Diffie-Hellman) with key sizes up to 3072-bits
- EC key pair generation, point validation, point multiplication (EC Diffie-Hellman), and ECDSA for NIST P-curves: P-192, P-224, P-256, P-384, and P-521. Brainpool curves: P-256, P-384, and P-512
- Key-tree function

The user crypto-processor also incorporates an TRNG. The user crypto-processor specifically supports an TRNG combined with an AES counter mode-based DRBG, compliant with NIST SP800-90A.

Many of the commonly used cryptographic operations available are certified by an independent third-party NIST-accredited security laboratory under the NIST cryptographic algorithm validation program (CAVP) scheme. This includes the AES, SHA, HMAC, ECDSA, RSA, DSA, and DRBG implementations, providing a high level of assurance that they are implemented correctly. The following table lists the CAVP validation numbers, see the NIST CAVP website for details on the specific algorithms and modes that are certified.

Table 9-1. Table 10-1: NIST CAVP Validation Numbers

Algorithm	CAVP Number
AES	3950
SHA1/2	3258
HMAC	2573
DSA	1077
RSA	2018
ECDSA	867
DRBG	1153

9.7 ECC Operation in RAMs

All RAM on the MPU (including caches, user crypto-processor, PCIe). Single bit errors are corrected and an interrupt can optionally be generated for the user. Double bit errors will result in an error being generated on the system bus an interrupt. The handling of a double bit error is up to the user – if data can be re-requested (for example, a PCIe double bit error) or in some cases a system reset may be required.

© 2024 Microchip Technology Inc. and its subsidiaries

10. Event System

10.1 Interrupts

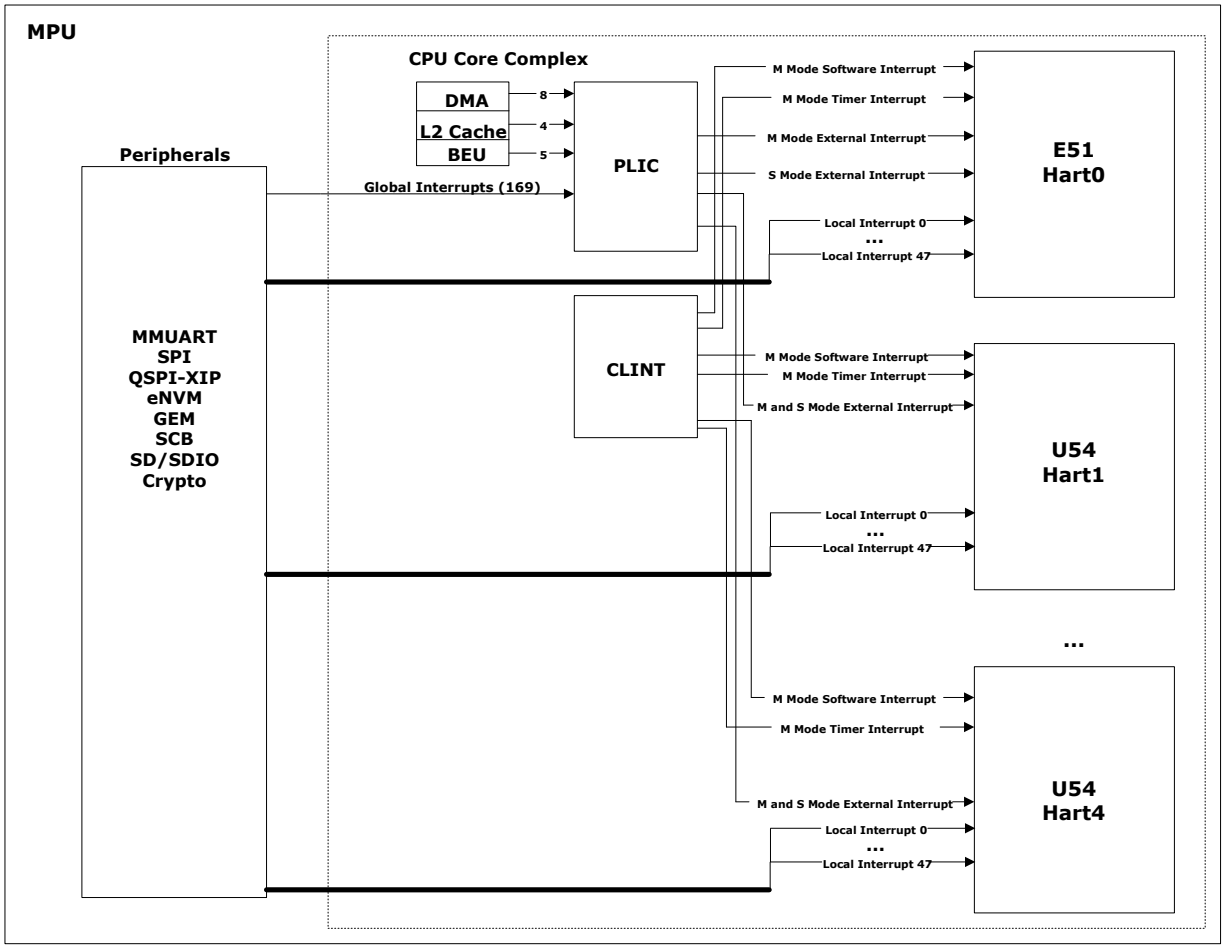
Each processor core supports Local and Global interrupts. 48 interrupts from peripherals are directly connected as Local interrupts to each processor core. Local interrupts are handled faster than the Global interrupts. The Core Local Interrupt Controller (CLINT) block generates Software and Timer Interrupts which are also Local interrupts.

169 interrupts from peripherals and 16 interrupts from the CPU Core Complex blocks—DMA Engine, BEU, and L2 Cache are connected to the Platform-Level Interrupt Controller (PLIC) as Global interrupts. The PLIC asserts Global interrupts to a specific processor core. The user can configure the PLIC registers to perform the following:

- Enable the required Global interrupts
- Route the interrupt to a specific core
- Assign priority to those interrupts
- Assign priority threshold levels

Some application critical Global interrupts can also be routed as Local interrupts. All interrupts are synchronized with the AXI/CPU clock domain for relaxed timing requirements. For a Hart, the latency of Global interrupts increases with the ratio of the core clock frequency to the clock frequency.

Figure 10-1. Interrupt Scheme



© 2024 Microchip Technology Inc. and its subsidiaries

Table 10-1 lists the Local and Global interrupts implemented in the MPU.

For example:

- The spi0 interrupt signal is a Global interrupt because it is not connected to any Hart as a Local interrupt. This interrupt signal is connected to the PLIC.
- The mac0_int interrupt signal is a Local interrupt to Hart1 and Hart2. It can also be enabled as a Global interrupt via the PLIC to Hart0, Hart3, and Hart4.

Table 10-1. Routing of Interrupts to Processor Cores

Interrupt	Width	Global_int	IRQ	Hart0	Hart1	Hart2	Hart3	Hart4	M2Fvect	M2F-Int	U54-Mask
MPU_INT_F2M[63:32]	32	[168:137]	[181:150]	[47:16]	—	—	—	—	—	—	—
MPU_INT_F2M[31:0]	32	[136:105]	[149:118]	—	[47:16]	[47:16]	[47:16]	[47:16]	—	—	MASKED
gpio0/2	14	[13:0]	[26:13]	—	—	—	—	—	[13:0]	0	—
gpio1/2	24	[37:14]	[50:27]	—	—	—	—	—	[37:14]	0	—
gpio0_non_direct	1	38	51	—	—	—	—	—	38	0	—
gpio1_non_direct	1	39	52	—	—	—	—	—	39	0	—
gpio2_non_direct	1	40	53	—	—	—	—	—	40	0	—
spi0	1	41	54	—	—	—	—	—	41	1	—
spi1	1	42	55	—	—	—	—	—	42	1	—
can0	1	43	56	—	—	—	—	—	43	1	—
can1	1	44	57	—	—	—	—	—	44	1	—
i2c0_main	1	45	58	—	—	—	—	—	45	2	—
i2c0_alert	1	46	59	—	—	—	—	—	46	2	—
i2c0_sus	1	47	60	—	—	—	—	—	47	2	—
i2c1_main	1	48	61	—	—	—	—	—	48	2	—
i2c1_alert	1	49	62	—	—	—	—	—	49	2	—
i2c1_sus	1	50	63	—	—	—	—	—	50	2	—
mac0_int	1	51	64	—	8	8	—	—	51	3	MASKED
mac0_queue1	1	52	65	—	7	7	—	—	52	3	MASKED
mac0_queue2	1	53	66	—	6	6	—	—	53	3	MASKED
mac0_queue3	1	54	67	—	5	5	—	—	54	3	MASKED
mac0_emac	1	55	68	—	4	4	—	—	55	3	MASKED
mac0_mmsl	1	56	69	—	3	3	—	—	56	3	MASKED
mac1_int	1	57	70	—	—	—	8	8	57	4	MASKED
mac1_queue1	1	58	71	—	—	—	7	7	58	4	MASKED
mac1_queue2	1	59	72	—	—	—	6	6	59	4	MASKED
mac1_queue3	1	60	73	—	—	—	5	5	60	4	MASKED
mac1_emac	1	61	74	—	—	—	4	4	61	4	MASKED
mac1_mmsl	1	62	75	—	—	—	3	3	62	4	MASKED
ddrc_train	1	63	76	—	—	—	—	—	63	9	—
scb_interrupt	1	64	77	15	—	—	—	—	64	7	—
peripheral_ecc_error[1]	1	65	78	14	—	—	—	—	65	6	—
peripheral_ecc_correct[1]	1	66	79	13	—	—	—	—	66	6	—
rtc_wakeup	1	67	80	—	—	—	—	—	67	11	—
rtc_match	1	68	81	—	—	—	—	—	68	11	—

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

Interrupt	Width	Global_int	IRQ	Hart0	Hart1	Hart2	Hart3	Hart4	M2Fvect	M2F-Int	U54-Mask
timer1	1	69	82	—	—	—	—	—	69	12	—
timer2	1	70	83	—	—	—	—	—	70	12	—
envm	1	71	84	12	—	—	—	—	71	13	—
qspi	1	72	85	—	—	—	—	—	72	13	—
usb_dma	1	73	86	—	—	—	—	—	73	14	—
usb_mc	1	74	87	—	—	—	—	—	74	14	—
mmc_main	1	75	88	—	—	—	—	—	75	15	—
mmc_wakeup	1	76	89	—	—	—	—	—	76	15	—
mmuart0	1	77	90	11	—	—	—	—	77	1	—
mmuart1	1	78	91	—	11	—	—	—	78	1	—
mmuart2	1	79	92	—	—	11	—	—	79	1	—
mmuart3	1	80	93	—	—	—	11	—	80	1	—
mmuart4	1	81	94	—	—	—	—	11	81	1	—
wdog0_mvrp	1	87	100	10	—	—	—	—	87	5	—
wdog1_mvrp	1	88	101	—	10	—	—	—	88	5	—
wdog2_mvrp	1	89	102	—	—	10	—	—	89	5	—
wdog3_mvrp	1	90	103	—	—	—	10	—	90	5	—
wdog4_mvrp	1	91	104	—	—	—	—	10	91	5	—
wdog0_tout	1	92	105	9	—	—	—	—	92	5	—
wdog1_tout	1	93	106	8	9	—	—	—	93	5	—
wdog2_tout	1	94	107	7	—	9	—	—	94	5	—
wdog3_tout	1	95	108	6	—	—	9	—	95	5	—
wdog4_tout	1	96	109	5	—	—	—	9	96	5	—
g5c_devrst	1	82	95	4	—	—	—	—	82	10	—
g5c_message	1	83	96	3	—	—	—	—	83	8	—
usoc_vc_interrupt	1	84	97	2	—	—	—	—	84	11	—
usoc_smb_interrupt	1	85	98	1	—	—	—	—	85	11	—
pll_event	1	86	99	0	—	—	—	—	86	6	—
mpu_fail	1	86	99	0	—	—	—	—	86	6	—
decode_error	1	86	99	0	—	—	—	—	86	6	—
lp_state_enter	1	86	99	0	—	—	—	—	86	6	—
lp_state_exit	1	86	99	0	—	—	—	—	86	6	—
ff_start	1	86	99	0	—	—	—	—	86	6	—
ff_end	1	86	99	0	—	—	—	—	86	6	—
scb_error	1	86	99	0	—	—	—	—	86	6	—
scb_fault	1	86	99	0	—	—	—	—	86	6	—
mesh_fail	1	86	99	0	—	—	—	—	86	6	—
volt_temp_alarm	1	98	111	—	—	—	—	—	98	No	—
athena_complete	1	99	112	—	—	—	—	—	NA	No	—
athena_alarm	1	100	113	—	—	—	—	—	NA	No	—
athena_buserror	1	101	114	—	—	—	—	—	NA	No	—
usoc_axic_us	1	102	115	—	—	—	—	—	102	11	—
usoc_axic_ds	1	103	116	—	—	—	—	—	103	11	—
reserved/spare	11	[104]	[117]	0	7	7	7	7	NA	—	—

© 2024 Microchip Technology Inc. and its subsidiaries

Note: The peripheral_ecc_error and peripheral_ecc_correct interrupts are driven by the EDAC_SR register. To implement ECC for peripherals, ECC interrupts are enabled individually by writing to the EDAC_INTEN_CR register. ECC interrupts are cleared by writing to the EDAC_SR register. ECC error count is available in EDAC_CNT_[peripheral] registers.

To enable all Local interrupts on the U54_1 core, set the FAB_INTEN_U54_1 register using the `SYSREG->FAB_INTEN_U54_1 = 0xffffffff`instruction. This instruction enables all MPU_INT_F2M[31:0] interrupts to interrupt U54_1 directly. Similarly, enable the Local interrupts on U54_2, U54_3, and U54_4 cores.

By default, all Local interrupts MPU_INT_F2M[63:32] are enabled on the E51 core.

10.2 Interrupt CSRs

When a hart receives an interrupt, the following events are executed:

1. The value of mstatus.MIE field is copied into mstatus.MPIE, then mstatus.MIE is cleared, effectively disabling interrupts.
2. The current value in the program counter (PC) is copied to the mepc register, and then PC is set to the value of mtvec. If vectored interrupts are enabled, PC is set to mtvec.BASE + 4 × exception code.
3. The Privilege mode prior to the interrupt is encoded in mstatus.MPP.
4. At this point, control is handed over to the software in the interrupt handler with interrupts disabled.

Interrupts can be re-enabled by explicitly setting mstatus.MIE, or by executing the `MRET` instruction to exit the handler. When the `MRET` instruction is executed:

1. The Privilege mode is set to the value encoded in mstatus.MPP.
2. The value of mstatus.MPIE is copied to mstatus.MIE.
3. The PC is set to the value of mepc.
4. At this point, control is handed over to software.

The Interrupt CSRs are described in the following sections. This document only describes the implementation of interrupt CSRs specific to the CPU Core Complex. For a complete description of RISC-V interrupt behavior and how to access CSRs, see The RISC-V Instruction Set Manual, Volume II: Privileged Architecture, Version 1.10

10.2.1 Machine STATUS Register

The mstatus register tracks and controls the current operating state of a Hart and tracks whether interrupts are enabled or not. Interrupts are enabled by setting the `MIE` bit and by enabling the required individual interrupt in the mie register described in the next section.

The mstatus register description related to interrupts is provided in the following table. The mstatus register also contains fields unrelated to interrupts. For a complete description of the mstatus register, see The RISC-V Instruction Set Manual, Volume II: Privileged Architecture, Version 1.10.

Table 10-2. Machine Status Register

Bits	Field Name	Attributes	Description
0	Reserved	WPRI	—
1	SIE	RW	Supervisor Interrupt Enable
2	Reserved	WPRI	—
3	MIE	RW	Machine Interrupt Enable
4	Reserved	WPRI	—
5	SPIE	RW	Supervisor Previous Interrupt Enable

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued			
Bits	**Field Name**	**Attributes**	**Description**
6	Reserved	WPRI	—
7	MPIE	RW	Machine Previous Interrupt Enable
8	SPP	RW	Supervisor Previous Privilege Mode
[10:9]	Reserved	WPRI	—
[12:11]	MPP	RW	Machine Previous Privilege Mode

10.2.2 Machine Interrupt Enable Register

Individual interrupts are enabled by setting the appropriate bit in the mie register described in the following table.

Table 10-3. Machine Interrupt Enable Register

Bits	Field Name	Attributes	Description
0	Reserved	WIRI	—
1	SSIE	RW	Supervisor Software Interrupt Enable
2	Reserved	WIRI	—
3	MSIE	RW	Machine Software Interrupt Enable
4	Reserved	WIRI	—
5	STIE	RW	Supervisor Timer Interrupt Enable
6	Reserved	WIRI	—
7	MTIE	RW	Machine Timer Interrupt Enable
8	Reserved	WIRI	—
9	SEIE	RW	Supervisor Global Interrupt Enable
10	Reserved	WIRI	—
11	MEIE	RW	Machine Global Interrupt Enable
[15:12]	Reserved	WIRI	—
16	LIE0	RW	Local Interrupt 0 Enable
17	LIE1	RW	Local Interrupt 1 Enable
18	LIE2	RW	Local Interrupt 2 Enable
...			
63	LIE47	RW	Local Interrupt 47 Enable

10.2.3 Machine Interrupt Pending Register

The machine interrupt pending (mip) register specifies interrupts which are currently pending.

Table 10-4. Machine Interrupt Pending Register

Bits	Field Name	Attributes	Description
0	Reserved	WPRI	—
1	SSIP	RW	Supervisor Software Interrupt Pending
2	Reserved	WPRI	—
3	MSIP	RO	Machine Software Interrupt Pending
4	Reserved	WPRI	—
5	STIP	RW	Supervisor Timer Interrupt Pending

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued			
Bits	**Field Name**	**Attributes**	**Description**
6	Reserved	WPRI	—
7	MTIP	RO	Machine Timer Interrupt Pending
8	Reserved	WPRI	—
9	SEIP	RW	Supervisor Global Interrupt Pending
10	Reserved	WPRI	—
11	MEIP	RO	Machine Global Interrupt Pending
[15:12]	Reserved	WPRI	—
16	LIP0	RO	Local Interrupt 0 Pending
17	LIP1	RO	Local Interrupt 1 Pending
18	LIP2	RO	Local Interrupt 2 Pending
...			
63	LIP47	RO	Local Interrupt 47 Pending

10.2.4 Machine Cause Register

When a trap is taken in the Machine mode, `mcause` is written with a code indicating the event that caused the trap. When the event that caused the trap is an interrupt, the most significant bit (MSb) of `mcause` is set to 1, and the least significant bits (LSb) indicate the interrupt number, using the same encoding as the bit positions in `mip`. For example, a Machine Timer Interrupt causes `mcause` to be set to

`0x8000_0000_0000_0007`. `Mcause` is also used to indicate the cause of synchronous exceptions, in which case the MSb of `mcause` is set to 0. This section provides the `mcause` register description and a list of synchronous Exception codes.

