Rev. 1
June 2010
WinPath3_MemTest_ApNote.odt

Application Note

Abstract

It is very important to ensure the integrity of the memory in all designs. Processor based memory tests alone are not adequate. Wintegra provides a data path software (dps) based stress test that must be used. This application note details the steps required to configure and run the memory stress test application. The application is based upon WinPath software and the WinPath 3 Packet Processor.
This application note is accompanied by software. The test software includes functionality to allow execution using linux. It can easily be ported to any operating system.

Table of Contents

1.0 Introduction

It is very important to ensure the integrity of the memory in all designs. Processor based memory tests alone are not adequate. The memory stress test application for the WinPath3 contains two different testing modules. The first module is a standard processor based that checks the basic functionality of the data buses. A second module stress tests the memory buses by spawning multiple threads that will transfer data between the specified source and destination addresses simultaneously. Users should note that these memory tests will perform a series of write commands which will overwrite data that was originally stored in the memory region tested. The stress test cannot be executed in conjunction with a WDDI application.

2.0 Terminology and conventions used in this document

The following conventions are used throughout the AppNote

Code examples are provided using courier font.

Example: `error_code = WP_IntEnable();`

When used in document text other than code examples, reserved user-visible function names, data structures, types and constants are indicated in bold, universal font format.

Example: **WP_DeviceDelete( )**

3.0 Configuring the Memory Stress Test

This section will list the information needed to configure the memory stress test application to run on your board.

3.1 Supported Hardware and Software

This version of the memory stress test application is only compatible with the WinPath3. There is a separate version available for the WinPath2. The WinPath3 application will allow external processors other than the MIPs processor to run the testing modules.

toolchain

This stress test was tested using using the CodeSourcery toolchain under WDDS 4.0. The user may have to install CodeSourcery and WDDS 4.0, and set the required environment variable as shown by following example code:

```
export
WP_PATH_LINUX_MIPS_WINMON_WINPATH3=/home/user/CodeSourcery/mip
s-linux/bin:/home/danny/CodeSourcery/mips-
elf/bin:/home/user/CodeSourcery/mips-linux/bin:$PATH
```

3.2 Required Addresses

The application is able to run the memory tests on four regions of memory: Parameter RAM, Packet RAM, Internal RAM, and Application RAM; thus the addresses for each of the memory regions must be known. Furthermore the location of the internal registers must also be known. The user will need to specify the memory addresses listed below.

The physical and virtual base addresses of the Parameter RAM.

The physical and virtual base addresses of the Packet RAM.
The physical and virtual base addresses of the Internal RAM.
The physical and virtual base addresses of the Application RAM.
The virtual base address of the Internal Register Space (RIF).

3.3 Required Memory sizes

The memory sizes of each of the four regions of memory will also need to be specified. It should be noted that when memory testing Internal RAM, only a small section of the entire Internal RAM memory size should be tested. Because the application itself uses Internal RAM, it is possible that the testing program will overwrite itself and cause an error. When testing the Application RAM, only unused memory should be passed into the application. The user will need to specify the memory sizes listed below.

Memory size of Parameter RAM.
Memory size of Packet RAM.
Memory size of unused Internal RAM.
Memory size of unused Application RAM.

4.0 Configuring the Memory Stress Test

A main.c file containing an example main() program is attached to the test package. The user can choose to either use the example program or compile the stress test within a separate program. Copy the attached files into a separate directory where you will build the stress test.

4.1 Configuring the example calling program.

The example calling program's default setup is to run both memory diagnostics and stress test on all memory regions. The default memory address and size definitions are configured for the WDS board using the MIPs processor. The user can change these definitions as needed within the main.c file in the section marked with “USER MODIFY”.

4.2 Configuring a separate calling program

To run the stress test under a separate calling program, the user must include memtest.h in their program and use the Mem_Test function to access the testing modules. The function prototype is shown below.

