MMC2107 - A 32-bit System on a Chip with FLASH

David Ruimy Gonzales
Senior Member of Technical Staff
Motorola Embedded Platform Solutions Austin, Texas
David.Gonzales@Motorola.com

Distribution channel customers designing high-performance, cost- and power-sensitive applications, such as vending machines, building management and heating-ventilation-air conditioning (HVAC) systems, exercise equipment and lighting control need highly integrated microcontrollers off the shelf which reduce system component count and fit their product requirements. There are also many wireless networking and infrastructure applications, internet appliances as well as special portable products that need off-the-shelf general purpose microcontrollers. These applications require microcontrollers running at speeds greater than 33mhz with on-chip RAM and FLASH memory as well as an array of sophisticated peripherals that are easy to program in a high level language.

This paper discusses the key elements of a general purpose microcontroller and their use in a real-time application. The interaction of the peripherals with the processor is essential for collecting, processing and outputting data. To better appreciate how these peripherals interact with the processor, a review of each of the major functions of a microcontroller will be discussed. The low power Motorola MMC2107 32-bit microcontroller architecture will be used as an example solution for designing a complex system using a minimum number of components.

System Overview

The most common elements of a single-chip microcontroller include the clock and reset logic, the processor, the memory which stores information, the interrupt controller to the processor, the peripherals and the external bus interface. The types of peripherals vary from microcontroller vendors depending on the market segment they are targeting. The most widely used in the majority of applications will be general purpose input/output ports, timers, serial interfaces and analog to digital converters. Having this host of resources on a single chip reduces visibility of their interaction with the processor so some sort of debug hooks are needed to help designers during hardware/software integration. Figure 1 illustrates the MMC2107 microcontroller architecture which includes all these commonly used elements needed to solve a variety of tasks.

The processor is the heart of the system

The central processing unit (CPU) which controls program flow is very important as it determines how fast you may compute values, the types on memory access, the method of development and whether it may be applied to a low power application. The arithmetic precision is also defined and the number of registers for temporary variable storage will dictate how efficient compiled C code will be. Interrupt handling is also important as the delay between an interrupt request and and the time it is serviced is critical in real-time applications.