Table 10-5. Machine Cause Register

Bits	Field Name	Attributes	Description
[62:0]	Exception Code	WLRL	A code identifying the last exception. See the following table.
63	Interrupt	WLRL	1 if the trap was caused by an interrupt; 0 otherwise.

Table 10-6. Interrupt Exception Codes

Interrupt	Exception Code	Description
1	0	Reserved
1	1	Supervisor software interrupt
1	2	Reserved
1	3	Machine software interrupt
1	4	Reserved
1	5	Supervisor timer interrupt
1	6	Reserved
1	7	Machine timer interrupt
1	8	Reserved
1	9	Supervisor Global interrupt
1	10	Reserved
1	11	Machine Global interrupt
1	12-15	Reserved

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued		
Interrupt	**Exception Code**	**Description**
1	16	Local Interrupt 0
1	17	Local Interrupt 1
1	18-62	...
1	63	Local Interrupt 47
0	0	Instruction address misaligned
0	1	Instruction access fault
0	2	Illegal Instruction
0	3	Breakpoint
0	4	Load address misaligned
0	5	Load access fault
0	6	Store/AMO address misaligned
0	7	Store/AMO access fault
0	8	Environment call from U-mode
0	9	Environment call from S-mode
0	10	Reserved
0	11	Environment call from M-mode
0	12	Instruction page fault
0	13	Load page fault
0	14	Reserved
0	15	Store/AMO page fault
0	16-31	Reserved

10.2.5 Machine Trap Vector Register

By default, all interrupts trap to a single address defined in the `mtvec` register. The interrupt handler must read `mcause` and handle the trap accordingly. The CPU Core Complex supports interrupt vectoring for defining an interrupt handler for each interrupt defined in `mie`. Interrupt vectoring enables all local interrupts to trap to exclusive interrupt handlers. With vectoring enabled, all global interrupts trap to a single global interrupt vector. Vectored interrupts are enabled when the `MODE` field of the `mtvec` register is set to 1. The following table lists the `mtvec` register description.

Table 10-7. Machine Trap Vector Register

Bits	Field Name	Attributes	Description
[1:0]	MODE	WARL	MODE determines whether or not interrupt vectoring is enabled. The field encoding of `mtvec.MODE` is as follows: 0: (Direct) All exceptions set PC to BASE 1: (Vectored) Asynchronous interrupts set PC to BASE + 4 × cause ≥2: Reserved
[63:2]	BASE[63:2][1]	WARL	Interrupt Vector Base Address. Must be aligned on a 128-byte boundary when MODE=1.

© 2024 Microchip Technology Inc. and its subsidiaries

Note:

8. BASE[1:0] is not present in this register and is implicitly 0.
   If vectored interrupts are disabled (`mtvec.MODE=0`), all interrupts trap to the `mtvec.BASE` address. If vectored interrupts are enabled (`mtvec.MODE=1`), interrupts set the PC to `mtvec.BASE + 4 × exception code`. For example, if a machine timer interrupt is taken, the PC is set to `mtvec.BASE + 0x1C`. The trap vector table is populated with jump instructions to transfer control to interrupt specific trap handlers. In Vectored Interrupt mode, BASE must be 128-byte aligned. All machine Global interrupts are mapped to exception code of 11. Thus, when interrupt vectoring is enabled, the PC is set to address `mtvec.BASE + 0x2C` for any Global interrupt. See the interrupt exception codes in Table 10-5.

10.3 Supervisor Mode Interrupts

For improved performance, the CPU Core Complex includes interrupt and exception delegation CSRs to direct the required interrupts and exceptions to Supervisor mode. This capability is enabled by `mideleg` and `medeleg` CSRs. Supervisor interrupts and exceptions can be managed via supervisor interrupt CSRs `stvec`, `sip`, `sie`, and `scause`. Machine mode software can also directly write to the sip register to pend an interrupt to Supervisor mode. A typical use case is the timer and software interrupts, which may have be to handled in both Machine and Supervisor modes. For more information about RISC-V supervisor interrupts, see The RISC-V Instruction Set Manual, Volume II: Privileged Architecture, Version 1.10.

By setting the corresponding bits in the `mideleg` and `medeleg` CSRs, the Machine mode software can delegate the required interrupts and exceptions to Supervisor mode. Once a delegated trap is asserted, `mcause` is copied into `scause` and `mepc` is copied into `sepc`, and then, the Hart traps to the `stvec` address in Supervisor mode. Local interrupts can not be delegated to Supervisor mode. The register description of the delegation and supervisor CSRs are described in the following sections.

10.3.1 Machine Interrupt Delegation Register (mideleg)

The register description of the `mideleg` register is provided in the following table.

Table 10-8. Machine Interrupt Delegation Register (mideleg)

Bits	Attributes	Description
0	WARL	Reserved
1	WARL	Supervisor software interrupt
[4:2]	WARL	Reserved
5	WARL	Supervisor timer interrupt
[8:6]	WARL	Reserved
9	WARL	Supervisor external interrupt
[63:10]	WARL	Reserved

10.3.2 Machine Exception Delegation Register (medeleg)

The register description of the `medeleg` register is provided in the following table.

Table 10-9. Machine Exception Delegation Register (medeleg)

Bits	Attributes	Description
0	WARL	Instruction address misaligned
1	WARL	Instruction access fault
2	WARL	Illegal Instruction
3	WARL	Breakpoint

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

Bits	Attributes	Description
4	WARL	Load address misaligned
5	WARL	Load access fault
6	WARL	Store/AMO address misaligned
7	WARL	Store/AMO access fault
8	WARL	Environment call from U-mode
9	WARL	Environment call from S-mode
[11:10]	WARL	Reserved
12	WARL	Instruction page fault
13	WARL	Load page fault
14	WARL	Reserved
15	WARL	Store/AMO page fault exception
[63:16]	WARL	Reserved

10.3.3 Supervisor STATUS Register (sstatus)

`sstatus` is a restricted view of `mstatus` described in 10.2.1. Machine STATUS Register (mstatus). Changes made to `sstatus` are reflected in `mstatus` and vice-versa but the Machine mode fields are not visible in `sstatus`. `Sstatus` also contains fields unrelated to interrupts, those fields are not covered in this document. The `sstatus` fields related to interrupts are described in the following table.

Table 10-10. Supervisor STATUS Register (sstatus)

Bits	Field Name	Attributes	Description
0	Reserved	WPRI	—
1	SIE	RW	Supervisor Interrupt Enable
[4:2]	Reserved	WPRI	—
5	SPIE	RW	Supervisor Previous Interrupt Enable
[7:6]	Reserved	WPRI	—
8	SPP	RW	Supervisor Previous Privilege Mode
[12:9]	Reserved	WPRI	—

Supervisor interrupts are enabled by setting the SIE bit in sstatus and by enabling the required individual supervisor interrupt in the sie register described in the following section.

10.3.4 Supervisor Interrupt Enable Register (sie)

The required supervisor interrupt (software, timer, and external interrupt) can be enabled by setting the appropriate bit in the `sie` register described in the following table.

Table 10-11. Supervisor Interrupt Enable Register (sie)

Bits	Field Name	Attributes	Description
0	Reserved	WIRI	—
1	SSIE	RW	Supervisor Software Interrupt Enable
[4:2]	Reserved	WIRI	—
5	STIE	RW	Supervisor Timer Interrupt Enable

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued			
Bits	**Field Name**	**Attributes**	**Description**
[8:6]	Reserved	WIRI	—
9	SEIE	RW	Supervisor External Interrupt Enable
[63:10]	Reserved	WIRI	—

10.3.5 Supervisor Interrupt Pending (sip)

The supervisor interrupt pending (`sip`) register indicates the interrupts that are currently pending.

Table 10-12. Supervisor Interrupt Pending Register (sip)

Bits	Field Name	Attributes	Description
0	Reserved	WPRI	—
1	SSIP	RW	Supervisor Software Interrupt Pending
[4:2]	Reserved	WPRI	—
5	STIP	RW	Supervisor Timer Interrupt Pending
[8:6]	Reserved	WPRI	—
9	SEIP	RW	Supervisor External Interrupt Pending
[63:10]	Reserved	WPRI	—

10.3.6 Supervisor Cause Register (scause)

When a trap is received in Supervisor mode, `scause` is written with a code indicating the event that caused the trap. When the event is an interrupt, the most significant bit (MSb) of scause is set to 1, and the least significant bits (LSb) indicate the interrupt number, using the same encoding as the bit positions in sip. For example, a Supervisor Timer interrupt causes `scause` to be set to 0x8000_0000_0000_0005. `Scause` is also used to indicate the cause of synchronous exceptions, if the MSb of `scause` is set to 0.

Table 10-13. Supervisor Cause Register (scause)

Bits	Field Name	Attributes	Description
[62:0]	Exception Code	WLRL	A code identifying the last exception. Supervisor Interrupt Exception codes are listed in 10.2.3. Machine Interrupt Pending Register .
63	Interrupt	WARL	1 if the trap was caused by an interrupt; 0 otherwise.

Table 10-14. Supervisor Interrupt Exception Codes

Interrupt	Exception Code	Description
1	0	Reserved
1	1	Supervisor software interrupt
1	2-4	Reserved
1	5	Supervisor timer interrupt
1	6-8	Reserved
1	9	Supervisor external interrupt
1	≥10	Reserved

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

Interrupt	Exception Code	Description
0	0	Instruction address misaligned
0	1	Instruction access fault
0	2	Illegal instruction
0	3	Breakpoint
0	4	Reserved
0	5	Load access fault
0	6	Store/AMO address misaligned
0	7	Store/AMO access fault
0	8	Environment call from U-mode
0	9-11	Reserved
0	12	Instruction page fault
0	13	Load page fault
0	14	Reserved
0	15	Store/AMO page fault
0	≥16	Reserved

10.3.7 Supervisor Trap Vector (stvec)

By default, all interrupts defined in `sie` trap to a single address defined in the `stvec` register. The interrupt handler must read `scause` and handle the interrupt accordingly. The CPU Core Complex supports interrupt vectors, which enables each interrupt to trap to its own specific interrupt handler. Vectored interrupts can be enabled by setting the `stvec.MODE` field to 1.

Table 10-15. Supervisor Trap Vector Register (stvec)

Bits	Field Name	Attributes	Description
[1:0]	MODE	WARL	MODE determines whether or not interrupt vectoring is enabled. The field encoding of `stvec.MODE` is as follows: 0: (Direct) All exceptions set PC to BASE 1: (Vectored) Asynchronous interrupts set PC to: BASE + 4 × cause. ≥2: Reserved
[63:2]	BASE[63:2]	WARL	Interrupt Vector Base Address. Must be aligned on a 128-byte boundary when MODE=1. **Note:** BASE [1:0] is not present in this register and is implicitly 0.

If vectored interrupts are disabled (`stvec.MODE=0`), all interrupts trap to the `stvec.BASE` address. If vectored interrupts are enabled (`stvec.MODE=1`), interrupts set the PC to `stvec.BASE` + 4 × exception code. For example, if a supervisor timer interrupt is taken, the PC is set to `stvec.BASE` + 0x14. Typically, the trap vector table is populated with jump instructions to transfer control to interrupt-specific trap handlers. In Vectored Interrupt mode, BASE must be 128-byte aligned.

All supervisor Global interrupts are mapped to exception code of 9. Thus, when interrupt vectoring is enabled, the PC is set to address `stvec.BASE` + 0x24 for any global interrupt. See the supervisor interrupt exception codes in Table 10-13 .

10.4 Interrupt Priorities

Local interrupts have higher priority than Global interrupts. If a Local and Global interrupt arrive in the same cycle, the Local interrupt is handled if enabled. Priorities of Local interrupts are determined by the Local interrupt ID, Local Interrupt 47 being the highest priority. For example, if Local Interrupt 47 and 6 arrive in the same cycle, Local Interrupt 47 is handled.

Exception code of the Local Interrupt 47 is also the highest and occupies the last slot in the interrupt vector table. This unique position in the vector table allows the interrupt handler of the Local

© 2024 Microchip Technology Inc. and its subsidiaries

Interrupt 47 to be placed in-line instead of a jump instruction. The jump instruction is required for other interrupts when operating in Vectored mode. Hence, Local Interrupt 47 must be used for the most critical interrupt in the system.

CPU Core Complex interrupts are prioritized in the following decreasing order of priority:

- Local Interrupt 47 to 0
- Machine Global interrupts
- Machine software interrupts
- Machine timer interrupts
- Supervisor Global interrupts
- Supervisor software interrupts
- Supervisor timer interrupts

Individual priorities of Global interrupts are determined by the PLIC, see 10.6. Platform Level Interrupt Controller.

10.5 Interrupt Latency

Interrupt latency is four cycles and depends on the number of cycles it takes from the signaling of the interrupt to the first instruction fetch of the handler. Global interrupts routed through the PLIC incur an additional latency of three cycles, where the PLIC is clocked by the user_clock. If the interrupt handler is cached or located in ITIM, the total latency (cycles) of a Global interrupt is

```
4 + 3 × [(core clock (Hz)/user_clock (Hz)].
```

Additional latency from a peripheral source is not included. Moreover, the Hart does not ignore an arithmetic instruction like “Divide” that is in the execution pipeline. Hence, if an interrupt handler tries to use a register which is the destination register of a divide instruction, the pipeline stalls until the completion of the divide instruction.

10.6 Platform Level Interrupt Controller

The PLIC supports 186 Global interrupts with 7 priority levels and complies with The RISC-V Instruction Set Manual, Volume II: Privileged Architecture, Version 1.10.

10.6.1 PLIC Memory Map

Table 10-16. PLIC Memory Map

Address	Width	Attributes	Description	Notes
0x0C00_0000 0x0C00_0004 0x0C00_0008 … 0x0C00_02D0	4B 4B 4B	RW RW RW	Reserved source 1 priority source 2 priority … source 186 priority	See Table 10-18.
0xC00_02D4 … 0x0C00_0FFF	—	—	Reserved	—
0xC00_1000 … 0x0C00_1014	4B … 4B	RO … RO	Start of pending array … Last word of pending array	Table 10-19
0x0C00_1018 … 0x0C00_1FFF	—	—	Reserved	—

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

Address	Width	Attributes	Description	Notes
0x0C00_2000 ... 0x0C00_2014	4B ... 4B	RW ... RW	Start of Hart 0 M-mode enables ... End of Hart 0 M-mode enables	Table 10-21
0x0C00_2018 ... 0x0C00_207F	—	—	Reserved	—
0x0C00_2080 ... 0x0C00_2094	4B ... 4B	RW ... RW	Hart 1 M-mode enables ... End of Hart 1 M-mode enables	
0x0C00_2100 ... 0x0C00_2114	4B ... 4B	RW ... RW	Hart 1 S-mode enables ... End of Hart 1 S-mode enables	Same layout as Hart 0 M-mode enables
0x0C00_2180 ... 0x0C00_2194	4B ... 4B	RW ... RW	Hart 2 M-mode enables ... End of Hart 2 M-mode enables	
0x0C00_2200 ... 0x0C00_2214	4B ... 4B	RW ... RW	Hart 2 S-mode enables ... End of Hart 2 S-mode enables	
0x0C00_2280 ... 0x0C00_2294	4B ... 4B	RW ... RW	Hart 3 M-mode enables ... End of Hart 3 M-mode enables	
0x0C00_2300 ... 0x0C00_2314	4B ... 4B	RW ... RW	Hart 3 S-mode enables ... End of Hart 3 S-mode enables	
0x0C00_2380 ... 0x0C00_2394	4B ... 4B	RW ... RW	Hart 4 M-mode enables ... End of Hart 4 M-mode enables	
0x0C00_2400 ... 0x0C00_2414	4B ... 4B	RW ... RW	Hart 4 S-mode enables ... End of Hart 4 S-mode enables	
0x0C00_2480 ... 0x0C1F_FFFF	—	—	Reserved	
0x0C20_0000 0x0C20_0004	4B 4B	RW RW	Hart 0 M-mode priority threshold Hart 0 M-mode claim/complete	
0x0C20_1000 0x0C20_1004	4B 4B	RW RW	Hart 1 M-mode priority threshold Hart 1 M-mode claim/complete	See Table 10-23 and Table 10-24.
0x0C20_2000 0x0C20_2004	4B 4B	RW RW	Hart 1 S-mode priority threshold Hart 1 S-mode claim/complete	
0x0C20_3000 0x0C20_3004	4B 4B	RW RW	Hart 2 M-mode priority threshold Hart 2 M-mode claim/complete	
0x0C20_4000 0x0C20_4004	4B 4B	RW RW	Hart 2 S-mode priority threshold Hart 2 S-mode claim/complete	

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued				
Address	Width	Attributes	Description	Notes
0x0C20_5000 0x0C20_5004	4B 4B	RW RW	Hart 3 M-mode priority threshold Hart 3 M-mode claim/complete	
0x0C20_6000 0x0C20_6004	4B 4B	RW RW	Hart 3 S-mode priority threshold Hart 3 S-mode claim/complete	—
0x0C20_7000 0x0C20_7004	4B 4B	RW RW	Hart 4 M-mode priority threshold Hart 4 M-mode claim/complete	
0x0C20_8000 0x0C20_8004	4B 4B	RW RW	Hart 4 S-mode priority threshold Hart 4 S-mode claim/complete	

10.6.2 Interrupt Sources

The CPU Core Complex exposes 186 Global interrupt signals, these signals are connected to the PLIC. The mapping of these interrupt signals to their corresponding PLIC ID's is provided in the following table.

Table 10-17. PLIC Interrupt ID Mapping

IRQ	Peripheral	Description
1	L2 Cache Controller	Signals when a metadata correction event occurs
2	L2 Cache Controller	Signals when an uncorrectable metadata event occurs
3	L2 Cache Controller	Signals when a data correction event occurs
4	L2 Cache Controller	Signals when an uncorrectable data event occurs
5	DMA Controller	Channel 0 Done
6	DMA Controller	Channel 0 Error
7	DMA Controller	Channel 1 Done
8	DMA Controller	Channel 1 Error
9	DMA Controller	Channel 2 Done
10	DMA Controller	Channel 2 Error
11	DMA Controller	Channel 3 Done
12	DMA Controller	Channel 3 Error
[181:13]	Off Core Complex	Connected to global_interrupts signal from MPU peripherals
182 183 184 185 186	Bus Error Unit Hart0 Bus Error Unit Hart1 Bus Error Unit Hart2 Bus Error Unit Hart3 Bus Error Unit Hart4	Bus Error Unit described in 5.11. Bus Error Unit.

