MIDAS: A Formally Constructed Virtual Machine

Stephen Wright

Department of Computer Science, University of Bristol, UK

stephen.wright@bris.ac.uk

Date	*Issue*	*Change*
4/6/2009	1	Original release

TABLE OF CONTENTS

1. Introduction

MIDAS (Microprocessor Instruction and Data Abstraction System) is a specification of an Instruction Set Architecture (ISA) capable of executing binary images compiled from the C language. It was developed to demonstrate a methodology for formal construction of various ISAs in Event-B via a generic model. It is intended to be representative of typical microprocessor ISAs, but using a minimal number of defined instructions, in order to make complete refinement practical. The intention is to simplify the number and complexity of the defined instructions at the cost of compiler complexity, run-time performance and code density, without compromising representativeness. There are two variants: a stack-based machine and a randomly accessible register array machine. The two variants employ the same instruction codes, the differences being limited to register file behavior. A pdf document giving an informal specification for both variants is available at:
http://www.cs.bris.ac.uk/Publications/pub_master.jsp?id=2001007

MIDAS is similar to the ISA of conventional microprocessors, and appropriate for hardware implementation However; it is intended to be initially implemented in software as an interpreting Virtual Machine (VM) executing on a host processor.

2. Related Publications

The following related publications are available:

- "Using EventB to Create a Virtual Machine Instruction Set Architecture", Abstract State Machines, B and Z, SpringerLink, 2008
- "Using Event-B to Create Instruction Set Architectures", [to appear in] Formal Aspects of Computing: Applicable Formal Methods, SpringerLink, 2009
- "Automatic Generation of C from Event-B", Workshop on Integration of Model-based Formal Methods and Tools, 2009
- "MIDAS Machine Specification", Bristol University http://www.cs.bris.ac.uk/Publications, 2009

3. Archive Description

The accompanying archive *MidasBundle.rar* contains two implementations of each MIDAS variant: a hand-coded C prototype and a detailed Event-B model capable being automatically translated to C via a supplied Rodin translation plug-in. A server providing program load and text output functions to all four VMs via a common protocol is provided. GCC compilers are supplied for each variant. All source code is included in the archive, allowing modification of either implementation of the ISA variants and their compilers.

The archive can be extracted using the *WinRAR* tool, available at:

http://www.rarlab.com.

The archive contains:

1) *Midas.zip*, a Rodin 0.8.2 model consisting of a generic Instruction Set Architecture (ISA) model, refined to the two MIDAS ISA variants.
2) *b2c.zip*, an Eclipse Java SDK project containing the Event-B to C translation tool "B2C".
3) *Gcc.rar*, a gcc tool-chain (i.e. compiler, assembler and linker) for compiling a supplied C test-suite to target the VM. It has been zipped using WinRAR.
4) *Execution.rar*, Visual Studio 5.0 projects, source code and batch files to compile the VM, compiler and server source code to Windows. A C test-suite and assembler bootstraps for compilation to the VMs is included. It has been zipped using *WinRAR*.

4. Host Environment

The only currently supported host operating system is *Windows XP*, although porting to other platforms is practical for all the components.

5. External Tools Installation

The flow of information through the various required tools is summarized in Figure 1.

