TOP

Verification Guide

Version #: **1.0**
Revision date: **October 11, 2010**

Version Control

Revision	Authors	Date	Comments
0.1	Harney Abrahim	08/Oct/2010	Initial Version
1.0	Vinícius Amaral	11/Oct/2010	Verification plan revised. Improved feature descriptions.

Table of Contents

TOP EMC08 DIGITAL MODULE

1. Introduction

The purpose of this document is to define the general terms for verification of Top Digital of Microcontroller EMC08 project. It is part of the training phase II in the Cadence IC flow in CT2. This document includes checking and coverage description, general features respect to verification environment. Added, there are the features and it description to be verified.

1.1 Verification Strategy

The methodology used is the eRM (e Reuse Methodology) that provides a template for partitioning the environment. This strategy codifies the best practice for developing e-based verification and ensures consistency, reusability and extensibility. The Specman tool will be used. It compiles and debuggers the E language.
E language is used for basically describe the verification environment, which has the capability to generate random test vectors, interface with code RTL to verify the functionality of the code.

1.2 Verification Flow

In order to generate the EMC08 instructions stimuli, the assembly language will be used. After create some set of instructions in assembly, the output of these instructions will be verified in the checkers. To do it, the assembly code needs to be translated in a hexadecimal file understood by the memory verilog. The following figure illustrates the diagram of verification flow, where the DUT (Design Under Test) is the TOP EMC08 block that we want to verify.

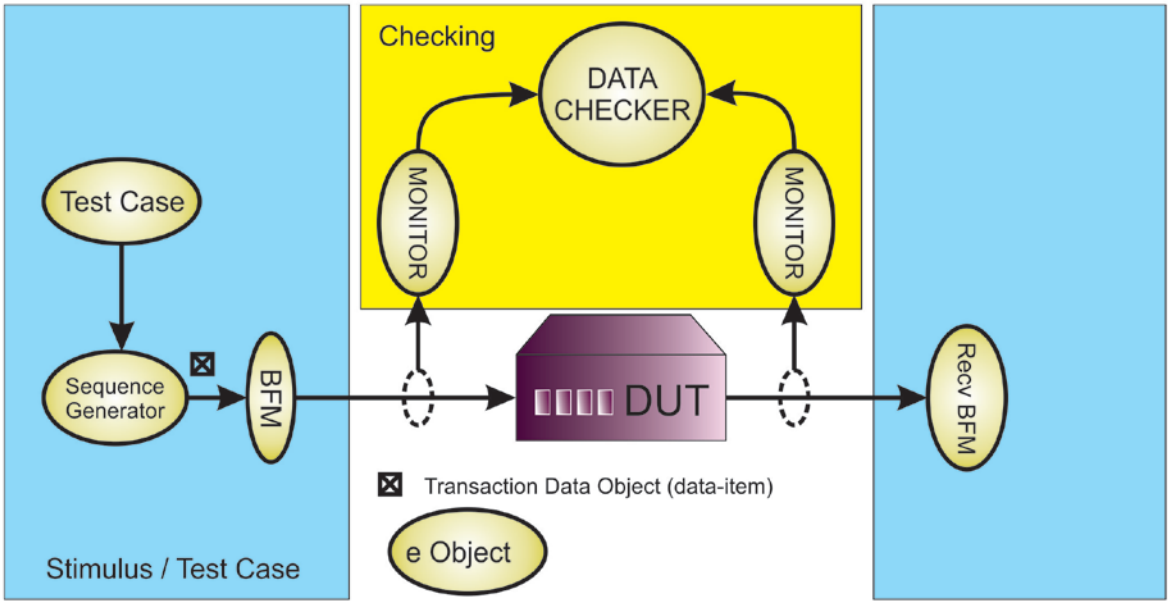

Figure 1 - Verification Flow Diagram

The DUT receives stimulus from input BFM (Bus Functional Model) and sends outputs to output BFM. These stimuli are generated by Sequence Generator according to whished Testcase that can control the order of sequences, types of stimulus, number of repetitions and so on. There are two

monitors that sample input and output DUT signals. They are passive units that do not change the values of signals. Monitors send sampled signals to Data Checker that compares DUT outputs with expected answers and gives a DUT error if it doesn't mismatch.

