Introduction to Reconfigurable Computing

Introduction to the development kits and design software

LECTURE 1

PEDRO SANTOS (SLIDES BY IOULIIA SKLIAROVA)

Universidade de Aveiro

Reconfigurable Computing

A computer architecture that facilitates faster and more complex computing by using reprogrammable integrated circuits to process data that solves problems and directs specified functions.

Offers the higher speed and efficiency of computer hardware to perform serial and parallel tasks combined with software's flexible capacity to allow changes to a circuit's purpose in the field to adjust for changing circumstances.

Core advantages:

- More functionality from simpler and smaller hardware designs
- Cost savings on low volume products and those whose useful life is extended by updating
- Shorter/faster development time-to-market

Original Idea

The earliest work on reconfigurable computer architecture was done at the University of California at Los Angeles – in 1960's Gerald Estrin proposed a standard processor extended with **an array of "variable" hardware**.

The main goal was to permit computations which are beyond the capabilities of present systems by providing an inventory of high speed substructures and rules for interconnecting them such that the entire system may be temporarily distorted into a problem oriented special purpose computer.

At that time, digital technology was not ready for such a revolutionary change.

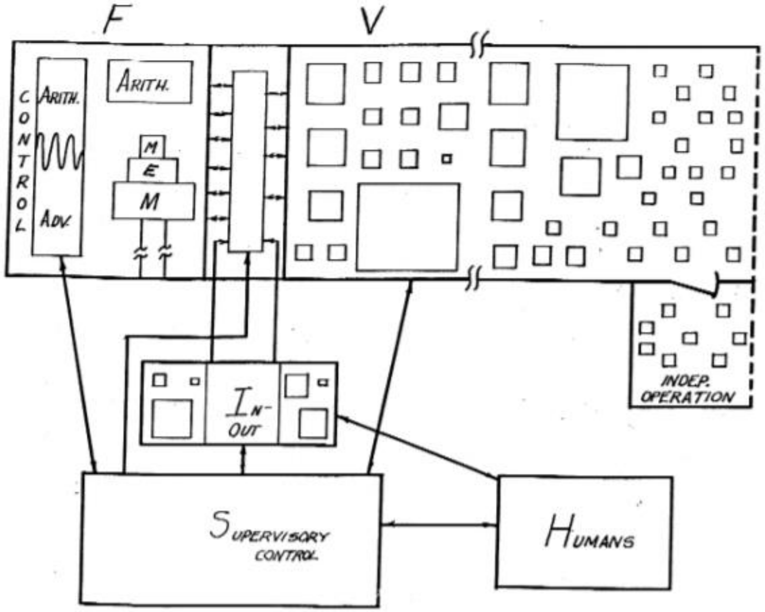


Figure 1. This diagram shows the relations between the fixed machine, the variable structure part, the input-output, the supervisory control, and the humans. (Courtesy of the 1960 Western Joint Computer Conference.)

Other Computing Models

Traditionally, computing was classified into General-Purpose Computing performed by a **General-Purpose Processor (GPP)** and Application-Specific Computing performed by an **Application-Specific Integrated Circuit (ASIC).**

Reconfigurable computing acts as a **trade-off between the two extreme characteristics of GPP and ASIC**, combining the advantages of both.

When compared to GPP:

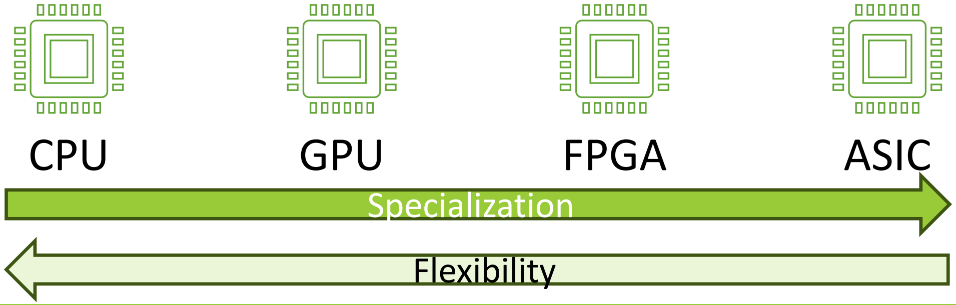
- provides ability to make substantial changes to the datapath itself in addition to the control flow
- can have better performance with respect to a software implementation in GPP but paying this in terms of time to implement

When compared to ASIC:

- possibility to adapt the hardware <during runtime> by "loading" a new circuit on the reconfigurable fabric
- can be used to design a system without requiring the same design time and complexity compared to a full custom solution but being beaten in terms of performance

CPU, GPU, ASIC, and FPGA

- CPU
  - Few cores but very fast, powerful and with extra functionalities
  - Good for: general-purpose processing
- GPU
  - Many small cores, each fairly limited
  - Good for: vectorial operations
- FPGA
  - Reconfigurable digital fabric
  - Good for: complex digital circuits for fast-performing applications (video, audio, etc.)
- ASIC
  - Dedicated digital fabric
  - Good for: specialized tasks that justify mass-production



Overview

Advantages / Features

- Flexible hardware/software co-design
- Customization (parallelism exploration, specialized control and datapaths)
- Fast prototyping
- Faster design cycles, lower risk, cheaper for small to medium size markets compared with ASICs
- Dynamic system upgrades (field updates for both hardware and software)

Target applications

- Application-specific hardware accelerators/coprocessors/processors
- Telecommunications, networks, image/audio processing, cryptography, space, automotive, medical, etc.