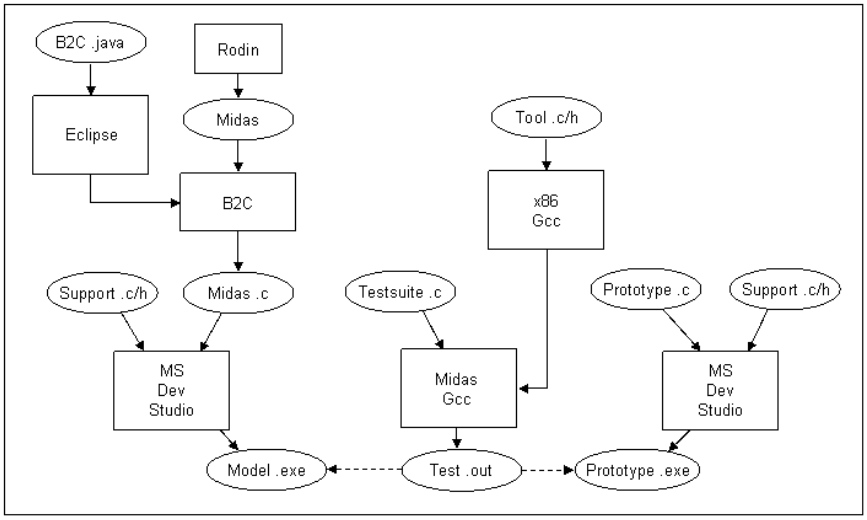

Figure 1: Development Tool Flow

The *Eclipse SDK* is used to compile Java source code to the *B2C* plug-in for *Rodin*. *Rodin* is then used to browse and edit the Midas model and generate a C output file via *B2C*. This *B2C*-generated Midas C file is compiled along with various supporting C files by *Visual C++* to create a VM as a *Windows* console executable. Midas *GCC* source code is compiled using the host *GCC* compiler to build the Midas *GCC* target tool-chain, which is then used to compile the demonstration C file to a Midas binary

executable, allowing it to be executed on the *Event-B* derived Midas VM. Separately hand-coded prototype C files are compiled using *Visual C++* to create a second VM Windows console executable, which is used to execute the same Midas demonstration binary. This process is applied to both Midas variants.

5.1 Rodin

Rodin 0.8.2 is needed for browsing of the Midas model and its discharged proof obligations, and automatic generation of C VM source code. Note that the following procedures assume the use of *Windows XP* and *Windows Internet Explorer 7*.

5.1.1. Java Installation

Rodin 0.8.2 requires installation of the *Java Runtime Environment 5.0 Update 19*. This is achieved by the following procedure:

- Navigate to http://java.sun.com/javase/downloads/5/jre
- Select "Windows" from the "Platform" drop-down, and select "Continue".
- Select "jre-1_5_0_19-windows-i586-p.exe" and then "Run". This will cause the installation to be downloaded and executed.
- Select the "Typical setup" radio button in the "License Agreement" page of the wizard and select "Accept".
- Select "Finish" when installation is complete.
- Verify that a new directory *C:\Program Files\Java\jre1.5.0_19* has been created.

5.1.2. Font Installation

The mathematical font used by Rodin is installed by the following procedure:

- Navigate to http://sourceforge.net/project/showfiles.php?group_id=108850.
- Locate the "Font: Brave Sans Mono" line and select "Download" on that line.
- Select "BraveSansMono-Roman-0.12.ttf".
- Select "Save" in the download wizard.
- Select C:\ for the save location and select "Save".
- Close the download wizard.
- Open the Windows Control Panel (in Start).
- Select the Fonts folder.
- Choose File > Install New Font.
- Select C:\ as the folder in the installation wizard.
- Select "Brave Sans Mono (TrueType)".
- Make sure the "Copy fonts to the Fonts" check box is selected.
- Select "OK".
- Verify that a new entry "Brave Sans Mono (TrueType)" has appeared in the Fonts folder.

5.1.3. Platform Installation

The *Rodin 0.8.2* main platform is now installed by the following procedure:

- Navigate to http://sourceforge.net/project/downloading.php?group_id=108850&filename=rodin-0.8.2-win32.zip&a=38548598.
- Select "Download RODIN" from the top of the page. You will be invited to download a file "rodin-0.8.2-win32.zip".
- Select "Save" in the download wizard.
- Select the Windows Desktop as the download destination.
- Select "Save".
- Close the download wizard.
- Open *rodin-0.8.2-win32.zip* by clicking on it on the Windows Desktop. It contains a single folder *rodin*.
- Drag and drop the *rodin* folder into the "C:\Program Files" folder.
- Rename the new folder "rodin" in "C:\Program Files" to "rodin 0.8.2".

5.1.4. Setup

In order to guarantee sufficient memory allocation for building of MIDAS under Rodin, a startup script is installed by the following procedure:

- Navigate to folder *C:\Program Files\rodin 0.8.2*.
- Right-click on the directory and create a new file *StartRodin.bat*.
- Right-click on *StartRodin.bat* and select "Edit".
- Insert the following line into the file: "C:\Program Files\rodin 0.8.2\rodin.exe" –vm java.exe –vmargs –Xms1000m –Xmx1000m
- Right-click on the file and select "Create Shortcut".
- Drag the new shortcut to the Windows Desktop.
- Create a new folder *C:\SJW\PhD\Development\RodinWorkspace*.
- Select the shortcut from the Desktop to start Rodin.
- Browse to *C:\SJW\PhD\Development\RodinWorkspace* in the "Workspace Launcher" wizard.
- Make sure the "Use this as the default and do not ask again" check box is selected.
- Select "OK" to start *Rodin*. The *Rodin* "Welcome" tab will be displayed.
- Close the "Welcome" tab by selecting the close cross.

