

IEE 2463
Sistemas Electrónicos Programables
Lecture: Introduction to Programmable Systems

Introduction to Electronic Programmable Systems



Electrical Engineering Department
Pontificia Universidad Católica de Chile
peclab.ing.uc.cl



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

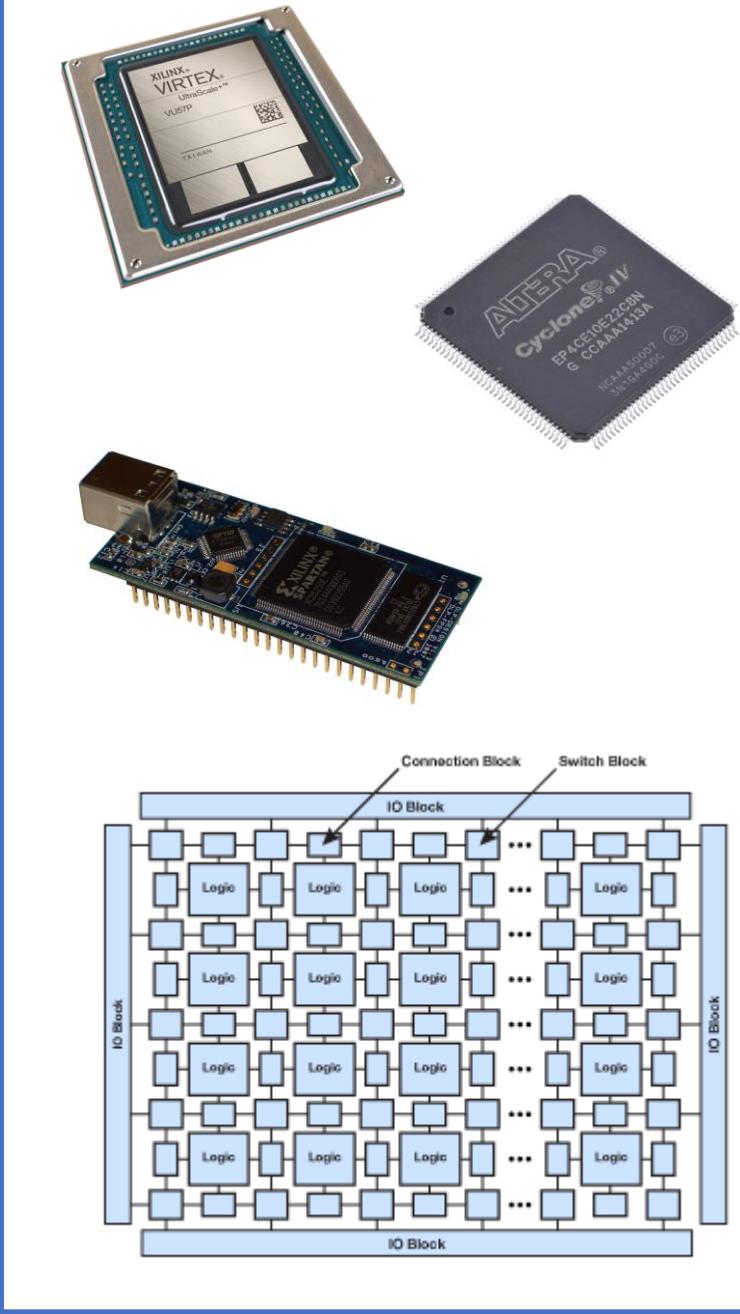
Objectives of this lecture

- Introduction to FPGAs
- Introduction to microprocessors
- Introduction to Zynq SoC devices
- Description of Zybo z7 (personal study)



Basic Concepts Programmable Logic Devices

This section provides a brief overview of the basic architecture of Programmable Logic Devices

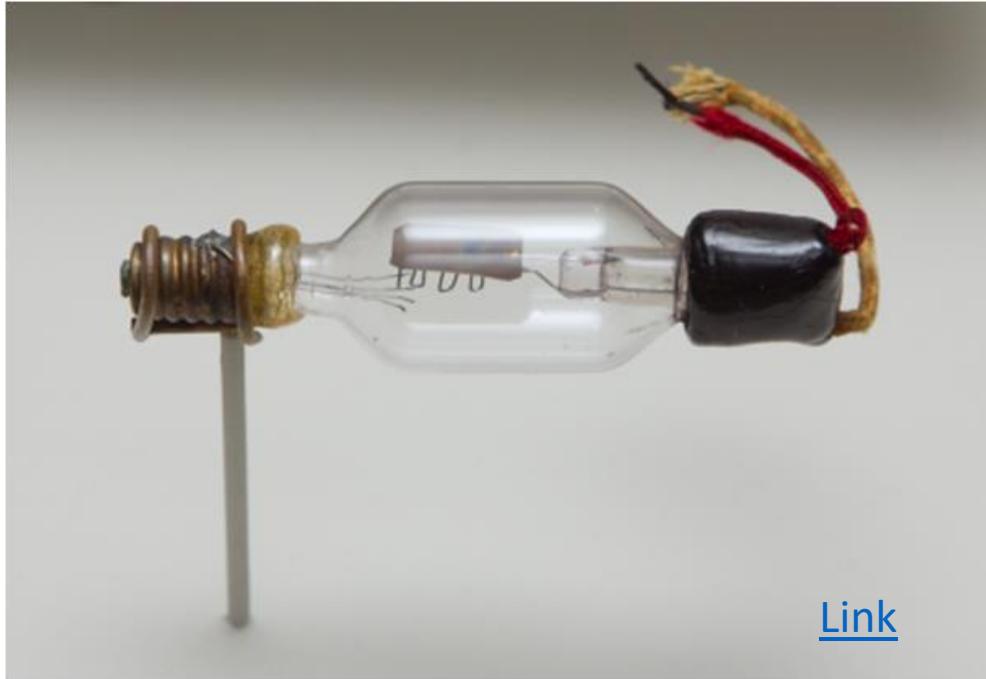


Need to Process Signals

Invention of the TRIODE (Triodo)

Audion (audión), Thermionic Tube (válvula termoiónica) or Vacuum Tube (tubo de vacío)

DeForest invented the Triode in 1907



[Link](#)

- The Triode was developed in 1906 as a diode and was patented by De Forest in 1908.
- A small electric current applied to the cathode (grid) generates a high voltage level in the anode.
- This means the TRIODE was able to amplify signals, specially useful for radio frequency signals with low amplitude.
- Triode was used to amplify telephone signals at large distance.

See this interesting [video](#)

Need to Process Signals

The Transistor

Shockley, Bardeen y Brattain invented the transistor(1947)
Nobel Prize in 1956



By Federal employee -
<https://clintonwhitehouse4.archives.gov/Initiatives/Millennium/capsule/mayo.html>,
Public Domain, <https://commons.wikimedia.org/w/index.php?curid=554340>

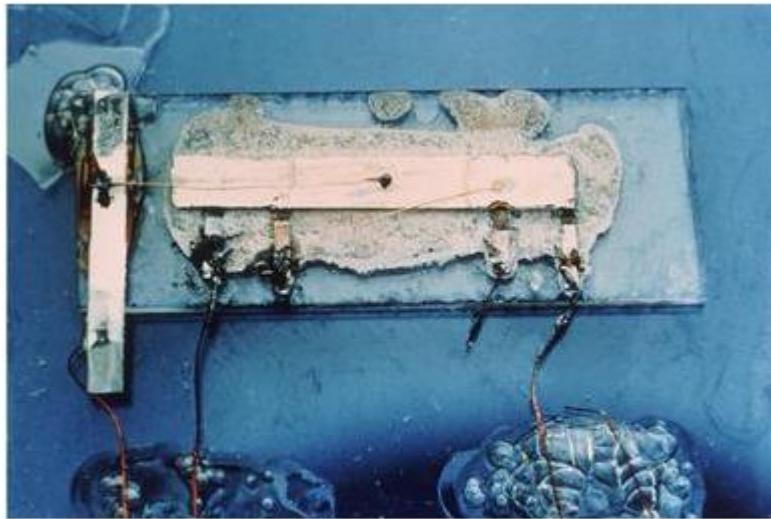


By AT&T; photographer: Jack St. (last part of name not stamped well enough to read),
New York, New York. - eBay itemphoto frontphoto back, Public Domain,
<https://commons.wikimedia.org/w/index.php?curid=17898468>

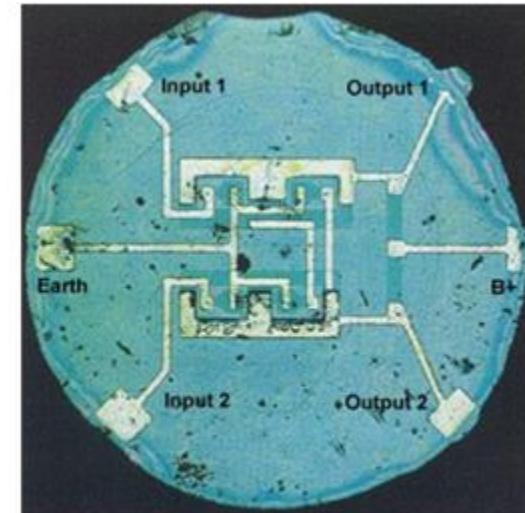
Need to Process Signals

The Integrate Circuit (IC)

Kilby y Noyce independently invented the Integrated Circuit (1958)
Nobel Prize for Kilby en 2000



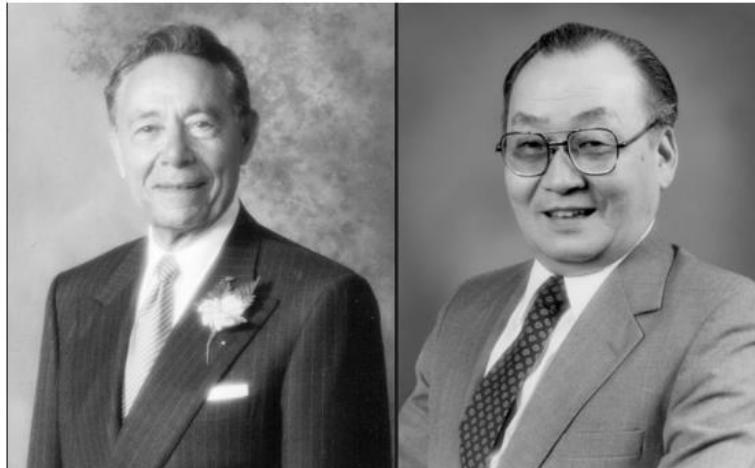
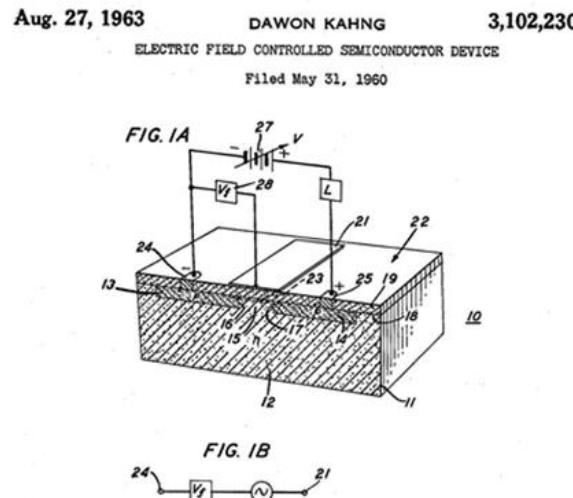
By Source, Fair use,
<https://en.wikipedia.org/w/index.php?curid=19287598>



<https://www.chiphistory.org/83-first-monolithic-silicon-ic-chip>

See this interesting [video](#)

Atalla y Kahng invented the MOSFET (1959)



Source: www.wikipedia.com

Martin Atalla

Dawon Kahng

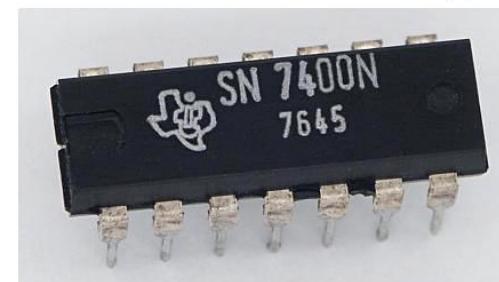
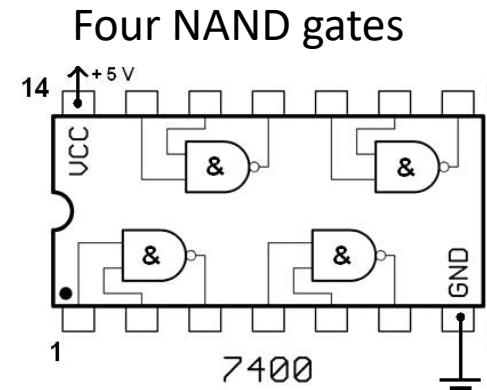
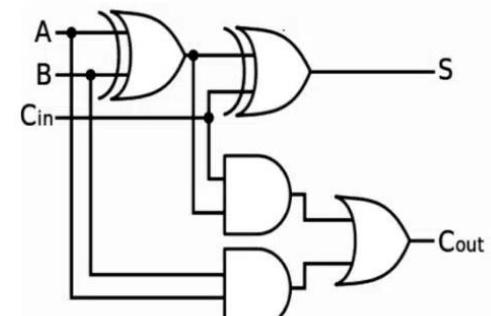
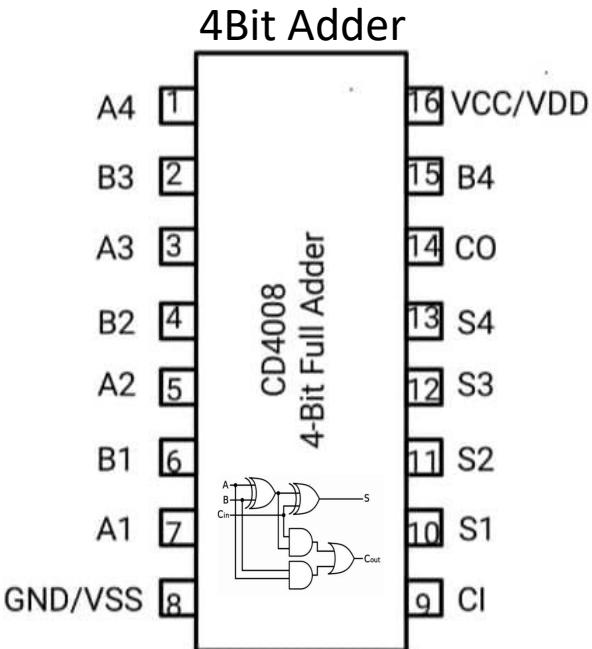
<https://www.computerhistory.org/siliconengine/metal-oxide-semiconductor-mos-transistor-demonstrated/>

Need to Process Signals

...in the old days

- Implementation of digital circuits was based on discrete logic (e.g. SN7400N).
- Several independent Integrated Circuits (ICs) were interconnected, each IC was only one gate or a reduced number of them. (1-bit adder for [instance](#)).
- The process was very error prone and complex.
- Connections among ICs were made based on Printed Circuit Boards.
- State machines were used to design circuits.

