Installation ManualInstallation Manual

UNI92K-4DMUNI92K-4DMW

|  |
| --- |
| ***UNI 53K*** |
| ***Software Requirements Specification*** |
| Rev. 0.01 |



Copyright ©UNITEST Corporation

**Table of Contents**

[1. Introduction 4](#_Toc45017283)

[1.1. Purpose 4](#_Toc45017284)

[1.2. Scope 4](#_Toc45017285)

[1.3. Definitions, Acronyms, and Abbreviations 4](#_Toc45017286)

[1.4. Reference 5](#_Toc45017287)

[1.5. Overview 5](#_Toc45017288)

[2. Overall Description 6](#_Toc45017289)

[2.1. Product Perspective 6](#_Toc45017290)

[2.1.1. System Interfaces 6](#_Toc45017291)

[2.1.2. User Interfaces 6](#_Toc45017292)

[2.1.3. Hardware Interfaces 7](#_Toc45017293)

[2.1.4. Software Interfaces 8](#_Toc45017294)

[2.1.5. Communications Interfaces 9](#_Toc45017295)

[2.1.6. Memory Constraints 9](#_Toc45017296)

[2.1.7. Operations 9](#_Toc45017297)

[2.2. User Characteristics 10](#_Toc45017298)

[2.3. Constraints 10](#_Toc45017299)

[3. Software Requirements Specification 11](#_Toc45017300)

[3.1. External Interface Requirements 11](#_Toc45017301)

[3.1.1. User Interfaces 11](#_Toc45017302)

[3.1.2. Hardware Interfaces 11](#_Toc45017303)

[3.2. Functional Requirements 11](#_Toc45017304)

[3.2.1. Device Entry Setup 11](#_Toc45017305)

[3.2.2. DUT Map 11](#_Toc45017306)

[3.2.3. Handler Sequence Setup 12](#_Toc45017307)

[3.2.4. Test Log 저장 방식 변경 12](#_Toc45017308)

[3.2.5. DIAG 12](#_Toc45017309)

[3.2.6. PCAL 13](#_Toc45017310)

[3.2.7. PCAL Initialization 13](#_Toc45017311)

[3.2.8. PCAL Load Result File 13](#_Toc45017312)

[3.2.9. PCAL Save Result File 13](#_Toc45017313)

[3.2.10. PCAL Apply 13](#_Toc45017314)

[3.2.11. Main Controller, Sub Controller, CPLD 변경 사항 수정 13](#_Toc45017315)

[3.2.12. Flash Download 14](#_Toc45017316)

[3.2.13. Boad ID 15](#_Toc45017317)

[3.2.14. Fixed Power Control(On BCB) 15](#_Toc45017318)

[3.2.15. ADC Control 15](#_Toc45017319)

[3.2.16. ADC Calibration 15](#_Toc45017320)

[3.2.17. ADC Calibration Check 16](#_Toc45017321)

[3.2.18. PPS Control 16](#_Toc45017322)

[3.2.19. PPS Calibration 17](#_Toc45017323)

[3.2.20. PPS Current Measure Calibration 17](#_Toc45017324)

[3.2.21. PPS Calibration Check 18](#_Toc45017325)

[3.2.22. PMU Control 19](#_Toc45017326)

[3.2.23. PMU Calibration 19](#_Toc45017327)

[3.2.24. PMU Current Measure Calibration 20](#_Toc45017328)

[3.2.25. PMU Calibration Check 21](#_Toc45017329)

[3.2.26. DC-TEST 21](#_Toc45017330)

[3.2.27. OCP 21](#_Toc45017331)

[4. Software Reuse 22](#_Toc45017332)

[5. Supporting Information 24](#_Toc45017333)

[5.1. Software Specific Requirement 24](#_Toc45017334)

[5.2. Revision History 24](#_Toc45017335)

# Introduction

본 문서는 UNIN53K Project의 S/W 요구사항을 설명하는 SRS(Software Requirements

Specification)이다.

## Purpose

UNI53K의 요구사항을 분석하고 정의하여 개발 방향 선정, Architecture 설계, Function Block 문서의 기준이 되기 위해 작성되었다. 문서의 사용자는 UNITEST UNI53K 개발자(SW, FW, HW, TE)를 대상으로 작성되며, FW, HW, TE의 요구 사항이 반영되었는지 확인 할 수 있는 기준이 된다.

또한 고객의 요구 사항이 설계에 반영 할 수 있도록 필요한 요구사항을 해당 문서에 기술 하도록 한다.

## Scope

UNI53K을 구성 할 GUI, FPGA, BCB(CTRL, CPLD), Utility(Pcal, Diag), GPIB 등의 동작을 규정 한다.

* Power Control 기능
* Power Calibration 기능
* System Diagnostic 기능
* FPGA Interface 기능
* Handler Interface 기능
* 사용자가 운영하는 Server와 통신 기능
* Control GUI 제공
* Test Log 저장 기능

## Definitions, Acronyms, and Abbreviations

GUI : Graphical User Interface

BCB : Base Control Board

DPB : Device Protocol Board

BPB : Back Plane Board

CTRL : Controller

CPLD : Complex Programmable Logic Device

GPIB : General Purpose Interface Bus

TB : Test Board

Pcal : Power Calibration

Diag : Diagnostic

FORCE V : 예상 Voltage, 기대 Voltage, 설정 Voltage

SENSE V : Read Voltage

## Reference

* UNI92K Operating\_User\_Guide KR R020 20191219.docx
* U53K\_HW\_Architecture\_20200609.pptx

## Overview

본 과제는 UNI92K를 base로 진행되는 과제이며, UNI92K와 구성이 유사하여, UNI92K에서 변경된 사항에 대하여 기술하는 방향으로 한다.

* 2장에서는 SYSTEM 전체의 사양과 SYSTEM 동작에 대하여 기술한다.
* 3장에서는 software 세부 요구사항과 UNI92K대비 변경된 사항에 대하여 기술한다.

# Overall Description

## Product Perspective

### System Interfaces

* UNI53K는 SYSTEM을 Control 할 수 있는 UNISDK로부터 BCB(Base Control Board)와 DPB(Device Protocol Board) 등과 Ethernet 통신을 이용하여 DATA를 주고 받는다.
* FPGA(on DPB)는 embedded system(linux)가 설치 되어 있어 HOST PC와 NFS를 이용하여 Ethernet 통신을 한다. 단 부하가 큰 LOG Data는 TCP Protocol을 이용하여 통신한다.
* BCB에는 Control 할 수 있는 CTRL, CPLD가 있으며, DPB에는 Control 할 수 있는 FPGA가 있다.
* GPIB통신을 이용하여 Handler를 제어한다.

SYSTEM

GPIB

(Handler)

Host PC

**FPGA(**embedded linux**)**

UNISDK

**Ctrl**

Server

그림 1, UNI53K 간략 Block Diagram

### User Interfaces

* 사용자에게 UNISDK(GUI)를 제공하여, 기능으로는 Test Program Load 및 Run Test 실행, Pcal, Diag, Firmware Download, Analyzer, Log Viewer 등과 같은 Utility 실행, GPIB 제어, 결과 Monitoring과 같은 기능을 제공한다.
* UNISDK(GUI)는 Engineer Mode를 제공 하며, 해당 Mode를 이용하여 System 운영에 권한을 선택 할 수 있다.
* System에 문제가 발생할 경우 Test Program을 강제 종료하는 Test Stop Mode을 제공하며 System 초기화를 위한 Initialize를 제공.
* UNI53K 동작과 관련된 LOG는 Dut Log tab을 이용하여 출력하고, Text Log를 제공한다.

