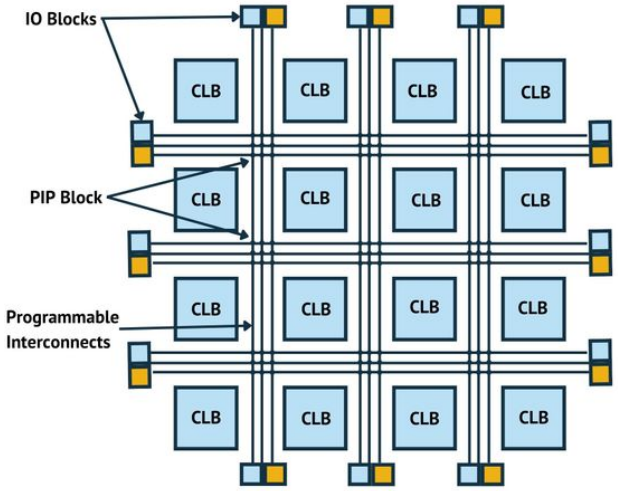
# Introduction to Zynq SoC

Erwin Setiawan

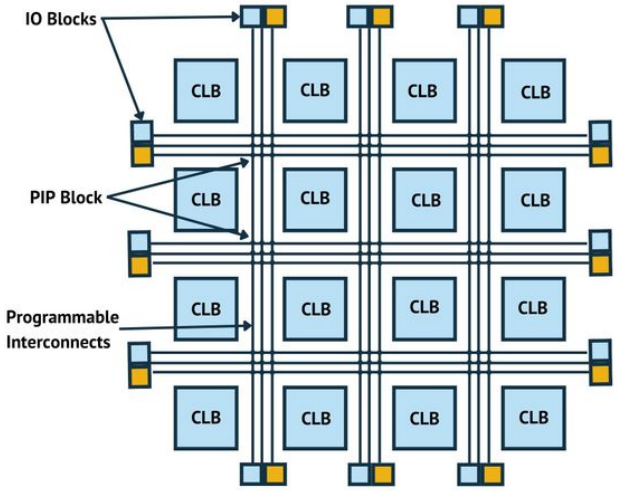
# What is FPGA

- Digunakan untuk menjalankan rangkaian digital yang dibuat dengan HDL (VHDL/Verilog).
- Internal structure dari FPGA terdiri dari block dasar configurable logic block (CLB).



# What is FPGA

- Digunakan untuk menjalankan rangkaian digital yang dibuat dengan HDL (VHDL/Verilog).
- Internal structure dari FPGA terdiri dari block dasar configurable logic block (CLB).




Semakin banyak jumlah CLB dalam FPGA, maka harganya semakin mahal.

## Zynq UltraScale+ RFSoC ZCU216 Evaluation Kit

by: AMD

AMD

Equipped with the industry's only single-chip adaptable radio device, the Zynq™ UltraScale+™ RFSoC ZCU216 evaluation kit, is the ideal platform for both rapid prototyping and high-performance RF application development.

Price: $15,546.00

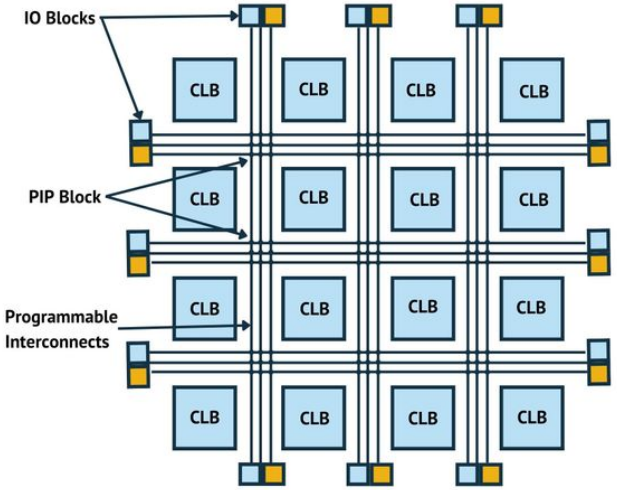
Part Number: [illegible]-ZCU216-V1-G

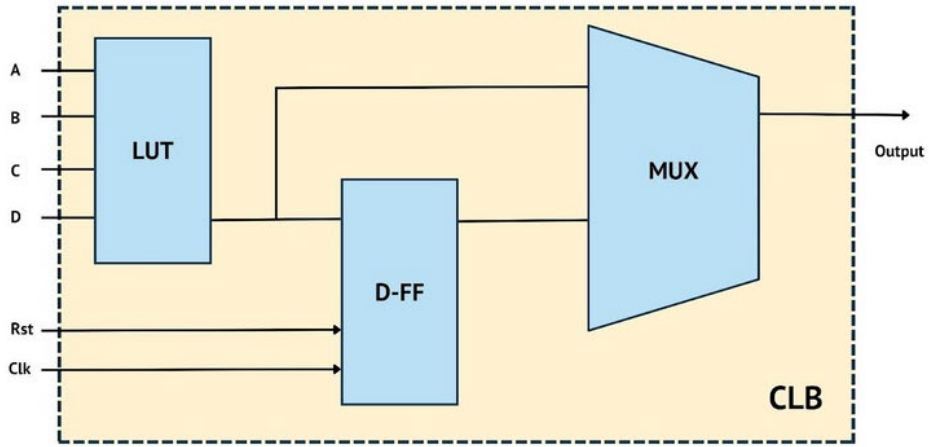
Lead Time: 6 weeks

Device Support: Zynq UltraScale+ RFSoC

# What is FPGA (cont.)

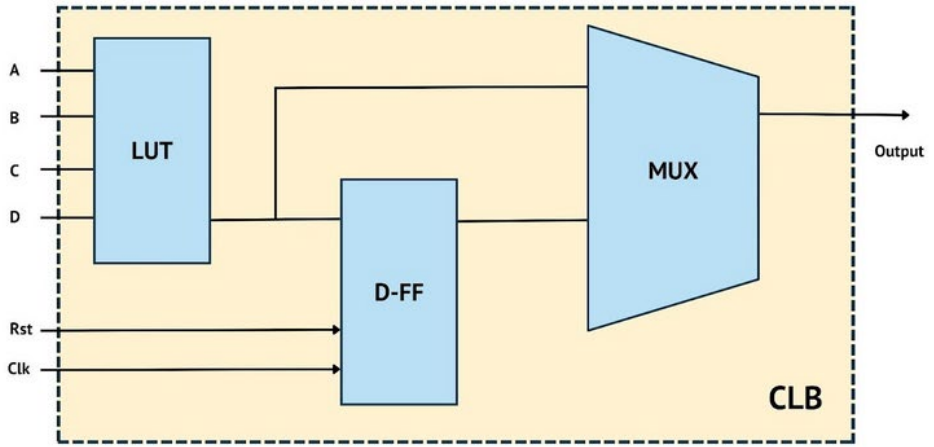
- Setiap brand FPGA (Altera, Xilinx) bisa memiliki arsitektur CLB yang berbeda.
- Block CLB ini merupakan gambaran sederhana CLB pada FPGA Xilinx 7 Series.



