**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | e200z3 |
| Core Size | 32-Bit Single-Core |
| Speed | 64MHz |
| Connectivity | CANbus, EBI/EMI, LINbus, SCI, SPI, UART/USART |
| Peripherals | DMA, POR, PWM, WDT |
| Number of I/O | 80 |
| Program Memory Size | 1.5MB (1.5M x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 94K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.14V ~ 1.32V |
| Data Converters | A/D 34x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 176-LQFP |
| Supplier Device Package | 176-LQFP (24x24) |
| Purchase URL | https://www.e-xfl.com/product-detail/stmicroelectronics/spc563m64l7coby |

The SPC563Mxx has a single level of memory hierarchy consisting of up to 94 KB on-chip SRAM and up to 1.5 MB of internal Flash memory. The SPC563Mxx also has an external bus interface (EBI) for 'calibration'[(b)].

On-chip modules include:

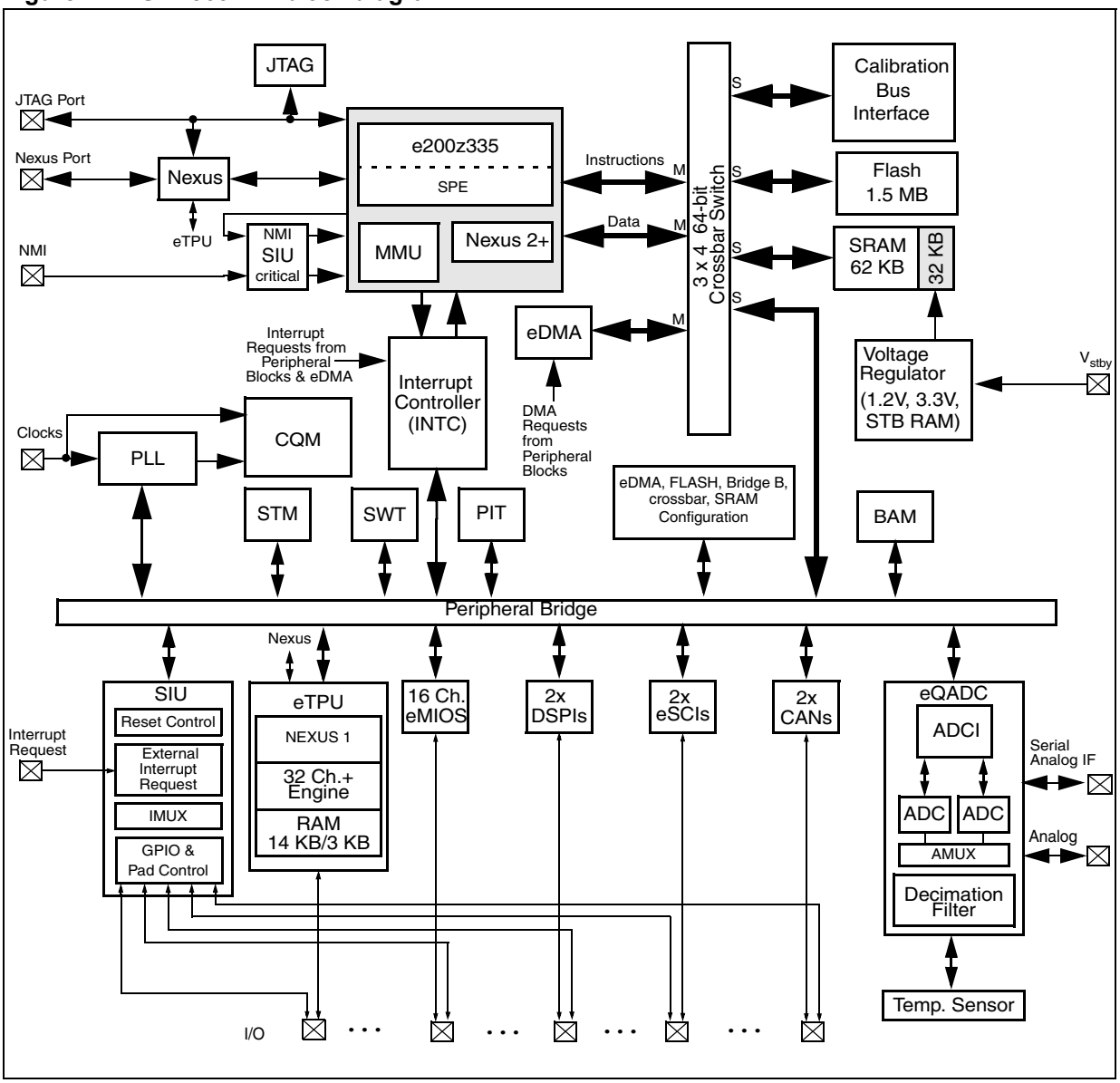
- Single issue, 32-bit Power Architecture technology compliant e200z335 CPU core complex
  - Includes Variable Length Encoding (VLE) enhancements for code size reduction
  - Floating Point Unit (FPU)
- 32-channel direct memory access controller (DMA)
- Interrupt controller (INTC) capable of handling 364 selectable-priority interrupt sources—191 peripheral interrupt sources, 8 software interrupts and 165 reserved interrupts.
- Frequency-modulated phase-locked loop (FMPLL)
- Calibration external bus interface (EBI)[b]
- System integration unit (SIU)
- Up to 1.5 MB on-chip flash with flash controller
  - Fetch Accelerator for single cycle flash access @80 MHz
- Up to 94 KB on-chip static RAM (including up to 32 KB standby RAM)
- Boot assist module (BAM)
- 32-channel second-generation enhanced time processor unit (eTPU2)
  - 32 standard eTPU channels
  - Architectural enhancements to improve code efficiency and added flexibility
- 16-channels enhanced modular input-output system (eMIOS)
- Enhanced queued analog-to-digital converter (eQADC)
- Decimation filter (part of eQADC)
- Silicon die temperature sensor
- Two deserial serial peripheral interface (DSPI) modules (compatible with Microsecond Channel)
- Two enhanced serial communication interface (eSCI) modules compatible with LIN
- Two Controller Area Network (FlexCAN) modules that support CAN 2.0B
- Nexus port controller (NPC) per IEEE-ISTO 5001-2003 standard
- IEEE 1149.1 (JTAG) support
- Nexus interface
- On-chip voltage regulator controller that provides 1.2 V and 3.3 V internal supplies from a 5 V external source
- Designed for LQFP100, LQFP144, LQFP176, and LBGA208 packages

