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"

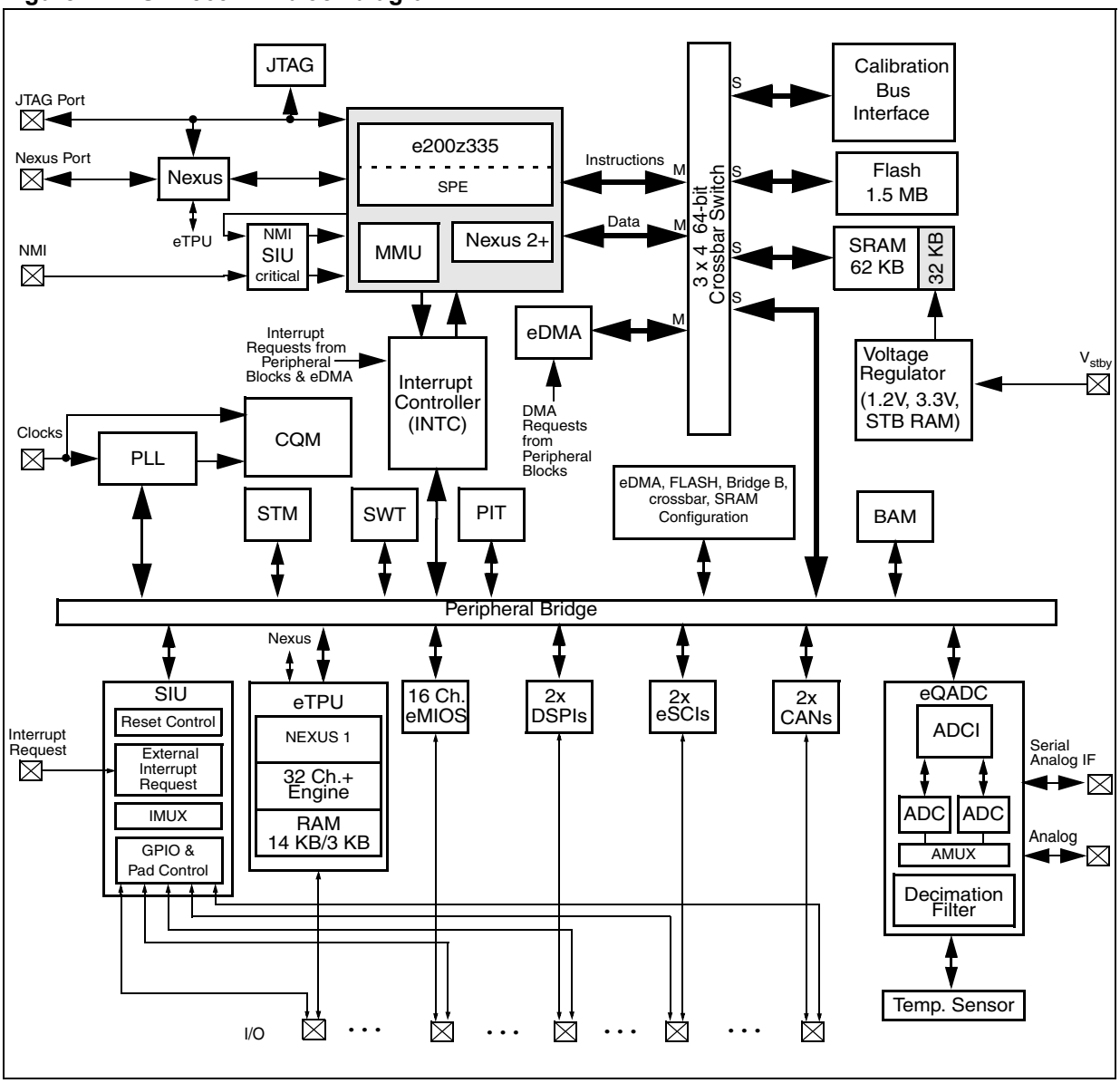
Details	
Product Status	Active
Core Processor	e200z3
Core Size	32-Bit Single-Core
Speed	64MHz
Connectivity	CANbus, EBI/EMI, LINbus, SCI, SPI, UART/USART
Peripherals	DMA, POR, PWM, WDT
Number of I/O	80
Program Memory Size	1.5MB (1.5M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	94K x 8
Voltage - 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/spc563m64l7cobr

List of tables

2 Block diagram

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

Figure 1. SPC563Mxx block diagram



  - 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

- Zero jitter triggering for queue 0. (Queue 0 trigger causes current conversion to be aborted and the queued conversions in the CBUFFER to be bypassed. Delay from Trigger to start of conversion is 13 system clocks + 1 ADC clock.)
- eQADC Result Streaming. Generation of a continuous stream of ADC conversion results from a single eQADC command word. Controlled by two different trigger signals; one to define the rate at which results are generated and the other to define the beginning and ending of the stream. Used to digitize waveforms during specific time/angle windows, e.g., engine knock sensor sampling.
- Angular Decimation. The ability of the eQADC to sample an analog waveform in the time domain, perform Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) filtering also in the time domain, but to down sample the results in the angle domain. Resulting in a time domain filtered result at a given engine

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

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.

