Installation ManualInstallation Manual

UNI92K-4DMUNI92K-4DMW

|  |
| --- |
| ***UNI92K - UFS*** |
| ***Software Requirements Specification*** |
| SRS Rev. 0.01 |



Copyright ©UNITEST Corporation

**Table of Contents**

[1. Introduction 3](#_Toc29298191)

[1.1. Purpose 3](#_Toc29298192)

[1.2. Scope 3](#_Toc29298193)

[1.3. Definitions, Acronyms, and Abbreviations 3](#_Toc29298194)

[1.4. Reference 3](#_Toc29298195)

[1.5. Overview 3](#_Toc29298196)

[2. Overall Description 4](#_Toc29298197)

[2.1. Product Perspective 4](#_Toc29298198)

[2.1.1. System Interfaces 4](#_Toc29298199)

[2.1.2. User Interfaces 5](#_Toc29298200)

[2.1.3. Software Interfaces 5](#_Toc29298201)

[2.1.4. Memory Constraints 5](#_Toc29298202)

[2.1.5. Operations 5](#_Toc29298203)

[2.2. User Characteristics 5](#_Toc29298204)

[2.3. Constraints 5](#_Toc29298205)

[3. Specific Requirements 6](#_Toc29298206)

[3.1. Functional Requirements 6](#_Toc29298207)

[3.1.1. Power Up/Down 6](#_Toc29298208)

[3.1.2. DRAM 7](#_Toc29298209)

[3.1.3. Item 구성 7](#_Toc29298210)

[3.1.4. Multi Queue 8](#_Toc29298211)

[3.1.5. FAIL MEMORY 8](#_Toc29298212)

[3.2. Performance Requirements 8](#_Toc29298213)

[3.2.1. Write / Read Performance 8](#_Toc29298214)

[4. Software Reuse 9](#_Toc29298215)

[5. Supporting Information 10](#_Toc29298216)

[5.1. Software Specific Requirement 10](#_Toc29298217)

[5.2. Revision History 10](#_Toc29298218)

# Introduction

본 문서는 UNI92K-UFS의 기능들을 설명하는 SRS(Software Requirements Specification)이다.

## Purpose

UNI92K-UFS의 요구사항을 분석하고 정의하여 개발 방향 선정, Architecture 설계, Function Block 문서의 기준이 되기 위해 작성되었다.

문서의 사용자는 UNITEST UNI92K-UFS 개발자(SW, FW, HW, TE)를 대상으로 작성한다.

SW 개발팀은 본 SRS에 따라 기능 설계 및 구현을 한다.

## Scope

UNI92K-UFS을 구성 할 GUI, Test Core, PGB(RG, ALPG, FM, CTRL, CPLD), Utility(Pcal, Diag, Tcal), PLC, Compiler 등의 동작을 규정 한다.

* Power Control 기능
* ALPG Interface 기능
* Chamber Interface, Emulator 기능
* S/W RA를 위한 FM Interface 기능
* 사용자가 운영하는 Server와 통신 기능

## Definitions, Acronyms, and Abbreviations

GUI : Graphical User Interface

PGB : Pattern Generator Board

PG : Pattern Generator

CTRL : Controller

TB : Test Board

## Reference

* UFS Spec - JESD220D.pdf
* UFS Spec - JESD223D.pdf

## Overview

본 문서는 다음과 같이 구성되었다.

2장에서는 UNI92K-UFS의 총괄 기술, 제품의 조망, 제품의 기능, 사용자의 특성, 제약사항, 가정과 의존사항, 요구사항의 유보를 살펴 본다.