	INVERT	AND	NAND	OR	NOR	EX - OR	EX - NOR
European							
American							
IBM ALD's							
Boolean	$Y = \bar{A}$	$Y = A \cdot B$	$Y = \bar{A} \cdot \bar{B}$	$Y = A + B$	$Y = \bar{A} + \bar{B}$	$Y = A + B$	$Y = \bar{A} + \bar{B}$
Truth Table	A L H H	Y H L H	A L H H	B L H H	Y L H H	A L H H	B L H H



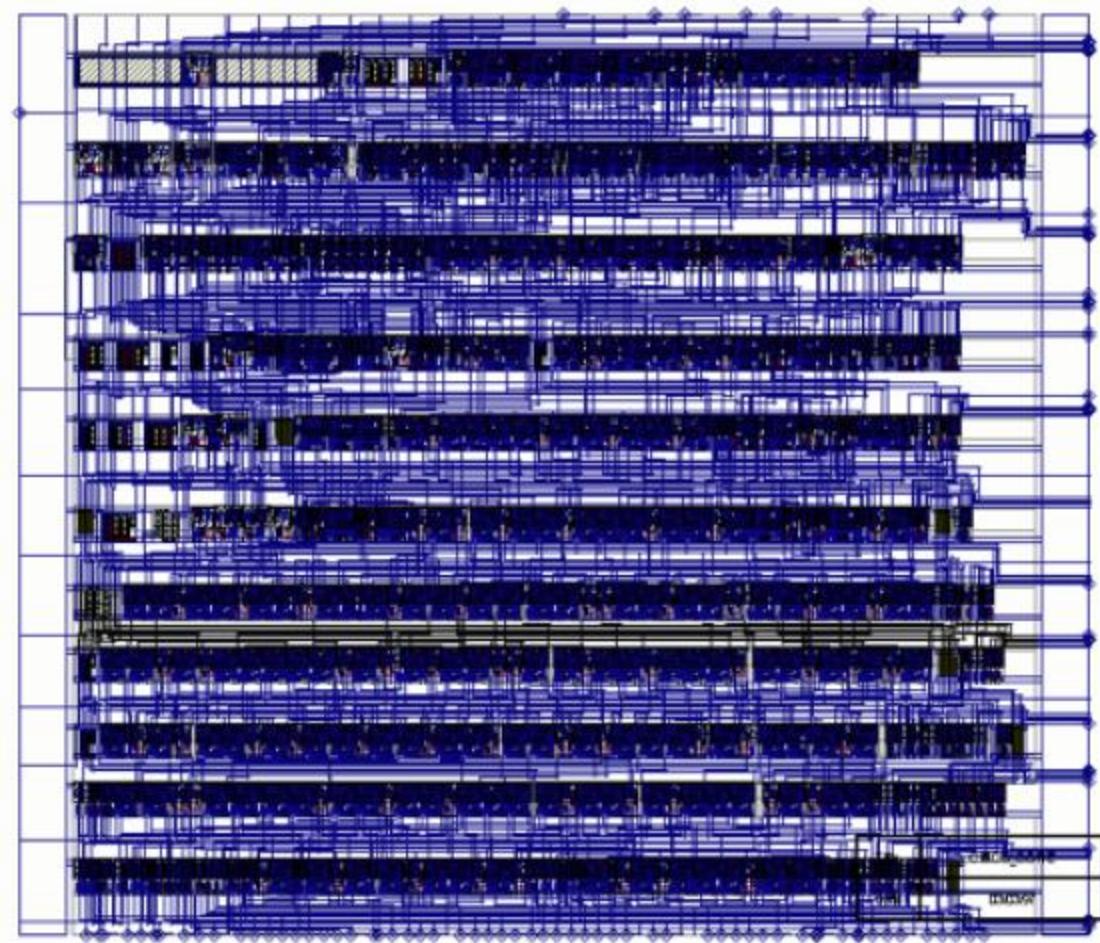
1970s PCB



Early Integrated Circuit based on Gate Arrays

Its Architecture

- Large numbers of miniaturized transistors and other electronic components are integrated together on the chip.
- Integrated circuits revolution started around the 70s and is still undergoing.
- Rows of gates - often identical in structure.
- Gates can be connected to create full customizable circuits.
- Once it is fabricated. It can not be changed.
- This results in circuits that are orders of magnitude smaller, faster, and less expensive than those constructed of discrete components, allowing a large transistor count.



IC based on rows of gates (dar part of the figure)

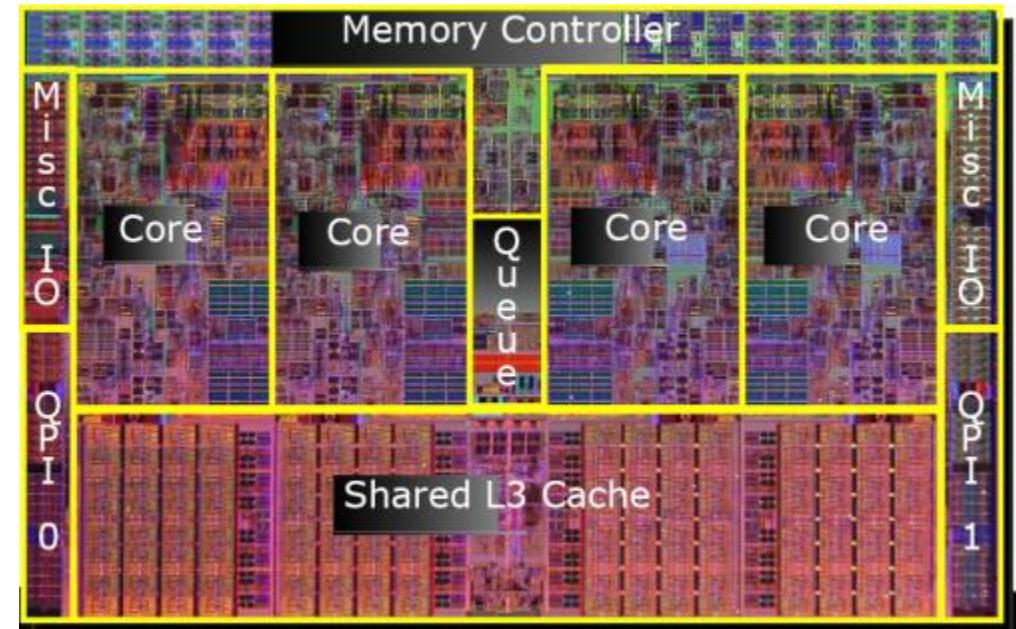
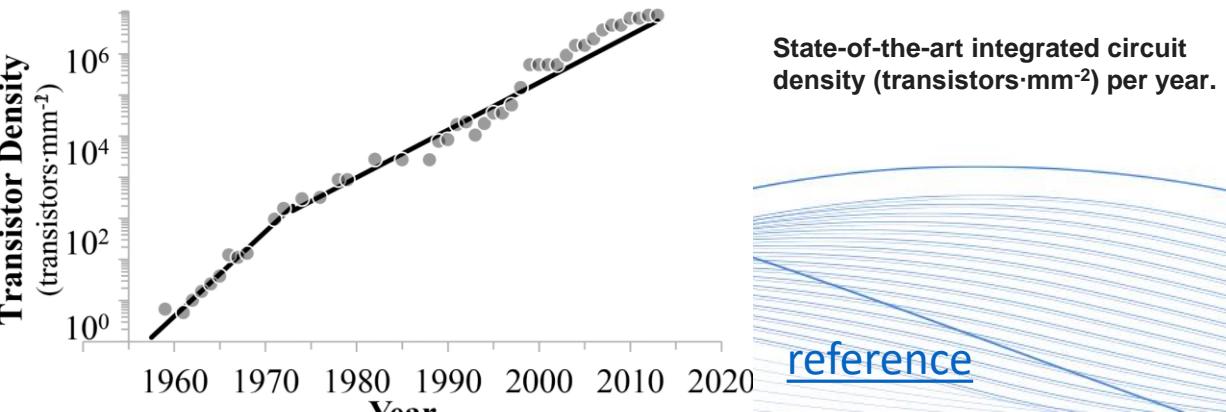
Modern Digital Design

Full customizable IC

- Figure shows modern intel i7 processor, which has 4 physical cores.
- Very high integration. More than 1,000,000,000 of transistors in 216mm².
- Very expensive to design and manufacture the masks.
- Once designed it can not be changed easily.
- Not viable unless the market is very very high (as it is).
- Most electronics industry can not afford a design like this for their own needs. ...

.....then how to do it?.

The answer is Programmable Logic Devices!



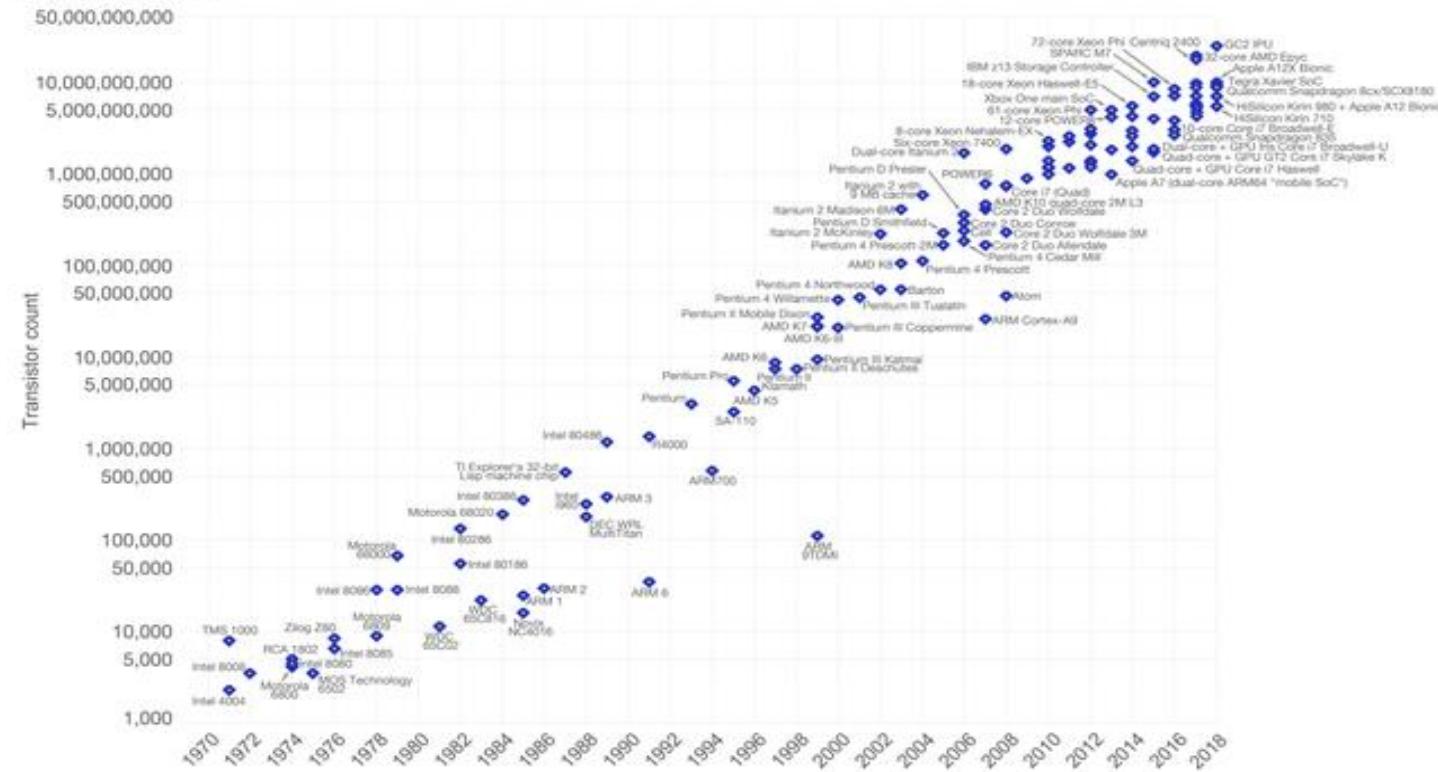
Intel i7, 216mm², 4core, 2M L3 Cache,
1.16 billion transistors.

Need to Process Signals

Moore Law

Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)

The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

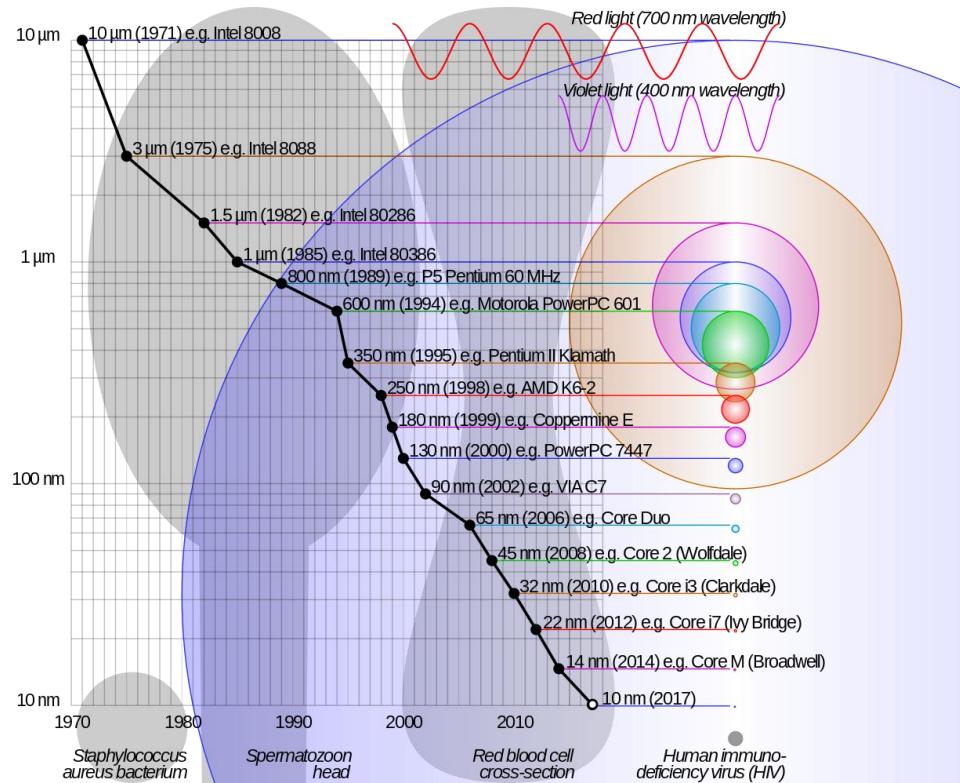
Our World
in Data

"The number of transistors within
comercial Integrated Circuits double
every two years"

Gordon Moore. *Cramming more
components onto integrated circuits,*
Electronics, Abril 1965



Necesidad de Procesar Moore Law

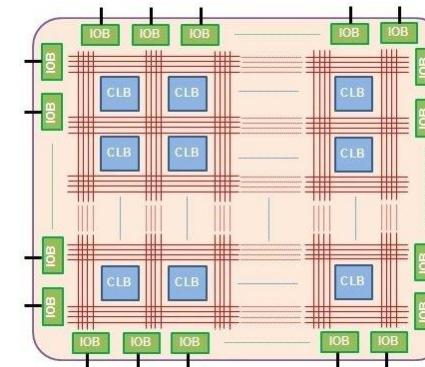


By Asimzb - Own work, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=62233162>

Progress of miniaturisation, and comparison of sizes of semiconductor manufacturing process nodes with some microscopic objects and visible light wavelengths

What is it?