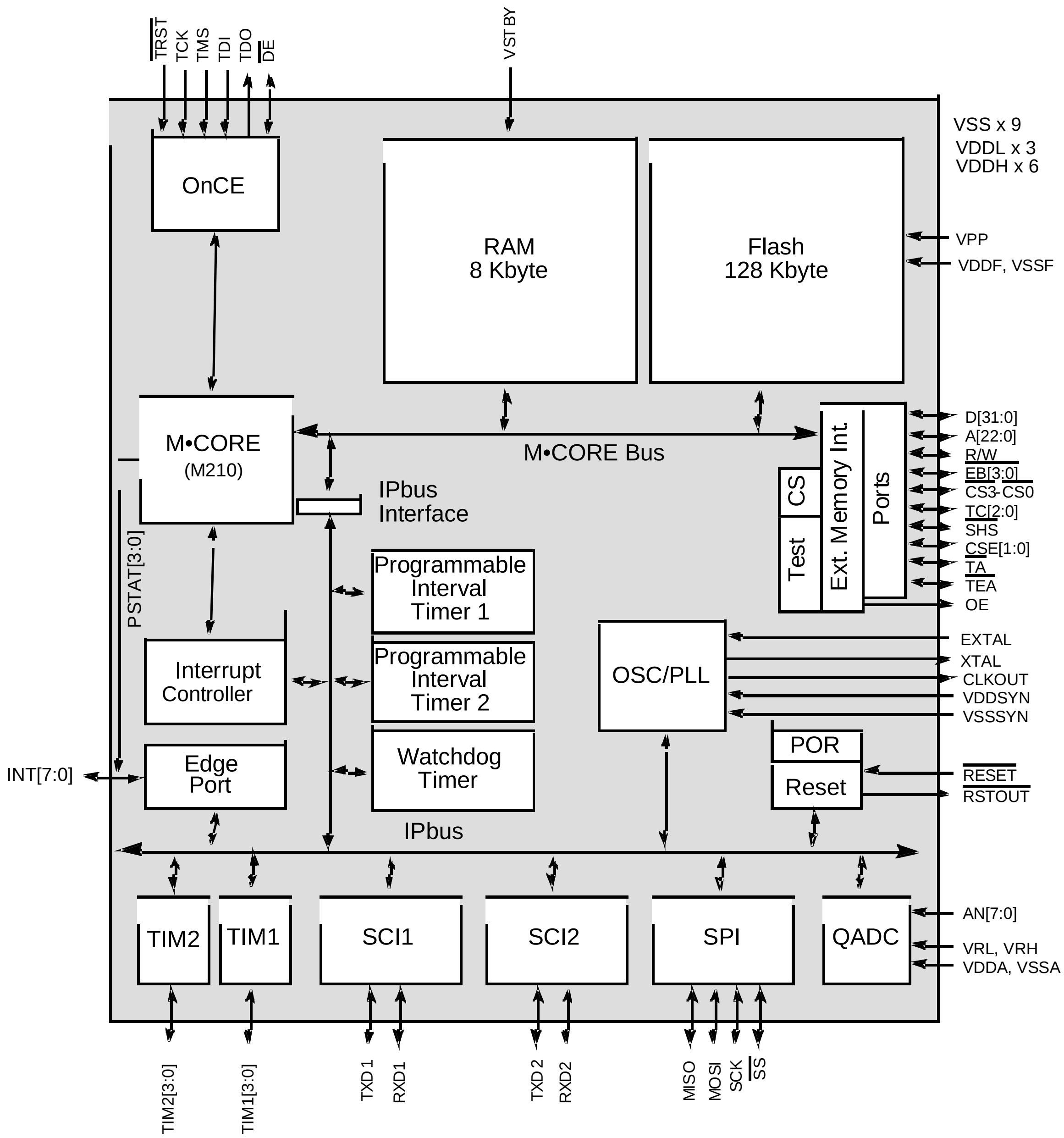

Figure 1 - MMC2107 Microcontroller Block Diagram

The MMC2107 implements a low power micro-RISC M-CORE™ architecture which supports full 32-bit integer arithmetic including a hardware multiplier and divider. It uses a 4 stage pipeline for efficient execution of streamlined 16-bit instructions thus permitting low interrupt service latency. There are a total of forty five 32-bit registers which are used for data storage and program context switch information facilitating efficient compiler generated code. Load and store opcodes permit single or multiple byte, half-word and word data movement. There are special instructions for determining the source of an interrupt and storage and retrieval of program state.

The M-CORE processor uses dynamic clock management to automatically power-down internal functions that are not in use on a clock-by-clock basis. It also incorporates three power-conservation operating modes, which are invoked via dedicated instructions.

On-Chip Memory Arrays

An embedded microcontroller ideally has on-chip memory arrays to reduce system component count, power consumption and electromagnetic interference due to toggling board signal traces not to mention overall system cost. SRAM is quite common for temporary program execution and variable storage while on-chip FLASH is used for permanent storage. Peripherals are commonly memory mapped to specific addresses so they are easily accessible by load and store opcodes in the instruction set.

The primary function of a FLASH is to serve as electrically programmable and erasable non-volatile memory (NVM) to store program instructions and/or data. It is a class of nonvolatile solid state silicon memory device consisting of an array of isolated elements, a means for selectively adding and removing charge to the elements electrically and a means of selectively sensing the stored charge in the elements. When power is removed from the device, the stored charge of the isolated elements are retained, thus it is ideal for saving system configuration parameters.

The MMC2107 has 8-Kbytes SRAM with standby voltage support for low power applications. When the chip is powered down, the contents of the SRAM array are maintained by the standby power supply, V_{STBY}. If the standby voltage falls below the minimum required voltage, the SRAM contents may be corrupted. The SRAM automatically switches to standby operation with no loss of data when the voltage on V_{DD} is below the voltage on V_{STBY}. In standby mode, the SRAM does not respond to any bus cycles.

NVM storage is accomplished with 128 Kbytes of FLASH implemented in eight array blocks of 16-Kbytes each. Having array blocks permits independent erase, address attributes restriction, and protection from program and erase for each array block. To improve program performance, the FLASH programs up to eight unique 64-byte pages simultaneously in eight separate array blocks. These 64 bytes are aligned to the low-order addresses to form a program page buffer. The FLASH memory can be read, programmed, and erased from a single external

V_{PP} programming voltage supply, thus minimizing external components for FLASH support.

Receiving and Servicing Real-Time Interrupts

A key task in an embedded microcontroller is the ability to rapidly service one or more interrupt inputs. Interrupts may come from a variety of sources and can vary successively in time from greater than 1 second to less than 10 microseconds. Therefore it is important to evaluate the worst case conditions of interrupt request to interrupt service delay. It is also important to understand how many interrupts may be serviced, whether you are able to turn them off and whether you are able to prioritize the interrupts in hardware and software.

Efficient interrupt handling is accomplished on the MMC2107 using a combination of M-CORE processor and interrupt controller features. As many as 40 interrupt sources may be serviced using 32 unique programmable priority levels for each interrupt source. Each interrupt source is maskable and each priority level can be programmmed for normal or fast interrupt requests where fast interrupt requests always have priority over normal interrupts. The interrupt input ports may be uniquely programmed to be edge or level triggered.

