

# Xilinx FPGA, TI DSP·MCU

## 기반의 회로 설계 및 임베디드 전문가 과정

최준호  
계획/성과  
3주차



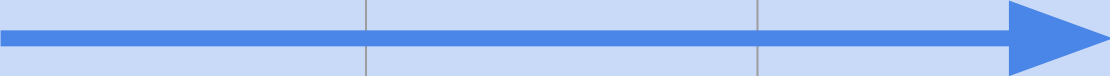
# 목차

- 내 역할
- 전체 일정
- FPGA 일정
  - FPGA 다음 주 목표
- DSP 일정
  - DSP 다음 주 목표
- 지난 주 성과

# 내 역할


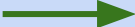
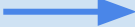
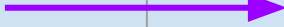

- DSP Linux Device Driver 개발 및 각 장치 구현
- FPGA Linux Porting 및 Linux Device Driver 개발 및 장치 개발 및 PL 구현

# 전체 일정

구분	파트	05月	06月	07月
FPGA	Firmware			
DSP	Firmware			
	LDD			



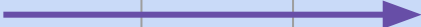
\*LDD : Linux Device Driver

# FPGA 일정

파트 \ 주차	03	04	05	06	07	08	09
FPGA 교재							
Linux Porting							
PL 구현							
Motor Control (PL or PS) 구현							
CAN, DMA 구현							

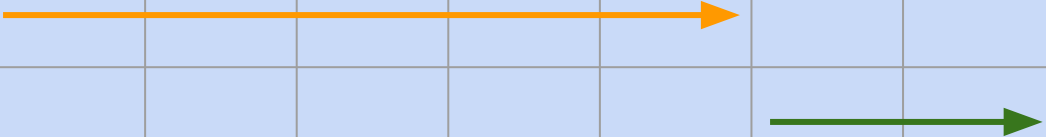
# FPGA 다음 주 목표

5월 4주차

파트 \ 일	22	23	24	25	26	27	28
Linux Porting 실습							
Linux Porting							
Zynq PL 조사 및 PL Control							

# DSP 일정

파트 \ 주차	03	04	05	06	07	08	09
LDD							
DSP, DMA 구현							



# DSP 다음 주 목표

5월 4주차

파트 \ 일	22	23	24	25	26	27	28
LDD 간단한 드라이버 작성 실습							
DSP or FPGA 데이터시트 연계한 LDD 작성							



지난 주 성과

# FPGA 지난, 이번 주 목표

8일

\*ZYNQ 이론 완료  
Overview PS 부분은 끝났고 PL 부분 공부하면 됨.

\*ZYNQ 펌웨어 학습  
우분투 Vivado setup 끝났고 산 교재 꼼꼼히 Overview에서 공부한 내용이랑 비교해보며 공부.  
확실히 겹치는 게 많아서 도움이 많이 됐다고 생각.  
내일 학습할 부분부터 실제 예제 구현이기 때문에 시간이 많이 들 수 있다고 생각함.  
실습 하면서도 아는 부분에 대해선 꼼꼼히 체크하면서 공부하는 게 필요하다고 생각.

9일

\*ZYNQ 이론 완료  
Overview 모두 끝냈지만 잊힐 것 같기도 하고 하여,  
이번 주 주말에 총 복습 한 번 하는 게 좋을 것 같다.  
복습할 땐 디지털 문서로 남겨보기로 한다.

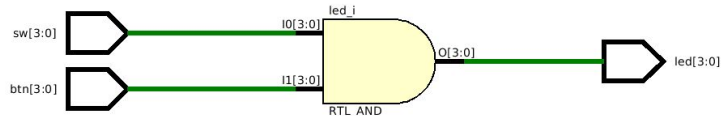
11일

\*ZYNQ 펌웨어 학습 완료(~51)  
Zybo와 vivado를 통해 교재에 나와 있는 대로 AND 상태일 때 LED 켜는 것을 실습해 봤다.  
처음 써보는 것이기도 하고 교재에 상세한 설명이 적은 부분도 있어 찾으며 하느라 애 먹었다.  
특히 JTAG 관련 이슈 때문에 시간을 많이 썼었던 것 같다.  
FPGA에서 사용하는 언어가 새로워서 FPGA의 앞날이 많이 어려울 수 있을 것도 같다.  
Verilog나 VHDL로 구현할 수 있는데, 나는 이번에 VHDL로 했지만 다음에 Verilog를 써보든가 해서 더 쉬운 언어를 쓰는 것이 좋겠다.

15일	16일	17일	18일	19일	20일	21일
Zynq 설계 기본 52~104	Custom IP 및 Timer 설계 105~156	Zynq 설계 응용 157~175	리눅스 포팅에 대한 조사 1/2	리눅스 포팅에 대한 조사 2/2	리눅스 포팅 실습 1/4	리눅스 포팅 실습 2/4

# FPGA 지난 주 성과(1)

## 4sws, 4btn AND -> 4led

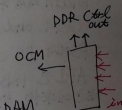


# FPGA 지난 주 성과(2)

## PS Inter-connect (= Inc)

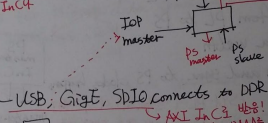
Inc는 많은 Peripheral(=PP)들이 논리적으로 연결되도록 하기 위해 만들어진 개념.  
즉, PP들이 서로 직접 연결되는 대신 Inc를 통해 연결됨.  
(물론 이 예외 Inc 없이 바로 연결되는 경우도 있음)

- 구조
- Programable Logic to Memory - 2 ports to DDR, 1 port to OCM SDRAM

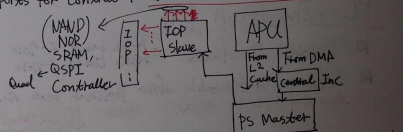


- Central Interconnect - Enables other interconnects to communicate with each other, handles all Inc traffic, handles all system traffic.

