

Thomas Mauer,
Digital Applications Engineer,
Texas Instruments

Implementing Industrial Interfaces with the Programmable Realtime Core in the Texas Instruments AM1808 ARM[®] Microprocessor

Introduction

The realtime requirements of microprocessor interfaces lie in the range of mere microseconds, if not nanoseconds. Conventional microprocessors are largely operated using a high-level operating system (HLOS), such as Embedded Linux™ or Windows® Embedded CE. However, this makes it very difficult to achieve a deterministic reaction time of less than one millisecond in the user application or kernel driver.

To achieve the necessary reaction times for a realtime interface, it is necessary to decouple the HLOS using a dedicated peripheral block. The interface in question is typically made available by the manufacture in hardware on the microprocessor (UART, SPI and I²C are examples). If no appropriate interface is available on the microprocessor, the realtime requirement must be decoupled from the HLOS using an external application-specific integrated circuit (ASIC) or a field programmable gate array (FPGA) providing the corresponding functionality.

Above all, the industrial sector presents an enormous range of complex applications with realtime requirements. The developer can only provide

the interface necessary to meet these requirements using an external ASIC or FPGA. This means using more space on the board and additional component costs.

The Texas Instruments (TI) AM1808 ARM microprocessor offers an alternative solution. This article describes a method that integrates the realtime requirements of an interface back into the microprocessor, without needing an ASIC or FPGA.

The programmable realtime core in the AM1808 MPU

The AM1808 is a low-power ARM9™ microprocessor based on an ARM926EJ-S™ RISC core, and may be clocked at up to 456 MHz. It includes a range of valuable on-chip peripherals, which may be usefully deployed in industrial applications: a 10/100 Mb/s Ethernet MAC (EMAC) with Management Data Input/Output (MDIO), USB interfaces, I²C interfaces, multichannel audio serial port (McASP) with 16 serializers, multichannel buffered serial ports (McBSP), SPI interfaces, 64-bit timer, host port interface, UARTs, enhanced PWM module, enhanced capture module, GPIO, LCD controller, connection to external memory, and asynchronous external peripheral modules.

An important distinguishing feature included in the peripheral devices is the programmable realtime core (Programmable Realtime Unit, PRU). It can access all system resources independent of the ARM core. The PRU may be freely programmed by the developer and is ideal for realtime interface, Layer 2 protocol processing and proprietary manufacturer protocols. Particularly in the field of industrial applications, which are distinguished by a large number of different interface protocols, the implementation of the realtime interface in the PRU may result in savings on external hardware components.

Figure 1 on the following page shows the PRU Subsystem (PRUSS):

The PRUSS contains two PRU cores consisting of independent 32-bit RISC processors with their own instruction and data memory. They are clocked at half the frequency of the ARM core. The zero-stage pipeline on the instruction memory allows the PRU to execute all commands in a single cycle. So, if the ARM core is clocked at 456 MHz, each PRU will run at 228 MHz, which corresponds to an instruction cycle time of only 4.4 ns. Knowledge of the

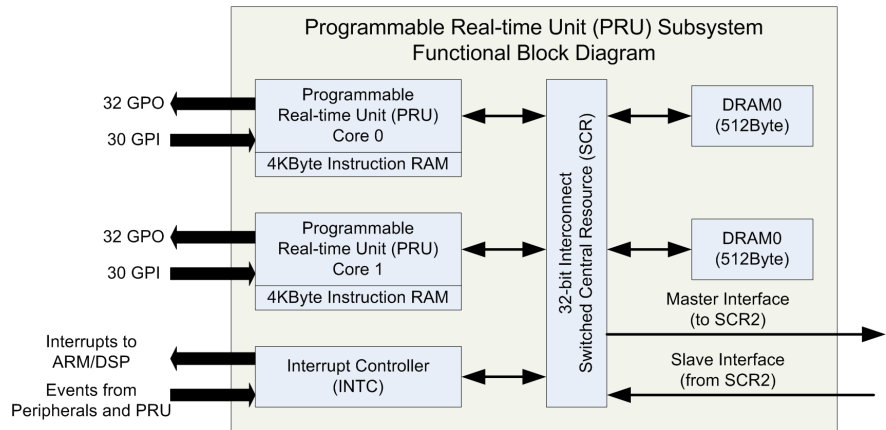


Figure 1.

instruction cycle time makes it possible to calculate the execution time of program algorithms in the PRU, and thus also to implement interfaces with realtime requirements.