b. The external bus interface is only accessible when using the calibration tool. It is not available on production packages.

# 2 Block diagram

*Figure 1* shows a top-level block diagram of the SPC563M64.

**Figure 1. SPC563Mxx block diagram**



# 3 Device overview

The following sections provide high level descriptions of the features found on the SPC563Mxx.

## 3.1 Device comparison

**Table 2. SPC563Mxx family device summary**

| Feature | SPC563M64 | SPC563M60P | SPC563M54P |
|---|---|---|---|
| Flash memory size (KB) | 1536 | 1024 | 768 |
| Total SRAM size (KB) | 94 | 64 | 48 |
| Standby SRAM size (KB) | 32 | 32 | 24 |
| Processor core | 32-bit e200z335 with SPE and FPU support | 32-bit e200z335 with SPE and FPU support | 32-bit e200z335 with SPE and FPU support |
| Core frequency (MHz) | 64/80 | 40/64/80 | 40/64 |
| Calibration bus width(1) | 16 bits | 16 bits | — |
| DMA (direct memory access) channels | 32 | 32 | 32 |
| eMIOS (enhanced modular input-output system) channels | 16 | 16 | 8 |
| eQADC (enhanced queued analog-to-digital converter) channels (on-chip) | Up to 34(2) | Up to 34(2) | Up to 32(2) |
| eSCI (serial communication interface) | 2 | 2 | 2 |
| DSPI (deserial serial peripheral interface) | 2 | 2 | 2 |
| Microsecond Channel compatible interface | 2 | 2 | 2 |
| eTPU (enhanced time processor unit) | Yes | Yes | Yes |
| Channels | 32 | 32 | 32 |
| Code memory (KB) | 14 | 14 | 14 |
| Parameter RAM (KB) | 3 | 3 | 3 |
| FlexCAN (controller area network)(3) | 2 | 2 | 2 |
| FMPLL (frequency-modulated phase-locked loop) | Yes | Yes | Yes |
| INTC (interrupt controller) channels | 364(4) | 364(4) | 364(4) |
| JTAG controller | Yes | Yes | Yes |
| NDI (Nexus development interface) level | Class 2+ | Class 2+ | Class 2+ |
| Non-maskable interrupt and critical interrupt | Yes | Yes | Yes |
| PIT (peripheral interrupt timers) | 5 | 5 | 5 |
| Task monitor timer | 4 channels | 4 channels | 4 channels |
| Temperature sensor | Yes | Yes | Yes |

## 3.2 Feature list

- Operating parameters
  - Fully static operation, 0 MHz – 80 MHz (plus 2% frequency modulation - 82 MHz)
  - –40 °C to 150 °C junction temperature operating range
  - Low power design
    - Less than 400 mW power dissipation (nominal)
    - Designed for dynamic power management of core and peripherals
    - Software controlled clock gating of peripherals
    - Low power stop mode, with all clocks stopped
  - Fabricated in 90 nm process
  - 1.2 V internal logic
  - Single power supply with 5.0 V –10% / +5% (4.5 V to 5.25 V) with internal regulator to provide 3.3 V and 1.2 V for the core
  - Input and output pins with 5.0 V –10% / +5% (4.5 V to 5.25 V) range
    - 35%/65% $V_{DDE}$ CMOS switch levels (with hysteresis)
    - Selectable hysteresis
    - Selectable slew rate control
  - Nexus pins powered by 3.3 V supply
  - Designed with EMI reduction techniques
    - Phase-locked loop
    - Frequency modulation of system clock frequency
    - On-chip bypass capacitance
    - Selectable slew rate and drive strength
- High performance e200z335 core processor
  - 32-bit *Power Architecture Book E* programmer's model
  - Variable Length Encoding Enhancements
    - Allows Power Architecture instruction set to be optionally encoded in a mixed 16 and 32-bit instructions
    - Results in smaller code size
  - Single issue, 32-bit *Power Architecture technology* compliant CPU
  - In-order execution and retirement
  - Precise exception handling
  - Branch processing unit
    - Dedicated branch address calculation adder
    - Branch acceleration using Branch Lookahead Instruction Buffer
  - Load/store unit
    - One-cycle load latency
    - Fully pipelined
    - Big and Little Endian support
    - Misaligned access support
    - Zero load-to-use pipeline bubbles
  - Thirty-two 64-bit general purpose registers (GPRs)