The Global interrupt signals are positive-level triggered. Any unused Global interrupts (inputs) must be tied to logic 0. In the PLIC, Global Interrupt ID 0 means "no interrupt", therefore, Global interrupts[0] corresponds to PLIC Interrupt ID 1.

10.6.3 Interrupt Priorities Register

Each PLIC interrupt source can be assigned a priority by writing to its 32-bit memory-mapped priority register. A priority value of 0 is reserved to mean "never interrupt" and effectively disables the interrupt. Priority 1 is the lowest active priority, and priority 7 is the highest. Ties between global interrupts of the same priority are broken by the Interrupt ID; interrupts with the lowest ID have the highest effective priority. The priority register description is provided in the following table.

© 2024 Microchip Technology Inc. and its subsidiaries

Table 10-18. PLIC Interrupt Priority Register

Base Address = 0x0C00_0000 + 4 × Interrupt ID				
Bits	Field Name	Attributes	Reset	Description
[2:0]	Priority	WARL	X	Sets the priority for a given global interrupt.
[31:3]	Reserved	WIRI	X	—

10.6.4 Interrupt Pending Bits

The current status of the interrupt source can be read from the pending bits in the PLIC. The pending bits are organized as 6 words of 32 bits, see the following table for the register 1 description. The pending bit for interrupt ID N is stored in bit (N mod 32) of word (N=32). The PLIC includes 6 interrupt pending registers, see the following table for the first register description and Table 10-20for the sixth register. Bit 0 of word 0, which represents the non-existent interrupt source 0, is hardwired to zero.

A pending bit in the PLIC can be cleared by setting the associated enable bit, then performing a claim as described in 10.6.7. Interrupt Claim Process.

Table 10-19. PLIC Interrupt Pending Register 1

Base Address = 0x0C00_1000				
PLIC Interrupt Pending Register 1 (pending 1)				
Bits	Field Name	Attributes	Reset	Description
0	Interrupt 0 pending	RO	0	Non-existent Global interrupt 0 is hardwired to zero
1	Interrupt 1 pending	RO	0	Pending bit for Global interrupt 1
2	Interrupt 2 pending	RO	0	Pending bit for Global interrupt 2
...				
31	Interrupt 31 pending	RO	0	Pending bit for Global interrupt 31

Table 10-20. PLIC Interrupt Pending Register 6

PLIC Interrupt Pending Register 6 (pending 6)				
Base Address = 0x0C00_1014				
Bits	Field Name	Attributes	Reset	Description
0	Interrupt 160 Pending	RO	0	Pending bit for Global interrupt 160
...				
25	Interrupt 186 Pending	RO	0	Pending bit for Global interrupt 186
[31:26]	Reserved	WIRI	X	—

10.6.5 Interrupt Enables

Each Global interrupt can be enabled by setting a bit in an Enable register. There are six Enable registers organized as a contiguous array of 32 bits (6 words). Bit 0 of enable word 0 represents the non-existent interrupt ID 0 and is hardwired to 0. 64-bit and 32-bit word accesses are supported in the RV64 systems.

Table 10-21. PLIC Interrupt Enable Register 1

PLIC Interrupt Enable Register 1 (enable 1)				
Base Address = 0x0C00_2000				
Bits	Field Name	Attributes	Reset	Description
0	Interrupt 0 Enable	RW	X	Non-existent Global interrupt 0 is hardwired to zero.
1	Interrupt 1 Enable	RW	X	Enable bit for Global interrupt 1
2	Interrupt 2 Enable	RW	X	Enable bit for Global interrupt 2

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued				
PLIC Interrupt Enable Register 1 (enable 1)				
Base Address = 0x0C00_2000				
Bits	Field Name	Attributes	Reset	Description
...				
31	Interrupt 31 Enable	RW	X	Enable bit for Global interrupt 31

Table 10-22. PLIC Interrupt Enable Register 6

PLIC Interrupt Enable Register 6 (enable 6)				
Base Address = 0x0C00_201C				
Bits	Field Name	Attributes	Reset	Description
0	Interrupt 160 Enable	RW	X	Enable bit for Global interrupt 160
...				
25	Interrupt 186 Enable	RW	X	Enable bit for Global interrupt 186
[31:26]	Reserved	WIRI	X	—

10.6.6 Priority Thresholds

An interrupt priority threshold can be set using the Threshold register. The Threshold register is a WARL field and a maximum threshold of 7 is supported. The processor core masks the PLIC interrupts that have a priority less than or equal to threshold. For example, a threshold value of zero permits all interrupts with non-zero priority, whereas a value of 7 masks all interrupts.

Table 10-23. PLIC Interrupt Priority Threshold Register

PLIC Interrupt Priority Threshold Register				
Base Address = 0x0C20_0000				
Bits	Field Name	Attributes	Reset	Description
[2:0]	Threshold	RW	X	Sets the priority threshold.
[31:3]	Reserved	WIRI	X	—

10.6.7 Interrupt Claim Process

Processor cores can claim an interrupt by reading the PLIC's Claim/Complete register (described in 10.6.8. Interrupt Completion), which returns the ID of the highest-priority pending interrupt or zero if there is no pending interrupt. A successful claim will also atomically clear the corresponding pending bit on the interrupt source. Processor cores can perform a claim at any time, even if the MEIP bit in the mip register is not set. The claim operation is not affected by the setting of the priority threshold register.

10.6.8 Interrupt Completion

To signal the completion of executing an interrupt handler, the processor core writes the received interrupt ID to the Claim/Complete register. The PLIC does not check whether the completion ID is the same as the last claim ID for that target. If the completion ID does not match an interrupt source that is currently enabled for the target, the completion is ignored.

Table 10-24. PLIC Interrupt Claim or Complete Register

Base Address = 0x0C20_0004				
Bits	Field Name	Attributes	Reset	Description
[31:0]	Interrupt Claim	RW	X	A read of zero indicates that no interrupts are pending. A non-zero read contains the ID of the highest pending interrupt. A write to this register signals completion of the interrupt ID written.

© 2024 Microchip Technology Inc. and its subsidiaries

10.7 Core Local Interrupt Controller

The CLINT includes memory-mapped CSRs for enabling software and timer interrupts. The CLINT register map is provided in the following table.

Table 10-25. CLINT Register Map

Address	Width	Attributes	Description	Notes
0x0200_0000 0x0200_0004 0x0200_0008 0x0200_000C 0x0200_0010	4B 4B 4B 4B 4B	RW RW RW RW RW	msip for Hart0 msip for Hart1 msip for Hart2 msip for Hart3 msip for Hart4	MSIP registers
0x0200_0014 ... 0x0200_3FFF	—	—	Reserved	—
0x0200_4000 0x0200_4008 0x0200_4010 0x0200_4018 0x0200_4020	8B 8B 8B 8B 8B	RW RW RW RW RW	mtimecmp for Hart0 mtimecmp for Hart1 mtimecmp for Hart2 mtimecmp for Hart3 mtimecmp for Hart 4	Timer compare register
0x0200_4028 ... 0x0200_BFF7	—	—	Reserved	—
0x0200_BFF8	8B	RW	mtime	Timer register
0x0200_C000 ... 0x0200_FFFF	—	—	Reserved	—

10.7.1 MSIP Register (msip)

Machine mode software interrupts per Hart are enabled by writing to the control register msip. Each msip register is a 32-bit long WARL register. The LSb of msip is reflected in the msip bit of the mip register. Other bits in each msip register are hardwired to zero. At Reset, msip registers are cleared to zero. Software interrupts allow inter-processor core communication in multi-Hart systems by enabling Harts to write to each other's msip bits.

10.7.2 Timer Register (mtime)

mtime is a 64-bit read-write register that counts the number of cycles of the rtc_toggle signal. A timer interrupt is pending whenever mtime is greater than or equal to the value in the mtimecmp register. The timer interrupt reflects in the mtip bit of the mip register described in 10.2.3. Machine Interrupt Pending Register (mip). At Reset, mtime is cleared to zero, the mtimecmp registers are not reset.

The mtime register increments every 1 µs (1 MHz), which is the input frequency of the RTC clock.

10.7.3 Supervisor Mode Delegation

By default, all interrupts trap to Machine mode including timer and software interrupts. Machine mode software and timer interrupts must be delegated to Supervisor mode. For more information, see 10.3. Supervisor Mode Interrupts.

10.8 Resets

The MPU can be reset by any of the following sources:

© 2024 Microchip Technology Inc. and its subsidiaries

- Power Cycle
- System Controller
- CPU Debugger
- E51 Watchdog

The following table lists all the Reset signals of the MPU.

Table 10-26. Reset Signals

Reason	Reset Reason Bit	Asserted By	Description
SCB_PERIPH_RESET	0	SCB	This is the POR signal. This signal fully resets the MPU. Additional bits in the SOFT-RESET register also allow the SCB registers to be reset.
SCB_MPU_RESET	1	SCB, CPU, MPU	This signal resets the full MPU including the CPU Core Complex, peripherals, and the entire AXI system. This signal does not reset SCB registers.
SCB_CPU_RESET	2	SCB, CPU, MPU	This signal resets only the CPU Core Complex. This Reset signal must be used carefully because in most cases the MPU requires resetting at the same time to clear outstanding AXI transactions.
DEBUGER_RESET	3	Debugger	This signal is asserted by the CPU Core Complex debugger and has the same effect as the SCB_MPU_RESET.
WDOG_RESET	5	Watchdog	This signal indicates that the watchdog (WDOG0) Reset has activated.
GPIO_RESET	6	MPU	This indicates that the MPU GPIO Reset was asserted, it will reset the GPIO blocks if the GPIOs are configured to be reset by this signal, it does not reset the MPU.
SCB_BUS_RESET	7	MPU	Indicates that SCB bus Reset occurred.
CPU_SOFT_RESET	8	MPU	Indicates CPU Core Complex Reset was asserted using the soft reset register.

There is an additional register SOFT_RESET_CR, which is used to Reset all MPU peripherals after the MPU Reset. The SOFT_RESET_CR register is described in 11. Memory Map. To view the register description of SOFT_RESET_CR, follow these steps:

1. Download and unzip the register map folder.
2. Using a browser, open the `pfsoc_regmap.htm` file from `<$download_folder>\Register Map\PF_SoC_RegMap_Vx_x`.
3. Select `PFSOC_MPU_TOP_SYSREG` and find the SOFT_RESET_CR register to view its description.

© 2024 Microchip Technology Inc. and its subsidiaries

11. Memory Map

The overall PIC64GX memory map consists of the following address spaces:

- CPU Core Complex address space
- Peripherals address space
- Memory address space

The address range 0x2000_0000 – 0x27FF_FFFF includes the default base addresses (LOW) of low-speed peripherals and base addresses of high-speed peripherals. The address range 0x2800_0000 – 0x2812_6FFF, includes the alternate base addresses (HIGH) of low-speed peripherals.

Table 11-1. Memory Map

	Description	Start Address	End Address	Attributes
CPU Core Complex	Debug	0x0000_0100	0x0000_0FFF	RWX
	E51 DTIM	0x0100_0000	0x0100_1FFF	RWXA
	Bus Error Unit 0	0x0170_0000	0x0170_0FFF	RW
	Bus Error Unit 1	0x0170_1000	0x0170_1FFF	RW
	Bus Error Unit 2	0x0170_2000	0x0170_2FFF	RW
	Bus Error Unit 3	0x0170_3000	0x0170_3FFF	RW
	Bus Error Unit 4	0x0170_4000	0x0170_4FFF	RW
	E51 Hart 0 ITIM	0x0180_0000	0x0180_1FFF	RWXA
	U54 Hart 1 ITIM	0x0180_8000	0X0180_EFFF	RWXA
	U54 Hart 2 ITIM	0x0181_0000	0x0181_6FFF	RWXA
	U54 Hart 3 ITIM	0x0181_8000	0X0181_EFFF	RWXA
	U54 Hart 4 ITIM	0x0182_0000	0x0182_6FFF	RWXA
	CLINT	0x0200_0000	0x0200_FFFF	RW
	Cache Controller	0x0201_0000	0x0201_0FFF	RW
	WCB	0x0202_0000	0x0202_0FFF	RW
	DMA Controller	0x0300_0000	0x030F_FFFF	RW
	L2-LIM	0x0800_0000	0x081F_FFFF	RWX
	L2 Zero Device	0x0A00_0000	0x0BFF_FFFF	RWXC
	PLIC	0x0C00_0000	0x0FFF_FFFF	RW

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

	Description	Start Address	End Address	Attributes
AMP Context A Peripherals	MMUART0	0x2000_0000	0x2000_0FFF	RW
	Watchdog0	0x2000_1000	0x2000_1FFF	RW
	MMUART1	0x2010_0000	0x2010_0FFF	RW
	Watchdog1	0x2010_1000	0x2010_1FFF	RW
	MMUART2	0x2010_2000	0x2010_2FFF	RW
	Watchdog2	0x2010_3000	0x2010_3FFF	RW
	MMUART3	0x2010_4000	0x2010_4FFF	RW
	Watchdog3	0x2010_5000	0x2010_5FFF	RW
	MMUART4	0x2010_6000	0x2010_6 FF	RW
	Watchdog4	0x2010_7000	0x2010_7FFF	RW
	SPI0	0x2010_8000	0x2010_8FFF	RW
	SPI1	0x2010_9000	0x2010_9 FF	RW
	I2C0	0x2010_A000	0x2010_AFFF	RW
	I2C1	0x2010_B000	0x2010_BFFF	RW
	CAN0	0x2010_C000	0x2010_CFFF	RW
	CAN1	0x2010_D000	0x2010_DFFF	RW
	MAC0	0x2011_0000	0x2011_0FFF	RW
	MAC1	0x2011_2000	0x2011_2FFF	RW
	GPIO0	0x2012_0000	0x2012_0FFF	RW
	GPIO1	0x2012_1000	0x2012_1FFF	RW
	GPIO2	0x2012_2000	0x2012_2FFF	RW
	RTC	0x2012_4000	0x2012_4FFF	RW
	Timer	0x2012_5000	0x2012_5FFF	RW
Peripherals	User Crypto	0x2200_0000	0x2201FFFF	RW
	eNVM - Config	0x2020_000	0x2020_0FFF	RW
	USB – Config	0x2020_1000	0x2020_1FFF	RW
	eNVM - Data	0x2022_0000	0x2023_FFFF	RW
	QSPI	0x2100_0000	0x21FF_FFFF	RW
	Athena	0x2200_0000	0x2201FFFF	RW

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

	Description	Start Address	End Address	Attributes
AMP Context B Peripherals	MMUART0	0x2800_0000	0x2800_0FFF	RW
	Watchdog0	0x2800_1000	0x2800_1FFF	RW
	MMUART1	0x2810_0000	0x2810_0FFF	RW
	Watchdog1	0x2810_1000	0x2810_1FFF	RW
	MMUART2	0x2810_2000	0x2810_2FFF	RW
	Watchdog2	0x2810_3000	0x2810_3FFF	RW
	MMUART3	0x2810_4000	0x2810_4FFF	RW
	Watchdog3	0x2810_5000	0x2810_5FFF	RW
	MMUART4	0x2810_6000	0x2810_6FFF	RW
	Watchdog4	0x2810_7000	0x2810_7FFF	RW
	SPI0	0x2810_8000	0x2810_8FFF	RW
	SPI1	0x2810_9000	0x2810_9FFF	RW
	I2C0	0x2810_A000	0x2810_AFFF	RW
	I2C1	0x2810_B000	0x2810_BFFF	RW
	CAN0	0x2810_C000	0x2810_CFFF	RW
	CAN1	0x2810_D000	0x2810_DFFF	RW
	MAC0	0x2811_0000	0x2811_0FFF	RW
	MAC1	0x2811_2000	0x2811_2FFF	RW
	GPIO0	0x2812_0000	0x2812_0FFF	RW
	GPIO1	0x2812_1000	0x2812_1FFF	RW
	GPIO2	0x2812_2000	0x2812_2FFF	RW
	RTC	0x2812_4FFF	0x2812_4FFF	RW
	Timer	0x2812_5FFF	0x2812_5FFF	RW
Peripherals	IOSCB-DATA	0x3000_0000	0x3FFF_FFFF	RWX
	IOSCB-CONFIGURATION	0x3708_0000	0x3708_0FFF	RWX
	Pin Multiplexer	0x4100_0000	0x4100_00FF	RW
	IHC	0x4800_0000	0x48FF_FFFF	RW
	Video Pipeline Control	0x4xxx_xxxx		RW
Video Pipeline 32-bit addressing	512 MB CPU Core Complex – F0 (AXI Switch Master Port M12)	0x6000_0000	0x7FFF_FFFF	RWX
DDR Memory 32-bit addressing	DDR Cached Access 1GB	0x8000_0000	0xBFFF_FFFF	RWXC
	DDR Non-Cached Access 256 MB	0xC000_0000	0xCFFF_FFFF	RWX
	DDR Non-Cached WCB Access 256 MB	0xD000_0000	0xDFFF_FFFF	RWX
Video Pipeline 32-bit addressing	512 MB CPU Core Complex – F1 (AXI Switch Master Port M13)	0xE000_0000	0xFFFF_FFFF	RWX

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

	Description	Start Address	End Address	Attributes
DDR Memory 39-bit addressing	DDR Cached Access 16 GB	0x10_0000_0000	0x13_FFFF_FFFF	RWXC
	DDR Non-Cached Access 16 GB	0x14_0000_0000	0x17_FFFF_FFFF	RWX
	DDR Non-Cached WCB Access 16 GB	0x18_0000_0000	0x1B_FFFF_FFFF	RWX
Video Pipeline 39-bit addressing	64 GB	0x20_0000_0000	0x2F_FFFF_FFFF	RWX
	64 GB	0x30_0000_0000	0x3F_FFFF_FFFF	RWX

Note: Memory Attributes: R – Read, W- Write, X – Execute, C – Cacheable, A – Atomics.

© 2024 Microchip Technology Inc. and its subsidiaries

12. Clocking

An off-chip 100 MHz or 125 MHz reference clock can be fed into the following PLLs:

- MPU PLL: Generates up to 625 MHz CPU clock, 80 MHz Standby.
- DDR PLL: Generates 400 MHz, actual frequency depends on the DDR type.
- SGMII PLL: Generates clocks for SGMII PHY and GEMs.
  Note: These PLLs are located at NW corner of the device close to the MPU. The MPU PLL supports up to 625 MHz on “STD” devices.

  These PLLs are used to generate the main clocks for the various blocks in the MPU. The majority of the MPU is clocked from a 600 MHz or less (CPU)/300 MHz or less (AMBA subsystem) clock derived from the MPU PLL via a clock divider.