Interrupt servicing may be autovectored or vectored interrupt requests. Vectored interrupts are generated based on priority level. Separate vector numbers are available for normal and fast interrupts to facilitate rapid service handling. Software can self-schedule interrupts using status information on pending interrupts. The interrupt controller is not affected by any low-power modes. All logic between the input sources and the M-CORE processor interrupt signals are combinatorial. This permits the M-CORE processor to wake up during low-power stop mode when all system clocks are stopped.

Communicating with other devices asynchronously

Many systems require a means for communicating with one or more host computers, terminals and other microcontrollers using the asynchronous RS232 protocol for serial communication. It allows information to be passed in an industry standard method using a minimum of two wires.

The MMC2107 has two on-chip serial communications ports with full-duplex operation and standard mark/space NRZ format. It implements a 13-bit baud rate selector with programmable 8-bit or 9-bit data format. There are separate receiver and transmitter interrupt requests with eight different flags for such tasks as framing error detection and hardware parity checking. Its asynchronous clock is sampled every 1/16 of a bit-time to insure reliable data input into its receive shift register.

Communicating with other devices synchronously

There are many special circuits that may be required during system integration that perform a special function. These standalone peripherals may range from analog to digital or digital to analog converters, serial flash memory or Liquid Crystal Display (LCD) arrays which implement a standard synchronous serial peripheral interface (SPI) protocol allowing for high speed full duplex transfers of data with the microcontroller. These high speed serial transfers are based on a multiple of the microcontroller clock.

The MMC2107 implements a 4 wire SPI interface where it may be configured as a master or slave with a programmable serial clock which may be an input or output. Maskable interrupts may be generated on every 8^{th} clock after the receiver becomes full. The receiver and transmitter are double buffered allowing full duplex transmissions. The output pin may be configured as open drain for wired-OR systems or as a CMOS output with full or reduced drive capability.

Measuring Events, keeping time and creating periodic pulses

Real-time systems use special hardware timers for keeping time, measuring and creating events and generating periodic interrupts. These timers help calculate and control events such as engine velocity and acceleration, waveform generation, periodic interrupt generation, real-time clocks and stepper motor controllers. These applications require timers with counters that can measure and create variable pulse widths and automatic reloading to reduce overhead from microcontroller service.

The MMC2107 provides such capabilities with two 16-bit timers with input capture and output compare circuits. There is a periodic interval timer with a 16-bit counter selectable as a free running or down counter.

Collecting Analog Information

Analog to digital converters (ADC) serve many different areas in embedded applications ranging from audio filtering, noise cancellation, frequency equalization or waveform analysis. ADCs come in different technologies, have different response times and accuracies and vary in the number of channels of input they may monitor.

The MMC2107 has a sophisticated queued ADC, the QADC, which can hold up to 40 command words and 40 10-bit conversion results. Operations can be queued, and the QADC automatically can convert multiplexed analog entries with up to 16 analog input channels. With additional external multiplexers, you can extend the QADC to service 44 distinct analog channels. You can set trigger sampling by an external trigger, a QADC timer, or via software. The QADC can sample two channels simultaneously and can generate an interrupt for each queue or sub-queue chain. There are 64 result registers that not only allow the user to obtain the results from all the channels, but also to make repeated multiple measurements on one channel. Two queues control the conversion order of the QADC channels.

System Features that Enhance your Application

There are still things to consider which increase reliability of your system and help increase battery life in portable applications. For example, if your application operates in harsh environments, the code may periodically receive high voltages or mechanical shock that may cause your system to stall or get lost. This may require some special timer that insures your application program continues to operate properly. System reset, external memory interface and debugging are also important functions needed.

The MMC2107 implements a 16-bit watchdog timer to insure application code operates properly. There are 3 low power modes of operation which allow powering down particular peripherals until an interrupt service from the peripheral is required. An on-chip phase lock loop (PLL) permits frequency scaling so that system frequency of operation is dynamically programmable. There may be six sources of reset including power on reset, external reset input, software reset, loss of clock reset or loss of PLL lock reset and watchdog timer reset.

Interfacing to external memories or circuits is facilitated via a full 32-bit external bidirectional data bus with 23 bit address bus and four chip select signals. If the external bus is not needed, the pins may be used as general purpose input/output lines. Debugging code is facilitated using a JTAG interface to a OnCE™ debug port for source level debugging and direct access to all on-chip resources in your application. Internal bus activity may be brought out via a show cycle mode for direct hookup to your Tektronix or Agilent logic analyzer. Software development tools are available from Metrowerks and Wind River Systems, which permit C/C++ coding and debugging of application programs.

Summary

The MMC2107 has become quite a popular off the shelf microcontroller for a variety of application areas due to its combination of on-chip SRAM and FLASH, M-CORE processor, digital and analog peripherals, system configuration circuits and resident debugging hooks. The device is ideal for low power applications, which require 16 or 32-bit precision arithmetic and efficient interrupt handling. For additional information visit the Motorola web site at www.mot.com/sps