The PLL has the following major features:

- Input clock frequency from 4 MHz to 20 MHz
- Voltage controlled oscillator (VCO) range from 256 MHz to 512 MHz, resulting in system clock frequencies from 16 MHz to 80 MHz with granularity of 4 MHz or better
- Reduced frequency divider (RFD) for reduced frequency operation without forcing the PLL to relock
- 3 modes of operation
  - Bypass mode with PLL off
  - Bypass mode with PLL running (default mode out of reset)
  - PLL normal mode
- Each of the three modes may be run with a crystal oscillator or an external clock reference
- Programmable frequency modulation
  - Modulation enabled/disabled through software
  - Triangle wave modulation up to 100 kHz modulation frequency
  - Programmable modulation depth (0% to 2% modulation depth)
  - Programmable modulation frequency dependent on reference frequency
- Lock detect circuitry reports when the PLL has achieved frequency lock and continuously monitors lock status to report loss of lock conditions
- Clock Quality Module
  - detects the quality of the crystal clock and cause interrupt request or system reset if error is detected
  - detects the quality of the PLL output clock. If an error is detected, causes a system reset or switches the system clock to the crystal clock and causes an interrupt request
- Programmable interrupt request or system reset on loss of lock

3.3.6 Calibration EBI

The Calibration EBI controls data transfer across the crossbar switch to/from memories or peripherals attached to the calibration tool connector in the calibration address space. The Calibration EBI is only available in the calibration tool. The Calibration EBI includes a memory controller that generates interface signals to support a variety of external memories. The Calibration EBI memory controller supports legacy flash, SRAM, and asynchronous memories. In addition, the calibration EBI supports up to three regions via chip selects (two chip selects are multiplexed with two address bits), along with programmed region-specific attributes. The calibration EBI supports the following features:

- 22-bit address bus (two most significant signals multiplexed with two chip selects)
- 16-bit data bus
- Multiplexed mode with addresses and data signals present on the data lines

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

The Flash memory provides the following features:

- Supports a 64-bit data bus for instruction fetch, CPU loads and DMA access. Byte, halfword, word and doubleword reads are supported. Only aligned word and doubleword writes are supported.
- Fetch accelerator
  - Architected to optimize the performance of the flash with the CPU to provide single cycle random access to the flash up to 80 MHz system clock speed
  - Configurable read buffering and line prefetch support
  - Four line read buffers (128 bits wide) and a prefetch controller
- Hardware and software configurable read and write access protections on a per-master basis
- Interface to the Flash array controller is pipelined with a depth of one, allowing overlapped accesses to proceed in parallel for interleaved or pipelined Flash array designs
- Configurable access timing allowing use in a wide range of system frequencies
- Multiple-mapping support and mapping-based block access timing (0-31 additional cycles) allowing use for emulation of other memory types
- Software programmable block program/erase restriction control
- Erase of selected block(s)
- Read page size of 128 bits (four words)
- ECC with single-bit correction, double-bit detection
- Program page size of 64 bits (two words)
- ECC single-bit error corrections are visible to software
- Minimum program size is two consecutive 32-bit words, aligned on a 0-modulo-8 byte address, due to ECC
- Embedded hardware program and erase algorithm
- Erase suspend
- Shadow information stored in non-volatile shadow block
- Independent program/erase of the shadow block

3.3.10 SRAM

The SPC563Mxx SRAM module provides a general-purpose up to 94 KB memory block. The SRAM controller includes these features:

- Supports read/write accesses mapped to the SRAM memory from any master
- 32 KB or 24 KB block powered by separate supply for standby operation
- Byte, halfword, word and doubleword addressable
- ECC performs single-bit correction, double-bit detection on 32-bit data element

3.3.11 BAM

The BAM (Boot Assist Module) is a block of read-only memory that is programmed once by ST and is identical for all SPC563Mxx MCUs. The BAM program is executed every time the

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.