Mem_Test function Prototype:

```
void Mem_Test( int test_select,
           datum Phys_srcAddress,
           datum Virt_srcAddress,
           datum Phys_intSpace,
           datum Virt_intSpace,
           datum Virt_intReg,
           int siuBus,
           int nReps,
           int nWinFarms,
           datum memSize,
           char *msg)
```

Parameters:

test_select:	**PROC_TEST** = processor based memory diagnostics **STRESS_TEST** = memory stress test
Phys_srcAddress:	Physical base address of Memory region tested
Virt_srcAddress:	Virtual base address of Memory region tested
Phys_intSpace:	Physical base address of Internal space
Virt_intSpace:	Virtual base address of Internal space
Virt_intReg:	Virtual base address of Internal registers
siuBus:	**SIU_BUS_PARAM** = Parameter Bus, **SIU_BUS_PACKET** = Packet Bus, **SIU_BUS_INTERNAL** = Internal Bus, **SIU_BUS_HOST_APP** = Application Bus
nReps:	Number of test repetitions
nWinFarms:	Number of winfarms to run **
memSize:	Size of the memory bank tested ***
msg:	Debug message

**Note: If nWinFarms = 0, a special case will execute where only a single thread will be run for debugging purposes.
***Note: See Section 2.4

4.3 Additional Configurations

The memory stress test has a verbose mode which prints out additional details about the operations executed. Users can activate verbose mode by adding the following line to the beginning of their main program:

```
#define VERBOSE_MODE
```

In normal mode, the stress test will select a memory location to test for each active thread. However, the user may also run a more comprehensive stress test that will stress test the entire memory region passed in. This full memory test is activated by adding the following line to the beginning of the main program:

```
#define FULL_SCAN
```

Note that when selecting full scan decrease nReps to a much smaller number than shown in the example, 10 should be sufficient. Otherwise the test will take an extraordinary amount of time to complete.

5.0 Compiling the Example

The example program can be configured to run the memory stress test under either linux or Winmon. This section will detail the steps required to compile and run this application. A full list of attached files is shown below.

Name	Location	Contents
memTest.c	winpath3_examples/memtest/memTest.c	Test Functions
memTest.h	winpath3_examples/memtest/memTest.h	FunctionDefinitions
generic.h	winpath3_examples/memtest/generic.h	Generic Definitions
siu_image.c	winpath3_examples/memtest/siu_image.c	siu_ProgramImage

main.c	winpath3_examples/memtest/main.c	Example program
makefile.winmon	winpath3_examples/makefile.winmon	Winmon Makefile
makefile.linux	winpath3_examples/makefile.linux	Linux Makefile
make.list	winpath3_examples/memtest/make.list	Makelist

5.1 Compiling and running under Winmon

To run the stress test stand alone under Winmon, the user must edit the attached makefile.winmon to suit their system settings. Instructions which must be altered are marked by "`user modify`". The addresses and memory sizes defined in the main.c file must also be changed to fit the specifications of the user's board. The section that must be altered is marked by "`USER MODIFY (NON-LINUX ONLY)`". Users running the Winpath 3 SL board have different memory configurations and must add the following line to their main program:

```
#define WDS_SL
```

Once all changes are applied, the user can compile the program from command line. An example command line is shown below.

example command line:

```
make -f makefile.winmon example=memtest WPI_TARGET=mips_winmon
WPI_HW_DEVICE=winpath3 WPI_BOARD=wds
```

Once compiled, the user can now load the generated .exe onto WinMon and run the memory stress test as an executable.

5.2 Compiling and running under Linux

To run the stress test stand alone under linux, the user must edit the attached makefile.linux to suit their system settings. Instructions which must be altered are marked by "`user modify`". The addresses and memory sizes defined in the main.c file must also be changed to fit the specifications of the user's board. Users running the Winpath 3 SL board have different memory configurations and must add the following line to their main program:

```
#define WDS_SL
```

Once all changes are applied, the user can compile the program from command line. An example command line is shown below.

example command line:

```
make -f makefile.linux example=memtest WPI_TARGET=mips_linux
WPI_HW_DEVICE=winpath3 WPI_BOARD=wds
```

The attached program plots the WinPath3 memory through linux's mmap functionality. Therefore the main calling program will retrieve and load the required memory addresses and sizes automatically. After compiling the program, the user can now load the program into the WinPath3 linux environment and run the memory stress test as an executable.

6.0 References

7.0 Glossary

Document History

Ver.	Released	Description
1.0	4th July 2010	First Draft

Contact Information

Wintegra Inc.
6850 Austin Center Blvd.
Suite 215
Austin, TX 78731
USA
Tel: +1 (512) 345-3808

Wintegra LTD.
Taya Center
6 Hamasger St.
Ra'anana 43653
Israel
Tel: +972 (9) 743-9998

Wintegra
International House, Stanley Blvd.
Hamilton Int. Technology Park
Blantyre, Glasgow. G72 OBN.
United Kingdom
Tel: +44 (0) 1698-404889

Proprietary Notice