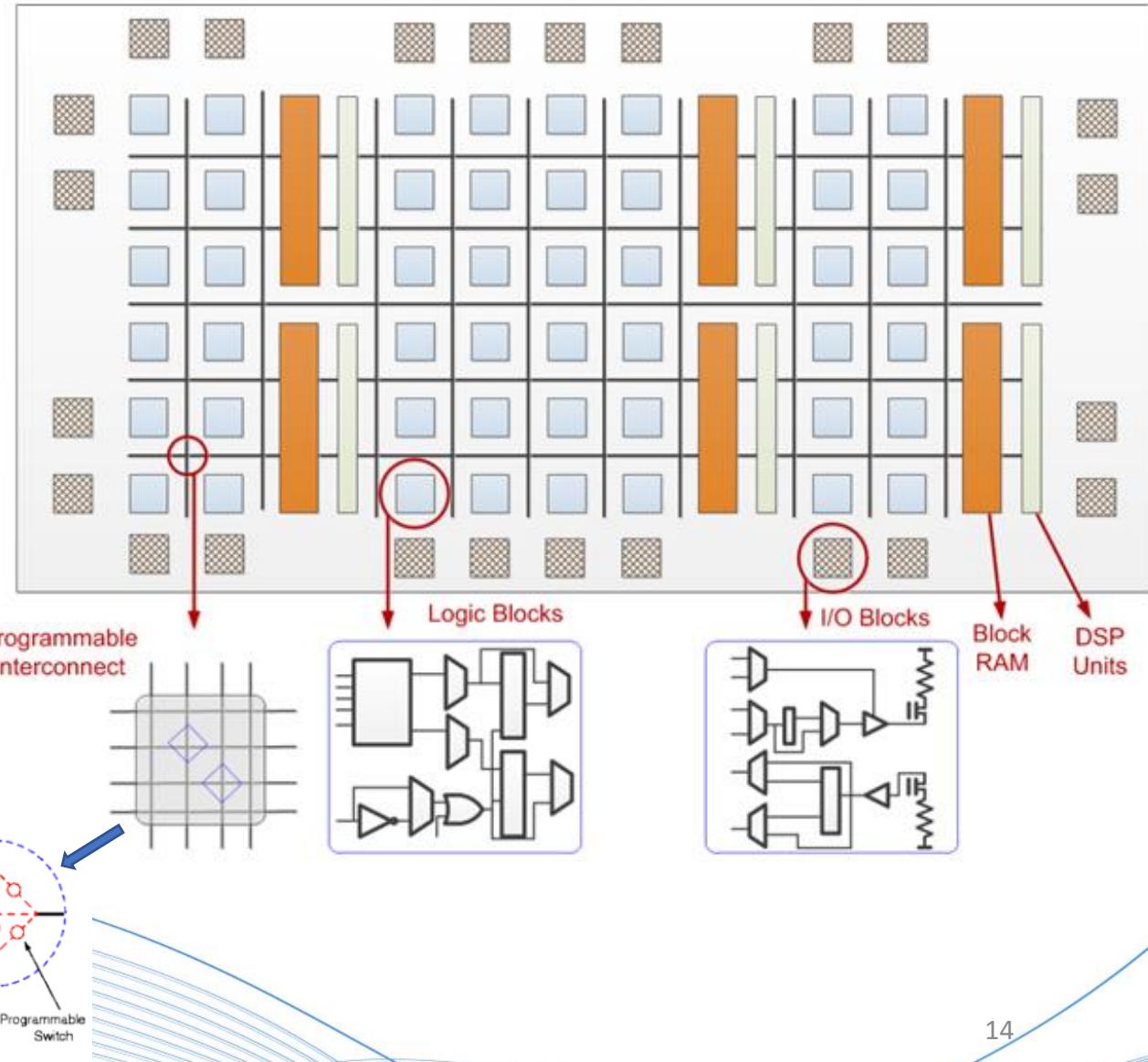
1. A PLD is a group of programmable logic gates, such as: AND, OR, NOR, NAND, XOR,XNOR, NOT, BUFFER, FLIPFLOPS , Look-up Tables (LUT) and multiplexers.
2. A PLD has no function or usability before its first programming
3. In increase of complexity and number of integrated logic components, PLD can be: Simple PLD (SPLD), Complex PLD (CPLD) and Field-Programmable Gate Array (FPGA). (history of PAL,GAL, PEEL to CPLD and FPGA [here](#)).
4. It allows hardware update to products! What?. E.g., adapt to new standard or add additional features.
5. Xilinx introduced the FPGA in 1985 XC2064. Altera followed, being the two biggest companies for CPLD and FPGA.



Field Programmable Gate Array (FPGA)

Its Architecture

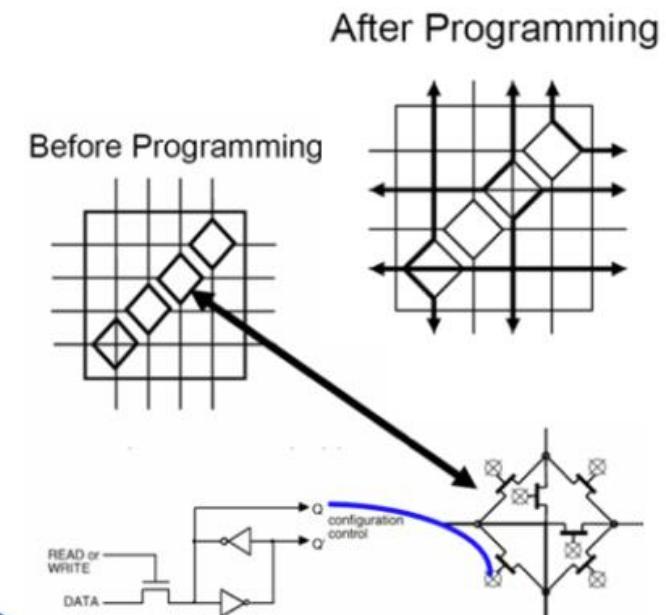
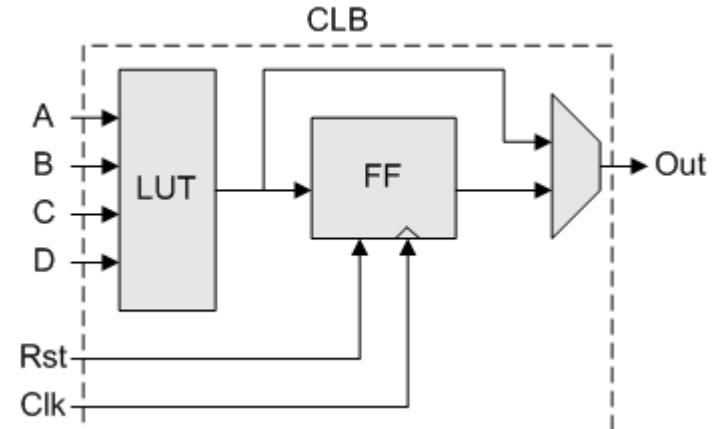
- An FPGA has three main parts
 - Configurable Logic Blocks: Implements logic functions
 - Programmable Interconnects: Implements routing.(also known as switch block)
 - Programmable I/O Blocks: They connect with external components.
- Think of configurable logic block (CLB) as a Lego brick which form you can choose!
- Each CLB is connected to a switch matrix to access the routing structure.
- Switch matrix acts as a programmable multiplexer.
- Today FPGA can also contain ALU, block RAM and DSP units. These are not logic gates.



Configurable Logic Block and INterconnect

Its Architecture

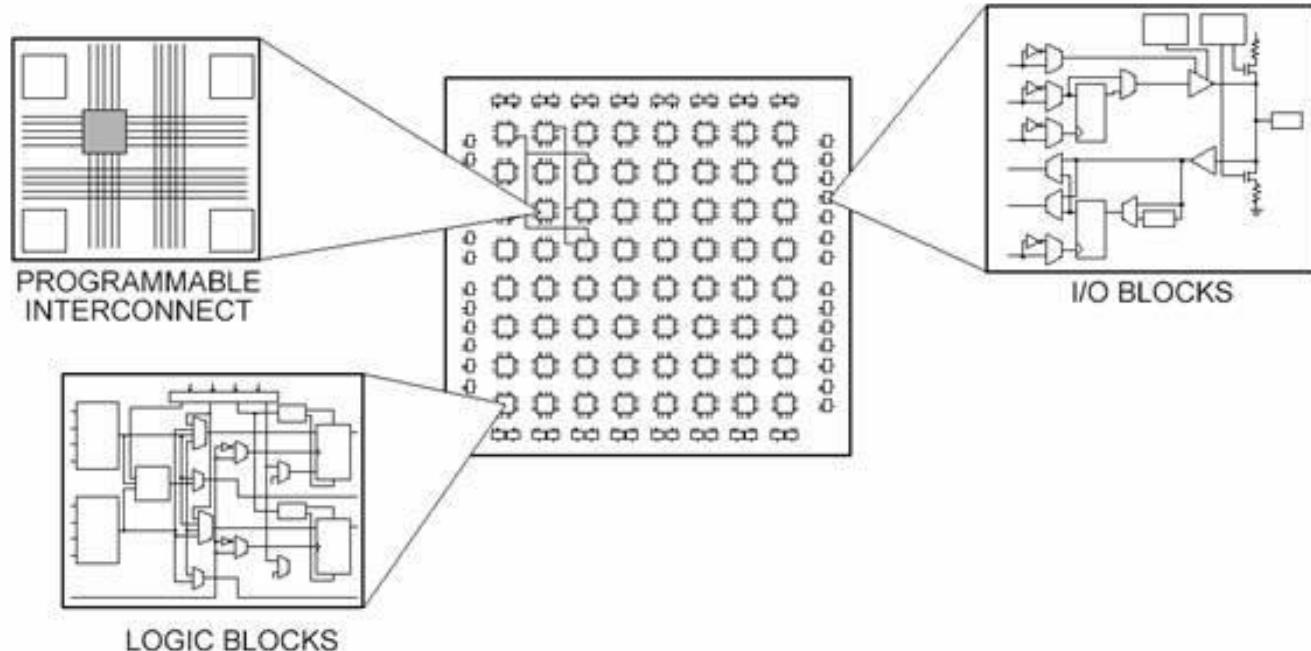
- Commonly a CLB has 4 inputs LUT and a D-FF.
- The LUT can be considered as a certain truth table.
- The LUT can be programmed by the user to implement any 4-input Boolean function.
- Each CLB posses a reset and clock, which drives it operation.
- Each switch on the Interconnect block is controlled by a 1 bit configuration register.
- By putting 1 or 0 in the register, different circuits can be created.



Programming an FPGA

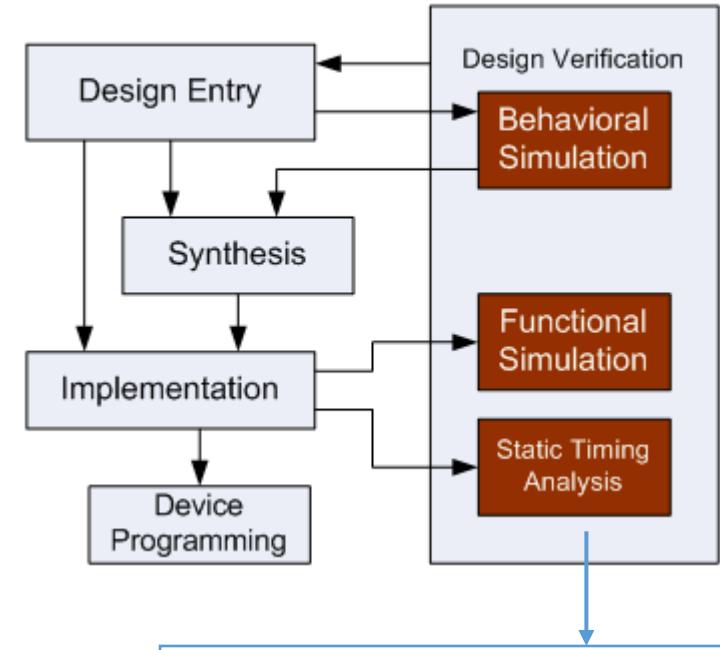
Conceptual understanding

- Program an FPGA is in fact a **configuration** of a circuit.
- After we design a circuit, we load a **bitstream** (1s and 0s) into the FPGA (not a program).
- **Bitstream** is a file generated by the software associated to the FPGA (VIVADO in our case) that configure the LUTs and Interconnects blocks.
- **Bitstream** file can be saved in a local flash memory (download to FPGA during power up) or downloaded from PC.



FPGA Design Flow

- **Design Entry:** It is the description of the logic circuit. It can be done graphically as a schematic, as a Finite State Machine (FSM) or with a Hardware Description Language (**HDL**).
 - Schematic is useful only for modest circuits. In general, HDL is used for larger circuits.
 - VHDL and Verilog are the most HDL. We use **VHDL** in this course.
 - For simplicity, another languages C-like languages, such as Handel-C or Impulse C can be used for as pseudo HDL. But we lose all control over the hardware description as the compiler does the translation to HDL. These languages are not recommended.
- **Logic Synthesis:** Translates the VHDL code into a device netlist format for depicting a complete circuit with logical elements. Synthesis involves checking the code syntax and analyzing the hierarchy of the design architecture. The code is compiled, and the netlist is saved as a .ngc file.



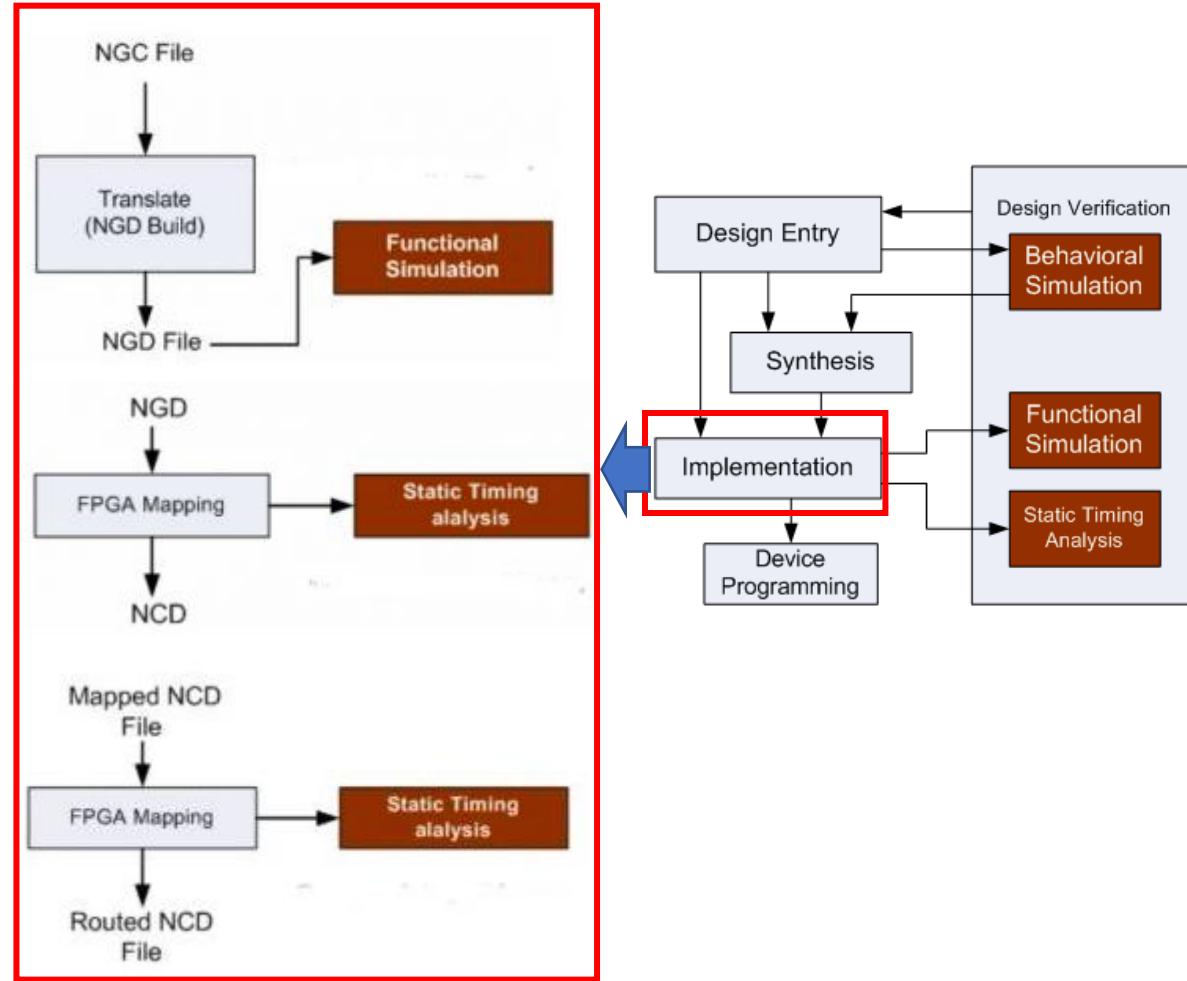
These simulations are done in order to emulate the behavior of the components by providing test patterns to the inputs of the design and observing the outputs.

