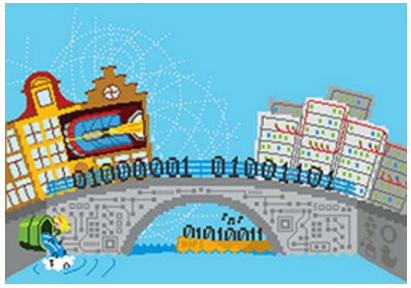
Introduction to SoC FPGA

And HardWorking Cores in Electronic Systems

Manoel Barros Marin (CERN BE-BI-QP)

(on behalf of the ISOTDAQ SoC FPGA Team)

International School of Trigger & Data Acquisition
(ISOTDAQ) 2017 @ Nikhef (Amsterdam)



Introduction to SoC FPGA

And HardWorking Cores in Electronic Systems

Outline:

- Introduction
- SoC FPGA
- Summary

Introduction to SoC FPGA

And HardWorking Cores in Electronic Systems

Outline:

- Introduction
- SoC FPGA
- Summary

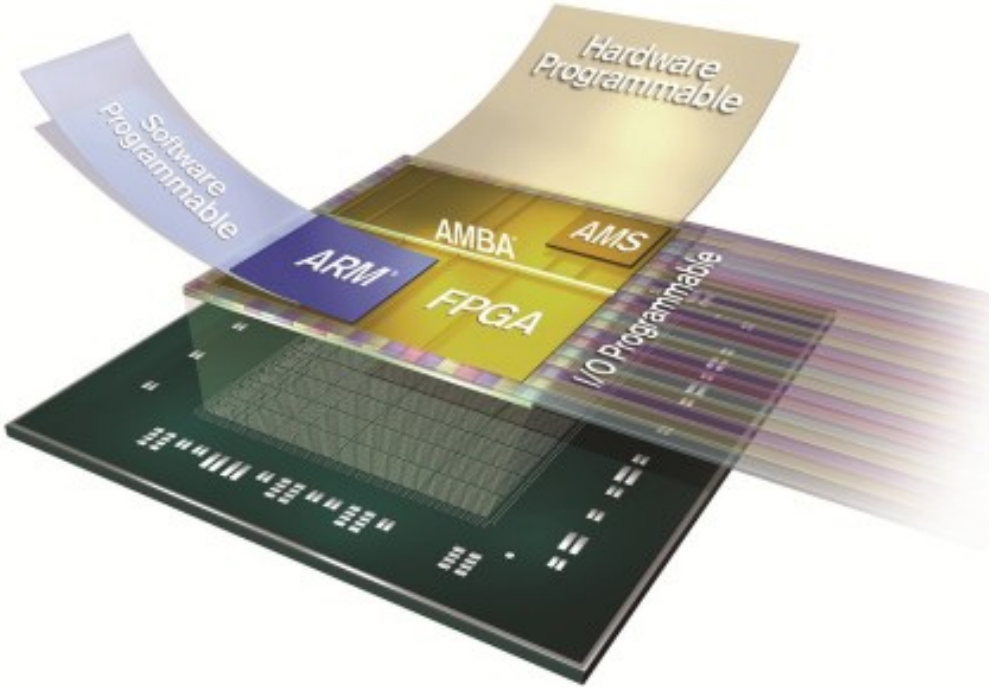
Introduction (1 of 13)

What is an SoC FPGA?

Introduction (1 of 13)

What is an SoC FPGA?

- Device featuring both System On Chip (SoC) and FPGA architectures into a single Integrated Circuit (IC)
- Also includes a rich set of peripherals (e.g. Multi-Gigabit Transceivers (MGT), DSP cores)
- Becoming more and more popular as HardWorking Cores (HWC) in Electronic Systems



Introduction (2 of 13)

But...

...what does this guy mean with HardWorking Cores in Electronic Systems???

Introduction (3 of 13)

HardWorking Core in Electronic Systems

- Device in charge of the control tasks and/or data processing of an electronic system

Introduction (3 of 13)

HardWorking Core in Electronic Systems

- **Device in charge of the control tasks and/or data processing of an electronic system**

Typical HardWorking Cores in Electronic Systems

- **Commercial-Off-The-Shelf (COTS):**
  - Fixed Architecture (Software design work flow):
    - MicroProcessor (μP)
    - MicroController Unit (MCU)
    - System On Chip (SoC)
    - Digital Signal Processor (DSP)
    - Graphics Processing Unit (GPU)
  - Configurable Logic (Logic design work flow):
    - Complex Programmable Logic Device (CPLD)
    - Field-Programmable Gate Array (FPGA)
- **Custom design (Microelectronics design work flow):**
  - Application Specific Integrated Circuit (ASIC)

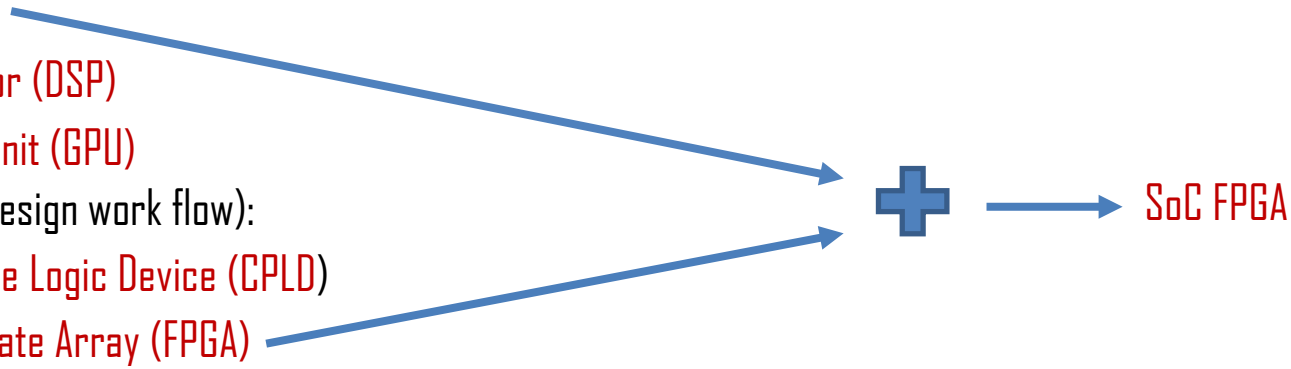
Introduction (3 of 13)

HardWorking Core in Electronic Systems

- **Device in charge of the control tasks and/or data processing of an electronic system**

Typical HardWorking Cores in Electronic Systems

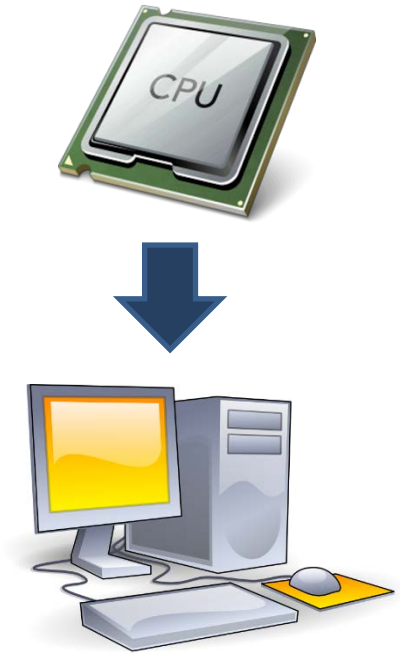
- **Commercial-Off-The-Shelf (COTS):**
  - Fixed Architecture (Software design work flow):
    - MicroProcessor (µP)
    - MicroController Unit (MCU)
    - System On Chip (SoC)
    - Digital Signal Processor (DSP)
    - Graphics Processing Unit (GPU)
  - Configurable Logic (Logic design work flow):
    - Complex Programmable Logic Device (CPLD)
    - Field-Programmable Gate Array (FPGA)
- **Custom design (Microelectronics design work flow):**
  - Application Specific Integrated Circuit (ASIC)