5.1.5. Prover Installation

Additional proving plug-ins are required if Proof Obligation discharge is required. These are installed by the following procedure:

- In the *Rodin* environment, select "Help > Software Updates > Find and Install". The "Feature Updates" wizard will appear.
- Select the "Search for new features to install" radio button and select "Next".
- Make sure the "B2Free Update Site" check box is selected and select "Finish". The "Search Results" wizard will appear.
- Expand the "B2Free update site" line and "Other" within that.
- Make sure the "B2Free Provers 0.8.3" check box is selected and select "Next".

- Select the “I accept terms in the license agreement” radio button and select “Next”. The “Installation” wizard will appear.
- Select “Finish”. The “Feature Verification” wizard will appear.
- Select “Install All”. An invitation to restart Rodin will appear.
- Select “Yes”.

5.2 Cygwin

The cygwin tools are required for compilation of the MIDAS target compilers. These are installed by the following procedure:

5.2.1. Installation

- Create a new directory *C:\Cygwin*.
- Navigate to http://cygwin.com.
- Select the “Install or update now” link and then “Save”.
- Browse to *C\:cygwin* as the destination directory and select “Save”.
- Navigate to *C\:cygwin* and select *setup.exe*. The “Cygwin Setup” wizard will appear.
- Select “Next”.
- Select the “Install from internet” radio button and select “Next”.
- Select *C\:cygwin* as the “Root directory”, the “All Users” radio button for “Install For”, and “DOS/text” for “Default text file type” ,and select “Next”.
- Select *C\:cygwin* as the “Local package directory” and select “Next”.
- Select “Direct connection” for “Select your internet connection”.
- Select a mirror site from the presented list and select “Next”. A list of packages will appear.
- Click on the “Default” next to the “Devel” package to select “Install”. Other dependant lines will also be automatically selected.
- Select “Next”. Downloading and installation will begin.
- Select “Finish” when download is complete.

5.2.2. Setup

- Open the Windows Control Panel (in Start).
- Select the System folder. The “System Properties” dialog will appear.
- Select the “Advanced” tab.
- Select “Environment Variables”. The “Environment Variables” dialog will appear.
- In the “System variables” window locate the “Path” variable and select it. The “Edit System Variable” dialog will appear.
- Add *C:\cygwin\bin* to the “Variable Value” line, separated by a semi-colon and select “OK”.
- Select “OK” in both the “Environment Variables” and “System Properties” dialogs.
- Navigate to *C:\cygwin*.
- Locate file *Cygwin.bat*

- Right-click on the file and select “Create Shortcut”.
- Drag the new shortcut to the Windows Desktop.

5.3 Microsoft Visual C++ 5.0

Microsoft Visual C++ 5.0 is required for compilation of the VMs and support server. Installation is performed by inserting of a *Visual C++ 5.0* installation CD and following the automatically executed installation procedure.

5.4 Eclipse SDK

The *Eclipse Software Development Kit* is required for compilation of the *B2C* plug-in for Rodin. This is installed by the following procedure:

5.4.1. Installation

- Navigate to http://www.eclipse.org/downloads.
- Locate the line “Eclipse IDE for Java Developers” and select the “Windows” link from this.
- Select a mirror site from those presented. The “File Download” wizard for a *.zip* file will appear.
- Select “Save” and select the Windows Desktop as the destination directory.
- Select “Save”. The download will commence.
- Open the *.zip* file by clicking on it. It contains a single folder *eclipse*.
- Drag and drop the eclipse folder into the *C:\Program Files* folder.
- Right-click on the file and select “Create Shortcut”.
- Drag the new shortcut to the Windows Desktop.

5.4.2. Setup

- Create a new folder *C:\SJW\PhD\Development\EclipseWorkspace*.
- Select the shortcut from the Desktop to start Eclipse.
- Browse to *C:\SJW\PhD\Development\EclipseWorkspace* in the “Workspace Launcher” wizard.
- Make sure the “Use this as the default and do not ask again” check box is selected.
- Select “OK” to start Eclipse. The Eclipse “Welcome” tab will be displayed.
- Close the “Welcome” tab by selecting the close cross.

6. MIDAS Installation

Source code for all MIDAS components is supplied in the *MidasBundle.rar* archive described in 3. This file should be downloaded and saved to the Windows Desktop.