FPGA Design FFlow

- **Implementation Design:**

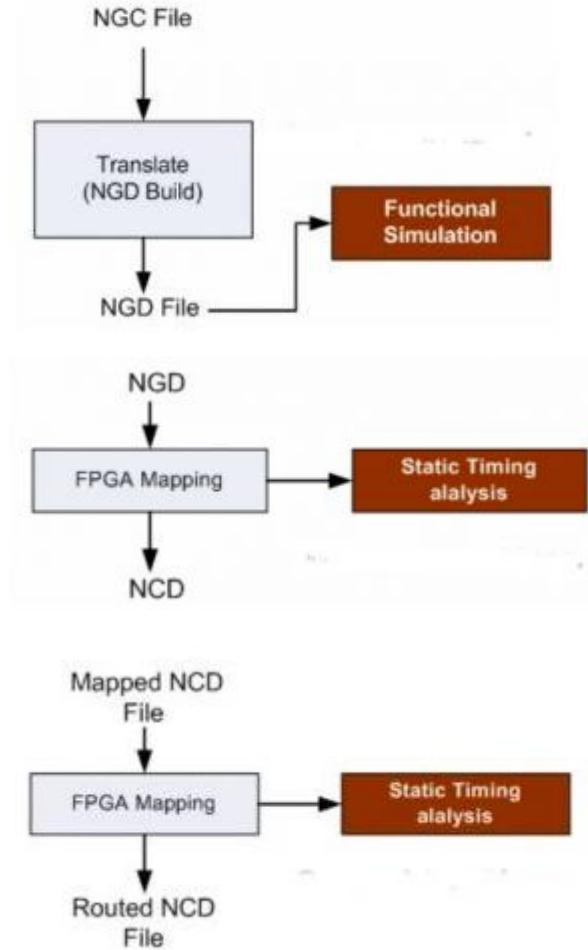
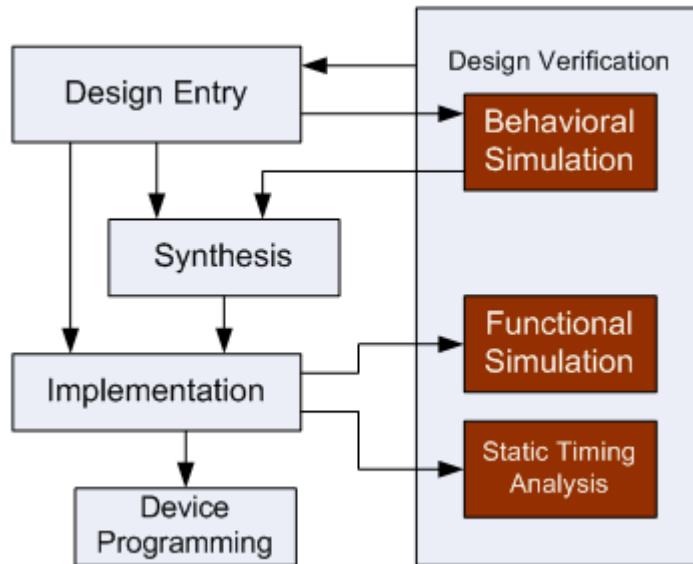
- **Translate:** The netlist .ngc file is transformed into an .ngd file which assign the ports to the physical elements.
- **Map:** Mapping the logic defined in .ngd into the components of the FPGA, generating .ncd file.
- **Place and Route:** It places the sub-blocks from previous process into the CLB of FPGA and make the connections according to the constraints. Generating “routed NCD file”

- **Device Programming:** To program the FPGA, the routed .ncd file is given to the *BitGen* generator program (or bitstream), which generate the **bitstream** file (1s and 0s) readable for the FPGA to be configured (this information prints the internal circuits of FPGA).



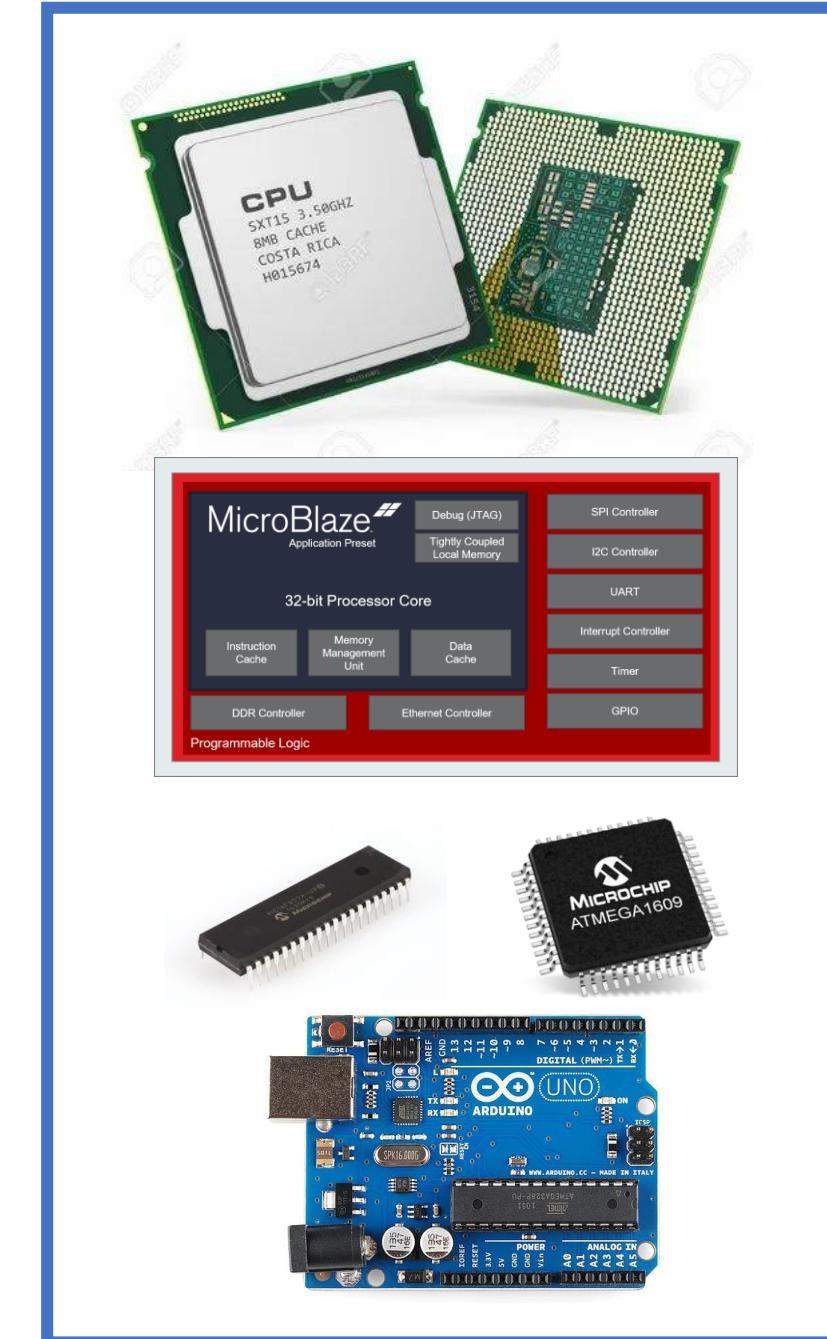
FPGA Design FLow

- 1. Design Entry**
- 2. Logic Synthesis**
- 3. Implementation Design**
- 4. Device Programming**



Basic Concepts Processing Systems Architecture

This section provides a brief overview of the basic architecture of a microprocessor and its principles of operation



Need to Process Data

Is the processor new?

Mechanical processors appeared already on the 1st century. They were analog mechanical computers.



By Andy Dingley - Own work, CC BY 3.0,
<https://commons.wikimedia.org/w/index.php?curid=4944626>

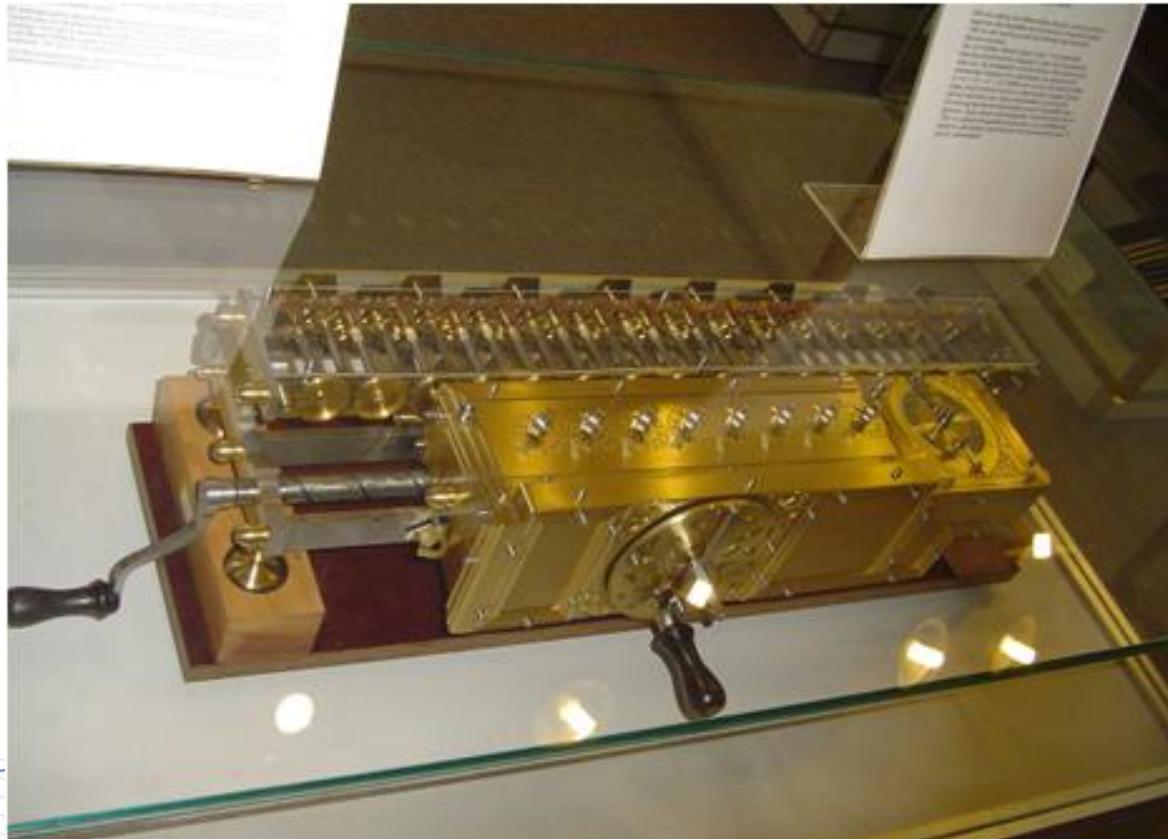


- By Marsyas (wikipedia user). Own work assumed (based on copyright claims)., CC BY 2.5, <https://commons.wikimedia.org/w/index.php?curid=469865>

Need to Process Data

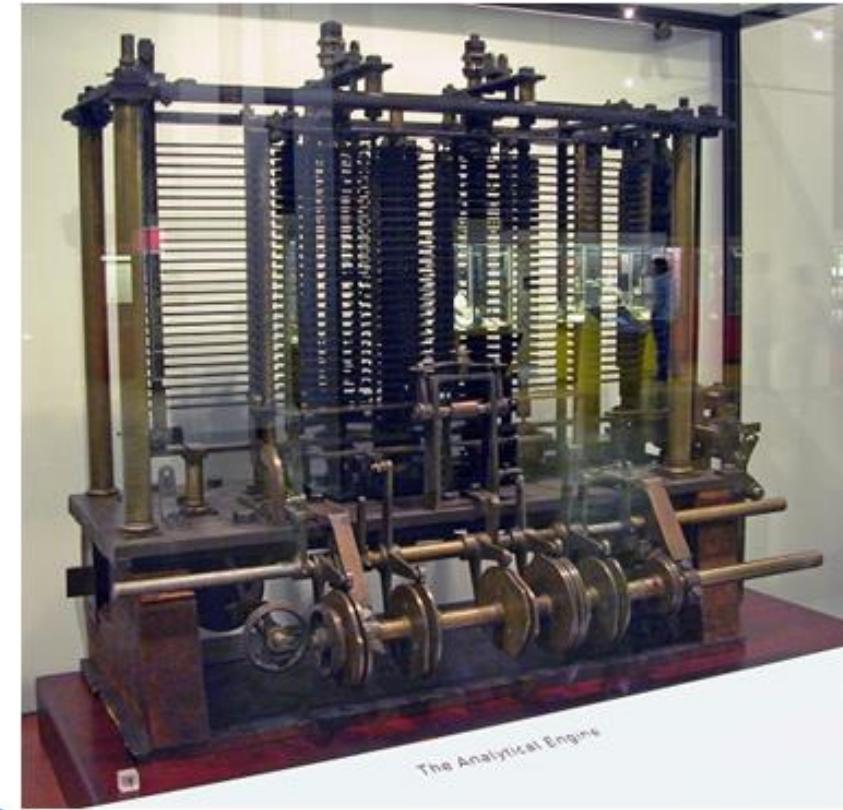
Is the processor new?

Stepped reckoner, Leibniz (1672-94)



By User:Kolossos - recorded by me in de:Technische Sammlungen der Stadt Dresden (with photo permission), CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=925505>

Analytical engine, Babbage (1837)

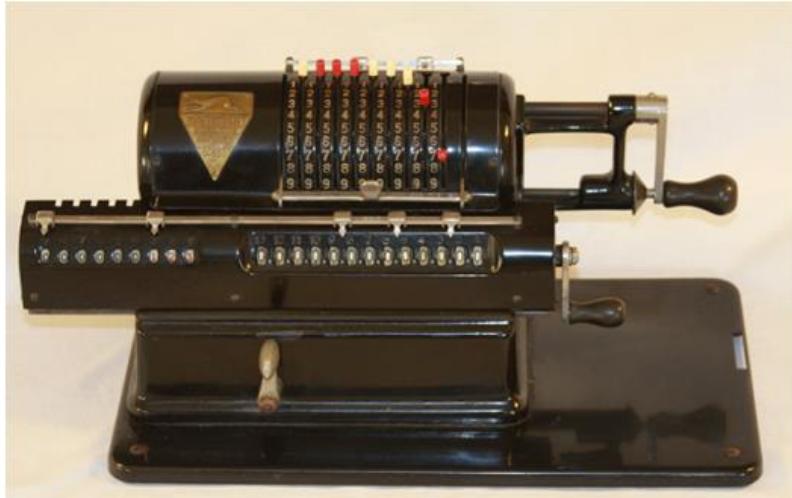


By Bruno Barral (ByB), CC BY-SA 2.5, <https://commons.wikimedia.org/w/index.php?curid=6839854>

Need to Process Data

Is the processor new?