  - Memory management unit (MMU) with 16-entry fully-associative translation look-aside buffer (TLB)
  - Separate instruction bus and load/store bus
  - Vectored interrupt support
  - Interrupt latency < 120 ns @ 80 MHz (measured from interrupt request to execution of first instruction of interrupt exception handler)
  - Non-maskable interrupt (NMI) input for handling external events that must produce an immediate response, for example, power down detection. On this device, the NMI input is connected to the Critical Interrupt Input. (May not be recoverable)
  - Critical Interrupt Input. For external interrupt sources that are higher priority than provided by the Interrupt Controller. (Always recoverable)
  - New 'Wait for Interrupt' instruction, to be used with new low power modes
  - Reservation instructions for implementing read-modify-write accesses
  - Signal processing extension (SPE) APU

    Operating on all 32 GPRs that are all extended to 64 bits wide

    Provides a full compliment of vector and scalar integer and floating point arithmetic operations (including integer vector MAC and MUL operations) (SIMD)

    Provides rich array of extended 64-bit loads and stores to/from extended GPRs

    Fully code compatible with e200z6 core
  - Floating point (FPU)

    IEEE 754 compatible with software wrapper

    Scalar single precision in hardware, double precision with software library

    Conversion instructions between single precision floating point and fixed point

    Fully code compatible with e200z6 core
  - Long cycle time instructions, except for guarded loads, do not increase interrupt latency
  - Extensive system development support through Nexus debug port
- Advanced microcontroller bus architecture (AMBA) crossbar switch (XBAR)
  - Three master ports, four slave ports

    Masters: CPU Instruction bus; CPU Load/store bus (Nexus); eDMA

    Slave: Flash; SRAM; Peripheral Bridge; calibration EBI
  - 32-bit internal address bus, 64-bit internal data bus
- Enhanced direct memory access (eDMA) controller
  - 32 channels support independent 8-bit, 16-bit, or 32-bit single value or block transfers
  - Supports variable sized queues and circular queues
  - Source and destination address registers are independently configured to post-increment or remain constant
  - Each transfer is initiated by a peripheral, CPU, or eDMA channel request
  - Each eDMA channel can optionally send an interrupt request to the CPU on completion of a single value or block transfer
- Interrupt controller (INTC)
  - 191 peripheral interrupt request sources
  - 8 software setable interrupt request sources

  - 9-bit vector
    Unique vector for each interrupt request source
    Provided by hardware connection to processor or read from register
  - Each interrupt source can be programmed to one of 16 priorities
  - Preemption
    Preemptive prioritized interrupt requests to processor
    ISR at a higher priority preempts ISRs or tasks at lower priorities
    Automatic pushing or popping of preempted priority to or from a LIFO
    Ability to modify the ISR or task priority. Modifying the priority can be used to implement the Priority Ceiling Protocol for accessing shared resources.
  - Low latency—three clocks from receipt of interrupt request from peripheral to interrupt request to processor
- Frequency Modulating Phase-locked loop (FMPLL)
  - Reference clock pre-divider (PREDIV) for finer frequency synthesis resolution
  - Reduced frequency divider (RFD) for reducing the FMPLL output clock frequency without forcing the FMPLL to re-lock
  - System clock divider (SYSDIV) for reducing the system clock frequency in normal or bypass mode
  - Input clock frequency range from 4 MHz to 20 MHz before the pre-divider, and from 4 MHz to 16 MHz at the FMPLL input
  - Voltage controlled oscillator (VCO) range from 256 MHz to 512 MHz
  - VCO free-running frequency range from 25 MHz to 125 MHz
  - Four bypass modes: crystal or external reference with PLL on or off
  - Two normal modes: crystal or external reference
  - Programmable frequency modulation
    Triangle wave modulation
    Register programmable modulation frequency and depth
  - Lock detect circuitry reports when the FMPLL has achieved frequency lock and continuously monitors lock status to report loss of lock conditions
    User-selectable ability to generate an interrupt request upon loss of lock
    User-selectable ability to generate a system reset upon loss of lock
  - Clock quality monitor (CQM) module provides loss-of-clock detection for the FMPLL reference and output clocks
    User-selectable ability to generate an interrupt request upon loss of clock
    User-selectable ability to generate a system reset upon loss of clock
    Backup clock (reference clock or FMPLL free-running) can be applied to the system in case of loss of clock
- Calibration bus interface (EBI)
  - Available only in the calibration package (496 CSP package)
  - 1.8 V to 3.3 V ± 10% I/O (1.6 V to 3.6 V)
  - Memory controller with support for various memory types
  - 16-bit data bus, up to 22-bit address bus
  - Selectable drive strength