6.1 MIDAS Event-B Model

Importing of the Midas model into Rodin is required for inspection and modification. This is achieved by the following procedure:

- Extract the *Midas.zip* file from *MidasBundle.rar* to the Windows desktop.

- Run *Rodin 0.8.2* using the procedure described in 5.1.4.
- Select File > Import in the Rodin environment. The "Import" wizard will appear.
- Expand the "General" line and select the "Existing projects into workspace" line.
- Select "Next".
- Select the "Select Archive" radio button and browse to *Midas.zip* on the Windows Desktop.
- Select "Open" and then "Finish". The import will commence. The model will be automatically rebuilt: this will take several minutes.

6.2 Execution Environment

The Execution bundle is required for execution of the MIDAS VMs. Importing this is achieved by the following procedure:

- Extract the *Execution.rar* file from *MidasBundle.rar* to the Windows desktop.
- Open *Execution.rar* by clicking on it. It contains a single folder *Execution*.
- Drag and drop the *Execution* folder into the "C:\SJW\PhD\Development" folder.
- Open the Visual C++ environment.
- Select "File > Open Workspace". The "Open Workspace" wizard will appear.
- Browse to *C:\SJW\PhD\Development\Execution\WindowsProj* and select *Execution.dsw*. The component projects of the workspace will be loaded.

6.3 Target GCC

The *Gcc* bundle is required for building the MIDAS VM executables or modifying the tool-chain. Importing this is achieved by the following procedure:

- Extract the *Gcc.rar* file from *MidasBundle.rar* to the Windows desktop.
- Open *Gcc.rar* by clicking on it. It contains a single folder *Gcc*.
- Drag and drop the *Gcc* folder into the "C:\SJW\PhD\Development" folder.

6.4 B2C

Importing of the *B2C* plug-in project into the Eclipse SDK is required for insertion into *Rodin*, inspection and modification.

6.4.1. Rodin Source Code

The B2C source code has dependencies within the Rodin source, so this must be installed. This is achieved by the following procedure:

- Navigate to http://sourceforge.net/project/showfiles.php?group_id=108850&package_id=181714.

- Select the “0.8.2” link and select “rodin-0.8.2-sources.zip” within that. The “File Download” wizard will appear.
- Select the Windows Desktop as the destination directory and select “Save”. The download will commence.
- Run *Eclipse* using the procedure described in 5.4.2.
- Select File > Import in the Eclipse environment. The “Import” wizard will appear.
- Expand the “General” line and select the “Existing projects into workspace” line.
- Select “Next”.
- Select the “Select Archive” radio button and browse to *rodin-0.8.2-sources.zip* on the Windows Desktop.
- Select “Open” and then “Finish”. The import will commence. The source code will be automatically rebuilt, yielding several warnings.

6.4.2. B2C Source Code

The B2C source code has may now be installed. This is achieved by the following procedure:

- Extract the *b2c.zip* file from *MidasBundle.rar* to the Windows desktop.
- Select File > Import in the Eclipse environment. The “Import” wizard will appear.
- Expand the “General” line and select the “Existing projects into workspace” line.
- Select “Next”.
- Select the “Select Archive” radio button and browse to b2c.zip on the Windows Desktop.
- Select “Open” and then “Finish”. The import will commence. The source code will be automatically rebuilt: this will take several minutes.
- Select “Open” and then “Finish”. The import will commence. The model will be automatically rebuilt.

7. Demonstration Procedure

7.1 Demo Executables Build

The MIDAS demo program source code is given in *Testsuite.c* in *C:\SJW\PhD\Development\Execution\TargetTestsuite*. The file may also be accessed via the *Testsuite files* project in the *Visual C++* workspace imported in 6.2.

7.1.1. Stack Variant

- Open a DOS command console. This may be achieved by running “cmd” from the “Run…” option in Start.
- Change directory to *C:\SJW\PhD\Development\Execution\Scripts* (with the command “cd C:\SJW\PhD\Development\Execution\Scripts”).

- Run the DOS batch file *BuildTestsuiteMidasStk.bat* (with the command "BuildTestsuiteMidasStk").
- Display directory *C:\SJW\PhD\Development\Execution\TargetTestsuite* (with the command "dir C:\SJW\PhD\Development\Execution\TargetTestsuite").
- Confirm that the timestamps for files *bootStk.o*, *Testsuite.s*, *Testsuite.o*, *Testsuite.out* and *Testsuite.txt* have been updated.