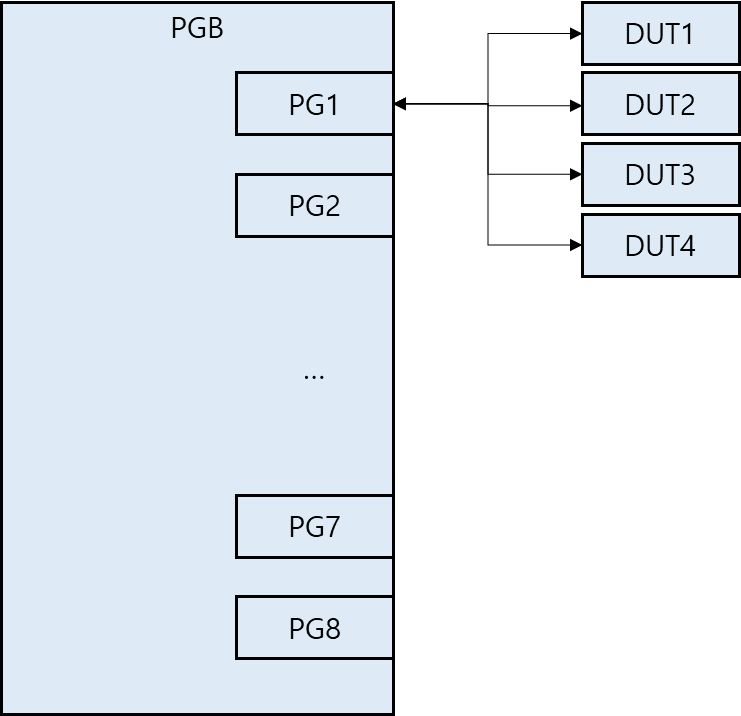
3장에서는 세부 요구사항에 대해 알아 본다. 각 절의 내용으로는 외부 인터페이스, 기능, 성능 요구사항, 논리적 데이터베이스 요구사항, 설계상의 제약사항, 소프트웨어 시스템 속성, 추가 설명을 살펴 본다.

# Overall Description

## Product Perspective

### System Interfaces

* UNI92K-UFS는 PG로부터 Device를 Control 한다.
* Device를 Control하는 User Program은 PG에서 구동한다.
* PGB를 Control 할 수 있는 CHIP으로는 PG, FPGA, CTRL 가 있다.



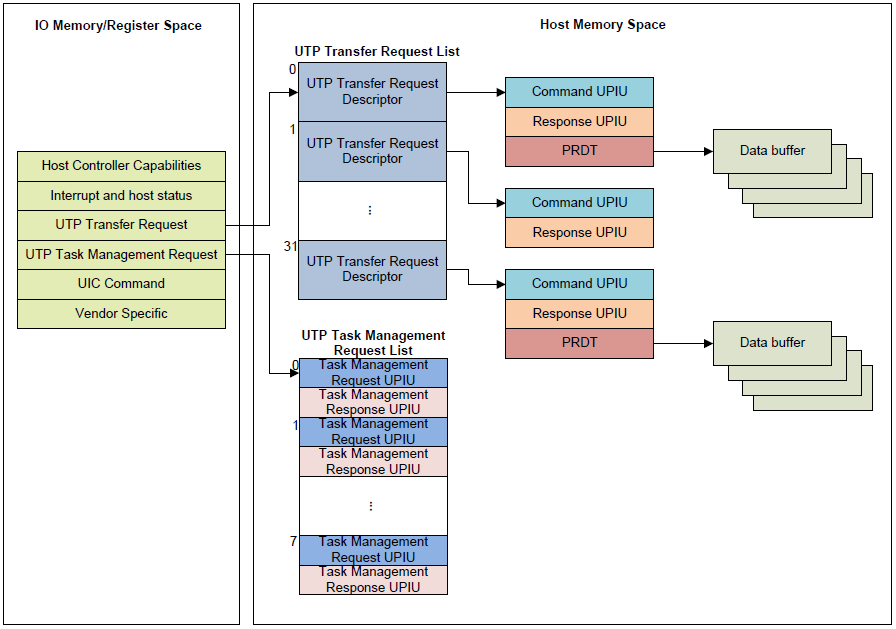


Figure 1 UNI92K-UFS 간략 구조

### User Interfaces

* 사용자에게 UNI92K-UFS Library를 제공하여 UFS Device Test, Test Log Display 등을 제공한다.

### Software Interfaces

* 개발 환경
  + Linux : 4.14.0-xilinx-v2018.2
  + Editor
    - Gedit : 3.22.0
    - VIM : 7.4.160-2
    - Eclipse : Oxygen.3 Release (4.7.3)

### Memory Constraints

* DRAM MEMORY
  + 용량 : PG당 2GB
  + DRAM은 UFS Test에 필요한 Memory를 어느정도 사용한다.