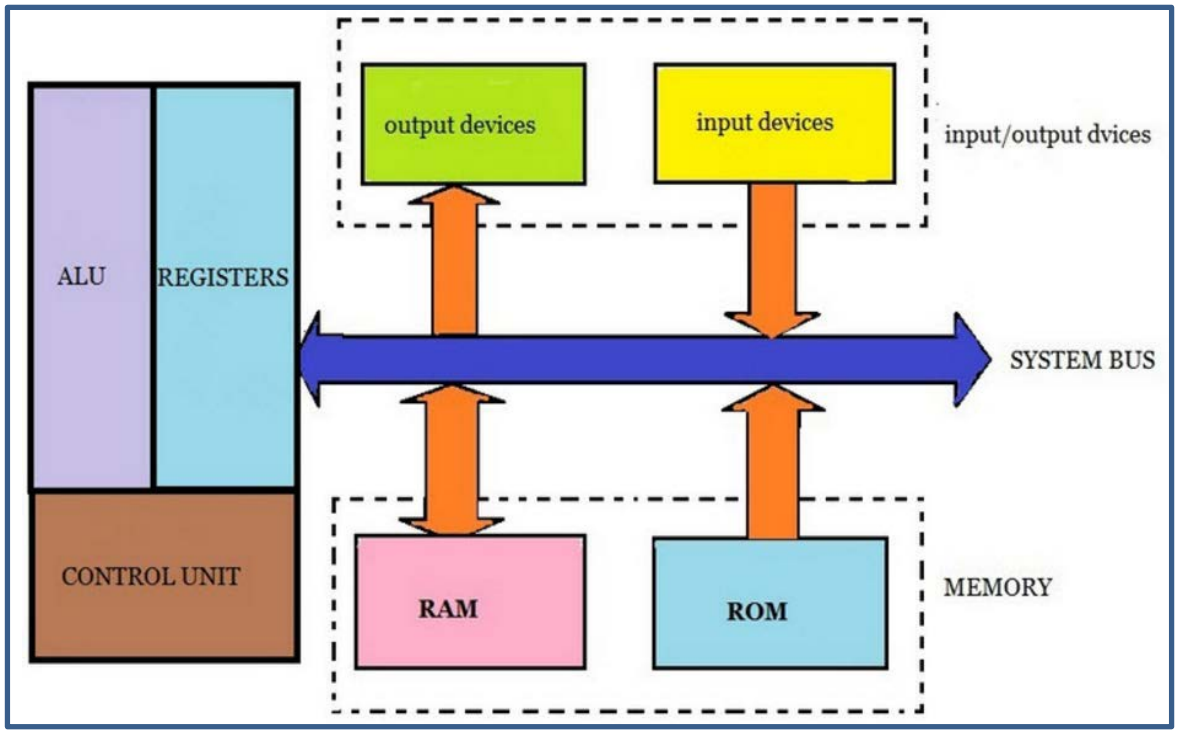
Introduction (4 of 13)

COTS: fixed architecture

- **Microprocessor (μP):**
  - Computer Central Processing Unit (CPU) on a single Integrated Circuit (IC) (e.g. Intel i7, AMD RYZEN)
  - Represents the core of a computer
  - HIGH computing power for general purpose applications
  - Single/Multi-thread processing



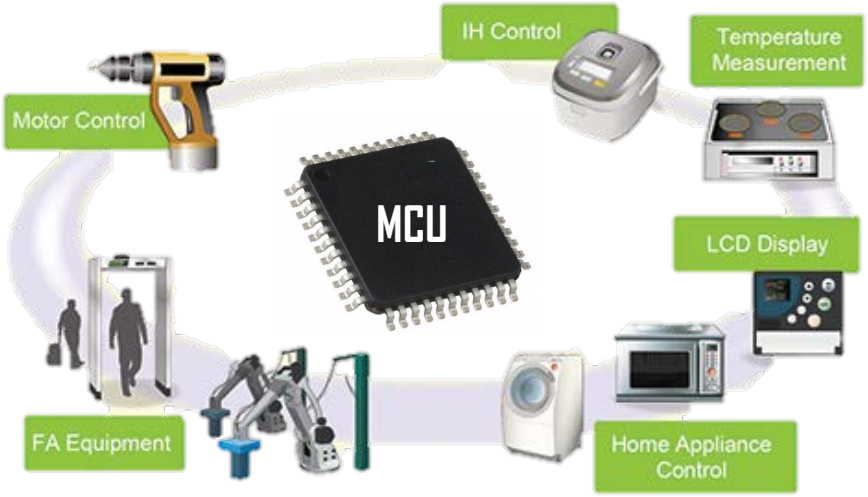
Example Diagram of Microprocessor



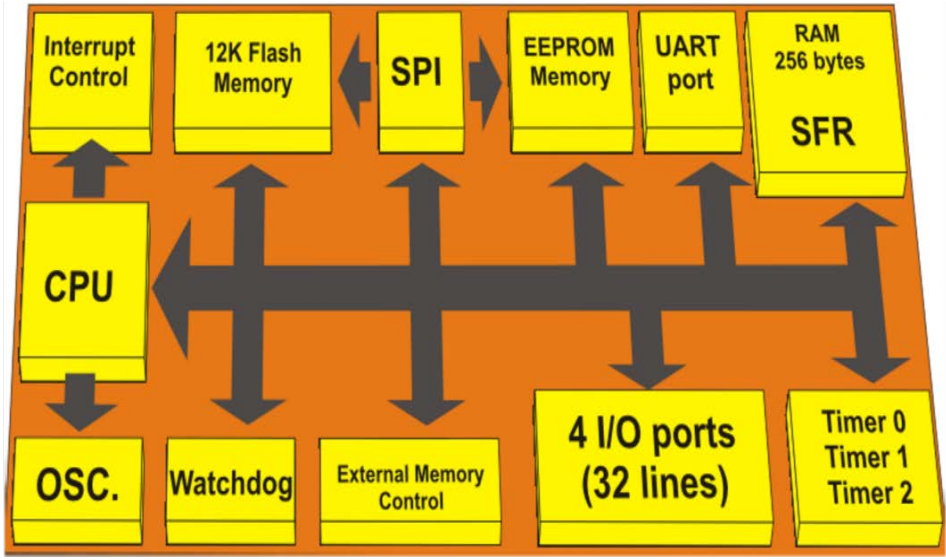
Introduction (5 of 13)

COTS: fixed architecture

- **Microcontroller (MCU):**
  - Small computer on a single integrated circuit (e.g. Atmel AT Mega, Microchip PIC)
  - Multiple peripherals (e.g. ADC, timer)
  - LOW computing power for general purpose applications (compared to µP)
  - May be used in Real Time applications (capable of low latency & Real Time Operating Systems (RTOS))
  - Very common in Embedded Systems (ES) (e.g. Arduino, modem, TV)
  - Single-thread processing



Example Diagram of Microcontroller

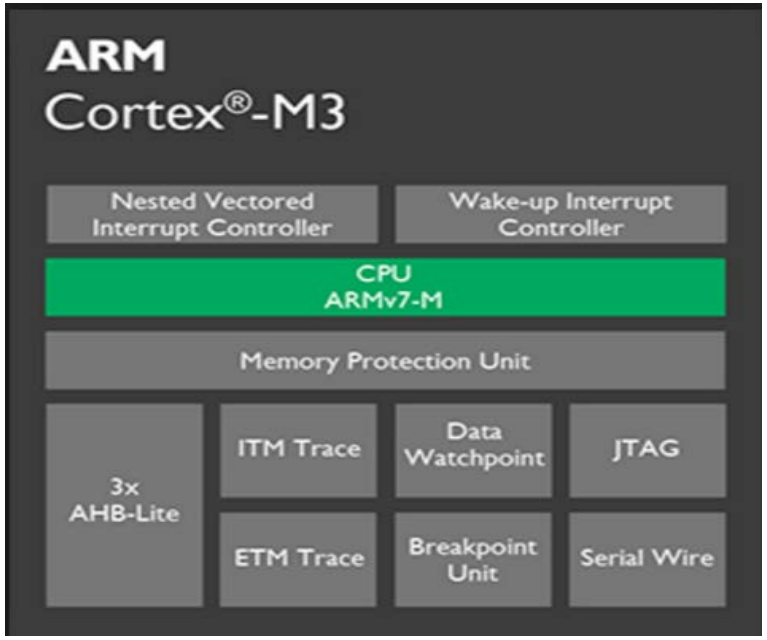


Introduction (6 of 13)

COTS: fixed architecture

- **System On Chip (SoC):**
  - Like a microcontroller with steroids ☺
  - ARM architecture has become standard
  - Multiple peripherals (e.g. GPU, DSP)
  - HIGH computing power for general purpose applications (capable to run Operating Systems (e.g. Linux))
  - May be used in Real Time applications (capable of low latency & Real Time Operating Systems (RTOS))
  - Very common in Embedded Systems (ES) (e.g. smartphones, tablet)
  - Single/Multi-thread processing