Overview

Limitations

- Overhead due to reconfiguration time and/or additional hardware circuit delays

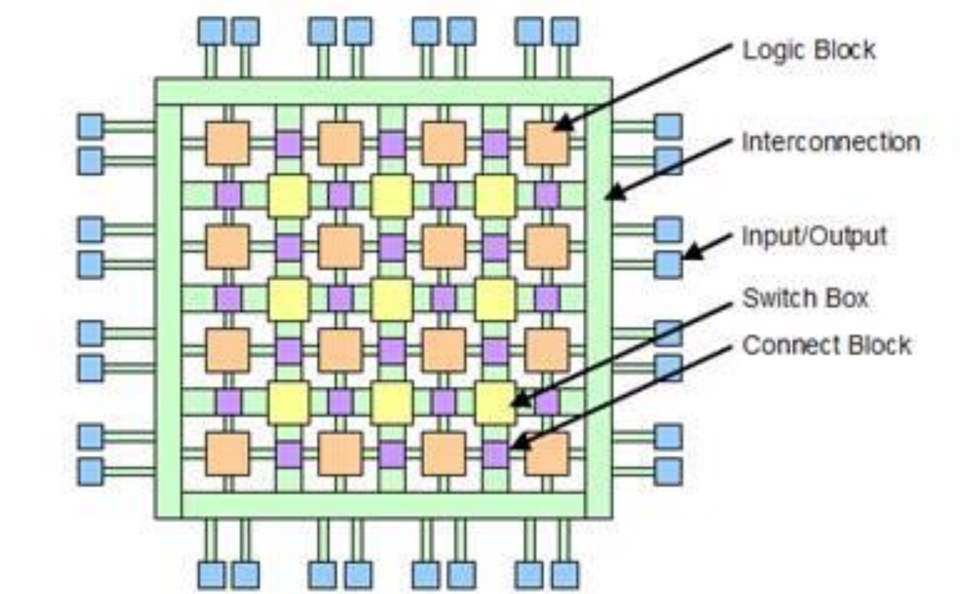
Supporting technologies

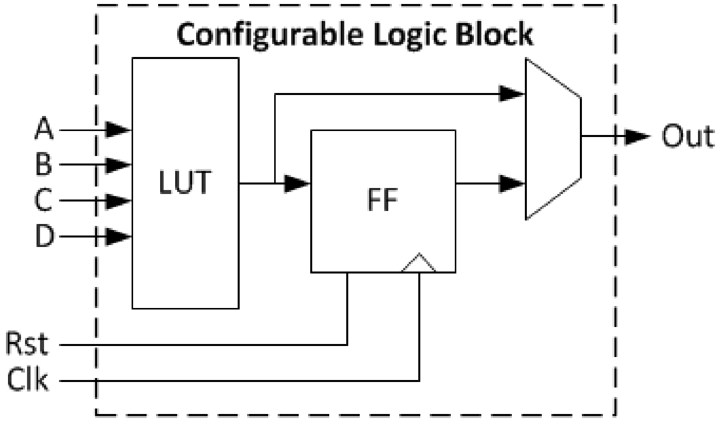
- High capacity FPGAs/Programmable SoC
- Processor customization frameworks
  - Fixed processor cores + extension coprocessors / flexible processors

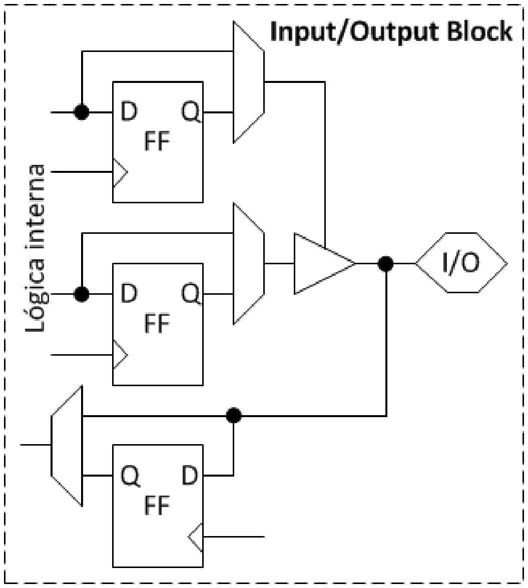
Reconfiguration approaches

- Static (at manufacturing, deployment or boot time)
- Dynamic
  - Total (at run-time between execution phases)
  - Partial (at runtime replacing some components, while others remain operational)

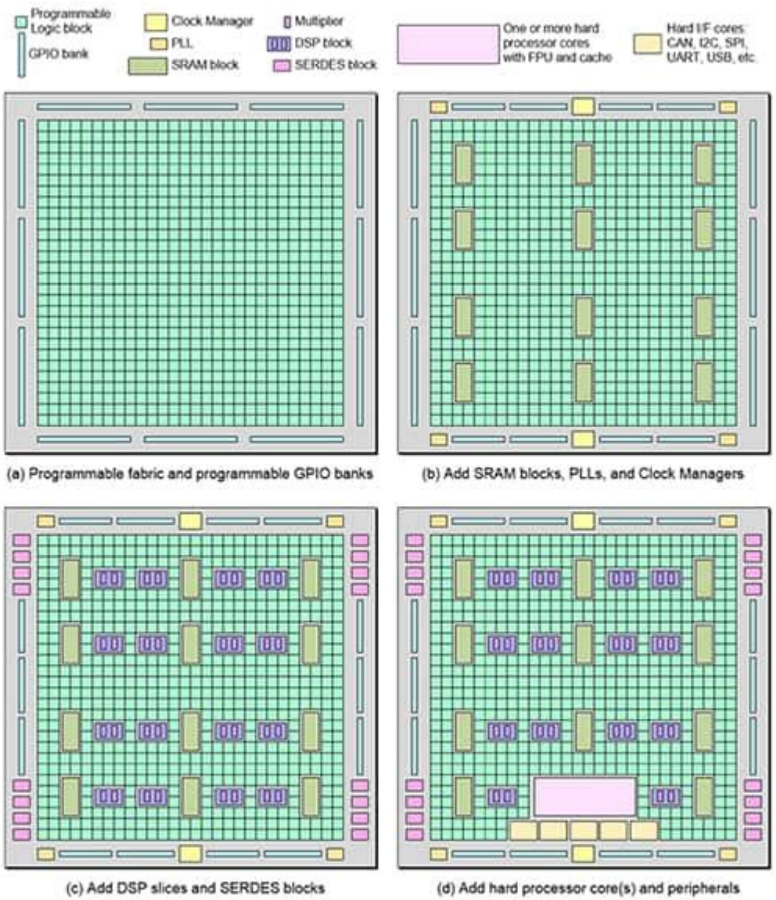
FPGA – Field-Programmable Gate Array







Modern FPGA



and hard processor cores and peripherals (d). (Image source: Max Maxfield)

6-input LUTs configurable as distributed memory

Embedded memory blocks - block RAM with built-in FIFO logic for on-chip data buffering

High-speed serial connectivity with built-in multi-gigabit transceivers

User configurable analog interfaces, incorporating analog-to-digital converters

DSP slices with dedicated multipliers for high-performance filtering

Clock management tiles
providing clock frequency synthesis, deskew, and jitter filtering functionality

Soft and hard processors

- => FPGAs with hard processor cores are referred to as programmable systems-on-chip (PSoC)

Xilinx Programmable Devices

The performance and capabilities of the programmable device offerings from Xilinx span from modest to extremely high:

- traditional FPGAs
- PSoCs (FPGA programmable fabric with a single hard core processor)
- MPSoCs (FPGA programmable fabric with a multiple hard core processors)
- RFSoCs (MPSoCs with RF capability)
- ACAPs (Adaptive Compute Acceleration Platforms)