A functional model receives the inputs from IN Monitor, synchronizes the DUT with the verifications environment, calculates the expected results and compares these results with DUT outputs, send to Data Checker by OUT Monitor. So, Data Checker compares DUT's results with expected and creates illegal buckets when they mismatch. The illegal buckets generate a DUT error.

1.3 Random Simulation

The testcases will be generated using a set of sequences. Depending on feature to be verified, the testcase controls the generation order of these sequences. The following table shows the action executed by each sequence. The testcase can generate any number of sequences in a random order.

Sequence	Action executed
SEQ_RESET	Generates a Reset signal random sequence.
SEQ_INST_RAND	This sequence sends a random instruction.
SEQ_INST_XXX	This sequence sends a valid instruction (where XXX is the instruction name) of the instruction set.
SEQ_ROM_RAM SEQ_RAM_RAM SEQ_RAM_ROM SEQ_RD_WR	Generates sequences that enables the access to the external memories ROM and RAM as much for reading as for writing at same time.
SEQ_SERIAL	Sends a set of instructions simulating a serial transmission and reception.
SEQ_TIMERS	Sends a set of instructions simulating a Timer use in all modes of operation.
SEQ_INTERRUPT	Sends a set of instructions simulating the interrupt use in all modes of operation by software and by hardware.
SEQ_PORTS	Sends a set of instructions simulating the ports use in all modes of operation for both internal block accesses and external.
SEQ_BAUD_RATE	Sends a set of instructions simulating a Baud Rate use in all modes of rate operation.
SEQ_TIMERS	Send a set of instructions simulating the Timers use in all modes of operation.

Table 1 – Sequences

2. References

- Project Definition, EMC08 SOW, Stated of work, Version 1.3 September 22, 2010
- Design Verification with e, Samir Palnitkar, Pretince Hall 2004
- BD02: Digital IC Design, Student Handout, Version 1.1, September 23, 2008
- Specman® Elite Basics for Verification Environment Developers Version 6.1.1, Lecture Manual, November 2, 2007.

3. Verification Plan

The Verification Plan describes the verification process that will be followed to verify top digital module. It details a list of required verifications that must be accomplished to validate the design specification.

In the verification environment, the following functions will be built:

Generation

Input stimuli are generated based on the designed constraints.

Driving Stimulus

After the test vectors are generates, they must be driven on the DUT, provide a simulator interface and the necessary mechanism to drive the DUT.

Collecting Output

After the stimulus is applied to the DUT, output is produced from the DUT. This output must be collected and checked.

Data Checking

After the output data are received from the DUT, the data must be checked. Data value checks compare the output data values against the expected data.

Coverage

Verification if the test plan goals have been met with the analysis the functional coverage results. Basic item coverage, transition item coverage, and cross coverage.

3.1 Design Details

This section describes basic details about verification plan and the design that will be verified. The Table 2 gives this information.

Component	Feature Description
Document Name	Top Verification Plan
Verification Methodology	e-RM (Bottom-up and Top-Down approach)
Verification Methods	Simulation
Simulation Components	
Application	Cadence Specman® Elite Version 6.1
Language	*e* language
Testcase file	EMC08/digital_A/top/verification/testbench/top_test.e
Bus Functional Model	EMC08/ digital_A/top/verification/testbench/top_bfm.e

Table 2 – Design Details

3.2 Feature List

The Table 3, shown below, lists the critical features of design that must be verified, including a description of the expected functional and the priority of each feature.

Feature Number	Feature Description	Testing Priority
EMC.Top.F1	The reset need to work property for all sub-blocks	10
EMC.Top.F2	The memories RAM and ROM need to be read and write by the core in the right time. The Lower 128 bytes of RAM which can be accessed by both direct and indirect addressing can be divided into 3 segments: Register Banks 0-3; Bit Addressable Location; and General	10