- Peripheral Master - USB, GigE, SDIO connects to DDR and PL



- Peripheral Slave - 2 ports for control for all Common Peripherals

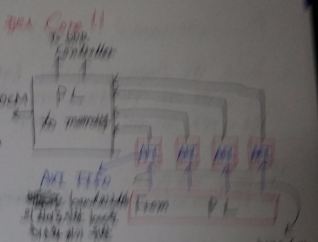


L2 cache → PS Master → IOP Slave  
Inc3.45  
Latency가 높을수록 이상!

## AXI Interfaces to PL

### AXI High Performance Ports (HPP)

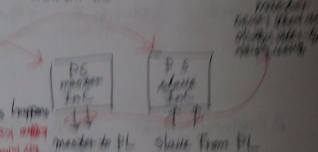
- 40-bit data bus: 32-bit data width to PL, 8-bit data width from PL
- OCM and DDR core out bus
- PS에 PS clock이 연결되어 Communication
- AXI는 HPP의 FIFO를 가지고 있어
- AXI는 HPP의 FIFO를 가지고 있어



### AXI General Purpose Ports

- 2 Masters from PS to PL
- 2 Slaves from PL to PS
- 32 bit data width
- Conversation and sync to PS clock happens in PS

PS에 PL의 클럭이 연결되어 있어  
마스터 1, 2가 PL에 클럭을 주면  
클럭이 연결된 마스터 1, 2가 동작한다.

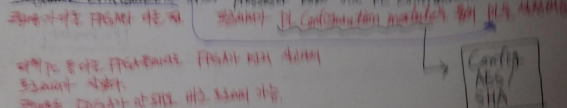


## PL - The FPGA Part

### Series 7 Fabric

Arithmetic based 2-nano and 2-nano  
Logic based 2-nano and 2-nano

### Software configurable after processor boot via PL configuration module



- Totally independent of PS - Spends on-chip power, manages clock and reset management, clock reset to PL, PL to PS, PL to PS, PL to PS.
- Debug port and management - coordinated with PS

# FPGA 지난 주 성과(3)

## PL Additional features

- ✎ Xilinx Analog to Digital Converter (XADC)
  - in all 2014-2020 dev
  - 1 Msps (sample per sec)
  - True 12 bit performance
  - Dual Converter for ~~12 bit~~ <sup>12 bit</sup> performance
- ✎ GTX - in 2-7030/7040 ...
- ✎ PCIe - in 2-7030/7040 ...
- Built-in chip temp and power sensors
  - 온도 센서
  - 전력 센서
  - 즉 바로 온도

## PS-PL Interfaces

- ✎ AMBA AXI interfaces for Primary data communication
  - 32-bit AXI master interfaces 2개
  - " " slave " 2개
  - 64/32-bit shared DDR memory - OCM에 빠르게 접근하기 위한 인터페이스 6개 (AFI slave interface)
- ✎ DMA, interrupts, events signals
  - CPU에 이벤트 신호 전송하기 위한 Processor Event bus
  - PL to IP interrupts to the PS GIC
  - PL을 위한 DMA 채널 신호 4개
  - 비동기적, 비정형 데이터 전송을 위한 신호들
- ✎ Extendable pin multiplexing (EMIO) allows unmapped PS PP to PL I/O
- ✎ Clock and resets
  - 100/200 MHz PS clock output to the PL 4개
  - PS reset outputs to the PL 4개
- ✎ Configuration and miscellaneous
  - clock output 4개
  - clock input 4개

## The PS Boot First

- ✎ The Device Configuration Unit (DEVCFG) initializes the PS and PL.
  - 프로그래머가 먼저 하는 일
- ✎ Processor memory boot sources
  - NOR - NAND - QSPI
  - PS가 부팅될 때까지 Cortex-A가 실행할 때까지 OCM에 저장된 ROM에
- JTAG (Not a memory device)
  - 실행 가능한 ROM에 저장된 NOR NAND QSPI 등 부트로더 로더를, 가변적 ROM에 저장된 부트로더 실행
- ✎ On chip boot loader in OCM ROM
  - Loads and executes first stage boot loader (FSBL)
  - FSBL boot source
    - ROM에 저장된 부트로더 실행
    - select via bootstrapping pins.
    - NOR, NAND, QSPI 등 하나 선택
    - 이 핀들을 통해서

## Configuration the PL

- ✎ The PL is configured after the PS boots
- ✎ performed by app software accessing the hardware Device Configuration Unit
  - Bitstream image transferred
    - PL 부트로더에 실행된 부트로더가
  - 100 MHz 32-bit P-CAP stream interface
    - 이후 부트로더 실행
    - (P-CAP, P-CAP)
  - Decryption/authentication hardware option
    - 부트로더 실행
    - 부트로더 실행 (부트로더 실행)

## Summary

- ✎ 2014-2020 Extensible Processing Platform 이다. (SOC + FPGA)
- ✎ PS는 hard silicon, dual core
  - PS boot
- ✎ PL은 Series 7 FPGA fabric
  - PL configuration
- ✎ PL은 PS의 AXI interface를 통해 (데이터 인터페이스)
- ✎ OCM에 FSBL 부트로더 실행을 위해 NOR, NAND, QSPI를 가변적으로

감사합니다