### Operations

* 사용자는 Test Program에 UFS Library API를 이용하여 작성하거나, User가 원하는 Program을 함께 생성 할 수 있다.
* 사용자가 만든 Test Program을 UNISDK에서 Load 하여 Run 동작시킨다.
* Test Program에 대한 결과를 Log로 남기며 결과값에 대한 내용은 GUI창을 통해 사용자에게 표시하며, Log File로 저장되어 따로 확인 할 수 있다.

## User Characteristics

사용자는 UNI92K와 UFS Spec의 전반적인 정보를 알고 있는 사람이어야 한다.

Test Program 작성자는 C 언어를 다룰 수 있어야 한다. Test Program의 Compiler는 gcc이며, UNI92K-UFS Library를 이용하여 Program을 구현해야 한다.

UNI92K Manual을 숙지하고 있는 사용자로 가정한다.

## Constraints

* Test Program을 작성할 때 UNI92K-UFS에서 제공하는 API 규격에 맞도록 작성한다.

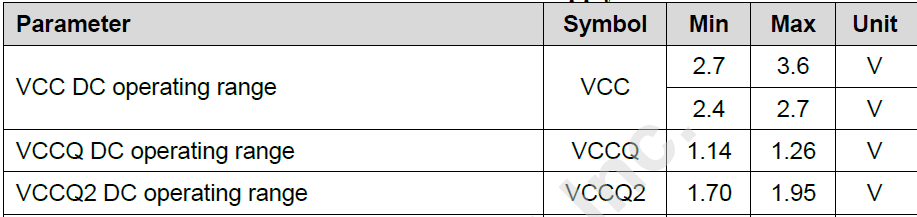
# Specific Requirements

## Functional Requirements

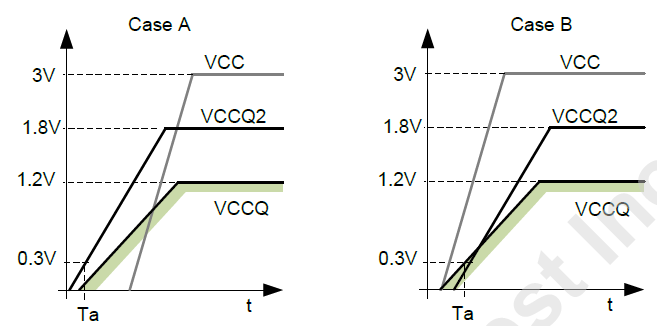
Power Up/Down, UFS Test에 관한 내용을 기입한다.

### Power Up/Down

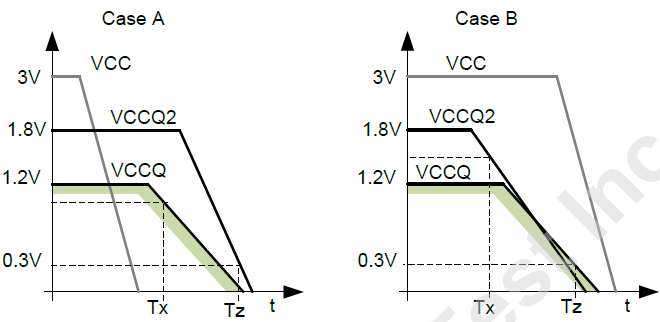
* Power Up/Down 은 UFS Spec에 맞게 인가한다.
* Power 정보



* Power Up Ramp



* Power Down Ramp



### DRAM

* DRAM은 PG당 2GB이며, DUT당 512MB 씩 할당한다.
* DUT에 할당된 Memory 중 일부는 UFS Test에 필요한 List를 작성하는데 사용한다.
* Data Engine 적용 전 / 후 DRAM Assign은 다음과 같다.
* Data Engine 적용 전