  The CPU cores, L2 Cache, and AMBA infrastructure are clocked from the MPU PLL through a set off dividers. During normal operation, the PLL clock is divided by 1 for the CPU cores, by 2 for the L2 Cache and AXI bus, and by 4 for the AHB/APB bus.

  At power-up and after MPU Reset, the MPU is clocked from the on-chip 80 MHz RC oscillator. This clock source can be switched to the MPU clock source dynamically during boot-up using the embedded firmware running on E51. There is no switching of clock sources at device power up. The MPU PLL remains operational.

  The SGMII PLL generates the necessary clocks required for the SGMII PHY block.

  The DDR PLL generates the necessary clocks required for the DDR PHY and for the DFI interface to the DDR controller in the MPU.

© 2024 Microchip Technology Inc. and its subsidiaries

13. System Interconnect

13.1 Branch Prediction

See 5.7. Branch Prediction.

13.2 AXI Switch

The AXI Switch provides 15 master ports, 9 slave ports, and QoS. Two slave ports—S5 and S6—are connected to AXI-to-AHB and AHB-to-APB bridges to interface with peripherals. The master and slave ports on the AXI Switch are listed in the following table.

Table 13-1. Master and Slave Ports

Master Port	Master Inputs	Slave Port	Slave Outputs
M4	Crypto Processor	S4	Crypto Processor
M5	GEM0	S5	AHB0
M6	GEM1	S6	AHB1
M7	USB	S7	DDR-Non Cacheable (NC)
M8	eMMC	S8	CPU Core Complex – MMIO
M9	SCB Bridge	S9	Trace Module
M10	CPU Core Complex – D0	—	—
M11	CPU Core Complex – D1	—	—
M12	CPU Core Complex – F0	—	—
M13	CPU Core Complex – F1	—	—
M14	CPU Core Complex – NC	—	—
M15	Trace Module	—	—

The following table lists the master and slave connectivity on the AXI switch.

Table 13-2. Master and Slave Connectivity

	S1	S2	S3	S4	S5	S6	S7	S8	S9
M1	✓	✓	✓	✓	✓	✓	✓	✓	—
M2	✓	✓	✓	✓	✓	✓	✓	✓	—
M3	—	—	—	—	—	—	✓	✓	—
M4	✓	✓	✓	—	✓	—	✓	✓	—
M5	✓	✓	—	✓	—	—	✓	✓	—
M6	✓	✓	—	✓	—	—	✓	✓	—
M7	✓	✓	—	✓	—	—	✓	✓	—
M8	✓	✓	—	✓	—	—	✓	✓	—
M9	✓	✓	✓	✓	✓	✓	✓	✓	✓
M10	—	—	✓	✓	✓	—	—	—	✓
M11	—	—	—	—	—	✓	—	—	—
M12	✓	—	—	—	—	—	—	—	—
M13	—	✓	—	—	—	—	—	—	—
M14	—	—	—	—	—	—	✓	—	—
M15	✓	✓	—	—	—	—	✓	—	—

© 2024 Microchip Technology Inc. and its subsidiaries

The following table lists address ranges of each slave port on the AXI switch.

Table 13-3. Address Ranges of Slaves

Slave Port	Number of Regions	Region 1		Region 2		Region 3	
		Start	End	Start	End	Start	End
S1	2	0x20_0000_0000	0x2F_FFFF_FFFF	0x6000_0000	0x7FFF_FFFF	—	—
S2	2	0x30_0000_0000	0x3F_FFFF_FFFF	0xE000_0000	0xFFFF_FFFF	—	—
S3	1	0x4000_0000	0x5FFF_FFFF	—	—	—	—
S4	1	0x2200_0000	0x2201_FFFF	—	—	—	—
S5	2	0x2000_0000	0x21FF_FFFF	0x3000_0000	0x3FFF_FFFF	—	—
S6	1	0x2800_0000	0x2FFF_FFFF	—	—	—	—
S7	2	0x14_0000_0000	0x1B_FFFF_FFFF	0xC000_0000	0xDFFF_FFFF	—	—
S8	3	0x10_0000_0000	0x13_FFFF_FFFF	0x8000_0000	0xBFFF_FFFF	0x0000_0000	0x1FFF_FFFF
S9	1	0x2300_0000	0x2303_FFFF	—	—	—	—

13.3 AXI Switch Arbitration

The AXI Switch arbitration is configured as listed in the following tables.

Table 13-4. Arbitration of Slave Ports

Slave Port	Slave	Arbitration Type Write and Read Address	Arbitration Type Write Data	Description
0	Default Slave	First come first served	First come first served	This is the default slave that responds to illegal addresses
1-8	Others	QoS Arbiter	Fair Among equals	—
9	Trace Slave	Priority	Priority	—

Table 13-5. Arbitration of Master Ports

Master Port	Master	Arbitration Type Write and Read Address	Arbitration Type Write Data	Description
1-8	Others	Fair among equals	Fair among equals	—
9	SCB/System Controller Master	Priority	Priority	System Controller comes with the highest priority
10-14	Others	Fair among equals	Fair among equals	—
15	Trace Master	Priority	Priority	Trace comes with the highest priority

The following rules are applicable:

- Priority: Highest priority and the lowest number client with the same priority wins.
- First Come First Serve: Clients are granted in the order they request, longer waiting client has the highest priority.
- Fair Among Equals: Two-tier arbitration. First tier is dynamic priority, second tier shares equally between clients of the same highest requesting priority on a cycle-by- cycle basis.

13.4 Quality of Service

The AXI switch uses a QoS scheme to control priorities in the switch and the DDR controller. The QoS is a 4-bit value and a higher QoS value represents a higher priority.

© 2024 Microchip Technology Inc. and its subsidiaries

Each master generates a QoS as listed in the following table:

Table 13-6. AXI Switch QoS

Port	Master	Master Directly supports QoS (Yes/No)	QoS Source
4	Crypto Processor	No	Set by System register, the value is fixed once set
5	GEM0	Yes	Set by Ethernet MAC
6	GEM1	Yes	Set by Ethernet MAC
7	USB	No	Set by System register, the value is fixed once set
8	MMC	No	Set by System register, the value is fixed once set
9	SCB Bridge	Yes	Set in System Controller SCB interface
10	CPLEX-D0	No	Set by System register, the value is fixed once set
11	CPLEX-D1	No	Set by System register, the value is fixed once set
12	CPLEX-F0	No	Set by System register, the value is fixed once set
13	CPLEX-F1	No	Set by System register, the value is fixed once set
14	CPLEX-NC	No	Set by System register, the value is fixed once set
15	TRACE	Yes	Set by Trace SMB

13.5 TileLink

TileLink is a chip-scale interconnect which provides multiple masters with coherent access to memory and other slave peripherals for low-latency and high throughput transfers. For more information, see TileLink Specification v1.7.1.

13.6 External Bus Interfaces

See 5.8. External Bus Interfaces.

13.7 AXI-to-AHB

The MPU supports AHB peripherals (QSPI, USB, eNVM, IOSCB) via Slave slot 5 (S5) of the AXI Switch, S5 is converted to AHB-Lite. S6 is also converted to AHB-Lite. These AHB buses are connected to a 5:1 AHB multiplexer to allow connection to the five AHB slaves in the system. The AHB clock is synchronous to the AXI clock, but the AHB clock is /2, /4, or /8 of the AXI clock. The MPU supports APB peripherals (CAN, MMUART, SPI, and I2C) and APB slaves.

13.8 AHB-to-APB

The MPU supports APB peripherals (CAN, MMUART, SPI, I2C) and configuration interfaces to other blocks (DDRC, AXI-SWITCH, ETHERNET) via the AHB-Lite bus generated from S5 of the AXI Switch, S5 is converted to APB using an AHB-to-APB bridge. The APB clock is synchronous (identical) to the AHB clock, and consequently to the AXI clock.

By default, the AHB-to-APB bridge operates in the non-posted Write mode, so the APB write cycle must complete (PREADY asserted) before HREADY is generated. In this mode, a slow responding APB device stalls the AHB bus and therefore, the AXI buses.

The System register bit SR_AHBAPB_CR.APBx_POSTED can be used to switch the AHB-to-APB bridge to the posted Write mode. In this mode, the AHB-to-APB bridge asserts the HREADY before the APB write cycle completes. This allows the CPU to move on before the slow peripheral completes the write. In the posted Write mode, CPU may start the next operation before the actual write completes leading to unexpected results.

© 2024 Microchip Technology Inc. and its subsidiaries

14. Package and Pinout

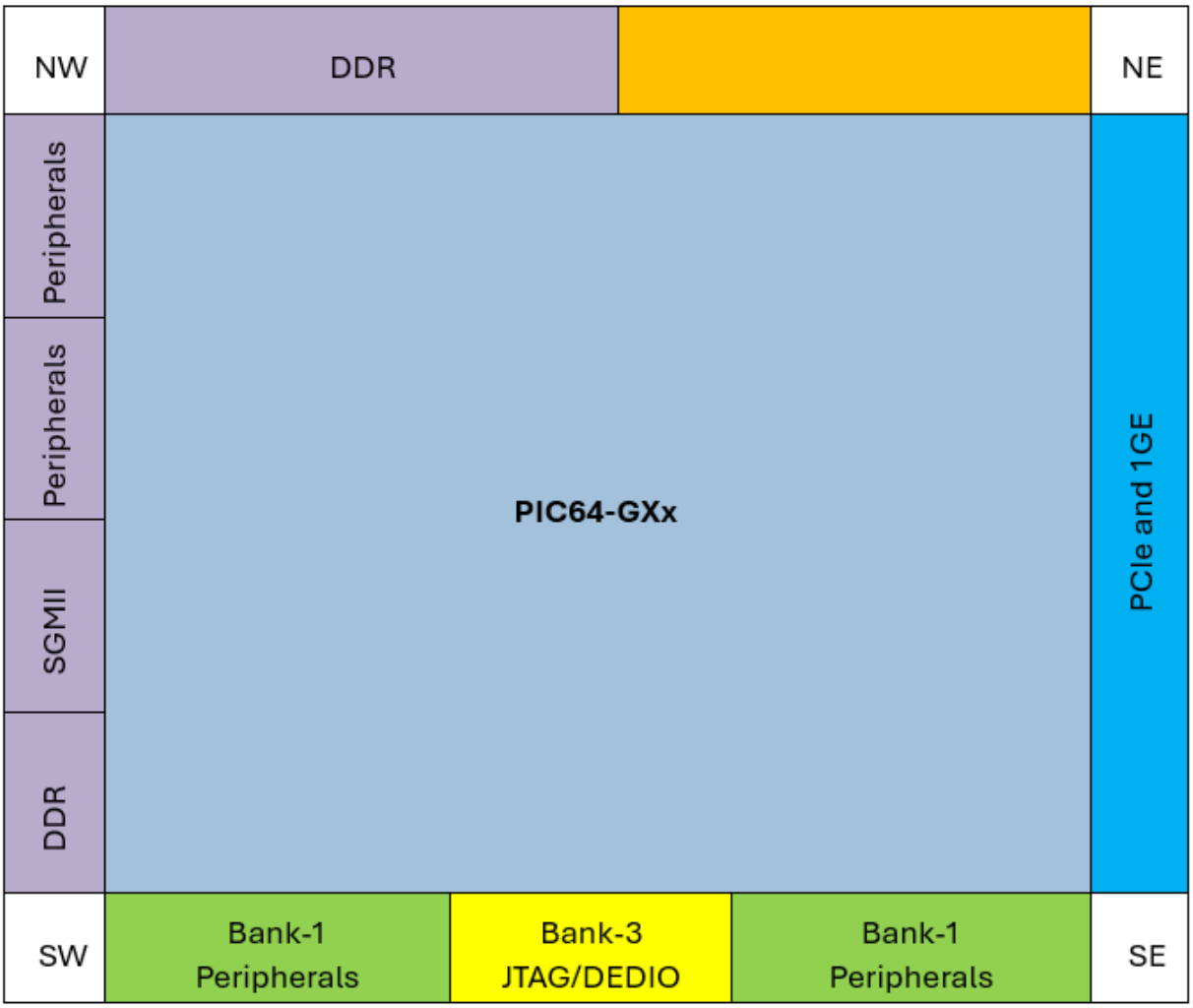
14.1 FCSG325 Package

14.1.1 FCSG325 Unique Features

The FCSG325 contains only two lanes of PCIe Gen 2 compared to the FCVG484 package.

14.1.2 Package Overview (FCSG325)

Figure 14-1. Package/IO Locations



14.1.3 Pin Assignments (FCSG325)

PKG.PIN	PIC64-GX Pin Names	Direction	I/O Type
B7	TMS	I	JTAG
A9	TDI	I	JTAG
A7	TDO	O	JTAG
A8	TCK	I	JTAG

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PKG.PIN	PIC64-GX Pin Names	Direction	I/O Type
B9	TRSTB	I	JTAG
E8	DEVRST_N	I	DEDIO
C6	Config_1 (Pull -Down)	I/O	DEDIO
B6	Config_2 (Pull -Down)	I/O	DEDIO
A5	Config_3 (No Connect)	O	DEDIO
D8	Config_4 (Pull Up)	I	DEDIO
E9	Config_5 (Pull -Down)	I	DEDIO
G9	Config_6 (Pull-Up to 4.7K)	I	DEDIO
G10	Config_7 (Pull -Down)	I	DEDIO
B19	GPIO_2_31	I/O	GPIO
B20	GPIO_2_30	I/O	GPIO
A17	HDMI Data2 P	I/O	HDMI
B17	HDMI Data2 N	I/O	HDMI
D18	HDMI Data1 P	I/O	HDMI
D17	HDMI Data1 N	I/O	HDMI
A19	HDMI Data0 P	I/O	HDMI
A18	HDMI Data0 N	I/O	HDMI
C21	GPIO_2_33	I/O	GPIO
C20	GPIO_2_32	I/O	GPIO
A20	HDMI Clock P	I/O	GPIO
B21	HDMI Clock N	I/O	GPIO
E17	MAC_0_MDIO/GPIO_2_1	I/O	Ethernet/GPIO
D16	MAC_0_MDC/GPIO_2_0	I/O	Ethernet/GPIO
D21	MAC_1_MDIO/GPIO_2_3	I/O	Ethernet/GPIO
E21	MAC_1_MDC/GPIO_2_2	I/O	Ethernet/GPIO
G17	SPI_0_SS0/GPIO_2_5	I/O	SPI/GPIO
G18	SPI_0_CLK/GPIO_2_4	I/O	SPI/GPIO
F19	MMUART_1_TXD/GPIO_2_9	I/O	MMUART/GPIO
E20	MMUART_1_RXD/GPIO_2_8	I/O	MMUART/GPIO
F18	PCIE_PERST#/GPIO_2_11	I/O	PCIe/GPIO
E18	PCIE_WAKE#/GPIO_2_10	I/O	PCIe/GPIO
F20	SPI_0_DI/GPIO_2_7	I/O	SPI/GPIO
F21	SPI_0_DO/GPIO_2_6	I/O	SPI/GPIO
U20	No Connect	I/O	No Connect
U21	No Connect	I/O	No Connect
T20	No Connect	I/O	No Connect
T21	No Connect	I/O	No Connect
T19	No Connect	I/O	No Connect
T18	No Connect	I/O	No Connect

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PKG.PIN	PIC64-GX Pin Names	Direction	I/O Type
U17	No Connect	I/O	No Connect
T17	No Connect	I/O	No Connect
R17	No Connect	I/O	No Connect
R18	No Connect	I/O	No Connect
U18	No Connect	I/O	No Connect
V18	No Connect	I/O	No Connect
AA20	No Connect	I/O	No Connect
Y20	No Connect	I/O	No Connect
AA19	No Connect	I/O	No Connect
Y19	No Connect	I/O	No Connect
V21	No Connect	I/O	No Connect
W20	No Connect	I/O	No Connect
AA18	No Connect	I/O	No Connect
AA17	No Connect	I/O	No Connect
W21	No Connect	I/O	No Connect
Y21	No Connect	I/O	No Connect
Y17	No Connect	I/O	No Connect
W16	No Connect	I/O	No Connect
Y16	No Connect	I/O	No Connect
Y15	No Connect	I/O	No Connect
U15	No Connect	I/O	No Connect
V15	No Connect	I/O	No Connect
AA15	No Connect	I/O	No Connect
Y14	No Connect	I/O	No Connect
V17	No Connect	I/O	No Connect
V16	No Connect	I/O	No Connect
V13	DDR_DM1	I/O	DDR
R14	DDR_DQ15	I/O	DDR
U14	DDR_DQ14	I/O	DDR
AA14	DDR_DQ13	I/O	DDR
AA13	DDR_DQ12	I/O	DDR
U12	DDR_DQS_N1	I/O	DDR
V12	DDR_DQS_P1	I/O	DDR
Y12	DDR_DQ11	I/O	DDR
AA12	DDR_DQ10	I/O	DDR
R13	DDR_DQ9	I/O	DDR
U13	DDR_DQ8	I/O	DDR
V11	DDR_DM0	I/O	DDR
W11	DDR_DQ7	I/O	DDR

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PKG.PIN	PIC64-GX Pin Names	Direction	I/O Type
Y11	DDR_DQ6	I/O	DDR
R11	DDR_DQ5	I/O	DDR
U9	DDR_DQ4	I/O	DDR
Y10	DDR_DQS_N0	I/O	DDR
AA10	DDR_DQS_P0	I/O	DDR
Y9	DDR_DQ3	I/O	DDR
AA9	DDR_DQ2	I/O	DDR
U10	DDR_DQ1	I/O	DDR
V10	DDR_DQ0	I/O	DDR
R9	DDR_VREF_IN	I	DDR_VREF
U8	DDR_A9	I/O	DDR
U6	DDR_A8	I/O	DDR
V7	DDR_A7	I/O	DDR
U7	DDR_A6	I/O	DDR
Y5	DDR_A5	I/O	DDR
AA5	DDR_A4	I/O	DDR
AA8	DDR_A3	I/O	DDR
V8	DDR_A2	I/O	DDR
Y6	DDR_A1	I/O	DDR
W6	DDR_A0	I/O	DDR
AA7	DDR_CK_N0	I/O	DDR
Y7	DDR_CK0/DDR_PLL0_OUT0	I/O	DDR
U5	DDR_BA1	I/O	DDR
AA3	DDR_BA0	I/O	DDR
V5	DDR3_WE_N	I/O	DDR
AA4	DDR_A16	I/O	DDR
W2	DDR_A15	I/O	DDR
Y3	DDR_A14	I/O	DDR
W1	DDR_A13	I/O	DDR
V1	DDR_A12	I/O	DDR
AA2	DDR_A11	I/O	DDR
Y2	DDR_A10	I/O	DDR
R4	DDR_ALERT_N	I/O	DDR
T4	DDR_PARITY	I/O	DDR
T5	DDR_ACT_N	I/O	DDR
T2	DDR_ODT0	I/O	DDR
R1	DDR_CKE0	I/O	DDR
U4	DDR_CS0	I/O	DDR
U2	DDR_BG1	I/O	DDR