Register addressable inputs and outputs

The PRUs feature 30 inputs and 32 outputs, which are addressed directly using registers. These inputs and outputs are tapped at the microprocessor pins after pin multiplexing for external signals has been configured in the AM1808 MPU. This allows each PRU to switch external signals within just a few nanoseconds and react to external signal changes with service routines. Application examples include reading in data after an external event, serial protocol generation with clock and data lines and the connection of external devices.

Interrupt controller and system events

Another component of the PRUSS is an interrupt controller (INTC) through which each PRU can receive events (interrupts) by on-chip peripherals and the ARM® core. If an event is triggered by the on-chip peripheral, it will be registered in the INTC and evaluated in the program flow in the PRU. This means that even complex protocols interacting with the on-chip peripherals can be implemented in the PRU. The serializers included with the McASP/McBSP ports are an example of this: The PRU processes the Layer 2 protocol on the byte level and uses the serializer ports for physical input and output.

The bit-test command allows each PRU to check whether an event was triggered. Evaluating the event number in the PRU then makes it possible to call the proper service function. The service function then acknowledges for the event in the INTC to allow reception of the next event.

Eight system events are additionally available in the INTC to trigger interrupts in the ARM core. These system events exist for effective inter-process communication (IPC) between the PRU and the ARM, avoiding unnecessary polling.

Internal bus architecture

Each PRU can also act as an independent bus master, just like the ARM® core and the EDMA controller. This means it is capable of accessing all system resources mapped in system memory, e.g., internal and external memory, UART, McASP, timer, and all other peripherals available to the system. To guarantee data consistency, each peripheral block should only be used by one bus master. The Linux™ kernel allows for resource allocation to prevent such dual use. The IPC is an exception, as data exchange takes place over a shared memory range.

The internal bus architecture (Switched Central Resources, SCR) of the AM1808 MPU, which is used to connect all master and peripheral devices with one another, allows for the parallel execution of multiple bus transactions without the bus masters blocking one another. This means the PRU can access internal memory and the ARM core external memory without reducing memory throughput.

Programming the realtime unit

The PRU is programmed in assembly language. Over 40 assembler commands are available for this purpose. They cover arithmetic, logic, flow control, memory access, bit and packet manipulation, and pattern search functions over a number of registers. In the AM1808 MPU, each PRU has 4 kB of instruction memory for up to 1024 assembly commands. The PRU program is loaded into instruction memory and started either by the ARM core, the EDMA controller or the second PRU. The PRU can be stopped at any time and loaded with a new program. This makes it possible to cover multiple functions with the PRU, dynamically loading them depending on the current application requirements. The PRUSS provides control registers to start and stop the PRU core. Another register provides information on the number of processed PRU command cycles.

PRU implementation example: Data acquisition and digital reduction filter for a delta-sigma modulator

Implementation of data acquisition and a reduction filter in the PRUSS allows a low-cost delta-sigma modulator to be employed in the solution presented here.

A delta-sigma modulator returns the average value of a time range, Δt . Oversampling in the modulator means several data values are required to generate a single output value. Many cases see the use of a SINC filter – an easy-to-implement and cascaded low-pass filter (see the application brief at focus.ti.com/lit/an/sbaa094/sbaa094.pdf). The modulator is suitable for low-frequency data acquisition, e.g., temperature measurements, audio signal processing or current flow metering at motor PWMs and generators.

The ADS1202 delta-sigma modulator used in this example has a dynamic range of 80 dB with an effective 12-bit resolution at 10 kHz signal bandwidth. While oversampling takes place at 10 MHz. It is operated with a 5-V supply. Its differential inputs allow sampling at transducers/magamps and low-voltage signals.

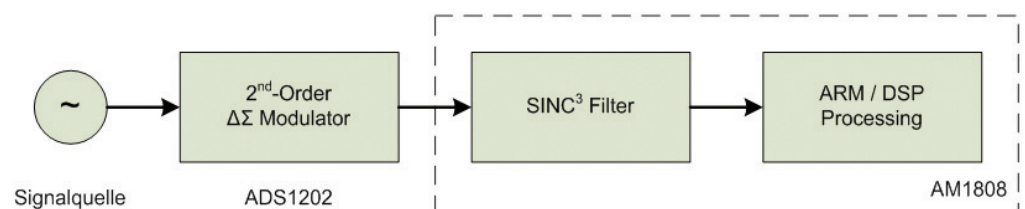


Figure 2.

The delta-sigma modulator is connected to the PRUSS using the data (MDAT) and clock (MCLK) lines to two input bits mapped in register R31. The bit-test command (QBBS – Quick Branch Bit Set, jump when bit is set) allows the PRU program to query the status of an input in a single clock cycle.