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **DRAM(2GB)** | | | |
| **Offset** | **DUT1 ( 512MB )** | **DUT2 ( 512MB )** | **DUT3 ( 512MB )** | **DUT4 ( 512MB )** |
| **0x0000\_0000** | Task management List ( 20DW X 8 ) 1KB | Task management List ( 20DW X 8 ) 1KB | Task management List ( 20DW X 8 ) 1KB | Task management List ( 20DW X 8 ) 1KB |
|  |
| **0x0000\_03FF** |
| **0x0000\_0400** | Request List ( 8DW X 32 ) 1KB | Request List ( 8DW X 32 ) 1KB | Request List ( 8DW X 32 ) 1KB | Request List ( 8DW X 32 ) 1KB |
|  |
| **0x0000\_07FF** |
| **0x0000\_0800** | UPIU ( 128KB x 32 ) 4MB | UPIU ( 128KB x 32 ) 4MB | UPIU ( 128KB x 32 ) 4MB | UPIU ( 128KB x 32 ) 4MB |
|  |
| **0x0040\_07FF** |
| **0x0040\_0800** | PRDT ( 4MB x 32 ) 128MB | PRDT ( 4MB x 32 ) 128MB | PRDT ( 4MB x 32 ) 128MB | PRDT ( 4MB x 32 ) 128MB |
|  |
| **0x0840\_07FF** |
| **0x0840\_0800** | Reserved | Reserved | Reserved | Reserved |
|  |
| **0x1FFF\_FFFF** |

* Data Engine 적용 후

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **DRAM(2GB)** | | | |
| **Offset** | **DUT1 ( 512MB )** | **DUT2 ( 512MB )** | **DUT3 ( 512MB )** | **DUT4 ( 512MB )** |
| **0x0000\_0000** | Task management List ( 20DW X 8 ) 1KB | Task management List ( 20DW X 8 ) 1KB | Task management List ( 20DW X 8 ) 1KB | Task management List ( 20DW X 8 ) 1KB |
|  |
| **0x0000\_03FF** |
| **0x0000\_0400** | PRDT ( 4MB x 32 ) 128MB | PRDT ( 4MB x 32 ) 128MB | PRDT ( 4MB x 32 ) 128MB | PRDT ( 4MB x 32 ) 128MB |
|  |
| **0x0800\_03FF** |
| **0x0800\_0400** | Reserved | Reserved | Reserved | Reserved |
|  |
| **0x1FFF\_FFFF** |

### Item 구성

* 기본적인 Item 구성은 다음과 같이 한다.

|  |  |
| --- | --- |
| **Item** | **설명** |
| item\_poweron | Device에 Power Up |
| item\_poweroff | Device에 Power Down |
| item\_linkstartup | Host initialize, Device reset, Device configuration |
| item\_deviceinfo | Device 정보 Read ( Max LBA 정보, Block Size 정보 등 ) |
| item\_setpowermode | Change Link Speed |
| item\_write | Data Write Item  - Data Engine 적용 전은 SW에서 처리  - Data Engine 적용 후는 FPGA에서 처리 |
| item\_read | Data Read Item  - Data Engine 적용 전은 SW에서 처리  - Data Engine 적용 후는 FPGA에서 처리 |
| 기타 item | UFS Spec 에 근거한 Item을 구성한다. |

### Multi Queue

* UFS는 32Queue까지 지원하기 때문에 Multi Queue를 지원한다.

### FAIL MEMORY

* Compare Enable 된 Read Item을 수행할 때 Compare Fail이 발생한 경우 Fail Memory에 Fail 정보가 적재된다.
* Fail Memory는 Data Engine이 적용된 이후 시점에 적용된다.

## Performance Requirements

* System Test Time Performance 향상을 위해서 System 상태 제어 조회의 최적화를 목표로 한다.

### Write / Read Performance

* Write / Read Performance 향상을 위해 Multi Queue 가 가능해야하고, Write / Read는 Data Engine에서 처리할 수 있도록 한다.
* Performance 향상을 위해 System Clock을 맞춰야한다.

# Software Reuse

* UNI91K-UFS를 BASE로 만든다.
* 재사용 범위는 다음과 같다.
* UFS Library
  + UNI91K의 기본 UFS Library를 사용하며, UNI92K System에 맞게 구성한다.
* UFS Spec Version
  + UFS 2.1과 UFS 3.0의 구분은 Build Compile 옵션으로 구분되게 한다.
* UART 기능
  + Device의 UART 지원여부에 따라 기능을 적용한다.
  + UART 지원할 경우 UNI92K System에 맞춰 구성한다.
* Device Power Up / Down
  + Device Power 제어는 UNI92K System에 맞춰 구성한다.

# Supporting Information

## Software Specific Requirement

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| 날짜 | 버전 | 설명 | 작성자 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| **UniTest Inc.**  27, Gigok-ro, Giheung-gu, Yongin-si, Gyeonggi-do, Korea  Zip Code: 446-930  Tel : 82-31-547-0300  Fax : 82-31-547-0468 |