  - Configurable bus speed modes
  - Bus monitor
  - Configurable wait states
- System integration unit (SIU)
  - Centralized GPIO control of 80 I/O pins
  - Centralized pad control on a per-pin basis
    Pin function selection
    Configurable weak pull-up or pull-down
    Drive strength
    Slew rate
    Hysteresis
  - System reset monitoring and generation
  - External interrupt inputs, filtering and control
  - Critical Interrupt control
  - Non-Maskable Interrupt control
  - Internal multiplexer subblock (IMUX)
    Allows flexible selection of eQADC trigger inputs (eTPU, eMIOS and external signals)
    Allows selection of interrupt requests between external pins and DSPI
- Error correction status module (ECSM)
  - Configurable error-correcting codes (ECC) reporting
  - Single-bit error correction reporting
- On-chip Flash memory
  - Up to 1.5 MB flash memory, accessed via a 64-bit wide bus interface
  - 16 KB shadow block
  - Fetch Accelerator
    Provide single cycle flash access at 80 MHz
    Quadruple 128-bit wide prefetch/burst buffers
    Prefetch buffers can be configured to prefetch code or data or both
  - Censorship protection scheme to prevent flash content visibility
  - Flash divided into two independent arrays, allowing reading from one array while erasing/programming the other array (used for EEPROM emulation)
  - Memory block:
    For SPC563M64: 18 blocks ($4 \times 16$ KB, $2 \times 32$ KB, $2 \times 64$ KB, $10 \times 128$ KB)
    For SPC563M60P: 14 blocks ($4 \times 16$ KB, $2 \times 32$ KB, $2 \times 64$ KB, $6 \times 128$ KB)
    For SPC563M54P: 12 blocks ($4 \times 16$ KB, $2 \times 32$ KB, $2 \times 64$ KB, $4 \times 128$ KB)
  - Hardware programming state machine
- On-chip static RAM
  - For SPC563M64: 94 KB general purpose RAM of which 32 KB are on standby power supply
  - For SPC563M60P: 64 KB general purpose RAM of which 32 KB are on standby power supply
  - For SPC563M54P: 48 KB general purpose RAM of which 24 KB are on standby

## 3.3 Feature details

### 3.3.1 e200z335 core

The e200z335 processor utilizes a four stage pipeline for instruction execution. The Instruction Fetch (stage 1), Instruction Decode/Register file Read/Effective Address Calculation (stage 2), Execute/Memory Access (stage 3), and Register Writeback (stage 4) stages operate in an overlapped fashion, allowing single clock instruction execution for most instructions.

The integer execution unit consists of a 32-bit Arithmetic Unit (AU), a Logic Unit (LU), a 32-bit Barrel shifter (Shifter), a Mask-Insertion Unit (MIU), a Condition Register manipulation Unit (CRU), a Count-Leading-Zeros unit (CLZ), a 32×32 Hardware Multiplier array, result feed-forward hardware, and support hardware for division.

Most arithmetic and logical operations are executed in a single cycle with the exception of the divide instructions. A Count-Leading-Zeros unit operates in a single clock cycle. The Instruction Unit contains a PC incrementer and a dedicated Branch Address adder to minimize delays during change of flow operations. Sequential prefetching is performed to ensure a supply of instructions into the execution pipeline. Branch target prefetching is performed to accelerate taken branches. Prefetched instructions are placed into an instruction buffer capable of holding six instructions.

Branches can also be decoded at the instruction buffer and branch target addresses calculated prior to the branch reaching the instruction decode stage, allowing the branch target to be prefetched early. When a branch is detected at the instruction buffer, a prediction may be made on whether the branch is taken or not. If the branch is predicted to be taken, a target fetch is initiated and its target instructions are placed in the instruction buffer following the branch instruction. Many branches take zero cycle to execute by using branch folding. Branches are folded out from the instruction execution pipe whenever possible. These include unconditional branches and conditional branches with condition codes that can be resolved early.

Conditional branches which are not taken and not folded execute in a single clock. Branches with successful target prefetching which are not folded have an effective execution time of one clock. All other taken branches have an execution time of two clocks. Memory load and store operations are provided for byte, halfword, and word (32-bit) data with automatic zero or sign extension of byte and halfword load data as well as optional byte reversal of data. These instructions can be pipelined to allow effective single cycle throughput. Load and store multiple word instructions allow low overhead context save and restore operations. The load/store unit contains a dedicated effective address adder to allow effective address generation to be optimized. Also, a load-to-use dependency does not incur any pipeline bubbles for most cases.

The Condition Register unit supports the condition register (CR) and condition register operations defined by the Power Architecture. The condition register consists of eight 4-bit fields that reflect the results of certain operations, such as move, integer and floating-point compare, arithmetic, and logical instructions, and provide a mechanism for testing and branching. Vectored and autovectored interrupts are supported by the CPU. Vectored interrupt support is provided to allow multiple interrupt sources to have unique interrupt handlers invoked with no software overhead.

The hardware floating-point unit utilizes the IEEE-754 single-precision floating-point format and supports single-precision floating-point operations in a pipelined fashion. The general purpose register file is used for source and destination operands, thus there is a unified

storage model for single-precision floating-point data types of 32 bits and the normal integer type. Single-cycle floating-point add, subtract, multiply, compare, and conversion operations are provided. Divide instructions are multi-cycle and are not pipelined.

The Signal Processing Extension (SPE) Auxiliary Processing Unit (APU) provides hardware SIMD operations and supports a full complement of dual integer arithmetic operation including Multiply Accumulate (MAC) and dual integer multiply (MUL) in a pipelined fashion. The general purpose register file is enhanced such that all 32 of the GPRs are extended to 64 bits wide and are used for source and destination operands, thus there is a unified storage model for 32×32 MAC operations which generate greater than 32-bit results.

