www.qorvo.com
lpw.support@qorvo.com

gpPoolMem Reference Manual

API Description

Version 2.10.2.0
November 15, 2021

Contents

Chapter 1

Introduction

This document describes in a formal manner the API interface that can be used to control all the functionality of the PoolMem component.

Chapter 2

Module Documentation

2.1 General PoolMem Definitions

The general PoolMem Definitions.
Maximal 3 different sizes of chunks available

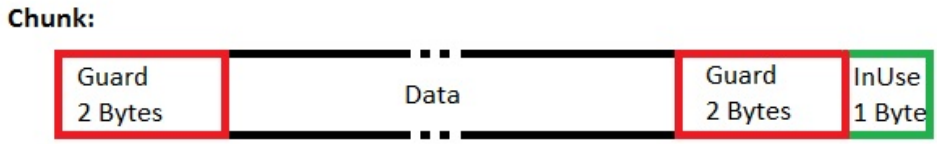

Figure 2.1: Single Chunk structure

2.2 General PoolMem Public Functions

The general public PoolMem functionality is implemented in these functions.

Functions

- void gpPoolMem_Init (void)

  This function initialize the chunks, it basically sets all the guards and the InUse flags on false.
- void gpPoolMem_Reset (void)

  This function is almost similar as PoolMem_Init; invoked at warm restart.
- void * gpPoolMem_Malloc (UInt8 comp_id, UInt32 nbytes, Bool try_)

  This function allocates a free chunk.
- void gpPoolMem_Free (void *pData)

  This function will free an allocated chunk.
- void gpPoolMem_Dump (Bool checkConsistency)

  This function will dump all PoolMem content and check consistency.

2.2.1 Detailed Description

2.2.2 Function Documentation

gpPoolMem_Dump()

```
void gpPoolMem_Dump (
              Bool checkConsistency )
```

Parameters

checkConsistency	Assert when consistency check fails

Returns

Assert When consistency is compromised

gpPoolMem_Free()

```
void gpPoolMem_Free (
              void * pData )
```

Parameters

pData	Pointer to Chunk

Returns

Assert When wrong pointer given
Assert When GuardEnd or GuardStart overwritten

gpPoolMem_Malloc()

```
void* gpPoolMem_Malloc (
              UInt8 comp_id,
              UInt32 nbytes,
              Bool try_ )
```

Parameters

nbytes	Minimal Chunk size
try	avoid assert and returning a NULL is handled

Returns

void$*$ Returns pointer to allocated chunk
Assert When no chunk free (depends on try)
Assert When nbytes $>$ size of biggest chunk
Assert When GuardEnd or GuardStart overwritten of free chunk