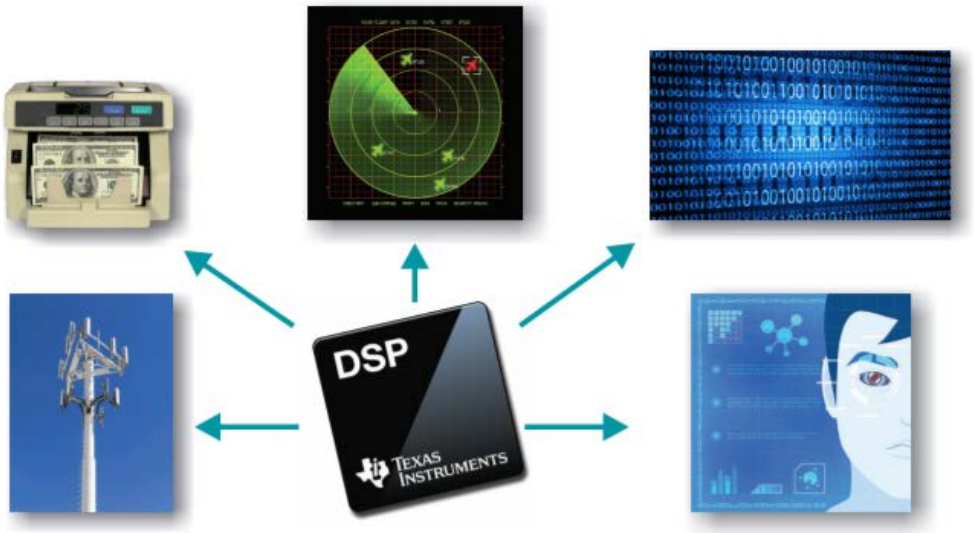
Example Diagram of SoC



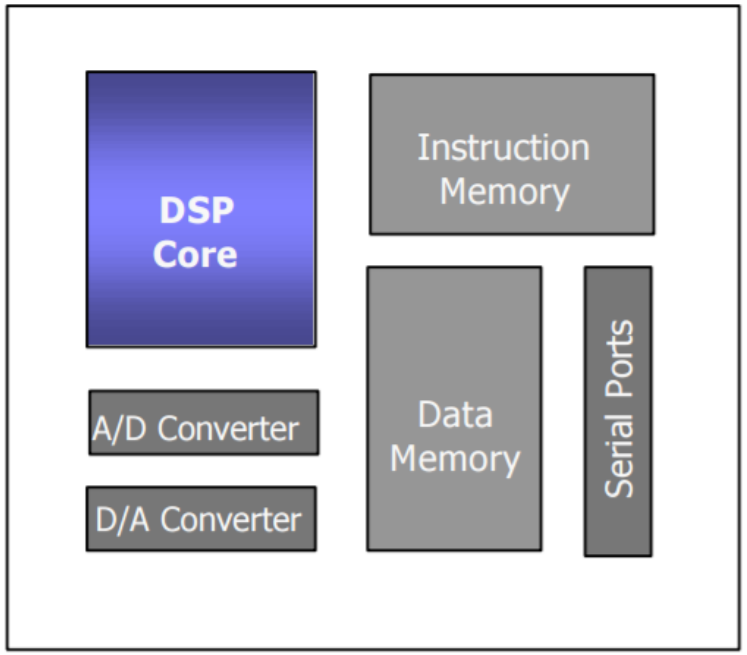
Introduction (7 of 13)

COTS: fixed architecture

- **Digital Signal Processor (DSP):**
  - Specialized microprocessor with its architecture optimized for digital signal processing (e.g. FFT, filters)
  - Dedicated blocks for mathematic operations (e.g. multiplier-accumulators (MAC), shift registers)
  - May have multiple peripherals (e.g. ADC, timer)
  - Separated Data/Instruction buses (Harvard architecture)
  - HIGH computing power for specific applications
  - May be used in Real Time applications (capable of low latency)
  - Single/Multi-thread processing



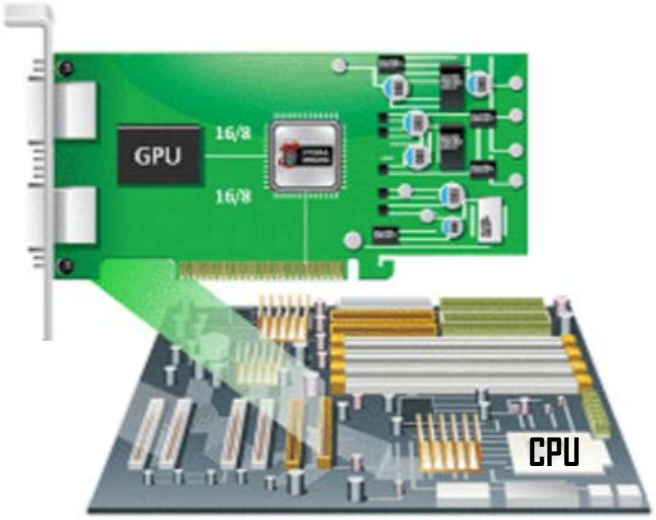
Example Diagram of DSP



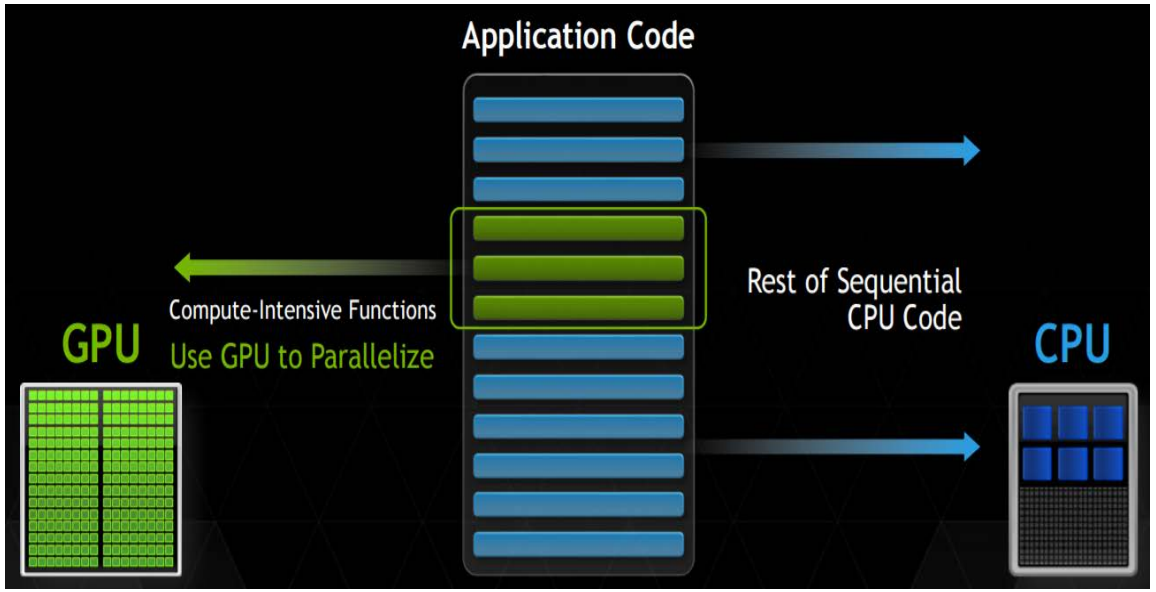
Introduction (8 of 13)

COTS: fixed architecture

- **Graphic Processor Unit (GPU):**
  - A GPU is a multi-core integrated circuit highly tuned for graphics generation
  - Used as co-processor in computers
  - Very common for graphics generation but becoming very popular for other applications (e.g. Medical, Physics)
  - VERY HIGH computing power for specific applications
  - Multi-thread processing
  - Complex parallel programming



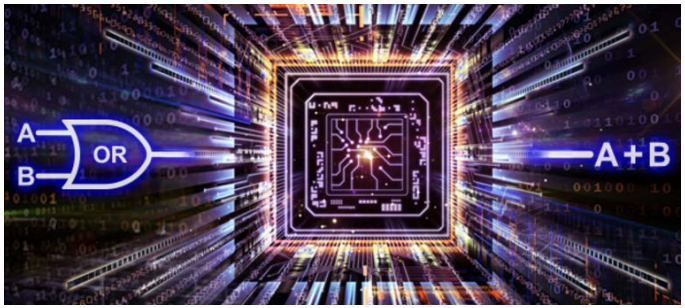
Example Diagram of Combined CPU + GPU Processing



