Actuating, Sensing and Control Mechatronic Systems

Grzegorz Góra PhD

D1-Lab 20

ggora@agh.edu.pl

http://home.agh.edu.pl/~ggora/

Department of Robotics and Mechatronics

Faculty of Mechanical Engineering and Robotics

AGH University of Science and Technology

AGENDA

Part I

1. FPGA - architecture of the system
2. Comparison of MCU and FPGA
3. FPGA – Ways of use
4. FPGA Applications
5. DE10-Lite - Development and Education Board
6. Example projects implemented using FPGA

Part II – Combinational circuits

CONDITIONS FOR PASSING THE COURSE

Passing the course – pass grade from laboratory, project, and exam:

laboratory (max 60 points)
+
project (max 40 points)
+
exam (motors, electronics, sensors).

Passing the laboratory and the project parts - based on the number of points. Grading scale:

- (0%-49%) - **2.0**
- (50%-60%) - **3.0**
- (61%-70%) - **3.5**
- (71%-80%) - **4.0**
- (81%-90%) - **4.5**
- (91%-100%) - **5.0**

Currently achieved points are visible on (website of Grzegorz Karpiel PhD, tab Dydaktyka):
http://home.agh.edu.pl/~gkarpiel/studenci.html

A point correction in case of an excused absence.

Final grade:

$$G_F = 0.35 \cdot G_{Lab} + 0.25 \cdot G_{Pro} + 0.4 \cdot G_{Exam}$$

G_{Lab}, G_{Pro} – final grades from laboratory and project;

G_{Exam} – arithmetic mean from parts: motors, electronics, and sensors.

COURSE ISSUES

Motors

- direct current motors (DC);
- stepper motors;
- linear motors;
- direct drives (BLDC, PMSM);
- synchronous motors AC;
- asynchronous motors AC;
- commutated motors AC;
- piezoelectric transducers;
- drivers and amplifiers;
- pneumatic and hydraulic actuators.

Electronics

- logic gates and flip-flops;
- combinational logic circuits design;
- sequential logic circuits design;
- microprocessors (construction and architecture);
- implementation of algorithms in FPGA;
- soft-processor Nios II.

Sensors

- sensors of non-electrical quantities (classic, MEMS);
- sensors in mechatronic systems;
- sensors as mechatronic systems.

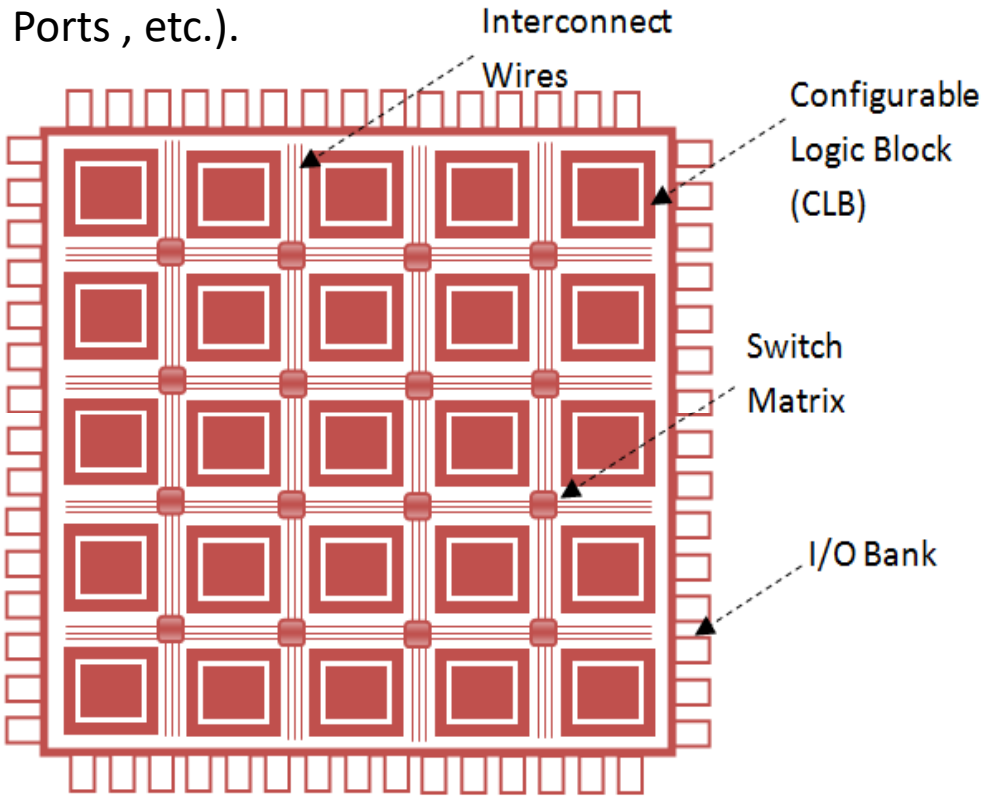
FPGA
(Field Programmable Gate Arrays)