	Purpose Area.	
EMC.Top.F3	The SFR space can be accessed only by direct addressing, occupying the block of addresses from 80H through FFH.	10
EMC.Top.F4	The Serial block need to work in both modes parallel with other instructions without conflicts with other instructions	8
EMC.Top.F5	The Baud Rate block needs to generate the rates for serial block with the correct timing as the Machine Cycle for other blocks.	10
EMC.Top.F6	The Ports blocks need to set correctly the ports operations with the correctly timing according to the instructions, without conflicts.	8
EMC.Top.F7	The external memory access needs to be controlled correctly by the Bus Control without conflicts with the internal access.	8
EMC.Top.F8	The Timers block needs to provide the correct count to the different operation modes synchronized with the CM and clock.	8
EMC.Top.F9	The Interrupt control needs to treat correctly interrupts in a property timing.	10

Table 3 – Feature List

3.3 Verification List

The verification list, described on Table, defines how the features (on Table 3) will be verified. This list contains all verifications that must be run to verify expected features.

Test number	Description	Feature Verified	Priority	Owner	Completation
EMC. Top.F1.V1	Verify if all sub-blocks are resetting correctly with a correct timing.	EMC. Top.F1	10	Harney	0%
EMC. Top.F2.V1	Verify the read and write operations in both RAM and ROM according to the read and write time.	EMC. Top.F2	10	Harney	0%
EMC. Top.F3.V1	Verify the timing and addressability to read and write in the SFR by the core and the internal blocks.	EMC. Top.F3	10	Harney	0%
EMC. Top.F4.V1	Verify the Serial block in mode0 (tx or rx) working in parallel with other instructions without conflicts	EMC. Top.F4	8	Harney	0%
EMC. Top.F4.V2	Verify the Serial block in mode2 (tx or rx) working in parallel with other instructions without conflicts	EMC. Top.F4	8	Harney	0%
EMC. Top.F4.V3	Verify the Serial block in mode2 (full-duplex) working in parallel with other instructions without conflicts	EMC. Top.F4	8	Harney	0%
EMC. Top.F5.V1	Verify the Baud rate Machine Cycle generation timing, and the rate modes	EMC. Top.F5	10	Harney	0%
EMC. Top.F6.V1	Verify all the ports by setting instructions to access the same port.	EMC. Top.F6	8	Harney	0%
EMC. Top.F6.V2	Verify with the data of a random instruction is being writing or reading in the ports property	EMC. Top.F6	8	Harney	0%
EMC. Top.F7.V1	Verify the external access of RAM by setting instructions to do it. The bus control need to control it property respecting the priority of each access	EMC. Top.F7	8	Harney	0%
EMC. Top.F8.V1	Verify the timers sync with the CM	EMC. Top.F8	8	Harney	0%
EMC.	Set timers to operate in mode 0 and 1 and	EMC.Top.F8	10	Harney	0%

Top.F8.V2	verify the correct counting, the overflow instant and the timer interrupt flags.				
EMC. Top.F8.V3	Verify the timers mode 2 operating by varying the PHT input	EMC. Top.F8	8	Harney	0%
EMC. Top.F9.V1	Verify the hardware interrupt control by forcing it occurs	EMC. Top.F9	10	Harney	0%
EMC. Top.F9.V2	Verify the software interrupt control by forcing it occurs	EMC. Top.F9	10	Harney	0%

Table 4 – Verification List

3.4 Top Feature 1 Description

The feature 1 of TOP verification is to verify if the reset is working property for all sub-blocks synchronously with the clock (positive and negative edge).

3.4.1 Verification Steps for F1.V1

For this feature, it will be stimulated the reset signal in a random way to stimulate all the sub-blocks to reset. It is expected that all sub-blocks goes to the initial state (this initial state can be found in the block guide of each sub-block).

3.5 Top Feature 2 Description

Since the access of ROM and RAM are critical in a design, the access of it by the core need to be done property in a correct timing. The Lower 128 bytes of RAM which can be accessed by both direct and indirect addressing can be divided into 3 segments: Register Banks 0-3; Bit Addressable Location; and General Purpose Area. This feature it is intended to verify the timing and addressability of memory accesses.

3.5.1 Verification Steps for F2.V1

For this feature will be stimulated a set of instructions to access both ROM and RAM. Since the access occurs, the checkers will verify the access timing as the data to be read or write. The operation of read and write needs to work property with the constraint access timing.