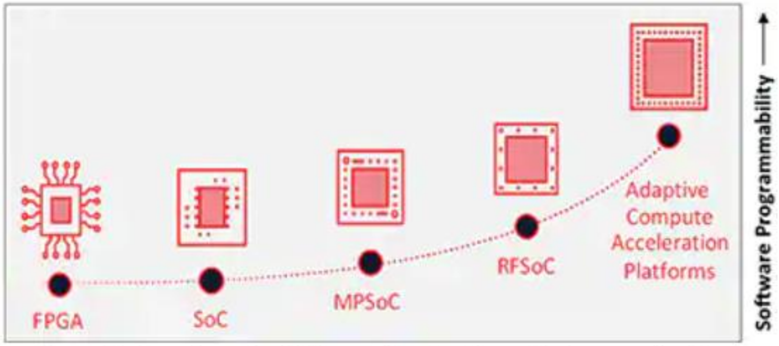


Figure 2: Over time, the Xilinx architectural portfolio has evolved from simple FPGAs containing only programmable fabric, to SoC devices in which the programmable fabric is augmented with a hard core processor, to MPSoCs with multiple processors, to RFSoC with RF capabilities, to the latest generation of ACAPs, which are targeted toward applications like AI. (Image source: Max Maxfield)

Xilinx FPGA

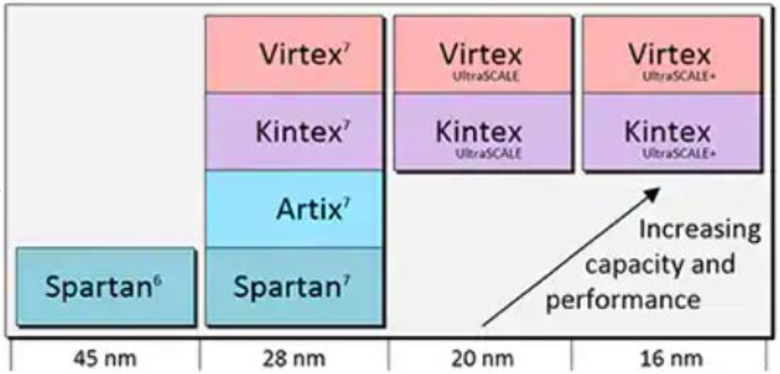


Figure 3: Xilinx FPGA offerings provide a comprehensive multi-node portfolio to address requirements across a wide set of applications. (Image source: Max Maxfield)

Table 4: **Artix-7 FPGA Feature Summary by Device**

Device	Logic Cells	Configurable Logic Blocks (CLBs)		DSP48E1 Slices(2)	Block RAM Blocks(3)			CMTs(4)	PCIe(5)	GTPs	XADC Blocks	Total I/O Banks(6)	Max User I/O(7)
		Slices(1)	Max Distributed RAM (Kb)		18 Kb	36 Kb	Max (Kb)						
XC7A12T	12,800	2,000	171	40	40	20	720	3	1	2	1	3	150
XC7A15T	16,640	2,600	200	45	50	25	900	5	1	4	1	5	250
XC7A25T	23,360	3,650	313	80	90	45	1,620	3	1	4	1	3	150
XC7A35T	33,280	5,200	400	90	100	50	1,800	5	1	4	1	5	250
XC7A50T	52,160	8,150	600	120	150	75	2,700	5	1	4	1	5	250
XC7A75T	75,520	11,800	892	180	210	105	3,780	6	1	8	1	6	300
XC7A100T	101,440	15,850	1,188	240	270	135	4,860	6	1	8	1	6	300
XC7A200T	215,360	33,650	2,888	740	730	365	13,140	10	1	16	1	10	500

Notes:
1. Each 7 series FPGA slice contains four LUTs and eight flip-flops; only some slices can use their LUTs as distributed RAM or SRLs.
2. Each DSP slice contains a pre-adder, a 25 x 18 multiplier, an adder, and an accumulator.
3. Block RAMs are fundamentally 36 Kb in size; each block can also be used as two independent 18 Kb blocks.
4. Each CMT contains one MMCM and one PLL.
5. Artix-7 FPGA Interface Blocks for PCI Express support up to x4 Gen 2.
6. Does not include configuration Bank 0.
7. This number does not include GTP transceivers.

Artix-7 CLB (Configurable Logic Block)

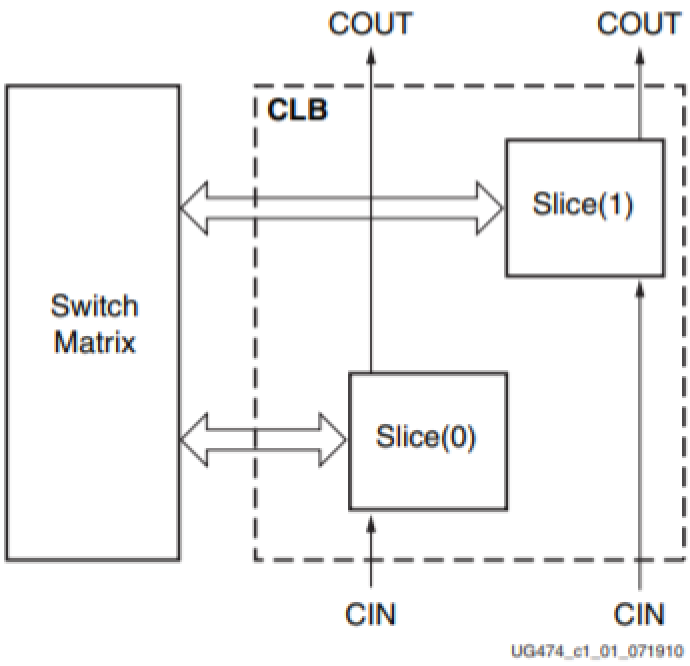
CLBs are the main logic resources for implementing sequential as well as combinatorial circuits.

Each CLB element is connected to a switch matrix for access to the general routing matrix.

A CLB element contains a pair of slices.

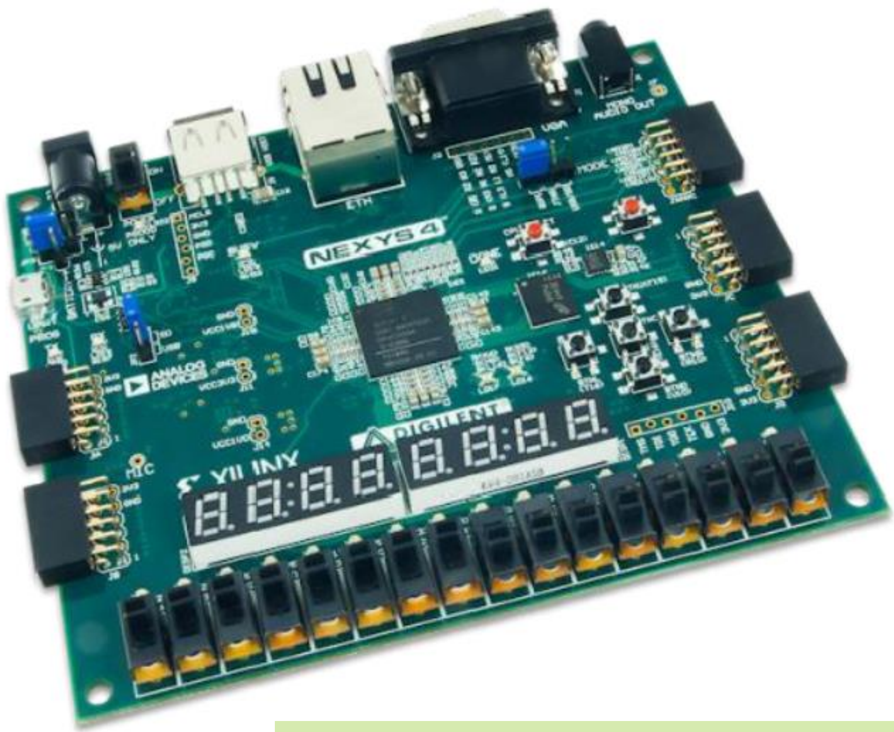
Every slice contains:

- Four logic-function generators (or look-up tables)
- Eight storage elements
- Wide-function multiplexers
- Carry logic



Nexys-4 Development Board

- 16 user switches
- 16 user LEDs
- Two 4-digit 7-segment displays
- USB-UART Bridge
- Two tri-color LEDs
- Micro SD card connector
- 12-bit VGA output
- PWM audio output
- PDM microphone
- 3-axis accelerometer
- Temperature sensor
- 10/100 Ethernet PHY
- 16MB CellularRAM
- Serial Flash
- Four Pmod ports
- Pmod for XADC signals
- Digilent USB-JTAG port for FPGA programming and communication
- USB HID Host for mice, keyboards and memory sticks



FPGA: xc7a100Tcsg324-1

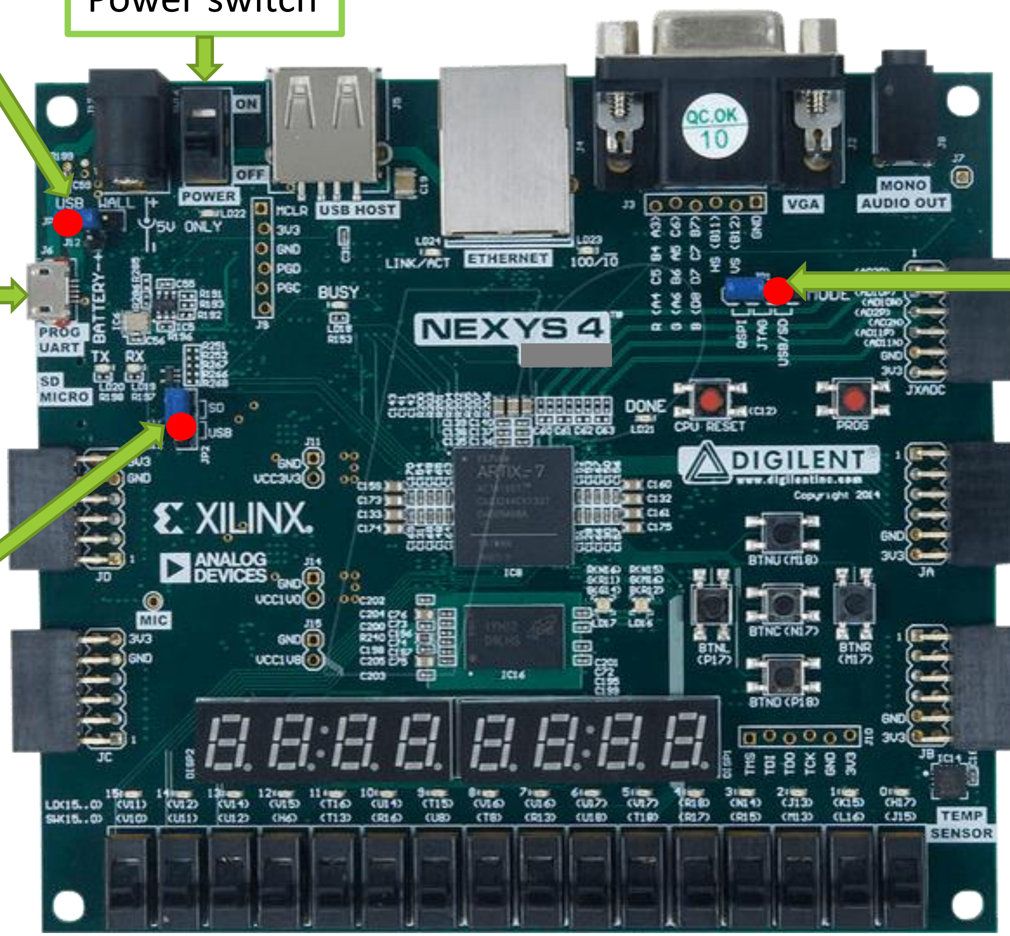
Nexys-4 Development Board

Power select jumper: **USB**

Power switch

USB cable for power supply and programming

External configuration jumper (SD / USB): **USB**



Programming mode jumper: **USB/SD**

Development Tools

Xilinx Vitis

- For portability of projects between different computers, it is recommended to install **version 2022.2**
- Installation instructions are available at the course website
- Requires about **65 GB** of disk space
- License: Vivado Design Suite 30-day evaluation license -> floating license available within the UA campus or with active VPN