The majority of both scalar and vector operations (including MAC and MUL) are executed in a single clock cycle. Both scalar and vector divides take multiple clocks. The SPE APU also provides extended load and store operations to support the transfer of data to and from the extended 64-bit GPRs.

The CPU includes support for Variable Length Encoding (VLE) instruction enhancements. This enables the classic Power Architecture instruction set to be represented by a modified instruction set made up from a mixture of 16- and 32-bit instructions. This results in a significantly smaller code size footprint without noticeably affecting performance. The Power Architecture instruction set and VLE instruction set are available concurrently. Regions of the memory map are designated as PPC or VLE using an additional configuration bit in each of Table Look-aside Buffers (TLB) entries in the MMU.

The CPU core is enhanced by the addition of two additional interrupt sources; Non-Maskable Interrupt and Critical Interrupt. These two sources are routed directly from package pins, via edge detection logic in the SIU to the CPU, bypassing completely the Interrupt Controller. Once the edge detection logic is programmed, it cannot be disabled, except by reset. The non-maskable Interrupt is, as the name suggests, completely un-maskable and when asserted will always result in the immediate execution of the respective interrupt service routine. The non-maskable interrupt is not guaranteed to be recoverable. The Critical Interrupt is very similar to the non-maskable interrupt, but it can be masked by other exceptional interrupts in the CPU and is guaranteed to be recoverable (code execution may be resumed from where it stopped).

The CPU core has an additional ‘Wait for Interrupt’ instruction that is used in conjunction with low power STOP mode. When Low Power Stop mode is selected, this instruction is executed to allow the system clock to be stopped. An external interrupt source or the system wake-up timer is used to restart the system clock and allow the CPU to service the interrupt.

### 3.3.2 Crossbar

The XBAR multi-port crossbar switch supports simultaneous connections between three master ports and four slave ports. The crossbar supports a 32-bit address bus width and a 64-bit data bus width.

The crossbar allows three concurrent transactions to occur from the master ports to any slave port; but each master must access a different slave. If a slave port is simultaneously requested by more than one master port, arbitration logic selects the higher priority master and grants it ownership of the slave port. All other masters requesting that slave port are stalled until the higher priority master completes its transactions. Requesting masters are treated with equal priority and are granted access to a slave port in round-robin fashion,

based upon the ID of the last master to be granted access. The crossbar provides the following features:

- 3 master ports:
  - e200z335 core complex Instruction port
  - e200z335 core complex Load/Store port
  - eDMA
- 4 slave ports
  - FLASH
  - calibration bus
  - SRAM
  - Peripheral bridge A/B (eTPU2, eMIOS, SIU, DSPI, eSCI, FlexCAN, eQADC, BAM, decimation filter, PIT, STM and SWT)
- 32-bit internal address, 64-bit internal data paths

### 3.3.3 eDMA

The enhanced direct memory access (eDMA) controller is a second-generation module capable of performing complex data movements via 32 programmable channels, with minimal intervention from the host processor. The hardware micro architecture includes a DMA engine which performs source and destination address calculations, and the actual data movement operations, along with an SRAM-based memory containing the transfer control descriptors (TCD) for the channels. This implementation is utilized to minimize the overall block size. The eDMA module provides the following features:

- All data movement via dual-address transfers: read from source, write to destination
- Programmable source and destination addresses, transfer size, plus support for enhanced addressing modes
- Transfer control descriptor organized to support two-deep, nested transfer operations
- An inner data transfer loop defined by a “minor” byte transfer count
- An outer data transfer loop defined by a “major” iteration count
- Channel activation via one of three methods:
  - Explicit software initiation
  - Initiation via a channel-to-channel linking mechanism for continuous transfers
  - Peripheral-paced hardware requests (one per channel)
- Support for fixed-priority and round-robin channel arbitration
- Channel completion reported via optional interrupt requests
- 1 interrupt per channel, optionally asserted at completion of major iteration count
- Error termination interrupts are optionally enabled
- Support for scatter/gather DMA processing
- Channel transfers can be suspended by a higher priority channel

### 3.3.4 Interrupt controller

The INTC (interrupt controller) provides priority-based preemptive scheduling of interrupt requests, suitable for statically scheduled hard real-time systems. The INTC allows interrupt request servicing from up to 191 peripheral interrupt request sources, plus 165 sources reserved for compatibility with other family members).

For high priority interrupt requests, the time from the assertion of the interrupt request from the peripheral to when the processor is executing the interrupt service routine (ISR) has been minimized. The INTC provides a unique vector for each interrupt request source for quick determination of which ISR needs to be executed. It also provides an ample number of priorities so that lower priority ISRs do not delay the execution of higher priority ISRs. To allow the appropriate priorities for each source of interrupt request, the priority of each interrupt request is software configurable.

When multiple tasks share a resource, coherent accesses to that resource need to be supported. The INTC supports the priority ceiling protocol for coherent accesses. By providing a modifiable priority mask, the priority can be raised temporarily so that all tasks which share the resource can not preempt each other.

Multiple processors can assert interrupt requests to each other through software setable interrupt requests. These same software setable interrupt requests also can be used to break the work involved in servicing an interrupt request into a high priority portion and a low priority portion. The high priority portion is initiated by a peripheral interrupt request, but then the ISR asserts a software setable interrupt request to finish the servicing in a lower priority ISR. Therefore these software setable interrupt requests can be used instead of the peripheral ISR scheduling a task through the RTOS.

