

# TUNA 2 Board Technical Report

21-Mar-2017

이태민, [leetaemin713@gmail.com](mailto:leetaemin713@gmail.com)

이도근, [matureelf@hanyang.ac.kr](mailto:matureelf@hanyang.ac.kr)

본 보고서는 TUNA 2.X 보드의 사용법을 간단하게 적은 것이다. 내용은 다음을 포함한다.

- Tool setting: PC에서 TUNA 보드를 접근할 때 필요한 도구에 대한 내용이다.
- Part system test: TUNA 보드의 핵심 기능과 주변 환경의 디버깅을 위한 내용이다.
- Full system: Ubuntu 동작하는 시스템의 동작에 대한 내용이다.
- Programming interface (TBD)

## Tool setting

본 테스트에 사용한 도구 및 버전은 아래와 같다.

- 보드: TUNA V2.1 (2016.04.29)
- JTAG: Xilinx Platform Cable USB II
- USB 드라이버: CP201X USB to UART Bridge VCP Drivers
- 터미널: Xshell 4, Baud Rate=115200
- 모니터: Alphascan AOC 2470 IPS MHL+DP
- XMD console: Xilinx SDK, Release Version 2016.2

각 도구는 아래의 link를 참조하면 추가 정보를 얻을 수 있다.

<https://www.xilinx.com/products/boards-and-kits/hw-usb-ii-g.html>

<http://www.silabs.com/products/mcu/pages/usbtouartbridgevcpdrivers.aspx>

<http://prod.danawa.com/info/?pcode=2516975&cate=112757>

<http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/2016-2.html>

### Part system test

TUNA 보드의 핵심 기능 3가지(HDMI, latency 모사, 비휘발성 모사)를 각각 테스트할 수 있도록 바이너리가 제공되어 있다. 테스트 방법은 SD카드에 바이너리를 복사해서 진행된다. SD카드는 linux fdisk tool을 이용해 첫 번째 partition은 1GB의 FAT으로, 두 번째 partition은 남은 영역 모두 Linux로 포맷하여 준비한다. 각각의 바이너리는 SD카드의 FAT partition에 복사되어야 된다.



- Figure 1 TUNA 보드 주변 장치 연결

보드에는 6개의 케이블을 연결하는데, 각각 아래와 같다. (그림 1 참조)

- 파워케이블, JTAG, HDMI, Ethernet, USB-UART (mini-usb), USB mouse (micro-usb)

보드의 USB-UART가 연결된 PC에서 터미널 프로그램 (ex. Xshell) 을 이용한다. 이 터미널을 통해 보드의 stdin/stdout을 입력/출력할 수 있다.



- Figure 2 HDMI 출력 테스트

## HDMI 테스트

제공된\_폴더/part/HDMI/BOOT.bin을 SD카드에 복사한 후 SD카드를 보드에 mount한다. 보드에 전원을 키면 터미널에는 log가 출력되고 모니터에는 아래의 그림과 반복적인 beep음이 출력된다. 터미널을 통해 '0'~'6'을 입력하면 출력 해상도를 조절할 수 있다.

## Latency 모사 테스트

제공된\_폴더/part/latency/BOOT.bin을 SD카드에 복사한 후 SD카드를 보드에 mount한다. 보드에 전원을 키면 터미널에 log가 출력된다.

TUNA 보드의 NVM 영역은 0x8000\_0000 ~ 0xBFFFF\_FFFF (1G)로 설정되어 있다. 유저는 이 영역에 대해 지연시간을 설정할 수 있다. 쓰기 지연 시간은 0x43C0\_0000 영역에 4-byte integer를 씬으로써 조절할 수 있고, 읽기 지연 시간은 0x43C0\_0004 영역에 4-byte integer로 조절한다. 처음 보드를 켤 경우 쓰기/읽기 지연 시간은 0 값이 임의로 설정되어 있다. (pre-defined 지연 시간에 대해서는 추가 정보 확인) 본 테스트 app은 쓰기와 읽기 지연 시간을 128\*iter 로 설정한 후 1024번의 4-byte 쓰기/읽기 요청에 대한 지연 시간을 측정한다.

## 비휘발성 모사 테스트

제공된\_폴더/part/nonvolatility/BOOT.bin을 SD카드에 복사한 후 SD카드를 보드에 mount한다. 보드에 전원을 키면 터미널에 log가 출력된다.