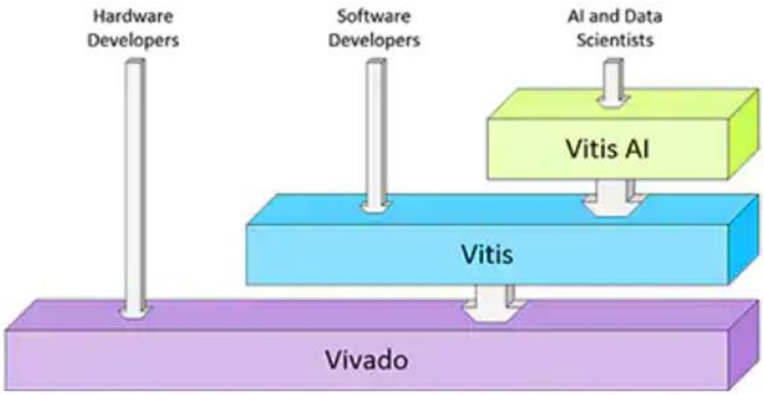
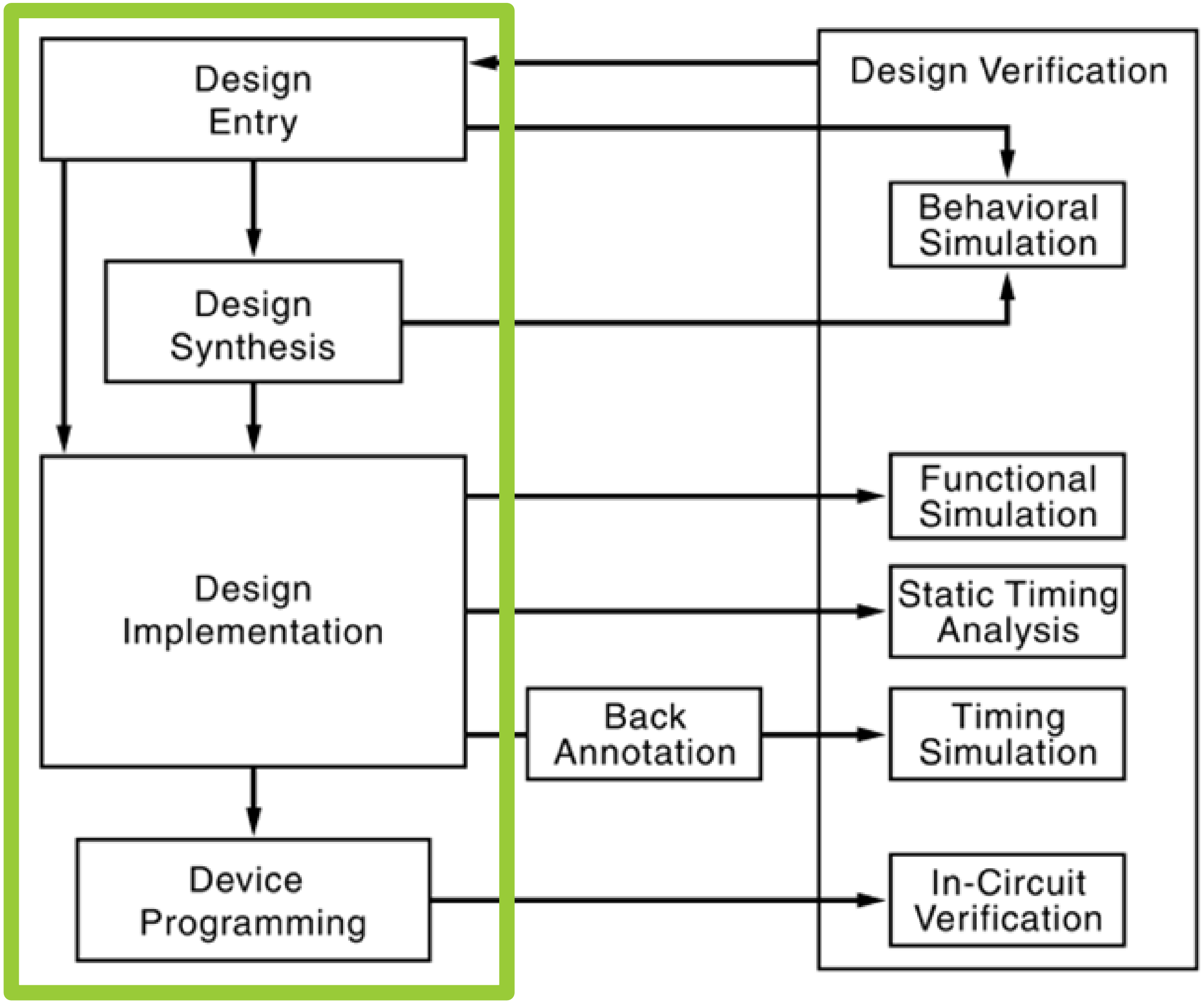


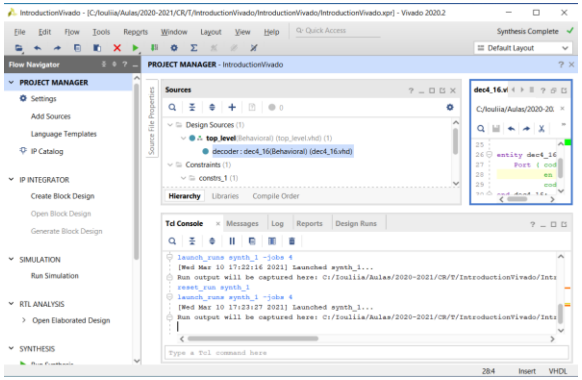
Figure 6: A high-level view of the Xilinx Vivado and Vitis design tool stack reflects how users can work with the tools at the most appropriate levels of abstraction. Hardware designers work with Vivado, software developers work with Vitis, and AI and data scientists work with Vitis AI. (Image source: Max Maxfield)