The INTC provides the following features:

- 356 peripheral interrupt request sources
- 8 software setable interrupt request sources
- 9-bit vector addresses
- Unique vector for each interrupt request source
- Hardware connection to processor or read from register
- Each interrupt source can be programmed to one of 16 priorities
- Preemptive prioritized interrupt requests to processor
- ISR at a higher priority preempts executing ISRs or tasks at lower priorities
- Automatic pushing or popping of preempted priority to or from a LIFO
- Ability to modify the ISR or task priority to implement the priority ceiling protocol for accessing shared resources
- Low latency—three clocks from receipt of interrupt request from peripheral to interrupt request to processor

This device also includes a non-maskable interrupt (NMI) pin that bypasses the INTC and multiplexing logic.

### 3.3.5 FMPLL

The FMPLL allows the user to generate high speed system clocks from a 4 MHz to 20 MHz crystal oscillator or external clock generator. Further, the FMPLL supports programmable frequency modulation of the system clock. The PLL multiplication factor, output clock divider ratio are all software configurable.

*Note: The calibration EBI must be configured in multiplexed mode when the extended Nexus trace is used on the VertiCal Calibration Systemcalibration tool. This is because Nexus signals and address lines of the calibration bus share the same balls in the calibration package.*

- Memory controller with support for various memory types:
  - Asynchronous/legacy flash and SRAM
- Bus monitor
  - User selectable
  - Programmable time-out period (with 8 external bus clock resolution)
- Configurable wait states (via chip selects)
- 3 chip-select (Cal_$\overline{CS}$[0], Cal_$\overline{CS}$[2:3]) signals (Multiplexed with 2 most significant address signals)
- 2 write/byte enable (WE[0:1]/BE[0:1]) signals
- Configurable bus speed modes
  - system frequency
  - 1/2 of system frequency
  - 1/4 of system frequency
- Optional automatic CLKOUT gating to save power and reduce EMI
- Selectable drive strengths; 10 pF, 20 pF, 30 pF, 50 pF

### 3.3.7 SIU

The SPC563Mxx SIU controls MCU reset configuration, pad configuration, external interrupt, general purpose I/O (GPIO), internal peripheral multiplexing, and the system reset operation. The reset configuration block contains the external pin boot configuration logic. The pad configuration block controls the static electrical characteristics of I/O pins. The GPIO block provides uniform and discrete input/output control of the I/O pins of the MCU. The reset controller performs reset monitoring of internal and external reset sources, and

MCU is powered-on or reset in normal mode. The BAM supports different modes of booting. They are:

- Booting from internal Flash memory
- Serial boot loading (A program is downloaded into RAM via eSCI or the FlexCAN and then executed)
- Booting from external memory on calibration bus

The BAM also reads the reset configuration half word (RCHW) from internal flash memory and configures the SPC563Mxx hardware accordingly. The BAM provides the following features:

- Sets up MMU to cover all resources and mapping all physical address to logical addresses with minimum address translation
- Sets up the MMU to allow user boot code to execute as either Power Architecture code (default) or as VLE code
- Detection of user boot code
- Automatic switch to serial boot mode if internal flash is blank or invalid
- Supports user programmable 64-bit password protection for serial boot mode
- Supports serial bootloading via FlexCAN bus and eSCI using fixed baudrate protocol
- Supports serial bootloading via FlexCAN bus and eSCI with auto baud rate sensing
- Supports serial bootloading of either Power Architecture code (default) or VLE code
- Supports booting from calibration bus interface
- Supports censorship protection for internal Flash memory
- Provides an option to enable the core watchdog timer
- Provides an option to disable the system watchdog timer

### 3.3.12 eMIOS

The eMIOS (Enhanced Modular Input Output System) module provides the functionality to generate or measure time events. The channels on this module provide a range of operating modes including the capability to perform dual input capture or dual output compare as well as PWM output.

- Event-triggered microengine:
  - Fixed-length instruction execution in two-system-clock microcycle
  - 14 KB of code memory (SCM)
  - 3 KB of parameter (data) RAM (SPRAM)
  - Parallel execution of data memory, ALU, channel control and flow control sub-instructions in selected combinations
  - 32-bit microengine registers and 24-bit wide ALU, with 1 microcycle addition and subtraction, absolute value, bitwise logical operations on 24-bit, 16-bit, or byte operands, single-bit manipulation, shift operations, sign extension and conditional execution
  - Additional 24-bit Multiply/MAC/Divide unit which supports all signed/unsigned Multiply/MAC combinations, and unsigned 24-bit divide. The MAC/Divide unit works in parallel with the regular microcode commands
- Resource sharing features support channel use of common channel registers, memory and microengine time:
  - Hardware scheduler works as a “task management” unit, dispatching event service routines by predefined, host-configured priority
  - Automatic channel context switch when a “task switch” occurs, i.e., one function thread ends and another begins to service a request from other channel: channel-specific registers, flags and parameter base address are automatically loaded for the next serviced channel
  - SPRAM shared between host CPU and eTPU2, supporting communication either between channels and host or inter-channel
  - Dual-parameter coherency hardware support allows atomic access to two parameters by host
- Test and development support features:
  - Nexus Class 1 debug, supporting single-step execution, arbitrary microinstruction execution, hardware breakpoints and watchpoints on several conditions
  - Software breakpoints
  - SCM continuous signature-check built-in self test (MISC — multiple input signature calculator), runs concurrently with eTPU2 normal operation
- System enhancements
  - Software watchdog with programmable timeout
  - Real-time performance information
- Channel enhancements
  - Channels 1 and 2 can optionally drive angle clock hardware
- Programming enhancements
  - Engine relative addressing mode

### 3.3.14 eQADC

The enhanced queued analog to digital converter (eQADC) block provides accurate and fast conversions for a wide range of applications. The eQADC provides a parallel interface to two on-chip analog to digital converters (ADC), and a single master to single slave serial interface to an off-chip external device. Both on-chip ADCs have access to all the analog channels.