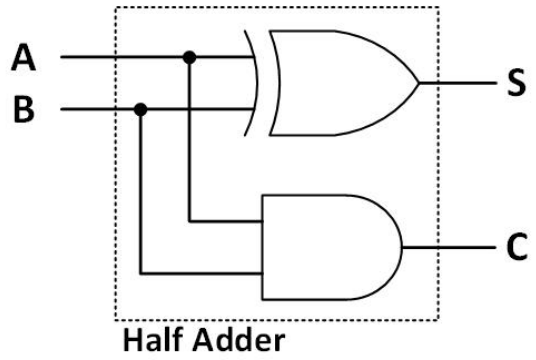
# What is FPGA (cont.)

- Terdiri dari LUT, D-FF, dan MUX.
- LUT biasa digunakan untuk rangkaian kombinasional dan D-FF untuk rangkaian sequensial.



# FPGA Compilation Flow

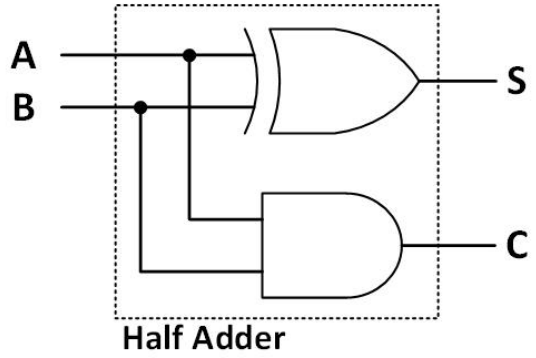
- Write Verilog code
- Simulation
- Synthesis
- Implementation
- Generate bit file




```verilog
module half_adder // Port declaration
    (
        input wire  a,
        input wire  b,
        output wire sum,
        output wire carry
    );

    // Module body
    assign sum = a ^ b; // Continuous assignment
    assign carry = a & b;

endmodule
```

# FPGA Compilation Flow (cont.)

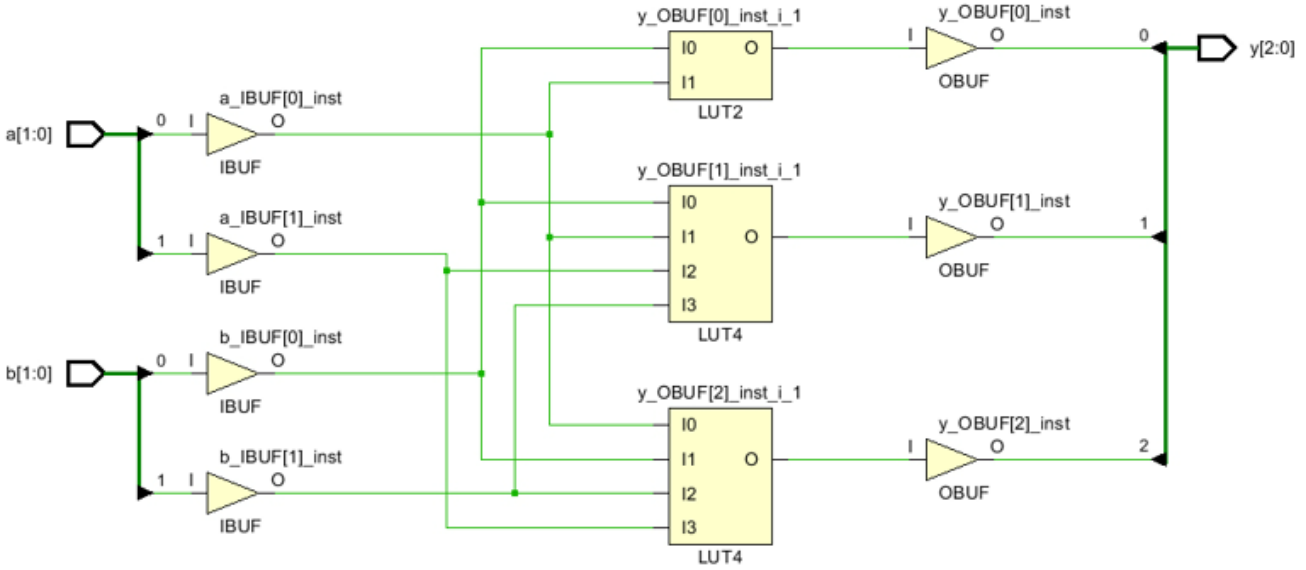
- Write Verilog code
- Simulation
- Synthesis
- Implementation
- Generate bit file



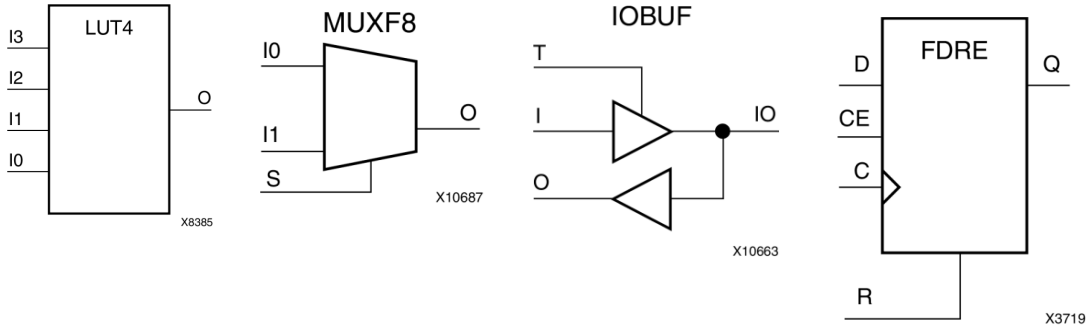

```verilog
module half_adder // Port declaration
    (
        input wire  a,
        input wire  b,
        output wire sum,
        output wire carry
    );

    // Module body
    assign sum = a ^ b; // Continuous assignment
    assign carry = a & b;

endmodule
```

# FPGA Compilation Flow (cont.)

- Write Verilog code
- Simulation
- Synthesis
- Implementation
- Generate bit file