FPGA Design Flow

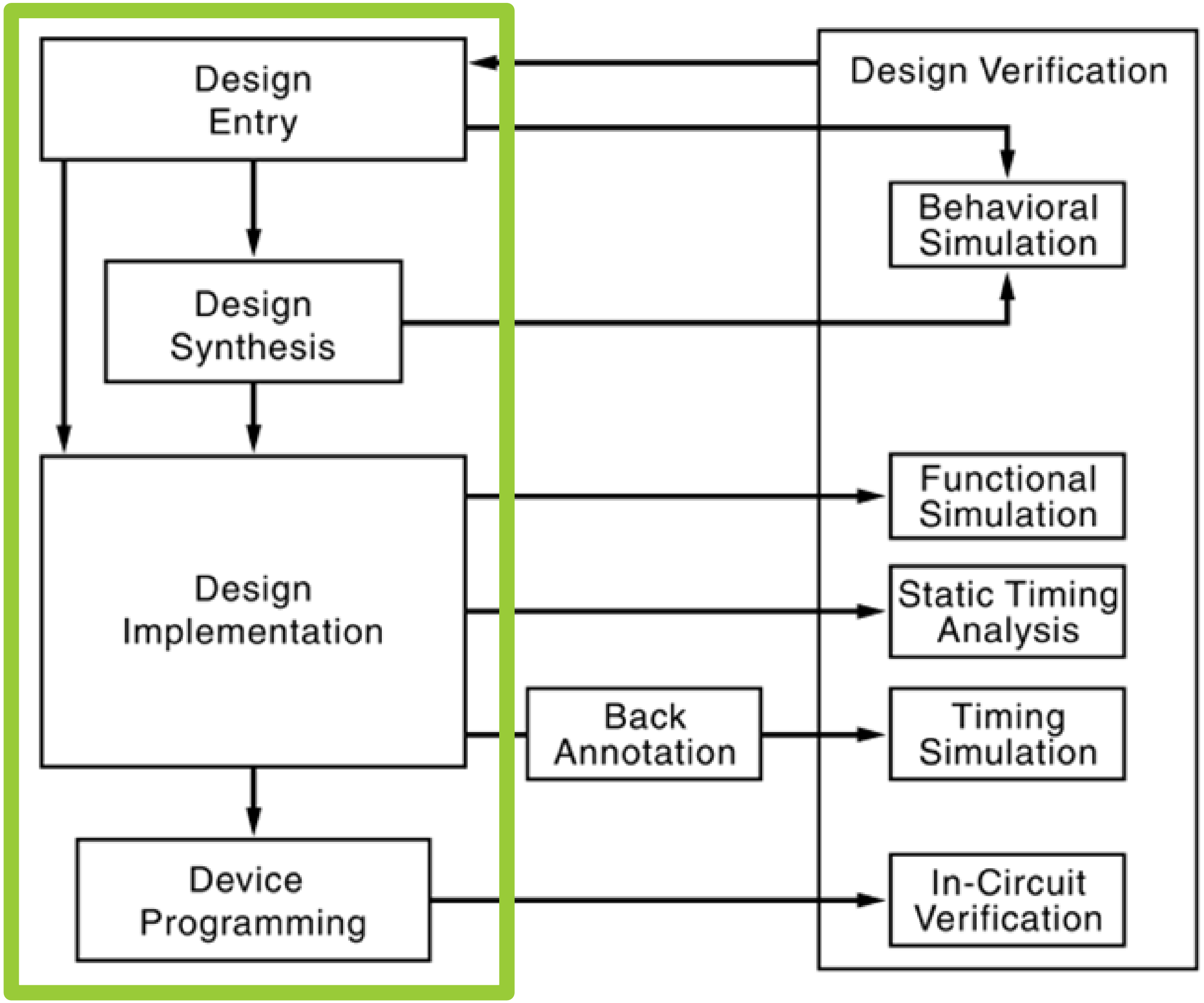


Design entry based on:

- Hardware description languages
- State diagrams
- Schematic capture
- C/C++ and other high level languages



Logic Synthesis

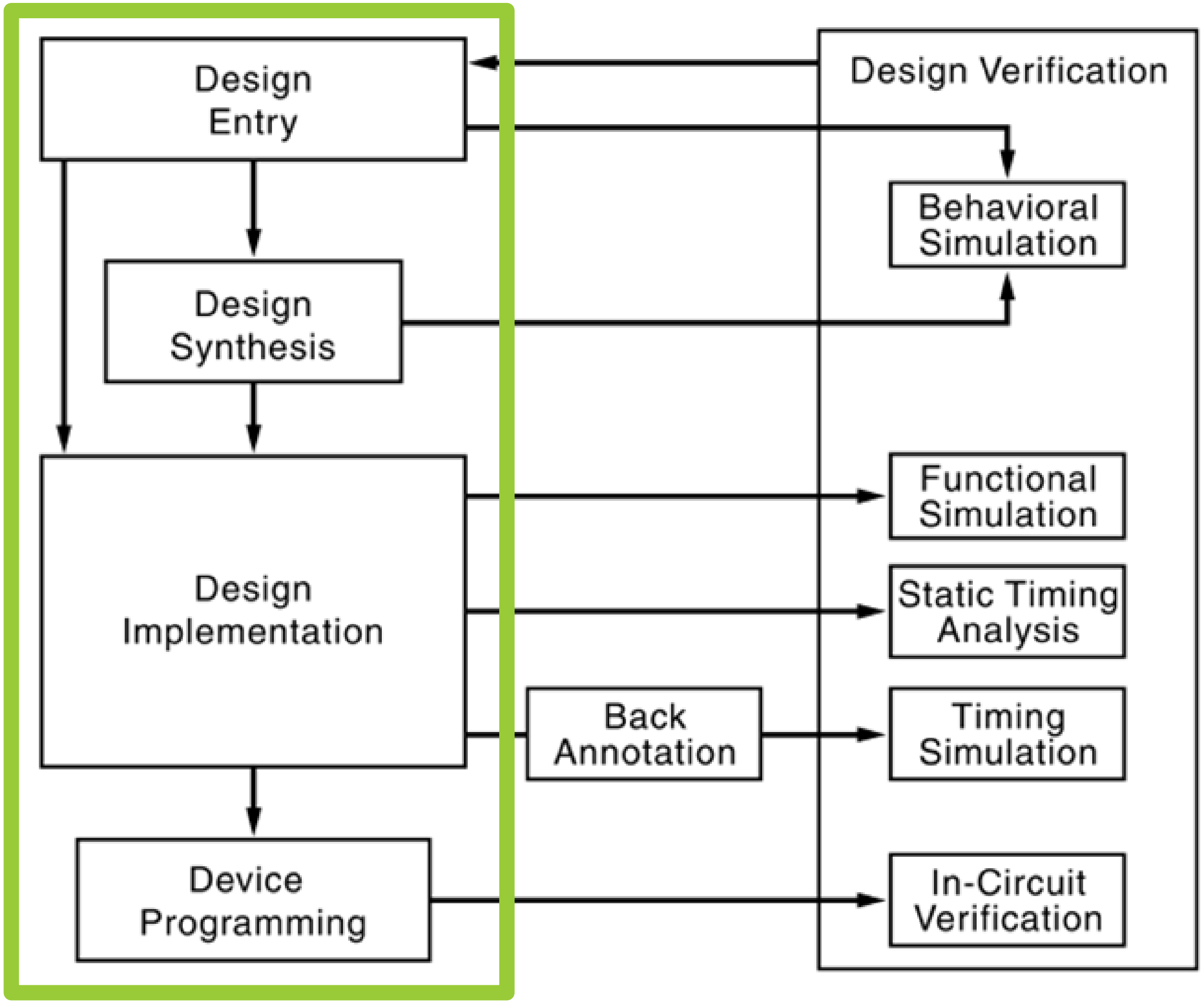


Results in a netlist (i.e. hardware components and their interconnections) that implement the modeled behavior and structure

Outputs

- Netlist
- Circuit performance and resource usage estimations

Implementation (Map, Place and Route)