7.1.2. Register Variant

- Open a DOS console and change directory to *C:\SJW\PhD\Development\Execution\Scripts*.
- Run the DOS batch file *BuildTestsuiteMidasReg.bat*.
- Display directory *C:\SJW\PhD\Development\Execution\TargetTestsuite*.
- Confirm that timestamps for files *bootReg.o*, *Testsuite.s*, *Testsuite.o*, *Testsuite.out* and *Testsuite.txt* have been updated.

7.2 Server Startup

The server provides executable loading and text output services for all executing VMs, and must therefore be running before target execution is started. The procedure for starting it is as follows:

- Open a DOS console and change directory to *C:\SJW\PhD\Development\Execution\Server\bin*.
- Run the server executable with the command "Server". The message "Server created OK." will appear.

7.3 Demo Program Execution (Prototype)

7.3.1. Stack Variant

- Make sure the test suite executable has been compiled for the stack variant using the procedure in 7.1.1.
- Make sure the server is running in a separate DOS console using the procedure in 7.2
- Open a DOS command console.
- Change directory to *C:\SJW\PhD\Development\Execution\CVirtualMachine\CMidasStkBin*.
- Run the prototype stack VM with the command "CMidasStk".
- Confirm that VM instrumentation is displayed in the VM DOS console, and test suite output text is displayed in the Server DOS console.

7.3.2. Register Variant

- Make sure the test suite executable has been compiled for the register variant using the procedure in 7.1.2.
- Make sure the server is running in a separate DOS console using the procedure in 7.2

- Change directory in the VM DOS console used in 7.3.1 to *C:\SJW\PhD\Development\Execution\CVirtualMachine\CMidasRegBin*.
- Run the prototype register VM with the command "CMidasReg".
- Confirm that VM instrumentation is displayed in the VM DOS console, and test suite output text is displayed in the Server DOS console.
- Compile the test suite back to the stack variant executable using the procedure in 7.1.1.
- Run the register VM again with the command "CMidasReg".
- Confirm that VM instrumentation is displayed in the VM DOS console, reporting that execution has ended with "Unknown opcode".

7.4 Demo Program Execution (Event-B)

7.4.1. Stack Variant

- Make sure the test suite executable has been compiled for the stack variant using the procedure in 7.1.1 and the server is running using the procedure in 7.2
- Change directory in the VM DOS console to *C:\SJW\PhD\Development\Execution\BVirtualMachine\BMidasStkBin*.
- Run the VM with the command "BMidasStk".
- Confirm that VM instrumentation is displayed in the VM DOS console, and test suite output text is displayed in the Server DOS console.

7.4.2. Register Variant

- Make sure the test suite executable has been compiled for the register variant using the procedure in 7.1.2 and the server is running using the procedure in 7.2
- Change directory in the VM DOS console to *C:\SJW\PhD\Development\Execution\BVirtualMachine\BMidasRegBin*.
- Run the VM with the command "BMidasReg".
- Confirm that VM instrumentation is displayed in the VM DOS console, and test suite output text is displayed in the Server DOS console.

7.5 Event Test Execution

7.5.1. Stack Variant

- Make sure the server is running using the procedure in 7.2
- Change directory in the VM DOS console to *C:\SJW\PhD\Development\Execution\BVirtualMachineTester\BMidasStk\bin*
- Run the prototype stack VM with the command "BMidasStkTester".
- Confirm that VM tests run and instrumentation is displayed confirming that all 108 events have been triggered.

7.5.2. Register Variant

- Make sure the server is running using the procedure in 7.2
- Change directory in the VM DOS console to *C:\SJW\PhD\Development\Execution\BVirtualMachineTester\BMidasReg\bin*.
- Run the prototype stack VM with the command "BMidasRegTester".
- Confirm that VM tests run and instrumentation is displayed confirming that all 112 events have been triggered.

8. Build Procedure

8.1 Building Support Server

All source code for support server may be accessed and built via the *Server files* project in the *Visual C++* workspace described in 6.2. Rebuilding of the server executable used in 7.2 is achieved using the following procedure:

- If the server is running in a DOS console after the procedure in 7.2, make sure that it has been stopped by selecting that console and entering "Ctrl-C".
- Open *Visual C++*.
- If the "Workspace" explorer is not already open in Visual C++, open it using "View > Workspace".
- Locate the "Server files" project in the Workspace explorer.
- Right-click on the project and select "Clean (selection only)".
- Right-click on the project and select "Build (selection only)".
- Navigate to *C:\SJW\PhD\Development\Execution\Server\bin* and confirm that the timestamp for file Server.exe has been updated.