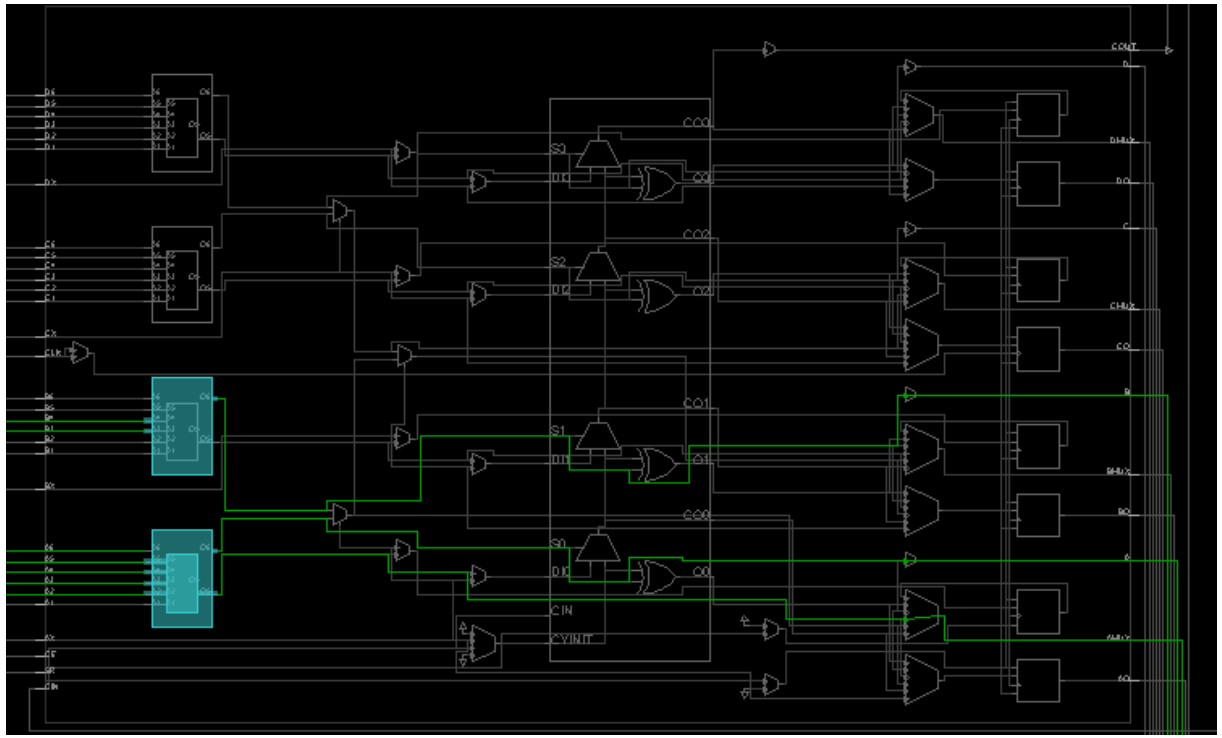

Proses synthesis mengubah kode Verilog menjadi primitives FPGA. Primitives pada FPGA yaitu LUT, MUX, carry, FF, DSP, Memory, IO, etc



# FPGA Compilation Flow (cont.)

- Write Verilog code
- Simulation
- Synthesis
- Implementation
- Generate bit file

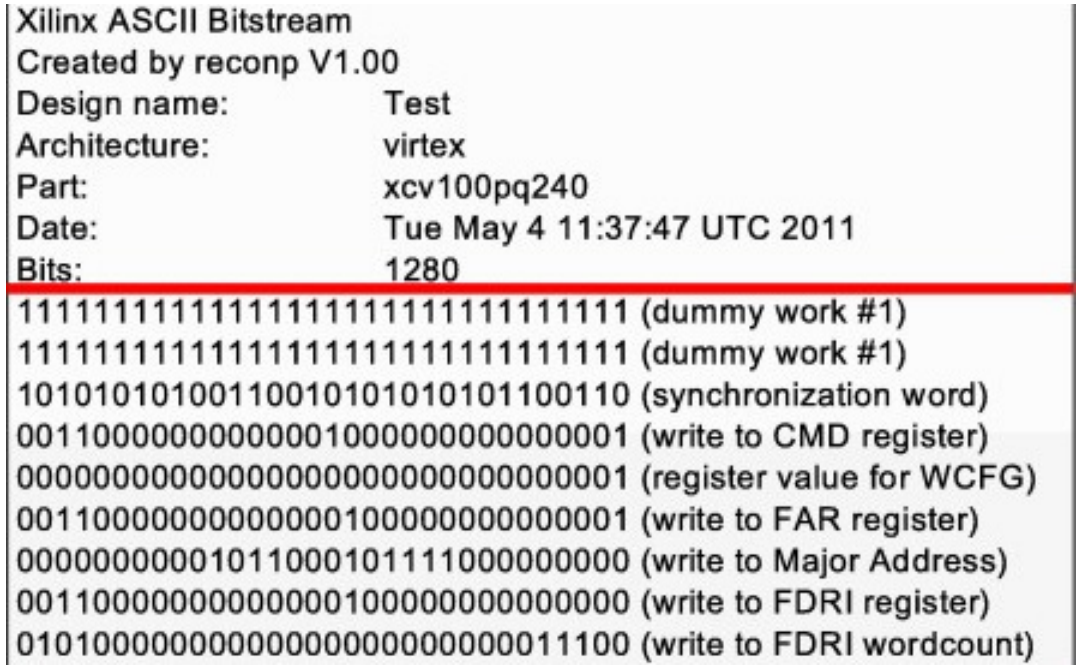
Pada proses implementasi, hasil synthesis akan dioptimasi, Dilakukan place and route untuk menghasilkan netlist serta jalur routing yang sesuai dengan target FPGA yang digunakan.



# FPGA Compilation Flow (cont.)

- Write Verilog code
- Simulation
- Synthesis
- Implementation
- Generate bit file

Pada proses generate bit file, akan dibuat file binary yang dapat diprogram ke FPGA.



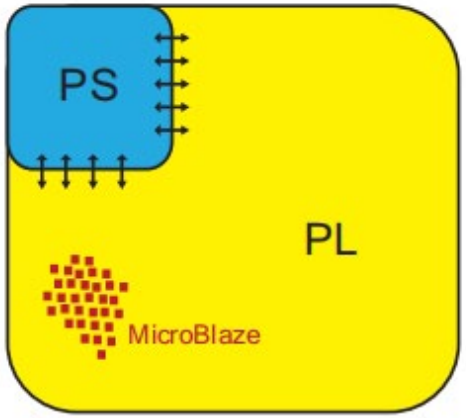
# What are the things that can be done with FPGAs

- Basic digital circuits
- Digital protocol implementation (SPI, I2C, UART, etc)
- Micro controller/processor (RISC-V, ARM, AVR, microblaze, etc)
- Wireless communication
- AI accelerator

# Soft Processor vs. Hard Processor

- Soft processor merupakan prosesor yang berupa kode Verilog yang diimplementasikan pada logic FPGA.
- Hard processor merupakan prosesor yang sudah difabrikasi menjadi silikon.