이 테스트 프로그램은 NVM 영역 일부(2MB)에 특정 값을 쓴 후 휘발되는지를 확인한다. TUNA 보드 NVM 영역의 데이터에 손실이 없기 위해서는 FPGA 이미지를 re-download하지 않은 채로 시스템을 재부팅해야 한다. 본 예제에서는 JTAG 하드웨어 디버거를 통해 시스템을 재부팅하는 방법을 쓰는데, 유저는 TCL 스크립트를 실행하는 방식이다.

Xilinx SDK의 XMD console을 열고 아래 커맨드를 입력한다.

- XMD% cd 제공된\_폴더/part/nonvolatility/tcl로
- XMD% source ltmin\_part.tcl

본 스크립트가 실행된 후 number of mismatch = 0이 되는 것, 즉 데이터가 보존되는 것을 확인할 수 있다.

## Full system

TUNA 보드의 핵심 기능 3가지 (HDMI, latency 모사, 비휘발성 모사)를 포함하는 예제 시스템이다. ARM용 Ubuntu인 linaro-ubuntu-desktop이 설치되어 있다. 제공된\_폴더/full/SD/fat의 내용을 SD카드의 첫 번째 파티션에 복사한 후, 제공된\_폴더/full/SD/ext에 있는 download 파일의 링크를 열어 SD카드의 두 번째 파티션에 복사한다.

- tar --strip-components=3 -C /마운트폴더/ -xzf linaro-precise-ubuntu-desktop-20121124-560.tar.gz binary/boot/filesystem.dir

SD카드를 보드에 mount 한 후 전원을 키면 터미널에는 root 사용자의 log가 출력되고 모니터에

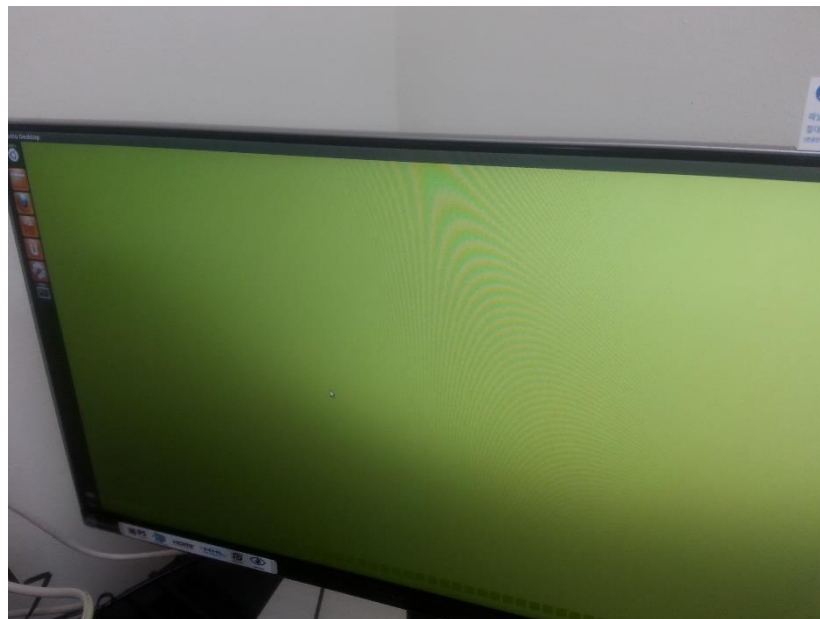
는 Ubuntu GUI가 출력된다.

Linux 환경에서는 SW stack의 복잡도 때문에 Latency 모사의 결과가 가시적이지 않다. 따라서 해당 테스트는 U-boot단에서 간접적으로 확인하는 것으로 대체한다.

- Zynq> md 0x80000000
- Zynq> mw 0x43c00004 0xa0000
- Zynq> md 0x80000000
- (읽는 속도 느려짐)

비휘발성 모사는 마찬가지로 TCL 코드를 실행함으로써 가능하다. 먼저 ubuntu를 poweroff한 후 Xilinx SDK의 XMD console을 열어 cd 제공된\_폴더/full/tcl 및 source ltmin\_reboot.tcl을 입력한다. 시스템이 재부팅되는 것을 볼 수 있다.

TCL 코드의 126번 줄을 보면 유저는 FPGA 이미지의 re-download 유무를 선택해 재부팅할 수 있다. FPGA re-download 방식은 DRAM 초기화 과정을 재실행하기 때문에 NVM의 비휘발성이 완전하지 않다. FPGA no-download 방식은 NVM의 데이터가 완전하게 보존되나 HDMI VDMA 문제로 ubuntu GUI가 동작하지 않는다. USER는 본인의 응용에 따라 재부팅 방식을 결정할 수 있다.



- Figure 3 Full system – Ubuntu GUI