The main components of FPGA devices:

- **LEs cells** *(Logic Elements) / **ALM** (Adaptive Logic Modules) / **CLBs** (Configurable Logic Blocks)*;
- **connection network** – provides local communication with neighboring LEs and global communication within the entire FPGA chip;
- **additional elements** (e.g. memory, multipliers, PLL loops, ADCs , etc.);
- **others** (power system, GPIO - General Input/Output Ports , etc.).

Differences between FPGA devices (and devices' families):

- number of LEs (*Logic Elements*);
- complexity level of LEs;
- device speed (maximum operating frequency, expressed as signal propagation time through LEs ~ 1-9 ns);
- the type and number of integrated equipments (memory, multipliers, PLL loops, ADCs, etc.);
- number of input/output ports - a type of footprint;
- version (commercial/industrial - depending on the operating temperature range).



FPGA
(Field Programmable Gate Arrays)

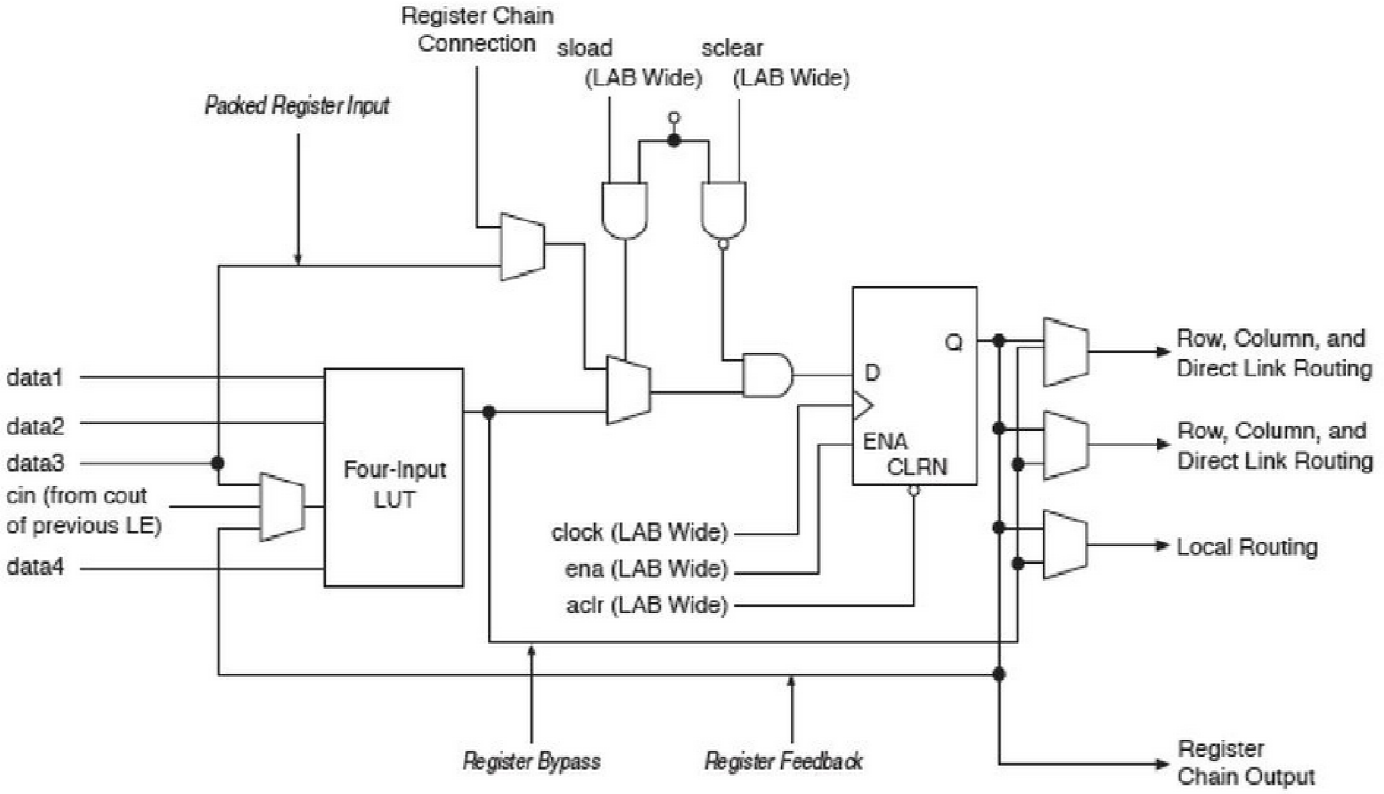
LEs cell:

• is based on a 4-bit Look Up Table (the 4 bit LUT allows to implement of any 4-bit boolean logic);

• the flip-flop allows to latch a signal (design sequential logic circuit);

• has additional elements for the internal configuration and signals multiplexing.

Altera Cyclone III
Logic Element (LE) – Normal Mode



FPGA vs. Microcontroller (MCU): Similarities and differences

	MCU 	FPGA 
Differences	▪ Fixed architecture imposed by the manufacturer ▪ A programming language is used as sequentially executed commands ▪ Multiple tasks are realized by switching between them ▪ Fixed, dedicated I/O ports, little possibility of changing functionality: e.g. PWM, I2C, SPI ▪ Built-in ADC and DAC converters	▪ User-configurable architecture ▪ A programming language used for internal device configuration ▪ Tasks are realized parallelly and simultaneously ▪ User-configurable I/O ports, high possibility of changing a functionality ▪ Most often without built-in ADC and DAC converters
Similarities	**▪ The digital circuit is in the form of an integrated circuit** **▪ Can realize implemented algorithms/tasks** **▪ Programmable by a dedicated language**	

FPGA vs. Microcontroller (MCU): Similarities and differences

	MCU 	FPGA 
Similarities	▪ **The digital circuit is in the form of an integrated circuit** ▪ **Can realize implemented algorithms** possibility of implementation algorithms: regulation and control systems, numerical calculations, communication with sensors and transducers by standard low-level interfaces (e.g. SPI, I2C, 1-Wire), communication with peripheral devices and systems (e.g. displays, drivers of electric motors) communication via standard interfaces (UART, Ethernet), etc. ▪ **Programmable by a dedicated language** C / C++ / asembler –MCU; VHDL / (System) Verilog – FPGA.	