Marchant Calculator (1923)



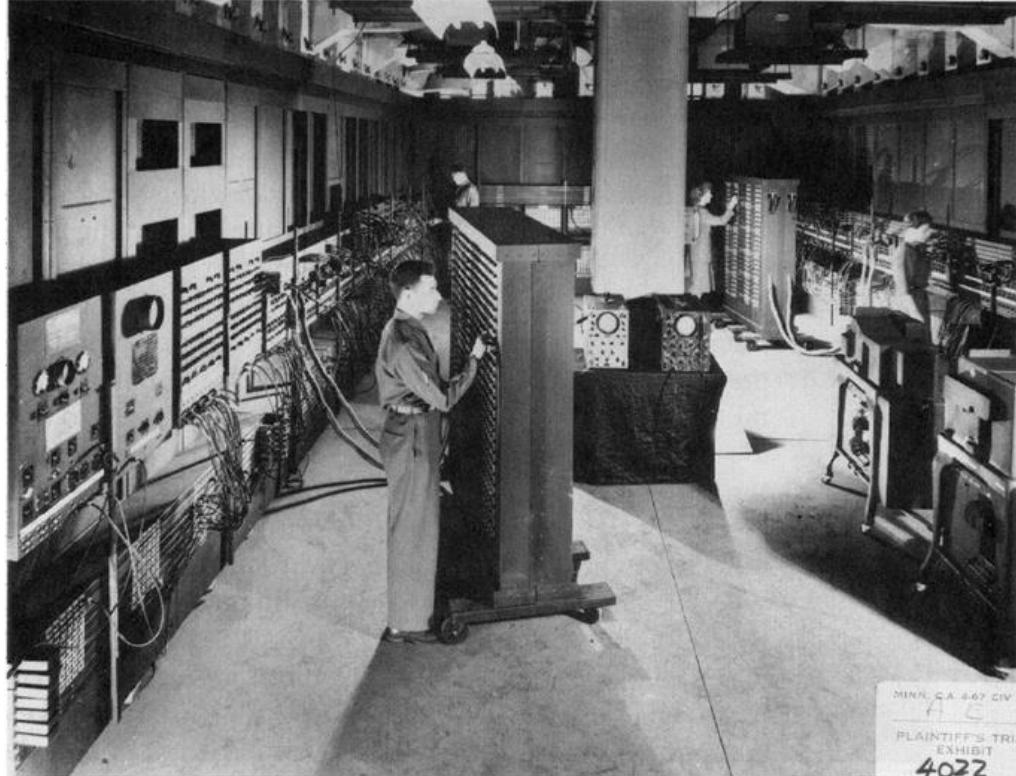
By Ezrdr - Own work, Public Domain,
<https://commons.wikimedia.org/w/index.php?curid=9510364>

Z1 electromech
computer (1938)



By ComputerGeek - de.wikipedia.org: 22:33, 27. Dez 2005 . . ComputerGeek (Diskussion)
. . 1037 x 778 (91664 Byte), CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=735841>

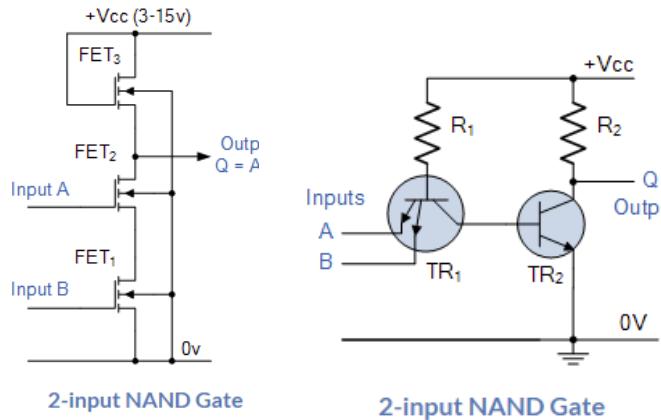
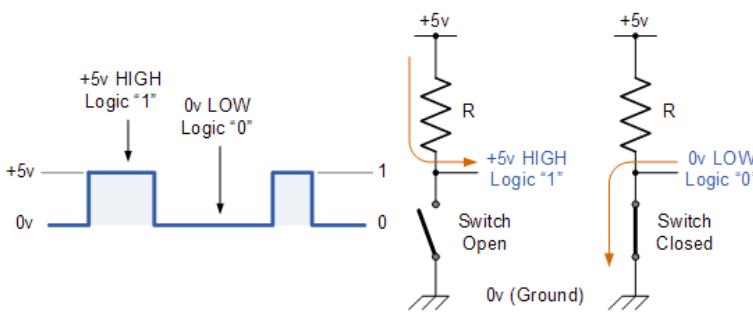
ENIAC – The first electronic computer 1946



- Operative in December 1945
- Used to calculate artillery tables.
- Equivalent to USD \$7E6 today.
- 20.000 vacuum tubes, 5E6 soldering hand made.
- 27 tons, 150kW

What is a Microprocessor?

- 1.- It is a device that can process and control data within a **single IC**.
- 2.- It contains a list of **program instructions** (dictionary) to perform **arithmetic operations**.
- 3.- It is **clock driven** and **register based**.
- 4.- The level of voltage within a microprocessor is translated to a **logical "1"** (>2.0V in TTL (transistor based) and >3.0 CMOS (MOSFET based)) or **logical "0"** (<0.8V in TTL and <1.5 CMOS).
- 5.- The simplest microprocessor contains **Arithmetic Logical Unit (ALU)**, **Registers** to save instructions (dictionary) and state of computation and data, and **buses** to move the information.

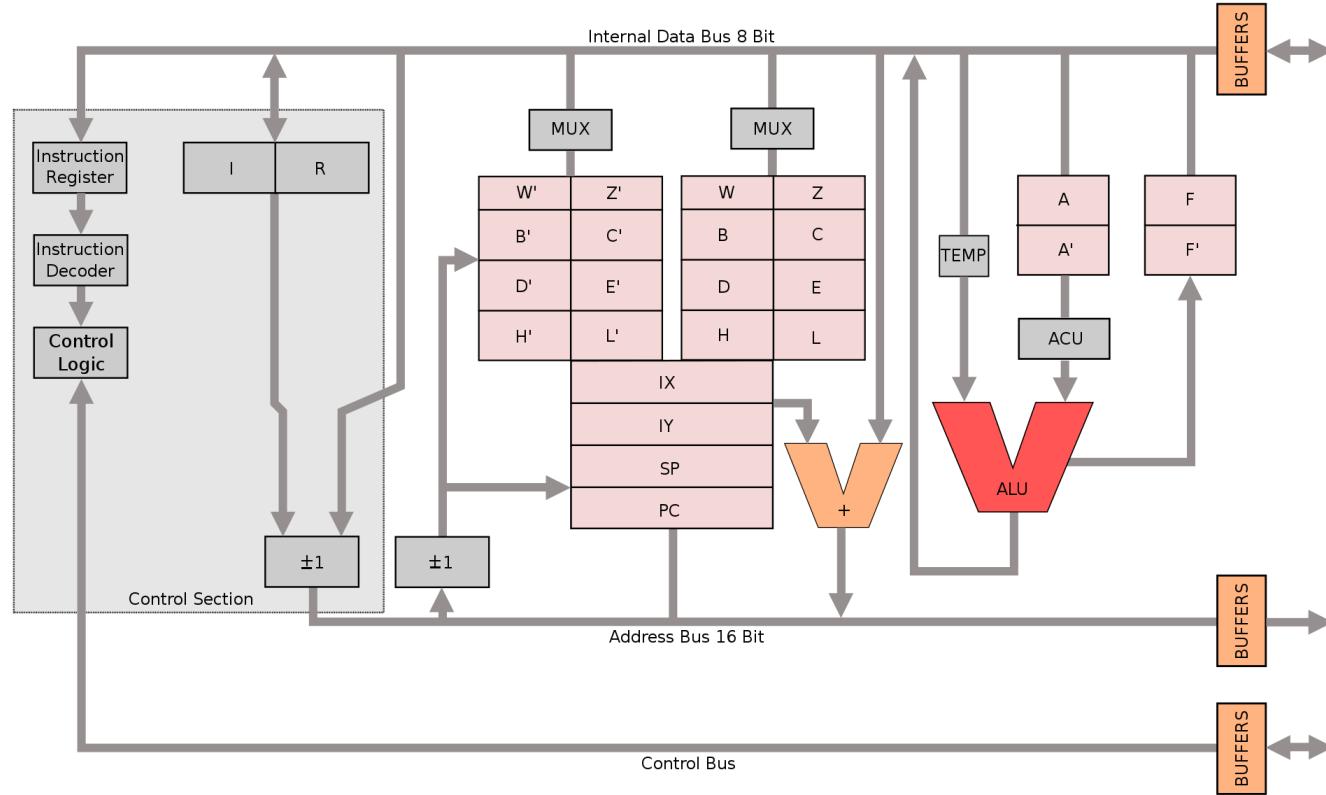


Link

Microprocessor

Basic Structure Zilog Z80

Z80 Architecture



[Link](#)

[More info](#)

Main Register Set		Alternate Register Set	
Accumulator	Flags	Accumulator	Flags
A	F	A'	F'
B	C	B'	B'
D	E	D'	E'
H	L	H'	L'

Interrupt Vector I	Memory Refresh R
Index Register IX	
Index Register IY	
Stack Pointer SP	
Program Counter PC	

General Purpose Registers

Special Purpose Registers

1.- **ALU:** Perform addition, subtractions, AND, OR operations.

2.- ALU sets flags in **status registers** to indicate state of the operation.

3.- The **control Logic** takes the code instructions from external memory and initiate a sequence of operations driven by ALU.

3.- **Word size** refers to the size of the data that can be moved within the buses. Started with **4-bits (0.5byte)** today **64bits (8 bytes)**. An 8 bit processor means that its words are 8bit length.

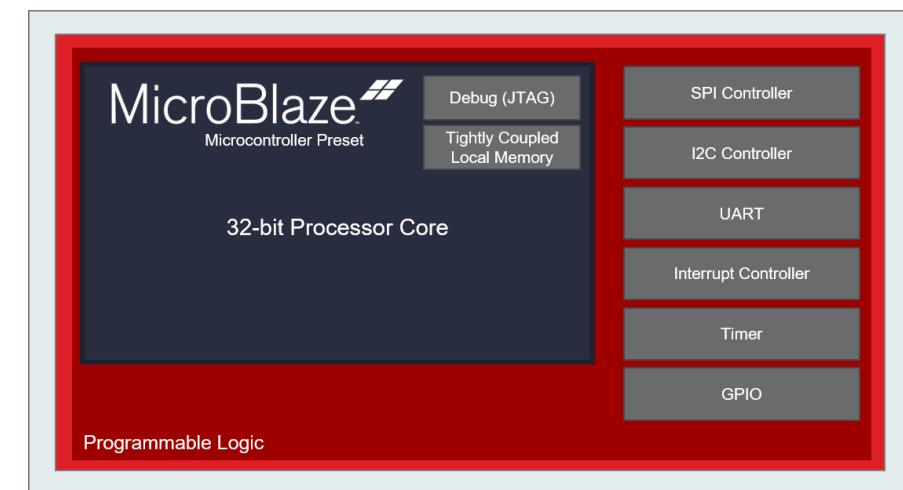
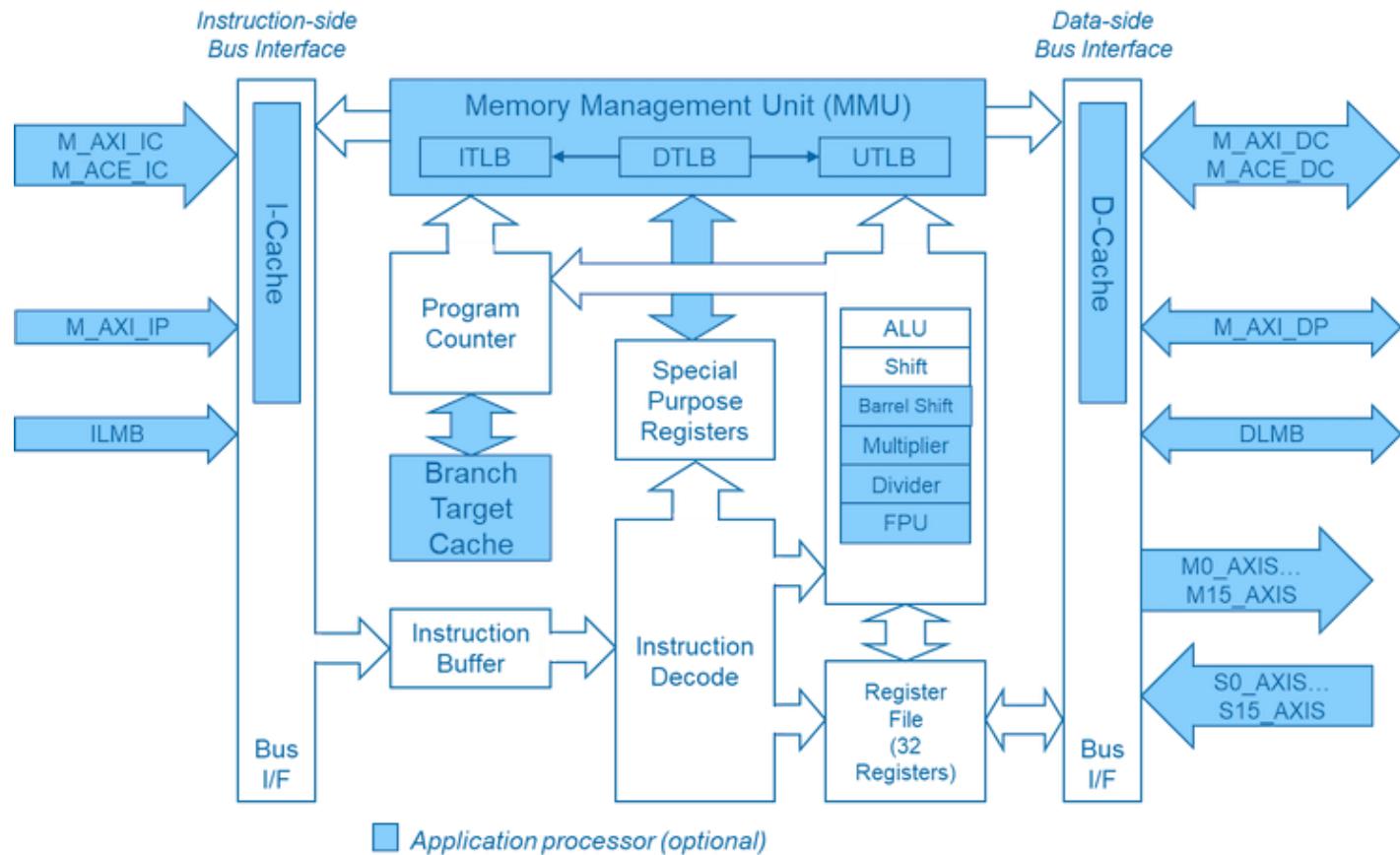
4.- **Floating point unit (FPU)** was introduced later to perform arithmetic calculation with floating point numbers.

5.- Today also internal memory, known as **cache memory**, is included to speed up the operation execution.