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PKG.PIN	PIC64-GX Pin Names	Direction	I/O Type
U1	DDR_BG0	I/O	DDR
T3	DDR_RAM_RST_N/DDR_PLL0_OUT1	I/O	DDR
R2	SGMII_RXN1	Input	SGMII
P2	SGMII_RXP1	Input	SGMII
P1	SGMII_TXN1	I/O	SGMII
N1	SGMII_TXP1	I/O	SGMII
M1	SGMII_RXN0	Input	SGMII
M2	SGMII_RXP0	Input	SGMII
N5	SGMII_TXN0	I/O	SGMII
N7	SGMII_TXP0	I/O	SGMII
P4	REFCLK_IN_N	Input	REFCLK
P5	REFCLK_IN_P	Input	REFCLK
L2	EMMC_CLK/SD_CLK	I/O	EMMC/SD
M4	EMMC_CMD/SD_CMD	I/O	EMMC/SD
L3	EMMC_DATA0/SD_DATA0	I/O	EMMC/SD
K1	EMMC_DATA1/SD_DATA1	I/O	EMMC/SD
M5	EMMC_DATA2/SD_DATA2	I/O	EMMC/SD
L5	EMMC_DATA3/SD_DATA3	I/O	EMMC/SD
K2	EMMC_STRB/SD_CD	I/O	EMMC/SD
H2	EMMC_RSTN/SD_WP	I/O	EMMC/SD
L7	EMMC_DATA4/SD_ PWR	I/O	EMMC/SD
K7	EMMC_DATA5/SD_VOLT_SEL	I/O	EMMC/SD
H1	EMMC_DATA6	I/O	EMMC/SD
K4	EMMC_DATA7	I/O	EMMC/SD
K5	GPIO_0_0/MIPI CAM RESET	I/O	GPIO/MIPI
J1	GPIO_0_1/MIPI CAM STANDBY	I/O	GPIO/MIPI
F3	USB_CLK/GPIO_1_0	I/O	USB/GPIO
G2	USB_DIR/GPIO_1_1	Input	USB/GPIO
H4	USB_NXT/GPIO_1_2	I/O	USB/GPIO
G5	USB_STP/GPIO_1_3	I/O	USB/GPIO
G1	USB_DATA0/GPIO_1_4	I/O	USB/GPIO
F2	USB_DATA1/GPIO_1_5	Input	USB/GPIO
F4	USB_DATA2/GPIO_1_6	I/O	USB/GPIO
E5	USB_DATA3/GPIO_1_7	Input	USB/GPIO
E2	USB_DATA4/GPIO_1_8	I/O	USB/GPIO
E1	USB_DATA5/GPIO_1_9	I/O	USB/GPIO
E4	USB_DATA6/GPIO_1_10	I/O	USB/GPIO
E6	USB_DATA7/GPIO_1_11	I/O	USB/GPIO
A2	I2C_1_SCL/GPIO_1_12/HDMI i2c SCL/MIPI_CAM_I2C_SCL	I/O	I2C/HDMI/MIPI

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PKG.PIN	PIC64-GX Pin Names	Direction	I/O Type
B1	I2C_1_SDA/GPIO_1_13/HDMI i2c SDA/MIPI_CAM_I2C_SDA	I/O	I2C/HDMI/MIPI
A3	MMUART_0_RXD (B)/GPIO_1_14	I/O	MMUART/GPIO
B5	MMUART_0_TXD (B)/GPIO_1_15	I/O	MMUART/GPIO
D6	SPI_1_CLK (B)/GPIO_1_16	I/O	SPI/GPIO
D5	SPI_1_SS0 (B)/GPIO_1_17	I/O	SPI/GPIO
A4	SPI_1_DO (B)/GPIO_1_18	I/O	SPI/GPIO
B4	SPI_1_DI (B)/GPIO_1_19	I/O	SPI/GPIO
B2	GPIO_1_20/HDMI HPD	I/O	HDMI/GPIO
C2	I2C_0_SCL (B)/GPIO_1_21/PCIE_SMCLK	I/O	I2C/GPIO/PCIE
D1	I2C_0_SDA (B)/GPIO_1_22/PCIE_SMDAT	I/O	I2C/GPIO/PCIE
C1	GPIO_1_23	I/O	GPIO
E10	QSPI_CLK/GPIO_2_13	I/O	QSPI/GPIO
E11	QSPI_SS0/GPIO_2_12	I/O	QSPI/GPIO
B10	MIPI_RX_P1	I/O	MIPI
A10	MIPI_RX_N1	I/O	MIPI
C11	MIPI_RX_P0	I/O	MIPI
B11	MIPI_RX_N0	I/O	MIPI
E12	QSPI_DATA1/GPIO_2_15	I/O	QSPI/GPIO
G12	QSPI_DATA0/GPIO_2_14	I/O	QSPI/GPIO
D11	QSPI_DATA3/GPIO_2_17	I/O	QSPI/GPIO
D12	QSPI_DATA2/GPIO_2_16	I/O	QSPI/GPIO
A12	MIPI_RX_CKP	I/O	MIPI
B12	MIPI_RX_CKN	I/O	MIPI
E15	MMUART_3_TXD/GPIO_2_19	I/O	MMUART/GPIO
E14	MMUART_3_RXD/GPIO_2_18	I/O	MMUART/GPIO
A13	MMUART_4_TXD/GPIO_2_21	I/O	MMUART/GPIO
A14	MMUART_4_RXD/GPIO_2_20	I/O	MMUART/GPIO
D13	CAN_1_RXBUS/GPIO_2_23	Out	CAN/GPIO
E13	CAN_1_TXBUS/GPIO_2_22	I/O	CAN/GPIO
B14	CAN_1_TX_EBL_N/GPIO_2_25	I/O	CAN/GPIO
B15	CAN_0_TX_EBL_N/GPIO_2_24	I/O	CAN/GPIO
D15	MMUART_2_TXD/GPIO_2_27	I/O	MMUART/GPIO
C16	MMUART_2_RXD/GPIO_2_26	I/O	MMUART/GPIO
A15	CAN_0_RXBUS/GPIO_2_29	I/O	CAN/GPIO
B16	CAN_0_TXBUS/GPIO_2_28	I/O	CAN/GPIO
N18	PCIE_REFCLK_P	Input	PCIe
N17	PCIE_REFCLK_N	Input	PCIe
P21	PCIE_RX2_P	HSI	PCIe
P20	PCIE_RX2_N	HSI	PCIe

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PKG.PIN	PIC64-GX Pin Names	Direction	I/O Type
M21	PCIE_TX2_P	HSO	PCIe
M20	PCIE_TX2_N	HSO	PCIe
J18	No Connect	HSI	No Connect
J17	No Connect	HSI	No Connect
K21	PCIE_RX0_P	HSI	PCIe
K20	PCIE_RX0_N	HSI	PCIe
H21	PCIE_TX0_P	HSO	PCIe
H20	PCIE_TX0_N	HSO	PCIe
G20	PCIE_VREF	N/A	PCIe
H18	PCIE_VREF	N/A	PCIe
K17	VDD_XCVR_CLK	N/A	VDD_XCVR_CLK
M17	VDD_XCVR_CLK	N/A	VDD_XCVR_CLK
L17	VDDA25	N/A	VDDA25
L19	VDDA25	N/A	VDDA25
J20	VDDA	N/A	VDDA
L18	VDDA	N/A	VDDA
L20	VDDA	N/A	VDDA
N20	VDDA	N/A	VDDA
P15	VDD18	N/A	VDD18
P7	VDD18	N/A	VDD18
R10	VDD18	N/A	VDD18
R12	VDD18	N/A	VDD18
F17	VDDAUX1	N/A	VDDAUX1
G11	VDDAUX1	N/A	VDDAUX1
G13	VDDAUX1	N/A	VDDAUX1
F5	VDDAUX2	N/A	VDDAUX2
G4	VDDAUX2	N/A	VDDAUX2
J5	VDDAUX4	N/A	VDDAUX4
J7	VDDAUX4	N/A	VDDAUX4
U16	VDDI0	N/A	VDDI0
V20	VDDI0	N/A	VDDI0
Y13	VDDI0	N/A	VDDI0
Y18	VDDI0	N/A	VDDI0
A11	VDDI1	N/A	VDDI1
B13	VDDI1	N/A	VDDI1
B18	VDDI1	N/A	VDDI1
B8	VDDI1	N/A	VDDI1
D10	VDDI1	N/A	VDDI1
D20	VDDI1	N/A	VDDI1

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PKG.PIN	PIC64-GX Pin Names	Direction	I/O Type
E16	VDDI1	N/A	VDDI1
G15	VDDI1	N/A	VDDI1
B3	VDDI2	N/A	VDDI2
D2	VDDI2	N/A	VDDI2
H5	VDDI2	N/A	VDDI2
E7	VDDI3	N/A	VDDI3
G8	VDDI3	N/A	VDDI3
J2	VDDI4	N/A	VDDI4
L4	VDDI4	N/A	VDDI4
M7	VDDI5	N/A	VDDI5
N2	VDDI5	N/A	VDDI5
R5	VDDI6	N/A	VDDI6
U11	VDDI6	N/A	VDDI6
V2	VDDI6	N/A	VDDI6
V6	VDDI6	N/A	VDDI6
Y1	VDDI6	N/A	VDDI6
Y4	VDDI6	N/A	VDDI6
Y8	VDDI6	N/A	VDDI6
A1	VSS	N/A	VSS
G7	VDD25	N/A	VDD25
H15	VDD25	N/A	VDD25
R15	VDD25	N/A	VDD25
R7	VDD25	N/A	VDD25
J10	VDD	N/A	VDD
J12	VDD	N/A	VDD
K11	VDD	N/A	VDD
K13	VDD	N/A	VDD
K15	VDD	N/A	VDD
K9	VDD	N/A	VDD
L10	VDD	N/A	VDD
L12	VDD	N/A	VDD
M11	VDD	N/A	VDD
M13	VDD	N/A	VDD
M15	VDD	N/A	VDD
M9	VDD	N/A	VDD
N10	VDD	N/A	VDD
N12	VDD	N/A	VDD
G21	VSS	N/A	VSS
H17	VSS	N/A	VSS

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PKG.PIN	PIC64-GX Pin Names	Direction	I/O Type
J21	VSS	N/A	VSS
K18	VSS	N/A	VSS
L21	VSS	N/A	VSS
M18	VSS	N/A	VSS
N21	VSS	N/A	VSS
P17	VSS	N/A	VSS
P18	VSS	N/A	VSS
R20	VSS	N/A	VSS
R21	VSS	N/A	VSS
A16	VSS	N/A	VSS
A21	VSS	N/A	VSS
A6	VSS	N/A	VSS
AA1	VSS	N/A	VSS
AA11	VSS	N/A	VSS
AA16	VSS	N/A	VSS
AA21	VSS	N/A	VSS
AA6	VSS	N/A	VSS
D14	VSS	N/A	VSS
D4	VSS	N/A	VSS
D7	VSS	N/A	VSS
D9	VSS	N/A	VSS
F1	VSS	N/A	VSS
G14	VSS	N/A	VSS
H7	VSS	N/A	VSS
J11	VSS	N/A	VSS
J13	VSS	N/A	VSS
J15	VSS	N/A	VSS
J4	VSS	N/A	VSS
J9	VSS	N/A	VSS
K10	VSS	N/A	VSS
K12	VSS	N/A	VSS
L1	VSS	N/A	VSS
L11	VSS	N/A	VSS
L13	VSS	N/A	VSS
L15	VSS	N/A	VSS
L9	VSS	N/A	VSS
M10	VSS	N/A	VSS
M12	VSS	N/A	VSS
N11	VSS	N/A	VSS

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued			
PKG.PIN	PIC64-GX Pin Names	Direction	I/O Type
N13	VSS	N/A	VSS
N15	VSS	N/A	VSS
N4	VSS	N/A	VSS
N9	VSS	N/A	VSS
R8	VSS	N/A	VSS
T1	VSS	N/A	VSS
V14	VSS	N/A	VSS
V4	VSS	N/A	VSS
V9	VSS	N/A	VSS

14.2 FCVG484 Package

© 2024 Microchip Technology Inc. and its subsidiaries

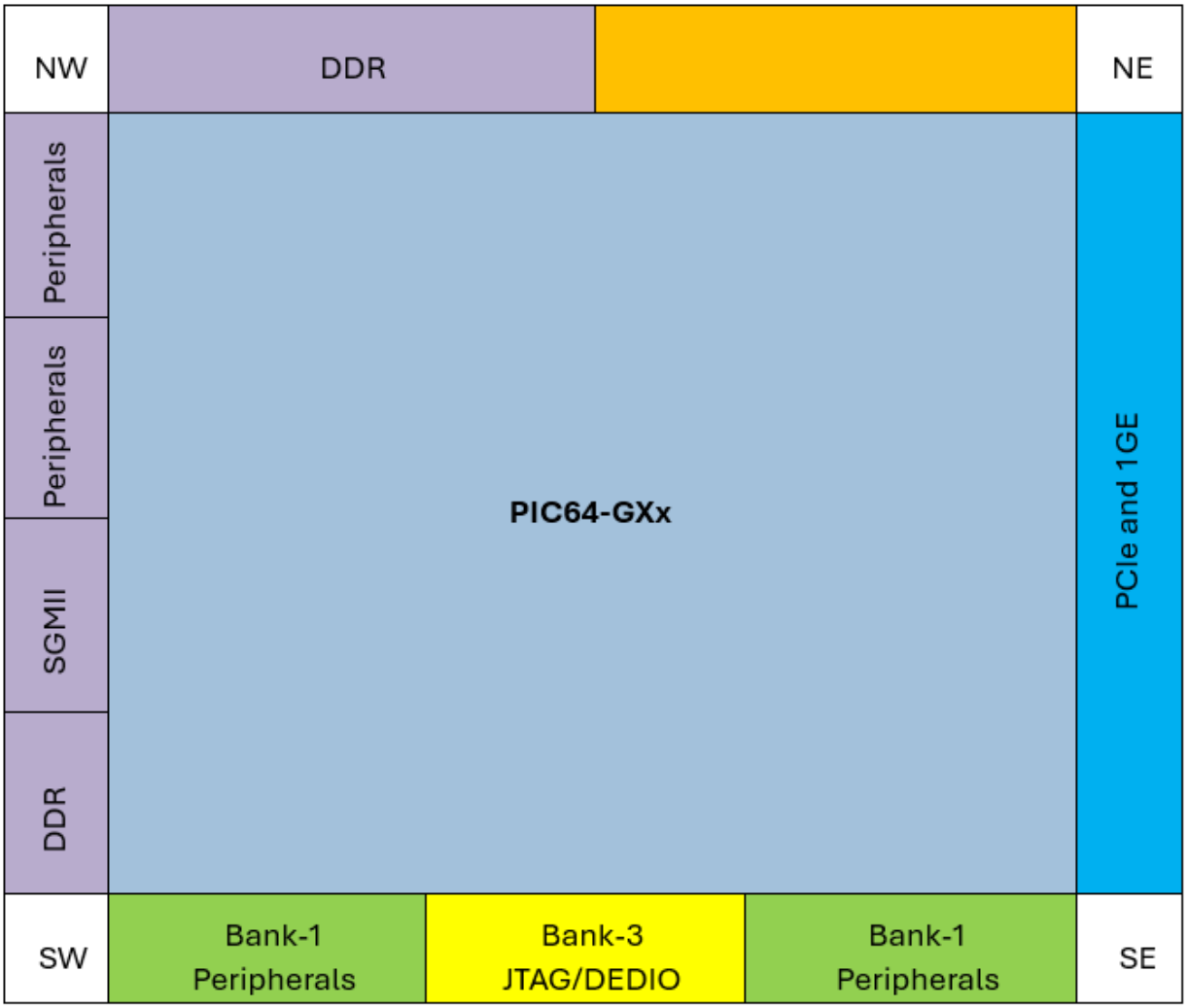
The following section describes the FCVG484 package.

14.2.1 FCVG484 Unique Features

The FCVG484 package contains four lanes of PCIe Gen 2.