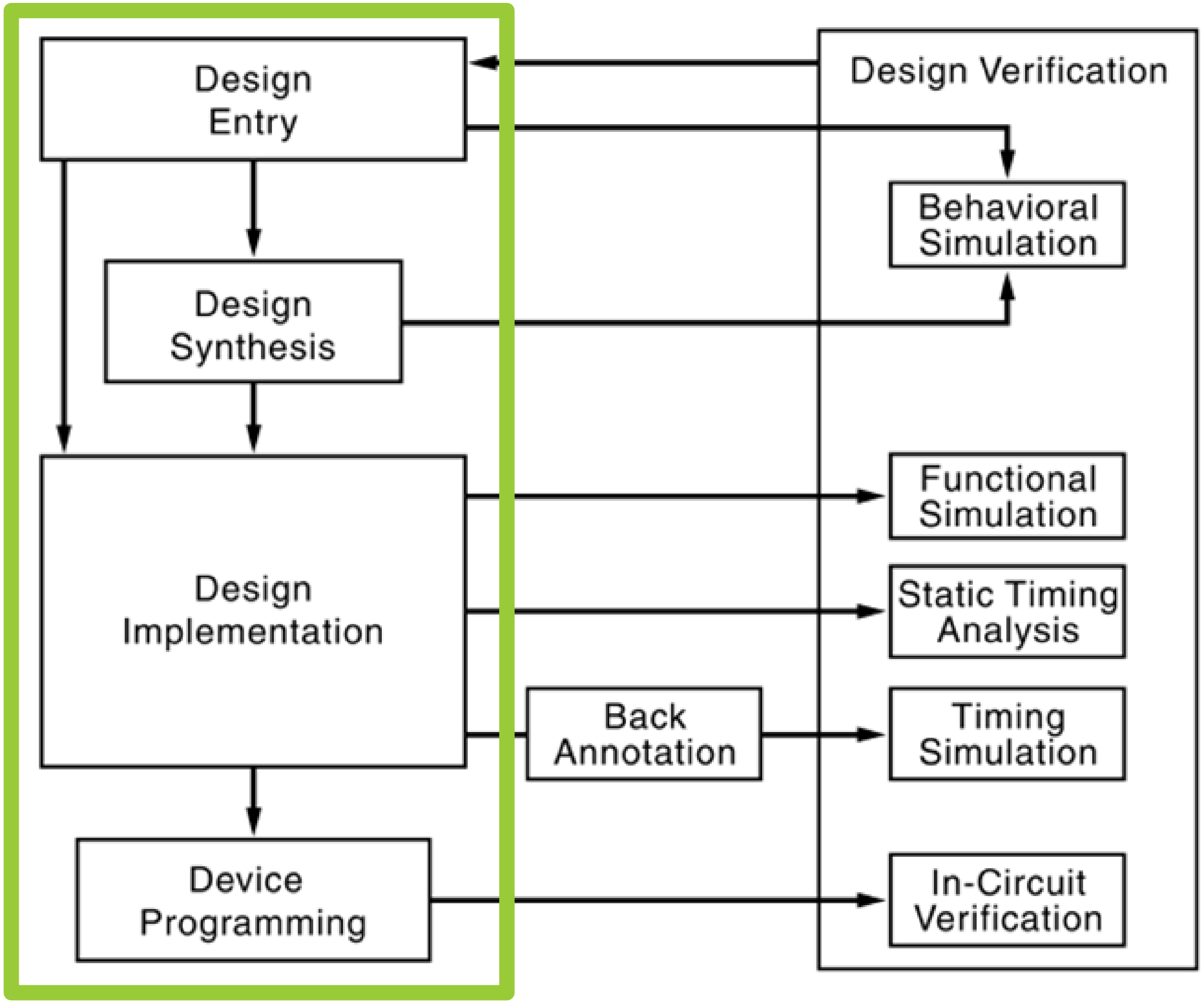
Maps the netlist in FPGA primitives

Places the primitives in specific FPGA locations

Routes the interconnections between primitives

Outputs
- FPGA configuration file
- Reports with FPGA resource usage, circuit delays, power consumption, …

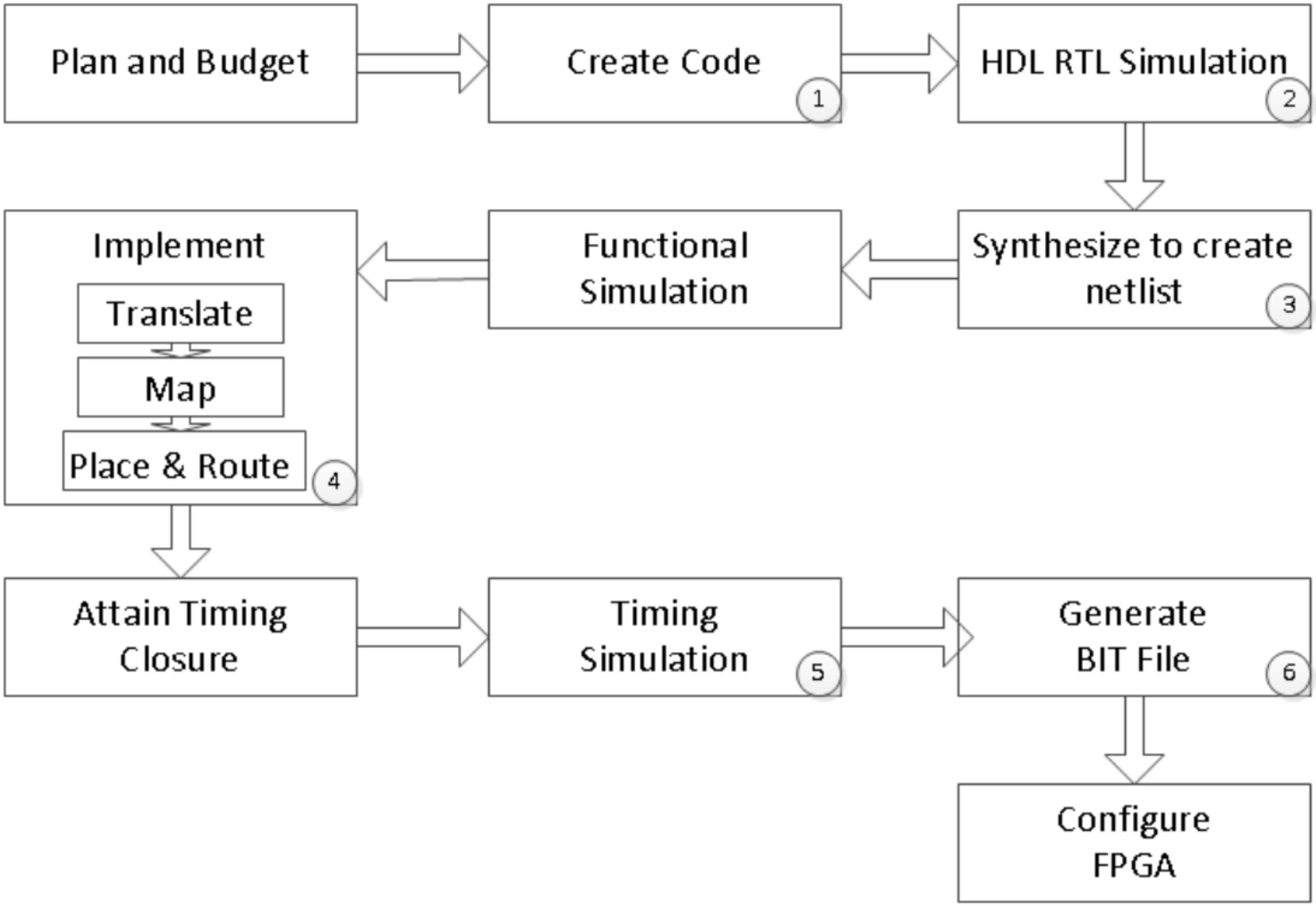
Device (FPGA) Programming



Transfers the configuration file to the FPGA

- Done with a software tool and a programming cable
- Most FPGAs are based on SRAM (volatile configuration)
- Some devices are based on non volatile FLASH memory

A typical workflow



System-on-Chip (SoC)

A SoC is a single integrated circuit integrating at least components like a processor and I/O peripherals to perform general user interface and housekeeping tasks and special hardware accelerators to handle computation-intensive operations.