3.6 Top Feature 3 Description

The SFR space can be accessed only by direct addressing, occupying the block of addresses from 80H through FFH. The feature 3 it is intended to verify the SFR read and write operation, since the timing and addressability need to be respected.

3.6.1 Verification Steps for F3.V1

For this feature will be stimulated random instructions to stimulate all sub-blocks to do some SFR read and write. The checkers need to verify if the SRF read and write operations are according to specification.

3.7 Top Feature 4 Description

The Serial block is responsible for receive and transmit data through the serial ports. It has the synchronous and asynchronous mode and can work with 8 (synchronous) or 9 (asynchronous) data bits.

The serial blocks have a special register called SCON which controls the serials modes and interrupts. Added, there is a serial buffer register (SBUF) which transmits and receives registers physically separately.

It is possible to set tree different baud rates depending on the operation modes. All of them are fixed rates according with the machine cycle.

The feature four it is intended to verify the Serial block functionality. This sub-block needs to work in parallel with other instructions in both operational modes. In mode 2, the serial needs to transmit and receives in a full-duplex way, while other instructions are being treated by the core.

3.7.1 Verification Steps for F4.V1

For this feature will be stimulated a set of instructions simulating a Serial transmission and reception in mode 0. Since this mode is half-duplex, two different sets of instructions need to be stimulated. One for the transmission and one for the reception. After set the Serial mode, other random instruction will be stimulated and need works in parallel with the Serial. The checkers will verify with there are no conflict among the random instruction and the serial.

3.7.2 Verification Steps for F4.V2

For this feature will be stimulated a set of instructions simulating a Serial transmission and reception in mode 2. Since this mode is full-duplex, three different sets of instructions needs to be stimulated. One for the transmission, one for the reception and one for both at the same time (other feature item). After set the Serial mode, other random instruction will be stimulated and need works in parallel with the Serial. The checkers will verify with there are no conflict among the random instruction and the serial.

3.7.3 Verification Steps for F4.V3

For this feature will be stimulated a set of instructions simulating a Serial transmission and reception in mode 2. The full duplex functionality will be stimulated simulating a transmission and reception at the same time. After set the Serial mode, other random instruction will be stimulated and need works in parallel with the Serial. The checkers will verify with there are no conflict among the random instruction and the serial.

3.8 Top Feature 5 Description

The baud rate module is responsible for the baud rate generation for the serial block and Machine cycle generation.

All generations occurs through the clock frequency division. In this implementation it works in two modes of operation, both fixes, according to the specification.

This feature it is intended to verify the Baud Rate sub-block signal generation. This block will generate the serial rates and the machine cycle to the other sub-blocks. Since the Machine cycle it is an important reference signal, the timing to be generated it need to be verified.

3.8.1 Verification Steps for F5.V1

For this feature will be stimulated a reset signal to begins the Machine Cycle generation. The checkers need to verify the timing of this signal (if it is according with the core functionality). Added, a set of instructions to change the serial rate will be stimulated. The checkers need to see if these rates are according to the specifications (1/32, 1/64, …).

3.9 Top Feature 6 Description

This module feature is responsible for verify the ports functionality.

The output drivers of Ports 0, 2 and 4, and the input buffers of Port 0, are used in accesses to external memory. In this application, Port 0 is bidirectional data bus. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise the Port 2 pins continue to emit the P2 SFR content.

Only Port 3 is multifunctional and serves functions of various special features.

The alternate functions can only be activated if the port direction is set to input for P3.0 and P3.2 and P3.3. For P3.1, it's necessary to set the I/O direction to output (P3EN) .

The data to be read and write needs to be right according to the instruction stimulated.

3.9.1 Verification Steps for F6.V1

For this feature will be stimulated a set of instruction to read or write in the same port. The checkers need to verify if the priority list is respected.

3.9.2 Verification Steps for F6.V2

For this feature will be stimulated a set of instructions to read and write random data in the ports. The checkers needs to verify if these data are in the right ports according to the instructions send.

3.10 Top Feature 7 Description

The Bus Control module is responsible to management of external signals of EMC08, they are: EA_b and PSEN_b.

The read strobe to external ROM, PSEN_b is used for all external program fetches. PSEN_b is not activated for internal program fetches.