14.2.2 Package Overview (FCVG484)

Figure 14-2. Package/IO Locations



14.2.3 Pin Assignments (FCVG484)

PIN	FCVG484 Pin Names	Direction	I/O Type
F8	TMS	I	JTAG
G9	TDI	I	JTAG
E8	TDO	O	JTAG
E9	TCK	I	JTAG
G8	TRSTB	I	JTAG
H7	DEVRST_N	I	DEDIO
E6	Config_1	I/O	DEDIO

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PIN	FCVG484 Pin Names	Direction	I/O Type
G7	Config_2	I/O	DEDIO
F7	Config_3	O	DEDIO
H10	Config_4	I	DEDIO
H9	Config_5	I	DEDIO
G10	Config_6	I	DEDIO
H11	Config_7	I	DEDIO
D8	MMUART_3_RXD/GPIO_2_18	I/O	MMUART/GPIO
D9	MMUART_3_TXD/GPIO_2_19	I/O	MMUART/GPIO
B8	MMUART_4_RXD/GPIO_2_20	I/O	MMUART/GPIO
A8	MMUART_4_TXD/GPIO_2_21	I/O	MMUART/GPIO
C9	CAN_1_TXBUS/GPIO_2_22	I/O	CAN/GPIO
C10	CAN_1_RXBUS/GPIO_2_23	I/O	CAN/GPIO
A11	CAN_0_TX_EBL_N/GPIO_2_24	I/O	CAN/GPIO
A10	CAN_1_TX_EBL_N/GPIO_2_25	I/O	CAN/GPIO
D11	MMUART_2_RXD/GPIO_2_26	I/O	MMUART/GPIO
C11	MMUART_2_TXD/GPIO_2_27	I/O	MMUART/GPIO
B9	CAN_0_TXBUS/GPIO_2_28	I/O	CAN/GPIO
B10	CAN_0_RXBUS/GPIO_2_29	I/O	CAN/GPIO
F15	NC	N/A	N/A
G14	NC	N/A	N/A
G15	NC	N/A	N/A
H15	NC	N/A	N/A
G17	NC	N/A	N/A
H17	NC	N/A	N/A
C12	PCIE_PERST_2#/GPIO_2_30	I/O	PCIe/GPIO
B12	GPIO_2_31	I/O	GPIO
E16	NC	N/A	N/A
D16	NC	N/A	N/A
A12	HDMI Data2 N	I/O	HDMI
A13	HDMI Data2 P	I/O	HDMI
C16	NC	N/A	N/A
D17	NC	N/A	N/A
A16	NC	N/A	N/A
A17	NC	N/A	N/A
D14	HDMI Data1 N	I/O	HDMI
D13	HDMI Data1 P	I/O	HDMI
B14	HDMI Data0 N	I/O	HDMI
B13	HDMI Data0 P	I/O	HDMI
B15	HDMI Clock N	I/O	HDMI

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PIN	FCVG484 Pin Names	Direction	I/O Type
A15	HDMI Clock P	I/O	HDMI
C15	GPIO_2_32	I/O	GPIO
C14	GPIO_2_33	I/O	GPIO
C19	NC		NC
C20	NC		NC
B20	NC		NC
B19	NC		NC
A20	NC		NC
A21	NC		NC
D21	NC		NC
D20	NC		NC
B21	NC		NC
B22	NC		NC
C22	NC		NC
D22	NC		NC
V22	NC		NC
W22	NC		NC
Y21	NC		NC
Y20	NC	I/O	NC
V21	NC	I/O	NC
W21	NC	I/O	NC
AA22	NC	I/O	NC
AA21	NC	I/O	NC
AB20	NC	I/O	NC
AB19	NC	I/O	NC
AA20	NC	I/O	NC
AB21	NC	I/O	NC
V19	NC	I/O	NC
V20	NC	I/O	NC
U17	NC	I/O	NC
T17	NC	I/O	NC
U18	NC	I/O	NC
U19	NC	I/O	NC
R16	NC	I/O	NC
T16	NC	I/O	NC
V16	NC	I/O	NC
V17	NC	I/O	NC
R14	NC	I/O	NC
R15	NC	I/O	NC

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PIN	FCVG484 Pin Names	Direction	I/O Type
AA18	NC	I/O	NC
AB18	NC	I/O	NC
W18	NC	I/O	NC
W19	NC	I/O	NC
Y18	NC	I/O	NC
Y19	NC	I/O	NC
AA17	NC	I/O	NC
AB17	NC	I/O	NC
Y16	NC	I/O	NC
AA16	NC	I/O	NC
W17	NC	I/O	NC
W16	NC	I/O	NC
AB15	NC	I/O	NC
AB14	NC	I/O	NC
AA15	NC	I/O	NC
Y15	NC	I/O	NC
AB13	NC	I/O	NC
AA13	NC	I/O	NC
W14	NC	I/O	NC
Y14	NC		NC
AB12	NC		NC
AA12	NC		NC
Y13	NC		NC
W13	NC		NC
V15	NC		NC
V14	NC		NC
T15	NC		NC
U15	NC		NC
V12	NC		NC
W12	NC		NC
U13	NC		NC
U14	NC		NC
T12	NC		NC
U12	NC		NC
T13	NC		NC
R12	NC		NC
T10	DDR_DM3	I/O	DDR
P8	DDR_DQ31	I/O	DDR
R8	DDR_DQ30	I/O	DDR

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PIN	FCVG484 Pin Names	Direction	I/O Type
R10	DDR_DQ29	I/O	DDR
R11	DDR_DQ28	I/O	DDR
V9	DDR_DQS_N3	I/O	DDR
U9	DDR_DQS_P3	I/O	DDR
U10	DDR_DQ27	I/O	DDR
T11	DDR_DQ26	I/O	DDR
T8	DDR_DQ25	I/O	DDR
U8	DDR_DQ24	I/O	DDR
Y10	DDR_DM2	I/O	DDR
W11	DDR_DQ23	I/O	DDR
Y11	DDR_DQ22	I/O	DDR
V10	DDR_DQ21	I/O	DDR
V11	DDR_DQ20	I/O	DDR
AA10	DDR_DQS_N2	I/O	DDR
AA11	DDR_DQS_P2	I/O	DDR
AB10	DDR_DQ19	I/O	DDR
AB9	DDR_DQ18	I/O	DDR
W9	DDR_DQ17	I/O	DDR
Y9	DDR_DQ16	I/O	DDR
AA7	DDR_DM1	I/O	DDR
AA8	DDR_DQ15	I/O	DDR
Y8	DDR_DQ14	I/O	DDR
AB8	DDR_DQ13	I/O	DDR
AB7	DDR_DQ12	I/O	DDR
W6	DDR_DQS_N1	I/O	DDR
Y6	DDR_DQS_P1	I/O	DDR
AA6	DDR_DQ11	I/O	DDR
AB5	DDR_DQ10	I/O	DDR
W7	DDR_DQ9	I/O	DDR
W8	DDR_DQ8	I/O	DDR
AB4	DDR_DM0	I/O	DDR
AB3	DDR_DQ7	I/O	DDR
AB2	DDR_DQ6	I/O	DDR
AA5	DDR_DQ5	I/O	DDR
Y5	DDR_DQ4	I/O	DDR
AA3	DDR_DQS_N0	I/O	DDR
AA2	DDR_DQS_P0	I/O	DDR
AA1	DDR_DQ3	I/O	DDR
Y1	DDR_DQ2	I/O	DDR

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PIN	FCVG484 Pin Names	Direction	I/O Type
Y4	DDR_DQ1	I/O	DDR
Y3	DDR_DQ0	I/O	DDR
V5	DDR_DM4	I	DDR
V1	DDR_DQS_N4	I/O	DDR
V2	DDR_DQS_P4	I/O	DDR
W1	DDR_DQ35	I/O	DDR
W2	DDR_DQ34	I/O	DDR
W3	DDR_DQ33	I/O	DDR
W4	DDR_DQ32	I/O	DDR
U7	DDR_A9	I/O	DDR
P7	DDR_A8	I/O	DDR
T7	DDR_A7	I/O	DDR
P6	DDR_A6	I/O	DDR
R5	DDR_A5	I/O	DDR
R6	DDR_A4	I/O	DDR
V6	DDR_A3	I/O	DDR
V7	DDR_A2	I/O	DDR
T6	DDR_A1	I/O	DDR
T5	DDR_A0	I/O	DDR
U4	DDR_CK_N0	I/O	DDR
U5	DDR_CK0/DDR_PLL0_OUT0	I/O	DDR
T3	DDR_BA1	I/O	DDR
P4	DDR_BA0	I/O	DDR
U3	DDR3_WE_N	I/O	DDR
R4	DDR_A16	I/O	DDR
R3	DDR_A15	I/O	DDR
P3	DDR_A14	I/O	DDR
P2	DDR_A13	I/O	DDR
P1	DDR_A12	I/O	DDR
R1	DDR_A11	I/O	DDR
U2	DDR_A10	I/O	DDR
T2	DDR_CK_N1	I/O	DDR
T1	DDR_CK1/DDR_PLL0_OUT0	I/O	DDR
N4	DDR_ALERT_N	I/O	DDR
M5	DDR_PARITY	I/O	DDR
N5	DDR_ACT_N	I/O	DDR
M4	DDR_ODT1	I/O	DDR
N2	DDR_CKE1	I/O	DDR
M3	DDR_CS1	I/O	DDR

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PIN	FCVG484 Pin Names	Direction	I/O Type
L1	DDR_ODT0	I/O	DDR
K1	DDR_CKE0	I/O	DDR
L3	DDR_CS0	I/O	DDR
M2	DDR_BG1	I/O	DDR
N1	DDR_BG0	I/O	DDR
L2	DDR_RAM_RST_N/DDR_PLL0_OUT1	I/O	DDR
K7	SGMII_RXN1	I/O	SGMII
K6	SGMII_RXP1	I/O	SGMII
N8	SGMII_TXN1	I/O	SGMII
M7	SGMII_TXP1	I/O	SGMII
L6	SGMII_RXN0	I/O	SGMII
L5	SGMII_RXP0	I/O	SGMII
N7	SGMII_TXN0	I/O	SGMII
N6	SGMII_TXP0	I/O	SGMII
L8	REFCLK_IN_N	I/O	SGMII
L7	REFCLK_IN_P	I/O	SGMII
J1	EMMC_CLK/SD_CLK	I/O	SD
K5	EMMC_CMD/SD_CMD	I/O	SD
H1	EMMC_DATA0/SD_DATA0	I/O	SD
J4	EMMC_DATA1/SD_DATA1	I/O	SD
K4	EMMC_DATA2/SD_DATA2	I/O	SD
J7	EMMC_DATA3/SD_DATA3	I/O	SD
K3	EMMC_STRB/SD_CD	I/O	SD
H4	EMMC_RSTN	I/O	SD
J6	EMMC_DATA4/SD_POW	I/O	SD
H6	EMMC_DATA5/SD_VOLT_SEL	I/O	SD
J3	EMMC_DATA6	I/O	SD
H2	EMMC_DATA7	I/O	SD
H5	GPIO_0_0/MIPI CAM RESET	I/O	MIPI/GPIO
J2	GPIO_0_1/MIPI CAM STANDBY	I/O	MIPI/GPIO
G2	USB_CLK/GPIO_1_0	I/O	USB/GPIO
F1	USB_DIR/GPIO_1_1	I/O	USB/GPIO
G5	USB_NXT/GPIO_1_2	I/O	USB/GPIO
G4	USB_STP/GPIO_1_3	I/O	USB/GPIO
F2	USB_DATA0/GPIO_1_4	I/O	USB/GPIO
E1	USB_DATA1/GPIO_1_5	I/O	USB/GPIO
G3	USB_DATA2/GPIO_1_6	I/O	USB/GPIO
F5	USB_DATA3/GPIO_1_7	I/O	USB/GPIO
D1	USB_DATA4/GPIO_1_8	I/O	USB/GPIO

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PIN	FCVG484 Pin Names	Direction	I/O Type
D2	USB_DATA5/GPIO_1_9	I/O	USB/GPIO
F6	USB_DATA6/GPIO_1_10	I/O	USB/GPIO
F3	USB_DATA7/GPIO_1_11	I/O	USB/GPIO
C1	I2C_1_SCL/GPIO_1_12/HDMI i2c SCL/MIPI_CAM_I2C_SCL	I/O	I2C/HDMI/MIPI
B1	I2C_1_SDA/GPIO_1_13/HDMI i2c SDA/MIPI_CAM_I2C_SDA	I/O	I2C/HDMI/MIPI
D3	MMUART_0_RXD (B)/GPIO_1_14	I/O	MMUART/GPIO
C2	MMUART_0_TXD (B)/GPIO_1_15	I/O	MMUART/GPIO
E5	SPI_1_CLK (B)/GPIO_1_16	I/O	SPI/GPIO
E4	SPI_1_SS0 (B)/GPIO_1_17	I/O	SPI/GPIO
B2	SPI_1_DO (B)/GPIO_1_18	I/O	SPI/GPIO
A2	SPI_1_DI (B)/GPIO_1_19	I/O	SPI/GPIO
B3	GPIO_1_20/HDMI HPD	I/O	HDMI /GPIO
A3	I2C_0_SCL (B)/GPIO_1_21/PCIE_SMCLK	I/O	I2C/GPIO
E3	I2C_0_SDA (B)/GPIO_1_22/PCIE_SMDAT	I/O	I2C/GPIO
D4	GPIO_1_23	I/O	GPIO
E14	MAC_1_MDC/GPIO_2_0	I/O	GPIO
E15	MAC_1_MDIO/GPIO_2_1	I/O	GPIO
F16	MAC_2_MDC/GPIO_2_2	I/O	GPIO
F17	MAC_2_MDIO/GPIO_2_3	I/O	GPIO
D19	SPI_0_CLK/GPIO_2_4	I/O	GPIO
E19	SPI_0_SS0/GPIO_2_5	I/O	GPIO
C17	MMUART_1_RXD/GPIO_2_6	I/O	GPIO
B17	MMUART_1_TXD/GPIO_2_7	I/O	GPIO
E18	PCIE_WAKE#/GPIO_2_8	I/O	GPIO
D18	PCIE_PERST_1#/GPIO_2_9	I/O	PCIe/GPIO
B18	SPI_0_DO/GPIO_2_10	I/O	GPIO
A18	SPI_0_DI/GPIO_2_11	I/O	GPIO
D6	QSPI_SS0/GPIO_2_12	I/O	QSPI/GPIO
D7	QSPI_CLK/GPIO_2_13	I/O	QSPI/GPIO
B4	MIPI_RX_N1	I/O	GPIO
C4	MIPI_RX_P1	I/O	GPIO
B5	MIPI_RX_N0	I/O	GPIO
C5	MIPI_RX_P0	I/O	GPIO
C7	QSPI_DATA0/GPIO_2_14	I/O	QSPI/GPIO
C6	QSPI_DATA1/GPIO_2_15	I/O	QSPI/GPIO
A5	QSPI_DATA2/GPIO_2_16	I/O	QSPI/GPIO
A6	QSPI_DATA3/GPIO_2_17	I/O	QSPI/GPIO
B7	MIPI_RX_CKN	I/O	GPIO
A7	MIPI_RX_CKP	I/O	GPIO

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PIN	FCVG484 Pin Names	Direction	I/O Type
E10	NC	N/A	N/A
F10	NC	N/A	N/A
F12	NC	N/A	N/A
F11	NC	N/A	N/A
H12	NC	N/A	N/A
G12	NC	N/A	N/A
D12	NC	N/A	N/A
E11	NC	N/A	N/A
E13	NC	N/A	N/A
F13	NC	N/A	N/A
G13	NC	N/A	N/A
H13	NC	N/A	N/A
N19	PCIE_REFCLK_P	HSI	PCIe
N20	PCIE_REFCLK_N	HSI	PCIe
T22	PCIE_TX3_P	HSO	PCIe
T21	PCIE_TX3_N	HSO	PCIe
R20	PCIE_RX3_P	HSI	PCIe
R19	PCIE_RX3_N	HSI	PCIe
M22	PCIE_RX2_P	HSI	PCIe
M21	PCIE_RX2_N	HSI	PCIe
P22	PCIE_TX2_P	HSO	PCIe
P21	PCIE_TX2_N	HSO	PCIe
L19	No Connect	HSI	No Connect
L20	No Connect	HSI	No Connect
H22	PCIE_TX1_P	HSO	PCIe
H21	PCIE_TX1_N	HSO	PCIe
K22	PCIE_RX1_P	HSI	PCIe
K21	PCIE_RX1_N	HSI	PCIe
G20	PCIE_RX0_P	HSI	PCIe
G19	PCIE_RX0_N	HSI	PCIe
F22	PCIE_TX0_P	HSO	PCIe
F21	PCIE_TX0_N	HSO	PCIe
J19	No Connect	HSI	No Connect
J20	No Connect	HSI	No Connect
H18	PCIE_VREF	N/A	PCIe
H19	PCIE_VREF	N/A	PCIe
K18	VDD_XCVR_CLK	N/A	VDD_XCVR_CLK
P18	VDD_XCVR_CLK	N/A	VDD_XCVR_CLK
J18	VDDA25	N/A	VDDA25

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PIN	FCVG484 Pin Names	Direction	I/O Type
M18	VDDA25	N/A	VDDA25
G21	VDDA	N/A	VDDA
J21	VDDA	N/A	VDDA
K19	VDDA	N/A	VDDA
L21	VDDA	N/A	VDDA
N21	VDDA	N/A	VDDA
P19	VDDA	N/A	VDDA
R21	VDDA	N/A	VDDA
P10	VDD18	N/A	VDD18
P12	VDD18	N/A	VDD18
P14	VDD18	N/A	VDD18
P16	VDD18	N/A	VDD18
R13	VDD18	N/A	VDD18
Y17	VSS	N/A	VSS
H14	VDDAUX1	N/A	VDDAUX1
H16	VDDAUX1	N/A	VDDAUX1
J13	VDDAUX1	N/A	VDDAUX1
J9	VDDAUX1	N/A	VDDAUX1
K8	VDDAUX2	N/A	VDDAUX2
L9	VDDAUX2	N/A	VDDAUX2
M10	VDDAUX4	N/A	VDDAUX4
N9	VDDAUX4	N/A	VDDAUX4
AA19	VDDI0	N/A	VDDI0
AB16	VDDI0	N/A	VDDI0
T14	VDDI0	N/A	VDDI0
V18	VDDI0	N/A	VDDI0
W15	VDDI0	N/A	VDDI0
Y12	VDDI0	N/A	VDDI0
Y22	VDDI0	N/A	VDDI0
K10	VDDAUX1	N/A	VDDAUX1
A19	VDDI1	N/A	VDDI1
A9	VDDI1	N/A	VDDI1
B16	VDDI1	N/A	VDDI1
B6	VDDI1	N/A	VDDI1
C13	VDDI1	N/A	VDDI1
C21	VDDI1	N/A	VDDI1
D10	VDDI1	N/A	VDDI1
E17	VDDI1	N/A	VDDI1
F14	VDDI1	N/A	VDDI1

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PIN	FCVG484 Pin Names	Direction	I/O Type
G11	VDDI1	N/A	VDDI1
C3	VDDI2	N/A	VDDI2
F4	VDDI2	N/A	VDDI2
G1	VDDI2	N/A	VDDI2
E7	VDDI3	N/A	VDDI3
J11	VDDI3	N/A	VDDI3
J5	VDDI4	N/A	VDDI4
K2	VDDI4	N/A	VDDI4
M6	VDDI5	N/A	VDDI5
M8	VDDI5	N/A	VDDI5
AA9	VDDI6	N/A	VDDI6
AB6	VDDI6	N/A	VDDI6
N3	VDDI6	N/A	VDDI6
R7	VDDI6	N/A	VDDI6
T4	VDDI6	N/A	VDDI6
U1	VDDI6	N/A	VDDI6
U11	VDDI6	N/A	VDDI6
V8	VDDI6	N/A	VDDI6
W5	VDDI6	N/A	VDDI6
Y2	VDDI6	N/A	VDDI6
Y7	VSS	N/A	VSS
H8	VDD25	N/A	VDD25
J17	VDD25	N/A	VDD25
L15	VDD25	N/A	VDD25
R17	VDD25	N/A	VDD25
R9	VDD25	N/A	VDD25
J15	VDD	N/A	VDD
K12	VDD	N/A	VDD
K14	VDD	N/A	VDD
K16	VDD	N/A	VDD
L11	VDD	N/A	VDD
L13	VDD	N/A	VDD
L17	VDD	N/A	VDD
M12	VDD	N/A	VDD
M14	VDD	N/A	VDD
M16	VDD	N/A	VDD
N11	VDD	N/A	VDD
N13	VDD	N/A	VDD
N15	VDD	N/A	VDD