FPGA vs. Microcontroller (MCU): Similarities and differences

	MCU 	FPGA 
Differences	▪ **Fixed architecture imposed by the manufacturer** ▪ A programming language is used as sequentially executed commands ▪ Multiple tasks are realized by switching between them ▪ Fixed, dedicated I/O ports, little possibility of changing functionality: e.g. PWM, I2C, SPI ▪ Built-in ADC and DAC converters	▪ **User-configurable architecture** ▪ A programming language used for internal device configuration ▪ Tasks are realized parallelly and simultaneously ▪ User-configurable I/O ports, high possibility of changing functionality ▪ Most often without built-in ADC and DAC converters
Similarities	▪ The digital circuit is in the form of an integrated circuit ▪ Can realize implemented algorithms ▪ Programmable by a dedicated language	

FPGA vs. Microcontroller (MCU): Similarities and differences

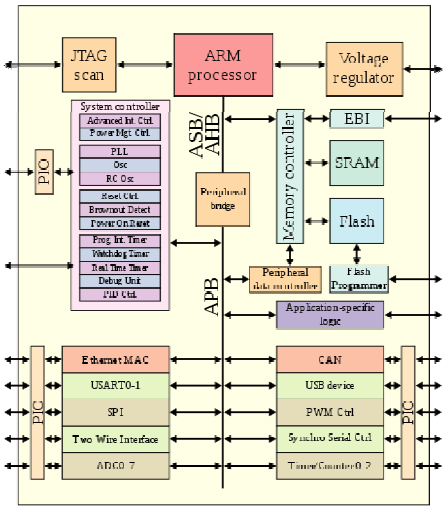
Differences

MCU

Fixed architecture imposed by the manufacturer

+ available modules/peripherals that are ready to work after simple configuration;

- functionality limited to components built-in by the manufacturer, user cannot add or change any elements of the architecture.



FPGA

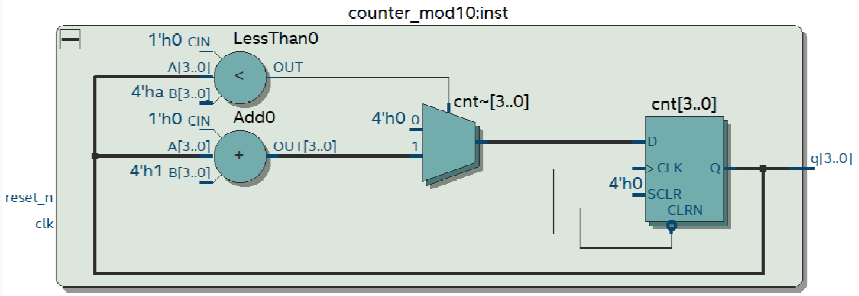
User-configurable architecture

+ the ability to implement any module;

- usually a longer implementation time.

Physical architecture : LEs, connection network, etc.

Hardware architecture: defined by the user, e.g. mod 10 counter:

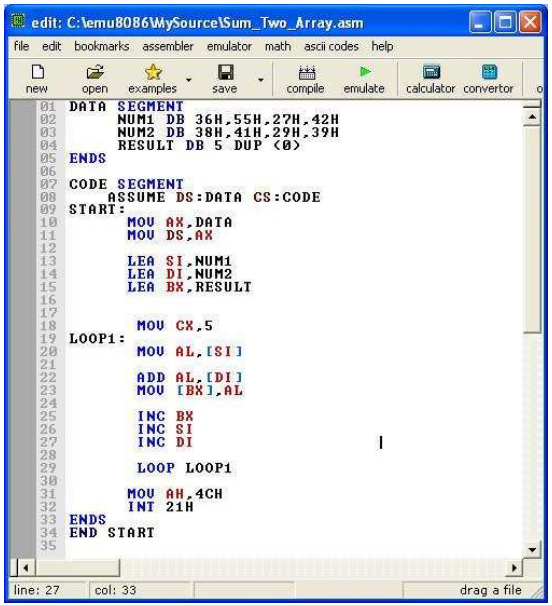


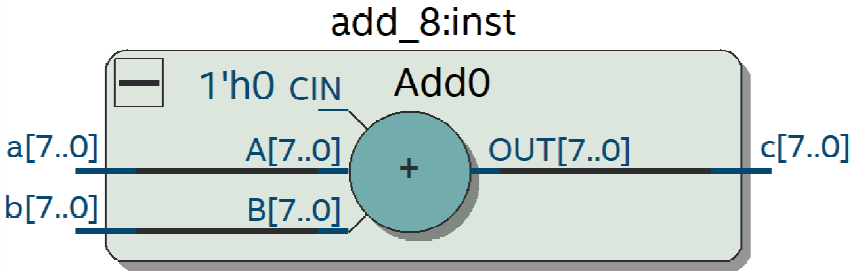
FPGA vs. Microcontroller (MCU): Similarities and differences