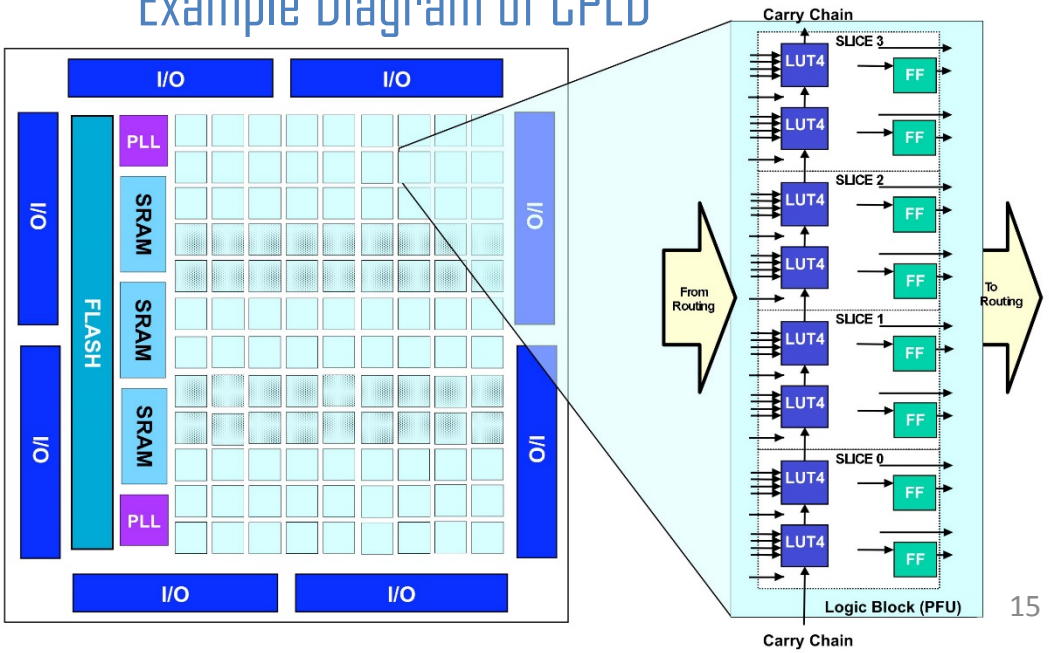
Introduction (9 of 13)

COTS: configurable logic

- **Complex Programmable Logic Device (CPLD):**
  - Programmable Logic Device (PLD)
  - Features logic elements (e.g. AND gate), registers, I/O blocks (it may feature PLLs & memories)
  - Non-volatile (Flash-based) configuration memory
  - Very common for interfacing (glue logic) and/or implementing SIMPLE processing cores
  - Very good option for Real Time applications (capable of low, fixed and deterministic latency)
  - LOW computing power for specific applications
  - Parallel processing



Example Diagram of CPLD

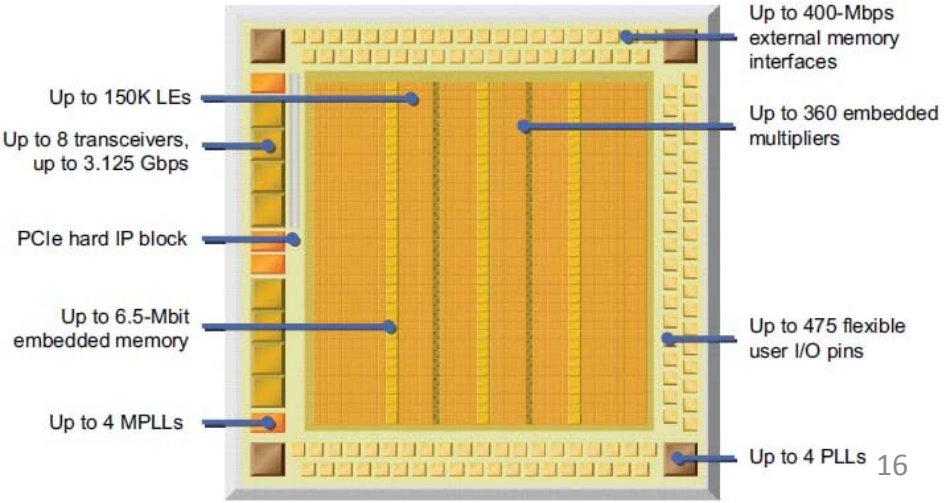


Introduction (10 of 13)

COTS: configurable logic

- **Field-Programmable Gate Array (FPGA):**
  - Like a CPLD with steroids ☺
  - Features logic elements (e.g. AND gate), registers, I/O blocks, PLLs, memories
    - But also features other dedicated hard-blocks (e.g. Multi-Gigabit Transceivers (MGT), DSP blocks)
    - Microprocessors may be implemented using logic elements and internal memory
  - Volatile (SRAM-based) or Non-volatile (Flash-based) configuration memory
  - Very common for interfacing (glue logic) and/or implementing COMPLEX processing cores
  - Very good option for Real Time applications (capable of low, fixed and deterministic latency)
  - VERY HIGH computing power for general purpose or specific applications
  - Parallel processing

Example Diagram of FPGA

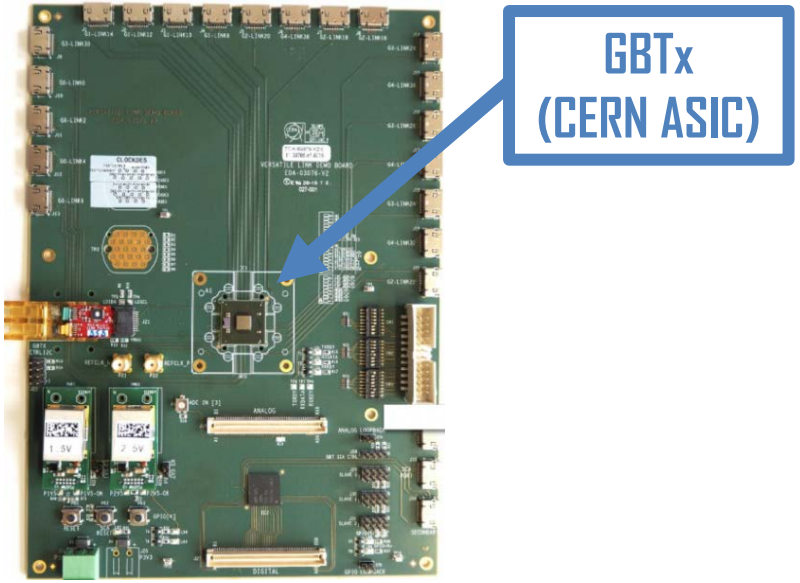


Introduction (11 of 13)

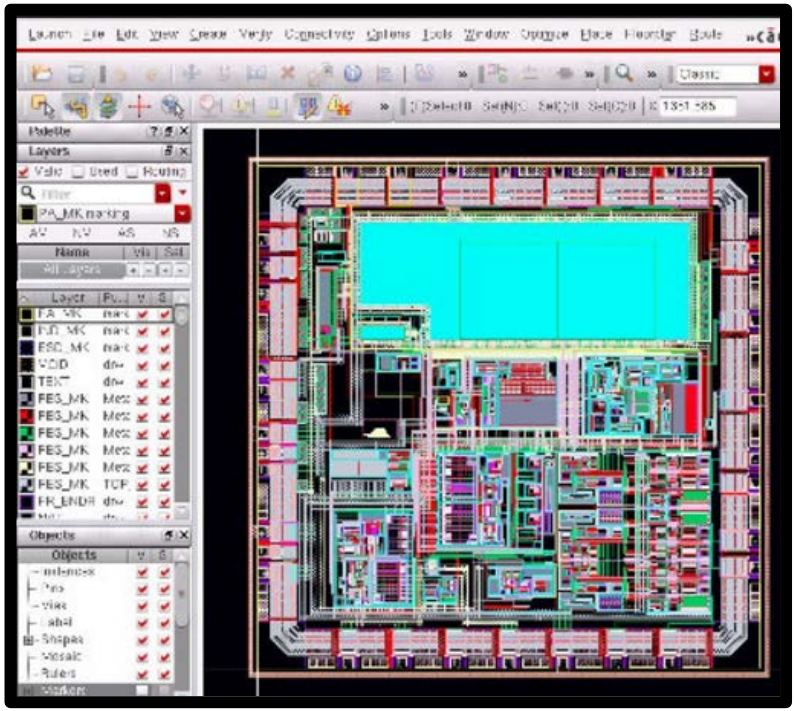
Custom design

- **Application Specific Integrated Circuit (ASIC):**
  - Integrated circuit customised for a particular use, rather than intended for begin general purpose
  - Used when COTS components do not suit the application
  - May feature any type of peripherals (digital and/or analogue)
  - Custom level of computing power
  - Single/Multi-threaded and/or Parallel processing