The lowest 4K bytes of Program Memory can be either in the on-chip ROM or in an external ROM. This selection is made by strapping the EA_b pin to either VCC or Vss. This project uses only 4K bytes of ROM internal.

This feature is responsible to verify the external memory access controlled by the sub-block Bus Control.

3.10.1 Verification Steps for F7.V1

For this feature will be stimulated a set of instructions to stimulated the external memory access. Since the access is done, the checkers needs to verify the priority access of each write or read operation.

3.11 Top Feature 8 Description

The microcontroller EMC08 has 3 timers (timer 0, timer 1 and timer 2), being two on general purpose (timer 0 and 1) and an on specific purpose (timer 2).

The timers 0 and 1 can assume the timer function or counter depending on the configurations attributed to the same by the application (software).

The timer 2 has his functionality focused for the section automotive being used as an angle counter in a jagged wheel in the which lacks a tooth that through his occurrence allows to synchronize the counting previously stored in a register with the current counting obtained by a turn of the jagged wheel, being then that validated result and stored in a register, and through these data stored in the register makes possible the counting of turns in the motor of the vehicle (RPM) and it provides to the system (CPU) to evaluate the automobile is accelerating or slowing down so that of ownership of those registrations to increase or to reduce the flow of injection of combustible mixture and the speed of the ignition, that system FlyWheel is called.

This feature is responsible to verify the timers synchronization and the mode 2 operation, since it is critical for this project.

3.11.1 Verification Steps for F8.V1

For this feature will be stimulated the reset signal and the change of operation modes to synchronize the timers (with the Machine Cycle). The checkers needs to see if the timers are synchronously with the CM.

3.11.2 Verification Steps for F8.V2

For this feature will be stimulated a set of instructions to verify the operation of Timers in mode 0 and 1. Random signal will be stimulated and checkers needs to see if the timers are counting property and overflow flags are set correctly.

3.11.3 Verification Steps for F8.V3

For this feature will be stimulated a set of instructions to verify the Timers mode 2 of operation. Random signal will be stimulated in the PHT input simulating the Digital Flywheel Tooth sensor.

3.12 Top Feature 9 Description

The Interrupt Controller module evaluate and decides whether an interrupt request must be generated to CPU.

This module can monitor up to 8 interrupt sources. These sources are Timer 0, Timer 1, Timer 2 , Serial Communication Port (transmit and receive), External Pin 0, External Pin 1 and Transceiver.

Interrupt sources can be individually configurable through IE and IP registers.

This feature is responsible to verify the interrupts operations. Both software and hardware interrupts need to be stimulated and be accomplished with the priority list.

3.12.1 Verification Steps for F9.V1

For this feature will be stimulated a set of instructions to generate a hardware interruption. The EMC08 provides 8 interrupt sources

The checkers needs to see if the priority list is respected and if the interrupt treatment is right.

3.12.2 Verification Steps for F9.V2

For this feature will be stimulated a set of instructions to generate a software interruption. The checkers needs to see if the priority list is respected and if the interrupt treatment is right.

4. Testbench

The testbench of the TOP module will be used for the functional verification of the design and for connect the DUT with the verification environment. Added, it will be used to stimulate the Design Under Test (DUT) and to capture the results generated by the DUT. The mixed approach (top-down + bottom-up) will help to verify first the critical part of the design (core, bus, etc.). After the other blocks (serial, baud rate, etc) will be added to the environment and verified.

4.1 Testbench Overview

In the simulation for TOP module, only one DUT will be verified. The entity monitors samples signals and extracts events, collects transaction, and data items. The BFM are an active entity which samples and drives the signals to stimulates the DUT.
The BFM and monitor should be independent even if they are monitoring the same signals. The monitor should never depend on the BFM, because there may be situations that do not require a BFM. Figure shows the block diagram of the testbench.

