14

Dual-Ported Memory Manager

14.1 Introduction

The **dual-ported memory manager** provides a mechanism for converting addresses between internal and external representations for multiple dual-ported memory areas (DPMA). The directives provided by the dual-ported memory manager are:

Name	Directive Description
port_create	Create a port
port_ident	Get ID of a port
port_delete	Delete a port
port_external_to_internal	Convert external to internal address
port_internal_to_external	Convert internal to external address

14.2 Background

A dual-ported memory area (DPMA) is an contiguous block of RAM owned by a particular processor but which can be accessed by other processors in the system. The owner accesses the memory using internal addresses, while other processors must use external addresses. **RTEMS** defines a **port** as a particular mapping of internal and external addresses.

There are two system configurations in which dual-ported memory is commonly found. The first is tightly-coupled multiprocessor computer systems where the

dual-ported memory is shared between all nodes and is used for inter-node communication. The second configuration is computer systems with intelligent peripheral controllers. These controllers typically utilize the DPMA for high-performance data transfers.

14.3 Operations

14.3.1 Creating a Port

The **port_create** directive creates a port into a DPMA with the user-defined name. The user specifies the association between internal and external representations for the port being created. **RTEMS** allocates a **Dual-Ported Memory Control Block** (DPCB) from the DPCB free list to maintain the newly created DPMA. **RTEMS** also generates a unique dual-ported memory port ID which is returned to the calling task. **RTEMS** does not initialize the dual-ported memory area or access any memory within it.

14.3.2 Obtaining Port IDs

When a port is created, **RTEMS** generates a unique port ID and assigns it to the created port until it is deleted. The port ID may be obtained by either of two methods. First, as the result of an invocation of the **port_create** directive, the task ID is stored in a user provided location. Second, the port ID may be obtained later using the **port_ident** directive. The port ID is used by other dual-ported memory manager directives to access this port.

14.3.3 Converting an Address

The **port_external_to_internal** directive is used to convert an address from external to internal representation for the specified port. The **port_internal_to_external** directive is used to convert an address from internal to external representation for the specified port. If an attempt is made to convert an address which lies outside the specified DPMA, then the address to be converted will be returned.

14.3.4 Deleting a DPMA Port

A port can be removed from the system and returned to **RTEMS** with the **port_delete** directive. When a port is deleted, its control block is returned to the DPCB free list.

14.4 Directives

This section details the dual-ported memory manager's directives. A subsection is dedicated to each of this manager's directives and describes the calling sequence, related constants, usage, and status codes.

14.4.1 PORT_CREATE - Create a port

CALLING SEQUENCE:

```
rtems_status_code rtems_port_create(
  rtems_name        name,
  void             *internal_start,
  void             *external_start,
  rtems_unsigned32  length,
  rtems_id         *id
);
```

DIRECTIVE STATUS CODES:

SUCCESSFUL	port created successfully
INVALID_NAME	invalid task name
INVALID_ADDRESS	address not on four byte boundary
TOO_MANY	too many DP memory areas created

DESCRIPTION:

This directive creates a port which resides on the local node for the specified DPMA. The assigned port id is returned in **id**. This port id is used as an argument to other dual-ported memory manager directives to convert addresses within this DPMA.

For control and maintenance of the port, **RTEMS** allocates and initializes an DPCB from the DPCB free pool. Thus memory from the dual-ported memory area is not used to store the DPCB.

NOTES:

The **internal_address** and **external_address** parameters must be on a four byte boundary.

This directive will not cause the calling task to be preempted.

14.4.2 PORT_IDENT - Get ID of a port

CALLING SEQUENCE:

```
rtems_status_code rtems_port_ident(
  rtems_name  name,
  rtems_id   *id
);
```

DIRECTIVE STATUS CODES:

SUCCESSFUL	port identified successfully
INVALID_NAME	port name not found

DESCRIPTION:

This directive obtains the port **id** associated with the specified **name** to be acquired. If the port name is not unique, then the port id will match one of the DPMAs with that name. However, this port id is not guaranteed to correspond to the desired DPMA. The port id is used to access this DPMA in other dual-ported memory area related directives.

NOTES:

This directive will not cause the running task to be preempted.

14.4.3 PORT_DELETE - Delete a port

CALLING SEQUENCE:

```
rtems_status_code rtems_port_delete(
  rtems_id id
);
```

DIRECTIVE STATUS CODES:

SUCCESSFUL	port deleted successfully
INVALID_ID	invalid port id

DESCRIPTION:

This directive deletes the dual-ported memory area specified by **id**. The DPCB for the deleted dual-ported memory area is reclaimed by **RTEMS**.

NOTES:

This directive will not cause the calling task to be preempted.

The calling task does not have to be the task that created the port. Any local task that knows the port id can delete the port.

14.4.4 PORT_EXTERNAL_TO_INTERNAL - Convert external to internal address

CALLING SEQUENCE:

```
rtems_status_code rtems_port_external_to_internal(
  rtems_id   id,
  void      *external,
  void     **internal
);
```

DIRECTIVE STATUS CODES:

SUCCESSFUL always successful

DESCRIPTION:

This directive converts a dual-ported memory address from external to internal representation for the specified **port**. If the given external address is invalid for the specified port, then the internal address is set to the given external address.

NOTES:

This directive is callable from an ISR.

This directive will not cause the calling task to be preempted.

14.4.5 PORT_INTERNAL_TO_EXTERNAL - Convert internal to external address

CALLING SEQUENCE:

```
rtems_status_code rtems_port_internal_to_external(
  rtems_id   id,
  void      *internal,
  void     **external
);
```

DIRECTIVE STATUS CODES:

SUCCESSFUL always successful

DESCRIPTION:

This directive converts a dual-ported memory address from internal to external representation so that it can be passed to owner of the DPMA represented by the specified **port**. If the given internal address is an invalid dual-ported address, then the external address is set to the given internal address.

NOTES:

This directive is callable from an ISR.

This directive will not cause the calling task to be preempted.