### Hardware Interfaces

* FPGA(on DPB)
  + Device : XC5VLX110-3FF1153
  + Pattern Line : 2048 Line

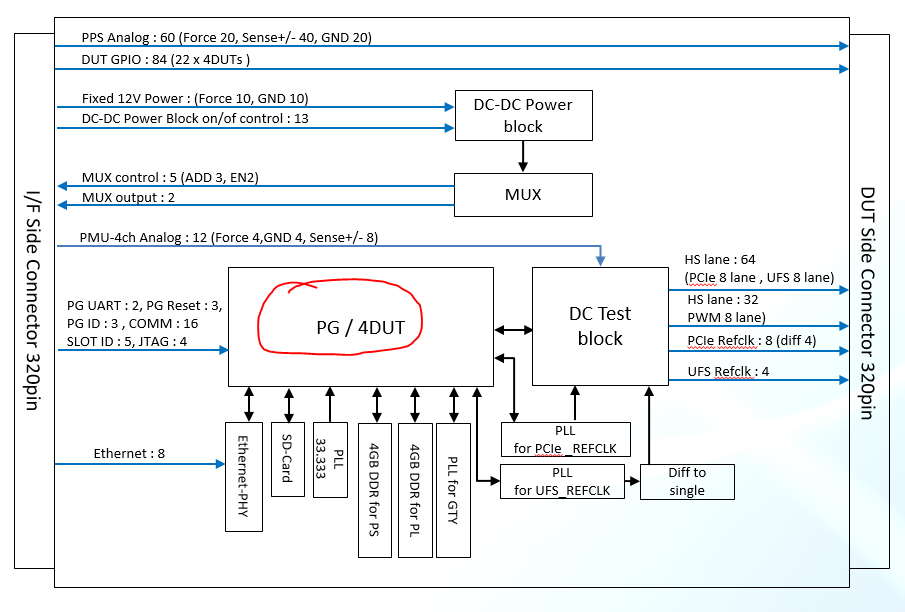


그림 2, DPB Block Diagram

* CTRL(on BCB)
  + Device : XC5VLX110T-FF1136

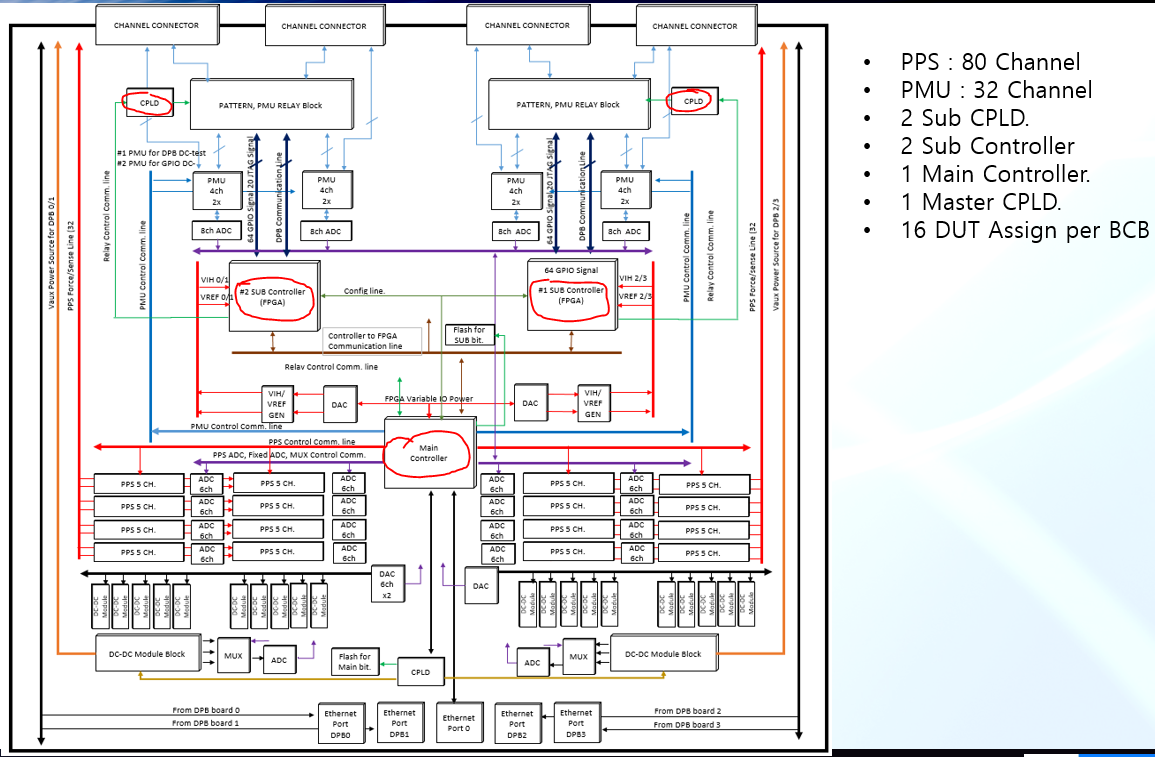


그림 3, BCB Block Diagram

* SUB CTRL(on BCB)
  + Device : XC5VLX110T-FF1136
* CPLD(on BCB)
  + Device : LCMX02-7000HC-4FG484I
* GPIB
* FLASH
* SERVER PLATFORM

### Software Interfaces

* 개발 환경
  + OS : Red Hat Enterprise Linux Server release 7.4
  + Kernel : 3.10.0-693.el7.x86\_64
  + GUI : Qt version 4.8.5
  + Editor
    - Gedit : 3.22.4
    - VIM : Vi IMproved 7.4

### Communications Interfaces

* PC -> PGB, PC -> FPGA
  + Ethernet : 10/100/1000 Mbps
* PC -> GPIB
  + NI PCIe-GPIB : 1.5 MB/s

### Memory Constraints

* HDD
  + 용량 : 1TB
  + 속도 : 7200RPM
  + CACHE : 64MB
* MEMORY
  + TYPE : DDR4
  + 용량 : 16GB
  + 동작클럭 : 2666MHz

### Operations

* 사용자는 OCP Test Program에서 Test Item을 작성해야 하며, UNISDK에서 제공하는 Recipe를 이용하여 User가 원하는 Test Recipe작성 할 수 있다.
* 사용자가 Recipe를 이용하여 만든 Test Recipe는 UNISDK에서 Load되고 Test Recipe는 각각의 OCP로 전송되며, 해당 Test Recipe의 Test Item이 각OCP에서 수행 된다.
* Test Recipe에 대한 결과를 Log로 남기며 결과값에 대한 내용은 GUI의 Dut Log tab을 통해 사용자에게 표시한다.
* 사용자는 Test Program을 통해서 Log를 만들 수 있으며, 해당 Log는 Terminal 또는 GUI의 Dut Log tab또는 Log File로 확인 할 수 있다.
* 사용자는 Compiler를 통하여 Socket, Pattern File 등을 생성할 수 있다.
* Test Program 실행 중 Error Level에 따라 그대로 진행하는 Ignore Mode와 Program을 강제로 종료하는 Test Stop Mode가 발생 할 수 있다.
* Handler는 Test Sequence에 따라 UNISDK에서 제어할 수 있다. *~~GUI의 Utility Handler CTRL을 이용하여 제어 할 수 있다.~~*
* 사용자는 GUI의 Utility 중 Error Status를 이용하여 Error List를 확인할 수 있으며, 각 Error의 Error level을 조절할 수 있다.
* 사용자는 GUI의 Utility 중 Power Calibration을 이용하여 ADC와 DAC의 Calibration 동작을 Start 할 수 있으며 Result 값을 GUI, Terminal, Text File로 확인 할 수 있다.
* 사용자는 GUI의 Utility 중 Power Calibration을 이용하여 ADC와 DAC의 Cal condition을 설정 할 수 있고, terminal 창에서 conf file을 직접 열어 설정 할 수 있다.
* 사용자는 GUI의 Utility 중 Diagnostic을 이용하여 DC, Channel, ADC, DAC, Clamp, Current Measure, BCB와 DPB ID, Chip Device Version, Chip Device Temperature를 Check 할 수 있다.
* 사용자는 GUI의 Utility 중 Diagnostic을 이용하여 Item Condition을 설정 할 수 있고, terminal 창에서 conf file을 직접 열어 설정 할 수 있다.