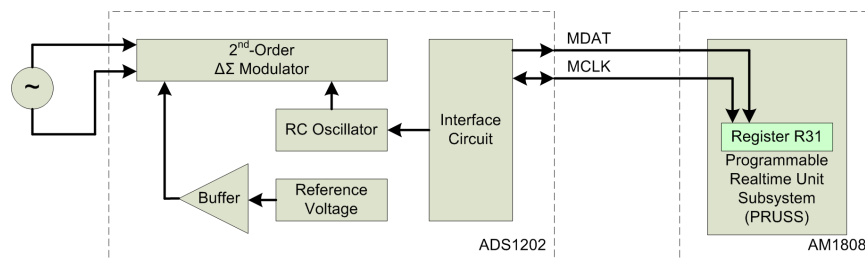


Figure 3.

The delta-sigma modulator delivers a new sample bit every 100 ns by oversampling the 10 MHz input signal. The sample bit must be fetched and processed by the PRU within this period of time before the next sample bit from the delta-sigma modulator is ready. If this does not occur, sample bits will be lost and the measured value is distorted.

A PRU clocked at 225 MHz can execute 22.5 commands during this 100 ns period. The implementation of data acquisition and filtering may not exceed this command budget.

The following functions are implemented in the PRU:

1. Wait for the next valid measured value to become available: Test the MCLK bit in register R31 for a transition from 0 to 1. This event triggers the following steps.
2. Process the MDAT bit sample from register R31 in the SINC filter integrator stage.
3. Call the SINC filter differentiation stage, depending on the selected reduction value (oversampling rate).
4. Save the measured value in local memory and, if necessary, signal event to the ARM for further measured value processing.
5. Test the measured value for min/max limit exceedance.

The diagram in Figure 4 on the next page lists the individual PRU tasks and compares the number of commands necessary between SINC² and SINC³ filters. The assembler commands for the SINC3 filter implementation are also displayed in a simplified form.

The first four functions are carried out for every sample. This part uses six (for SINC2) or seven (SINC3) commands from a budget of 22.5 commands per bit sample. Depending on the selected oversampling rate, the program either waits for the next sample bit or calls the SINC filter differentiation stage. The next measuring result only becomes ready for further processing after the SINC filter differentiation stage. For this reason, the measured value min/max check and saving the sample to local memory only takes place after this step. In this example, it can be recognized that a SINC3 filter implementation with 21 commands uses almost the entire budget – to reduce the budget load, the min/max check can be postponed to the next sample bit cycle, which will only require seven cycles.