FPGA with soft processor
(e.g. MicroBlaze)




Zynq Architecture
(optional MicroBlaze)

# Soft Processor vs. Hard Processor

- Dibandingkan hard processor, soft processor memiliki kecepatan clock yang terbatas sekitar 100-200MHz.

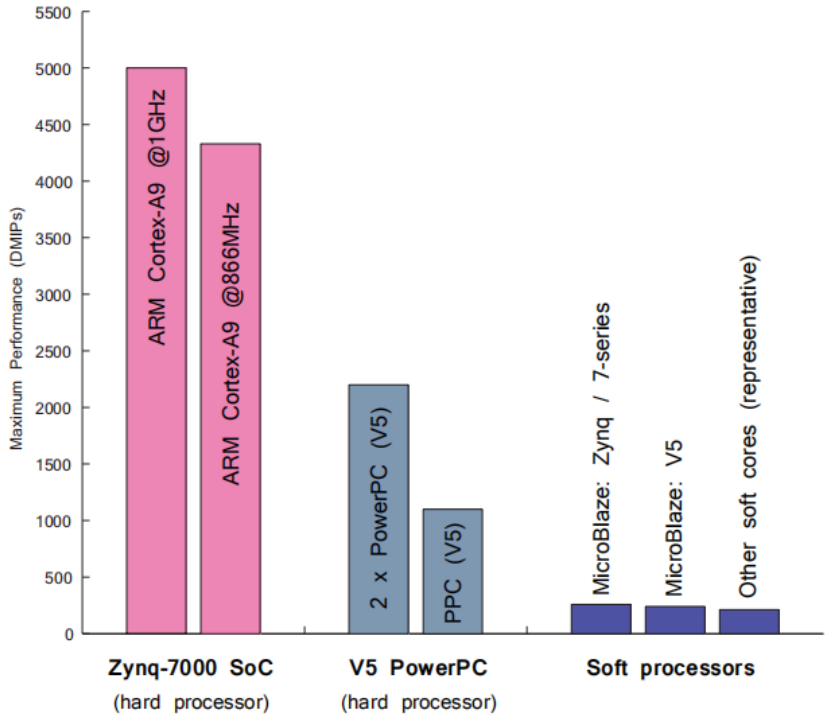
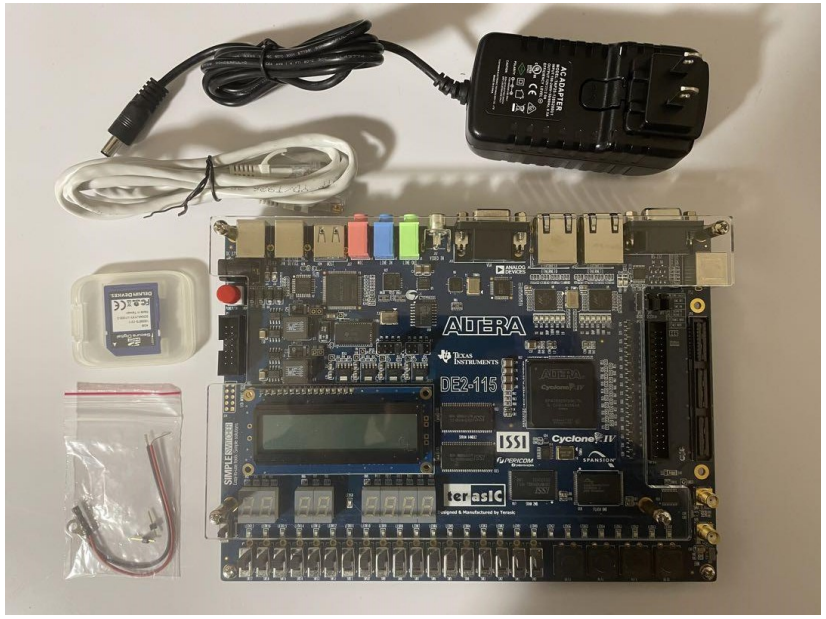


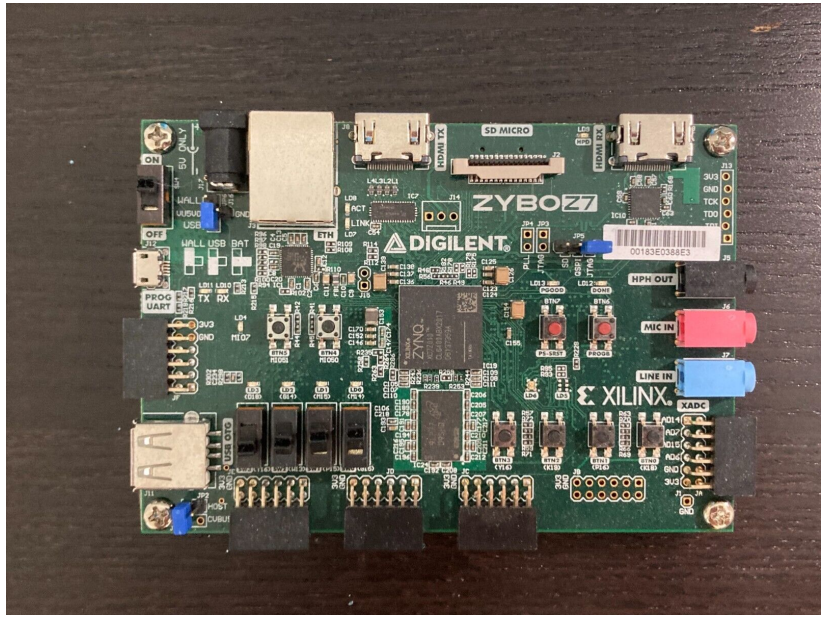

*Figure 4.3: Performance comparison of hard and soft processor options (indicative only - extrapolated and based on best case)*

# FPGA vs. SoC FPGA

- SoC FPGA biasanya memiliki hard processor.