## User Characteristics

사용자는 UNI53K System의 전반적인 정보를 알고 있는 사람이어야 한다.

OCP Program 작성자는 C 언어를 다룰 수 있어야 한다. UNISDK의 Compiler는 gcc/g++, OCP Program의 Cross Compiler는 gcc/g++이며, UNISDK에서 제공하는 Library및 OCP Program에서 제공 하는 Test Item을 이용하여 Program을 구현해야 한다.

UNI53K Manual을 숙지하고 있는 사용자로 가정한다.

## Constraints

* Test Program을 작성할 때 UNISDK에서 제공하는 api 규격에 맞도록 작성한다.
* Test Program이 실행 중일 때는 Utility와 같은 기능들을 사용할 수 없다.
* Test Item을 작성할 때 OCP에서 제공하는 Test Item 규격에 맞도록 작성한다.
* Recipe GUI를 이용 하여 Test Recipe 작성시 Recipe작성 규칙에 따라 작성한다.
* 사용하고자 하는 각 Device별 Entry setup에 주의 한다.
* Handler 특징 이해하며, 동작 Sequence를 확인 한다.

# Software Requirements Specification

## External Interface Requirements

### User Interfaces

* UNISDK와 통신하는 SERVER
* UNISDK와 통신하는 Handler

### Hardware Interfaces

* Ethernet
  + Protocol : TBD ( TCP/IP, UDP, FTP )
* GPIB
  + Protocol : NI PCIe-GPIB : 1.5 MB/s

## Functional Requirements

Device Entry Setup, DUT Map, Handler Sequence Setup, Test Log 저장 방식 변경, Diag,

Main CTRL, Sub CTRL, CPLD(Pattern, PMU Relay), Fixed Power Control(on BCB), ADC Control,

ADC Calibration, PPS Control, PPS Calibration, PMU Control, PMU Calibration, Dc Test,

Board Power Control(on DPB), FPGA Configuration SEQ Control, FPGA Uart Concept, SERVER,

Compiler에 대한 내용을 기입한다.

### Device Entry Setup

* UNISDK실행시 Test를 진행할 Device Type를 사용자가 지정 할 수 있도록 GUI Entry Setup을 제공 하고 있다.
* SYSTEM Type과 TB Type(PCIE, ASI, UFS), Device Type 그리고 BCB, DPB, DUT수량을 결정 할 수 있다.

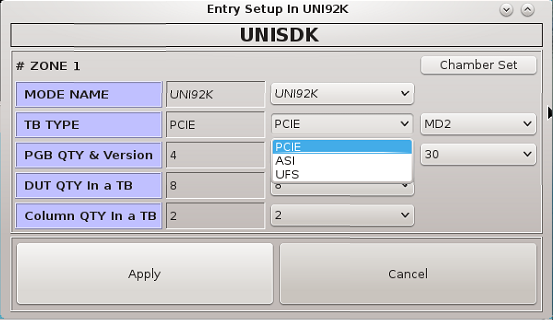


그림 4, UNISDK Entry Setup

### DUT Map

* 사용자가 Entry Setup에서 BCB, DPB, DUT, 수를 결정 하면 UNISDK에서는 해당 정보를 바탕으로 GUI에 DUT MAP을 표시 한다.
* GUI의 DUT MAP을 클릭 해서 Memo기능을 사용 하거나 Test Log를 확인 할 수 있다.
* 추후 UNI53K에 맞는 DUT MAP 적용 예정

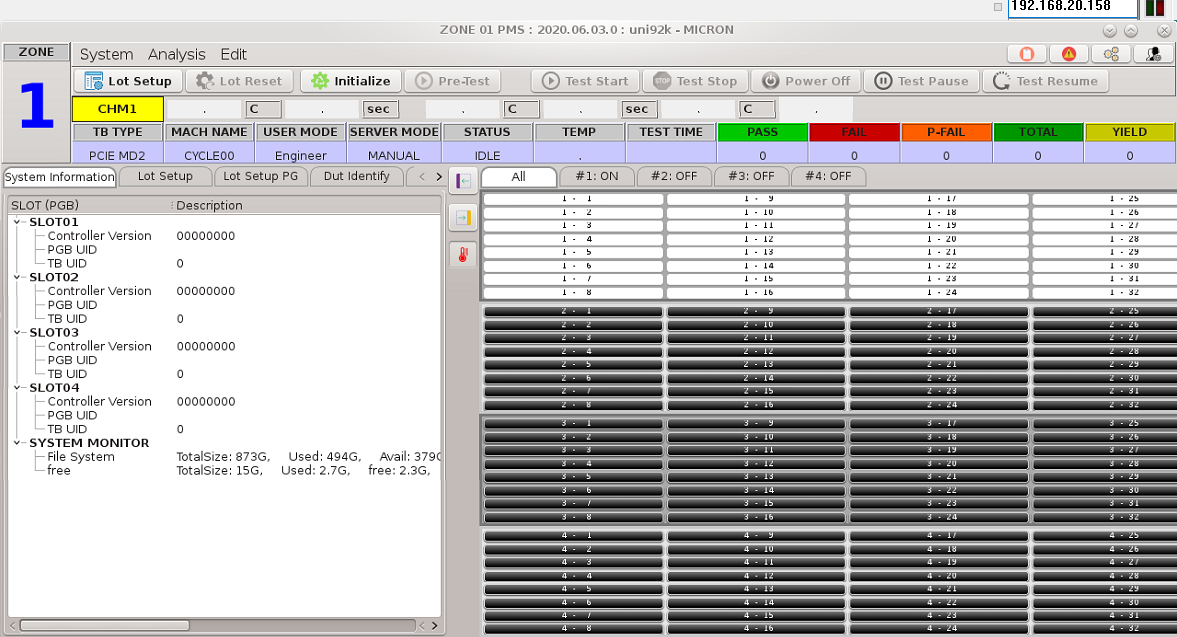


그림 5, UNI92K OS

### Handler Sequence Setup

* GPIB 통신 확인
* GPIB 통신 Scenario 확인
* Handler 업체와 협의 하여 Handler사용 Sequence 확립 필요

### Test Log 저장 방식 변경

* UNI92K과제 대비 FPGA가 많아짐에 따라 기존 방식(NFS를 이용한 Test Log file 저장)의 Test Log 저장 방식으로는 Test Time에 영향으로 줄 것으로 판단
* UNI53K에서는 Test Log 저장 방식을 TCP Protocol로 Packet을 전송 하여 OS에서 취합 하는 방식으로 변경 예정
* 또한 하기 그림과 같이 Network 구성이 변경되어야 지연 없는 TCP 통신이 가능함.

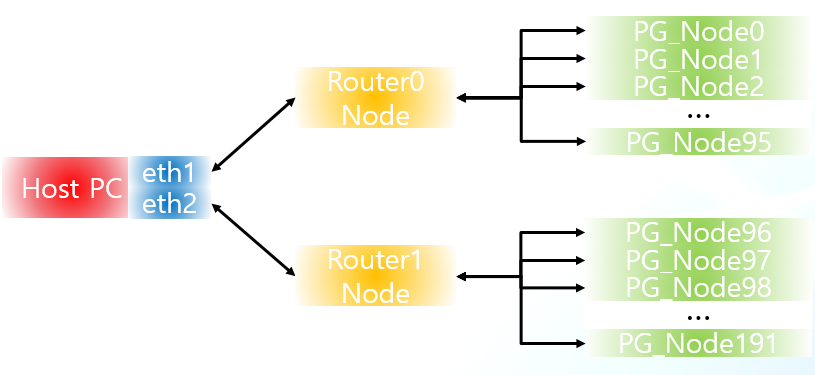


그림 6, UNI53K PG Network 구성

### DIAG

* DIAG 에서는 아래의 항목에 대한 이상 유무를 판단 한다.