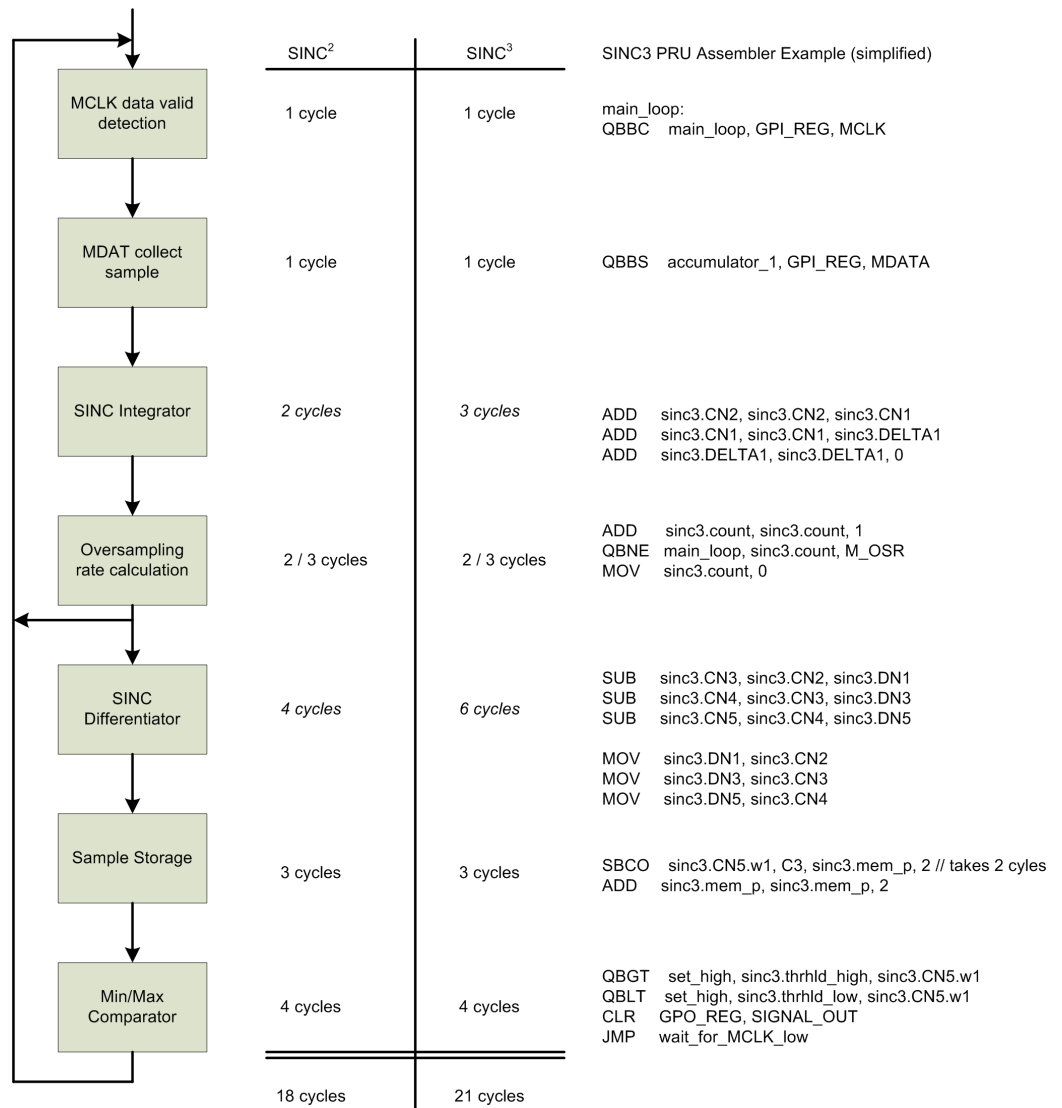


Figure 4.

Summary and outlook

The programmable realtime core in the AM1808 MPU makes it possible to implement customer-specific interfaces and protocols without any need for additional components. The realtime interfaces programmed by the PRUSS work independently from the ARM[®] core. This allows the ARM core to run a HLOS such as Embedded Linux[™] or Windows[®] CE Embedded with access to the independent realtime interfaces in the PRUSS.

Other PRU applications include additional UARTs (SoftUART). The AM1808 MPU already provides three UARTs as on-chip peripherals. However, certain applications such as an IO link master require 4 or 8 UARTs. In this case, the ARM core can work with the PRUSS and the McASP peripheral to make additional SoftUARTs available. Mistral (www.mistralsolutions.com) offers a SoftUART driver for download from their website. It supports Linux and can run up to four additional SoftUARTs per PRU.

Mistral has also developed a CAN bus driver based on the PRUSS, which emulates a CAN interface using the PRU inputs and outputs. This is particularly well suited for customer internal-bus systems with reduced transmission speeds. The driver is available with documentation for download from Mistral on their website.

It is also possible to integrate complex realtime interfaces such as PROFIBUS in the PRUSS. TI already offers a certified PROFIBUS slave solution with the AM1810 microprocessor.

The PRUSS is also included in other TI microprocessors and digital signal processors (DSPs). For more information on the PRUSS, see the **TI Wiki Page**, and the product pages for the **AM1808**, **TMS320C6748** and **OMAP-L138** processors.

Links:

1. Combining the ADS1202 with an FPGA Digital Filter for Current Measurement in Motor Control Applications: focus.ti.com/lit/an/sbaa094/sbaa094.pdf
2. PRUSS Wiki Page: processors.wiki.ti.com/index.php/Programmable_Realttime_Unit_Subsystem
3. AM1808 MPU: focus.ti.com/docs/prod/folders/print/am1808.html
4. AM1810 MPU with integrated PROFIBUS: focus.ti.com/docs/prod/folders/print/am1810.html
5. OMAP-L138 processor: focus.ti.com/docs/prod/folders/print/omap-l138.html
6. TMS320C6748 processor: focus.ti.com/docs/prod/folders/print/tms320c6748.html

Important Notice: The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI's standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer's applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company's products or services does not constitute TI's approval, warranty or endorsement thereof.

All trademarks are the property of their respective owners.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
RF/IF and ZigBee® Solutions	www.ti.com/lprf

Applications

Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Transportation and Automotive	www.ti.com/automotive
Video and Imaging	www.ti.com/video
Wireless	www.ti.com/wireless-apps

TI E2E Community Home Page

e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2011, Texas Instruments Incorporated