The eQADC prioritizes and transfers commands from six command conversion command ‘queues’ to the on-chip ADCs or to the external device. The block can also receive data from the on-chip ADCs or from an off-chip external device into the six result queues, in parallel, independently of the command queues. The six command queues are prioritized with Queue_0 having the highest priority and Queue_5 the lowest. Queue_0 also has the added ability to bypass all buffering and queuing and abort a currently running conversion on either ADC and start a Queue_0 conversion. This means that Queue_0 will always have a deterministic time from trigger to start of conversion, irrespective of what tasks the ADCs were performing when the trigger occurred. The eQADC supports software and external hardware triggers from other blocks to initiate transfers of commands from the queues to the on-chip ADCs or to the external device. It also monitors the fullness of command queues and result queues, and accordingly generates DMA or interrupt requests to control data movement between the queues and the system memory, which is external to the eQADC.

The ADCs also support features designed to allow the direct connection of high impedance acoustic sensors that might be used in a system for detecting engine knock. These features include differential inputs; integrated variable gain amplifiers for increasing the dynamic range; programmable pull-up and pull-down resistors for biasing and sensor diagnostics.

The eQADC also integrates a programmable decimation filter capable of taking in ADC conversion results at a high rate, passing them through a hardware low pass filter, then down-sampling the output of the filter and feeding the lower sample rate results to the result FIFOs. This allows the ADCs to sample the sensor at a rate high enough to avoid aliasing of out-of-band noise; while providing a reduced sample rate output to minimize the amount DSP processing bandwidth required to fully process the digitized waveform.

bound for the same ADC, the higher priority Queue is always served first
  - Queue_0 can bypass all prioritization, buffering and abort current conversions to start a Queue_0 conversion a deterministic time after the queue trigger
  - Streaming mode operation of Queue_0 to execute some commands several times
  - Supports software and hardware trigger modes to arm a particular Queue
  - Generates interrupt when command coherency is not achieved
- External hardware triggers
  - Supports rising edge, falling edge, high level and low level triggers
  - Supports configurable digital filter
- Supports four external 8-to-1 muxes which can expand the input channels to 56 channels total

### 3.3.15 DSPI

The deserial serial peripheral interface (DSPI) block provides a synchronous serial interface for communication between the SPC563Mxx MCU and external devices. The DSPI supports pin count reduction through serialization and deserialization of eTPU and eMIOS channels and memory-mapped registers. The channels and register content are transmitted using a SPI-like protocol. This SPI-like protocol is completely configurable for baud rate, polarity and phase, frame length, chip select assertion, etc. Each bit in the frame may be configured to serialize either eTPU channels, eMIOS channels or GPIO signals. The DSPI can be configured to serialize data to an external device that supports the Microsecond Channel protocol. There are two identical DSPI blocks on the SPC563Mxx MCU. The DSPI ouput pins support 5 V logic levels or Low Voltage Differential Signalling (LVDS) according to the Microsecond Channel specification.

The DSPIs have three configurations:

- Serial Peripheral Interface (SPI) configuration where the DSPI operates as an up to 16-bit SPI with support for queues
- Enhanced Deserial Serial Interface (DSI) configuration where DSPI serializes up to 32 bits with three possible sources per bit
  - eTPU, eMIOS, new virtual GPIO registers as possible bit source
  - Programmable inter-frame gap in continuous mode
  - Bit source selection allows microsecond channel downstream with command or data frames up to 32 bits
  - Microsecond channel dual receiver mode
- Combined Serial Interface (CSI) configuration where the DSPI operates in both SPI and DSI configurations interleaving DSI frames with SPI frames, giving priority to SPI frames

For queued operations, the SPI queues reside in system memory external to the DSPI. Data transfers between the memory and the DSPI FIFOs are accomplished through the use of the eDMA controller or through host software.

The DSPI supports these SPI features:

- Full-duplex, synchronous transfers
- Selectable LVDS Pads working at 40 MHz for SOUT and SCK pins
- Master and Slave Mode
- Buffered transmit operation using the TX FIFO with parameterized depth of 4 entries
- Buffered receive operation using the RX FIFO with parameterized depth of 4 entries
- TX and RX FIFOs can be disabled individually for low-latency updates to SPI queues
- Visibility into the TX and RX FIFOs for ease of debugging
- FIFO Bypass Mode for low-latency updates to SPI queues
- Programmable transfer attributes on a per-frame basis:
  - Parameterized number of transfer attribute registers (from two to eight)
  - Serial clock with programmable polarity and phase
  - Various programmable delays:
    PCS to SCK delay
    SCK to PCS delay
    Delay between frames
  - Programmable serial frame size of 4 to 16 bits, expandable with software control
  - Continuously held chip select capability
- 6 Peripheral Chip Selects, expandable to 64 with external demultiplexer
- Deglitching support for up to 32 Peripheral Chip Selects with external demultiplexer
- DMA support for adding entries to TX FIFO and removing entries from RX FIFO:
  - TX FIFO is not full (TFFF)
  - RX FIFO is not empty (RFDF)
- 6 interrupt conditions:
  - End of queue reached (EOQF)
  - TX FIFO is not full (TFFF)
  - Transfer of current frame complete (TCF)
  - Attempt to transmit with an empty Transmit FIFO (TFUF)
  - RX FIFO is not empty (RFDF)
  - FIFO Underrun (slave only and SPI mode, the slave is asked to transfer data when the TxFIFO is empty)
  - FIFO Overrun (serial frame received while RX FIFO is full)
- Modified transfer formats for communication with slower peripheral devices
- Continuous Serial Communications Clock (SCK)
- Power savings via support for Stop Mode
- Enhanced DSI logic to implement a 32-bit Timed Serial Bus (TSB) configuration, supporting the Microsecond Channel downstream frame format

The DSPIs also support these features unique to the DSI and CSI configurations:

- 2 sources of the serialized data:
  - eTPU_A and eMIOS output channels
  - Memory-mapped register in the DSPI
- Destinations for the deserialized data:
  - eTPU_A and eMIOS input channels
  - SIU External Interrupt Request inputs
  - Memory-mapped register in the DSPI
- Deserialized data is provided as Parallel Output signals and as bits in a memory-mapped register
- Transfer initiation conditions:
  - Continuous
  - Edge sensitive hardware trigger
  - Change in data
- Pin serialization/deserialization with interleaved SPI frames for control and diagnostics
- Continuous serial communications clock
- Support for parallel and serial chaining of up to four DSPI blocks

### 3.3.16 eSCI

The enhanced Serial Communications Interface (eSCI) allows asynchronous serial communications with peripheral devices and other MCUs. It includes special support to

**Table 4. Revision history**

| Date | Revision | Description |
|---|---|---|
| 08-Jun-2011 | 5 | Replaced in all document "... Architecture Book E compliant..." with "... Architecture technology compliant..."<br>Updated "Introduction": Added "Document overview" and "Description" sections.<br>Made the following changes in the "Description" section:<br>– Added "Floating Point Unit (FPU)" bullet under "Single issue, 32-bit Power Architecture technology compliant e200z335 CPU core complex".<br>– Changed "eTPU" to "eTPU2" in the "32-channel second-generation enhanced time processor unit" sentence.<br>– Changed the "Available in LQFP100, LQFP144, LQFP176 and LBGA208" sentence to "Designed for LQFP100, LQFP144, LQFP176, and LBGA208 packages".<br>Replaced all instances of "Microsecond Bus" with "Microsecond Channel".<br>Replaced all instances of "SPC563M54" and "SPC563M60" with "SPC563M54P"and "SPC563M60P", respectively.<br>Changed the "Standby SRAM size" of the SPC563M60P device from "24" to "32".<br>Replaced all instances of "downlink" with "downstream" and "uplink" with "upstream".<br>Made the following changes in the "Flash" section:<br>Changed "Program page size of 128 bits (four words) to accelerate programming" to "Program page size of 64 bits (two words)".<br>Changed "Erase suspend, program suspend and erase-suspended program" to "Erase suspend".<br>Made editorial changes in the "BAM" section: Updated the conditional tags in the "Supports serial bootloading..." sentences.<br>Changed "Shared time bases with the eTPU through the counter buses" to "Shared time bases with the eTPU2 through the counter buses" in the "eMIOS" section.<br>Removed the following sentences from the "eTPU2" section:<br>"For Monaco 1.5M, the eTPU2 has been further enhanced with these features:"<br>"Timebases and channels are run at full system clock speed"<br>"Programmable channel mode allows customization of channel function"<br>"More flexibility in requesting DMA and interrupt service"<br>"Channel flags can be tested"<br>Added the following sentence in the "eQADC" section under the "Priority based Queues" bullet: "Streaming mode operation of Queue_0 to execute some commands several times".<br>Changed the following sentences in the "DSPI" section:<br>"The DSPI can be configured ... that implements ... the Microsecond Bus protocol" to "The DSPI can be configured ... that supports ... the Microsecond Channel protocol".<br>"The DSPI pins support 5 V logic levels or Low Voltage Differential Signalling (LVDS) to improve high speed operation." to "The DSPI ouput pins support 5 V logic levels or Low Voltage Differential Signalling (LVDS) according to the Microsecond Channel specification." |

**Table 4. Revision history**

| Date | Revision | Description |
|---|---|---|
| 08-Jun-2011 | 5 | Made the following changes in the DSPI section:<br>Changed “bus downlink” to “channel downstream”, “Microsecond bus” to “Microsecond channel”, “SIN” to “LVDS pads are for outputs only”, “1 to 16” to “4”, “bus standard” to “channel upstream”, and “17-pin Full Port interface in calibration tool calibration package” to “17-pin Full Port interface in calibration package used on calibration tool boards”.<br>Updated the Order codes table summary table and figures.<br>Changed the “Processor core” information of the SPC563M54P device to: “32-bit e200z335 with SPE and FPU support”.<br>Added the following to the “Available only in the calibration package” sentence in the Features section (496 CSP package). |
| 17-Sep-2013 | 6 | Updated Disclaimer |