Modular Test Code for Microcontroller Units with ROM

Lincoln R. Nunes[1], Arthur H. C. Oliveira[2], Alexandre S. Santiago[2] and Rubens Takiguti[2]

Motorola
Semiconductor Products Sector – SPS
[1] Somerset Design Center
7700 W Parmer Ln
Austin - TX - 78729 - USA
[2]Brazil Semiconductor Technology Center - BSTC
Rodovia SP-340, km 128,7-A - Tanquinho
13.820-000 - Jaguariúna - SP - Brazil

Abstract

This paper describes the concept of a modular test code developed to reside in Microcontroller Units (MCUs) with Read-Only Memory (ROM) to make easier the test bench evaluation of integrated circuit (IC) prototypes designed at Brazil Semiconductor Technology Center (BSTC). MCU standard input/output ports are used to select independent operating modes and to transfer data from/to a host-computer. At test bench, limited instrument resources and MCU Random-Access Memory (RAM) small size make it difficult the use of the MCU built-in test mode for evaluation purposes. As the modular test code does not depend on the MCU built-in test mode to run, it is a useful tool for testing internal device modules functionally during prototype stage. This modular test code is a complement to the IC test before the Automatic Test Equipment (ATE) program is ready.

I - INTRODUCTION

Microcontroller Units (MCUs) are devices which integrate into a single chip Central Processor Unit (CPU) and several modules such as RAM, ROM or another type of non-volatile memory, Serial Communications Interface (SCI), Analog-to-Digital (A/D) Converter, Low-Voltage Inhibit (LVI), Computer Operating Properly (COP), Input/Output (I/O) Ports, Timer, Clock Generator, etc. Depending on the MCU type, some of those modules are missing or more specific ones are included. Fig. 1 shows a generic MCU block diagram.

Motorola MCUs have their built-in test mode called Monitor ROM (MON), allowing a complete test of the device through a standard RS-232 serial interface. The built-in MON receives and executes commands from a host-computer and is able to access any memory address. The MCU can write and execute, in built-in MON, host-computer code in RAM and in Electrically-Eraseable Programmable ROM (EEPROM) while all its pins (except the serial communication pin for built-in MON) retain normal operating mode functions.

At production sites, most of the Motorola MCUs are tested through built-in MON, using ATEs. Lots of test patterns are required which, in some cases, are very long (in number of cycles and time). But ATEs have powerful resources like number of channels and timing generators, memory depth, development and debugging tools, etc. Within this environment, production test program is developed and actual prototypes are used to debug it.

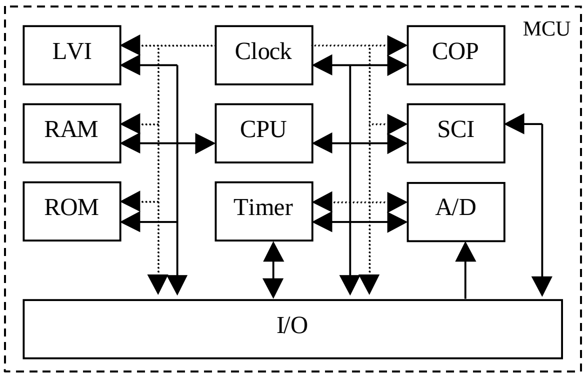

Fig. 1 – Generic MCU Block Diagram

At test bench instead, power supplies, signal generators, oscilloscopes and logic analyzers resources are limited compared to ATEs for evaluating MCU prototypes.

Different from MCUs with EEPROM, MCUs with ROM can not be reprogrammed and, for that reason, need to execute some test code in RAM through built-in MON for testing internal device modules functionally. As RAM size of MCUs is generally small, it is not possible to implement more complex test codes for MCUs with ROM using that approach.

To overcome the limitations described before, a modular test code was developed to reside in MCUs with ROM, aiming at the functional evaluation of devices at test bench during prototype stage.

II - BASIC CONCEPTS

The modular test code consists of code routines written as modules and integrated into a main program. Each module defines an operating mode and is responsible for testing functionally some internal modules of the MCU under test.

Operating modes run independent from each other in order to allow the evaluation of as many MCU modules as possible even if one or more modules have failures (with the exception of the CPU and ROM modules). In some cases, one MCU module is used to stimulate another one to make the evaluation easier or faster. Also, some MCU modules are exercised in more than one operating mode.

