1

Calling Conventions

1.1 Introduction

Each high-level language compiler generates subroutine entry and exit code based upon a set of rules known as the compiler's **calling convention**. These rules address the following issues:

- *register preservation and usage*
- *parameter passing*
- *call and return mechanism*

A compiler's calling convention is of importance when interfacing to subroutines written in another language either assembly or high-level. Even when the high-level language and target processor are the same, different compilers may use different calling conventions. As a result, calling conventions are both processor and compiler dependent.

1.2 Processor Background

The MC68020 architecture supports a simple yet effective call and return mechanism. A subroutine is invoked via the branch to subroutine (**bsr**) or the jump to subroutine (**jsr**) instructions. These instructions push the return address on the current stack. The return from subroutine (**rts**) instruction pops the return address off the current stack and transfers control to that instruction. It is is important to note that the MC68020 call and return mechanism does not

automatically save or restore any registers. It is the responsibility of the high-level language compiler to define the register preservation and usage convention.

1.3 Calling Mechanism

For specific information regarding the parameter passing convention for Tartan MC68020 Ada, refer to the **Calling Conventions** section of the **Implementation Details** in the **Compilation System** manual.

1.4 Register Usage

For specific information regarding the parameter passing convention for Tartan MC68020 Ada, refer to the **Calling Conventions** section of the **Implementation Details** in the **Compilation System** manual.

1.5 Parameter Passing

For specific information regarding the parameter passing convention for Tartan MC68020 Ada, refer to the **Parameter Passing Conventions** section of the **Implementation Details** in the **Compilation System** manual.

1.6 User-Provided Routines

All user-provided routines invoked by **RTEMS**, such as user extensions, device drivers, and MPCI routines, must also adhere to these calling conventions.