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | Unit | Section | Item |
| 1 | BCB |  | Fixed Power Check |
| 2 |  | ADC(PPS, PMU) Check |
| 3 |  | DAC Check |
| 4 |  | PPS Check |
| 5 |  | PMU VS, VSIM Check |
| 6 |  | PMU IS, ISVM Check |
| 7 |  | RELAY(PPS, PMU Cal) Check |
| 8 | DPB |  | DPB Power Check |
| 9 |  | PL\_DRAM Check |
| 10 |  | FW Download Check |

표 1, Diag 항목

### PCAL

* PCAL 에서는 아래의 항목에 대하여 Calibration을 진행한다.

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | Unit | Section | Item |
| 1 | BCB | ADC | ADC Calibration |
| 2 | DAC | DAC Calibration |
| 3 | PMU | PMU Calibration |
|  |  |  |

표 2, PCAL 항목

### PCAL Initialization

* Pcal Result 값인 ADC, DAC의 Gain, Offset 값을 저장하는 모든 변수를 초기화 한다.
* ADC의 초기화 값 GAIN : 1.0, Offset : 0.0
* DAC의 초기화 값 GAIN : 1.0, Offset : 0.001

### PCAL Load Result File

* Pcal Result File을 load 하여 ADC, DAC의 Gain, Offset 값을 변수에 저장한다.

### PCAL Save Result File

* ADC Cal 결과와 DAC Cal 결과를 사용자가 저장한 경로와 File 이름으로 저장한다.

### PCAL Apply

* Result File을 Load 한다.

### Main Controller, Sub Controller, CPLD 변경 사항 수정

* 해당 과제의 Base과제인 UNI92K 대비 수정된 내용에 대하여 파악 및 수정 필요
* 아래의 항목에서 수정이 필요한 항목 정리 필요

|  |  |  |
| --- | --- | --- |
| 구분 | Unit | Item |
| 1 | Main  Ctrl | PPS Control |
| 2 | ADC for PPS |
| 3 | SPI Flash for Main controller |
| 4 | SPI Flash for SUB controller |
| 5 | BCB ID |
| 6 | ADC for Fixed power |
| 7 | DPB ID |
| 8 | CPLD | PPS CAL Relay Control |
| 9 | Fixed power on/off |
| 10 | #1,#2 Sub Controller | 2 PMU for GPIO DC-Test |
| 11 | 2 PMU ADC for GPIO DC-Test |
| 12 | GPIO for 8 DUT |
| 13 | Fixed power on/off on 2 DPB |
| 14 | PMU CAL |
| 15 | #1,#2 CPLD | GPIO Signal Relay |
| 16 | PMU Force/sense MUX |
|  |  |

표 3, Ctrl변경 확인 필요 항목

### Flash Download

* Main Ctrl, Sub Ctrl, CPLD에서 사용될 image를 SPI Download 할 수 있는 API를 제공한다.

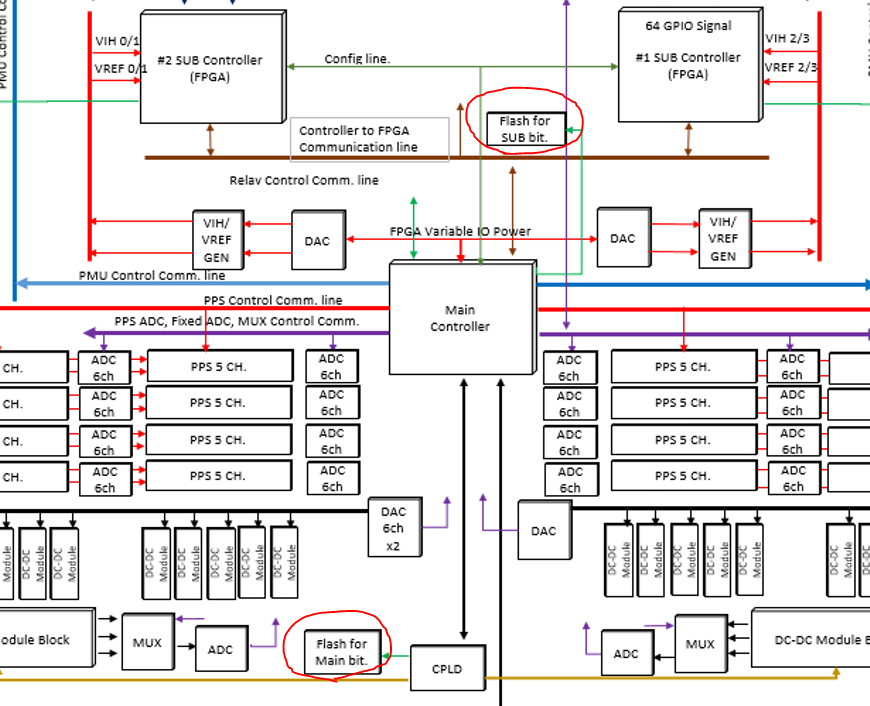


그림 7, Flash Block Diagram

### Boad ID

* BCB, DPB ID를 읽어 올 수 있는 API를 제공한다.

### Fixed Power Control(On BCB)

* BCB Booting시 CPLD에서 Fixed power를 인가 하게 되며, Test 종료시점이나 초기화 시점, OS에서 Fixed power Control이 필요한 경우를 대비해 관련 API or Testitem을 제공한다.

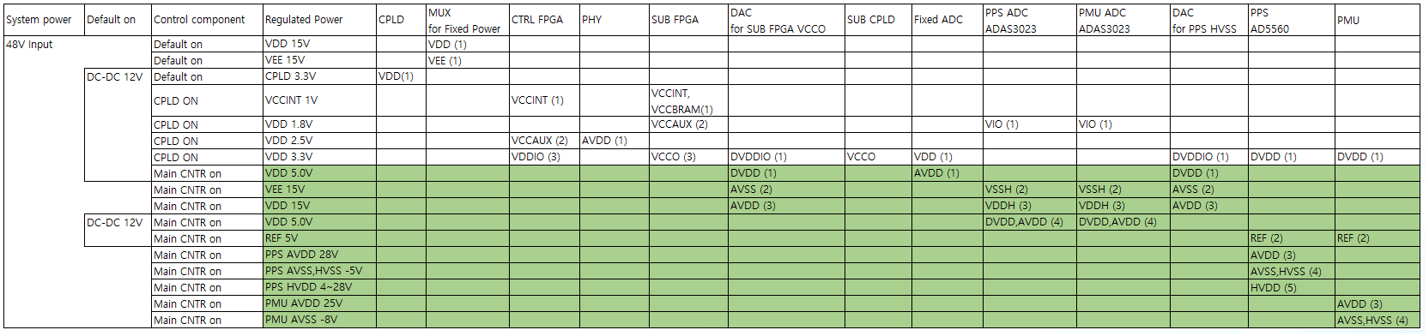


그림 8, UNI53K BCB Power 종류

### ADC Control

* Fixed Power, PPs, PMU에서 인가되는 전압을 읽기 위해 각각의 ADC Channel이 존재한다.
* 각각의 ADC Channel에 접근해 ADC Value를 읽어올 API를 제공 한다.