	MCU 	FPGA 
Differences	▪ Fixed architecture imposed by the manufacturer ▪ **A programming language is used as sequentially executed commands** ▪ Multiple tasks are realized by switching between them ▪ Fixed, dedicated I/O ports, little possibility of changing functionality: e.g. PWM, I2C, SPI ▪ Built-in ADC and DAC converters	▪ User-configurable architecture ▪ **A programming language used for internal device configuration** ▪ Tasks are realized parallelly and simultaneously ▪ User-configurable I/O ports, high possibility of changing functionality ▪ Most often without built-in ADC and DAC converters
Similarities	▪ The digital circuit is in the form of an integrated circuit ▪ Can realize implemented algorithms ▪ Programmable by a dedicated language	

FPGA vs. Microcontroller (MCU): Similarities and differences

	MCU 	FPGA 
Differences	A programming language is used as sequentially executed commands c = a + b; // C language	A programming language used for internal device configuration c <= a + b; // VHDL language

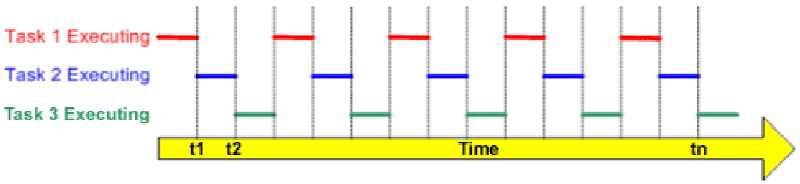
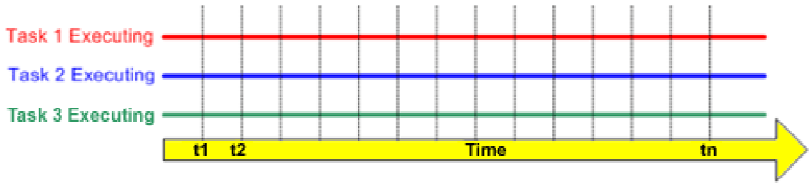




FPGA vs. Microcontroller (MCU): Similarities and differences

	MCU 	FPGA 
Differences	▪ Fixed architecture imposed by the manufacturer ▪ A programming language is used as sequentially executed commands ▪ **Multiple tasks are realized by switching between them** ▪ Fixed, dedicated I/O ports, little possibility of changing functionality: e.g. PWM, I2C, SPI ▪ Built-in ADC and DAC converters	▪ User-configurable architecture ▪ A programming language used for internal device configuration ▪ **Tasks are realized parallelly and simultaneously** ▪ User-configurable I/O ports, high possibility of changing functionality ▪ Most often without built-in ADC and DAC converters
Similarities	▪ The digital circuit is in the form of an integrated circuit ▪ Can realize implemented algorithms ▪ Programmable by a dedicated language	

FPGA vs. Microcontroller (MCU): Similarities and differences

	MCU 	FPGA 
Differences	**Multiple tasks are realized by switching between them** ▪ each task is allocated a specific time slot; ▪ increasing the number of tasks have an impact on the handling time of the other tasks; ▪ usually, necessity to given priorities to tasks; ▪ some problems in real-time systems with the keeping a constant and high frequency of operations. 	**Tasks are realized parallelly and simultaneously** ▪ different part of the hardware is responsible for each task; ▪ adding a new task does not affect the other tasks; ▪ the only consequence of adding a new task is using an additional FPGA resources; ▪ often the user has the ability to optimize algorithms in terms of speed (work frequency) or hardware resources. 