Example of ASIC



Example of ASIC layout



Introduction (12 of 13)

Comparison of typical of HardWorking Cores in Electronic Systems (1 of 2)

Device	Functionality	Main Application	Architecture	Processing	Computing Power	Design Work Flow	Cost Range	Real Time	Form Factor
uP	General Purpose	Computers	Fixed	Single/Multi-thread	High	Software	Mid-High	No	Small
MCU	General Purpose	Low-Mid Range ES	Fixed	Single-thread	Low	Software	Low	Yes	Very Small
SoC	General Purpose	Mid-High Range ES	Fixed	Single/Multi-thread	High	Software	Low-Mid	Yes	Very Small
DSP	Application Specific	Digital Signal Processing	Fixed	Single/Multi-thread	High	Software	Low-Mid	Yes	Small
GPU	Application Specific	Intensive Processing	Fixed	Multi-thread	High	Software	Mid-High	No	Small

Introduction (13 of 13)

Comparison of typical of HardWorking Cores in Electronic Systems (2 of 2)

Device	Functionality	Main Application	Architecture	Processing	Computing Power	Design Work Flow	Cost Range	Real Time	Form Factor
CPLD	General Purpose / Application Specific	Glue Logic, Basic Processing	Configurable	Parallel	Low	Logic Design	Low	Yes	Small
FPGA	General Purpose / Application Specific	Glue Logic, Control, Intensive Processing	Configurable	Parallel	High	Logic Design	Mid-High	Yes	Small
SoC FPGA	General Purpose / Application Specific	Glue Logic, Mid-High Range ES	Fixed & Configurable	Single/Multi-thread & Parallel	High	Software & Logic Design	Mid-High	Yes	Very Small
ASIC	Application Specific	Multiple Applications	Custom	Custom	Custom	ASIC	Based On Order Size	Yes	Custom

Introduction to SoC FPGA

And HardWorking Cores in Electronic Systems

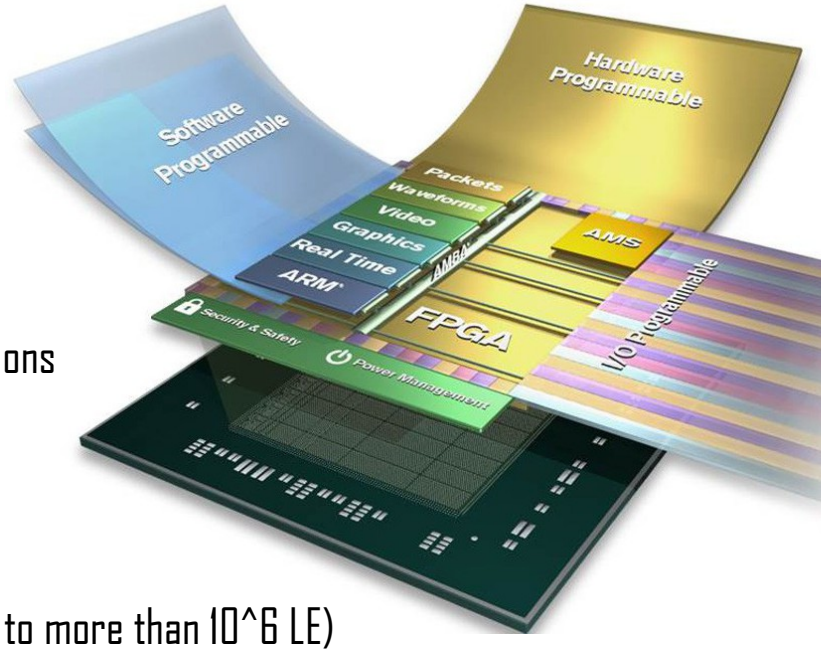
Outline:

- Introduction
- **SoC FPGA**
- Summary

SoC FPGA (1 of 6)

Main Features

- **SoC FPGA integrates both System On Chip (SoC) and FPGA architectures into a single device:**
  - Advantages with respect to discrete solution (SoC + FPGA on the same PCB):
    - Higher integration
    - Lower power consumption
    - Higher bandwidth between processor and FPGA
  - SoC part:
    - ARM microprocessor architecture has become standard
    - Rages from single to multi-core (e.g. 4 cores) configurations
    - Several memory levels (e.g. cache L1 & L2, DDR)
    - Multiple peripherals (e.g. SPI master, USB)
  - FPGA part:
    - Ranges from low to high count of Logic Elements (LE) (up to more than 10^6 LE)
    - Multiple peripherals (e.g. Multi-Gigabit Transceivers, DSP blocks)



SoC FPGA (2 of 6)

When do we should use an SoC FPGA?

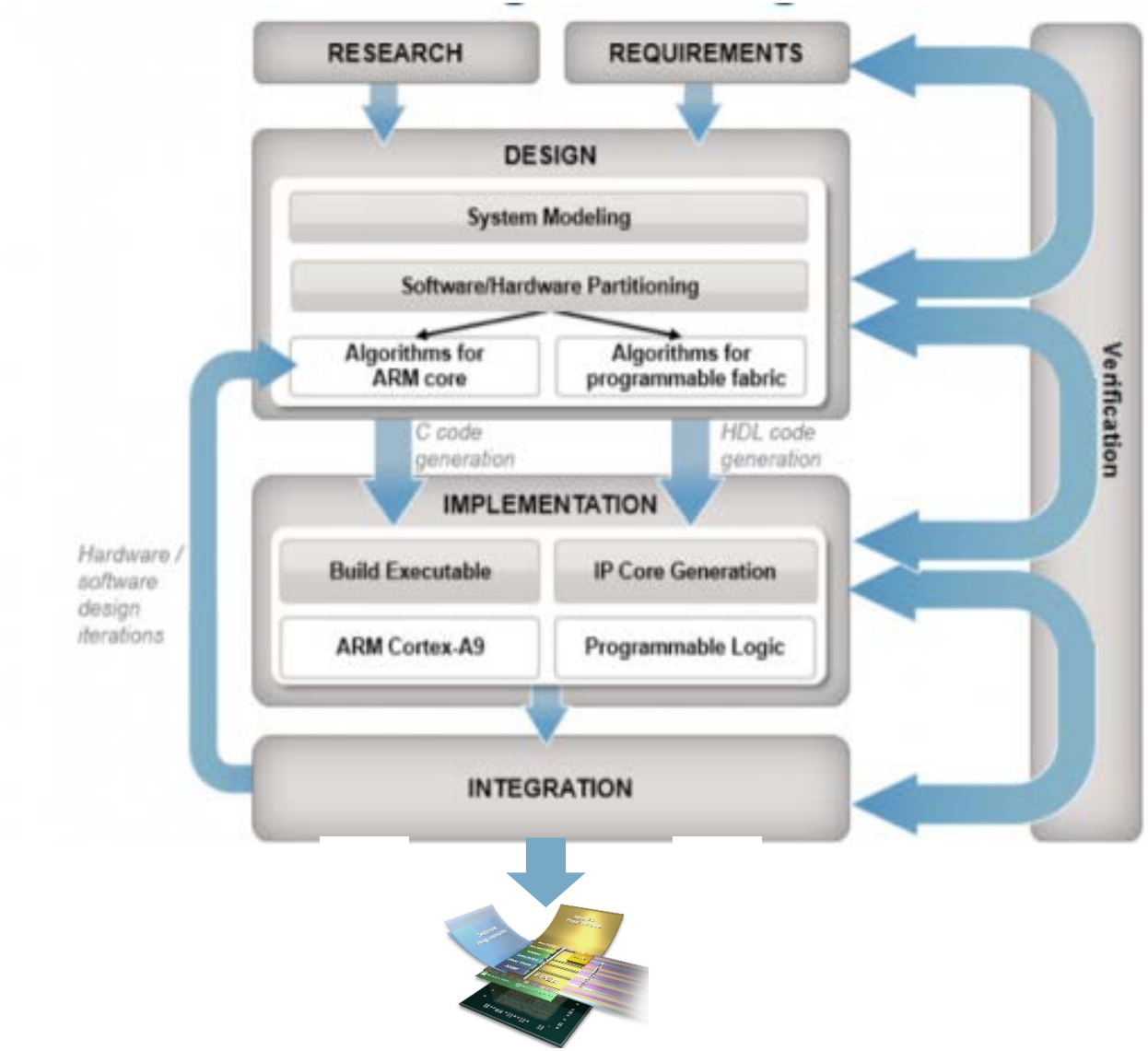
SoC FPGA (2 of 6)

When do we should use an SoC FPGA?