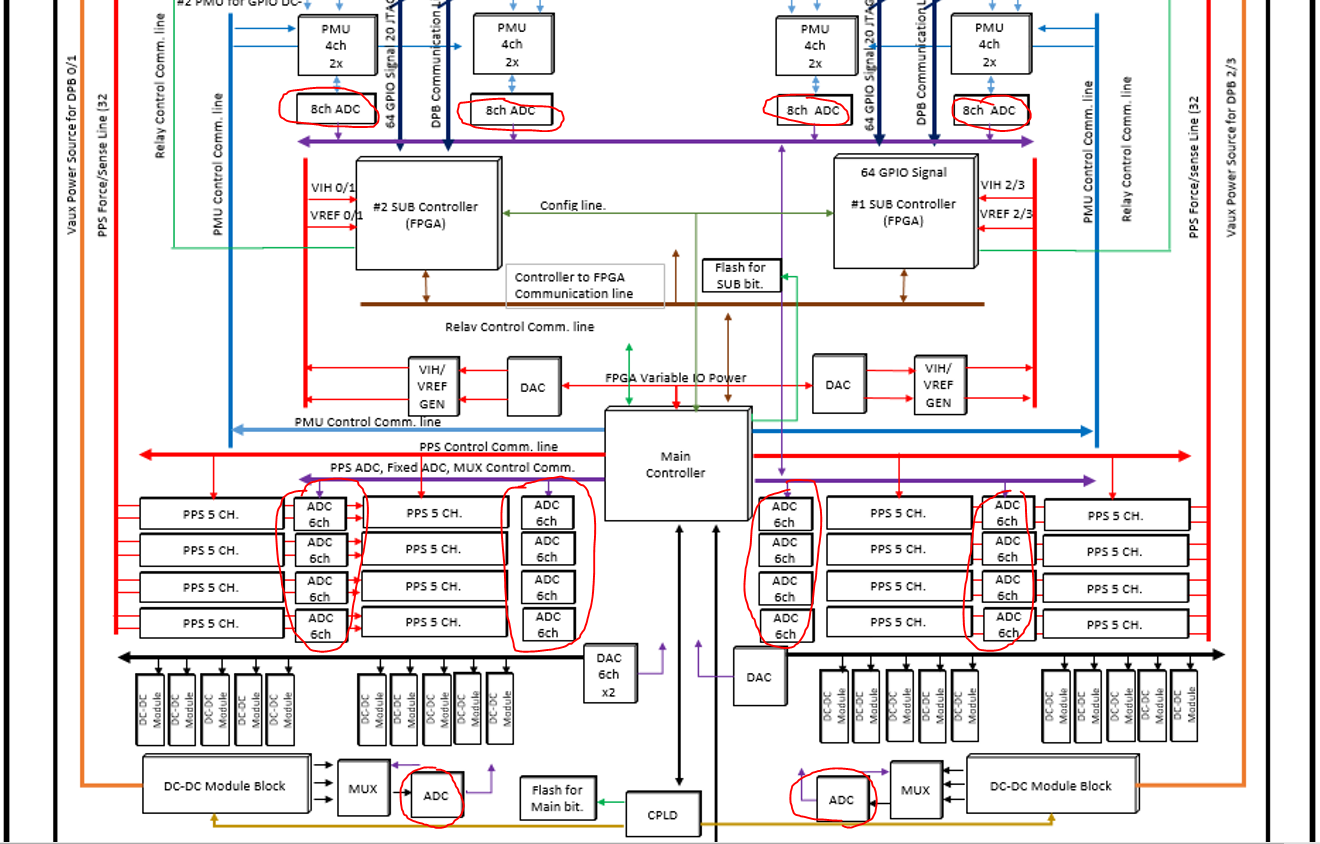


그림 9, ADC HW구성 Block Diagram

### ADC Calibration

* ADC는 DAC보다 먼저 해야 한다.
* PGB가 MASK 상태이면 해당 PGB는 CAL을 진행하지 않는다.
* POWER CHIP의 GAIN, OFFSET 값을 구해야 한다.
* READ 때는 사용자가 기입한 MEASURE 횟수만큼 읽은 후 MEASURE 횟수로 나눠서 평균값을 적용한다.
* GAIN : (FORCE REF V – FORCE GND V) / ( SENSE REF V – SENSE GND V )
* OFFSET : FORCE GND V – ( GAIN \* SENSE GND V )
* 해당 CHIP의 GAIN과 OFFSET 값을 저장한다.

### ADC Calibration Check

* READ ADC Channel에 ADC CAL로 구해진 GAIN을 곱하고 OFFSET 값을 더한다.
* 사용자가 입력한 Tolerance + FORCE V 와 Tolerance - FORCE 값 사이에 (READ ADC \* GAIN + OFFSET)이 들어가면 PASS 그게 아니라면 FAIL이다.
* GUI와 Terminal 창에 결과를 출력하고, Log로 저장한다.

### PPS Control

* Test 진행을 위해 각 DUT voltage를 인가 하기 위하여, PPS 를 제어하는 API를 제공
* DUT당 5 channels로 총 3840 Channel, 0 ~ 13.8V, Max. 1.0A
* PPS current measurement는 5uA/25uA/250uA/2.5mA/25mA/500mA/1.0A 가능

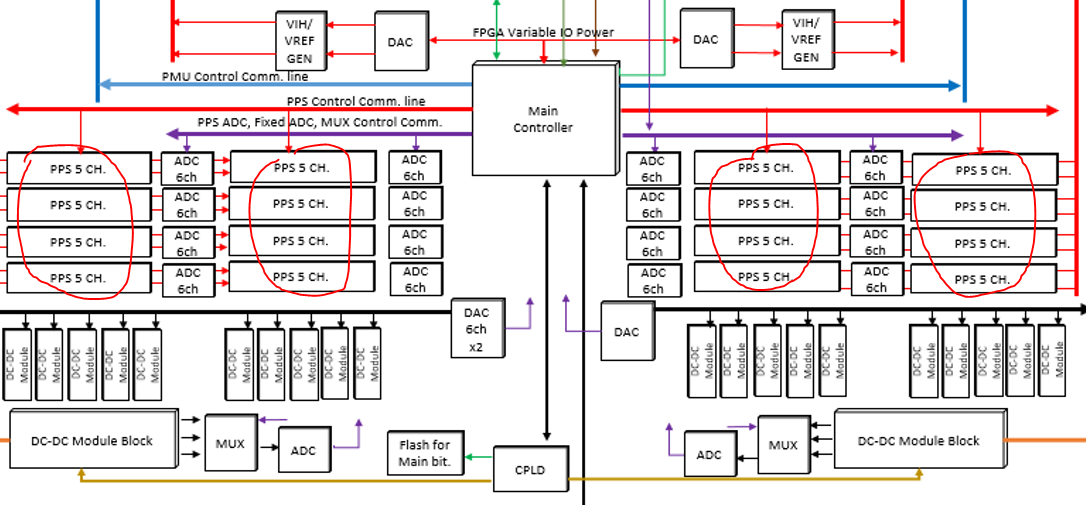


그림 10, PPS HW구성 Block Diagram

* PPS Output Range 0~13V 로 PPS 전원을 가변 하여 PPS TJ(발열)을 줄이기 위하여 PPS Power를 Share 한다.

