6

Processor Dependent Information Table

6.1 Introduction

Any highly processor dependent information required to describe a processor to **RTEMS** is provided in the **CPU Dependent Information Table**. This table is not required for all processors supported by **RTEMS**. This chapter describes the contents, if any, for a particular processor type.

6.2 CPU Dependent Information Table

The MC68020 version of the **RTEMS CPU Dependent Information Table** contains a single pointer to the MC68020's interrupt vector table. This address is provided to allow **RTEMS** to manipulate the MC68020's **Interrupt Vector Table**. This manipulation allows the automatic vectoring to the **RTEMS** Interrupt Manager when an interrupt occurs. The **Ada** record definition is given here:

```
type CPU_TABLE is
   record
      PRETASKING_HOOK                  : RTEMS.ADDRESS;
      PREDRIVER_HOOK                   : RTEMS.ADDRESS;
      POSTTASKING_HOOK                 : RTEMS.ADDRESS;
      IDLE_TASK                        : RTEMS.ADDRESS;
      DO_ZERO_OF_WORKSPACE             : RTEMS.BOOLEAN;
      INTERRUPT_STACK_SIZE             : RTEMS.UNSIGNED32;
      EXTRA_SYSTEM_INITIALIZATION_STACK : RTEMS.UNSIGNED32;
      INTERRUPT_VECTOR_TABLE           : RTEMS.ADDRESS;
   end record;
```

pretasking_hook is the address of the user provided routine which is invoked once **RTEMS** initialization is complete but before interrupts and tasking are enabled. This field may be **NULL** to indicate that the hook is not utilized.

predriver_hook is the address of the user provided routine which is invoked with tasking enabled immediately before the MPCI and device drivers are initialized. **RTEMS** initialization is complete, interrupts and tasking are enabled, but no device drivers are intialized. This field may be **NULL** to indicate that the hook is not utilized.

postdriver_hook is the address of the user provided routine which is invoked with tasking enabled immediately after the MPCI and device drivers are initialized. **RTEMS** initialization is complete, interrupts and tasking are enabled, and the device drivers are intialized. This field may be **NULL** to indicate that the hook is not utilized.

idle_task is the address of the optional user provided routine which is used as the system's IDLE task. If this field is not NULL, then the **RTEMS** default IDLE task is not used. This field may be **NULL** to indicate that the default IDLE is to be used.

do_zero_of_workspace indicates whether **RTEMS** should zero the Workspace as part of its initialization. If set to **TRUE**, the Workspace is zeroed. Otherwise, it is not.

interrupt_stack_size is the size of the **RTEMS** allocated interrupt stack in bytes. This value must be at least as large as **MINIMUM_STACK_SIZE**.

extra_system_ initialization_stack is the extra stack space allocated for the **RTEMS** system initialization task in bytes. The system initialization task invokes the MPCI and device driver initialization routines and may require extra stack space on some targets.

interrupt_vector_table is the base address of the CPU's Exception Vector Table.