8.2 Building Prototype VMs

All source code for the prototype VMs may be accessed and built via the "CMidasStk files" and "CMidasReg files" projects in the *Visual C++* workspace. Rebuilding of the VM executables used in 7.3.1and 7.3.2 is achieved using the following procedure:

8.2.1. Stack Variant

- Open *Visual C++*.
- Locate the *CMidasStk files* project in the Workspace explorer.
- Right-click on the project and select "Clean (selection only)".
- Right-click on the project and select "Build (selection only)".
- Navigate to *C:\SJW\PhD\Development\Execution\CVirtualMachine\CMidasStkBin* and confirm that the timestamp for *CMidasStk.exe* has been updated.

8.2.2. Register Variant

- Open *Visual C++*.
- Locate the *CMidasReg files* project in the Workspace explorer.
- Right-click on the project and select "Clean (selection only)".
- Right-click on the project and select "Build (selection only)".
- Navigate to *C:\SJW\PhD\Development\Execution\CVirtualMachine\CMidasRegBin* and confirm that the timestamp for *CMidasReg.exe* has been updated.

8.3 Installing B2C

- Open the Eclipse environment described in 5.4 and import the *B2C* source code using the procedure in 6.4.
- In the Eclipse environment select "File > Export".
- Expand the cross of "Plug-in Development" and select "Deployable plug-ins and fragments".
- Select "Next". The "Export" dialog will appear.
- Select only the "b2c (0.1.0)" check box and select *C:\Program Files\rodin 0.8.2* as the destination directory.
- Select "Finish". Build and installation of the plug-in will commence.
- Open *Rodin* using the procedure in 5.1. If it was currently open, close it and re-open it.
- Confirm that a *B2C* icon and drop-down have been installed in the Rodin toolbar.

8.4 Generating Event-B VM Source Code

Two C source files are automatically generated from the Midas model via *B2C*, for compilation into the Event-B VMs, achieved using the following procedure:

- Open *Rodin* using the procedure in 5.1.
- Ensure that the Midas model has been imported using the procedure from 6.1.
- Ensure that *B2C* has been installed using the procedure in 8.3.
- Select the *B2C* icon in the *Rodin* environment. Instrumentation will be displayed in the *DOS* console started by the batch file used to start *Rodin*.
- Navigate to *C:\SJW\PhD\Development\RodinWorkspace* and confirm that two new files *MidasStkMchB2C.c* and *MidasRegMchB2C.c* have been generated.

8.5 Building Event-B VMs

All source code for the prototype VMs may be accessed and built via the *BMidasStk files* and *BMidasReg files* projects in the *Visual C++* workspace. Rebuilding of the VM executables used 7.4.1 in 7.4.2 and is achieved using the following procedure:

8.5.1. Stack Variant

- Open *Visual C++*.

- Ensure that the file *MidasStkMchB2C.c* has been generated using the procedure in 8.4.
- Locate the *CMidasStk files* project in the Workspace explorer.
- Right-click on the project and select “Clean (selection only)”.
- Right-click on the project and select “Build (selection only)”.
- Navigate to *C:\SJW\PhD\Development\Execution\BVirtualMachine\BMidasStkBin* and confirm that the timestamp for *BMidasStk.exe* has been updated.

8.5.2. Register Variant

- Open *Visual C++*.
- Ensure that the file *MidasRegMchB2C.c* has been generated using the procedure in 8.4.
- Locate the *BMidasReg files* project in the Workspace explorer.
- Right-click on the project and select “Clean (selection only)”.
- Right-click on the project and select “Build (selection only)”.
- Navigate to *C:\SJW\PhD\Development\Execution\BVirtualMachine\BMidasRegBin* and confirm that the timestamp for *BMidasReg.exe* has been updated.

8.6 Building Event Test Executer

All source code for host executables used to load and run the VMs under test in 7.5.1 and 7.5.2 may be accessed and built via the *CMidasStkTester files* and *CMidasRegTest files* projects in the *Visual C++* workspace. Rebuilding of them is achieved using the following procedure:

8.6.1. Stack Variant

- Locate the *CMidasStkTester files* project in the Workspace explorer.
- Right-click on the project and select “Clean (selection only)”.
- Right-click on the project and select “Build (selection only)”.
- Navigate to *C:\SJW\PhD\Development\Execution\BVirtualMachineTester\BMidasStk\bin* and confirm that the timestamp for *BMidasStkTester.exe* has been updated.