- **SoC FPGA may be a good candidate when:**
  - **New System:**
    - Requiring both processor-based AND programmable logic-based in you system
  - **System Update :**
    - The existing design use an FPGA and a separate microprocessor
    - The existing design use a proprietary ASIC that includes a microprocessor
    - The existing design has a microprocessor but the application would benefit from a tailored peripheral set

SoC FPGA (3 of 6)

Work flow: gateware/software co-design



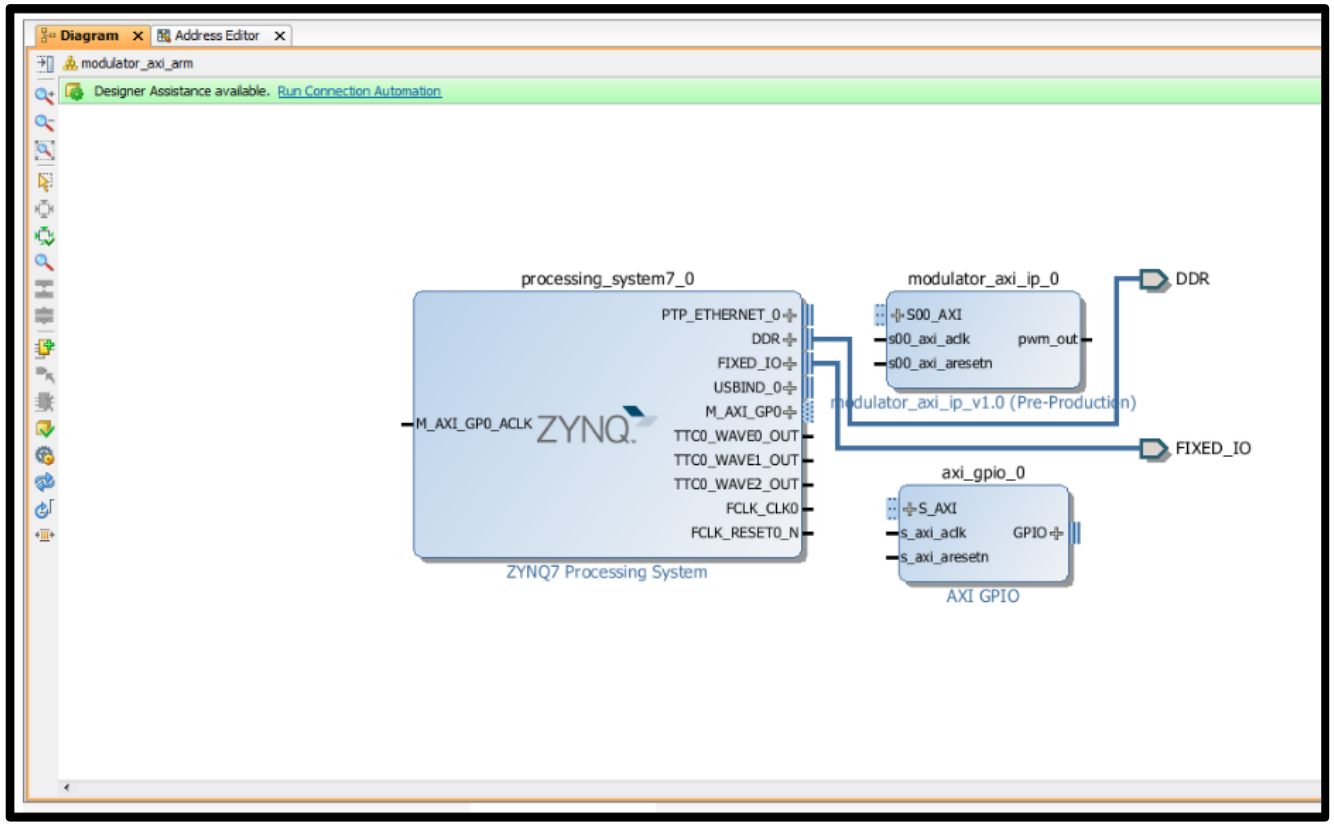
SoC FPGA (4 of 6)

Gateware

SoC FPGA (4 of 6)

Example of Schematic

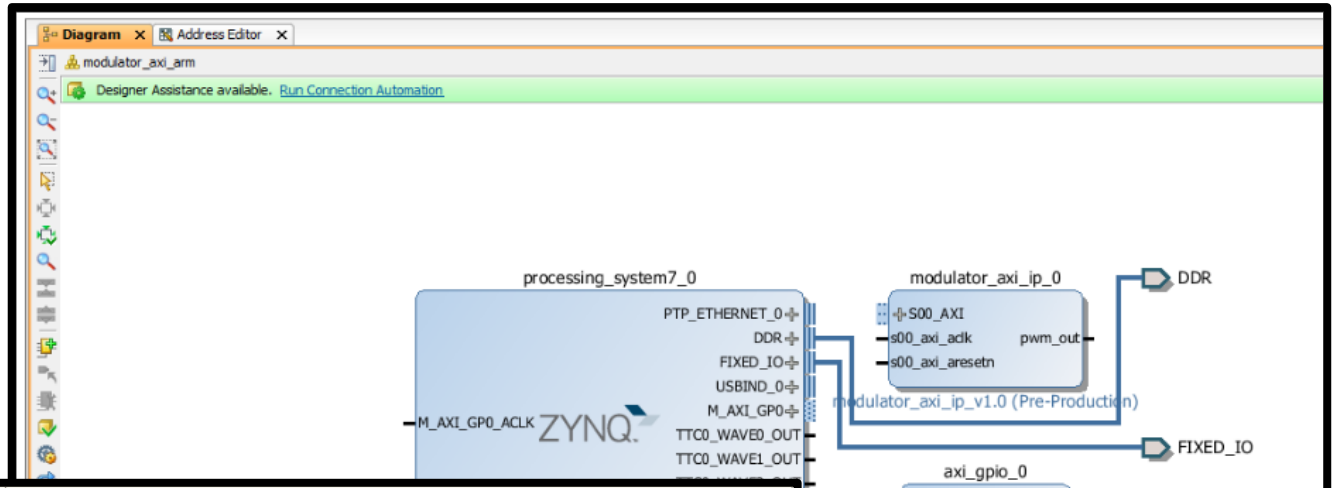
Gateware



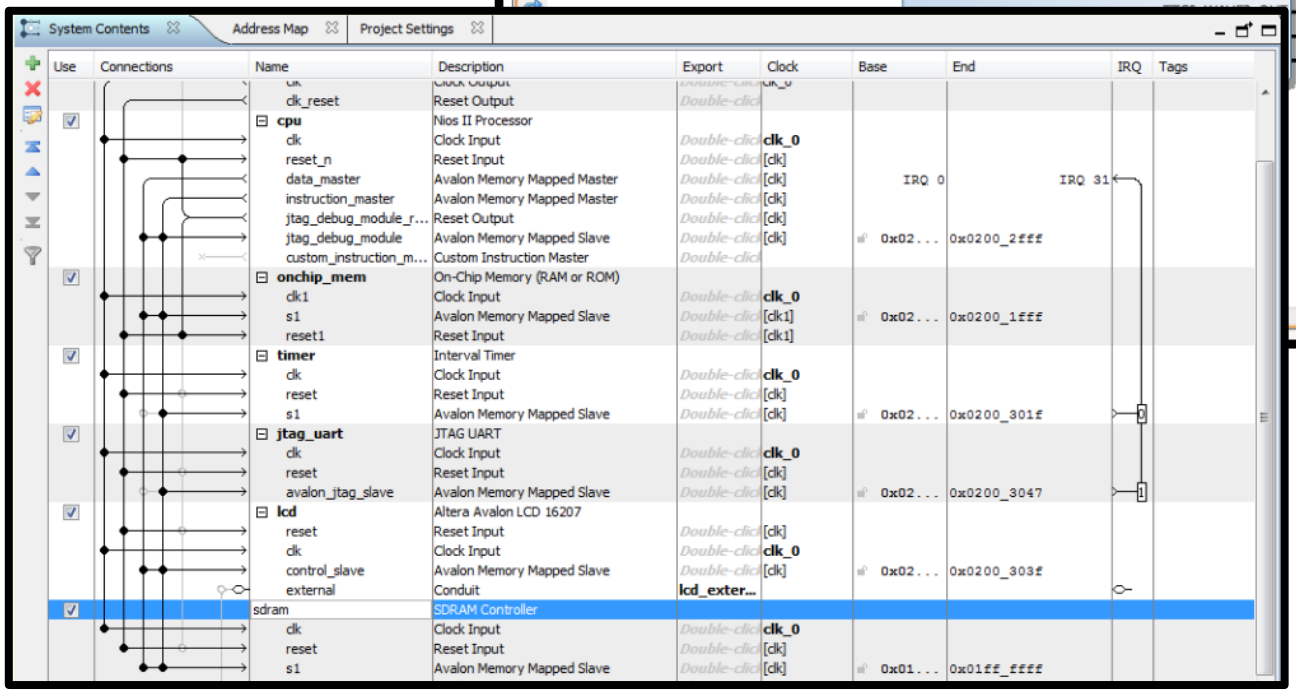
SoC FPGA (4 of 6)