The eMIOS provides the following features:

- 16 channels (24-bit timer resolution)
- For compatibility with other family members selected channels and timebases are implemented:
  - Channels 0 to 6, 8 to 15, and 23
  - Timebases A, B and C
- Channels 1, 3, 5 and 6 support modes:
  - General Purpose Input/Output (GPIO)
  - Single Action Input Capture (SAIC)
  - Single Action Output Compare (SAOC)
- Channels 2, 4, 11 and 13 support all the modes above plus:
  - Output Pulse Width Modulation Buffered (OPWMB)
- Channels 0, 8, 9, 10, 12, 14, 15, 23 support all the modes above plus:
  - Input Period Measurement (IPM)
  - Input Pulse Width Measurement (IPWM)
  - Double Action Output Compare (set flag on both matches) (DAOC)
  - Modulus Counter Buffered (MCB)
  - Output Pulse Width and Frequency Modulation Buffered (OPWFMB)
- Three 24-bit wide counter buses
  - Counter bus A can be driven by channel 23 or by the eTPU2 and all channels can use it as a reference
  - Counter bus B is driven by channel 0 and channels 0 to 6 can use it as a reference
  - Counter bus C is driven by channel 8 and channels 8 to 15 can use it as a reference
- Shared time bases with the eTPU2 through the counter buses
- Synchronization among internal and external time bases

3.3.13 eTPU2

The eTPU2 is an enhanced co-processor designed for timing control. Operating in parallel with the host CPU, eTPU2 processes instructions and real-time input events, performs output waveform generation, and accesses shared data without host intervention. Consequently, for each timer event, the host CPU setup and service times are minimized or eliminated. A powerful timer subsystem is formed by combining the eTPU2 with its own instruction and data RAM. High level assembler/compiler and documentation allows customers to develop their own functions on the eTPU2.

- 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.

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.

interface to Local Interconnect Network (LIN) slave devices. The eSCI block provides the following features:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- 13-bit baud rate selection
- Programmable 8-bit or 9-bit, data format
- Programmable 12-bit or 13-bit data format for Timed Serial Bus (TSB) configuration to support the Microsecond Channel upstream
- Automatic parity generation
- LIN support
  - Autonomous transmission of entire frames
  - Configurable to support all revisions of the LIN standard
  - Automatic parity bit generation
  - Double stop bit after bit error
  - 10- or 13-bit break support
  - Separately enabled transmitter and receiver
  - Programmable transmitter output parity
  - 2 receiver wake up methods:
    Idle line wake-up
    Address mark wake-up
- Interrupt-driven operation with flags
- Receiver framing error detection
- Hardware parity checking
- 1/16 bit-time noise detection
- DMA support for both transmit and receive data
  - Global error bit stored with receive data in system RAM to allow post processing of errors

3.3.17 FlexCAN

The SPC563Mxx MCU contains two controller area network (FlexCAN) blocks. The FlexCAN module is a communication controller implementing the CAN protocol according to Bosch Specification version 2.0B. The CAN protocol was designed to be used primarily as a vehicle serial data bus, meeting the specific requirements of this field: real-time processing, reliable operation in the EMI environment of a vehicle, cost-effectiveness and required bandwidth. FlexCAN module ‘A’ contains 64 message buffers (MB); FlexCAN module ‘C’ contains 32 message buffers.

The FlexCAN module provides the following features:

- Full Implementation of the CAN protocol specification, Version 2.0B
  - Standard data and remote frames
  - Extended data and remote frames
  - Zero to eight bytes data length
  - Programmable bit rate up to 1 Mbit/s
- Content-related addressing
- 64 / 32 message buffers of zero to eight bytes data length
- Individual Rx Mask Register per message buffer
- Each message buffer configurable as Rx or Tx, all supporting standard and extended messages
- Includes 1056 / 544 bytes of embedded memory for message buffer storage
- Includes a 256-byte and a 128-byte memories for storing individual Rx mask registers
- Full featured Rx FIFO with storage capacity for six frames and internal pointer handling
- Powerful Rx FIFO ID filtering, capable of matching incoming IDs against 8 extended, 16 standard or 32 partial (8 bits) IDs, with individual masking capability
- Selectable backwards compatibility with previous FlexCAN versions
- Programmable clock source to the CAN Protocol Interface, either system clock or oscillator clock
- Listen only mode capability
- Programmable loop-back mode supporting self-test operation
- 3 programmable Mask Registers
- Programmable transmit-first scheme: lowest ID, lowest buffer number or highest priority
- Time Stamp based on 16-bit free-running timer
- Global network time, synchronized by a specific message
- Maskable interrupts
- Warning interrupts when the Rx and Tx Error Counters reach 96
- Independent of the transmission medium (an external transceiver is assumed)
- Multi master concept
- High immunity to EMI
- Short latency time due to an arbitration scheme for high-priority messages
- Low power mode, with programmable wake-up on bus activity