8.6.2. Register Variant

- Locate the *CMidasRegTester files* project in the Workspace explorer.
- Right-click on the project and select “Clean (selection only)”.
- Right-click on the project and select “Build (selection only)”.
- Navigate to *C:\SJW\PhD\Development\Execution\BVirtualMachineTester\BMidasReg\bin* and confirm that the timestamp for *BMidasRegTester.exe* has been updated.

8.7 Building Event Test Executables

The source code for the target executables loaded and run in 7.5.1 and 7.5.2 is given in *C:\SJW\PhD\Development\Execution\BVirtualMachineTester\BMidasStk\target* and *C:\SJW\PhD\Development\Execution\BVirtualMachineTester\BMidasReg\target*. Rebuilding of them is achieved using the following procedure:

8.7.1. Stack Variant

- Open a *DOS* console and change directory to *C:\SJW\PhD\Development\Execution\BVirtualMachineTester\BMidasStk*.
- Run the *DOS* batch file *BuildMidasStkTargetExecs.bat*.
- Display directory *C:\SJW\PhD\Development\Execution\BVirtualMachineTester\BMidasStk\target*.
- Confirm that the timestamps for all .txt files in the directory have been updated.

8.7.2. Register Variant

- Open a *DOS* console and change directory to *C:\SJW\PhD\Development\Execution\BVirtualMachineTester\BMidasReg*.
- Run the *DOS* batch file *BuildMidasRegTargetExecs.bat*.
- Display directory *C:\SJW\PhD\Development\Execution\BVirtualMachineTester\BMidasReg\target*.
- Confirm that the timestamps for all .txt files in the directory have been updated.

8.8 Building MIDAS GCC

Although the *GCC binutils* and *gcc* source code consists of many files, those MIDAS-specific parts usually needed for modification of the tools are accessible via the *StkBinutils files*, *StkGcc files*, *RegBinutils files* and *RegGcc files* projects in the *Visual C++* workspace. Rebuilding of them is then performed using the following procedure:

8.8.1. Stack Variant

- Open a *cygwin* console by selecting the icon created in 5.4.2.
- Change directory within the cygwin console to */cygdrive/c/SJW/PhD/Development/Gcc/midas* (using the command “cd /cygdrive/c/SJW/PhD/Development/Gcc/midas/”).
- Clean the build environment, by entering the command “make clean” at the cygwin command line.
- Using Windows Explorer, navigate to *C:\SJW\PhD\Development*.
- Right-click on directory *Gcc* and select “Properties”.
- Uncheck the “Read-only” check-box in “Attributes” and select “OK”.

- Select the “Apply changes to this folder, sub-folders and files” radio button and select “OK”.
- Configure the build environment, by entering the command “sh ../binutils-2.16/configure --target=midas-elf --prefix=/cygdrive/c/SJW/PhD/Development/Gcc/midas” at the cygwin command line.
- Build the binutils by entering the command “make” at the cygwin command line. This will take several minutes.
- Install the binutils binaries by entering the command “make install” at the cygwin command line.
- Display directory *C:\SJW\PhD\Development\Gcc\midas\bin*.
- Confirm that the timestamps for the following files in the directory have been updated: midas-elf-addr2line.exe, midas-elf-ar.exe, midas-elf-as.exe, midas-elf-c++filt.exe, midas-elf-ld.exe, midas-elf-nm.exe, midas-elf-objcopy.exe, midas-elf-objdump.exe, midas-elf-ranlib.exe, midas-elf-readelf.exe, midas-elf-size.exe, midas-elf-strings.exe and midas-elf-strip.exe.
- Clean the build environment again, by entering the command “make clean” at the cygwin command line.
- Again navigate to *C:\SJW\PhD\Development,* right-click on directory *Gcc,* select “Properties” and remove all read-only properties.
- Configure the build environment, by entering the command “sh ../gcc-4.0.2/configure –build=i686-pc-cygwin –host=i686-pc-cygwin --target=midas-elf --prefix=/cygdrive/c/SJW/PhD/Development/Gcc/midas –enable-languages=c” at the cygwin command line.
- Build the gcc compiler by entering the command “make” at the cygwin command line. This will take several minutes.
- Under some cygwin installations, the make may fail with the error message “./configargs.h:2: error: mssing terminating “ character”. This is due to a bug in the cygwin tools. If this occurs, remove the carriage-return from the end of line 2 of file =/cygdrive/c/SJW/PhD/Development/Gcc/midas/gcc/configargs.h with an editor such as Wordpad, and run *make* again. The build will end with message “make[2]: *** [libgcc/./_muldi3.o] Error 1”.
- Display directory *C:\SJW\PhD\Development\Gcc\midas\gcc*.
- Confirm that the timestamps for the following files in the directory have been updated: cc1.exe and xgcc.exe.
- Install the driver executable, by entering the command “cp /cygdrive/c/SJW/PhD/Development/Gcc/midas/gcc/cc1.exe /cygdrive/c/SJW/PhD/Development/Gcc/midas/bin/cc1.exe”.
- Install the compiler executable, by entering the command “cp /cygdrive/c/SJW/PhD/Development/Gcc/midas/gcc/xgcc.exe /cygdrive/c/SJW/PhD/Development/Gcc/midas/bin/midas-elf-gcc.exe”.