Altera DE2, DE4, hanya berisi FPGA saja




Xilinx Zynq, berisi hard processor ARM

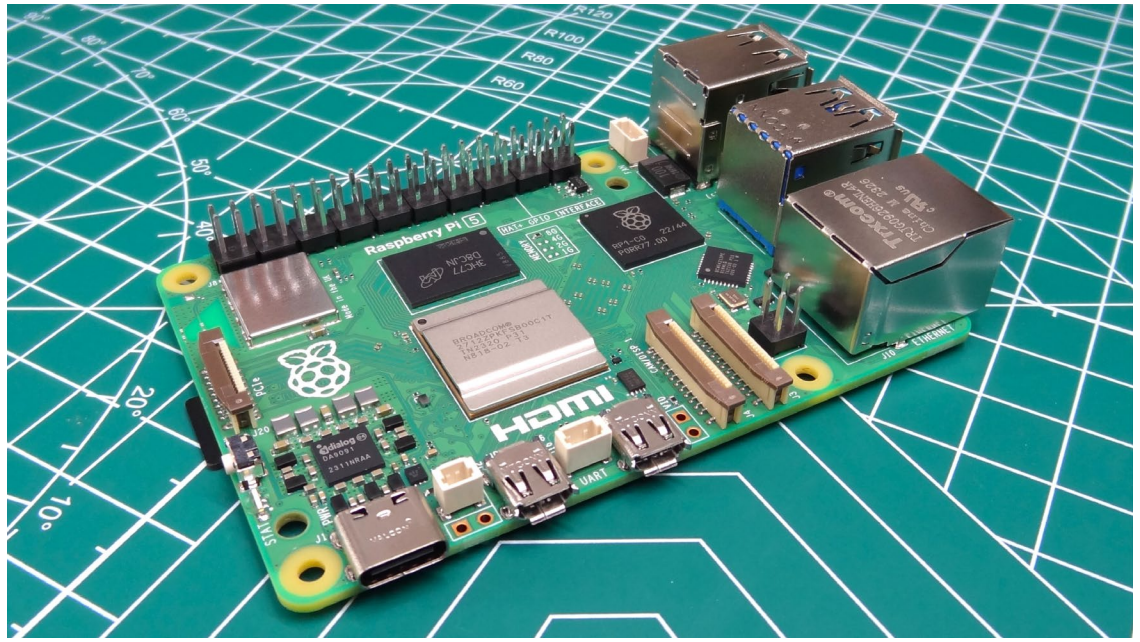
# FPGA vs. SoC FPGA

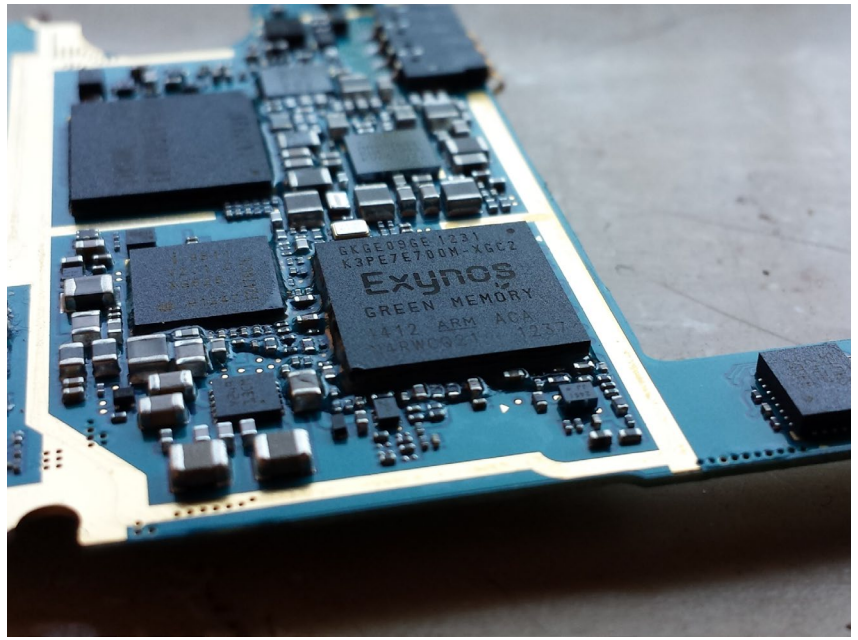
- Contoh FPGA: Altera DE2
  - Hanya memiliki resource FPGA
  - Tidak ada had processor
  - Jika membutuhkan processor bias menggunakan soft processor NIOS II.

- Contoh SoC FPGA: Digilent Xilinx Zybo
  - Memiliki resource FPGA
  - Terdapat hard processor ARM cortex A9

# SoC FPGA vs. SoC

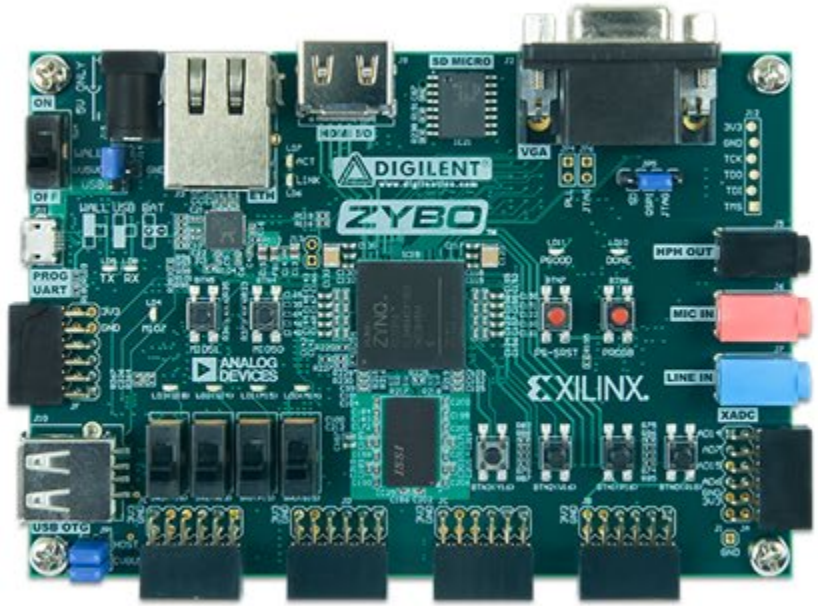
- SoC pada umumnya dipakai di single board computer seperti raspberry, sampai ke gadget seperti smartphone/tablet.
- Jenis SoC ini biasanya tidak memiliki programmagle logic (FPGA).