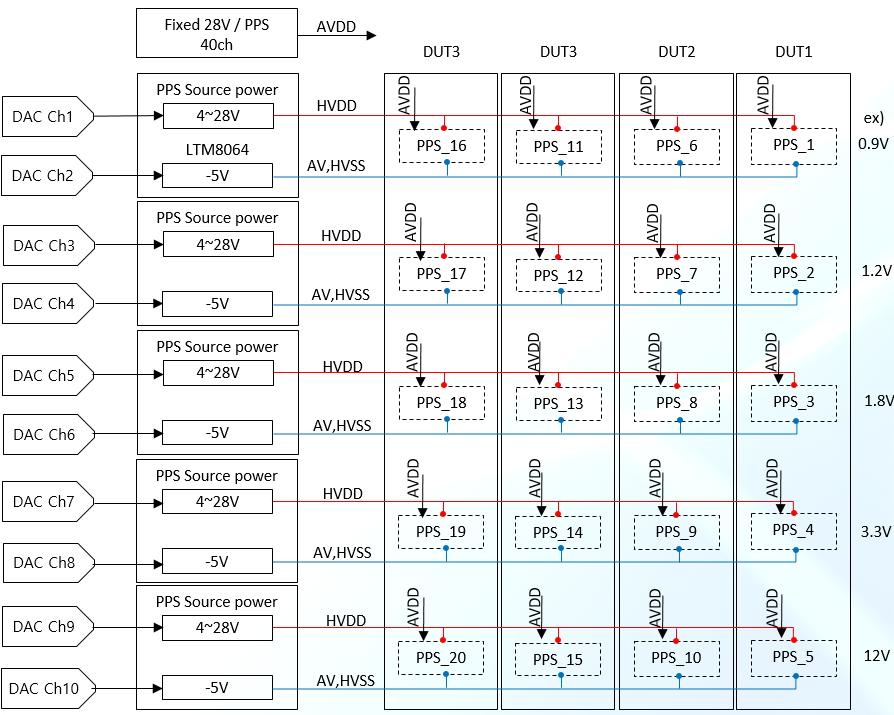


그림 11, PPS Power Share Block Diagram

* PPS Input power scheme

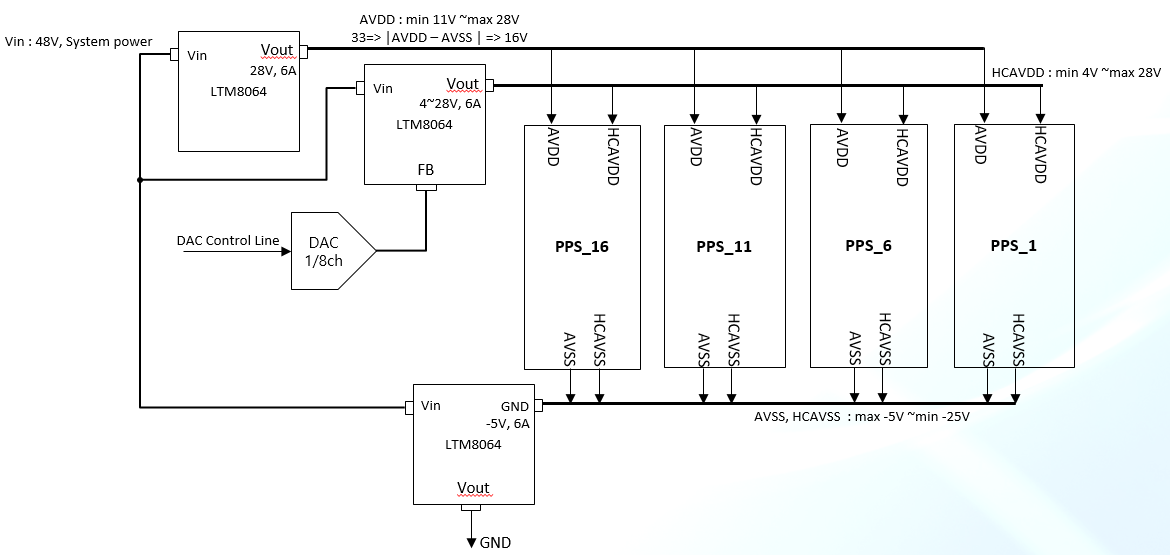


그림 12, PPS Input power scheme Block Diagram

* + Block Diagram은 PPS 4CH/Group표현이고 동일 구조로 전체 PPS 80CH,

20Group으로 구성

* + External Force 500mA, 1.2A Range는 TJ((39’C/W)를 줄이기 위해 HCAVDD는

전원 가변 되도록 설계

* + PPS Output Range : 0~25V

### PPS Calibration

* ADC Cal 적용이 된 상태에서 실행 되어야 한다.
* PGB가 MASK 상태이면 해당 PGB는 CAL을 진행하지 않는다.
* 사용자가 선택한 Power 종류만 진행할 수 있도록 한다.
* CAL은 사용자가 입력한 MIN Voltage부터 시작해 Step 범위만큼 증가 시키면서 MAX Voltage까지 실행한다.
* 사용자가 입력한 Measure 횟수는 Set Power 후 측정을 Measure 횟수만큼 반복 한 후 Measure 횟수로 나눈다.
* GAIN & OFFSET구하기 (H/W에게 자문 구하기)
  + Pps\_Meas\_Count : (MAX V – MIN V) / STEP V
  + SumF : MIN V ~ MAX V 까지 Force V 총합
  + SumS : MIN V ~ MAX V 까지 Sense V 총합
  + SumFMF : MIN V ~ MAX V 까지 (Force V \* Force V) 총합
  + SumFMS : MIN V ~ MAX V 까지 (Force V \* Sense V) 총합
  + GAIN : ((Dac\_Meas\_Count \* SumFMF) – (SumF \* SumF)) / ((Dac\_Meas\_Count \* SumFMS) – (SumF \* SumS))
  + OFFSET : (SumF - ( GAIN \* SumS)) / Dac\_Meas\_Count
  + Channel별 GAIN, OFFSET 값을 저장한다.
  + Channel별 GAIN, OFFSET 값을 Terminal 창에 출력하고, Log에 저장한다.

### PPS Current Measure Calibration

* ADC Cal 적용이 된 상태에서 실행 되어야 한다.
* PGB가 MASK 상태이면 해당 PGB는 CAL을 진행하지 않는다.
* Current Cal을 진행 하기 위하여, Cal 저항 변경해가며 진행 해야 한다. 이를 위해서 저항 전단에 위치한 스위치를 Control 해가며 Cal을 진행해야 한다.
* CAL은 사용자가 입력한 MIN Voltage부터 시작해 Step 범위만큼 증가 시키면서 MAX Voltage까지 실행한다.
* 사용자가 입력한 Measure 횟수는 Set Power 후 측정을 Measure 횟수만큼 반복 한 후 Measure 횟수로 나눈다.
* GAIN & OFFSET구하기
  + Pps\_Meas\_Count : (MAX V – MIN V) / STEP V
  + SumF : MIN V ~ MAX V 까지 Force V 총합
  + SumS : MIN V ~ MAX V 까지 Sense V 총합
  + SumFMF : MIN V ~ MAX V 까지 (Force V \* Force V) 총합
  + SumFMS : MIN V ~ MAX V 까지 (Force V \* Sense V) 총합
  + GAIN : ((Dac\_Meas\_Count \* SumFMF) – (SumF \* SumF)) / ((Dac\_Meas\_Count \* SumFMS) – (SumF \* SumS))
  + OFFSET : (SumF - ( GAIN \* SumS)) / Dac\_Meas\_Count
  + Channel별 GAIN, OFFSET 값을 저장한다.
  + Channel별 GAIN, OFFSET 값을 Terminal 창에 출력하고, Log에 저장한다.

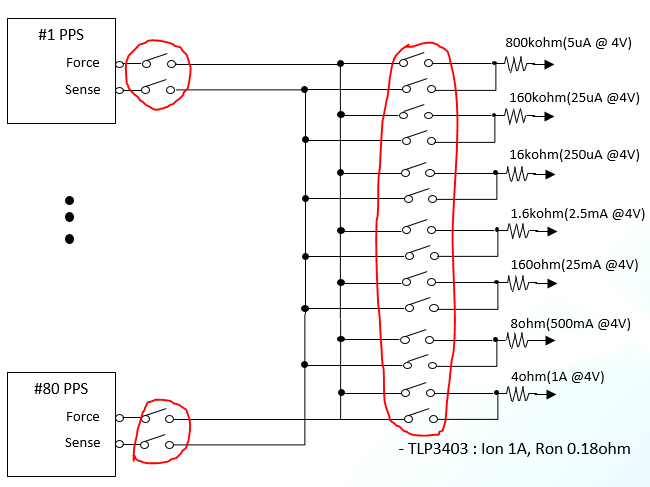


그림 13, PPS Current measure Cal Block Diagram

### PPS Calibration Check

* LOAD ADC, PPS CAL RESULT가 되어 있는 상태에서 진행 한다.
* 사용자가 입력한 Measure 횟수는 Set Power 후 측정만 Measure 횟수만큼 반복 한 후 Measure 횟수로 나눈다.
* Power Read 하고 ADC Cal에서 GAIN과 OFFSET 값을 적용한다.
* 위의 값을 가지고 PASS / FAIL을 결정한다.
* GUI와 Terminal창에 MEASURE 값, ERROR 값, PASS / FAIL을 출력하고 Log에 저장한다.