The simplest microprocessor contains:
Arithmetic Logical Unit (ALU)
Registers (special and general purpose)
Buses to move the information, address and control signals.

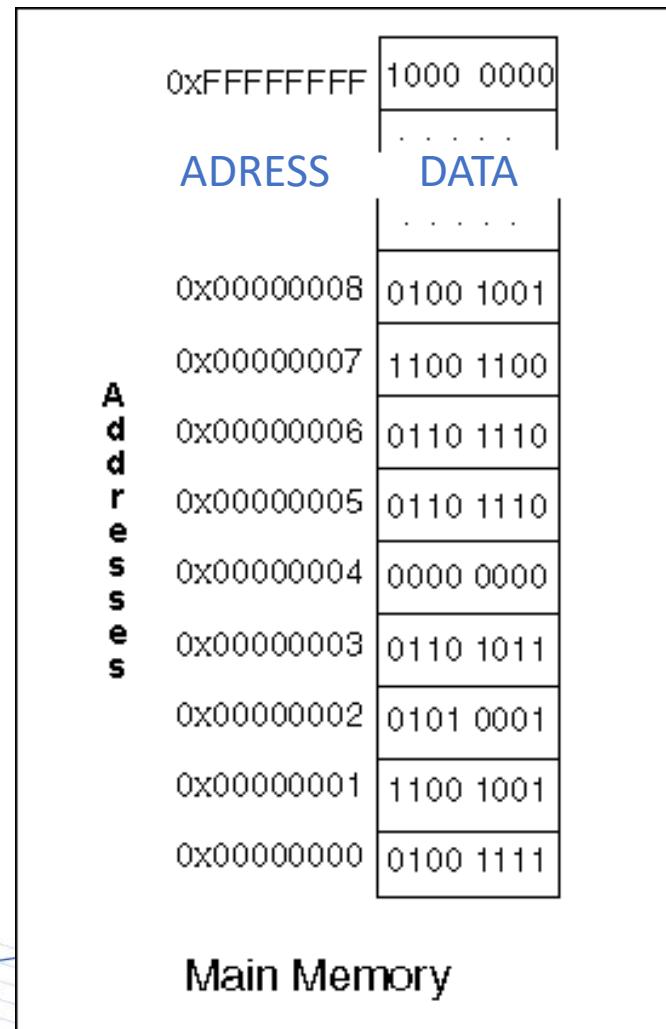
Microprocessor

Microblaze Xilinx



Addesses

Definition of a Memory

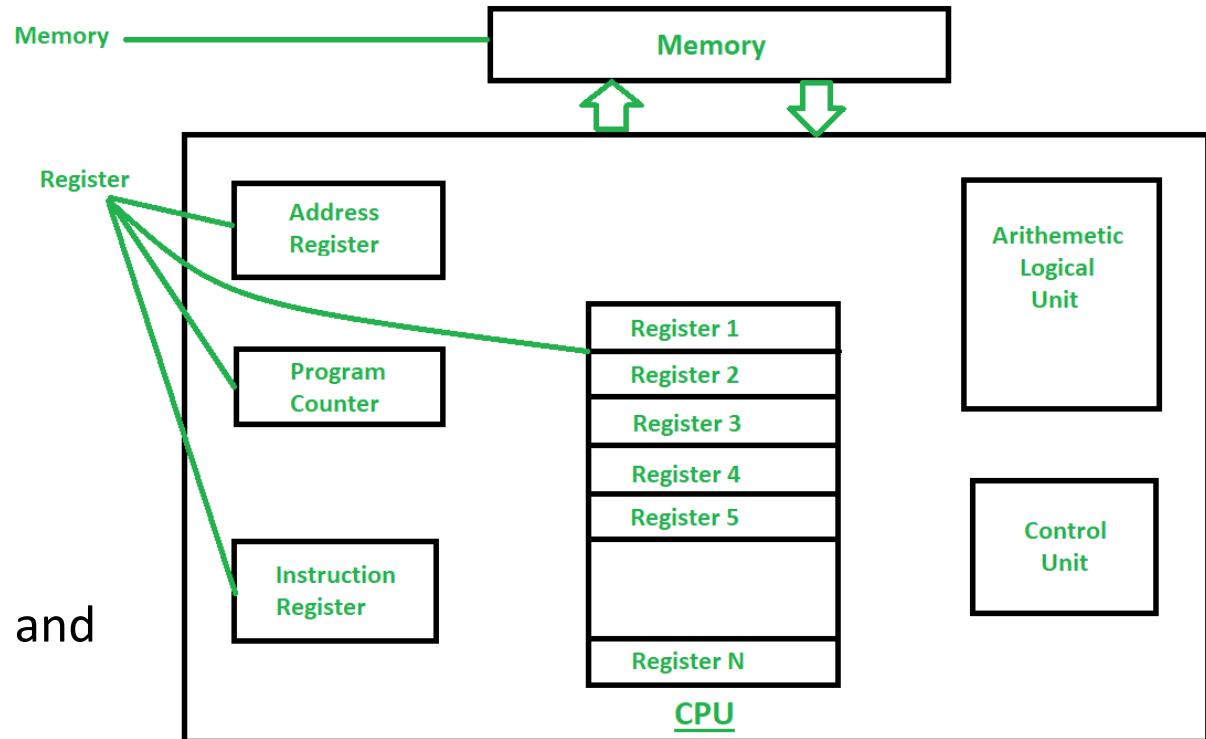


- Registers and instructions are data stored within processor memory.
 - Each memory is defined as “[word length](#)” (number of bits) and size (number of words).
 - Each word has a physical position. To identify this position the “[Memory Address](#)” is defined.
- To identify the address of a data within the memory we use its [address](#) (usually hexadecimal number).
- Each data contained in a specific memory address can be read or write.
- A variable in a program can be a memory address.

Registers of a Microprocessor

What is that?

- It is a **small piece of memory** within the processor with direct and very fast access.
 - Usually a processor has a **couple of dozen** of registers.
 - These are connected to the data bus.
- Registers contain **key information** for the operation and status of the processor.
- There are three types of registers (it is a nice simplification):
 - **General purpose** registers (to save user defined numbers). They can be fully read or written. Not intended for large data storage. They are key data with fast access.
 - **Input/Output (I/O) Registers**: control Access to memory and peripherals or the uP.
 - **Special Registers**: Usually give status of the uP.
 - Status register (status) and control, can not be fully written. Some bits are just readable to know if the processor is available or not for a certain operation.
- As any data, each register has a “memory address” (Address Register). Important to read it or write it!



Instructions of a Microprocessor

Buses

It provides interconnection among processor and other devices (memory, peripheral, other processors, etc).

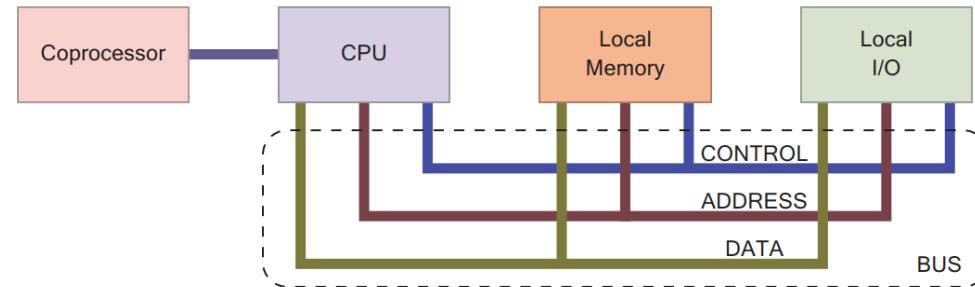
Its basic architecture is composed of:

1.-Adress bus: Transfer memory adress or target port identification number to I/O devices.

2.-Data bus: Transfer the data.

3.-Control bus: governs the control and timing signals, synchronising operations. Control signals: interrupt request and acknowledgements.

4.- Bus arbiter: Control the access of all connected modules to the bus. For simultaneously request from **bus masters**, it gives access to the one with highest priority, other waits for access (more info [here](#))



Bus Master: Can request access to the bus and is responsible for initiating the data transfer, provide the address and control signals.

Bus slave: Can not request access to the bus. It decode the control and address bus and when addressed can read or write the data bus.

Extra: In modern systems, with coprocessors and multiple cores, processor bus and system bus are used respectively.



SoC Zynq 7000 Device Overview

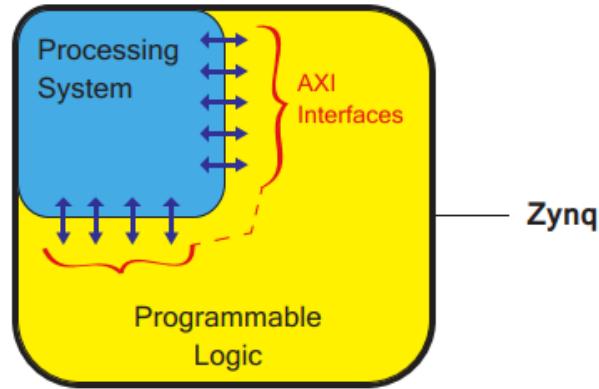
This section provides a brief overview of
the Xilinx Zynq Device



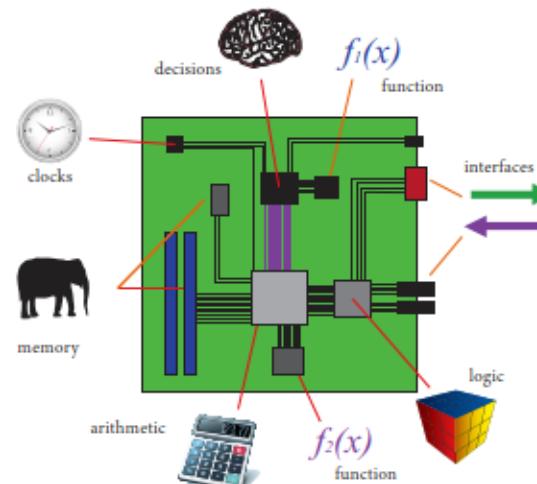
What is a SoC?

- Hardware:** It combines **Programable Logic (PL)** and **Processing System (PS)** units together with all required complementary digital system (memory, buses, etc) in a single chip.
- Integration:** It is the highest level of integration in a Programmable Electronic System. Avoiding PCB and its interconnection (system-on-board)

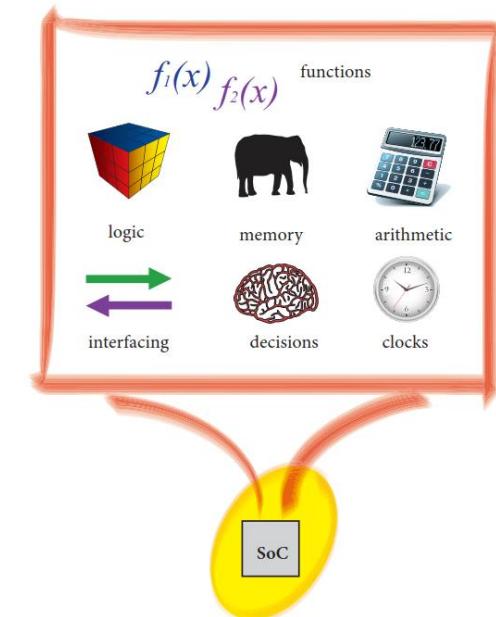
Zynq SoC Architecture



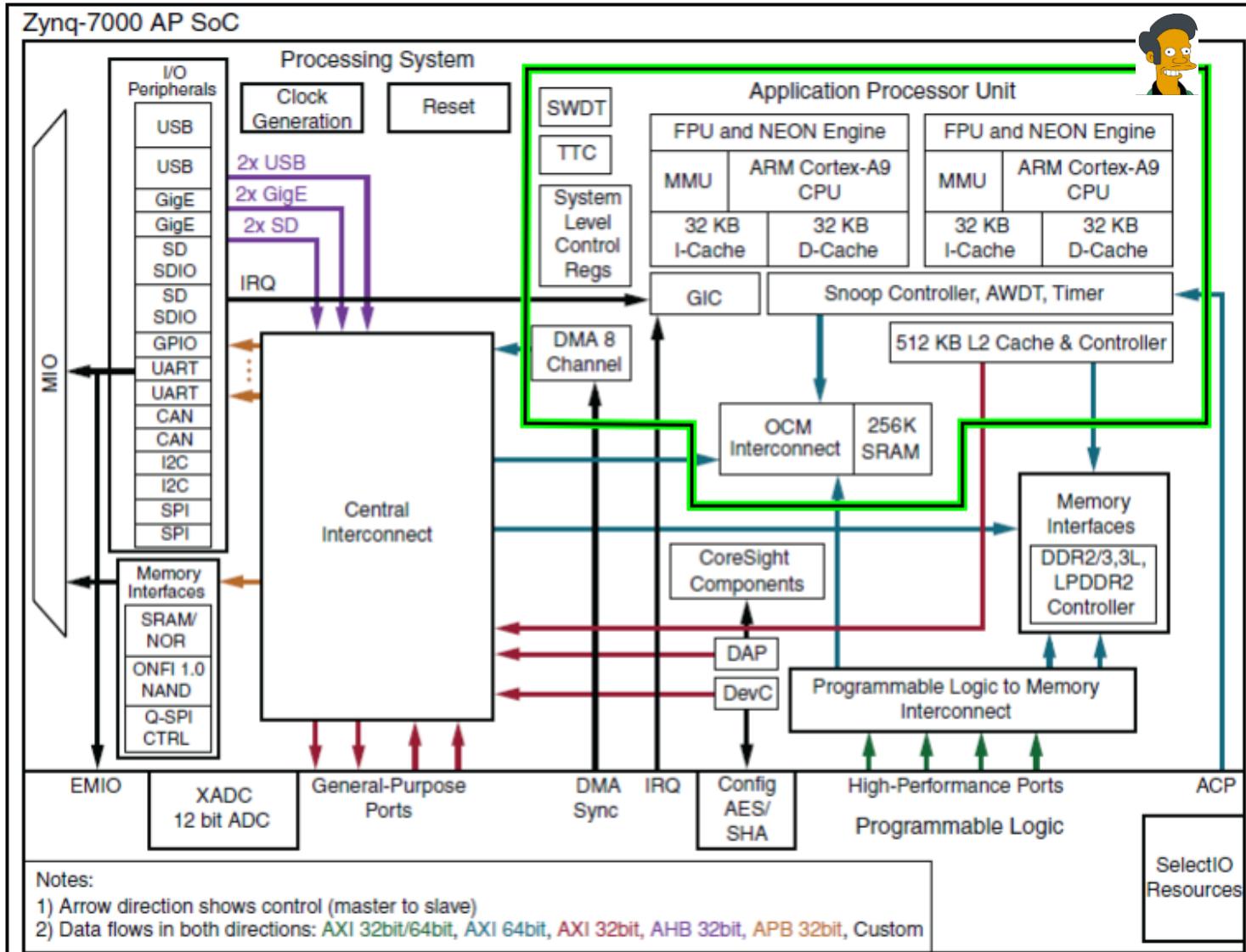
System-on-board



System-on-chip



Design Flow on Zynq SoC



Application Processing Unit (APU): Contains the ARM and other processing resources and peripheral. Another term for referring to a Microprocessor (or part of it).

PS: Mainly contains APU, Memory interfaces, I/O peripherals and connection to PL (central interconnect).

PL: Besides logic gates, contains 12bits XADC for analog to digital conversion and connection to PS (ACP, HP ports).