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PIN	FCVG484 Pin Names	Direction	I/O Type
N17	VDD	N/A	VDD
E20	VSS	N/A	VSS
E21	VSS	N/A	VSS
E22	VSS	N/A	VSS
F18	VSS	N/A	VSS
F19	VSS	N/A	VSS
F20	VSS	N/A	VSS
G18	VSS	N/A	VSS
G22	VSS	N/A	VSS
H20	VSS	N/A	VSS
J22	VSS	N/A	VSS
K20	VSS	N/A	VSS
L18	VSS	N/A	VSS
L22	VSS	N/A	VSS
M19	VSS	N/A	VSS
M20	VSS	N/A	VSS
N18	VSS	N/A	VSS
N22	VSS	N/A	VSS
P20	VSS	N/A	VSS
R18	VSS	N/A	VSS
R22	VSS	N/A	VSS
T18	VSS	N/A	VSS
T19	VSS	N/A	VSS
T20	VSS	N/A	VSS
U20	VSS	N/A	VSS
U21	VSS	N/A	VSS
U22	VSS	N/A	VSS
A1	VSS	N/A	VSS
A14	VSS	N/A	VSS
A22	VSS	N/A	VSS
A4	VSS	N/A	VSS
AA14	VSS	N/A	VSS
AA4	VSS	N/A	VSS
AB1	VSS	N/A	VSS
AB11	VSS	N/A	VSS
AB22	VSS	N/A	VSS
B11	VSS	N/A	VSS
C18	VSS	N/A	VSS
C8	VSS	N/A	VSS

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PIN	FCVG484 Pin Names	Direction	I/O Type
D15	VSS	N/A	VSS
D5	VSS	N/A	VSS
E12	VSS	N/A	VSS
E2	VSS	N/A	VSS
F9	VSS	N/A	VSS
G16	VSS	N/A	VSS
G6	VSS	N/A	VSS
H3	VSS	N/A	VSS
J10	VSS	N/A	VSS
J12	VSS	N/A	VSS
J14	VSS	N/A	VSS
J16	VSS	N/A	VSS
J8	VSS	N/A	VSS
K11	VSS	N/A	VSS
K13	VSS	N/A	VSS
K15	VSS	N/A	VSS
K17	VSS	N/A	VSS
K9	VSS	N/A	VSS
L10	VSS	N/A	VSS
L12	VSS	N/A	VSS
899L14	VSS	N/A	VSS
L16	VSS	N/A	VSS
L4	VSS	N/A	VSS
M1	VSS	N/A	VSS
M11	VSS	N/A	VSS
M13	VSS	N/A	VSS
M15	VSS	N/A	VSS
M17	VSS	N/A	VSS
M9	VSS	N/A	VSS
N10	VSS	N/A	VSS
N12	VSS	N/A	VSS
N14	VSS	N/A	VSS
N16	VSS	N/A	VSS
P11	VSS	N/A	VSS
P13	VSS	N/A	VSS
P15	VSS	N/A	VSS
P17	VSS	N/A	VSS
P5	VSS	N/A	VSS
P9	VSS	N/A	VSS

© 2024 Microchip Technology Inc. and its subsidiaries

.....continued

PIN	FCVG484 Pin Names	Direction	I/O Type
R2	VSS	N/A	VSS
T9	VSS	N/A	VSS
U16	VSS	N/A	VSS
U6	VSS	N/A	VSS
V13	VSS	N/A	VSS
V3	VSS	N/A	VSS
W10	VSS	N/A	VSS
W20	VSS	N/A	VSS

14.3 Supply Pins

The following table lists multiple power supply pins required for proper device operation.

Table 14-1. Supply Pins Description

Name	Description	Operating Voltage
PCIE_VREF	Voltage reference for PCIe	0.9V/1.25V
VDD_PCIE_CLK	Power for PCIE reference clock input buffers	2.5V/3.3V
VDDA25	Transceiver PLL power	2.5V
VDDA	Power for transceiver Tx and Rx lanes 0, 1, 2, and 3	1.0V/1.05V
VSS	Core digital ground	NA
VDD	Device core digital supply	1.0V/1.05V
VDDIx (JTAG Bank)	Supply for I/O circuits in a bank	1.8V/2.5V/3.3V
VDDIx (GPIO Banks)	Supply for I/O circuits in a bank	1.2V/1.5V/1.8V/2.5V/3.3V
VDDIx (IO Banks)	Supply for MPU I/O circuits in a bank	1.2V/1.5V/1.8V/2.5V/3.3V
VDDIx (SGMII Banks)	Supply for MPU SGMII circuits in a bank	2.5V/3.3V
VDDIx (DDR Bank)	Supply for MPU DDR circuits in a bank	1.1V/1.2V/1.5V/1.8V
VDDIx (HSIO Banks)	Supply for HSIO I/O circuits in a bank	1.2V/1.5V/1.8V

© 2024 Microchip Technology Inc. and its subsidiaries

15. PIC64GX Boot Modes Fundamentals

This section describes the boot modes of the PIC64GX Microprocessor Subsystem. The Security Controller controls the start-up of the CPU Core Complex harts based on the selected boot mode.

- *No-Boot* is used by blank devices or when debugging embedded software where code should not be executed on power up
- *Secure Boot* implements Microchip supplied factory secure boot authentication of the eNVM content

Once the Security Controller has authenticated the image in the eNVM, it hands control over to the E51 Monitor hart, to perform bring-up of the application harts. Normally, the E51 is running the Hart Software Services (HSS), which is responsible for loading a suitable mission workload for the application harts.

15.1 No-Boot Overview

PIC64GX No-Boot mode is used when debugging embedded software where code should not be executed on power up. It puts the PIC64GX in a mode where all harts execute a loop waiting for the debugger to connect through JTAG or for the device to be programmed.

Note that devices set to No-Boot mode will not have the HSS running on the E51 Monitor hart to perform DDR training.

15.2 No-Boot Mode Sequence

On power up, the Security Controller starts up and holds the PIC64GX in reset until it has completed configuring the device. It executes ROM code which configures the Core Complex based on configuration data structures stored in its private Non-Volatile Memory (pNVM).

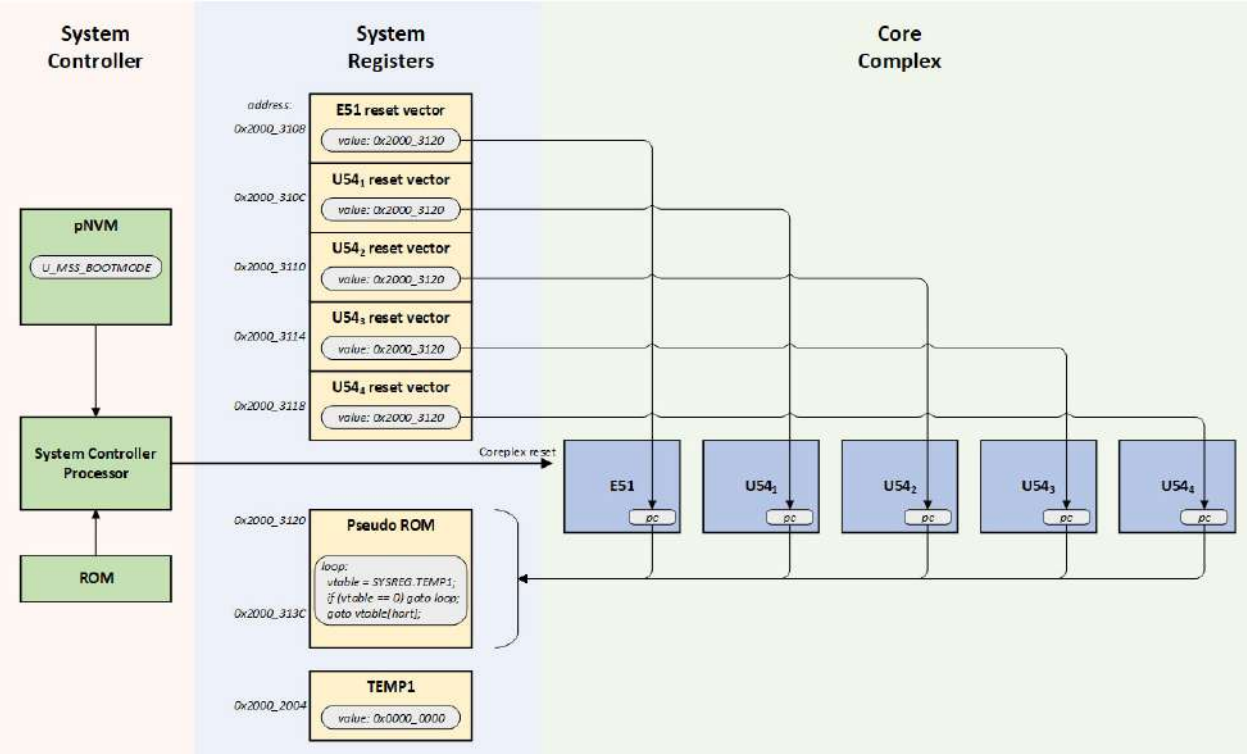
In No-Boot mode, the Security Controller only uses the CORE_COMPLEX_BOOTMODE configuration item stored in pNVM to control the Core Complex boot process: The CORE_COMPLEX_BOOTMODE configuration item determines whether boot mode 0, 1, 2 or 3 is used.

The boot mode 0 sequence is as follows:

1. Registers have the following reset values:
   - Pseudo BOOTROM system registers is a code loop.
   - System registers' reset vectors point to the base address of the pseudo BOOTROM
   - TEMP1 system register is zero.
2. The Security Controller releases the Core Complex reset causing all harts to execute the code found in the pseudo BOOTROM system registers.
   - The loop will keep executing until the content of the TEMP1 system register remains set to zero.

© 2024 Microchip Technology Inc. and its subsidiaries

Figure 15-1. Boot Mode 0 Sequence



The Core Complex harts will remain executing the pseudo BOOTROM loop until the debugger sets the hart's program counters to new values.

Notes:

- The Security Controller's pNVM content can only be modified through a programming bitstream. Neither pNVM nor sNVM content are directly accessible from the Core Complex.
- The Core Complex default clock configuration is 80MHz using the SCB clock source.

15.3 Standard Boot Overview

Standard Boot Mode is used where the PIC64GX harts start executing non-secured code from eNVM on power up.

15.4 Standard Boot Mode Sequence

On power up, the Security Controller starts up and holds the PIC64GX in reset until it has completed configuring the device. It executes ROM code which configures the Core Complex based on configuration data structures stored in its private Non-Volatile Memory (pNVM).

In Standard Boot mode, the Security Controller uses:

- CORE_COMPLEX_BOOTMODE: determines which boot mode (No-Boot, Standard Boot or Secure Boot) is used.
- CORE_COMPLEX_BOOTCFG: contains boot configuration specific to the requested boot mode. In the case of Standard Boot Mode, it contains the reset vectors for all 5 harts.

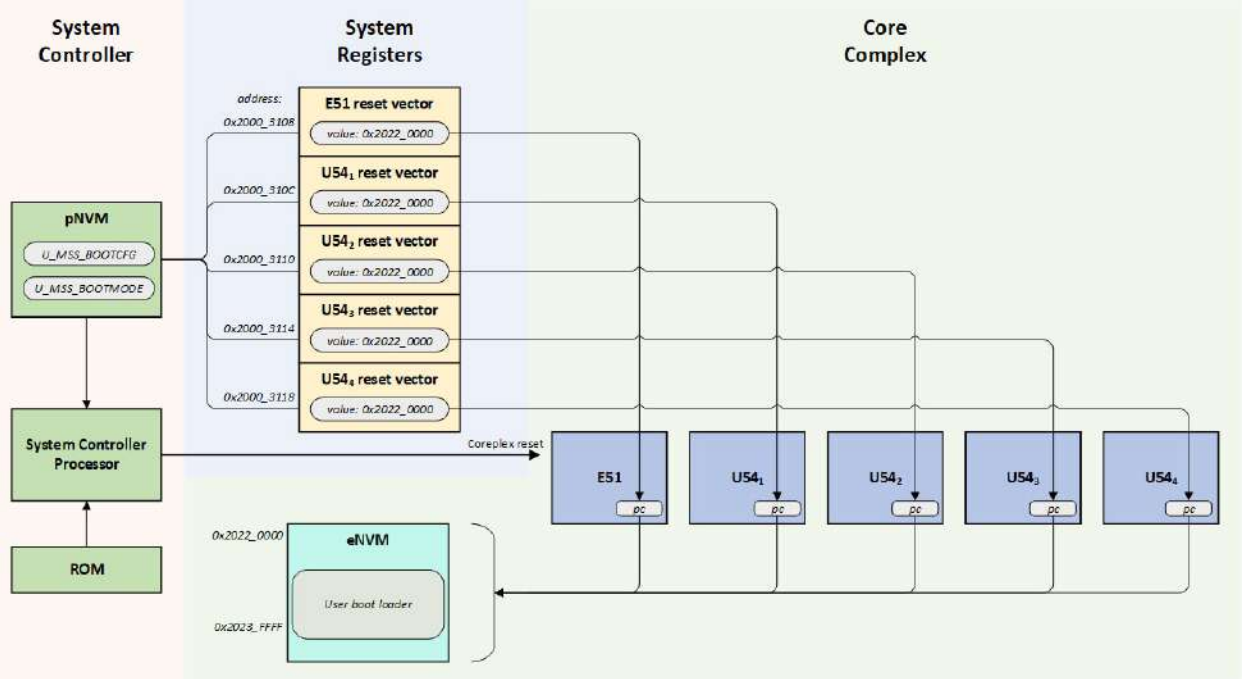
For Standard Boot mode, the Security Controller:

1. Reads the value of CORE_COMPLEX_BOOTMODE and proceeds to the following step if boot mode is 1.

© 2024 Microchip Technology Inc. and its subsidiaries

2. Sets the content of the System Registers' reset vector register from the values found in CORE_COMPLEX_BOOTCFG configuration data structure held in pNVM.
3. Releases the Core Complex reset causing all harts to execute the code found in eNVM.

Figure 15-2. Standard Boot Mode Sequence



Notes:

- The Security Controller's pNVM content can only be modified through a programming bitstream. The pNVM content is not directly accessible from the Core Complex.
- The Core Complex default clock configuration is 80 MHz using the SCB clock source.

15.5 Secure Boot Overview

Secure Boot mode implements Microchip supplied factory secure boot authentication of the eNVM content. It uses the Elliptic Curve Digital Signature Algorithm (ECDSA) to authenticate the signature of a Secure Boot Image Certificate (SBIC) as part of booting the system. The PIC64GX E51 monitor processor and U54 application processors will not be started if authentication fails.

Secure Boot mode only supports authentication of the eNVM content. No encryption/decryption of the eNVM content is used.

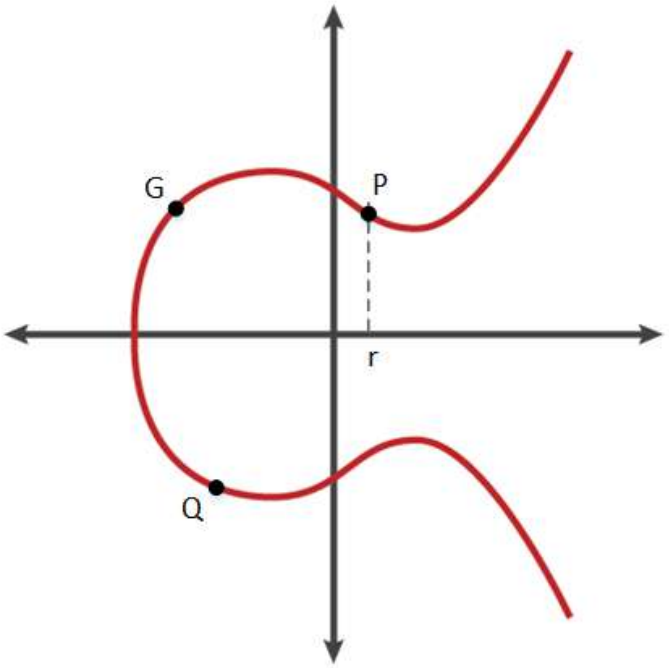
15.6 Elliptic Curve Digital Signature Algorithm (ECDSA) Refresher

Attention: This is intended to be a very high-level algorithm overview of ECDSA. It glosses over some of the finer details and conveniently ignores some of the mathematical properties required of the various actors in this play.

The Elliptic Curve Digital Signature Algorithm (ECDSA) is used to sign and authenticate certificates. It uses elliptic curve point multiplication by a scalar number as a one-way function, which is in practice impossible to reverse, to generate asymmetric key pairs and authenticate messages. The following diagram illustrates how difficult it would be to retrieve the scalar value used in the point multiplication to generate points P and Q from base point G. Such a scalar value is used, among other things, as a private key in the ECDSA.

© 2024 Microchip Technology Inc. and its subsidiaries

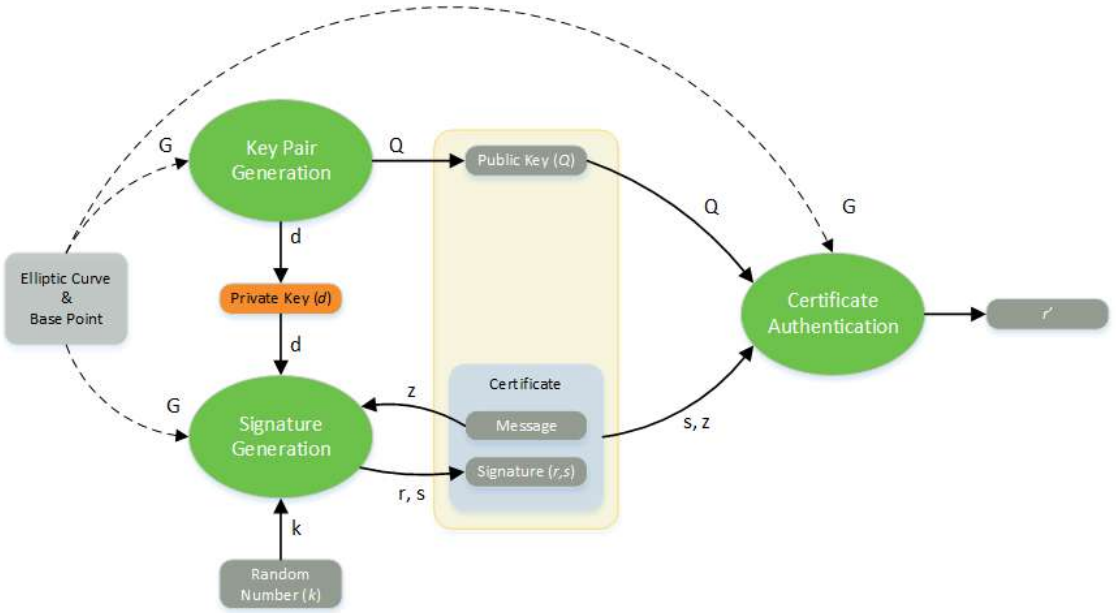
Figure 15-3. TBD Scalar Value Retrieval Example



ECDSA is used to generate a certificate containing a signature for a message using a private key. The associated public key is then used to authenticate the content of the certificate to check that the message has not been tampered with.