Operating mode selection is done by connecting specific digital input port pins of the MCU under test to low or high level before powering-on it.

III - OPERATING MODES

A simplified flow chart of the modular test code for MCUs with ROM is shown in Fig. 2.

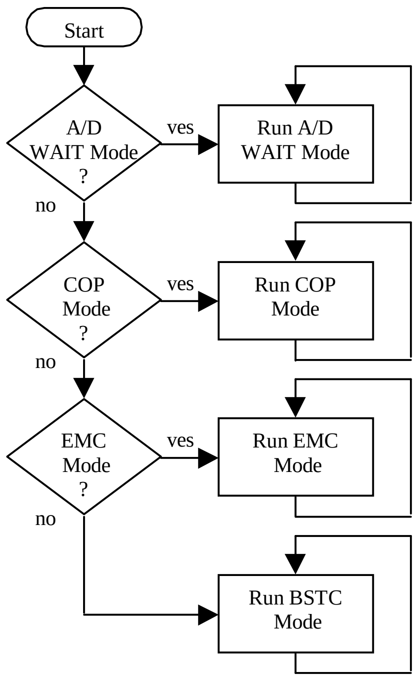

Fig. 2 - Modular Test Code Flow Chart

Four operating modes are available: A/D Wait, COP, BSTC and EMC.

A/D WAIT Mode

This operating mode intends to evaluate the internal A/D Converter of the MCU under test. By using, a host-computer, a single byte stating how many conversions to perform is sent to the MCU through its SCI. After receiving it, the MCU goes to its WAIT state and starts converting the analog voltage applied to a specific analog input port. When each conversion is completed, an Interrupt Request (IRQ) is generated bringing the MCU to its normal state and sending back the converted byte.

While in WAIT state, the MCU has its CPU clocks disabled and peripherals ones still running, allowing the A/D Converter to work at a low-noise condition necessary for characterization purposes.

There is an option, selected by connecting a specific MCU digital input port pin to a proper level, in which the converted bytes are sent continuously to the host-computer through the SCI, speeding up the communication.

By calculating the mean value of the converted bytes for each analog input voltage applied, it is possible to create Differential Non-Linearity (DNL) and Integral Non-Linearity (INL) graphs, for example, verifying the A/D Converter performance completely.

COP Mode

This operating mode holds the MCU under test in an endless null loop while the COP module is enabled. When the COP free-running counters overflow, an internal reset is generated by the MCU which asserts the Reset (RST) pin.

By monitoring the RST pin externally, it is possible to check the COP functionality by measuring time-out period. The main reason for implementing the COP mode is that the COP module, in MON mode, is disabled and therefore not testable.

EMC Mode

This operation mode verifies EMC susceptibility of the MCU under test. Through a standard digital output port, a single pulse is generated to trigger an external Schaffner generator responsible for creating a high-voltage noisy environment around the MCU ports. After that, a loop which exercises sequentially CPU, RAM, SCI, I/O Ports and Timer is run. Some MCU standard digital output ports are connected to LEDs for signalizing the current stage of the loop.

In case of an accidental reset of the MCU caused by COP, LVI, invalid instruction or address, the same LEDs indicate this occurrence and the MCU is halted.

BSTC Mode

This operating mode is used to exercise at the same time several internal modules of the MCU under test. The Timer is programmed to generate, through a standard digital output port, a Pulse Width Modulation (PWM) signal whose duty-cycle is proportional to the A/D conversion of the voltage applied to a specific analog input port. All RAM contents are sent out through the SCI continuously. Also, an alternative debug routine was developed using standard digital MCU ports and is available for use, allowing the same functions as built-in MON.

Alternative Debug Routine

The built-in MON already mentioned is responsible for a basic serial communication with a development or test system. Its serial protocol uses RS-232 format. A pre-defined communication frequency though brings limitations in some cases.

In a non-fixed frequency system it is necessary to provide an alternative protocol not strongly frequency dependent like RS-232. Examples of non-fixed frequency systems are low-power applications where the system frequency is lowered during stand-by and internal clock generators where locking may take a considerable number of cycles before stabilizing.

A simple, yet effective, protocol was then written and implemented in the test code to cover such cases.

The new protocol is software based and uses 4 signals to establish communication:

- SDIN - MCU under test data input
- SDOUT- MCU under test data output
- SCLK - data sample clock
- STRI - command execution trigger

Command and data are serially sent from host to the MCU through SDIN and sampled with SCLK. A FIFO receives and stores the bits sent from the host. SDOUT has the oldest bit in the FIFO.