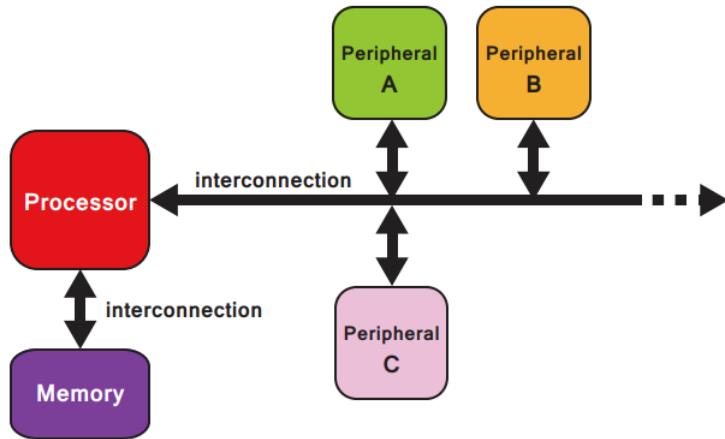
There is a need of internal buses architecture to interconnect all components of a SoC, some are very fast and other rather slow. **How to do that? -> AXI (will see later)**

Technical Reference Manual Zynq-SoC [here](#).

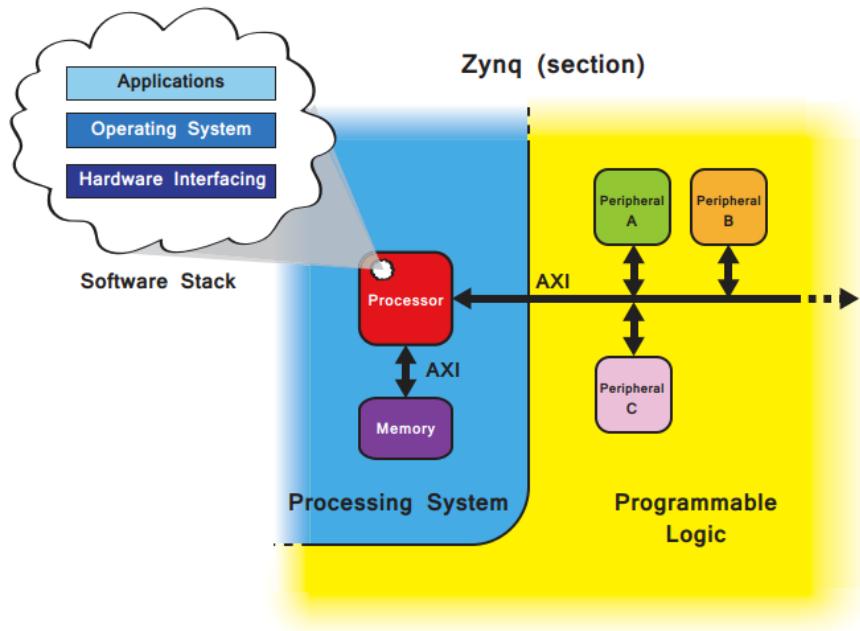
Important Terms

- **ASICs:** Application Specific Integrated Circuit (ADC, DAC, RF application ICs, etc.). First elements **knows as SoCs.**
- **ASIC-based SoCs:** It represents a system on board.
- **System-on-Programmable-Chip:** A specific type of SoC which can be reconfigured or reprogrammable (PLD).
- **All-Programmable SoC (APSoC):** Programmable SoC composed of PS and PL (equivalent to FPGA).
- **Why is a APSoC needed at all?:** Low cost compared to ASIC-based SoC, rapid prototyping, more flexible, short time-to-market, avoiding hardware problems with communications.
- **Zynq-SoC:** A family of APSoC designed and manufactured by the company Xilinx.

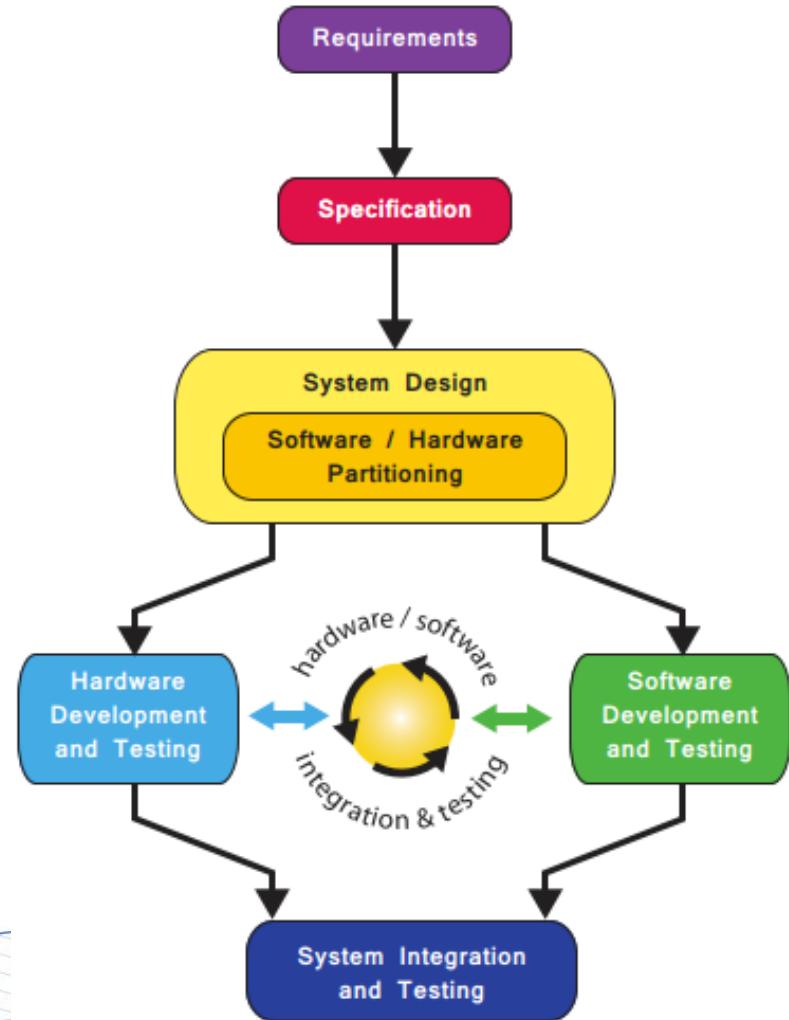
Simple Anatomy of Embedded SoC



A Zynq Embedded SoC.

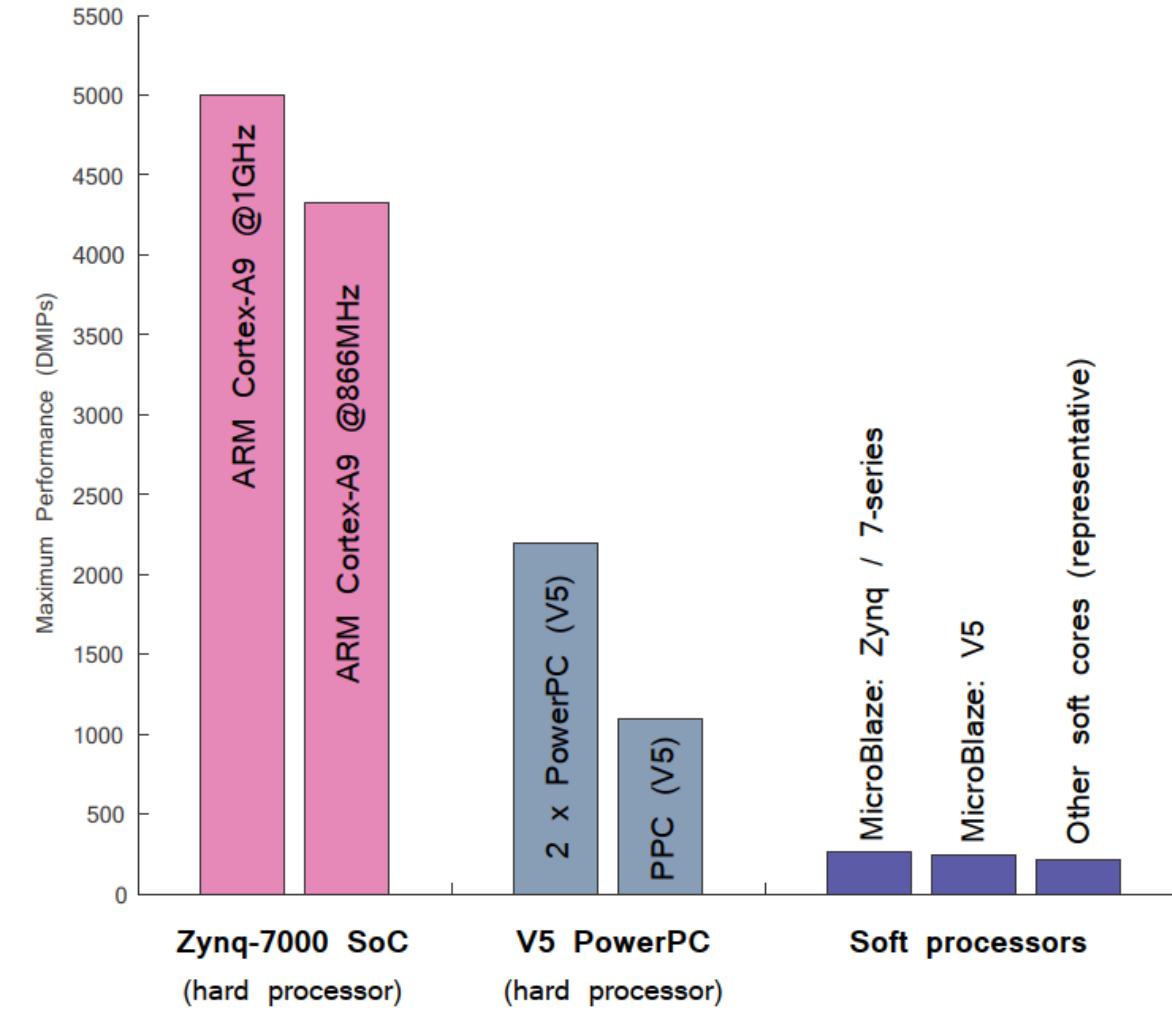


- Processor is the central element of the system.
- Communication between system elements is via interconnection
- Interconnection can be direct, point-to-point or buses (using protocols to Access the bus).
- Peripherals are functional components residing away from the processor, and in general these perform one of three functions:
 - (i) coprocessors — elements that supplement the primary processor, usually optimized for a certain task;
 - (ii) cores for interacting with external interfaces, e.g. connecting to LEDs and switches, codecs, etc.
 - (iii) additional memory elements.
- PS: Fixed architecture. Processor system memory.
- PL: Flexible and programmable architecture. **It can be memory, processor or anything you like =)**
- AXI: Is the interface linking PS and PL.



- Reuse: As programmer always think on blocks that can be re-used.
- IP-Cores: Intellectual property functional blocks speed-up time-to-market process (peripheral). Xilinx provide libraries, open-source libraries or third parties. **DO NOT RE-INVENT THE WHEEL.**
- PS is programmed in C code. (software development)
- PL is programmed in VHDL. (Hardware development)
- PL can be programmed in C using *High Level Synthesis (HDL)* tool from Xilinx. This avoid the programmer to learn VHDL, but he/she relies on Xilinx tool, no knowing what is the real source code. **In this course you must learn VHDL!**
- Partitioning process: Here the programmer assign the task to PS and PL.

Why a SoC and not just FPGA with soft processor?



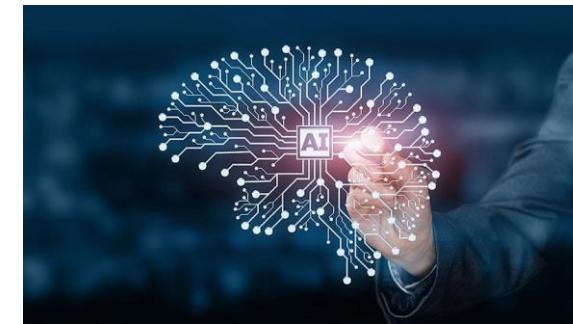
What can Zynq do for me that an FPGA with a soft processor can't?

- Processors performance is measured today in DMIPS.
- DMIPS ([Dhrystone Millions of Instructions Per Secons](#)): Numbers of MIPS when running a specific test named Dhrystone. This test stress the processor to execute several operations.
- A soft processor can not compete with hard processor regarding performance.
- PowerPC is a processor manufacturer (see [here](#))



SoC Applications

Overview



Motivación-What can I do with a SoC?



Advanced Assistance System (ADAS): Manages driving safety systems, e.g., traffic sign recognition, lane departure warning (LDWS), parking assistance, driver attention monitoring, etc. Brands: Mercedes-Benz, BYD.



Communications: Management of data-intensive systems (FPGAs). Applications such as radar, satellite transmission, GPS, and all possible communication systems. (NASA)

Motivación-What can I do with a SoC?

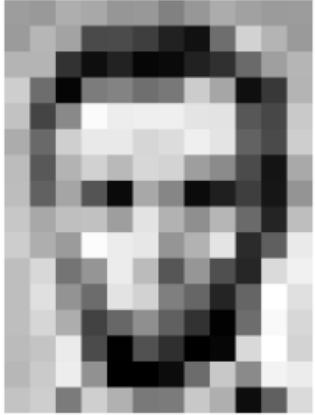


Defense and Aerospace: Communications, image processing, navigation. Aircraft electrification, on-board flight systems, satellite and ground communication. (radar, aircraft, missiles, space)



Power Electronics: Programming control algorithms for managing large amounts of energy. Real-time processing. Motor control, photovoltaic plants, wind farms, etc.

Motivación-What can I do with a SoC?



157	153	174	168	150	162	129	151	172	161	155	156
155	182	163	74	75	62	39	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	197	251	237	239	239	228	227	87	71	201
172	108	207	233	233	214	220	230	228	98	74	206
188	88	179	209	185	215	211	158	199	78	20	169
189	97	165	64	10	168	134	11	31	62	22	148
199	168	191	193	158	227	179	149	182	106	36	190
205	174	155	262	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	168	227	211	127	120	94	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	39	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	197	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	230	228	98	74	206
188	88	179	209	185	215	211	158	199	78	20	169
189	97	165	64	10	168	134	11	31	62	22	148
199	168	191	193	158	227	179	149	182	106	36	190
205	174	155	262	236	231	149	178	228	43	95	234
190	216	116	148	236	187	86	150	79	38	218	241
190	224	147	168	227	210	127	102	96	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218



© Xilinx



© Xilinx

Image and Video Processing: Parallelization of data processing, broadcast, video compression, surveillance, embedded vision (requires PL and PS)

Medical Applications: Computer Tomography (CT) scanners, ultrasound and Magnetic Resonance Imagers (MRIs). Also an application of image processing.

<https://www.xilinx.com/about/xcell-publications/xcell-journal.html>

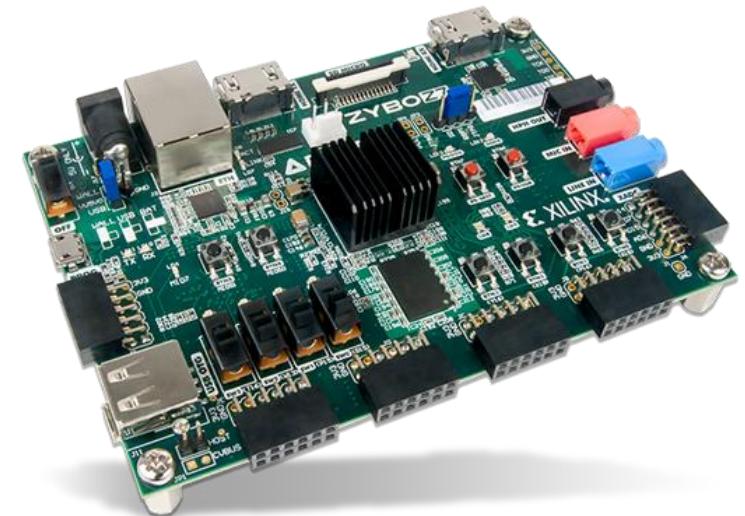


Personal Study



Zybo Z7 Overview

This section provides a brief overview of the ZYBO Z7 platform.