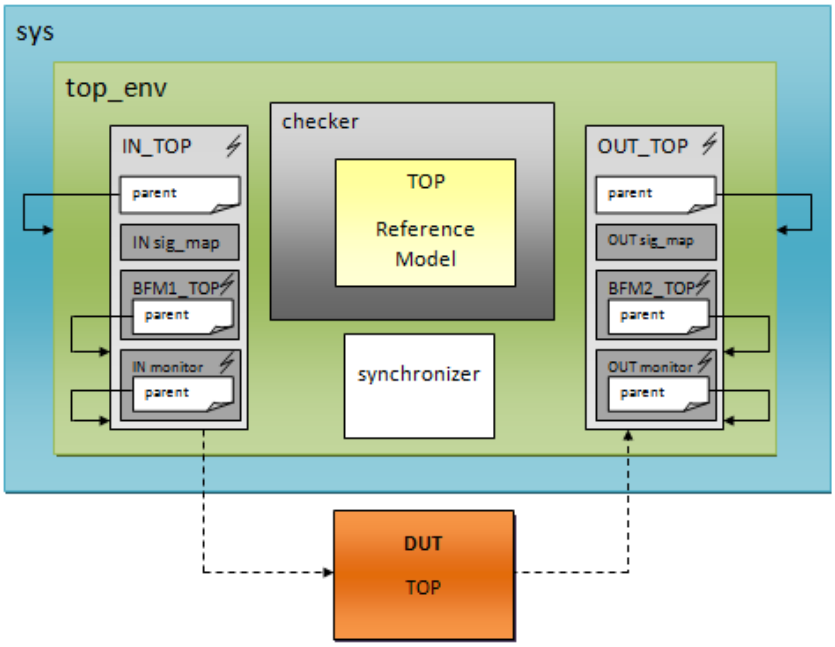

Figure 2 - Block Diagram of testbench

4.2 Partitioning

N/A Since this guide is to verify the TOP block.

4.3 Global Routines

The *e*RM methodology defines all global routines. See *e*RM methodology reference.

4.4 External Interface Functions

The *e*RM methodology defines all external interface functions.

4.5 Memory Map

The SOW (Statement of the Work) defines the memory map.

4.6 Verification and Debug Registers

To be defined. Debug registers were not defined yet.

4.7 Clocking

For define if it will be an extra sign to indicate the cycle of machine. At the beginning, all the modules will hardly have a sign of the global clock.

4.8 Reset

For the specifications of the project it will be active in 0 synchronous with the clock.

4.9 Termination

The *e*RM methodology defines all termination tasks.

5. Drivers and Monitors

The driver object performs the function of taking the stimulus data objects one at a time and applying them to the DUT until all stimuli has been applied. Typically, the driver object is similar to an HDL bus functional model. In case of a networking environment, it is also known as a port object.

Monitor is a unit instance that passively monitors (looks at) the DUT signals and supplies interpretation of the monitored activity to the other components of the agent.

The *e*RM methodology defines drivers and monitors.

5.1 Driver and Monitor Overview

In the TOP environment only one monitor and one driver will be implemented. The different stimulus will be provided by the different generated sequences. Each sequence will stimulate different design functionality.

The monitors will be passive units which can emit events when they notice interesting things happening in the DUT or on the DUT interface. They can also check for correct behavior or collect coverage.

6. Models

After the simulation in the TOP module is completed, coverage output is produced. Coverage output from Specman Elite can be viewed graphically or in a text file. Coverage output can also be accumulated over multiple test runs to view cumulative coverage using ICCR tool for example.

6.1 Functional Models

For the TOP verification a functional model will be implemented according to the operation inputs. Since the microcontroller can work in many ways, some common functionality will be mapped to build a functional model to verify it.
After the main functionalities are working property, it is intended to build a general functional model to verify randomly any input.

6.2 Memory Models

The TOP verification will adopt a binary file to both ROM and RAM memory models. The ROM file will be generated by a INTEL 8051 compiler which translates the mnemonic instructions in binary instructions understood by the design, since the instructions set are the same. The RAM model will be a binary file generated randomly.

6.3 Stub Models

Specman Elite and the Simulator talk to each other through an interface that includes a special file called a stubs file. In addition to the stubs file, Specman Elite and the Simulator also communicate through mechanisms such as Verilog Programming Language Interface (PLI) or the VHDL Foreign Language Interface (FLI).

For this project the testbench will be implemented as a stub file.

6.4 IO and Pads

TBD.

7. Additional Information

Not applicable.