### PMU Control

* PMU HW구성

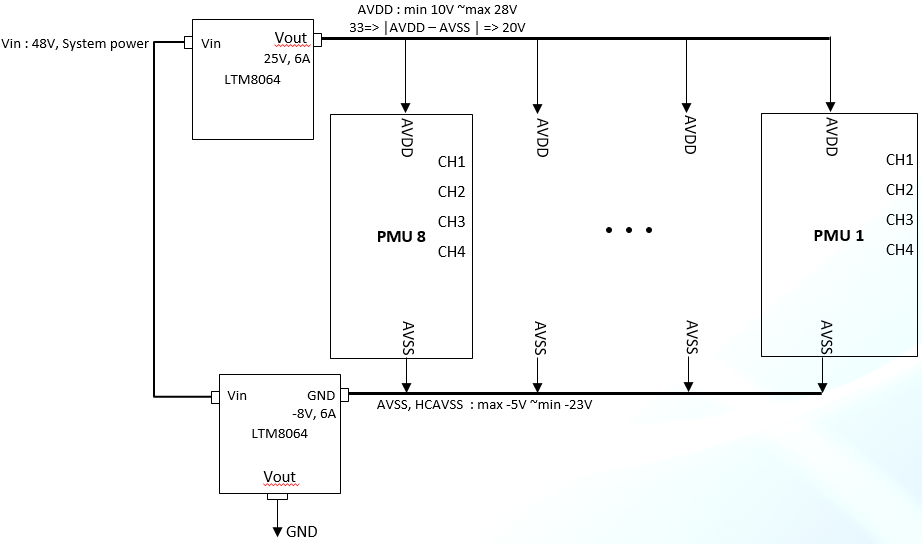


그림 14, PMU HW 구성 Block Diagram

* + PMU 총 개수는 4ch/PMU 8개 32ch, DUT 마다 GPIO, HS DC-Test 용 2ch 할당
  + Vout Range : -3 ~ 20V
  + Current Range : +/- 5uA, 20uA, 200uA, 2uA, 25mA
* VSIM, ISVMm VM 제어 필요

### PMU Calibration

* ADC Cal 적용이 된 상태에서 실행 되어야 한다.
* PGB가 MASK 상태이면 해당 PGB는 CAL을 진행하지 않는다.
* Current Cal을 진행 하기 위하여, Cal 저항 변경해가며 진행 해야 한다. 이를 위해서 저항 전단에 위치한 스위치를 Control 해가며 Cal을 진행해야 한다.
* CAL은 사용자가 입력한 MIN Voltage부터 시작해 Step 범위만큼 증가 시키면서 MAX Voltage까지 실행한다.
* 사용자가 입력한 Measure 횟수는 Set Power 후 측정을 Measure 횟수만큼 반복 한 후 Measure 횟수로 나눈다.
* GAIN & OFFSET구하기
  + Pmu\_Meas\_Count : (MAX V – MIN V) / STEP V
  + SumF : MIN V ~ MAX V 까지 Force V 총합
  + SumS : MIN V ~ MAX V 까지 Sense V 총합
  + SumFMF : MIN V ~ MAX V 까지 (Force V \* Force V) 총합
  + SumFMS : MIN V ~ MAX V 까지 (Force V \* Sense V) 총합
  + GAIN : ((Dac\_Meas\_Count \* SumFMF) – (SumF \* SumF)) / ((Dac\_Meas\_Count \* SumFMS) – (SumF \* SumS))
  + OFFSET : (SumF - ( GAIN \* SumS)) / Dac\_Meas\_Count
  + Channel별 GAIN, OFFSET 값을 저장한다.
  + Channel별 GAIN, OFFSET 값을 Terminal 창에 출력하고, Log에 저장한다.

### PMU Current Measure Calibration

* ADC Cal 적용이 된 상태에서 실행 되어야 한다.
* PGB가 MASK 상태이면 해당 PGB는 CAL을 진행하지 않는다.
* Current Cal을 진행 하기 위하여, Cal 저항 변경해가며 진행 해야 한다. 이를 위해서 저항 전단에 위치한 스위치를 Control 해가며 Cal을 진행해야 한다.
* CAL은 사용자가 입력한 MIN Voltage부터 시작해 Step 범위만큼 증가 시키면서 MAX Voltage까지 실행한다.
* 사용자가 입력한 Measure 횟수는 Set Power 후 측정을 Measure 횟수만큼 반복 한 후 Measure 횟수로 나눈다.
* GAIN & OFFSET구하기
  + Pmu\_Meas\_Count : (MAX V – MIN V) / STEP V
  + SumF : MIN V ~ MAX V 까지 Force V 총합
  + SumS : MIN V ~ MAX V 까지 Sense V 총합
  + SumFMF : MIN V ~ MAX V 까지 (Force V \* Force V) 총합
  + SumFMS : MIN V ~ MAX V 까지 (Force V \* Sense V) 총합
  + GAIN : ((Dac\_Meas\_Count \* SumFMF) – (SumF \* SumF)) / ((Dac\_Meas\_Count \* SumFMS) – (SumF \* SumS))
  + OFFSET : (SumF - ( GAIN \* SumS)) / Dac\_Meas\_Count
  + Channel별 GAIN, OFFSET 값을 저장한다.
  + Channel별 GAIN, OFFSET 값을 Terminal 창에 출력하고, Log에 저장한다.

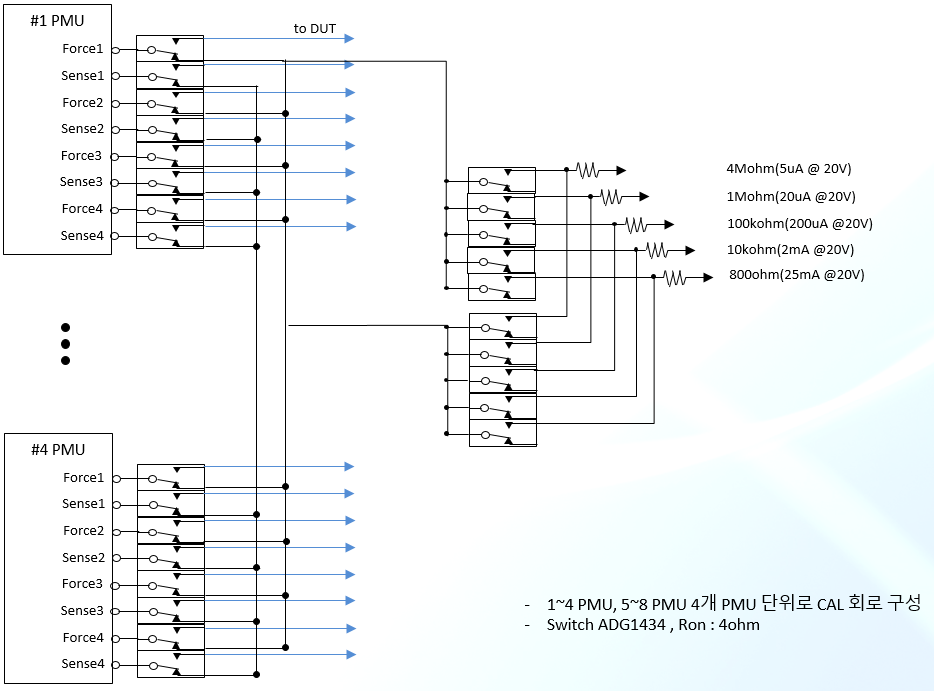


그림 15, PMU Current measure Cal Block Diagram

* + 1~4 PMU, 5~8 PMU 4개 PMU 단위로 CAL 회로 구성
  + Switch ADG1434 , Ron : 4ohm

### PMU Calibration Check

* LOAD ADC, PMU CAL RESULT가 되어 있는 상태에서 진행 한다.
* 사용자가 입력한 Measure 횟수는 Set Power 후 측정만 Measure 횟수만큼 반복 한 후 Measure 횟수로 나눈다.
* Power Read 하고 ADC Cal에서 GAIN과 OFFSET 값을 적용한다.
* 위의 값을 가지고 PASS / FAIL을 결정한다.
* GUI와 Terminal창에 MEASURE 값, ERROR 값, PASS / FAIL을 출력하고 Log에 저장한다.