- **ZYNQ Processor**

- 667 MHz dual-core Cortex-A9 processor
- High-bandwidth peripheral controllers: 1G Ethernet, USB 2.0, SDIO
- Low-bandwidth peripheral controllers: SPI, UART, CAN, I2C
- Programmable logic equivalent to Artix-7 FPGA

- **Memory**

- 1 GB DDR3L with 32-bit bus @ 1066 MHz
- microSD slot

- **Audio and Video**

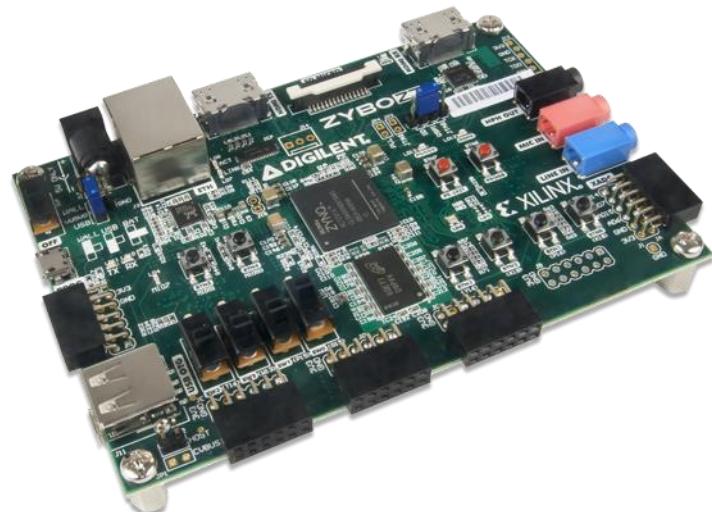
- Pcam camera connector with MIPI CSI-2 support
- HDMI sink port (input) with/without* CEC
- HDMI source port (output) with CEC
- Audio codec with stereo headphone, stereo line-in, and microphone jacks

- **Switches, Push-buttons, and LEDs**

- 6 push-buttons (2 processor connected)
- 4 slide switches
- 5 LEDs (1 processor connected)
- 2 RGB LEDs (1*)

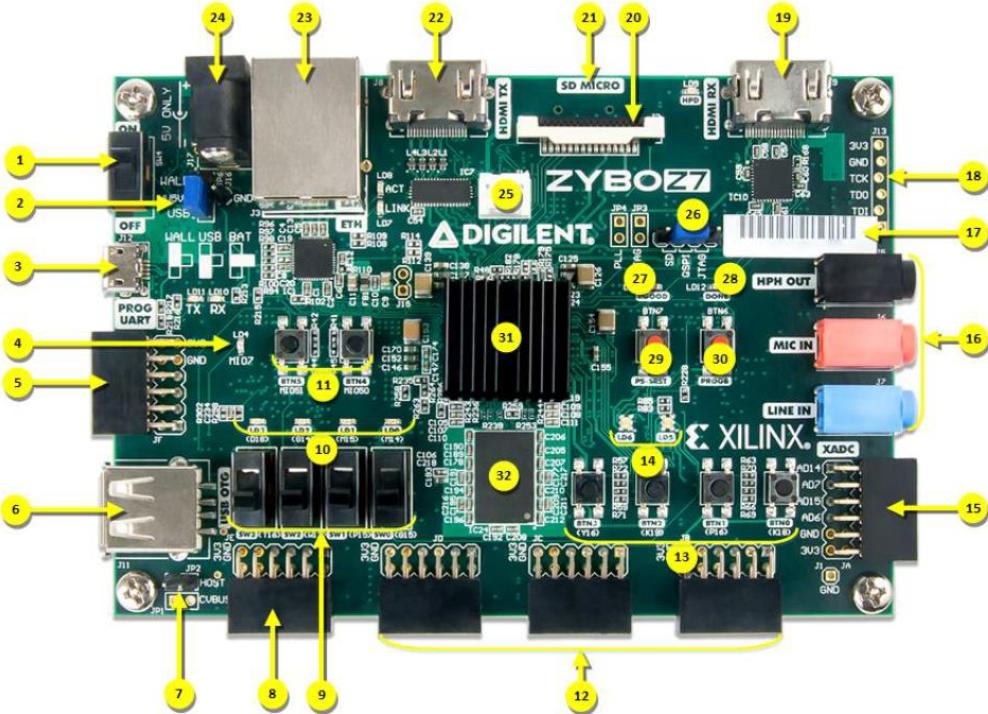
- **Expansion Connectors**

- 6 Pmod ports (5*)
- 8 Total Processor I/O
- 40 Total FPGA I/O (32*)
- 4 Analog capable 0-1.0V differential pairs to XADC



Plataforma de Programación

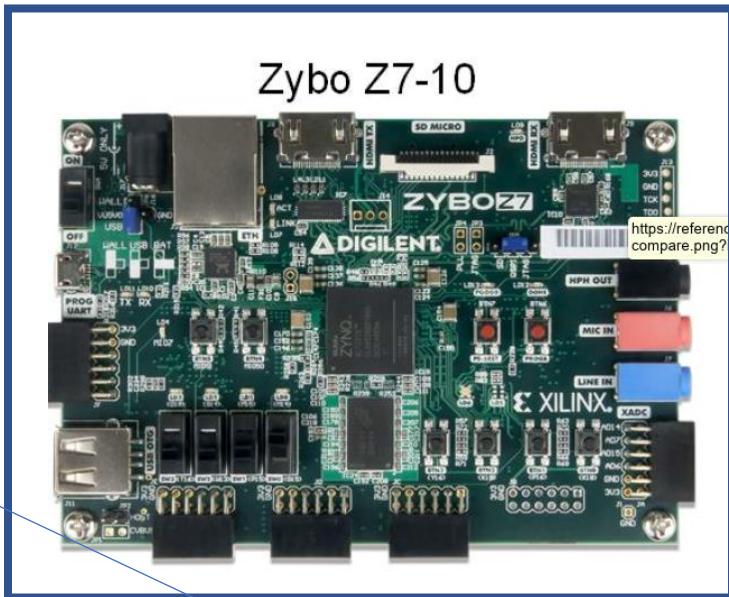
Zybo Z7



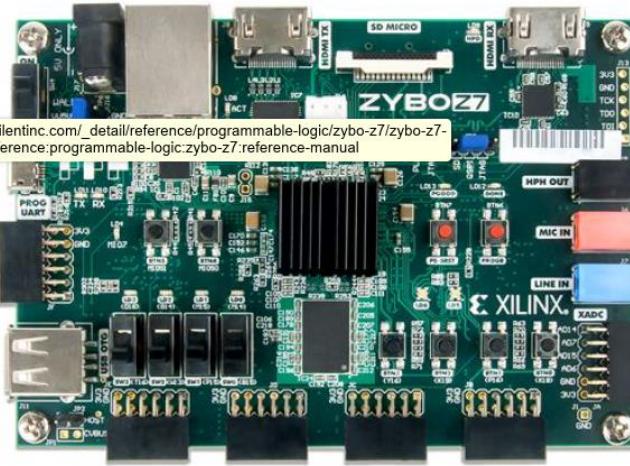
Callout	Description	Callout	Description
1	Power Switch	17	Unique MAC address label
2	Power select jumper	18	External JTAG port
3	USB JTAG/UART port	19	HDMI input port
4	MIO User LED	20	Pcam MIPI CSI-2 port
5	MIO Pmod port	21	microSD connector (other side)
6	USB 2.0 Host/OTG port	22	HDMI output port
7	USB Host power enable jumper	23	Ethernet port
8	Standard Pmod port	24	External power supply connector
9	User switches	25	Fan connector (5V, three-wire) *
10	User LEDs	26	Programming mode select jumper
11	MIO User buttons	27	Power supply good LED
12	High-speed Pmod ports *	28	FPGA programming done LED
13	User buttons	29	Processor reset button
14	User RGB LEDs *	30	FPGA clear configuration button
15	XADC Pmod port	31	Zynq-7000
16	Audio codec ports	32	DDR3L Memory

* denotes difference between Z7-10 and Z7-20.

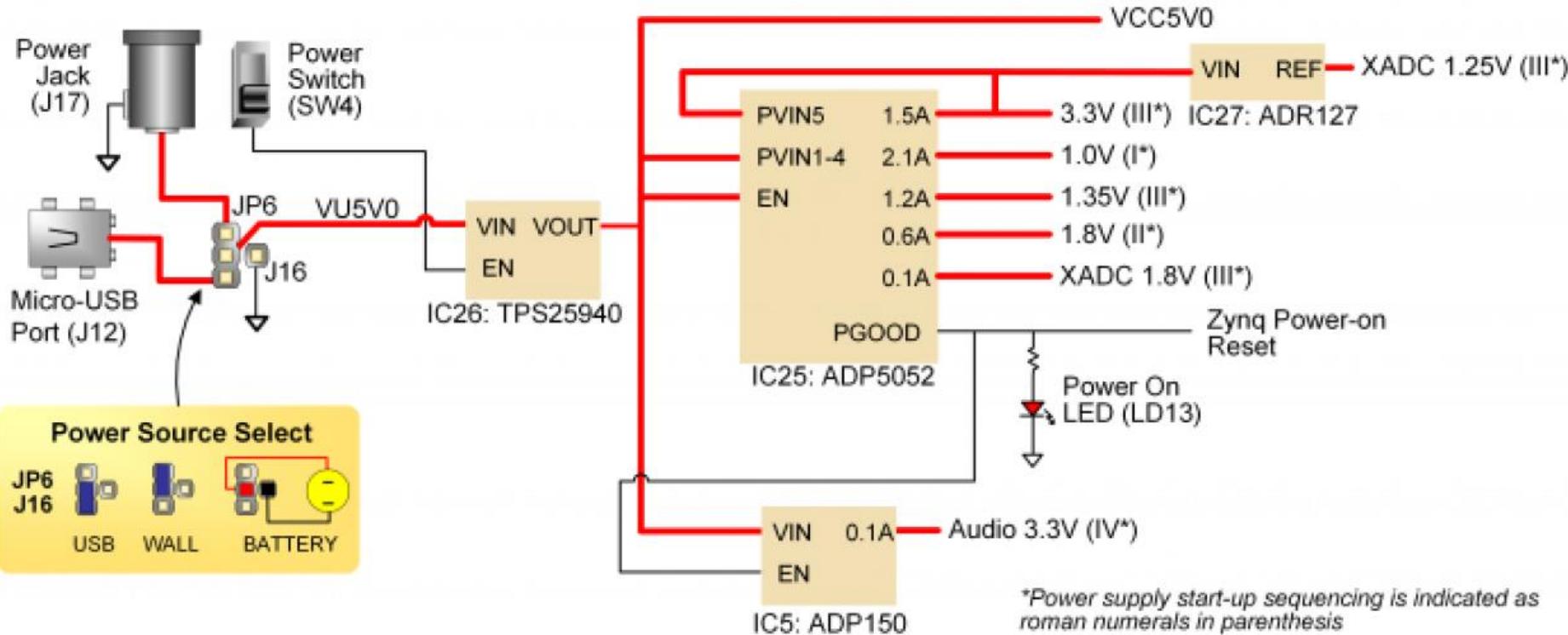
We use this!



Zybo Z7-20

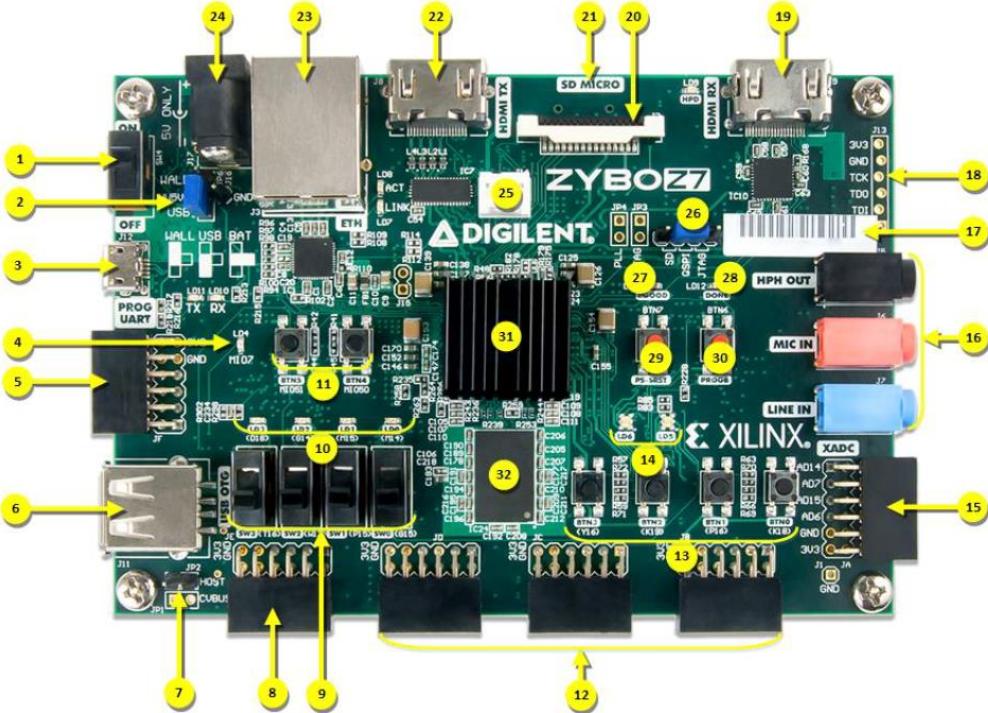


Product Variant	Zybo Z7-10	Zybo Z7-20
Zynq Part	XC7Z010-1CLG400C	XC7Z020-1CLG400C
1 MSPS On-chip ADC	Yes	Yes
Look-up Tables (LUTs)	17,600	53,200
Flip-Flops	35,200	106,400
Block RAM	270 KB	630 KB
Clock Management Tiles	2	4
Total Pmod ports	5	6
Fan connector	No	Yes
Zynq heat sink	No	Yes



Plataforma de Programación

Zybo Z7



Callout	Description	Callout	Description
1	Power Switch	17	Unique MAC address label
2	Power select jumper	18	External JTAG port
3	USB JTAG/UART port	19	HDMI input port
4	MIO User LED	20	Pcam MIPI CSI-2 port
5	MIO Pmod port	21	microSD connector (other side)
6	USB 2.0 Host/OTG port	22	HDMI output port
7	USB Host power enable jumper	23	Ethernet port
8	Standard Pmod port	24	External power supply connector
9	User switches	25	Fan connector (5V, three-wire) *
10	User LEDs	26	Programming mode select jumper
11	MIO User buttons	27	Power supply good LED
12	High-speed Pmod ports *	28	FPGA programming done LED
13	User buttons	29	Processor reset button
14	User RGB LEDs *	30	FPGA clear configuration button
15	XADC Pmod port	31	Zynq-7000
16	Audio codec ports	32	DDR3L Memory

* denotes difference between Z7-10 and Z7-20.



Electrical Engineering Department
Pontificia Universidad Católica de Chile
peclab.ing.uc.cl