8.8.2. Register Variant

- Open a *cygwin* console by selecting the icon created in 5.4.2.

- Change directory within the cygwin console to */cygdrive/c/SJW/PhD/Development/Gcc/midasreg* (using the command "cd /cygdrive/c/SJW/PhD/Development/Gcc/midasreg/").
- Clean the build environment, by entering the command "make clean" at the cygwin command line.
- Using Windows Explorer, navigate to *C:\SJW\PhD\Development*.
- Right-click on directory *Gcc* and select "Properties".
- Uncheck the "Read-only" check-box in "Attributes" and select "OK".
- Select the "Apply changes to this folder, sub-folders and files" radio button and select "OK".
- Configure the build environment, by entering the command "sh ../binutils-2.16/configure --target=midasreg-elf --prefix=/cygdrive/c/SJW/PhD/Development/Gcc/midasreg" at the cygwin command line.
- Build the binutils by entering the command "make" at the cygwin command line. This will take several minutes.
- Install the binutils binaries by entering the command "make install" at the cygwin command line.
- Display directory *C:\SJW\PhD\Development\Gcc\midasreg\bin*.
- Confirm that the timestamps for the following files in the directory have been updated: midasreg-elf-addr2line.exe, midasreg-elf-ar.exe, midasreg-elf-as.exe, midasreg-elf-c++filt.exe, midasreg-elf-ld.exe, midasreg-elf-nm.exe, midasreg-elf-objcopy.exe, midasreg-elf-objdump.exe, midasreg-elf-ranlib.exe, midasreg-elf-readelf.exe, midasreg-elf-size.exe, midasreg-elf-strings.exe and midasreg-elf-strip.exe.
- Clean the build environment again, by entering the command "make clean" at the cygwin command line.
- Again navigate to *C:\SJW\PhD\Development,* right-click on directory *Gcc,* select "Properties" and remove all read-only properties.
- Configure the build environment, by entering the command "sh ../gcc-4.0.2/configure –build=i686-pc-cygwin –host=i686-pc-cygwin --target=midasreg-elf --prefix=/cygdrive/c/SJW/PhD/Development/Gcc/midasreg –enable-languages=c" at the cygwin command line.
- Build the gcc compiler by entering the command "make" at the cygwin command line. This will take several minutes.
- Under some cygwin installations, the make may fail with the error message "./configargs.h:2: error: mssing terminating " character". This is due to a bug in the cygwin tools. If this occurs, remove the carriage-return from the end of line 2 of file =/cygdrive/c/SJW/PhD/Development/Gcc/midasreg/gcc/configargs.h with an editor such as Wordpad, and run *make* again. The build will end with message "make[2]: *** [libgcc/./_muldi3.o] Error 1".
- Display directory *C:\SJW\PhD\Development\Gcc\midasreg\gcc*.
- Confirm that the timestamps for the following files in the directory have been updated: cc1.exe and xgcc.exe.

- Install the driver executable, by entering the command “cp /cygdrive/c/SJW/PhD/Development/Gcc/midasreg/gcc/cc1.exe /cygdrive/c/SJW/PhD/Development/Gcc/midasreg/bin/cc1.exe”.
- Install the compiler executable, by entering the command “cp /cygdrive/c/SJW/PhD/Development/Gcc/midasreg/gcc/xgcc.exe /cygdrive/c/SJW/PhD/Development/Gcc/midasreg/bin/midasreg-elf-gcc.exe”.