When all bits are stored in the FIFO, the upper bits are taken by the MCU as command and the eight lower bits is the associated data byte. The size of the FIFO can easily be modified to support different ranges of command and data.

A similar FIFO is in the host side. If the host wants to read data from the MCU, the serial shift will be able to bring the MCU data through SDOUT.

When the complete pair command/data is loaded in the MCU FIFO, a STRI pulse from the host will start the command execution in the MCU.

Read/write transfer speed is lower in this implementation compared to RS-232. The speed is limited to the detectability of SCLK and STRI by the MCU test code. On the other hand robustness is achieved with the verifiable communication link (SDIN/SDOUT chain), frequency independence (SCLK commanded) and execution control (STRI pulse).

An effective set of commands completes the alternative debug routine (for example, read / write all accessible addresses, execute pieces of user code or RAM loaded code).

The alternative debug routine can easily be included inside a customer code if desired. MCUs with ROM can be particularly hard to test if some trouble comes up and the developed debug routine can be of great help.

IV - RESULTS

The modular test code was used in 8-bit ROM MCUs developed at Brazil Semiconductor Technology Center (BSTC) of Motorola. Its size is around 1.5 kBytes. Those MCUs targeted applications such as appliances, general purpose and PWM motor control.

A LabVIEW™ application running in a host-computer was developed to support the characterization of a 10-bit A/D Converter internal module. This application communicates with the MCU under test through RS-232 and controls test bench instruments through General Purpose Interface Bus (GPIB). Fig. 3 shows the test bench block diagram used for characterizing this MCU A/D Converter module using a evaluation board.

In this case, the MCU under test is operating in the A/D Wait mode. Jumpers are used to configure the operating mode and LEDs signalize MCU state. Power Supply is responsible for powering all evaluation board circuits. The LabVIEW™ application programs the Precision Voltage Source for sweeping from 0 to 5V, with 0.5mV increments and, for each voltage applied to the MCU A/D input pin, reads bytes with 10 converted values sent out from MCU through

Serial Interface. A Voltmeter measures exactly the voltage at an MCU A/D Converter input pin while an Oscilloscope monitors transmit/receive serial data. The complete characterization of this A/D Converter took 20 minutes approximately.

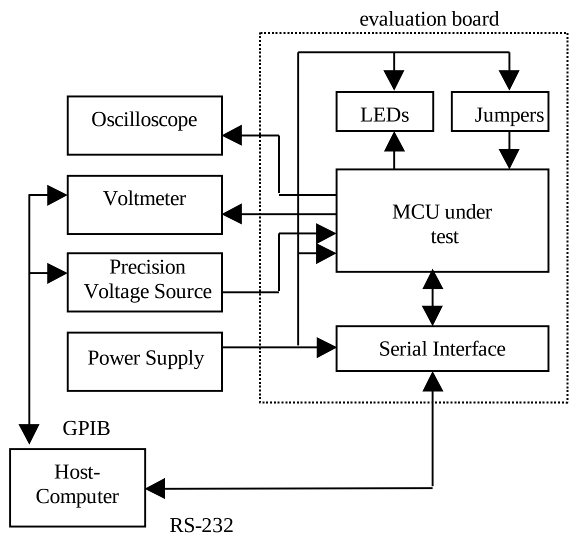

Fig. 3 - Test Bench Block Diagram

Also, the modular test code proved its effectiveness in a chip with a failing Clock Generator module (non-locking frequency). Using the routine it was possible to test the other modules, allowing their functional validation that would otherwise be unreachable through regular RS-232 communication through MON mode. That ability allowed a more reliable correction run of silicon with all associated savings of time, resources, mask sets, lot reruns, overall costs.

The modular test code is programmed at the initial MCU ROM prototypes, before a customer application code availability. Once the new device is characterized, tested and qualified it is ready for production and customer use.

V - CONCLUSIONS

The implementation of this concept of modular test code in ROM of MCUs proved to be extremely valuable on BSTC designs, allowing a faster prototype evaluation without the need of complex equipments and generating data for the production test code development. Functionality of MCU modules, were proved during debug with test code, indicating failing modules early during prototype stage and allowing a fast correction of designs. It is a complementary and alternative way to Motorola built-in MON and a fast evaluation solution while ATE test program is being developed.