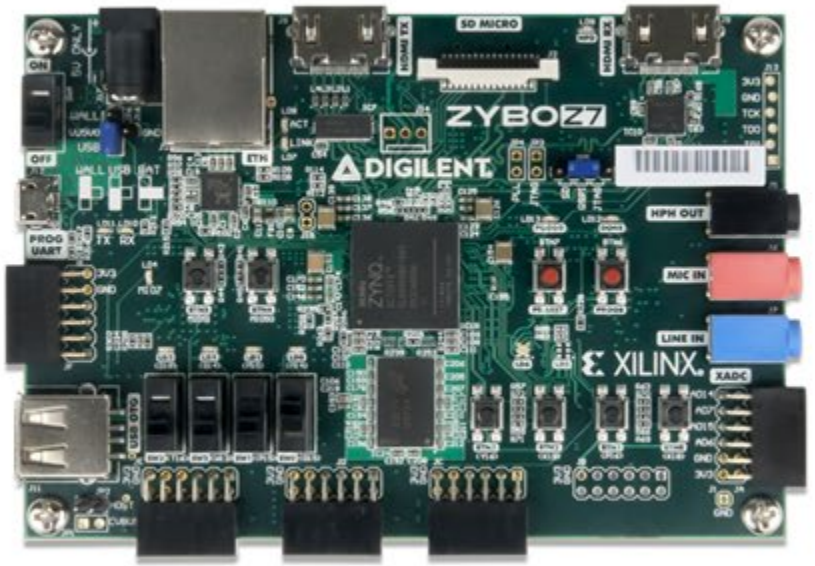



# Xilinx Zynq Board (ZyBo)

ZYBO

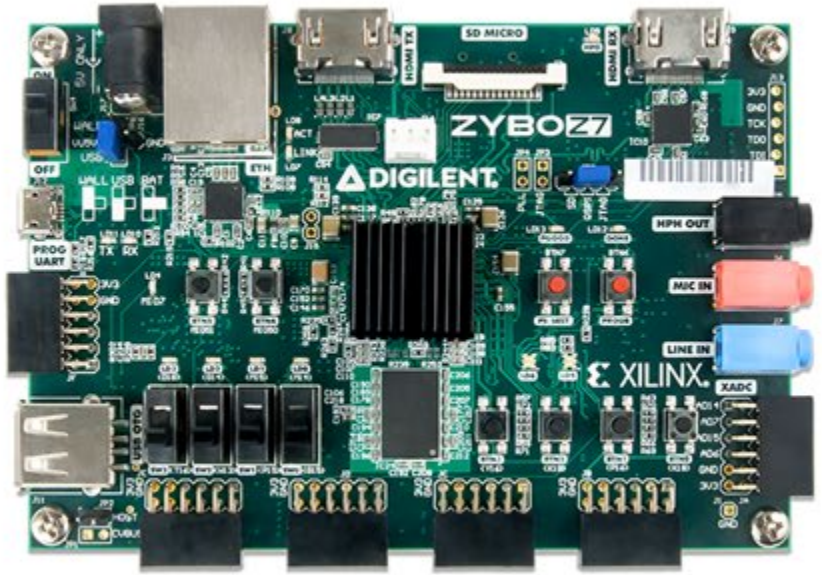



Zybo Z7-10




Part:
XC7Z010-1CLG400C

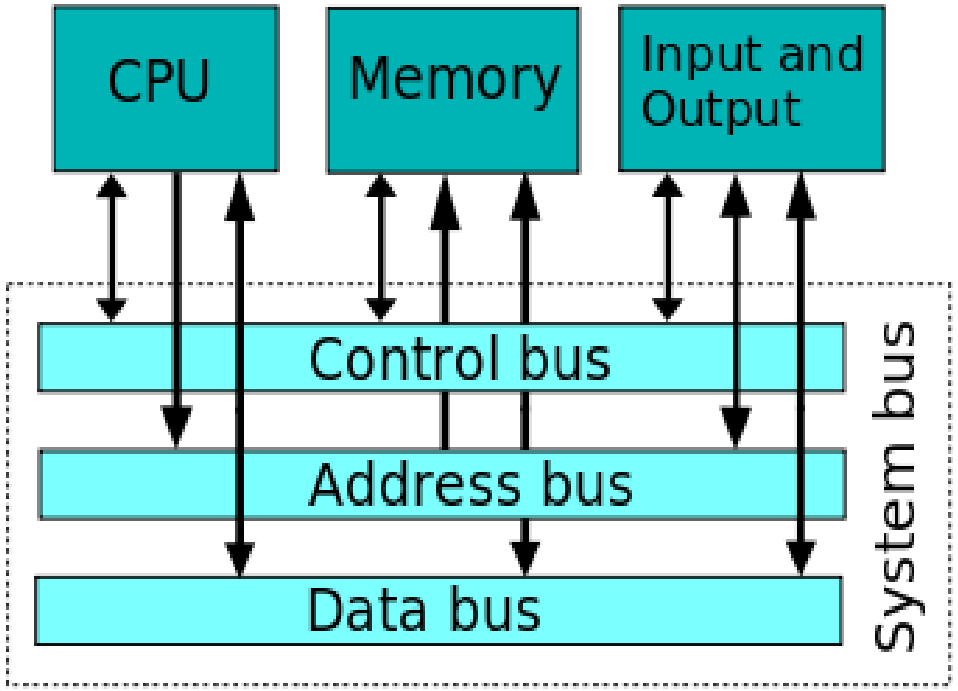
Zybo Z7-20




Part:
XC7Z020-1CLG400C

# Arsitektur Komputer

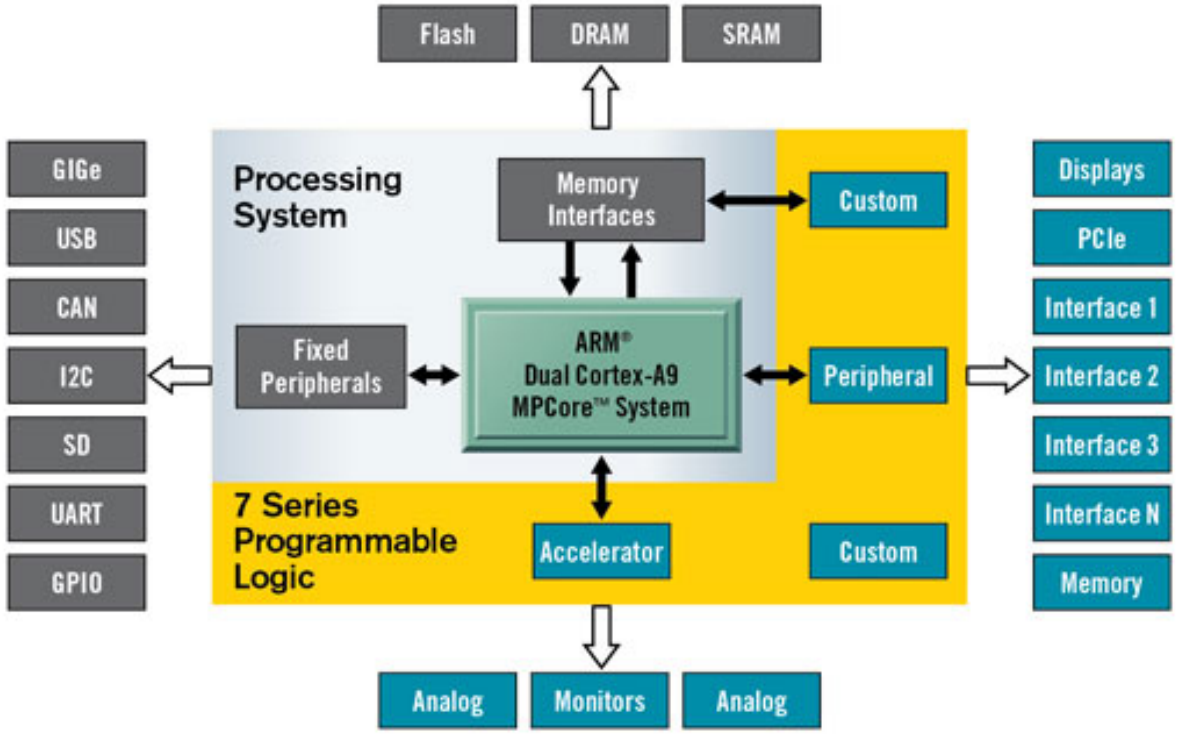
- Arsitektur computer pada umumnya terdiri dari: CPU (processor), memory, I/O, dan System Bus.



# Arsitektur Zynq

- Arsitektur zynq sama seperti arsitektur kompuer pada umumnya, tetapi ada tambahan FPGA.

**Processing System (PS)**
**CPU (Processor)**




**Programmable Logic (PL)**
**FPGA**

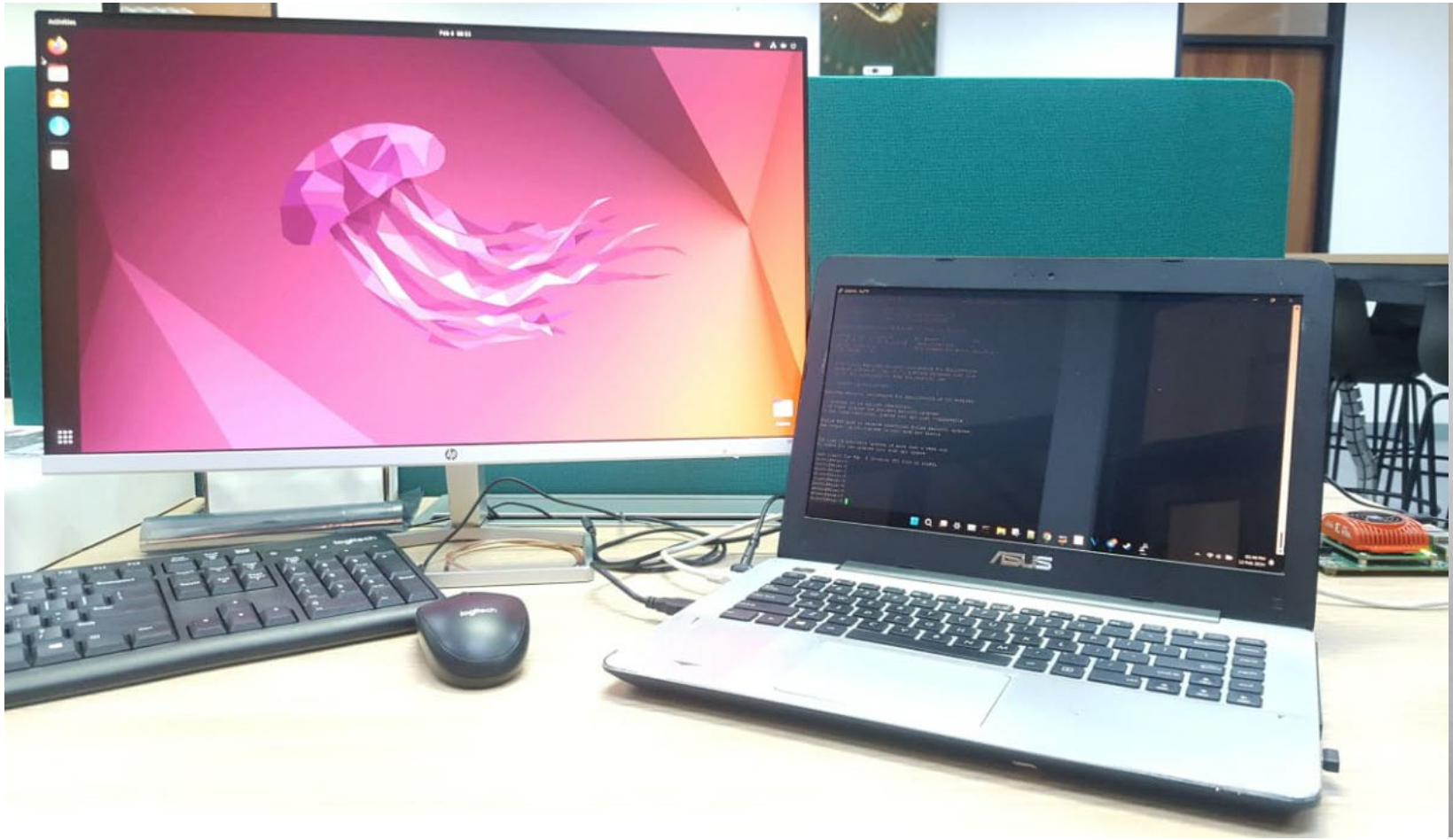
# System Bus

- AHB (Advanced High-performance Bus)
  - Ethernet, SDIO, USB
- APB (Advanced Peripheral Bus)
  - UART, SPI, I2C, GPIO
- AXI (Advanced Extensible Interface)  (PS to PL interface)
  - AXI memory mapped
    - AXI-Full
    - AXI-Lite
  - AXI stream

# PYNQ Framework

- PYNQ (Python productivity for Zynq) merupakan high level framework dan ekosistem hardware (berbasis Xilinx Zynq SoC FPGA).
- Analoginya seperti Arduino. Arduino terdiri dari framework, library dan ekosistem hardware-nya.
- PYNQ framework berjalan pada Operating System Linux untuk board tersebut.
- PYNQ melakukan abstraksi terhadap kompleksitas linux yang memungkinkan user membuat program aplikasi embedded Linux yang terintegrasi dengan FPGA secara lebih mudah.