A SoC integrates a processor, memory modules, I/O peripherals, and custom hardware accelerators into a single chip.

We will construct several SoCs containing a Xilinx MicroBlaze soft-core processor, memory-mapped I/O subsystem that can incorporate custom I/O cores, ...

Hardware-Software Co-Design

The FPGA technology allows us to tailor the processor, select only the needed I/O peripherals, create a custom I/O interface, and develop specialized hardware accelerators for computation-intensive tasks.

Both the hardware and software can be customized to match specific needs.

The methodology of exploiting the trade-offs between hardware and software and developing and integrating them concurrently is referred to as **hardware-software co-design**.

Development Flow of SoC

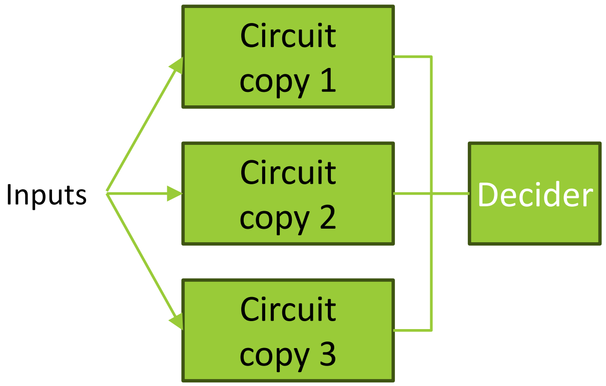
- Partition the tasks to software routines and hardware accelerators
- Design user custom cores if needed
- **Develop the hardware**
- Develop the software
- Implement the hardware and software and perform testing.

Where to find FPGAs?

- **Lab: Prototyping digital circuits**
  - Before sending for production as ASIC
  - Mostly circuit related aspects; timing and resources will inevitably be different

- **Low volume applications requiring high-speed processing**
  - Low volume means it is not enough to justify manufacturing an ASIC
  - E.g. image and audio processing

- **Radiation intensive environments**
  - Intense radiation can cause bit-flips (1 → 0 or 0 → 1)
  - One way to improve resilience to bit faults is replicating the same circuit (say, 3 times)
  - All circuit copies receive the same input; in the end, a decider compares the results and applies a majority vote
  - Space, airplanes, and… particle accelerators? → CERN



Where to find FPGAs (and Job Market)

esa | CERN | TEKEVER

☰ 🔍 → THE EUROPEAN SPACE AGENCY · esa

ENABLING & SUPPORT

The use of reprogrammable FPGAs in space

28918 VIEWS 54 LIKES

ESA / Enabling & Support / Space Engineering & Technology / Microelectronics

Reprogrammable (SRAM based) FPGA (RFPGA), featuring high flexibility, combined with high performance and complexity become increasingly important also for space applications. With satellite lifetimes increased far beyond 10 years, much longer than the validity of telecom standards, reprogrammability in flight becomes a stringent requirement. If software solutions are not possible, RFPGA may soon be the only solution.

Notwithstanding to the general methodology and recommendations outlined in the documents in the "Microelectronics Development Methodology" page, which are mostly applicable to FPGAs as well, specific considerations apply to the use of FPGA in space. This is mostly due to the fact that, in contrary to ASIC or one-time (antifuse) programmable FPGA, the configuration of RFPGA is stored in an SRAM, which is sensitive to SEU's. The radiation behaviour of RFPGA and methods for SEU mitigation is investigated in several studies.

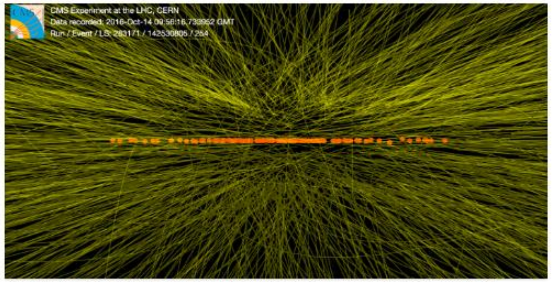
This page also presents lessons learned from space FPGA developments not directly related to radiation-induced issues.

ABOUT NEWS

From capturing collisions to avoiding them

How CERN machine-learning techniques could improve autonomous vehicles

29 AUGUST, 2019 | By Kate Kahle



Around 100 simultaneous proton–proton collisions in an event recorded by the CMS experiment (Image: Thomas McCauley/CMS/CERN)

With about one billion proton–proton collisions per second at the Large Hadron Collider (LHC), the LHC experiments need to sift quickly through the wealth of data to choose which collisions to analyse. To cope with an even higher number of collisions per second in the future, scientists are investigating computing methods such as machine-learning techniques. A new collaboration is now looking at how these techniques deployed on chips known as field-programmable gate arrays (FPGAs) could apply to autonomous driving, so that the fast decision-making used for particle collisions could help prevent collisions on the road.

FPGAs have been used at CERN for many years and for many applications. Unlike the central processing unit of a laptop, these chips follow simple instructions and process many parallel tasks at once. With up to 100 high-speed serial links, they are able to support high-bandwidth inputs and outputs. Their parallel processing and re-programmability make them suitable for machine-learning applications.

An FPGA-based readout card for the CMS tracker (Image: John Coughlan/CMS/CERN)

in 🔍 Search · Home · My Network

TEKEVER

Field-Programmable Gate Arrays Engineer

Porto, Portugal · 1 week ago

Hybrid · Full-time · Mid-Senior level

201-500 employees · Information Technology & Services

See how you compare to 20 applicants. Try Premium for €0

No longer accepting applications

People you can reach out to

Bruno and others in your network

About the job

Are you ready to revolutionise the world with TEKEVER?

Join us, the European leader in unmanned technology, where cutting-edge advancements r
We offer a unique surveillance-as-a-service solution that provides real-time intelligence, en
saving lives. TEKEVER is setting new standards in intelligence services, data and AI technolo

See more

Universidade de Aveiro

Final Remarks

At the end of this lecture you should be able to:

- Describe the main features of reconfigurable computing
- Outline a typical FPGA architecture
- Know difference between FPGA and PSoC
- Identify the main steps of FPGA design flow
- Define software-hardware co-design

To do:

- Install Xilinx Vitis 2022.2