All ECDSA steps are performed using a publicly agreed elliptic curve and base point (G) on that curve suitable for this purpose. The base point (G) is used to generate other points on the curve through point multiplication to generate keys and check signatures.

Figure 15-4. TBD ECDSA Sequence



© 2024 Microchip Technology Inc. and its subsidiaries

Parameter	Description
curve	Agreed elliptic curve used for signing and authenticating messages.
G	Base point on the curve. Used to generate other points on the curve though point multiplication. The value of the base point is known to both the signing and authenticating sides.
d	Private key. This is a large integer used in elliptic curve scalar point multiplications. The private key is only known by the signing side.
Q	Public key. This is a point on the elliptic curve. It is known to both signing and authenticating sides.
k	Secret random scalar number (greater than zero) used to generate the certificate's signature. The value of this random number is only known by the signing side.
z	Hash of the message being signed.
r	Part of the signature included in the certificate.
s	Part of the signature included in the certificate used to reconstruct r'.
r'	Value computed from the message's hash during authentication. Should match the signature's r value for authentication to be successful.

Note: The NIST P-384 curve is used for Secure Boot mode ECDSA.

15.6.1 Key Pair Generation

***Note about mathematical notation**: This document uses "." for scalar multiplication and "*" for elliptic curve point multiplication.*

The signing side generates a private/public key pair by randomly selecting a number (d) within the order of the agreed upon elliptic curve. The private key (d) is then used in an elliptic curve point multiplication with the base point (G) to produce the public key (Q): Q = d * G. The private key is a scalar number. The public key is a point on the curve.

15.6.2 Signature Generation

The ECDSA signature is generated using a secret random number (k). This random number is only known by the signing side. It must be changed every time a new signature is generated to prevent an attacker from retrieving sufficient data to reconstruct the signing private key.

The ECDSA signature is made up of two scalar values (integers): (r) and (s). The value of (r) is the x-axis of the point on the curve computed by point multiplication of the base point (G) by the secret random number (k):

$(x, y) = k*G$

$r = x$

The (s) part of the signature is computed using the hash (z) of the message to sign, the private key (d) and the secret random number (k):

z = hash of message

$s = (z + r.d)\ k^{-1}$

The (s) part of the signature is designed such that it can be used to reconstruct the value of (r) using the public key (Q) and the hash of the message (z).

15.6.3 Certificate Authentication

The certificate authentication is performed using the agreed curve and base point (G) by computing the signature check value (r') from the hash of the message (z), the (s) part of the signature and the public key (Q) using the following equations:

$u_1 = z.s^{-1}$

$u_2 = r.s^{-1}$

© 2024 Microchip Technology Inc. and its subsidiaries

$(x, y) = (u_1 * G) + (u_2 * Q)$

$r' = x$

The authentication is successful if the computed value (r') matches the (r) value of the certificate's signature.

The magic of ECDSA is the ability of the authenticating side to recompute the same point (P) on the curve using the hash of the message (z), the signature (s) and the public key as the signing side using the random number (k) and the curve's base point (G).

$(x, y) = (u_1 * G) + (u_2 * Q) = k * G$

Please refer to the ECDSA Wikipedia page for a more detailed explanation of the correctness of the algorithm.

15.7 Checking The Secure Boot Image Certificate

Secure boot mode uses the Elliptic Curve Digital Signature Algorithm (ECDSA) to sign a Secure Boot Image Certificate (SBIC). The SBIC is stored in eNVM alongside the executable being booted. The SBIC is authenticated at system boot time. The boot process is stopped if the SBIC authentication fails.

Two optional checks can be done by the Security Controller before authenticating the SBIC:

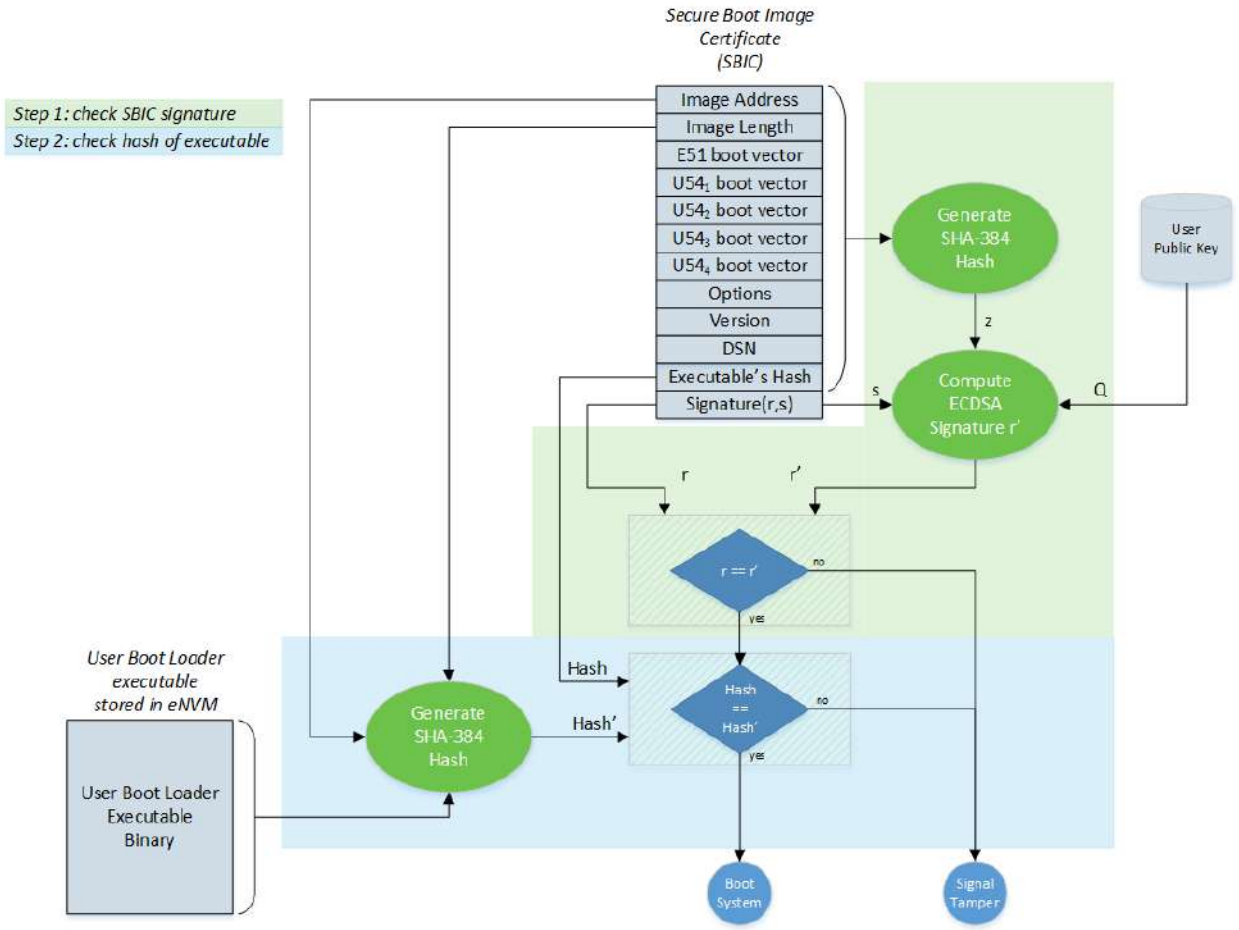
- An optional check of the Device Serial Number (DSN) is done if the SBIC's DSN field is non-zero. This option can be used to bind the SBIC to a specific individual PIC64GX device.
- An optional certificate revocation check is done if the SBIC's Version field is non-zero. The value of the SBIC's Version field is compared against a revocation threshold value. The system will only boot if the SBIC's Version is greater or equal to the revocation threshold. The revocation threshold is programmed as part of a bitstream. This option can prevent old valid certificates from being used.

The actual ECDSA signature authentication is orchestrated by the Security Controller if the above optional checks were successful. The authenticxation is done in two steps:

- The Security Controller verifies the SBIC signature using the ECDSA algorithm. It uses the (s) part of the signature, the user private key and the hash of the SBIC content to compute (r'). The SBIC is authenticated if the computed (r') value is equal to the (r) part of the SBIC's signature.
- If the SBIC signature authentication is successful, the hash of the User Boot Loader is computed and compared against the hash contained in the SBIC. The computed hash matching the SBIC's hash field indicates that the User Boot Loader has not been tampered with and can be executed.

© 2024 Microchip Technology Inc. and its subsidiaries

Figure 15-5. Secure Boot Image Certificate Check



The Security Controller causes the RISC-V harts to jump the addresses defined in the SBIC's boot vector fields when authentication is successful. The User Boot Loader is not executed if any authentication step fails.

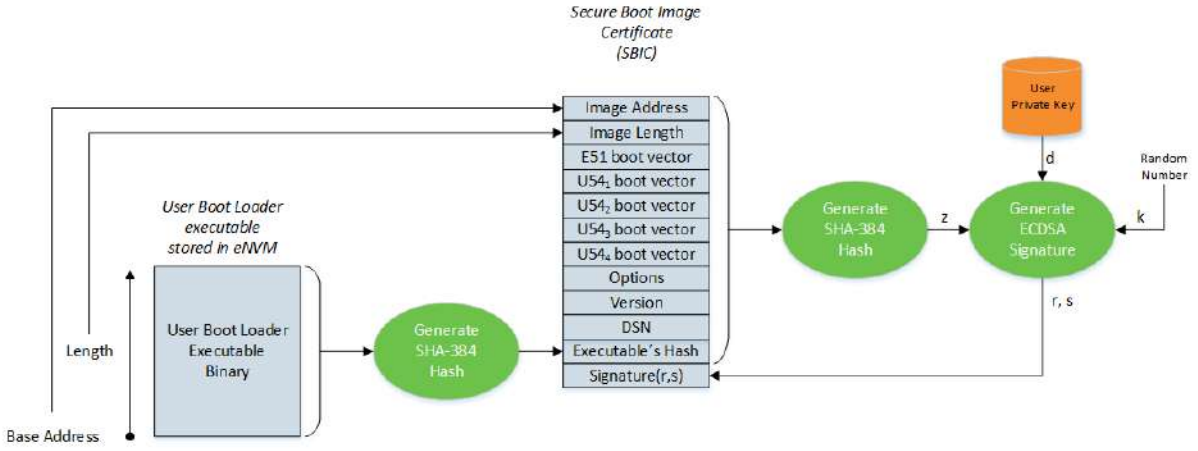
15.8 Generating the Secure Boot Image Certificate

The Secure Boot Image Certificate (SBIC) is constructed to contain information allowing to authenticate a User Boot Loader (UBL) located in eNVM. The SBIC contains the address in eNVM and length in bytes of the of the User Boot Loader alongside the hash value of the UBL's binary.

The SBIC also contains the boot vector addresses from which each hart will execute upon successful authentication of the certificate. It also contains options to bind itself to an individual PIC64GX device using the Device Serial Number (DSN), and the option to revoke the SBIC based on the SBIC's Version field.

© 2024 Microchip Technology Inc. and its subsidiaries

Figure 15-6. Generating the Secure Boot Image Certificate



Two distinct SHA-384 hash values are used to authenticate the User Boot Loader:

- The SBIC includes a SHA-384 hash of the User Boot Loader executable binary contained in eNVM.
- A SHA-384 hash (z) of the SBIC content, except the signature, is used to sign the certificate.

The signature is generated from the SBIC's hash (z), the user private key and a random number using the ECDSA algorithm. The generated signature is made up of two parts (r) and (s). Signature part (s) is used during the signature check to recompute the (r) part of the signature using the hash (z) of the SBIC content and the public key (Q). Authentication succeeds if the recomputed (r') matches the (r) part of the SBIC's signature.

Note: Private key management is not covered by this document.

© 2024 Microchip Technology Inc. and its subsidiaries

16. Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Table 16-1. Revision Histroy

Revision	Date	Description
A	07/2024	Initial revision

© 2024 Microchip Technology Inc. and its subsidiaries

Microchip Information

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

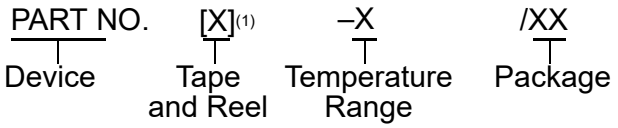
Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

© 2024 Microchip Technology Inc. and its subsidiaries

Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Device:	Device A, Device B, etc	
Tape and Reel Option:	Blank	= Standard packaging (tube or tray)
	T	= Tape and Reel(1)
Temperature Range:	I	= -40°C to +85°C (Industrial)
	E	= -40°C to +125°C (Extended)
Package:(2)	JQ	= UQFN
	P	= PDIP
	ST	= TSSOP
	SL	= SOIC-14
	SN	= SOIC-8
	RF	= UDFN
Pattern:	QTP, SQTP℠ (Serial Quick Turn Programming capability), Code or Special Requirements (blank otherwise)	

- Device A - I/P Industrial temperature, PDIP package
- Device B - E/SS Extended temperature, SSOP package

PIS_NOTES

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED

© 2024 Microchip Technology Inc. and its subsidiaries

WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, TimeCesium, TimeHub, TimePictra, TimeProvider, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, EyeOpen, GridTime, IdealBridge, IGaT, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, MarginLink, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mSiC, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, Power MOS IV, Power MOS 7, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, Trusted Time, TSHARC, Turing, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2024, Microchip Technology Incorporated and its subsidiaries. All Rights Reserved.

ISBN: 978-1-6683-4870-3

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore,

Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

© 2024 Microchip Technology Inc. and its subsidiaries

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office	**Australia - Sydney**	**India - Bangalore**	**Austria - Wels**
2355 West Chandler Blvd.	Tel: 61-2-9868-6733	Tel: 91-80-3090-4444	Tel: 43-7242-2244-39
Chandler, AZ 85224-6199	**China - Beijing**	**India - New Delhi**	Fax: 43-7242-2244-393
Tel: 480-792-7200	Tel: 86-10-8569-7000	Tel: 91-11-4160-8631	**Denmark - Copenhagen**
Fax: 480-792-7277	**China - Chengdu**	**India - Pune**	Tel: 45-4485-5910
Technical Support:	Tel: 86-28-8665-5511	Tel: 91-20-4121-0141	Fax: 45-4485-2829
www.microchip.com/support	**China - Chongqing**	**Japan - Osaka**	**Finland - Espoo**
Web Address:	Tel: 86-23-8980-9588	Tel: 81-6-6152-7160	Tel: 358-9-4520-820
www.microchip.com	**China - Dongguan**	**Japan - Tokyo**	**France - Paris**
Atlanta	Tel: 86-769-8702-9880	Tel: 81-3-6880- 3770	Tel: 33-1-69-53-63-20
Duluth, GA	**China - Guangzhou**	**Korea - Daegu**	Fax: 33-1-69-30-90-79
Tel: 678-957-9614	Tel: 86-20-8755-8029	Tel: 82-53-744-4301	**Germany - Garching**
Fax: 678-957-1455	**China - Hangzhou**	**Korea - Seoul**	Tel: 49-8931-9700
Austin, TX	Tel: 86-571-8792-8115	Tel: 82-2-554-7200	**Germany - Haan**
Tel: 512-257-3370	**China - Hong Kong SAR**	**Malaysia - Kuala Lumpur**	Tel: 49-2129-3766400
Boston	Tel: 852-2943-5100	Tel: 60-3-7651-7906	**Germany - Heilbronn**
Westborough, MA	**China - Nanjing**	**Malaysia - Penang**	Tel: 49-7131-72400
Tel: 774-760-0087	Tel: 86-25-8473-2460	Tel: 60-4-227-8870	**Germany - Karlsruhe**
Fax: 774-760-0088	**China - Qingdao**	**Philippines - Manila**	Tel: 49-721-625370
Chicago	Tel: 86-532-8502-7355	Tel: 63-2-634-9065	**Germany - Munich**
Itasca, IL	**China - Shanghai**	**Singapore**	Tel: 49-89-627-144-0
Tel: 630-285-0071	Tel: 86-21-3326-8000	Tel: 65-6334-8870	Fax: 49-89-627-144-44
Fax: 630-285-0075	**China - Shenyang**	**Taiwan - Hsin Chu**	**Germany - Rosenheim**
Dallas	Tel: 86-24-2334-2829	Tel: 886-3-577-8366	Tel: 49-8031-354-560
Addison, TX	**China - Shenzhen**	**Taiwan - Kaohsiung**	**Israel - Hod Hasharon**
Tel: 972-818-7423	Tel: 86-755-8864-2200	Tel: 886-7-213-7830	Tel: 972-9-775-5100
Fax: 972-818-2924	**China - Suzhou**	**Taiwan - Taipei**	**Italy - Milan**
Detroit	Tel: 86-186-6233-1526	Tel: 886-2-2508-8600	Tel: 39-0331-742611
Novi, MI	**China - Wuhan**	**Thailand - Bangkok**	Fax: 39-0331-466781
Tel: 248-848-4000	Tel: 86-27-5980-5300	Tel: 66-2-694-1351	**Italy - Padova**
Houston, TX	**China - Xian**	**Vietnam - Ho Chi Minh**	Tel: 39-049-7625286
Tel: 281-894-5983	Tel: 86-29-8833-7252	Tel: 84-28-5448-2100	**Netherlands - Drunen**
Indianapolis	**China - Xiamen**		Tel: 31-416-690399
Noblesville, IN	Tel: 86-592-2388138		Fax: 31-416-690340
Tel: 317-773-8323	**China - Zhuhai**		**Norway - Trondheim**
Fax: 317-773-5453	Tel: 86-756-3210040		Tel: 47-72884388
Tel: 317-536-2380			**Poland - Warsaw**
Los Angeles			Tel: 48-22-3325737
Mission Viejo, CA			**Romania - Bucharest**
Tel: 949-462-9523			Tel: 40-21-407-87-50
Fax: 949-462-9608			**Spain - Madrid**
Tel: 951-273-7800			Tel: 34-91-708-08-90
Raleigh, NC			Fax: 34-91-708-08-91
Tel: 919-844-7510			**Sweden - Gothenberg**
New York, NY			Tel: 46-31-704-60-40
Tel: 631-435-6000			**Sweden - Stockholm**
San Jose, CA			Tel: 46-8-5090-4654
Tel: 408-735-9110			**UK - Wokingham**
Tel: 408-436-4270			Tel: 44-118-921-5800
Canada - Toronto			Fax: 44-118-921-5820
Tel: 905-695-1980			
Fax: 905-695-2078			

© 2024 Microchip Technology Inc. and its subsidiaries