# Linux OS pada Board Kria KV260

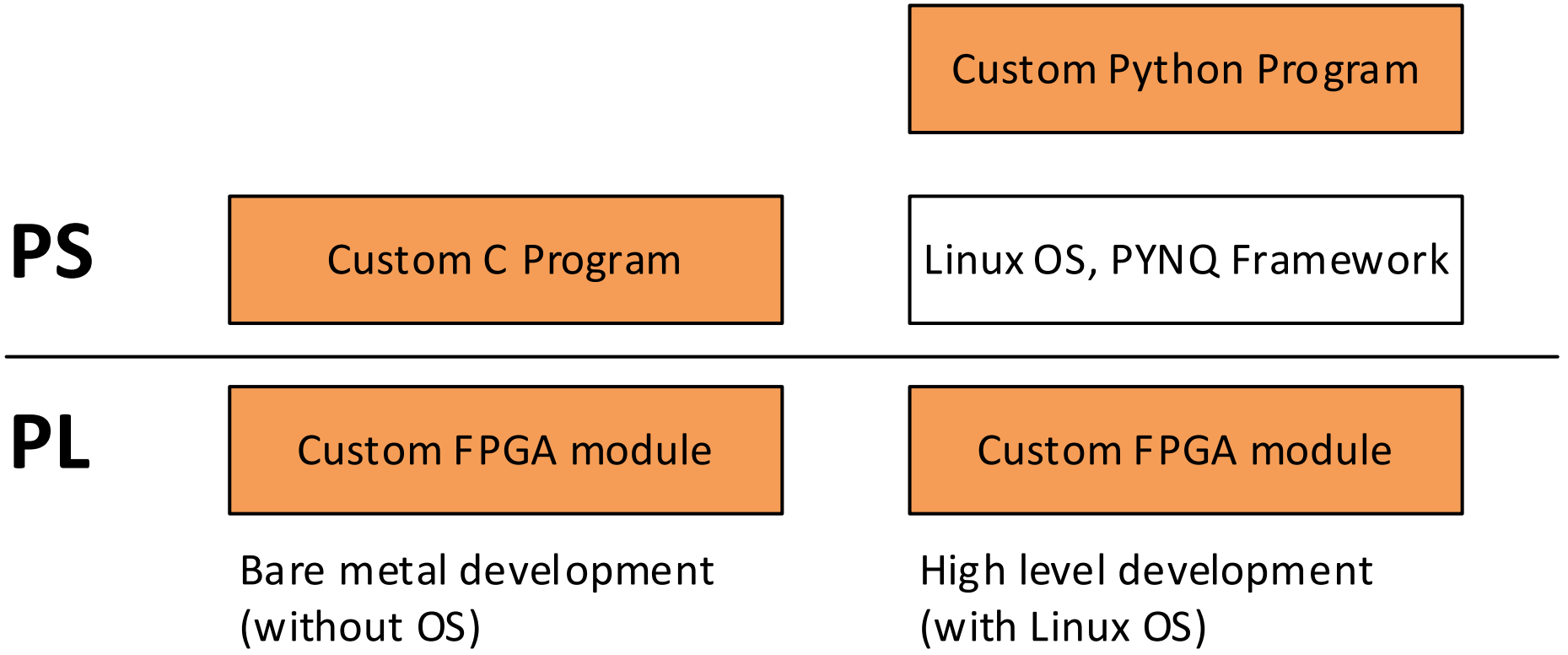


# Development Flow

- Development pada Zynq terdiri dari dua tahap, yaitu development hardware/FPGA/programmable logic (PL) dan software/firmware/Processing system (PS).
- Development hardware:
  - Membuat modul RTL dengan verilog atau VHDL yang nantinya akan diprogram ke FPGA/PL.
- Development software:
  - Membuat program (C, Python) yang nantinya akan diprogram ke PS.

# Development Flow

- Ada 2 tipe development pada Zynq.



# Software Tools for ZyBo (Bare Metal)

- Disarankan menggunakan Vivado **versi 2019.1**:
- Vivado versi 2019.2 - terbaru:
  - **Tidak support** karena tidak pakai SDK, tapi menggunakan Vitis.

# Xilinx Vivado

r/FPGA • 3 yr. ago
KyotoJayStation

one month to go until Vivado 2021.1, let's play bingo

Meme Friday

| V I V A D O | | | | |
|---|---|---|---|---|
| (things that still won't be fixed in the 2021.1 edition) | | | | |
| new synth bugs | can't mouseover struct/IF to see value | source control unfriendly projects | only one error shown at a time | waveform radix save broken |
| sim doesn't remember layout when resimulating | still no dark theme | sim errors only show in TCL console | new sim bugs | HW debug randomly doesn't work |
| file lock requires reboot | MIG config UI is still broken | IT'S BUGGY | vague or wrong error messages | SV still not fully supported |
| GUI doesn't remember theme | new impl bugs | still no proper auto completion | font settings broken | memory leaks |
| slow button response | previously working project no longer works | GUI is still slow | generated IP produces too many warnings | random crashes |

# Instalasi Vivado 1

- Offline installer:
  - Ukuran 21GB (versi 2019.1)
  - https://www.xilinx.com/member/forms/download/xef-vivado.html?filename=Xilinx_Vivado_SDK_2019.1_0524_1430.tar.gz

# Instalasi Vivado 2

- Untuk kuliah ini cukup pakai **Vivado versi free** License bisa registrasi ke website Xilinx (free).
- Jangan menginstal Vivado di **folder atau sub-folder-nya yang memiliki space**. Karena biasanya akan error ketika proses synthesis.
- Contoh:
  - C:\Program Files **(Error)**
  - C:\ProgramFiles **(OK)**
  - C:\ProgramFiles\Kuliah VLSI\Xilinx **(Error)**
  - C:\Xilinx **(OK)**
- Pastikan **checklist opsi SDK** saat menginstall Vivado.

# Membuat Project Vivado

- Jangan membuat project Vivado di folder seperti: folder **Users di C:, folder Program Files, atau folder system yang sifatnya protected atau read only** karena biasanya akan error ketika proses synthesis.
- Jangan membuat project Vivado di folder atau sub-folder-nya yang memiliki space.
- Contoh:
  - D:\Kuliah VLSI\Vivado **(Error)**
  - D:\Kuliah_VLSI\Vivado **(OK)**
  - D:\Kuliah_VLSI **(OK)**

Link Materi

bit.ly/3Ya3Jg7