3.3.18 System timers

The system timers provide two distinct types of system timer:

- Periodic interrupts/triggers using the Peripheral Interrupt Timer (PIT)
- Operating system task monitors using the System Timer Module (STM)

Peripheral Interrupt Timer (PIT)

The PIT provides five independent timer channels, capable of producing periodic interrupts and periodic triggers. The PIT has no external input or output pins and is intended to be used to provide system ‘tick’ signals to the operating system, as well as periodic triggers for eQADC queues. Of the five channels in the PIT, four are clocked by the system clock, one is

Note: This feature is imprecise due to CPU pipelining.

  - Subset of Power Architecture software debug facilities with OnCE block (Nexus class 1 features)
- eTPU development support features
  - IEEE-ISTO 5001-2003 standard class 1 compliant for the eTPU
  - Nexus based breakpoint configuration and single step support (JTAG feature of the eTPU)
- Run-time access to the on-chip memory map via the Nexus read/write access protocol. This feature supports accesses for run-time internal visibility, calibration variable acquisition, calibration constant tuning, and external rapid prototyping for powertrain automotive development systems.
- All features are independently configurable and controllable via the IEEE 1149.1 I/O port
- Power-on-reset status indication during reset via MDO[0] in disabled and reset modes

JTAG

The JTAGC (JTAG Controller) block provides the means to test chip functionality and connectivity while remaining transparent to system logic when not in test mode. Testing is performed via a boundary scan technique, as defined in the IEEE 1149.1-2001 standard. All data input to and output from the JTAGC block is communicated in serial format. The JTAGC block is compliant with the IEEE 1149.1-2001 standard and supports the following features:

- IEEE 1149.1-2001 Test Access Port (TAP) interface 4 pins (TDI, TMS, TCK, and TDO)
- A 5-bit instruction register that supports the following IEEE 1149.1-2001 defined instructions:
  - BYPASS, IDCODE, EXTEST, SAMPLE, SAMPLE/PRELOAD, HIGHZ, CLAMP
- A 5-bit instruction register that supports the additional following public instructions:
  - ACCESS_AUX_TAP_NPC
  - ACCESS_AUX_TAP_ONCE
  - ACCESS_AUX_TAP_eTPU
  - ACCESS_CENSOR
- 3 test data registers to support JTAG Boundary Scan mode
  - Bypass register
  - Boundary scan register
  - Device identification register
- A TAP controller state machine that controls the operation of the data registers, instruction register and associated circuitry
- Censorship Inhibit Register
  - 64-bit Censorship password register
  - If the external tool writes a 64-bit password that matches the Serial Boot password stored in the internal flash shadow row, Censorship is disabled until the next system reset.

4 Orderable parts

Table 3. Order codes

Order code	Flash/SRAM (Kbytes)	Package	Speed (MHz)
SPC563M60L5CPBR	1024 / 64	LQFP144 Pb-free	64
SPC563M60L5CPBY			
SPC563M60L5CPAR	1024 / 64	LQFP144 Pb-free	80
SPC563M60L5CPAY			
SPC563M60L7CPBR	1024 / 64	LQFP176 Pb-free	64
SPC563M60L7CPBY			
SPC563M60L7CPAR	1024 / 64	LQFP176 Pb-free	80
SPC563M60L7CPAY			
SPC563M64L5COBR	1536 / 94	LQFP144 Pb-free	64
SPC563M64L5COBY			
SPC563M64L5COAR	1536 / 94	LQFP144 Pb-free	80
SPC563M64L5COAY			
SPC563M64L7COBR	1536 / 94	LQFP176 Pb-free	64
SPC563M64L7COBY			
SPC563M64L7COAR	1536 / 94	LQFP176 Pb-free	80
SPC563M64L7COAY			

Table 4. Revision history

Date	Revision	Description
18-May-2011	4	Internal review. Block diagram updated: – Test Controller removed. Device Summary (device comparison) table updated: – The SPC563M64 will not be offered in an LQFP100 package. – Footnote added clarifying the number of eQADC channels. Feature list: – “Single power supply” applies to all packages. (previously stated 100- and 144-pin packages) – “Nexus pins powered by 3.3 V supply” applies to all packages. (previously stated LBGA208 and LQFP176 packages) – Calibration pin voltages deleted – Details added to eTPU and eQADC features – eMIOS features updated – eTPU renamed to ETPU2; features updated Orderable parts table updated

Table 4. Revision history

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