Gateware

Example of Schematic



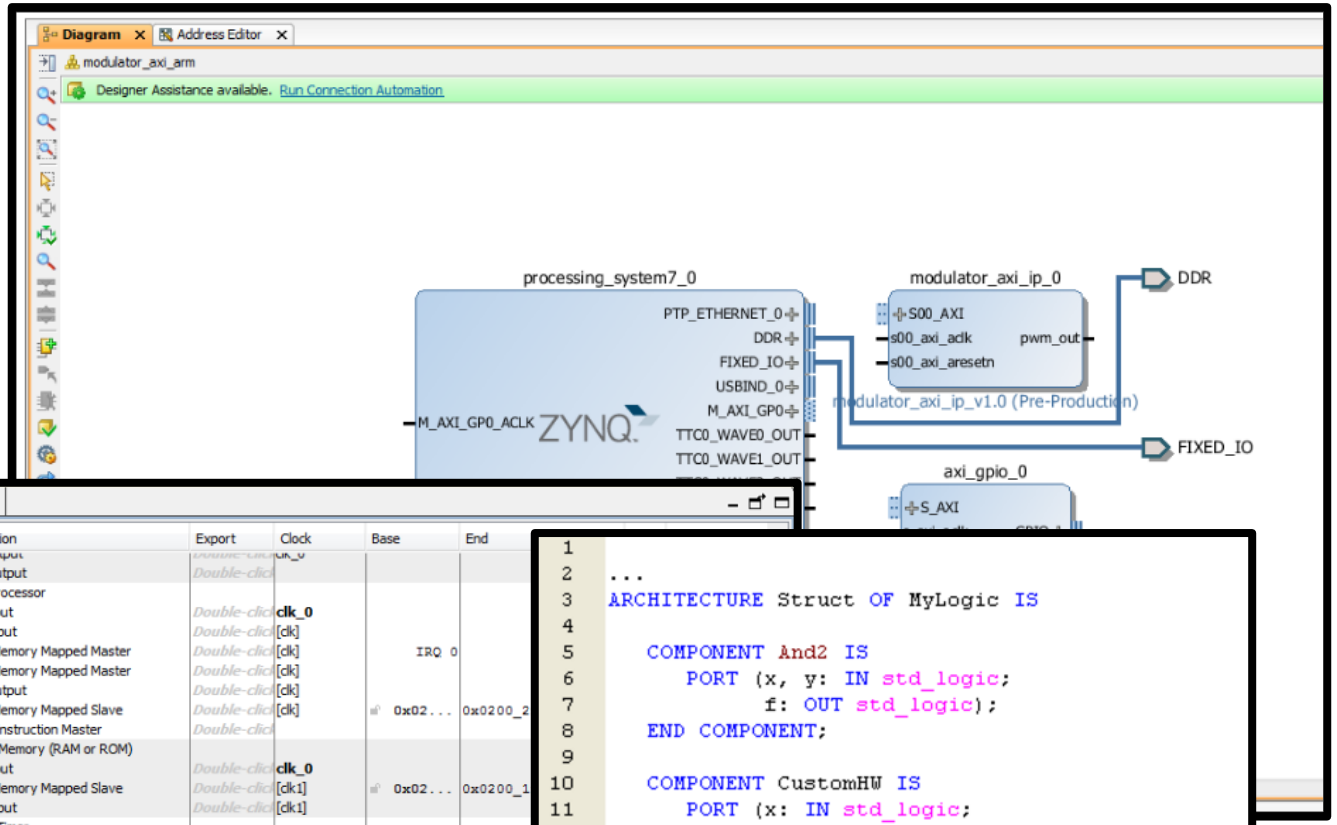
Example of Connectivity



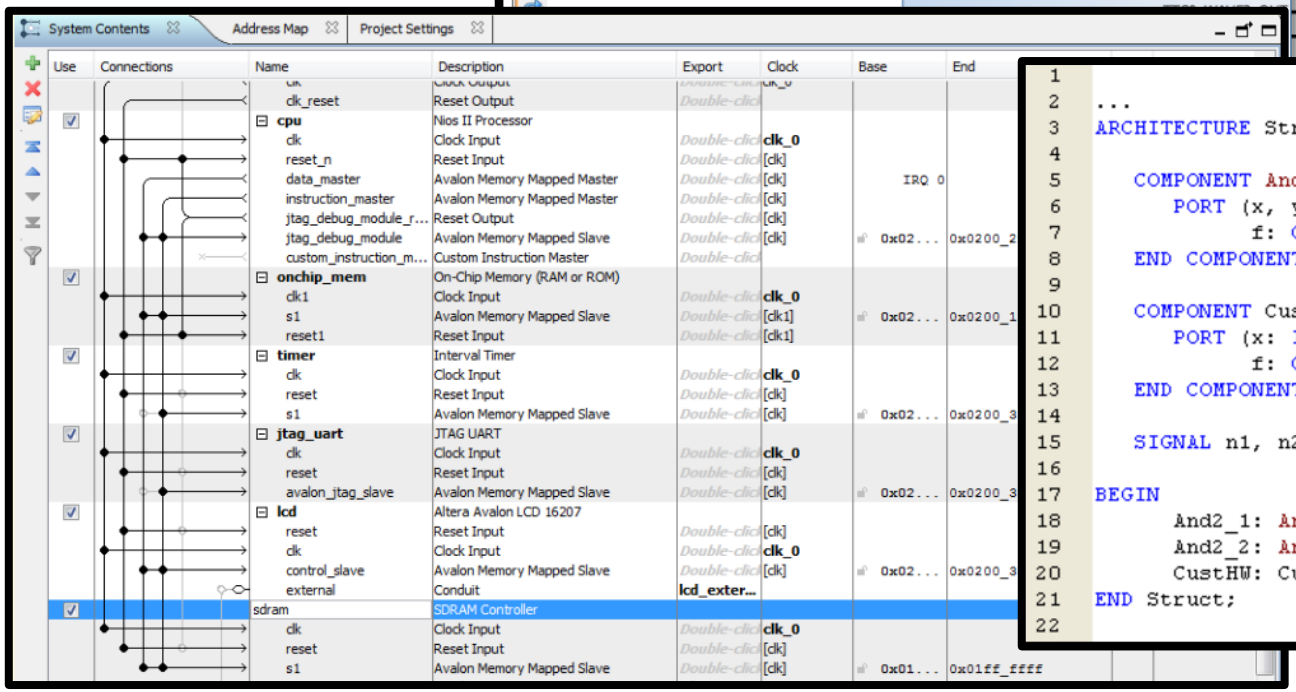
SoC FPGA (4 of 6)