### DC-TEST

* DPB Board에 위치
* PMU Force line 은 BCB 로 부터 받음
* 아직 자세한 내용을 파악하기 어려움 추후 내용 보강 예정

### OCP

* 각 SUB 과제별(PCIE, ASI, UFS) OCP Source tree UNI53K로 변경 필요
* Base 과제인 UNI92K에서 HW가 변경 되거나 추가된 부분을 수정 필요 사항
* 각 SUB 과제별(PCIE, ASI, UFS) OCP Source 수정 시 고려 사항
  + GPIO Control
    - DUT별 21CH
  + PPS Control
    - DUT별 5CH
    - 0 ~ 13.8V, Max. 1.0A
    - Force, Sense, Power Sharing
  + PMU Control
    - DUT별 2CH(?)
    - -2V ~ 20V, Max. ±25mA
    - PMU current Measure Range : 5uA/20uA/200uA/2mA/25mA
    - VSIM / ISVM / VM
    - DC-TEST
  + Test Log 저장 방식 변경
    - Base Project대비 Test Log 저장 방식이 변경됨
    - 기존 NFS를 이용 하여 Host PC에 Test Log File을 저장 하던 방식에서 TCP protocol에서 packet을 전송하여 Host PC에 Test Log를 저장 하도록 수정 필요
      * OCP에서 TCP protocol을 이용 하여 Packet을 전송.
      * OS에서는 OCP에서 보낸 TCP Packet를 Parsing하여 각 DUT별 Test Log file을 생성.
  + PG UART 동작 Concept 변경
    - 기존 장비들은 해당 Uart Port 를 연결하기 위해 Cable 을 PC에 연결
    - 기존 동작 Concept은 SYSTEM에서는 연결이 불가능 할 수도 있음
    - 이러한 문제점을 해결 하기 위해 UART to Ethernet Path 확보 필요

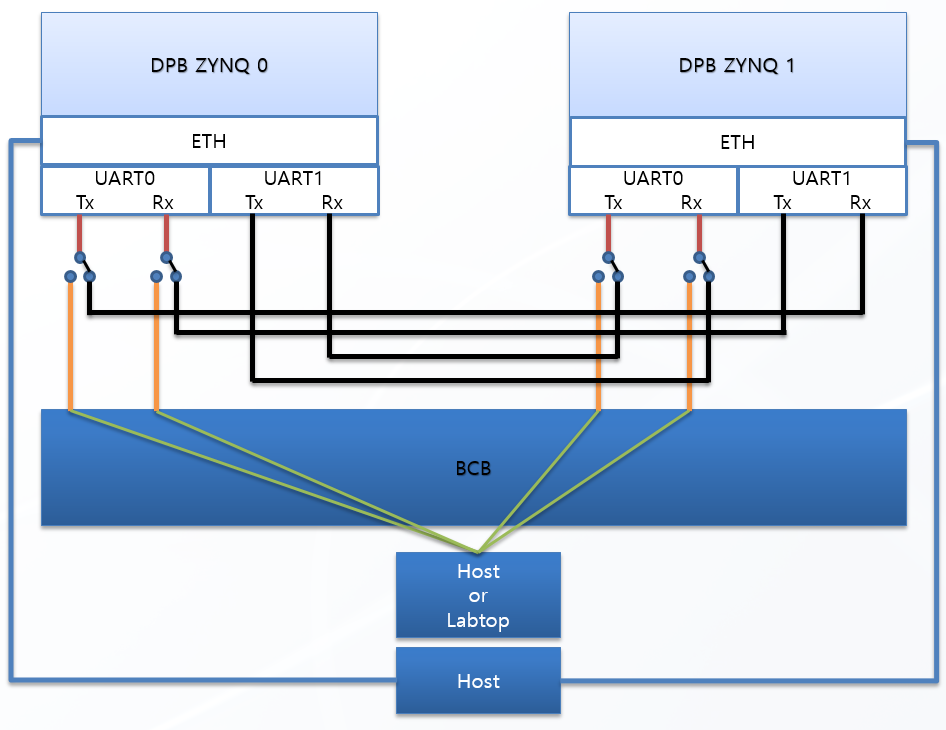


그림 16, UART to Ethernet Convertor Block Diagram

* + - * 스위치가 연결 되어 있어 아래의 UART PATH을 결정 할 수 있다.
        + PG0, PG1의 Uart Tx, Rx가 Cross로 연결되는 PATH
        + USB Port로 바로 연결 되는 PATH
      * PG0, PG1의 UART가 Cross로 연결될 경우 OCP에서 Cross연결된 UART의 LOG를 HOST PC에 저장.

# Software Reuse

* UNI92K를 BASE로 만든다.
* USER API는 UNI92K를 BASE로 하며, HW 변경으로 인한 사항을 추가 한다.
  + API 관련 내용은 API 문서를 참조한다.
* GUI의 Mainframe Layout은 기존 Uni92k의 GUI를 약간의 수정으로 사용 하며, Utility는 기존 GUI를 사용 한다.
* Pcal, Diag는 GUI 변경 최소화로 진행하고, Item 관련 내용에 대해서 추가 Update 한다.
* F/W Download는 Uni92k GUI및 기능을 그대로 사용 하도록 한다.
* Analysis Utility는 기존 GUI를 사용 하며, 기능도 기존 기능을 그대로 사용 한다.
* Lot Setup, Lot Reset, Initialize, Test Start등 Test 관련 기능들은 기존 GUI및 기능을 그대로 사용 한다.
* Recipe Utility는 기존 GUI를 사용 하며, 기능도 기존 기능을 그대로 사용 한다.
* OCP Test Program은 상기(3.2.27)에 언급한 내용을 제외한 나머지를 그대로 사용한다.
  + PL의 Register 변경 및 내용 추가 시 해당 내용 수정 필요
* HOST <-> OCP간 통신 하는 내용중 Test Log를 제외한 나머지는 기존 Concept를 사용한다.
* Network 구성은 UNI92K와 거의 같지만 SYSTEM에서 PG가 많아질 경우에 하기 그림 과 같이 구성 해야 한다.(PG기준으로 구성 되었으며, Main CTRL은 구성에 포함 하지 않음)

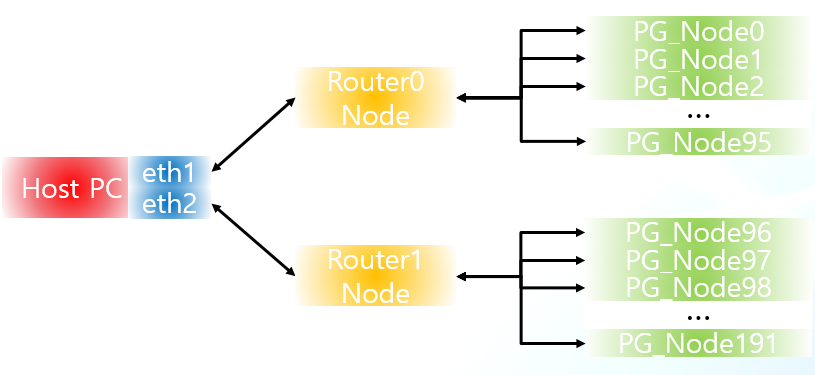


그림 17, UNI53K PG Network 구성

* SERVER Part ( TBD )
* Pattern Compiler는 Instruction map에 따라 변경 할 수 있다.
* Socket Compiler는 Pin map, Channel map에 따라 변경 가능 하다.

# 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 |