Gateware

Example of Schematic



Example of Connectivity



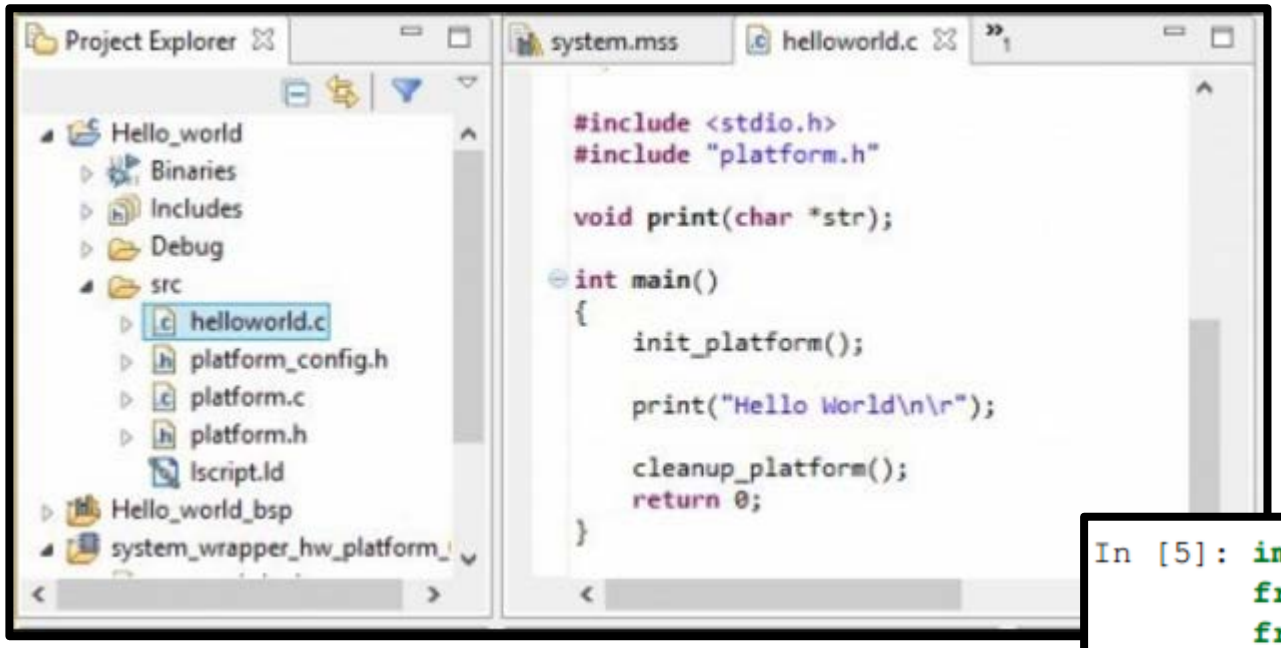

```
1
2   ...
3   ARCHITECTURE Struct OF MyLogic IS
4
5      COMPONENT And2 IS
6         PORT (x, y: IN std_logic;
7               f: OUT std_logic);
8      END COMPONENT;
9
10     COMPONENT CustomHW IS
11        PORT (x: IN std_logic;
12              f: OUT std_logic);
13     END COMPONENT;
14
15     SIGNAL n1, n2: std_logic;
16
17  BEGIN
18        And2_1: And2 PORT MAP (    ,     ,    );
19        And2_2: And2 PORT MAP (    ,     ,    );
20        CustHW: CustomHW PORT MAP (    ,    );
21  END Struct;
22
```

Example of HDL

SoC FPGA (5 of 6)

Embedded software: programming languages

Example of C code



Example of Python code

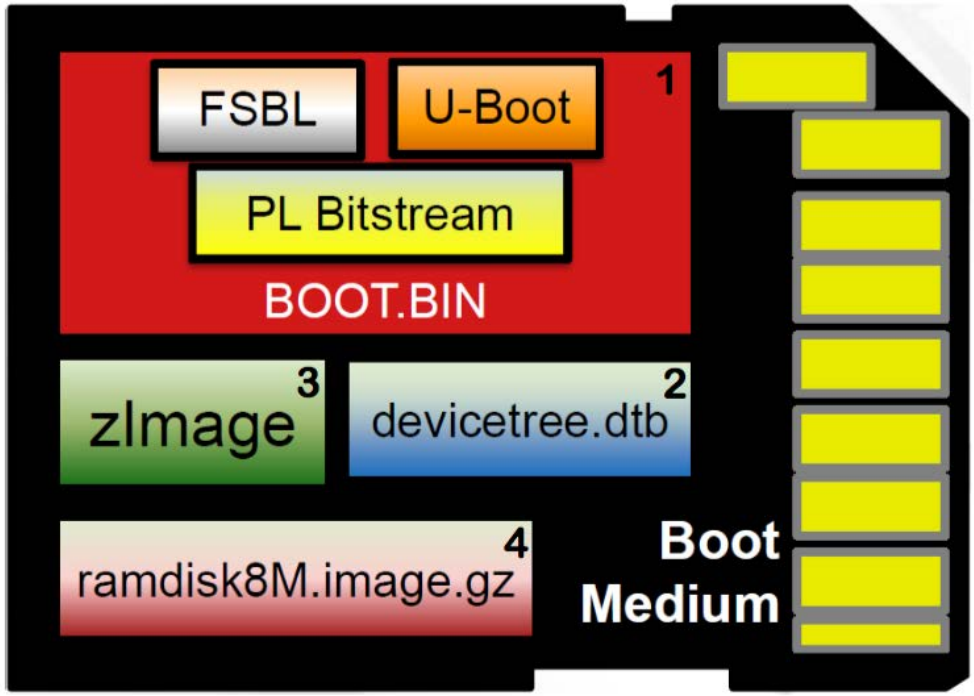
```
In [5]: import time
        from pynq.board import LED
        from pynq.board import Button

        led0 = LED(0)
        for i in range(20):
            led0.toggle()
            time.sleep(.1)
```

SoC FPGA (6 of 6)

Embedded software: Linux operating system

- **BOOT.BIN:**
  - First Stage Boot Loader (FSBL) (a.k.a. boot manager)
  - U-Boot
  - Programmable Logic (PL) Bitstream
- **zImage (linux kernel)**
- **devicetree.dtb**
- **ramdisk8M.image.gz (file system)**



Introduction to SoC FPGA

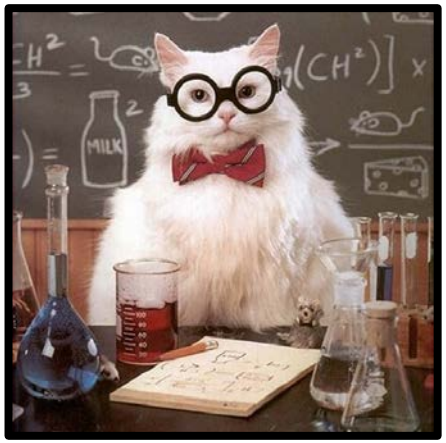
And HardWorking Cores in Electronic Systems

Outline:

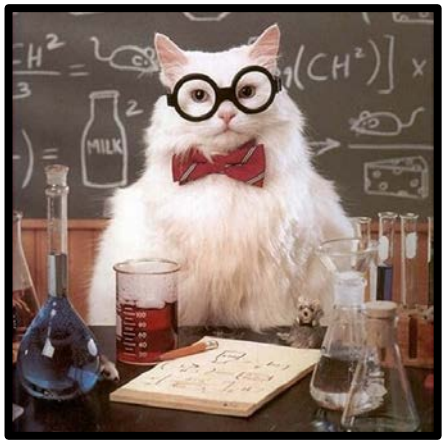
- Introduction
- SoC FPGA
- Summary

Summary

Summary

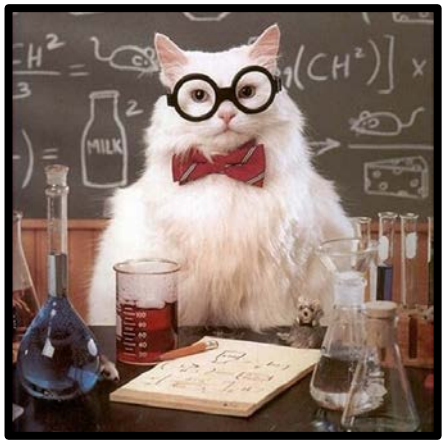


Summary



- **HardWorking cores in Electronics systems:**
  - Multiple types of hardworking cores in electronics systems (e.g. µP, SoC, FPGA)
  - Which one is the best? …it depends… all of them have pros and cons
  - It is vey important to chose the appropriate hardworking core for your application

Summary



- **HardWorking cores in Electronics systems:**
  - Multiple types of hardworking cores in electronics systems (e.g. µP, SoC, FPGA)
  - Which one is the best? …it depends… all of them have pros and cons
  - It is vey important to chose the appropriate hardworking core for your application
- **SoC FPGA :**
  - SoC FPGA is a new generation of hardworking cores featuring SoC and FPGA architectures in one IC
  - HIGH computing power
  - Very good candidate when requiring both processor-based and programmable logic-based in you system
  - Workflow based on hardware/software co-design
  - You can run Linux on it!!

Remember that no matter how much you'll struggle...

...In 30 years, your system will look like this!

Acknowledges

- Markus Joos (CERN) & organisers of ISOTDAQ-17
- Rhodri Jones, Thibaut Lefevre, Andrea Boccardi & other colleagues from CERN BE-BI-QP
- The other members of the ISOTDAQ-17 SoC FPGA Team:
  - Andrea Borga (NIKHEF)
  - Ton Damen (NIKHEF)
  - Peter Jansweijer (NIKHEF)
  - Elena-Sorina Lupu (EPFL)

And now...

And now...

Live Demo!!