FPGA vs. Microcontroller (MCU): Similarities and differences

	MCU 	FPGA 
Differences	▪ Fixed architecture imposed by the manufacturer ▪ A programming language is used as sequentially executed commands ▪ Multiple tasks are realized by switching between them ▪ **Fixed, dedicated I/O ports, little possibility of changing functionality: e.g. PWM, I2C, SPI** ▪ Built-in ADC and DAC converters	▪ User-configurable architecture ▪ A programming language used for internal device configuration ▪ Tasks are realized parallelly and simultaneously ▪ **User-configurable I/O ports, high possibility of changing functionality** ▪ Most often without built-in ADC and DAC converters
Similarities	▪ The digital circuit is in the form of an integrated circuit ▪ Can realize implemented algorithms ▪ Programmable by a dedicated language	

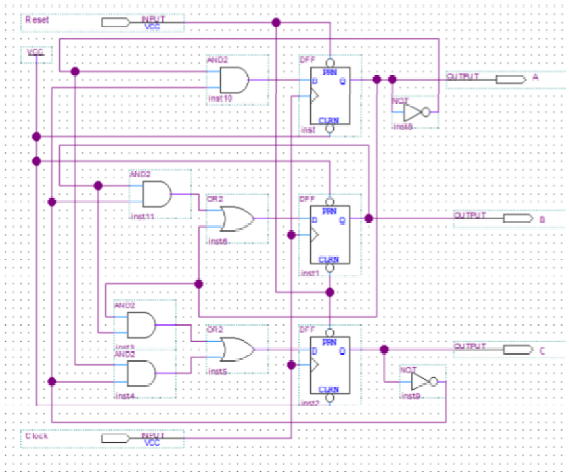
FPGA vs. Microcontroller (MCU): Similarities and differences

	MCU 	FPGA 
Differences	▪ Fixed architecture imposed by the manufacturer ▪ A programming language is used as sequentially executed commands ▪ Multiple tasks are realized by switching between them ▪ Fixed, dedicated I/O ports, little possibility of changing functionality: e.g. PWM, I2C, SPI ▪ **Built-in ADC and DAC converters**	▪ User-configurable architecture ▪ A programming language used for internal device configuration ▪ Tasks are realized parallelly and simultaneously ▪ User-configurable I/O ports, high possibility of changing functionality ▪ **Most often without built-in ADC and DAC converters**
Similarities	▪ The digital circuit is in the form of an integrated circuit ▪ Can realize implemented algorithms ▪ Programmable by a dedicated language	

FPGA – Ways of use

Block schematic

- Basic logic elements
(e.g. logic gates, flip-flops)
- Hardware blocks generated by the auxiliary tools
(e.g. counters, multiplexers)
- Custom hardware blocks
(implemented in HDL by the user)



HDL
(Hardware Description Languages)

The most commonly used:

- VHDL
- (System) Verilog
- System C

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.numeric_std.all;
4
5 entity signed_adder is
6   port
7   (
8     aclr : in   std_logic;
9     clk  : in   std_logic;
10    a    : in   std_logic_vector;
11    b    : in   std_logic_vector;
12    q    : out  std_logic_vector
13  );
14 end signed_adder;
15
16 architecture signed_adder_arch of signed_adder is
17   signal q_s : signed(a'high+1 downto 0); -- extra bit wide
18
19 begin -- architecture
20   assert(a'length >= b'length)
21     report "Port A must be the longer vector if different sizes!"
22     severity FAILURE;
23   q <= std_logic_vector(q_s);
24
25   adding_proc:
26   process (aclr, clk)
27     begin
28       if (aclr = '1') then
29         q_s <= (others => '0');
30       elsif rising_edge(clk) then
31         q_s <= ('0'&signed(a)) + ('0'&signed(b));
32       end if; -- clk'd
33     end process;
34
35 end signed_adder_arch;
```

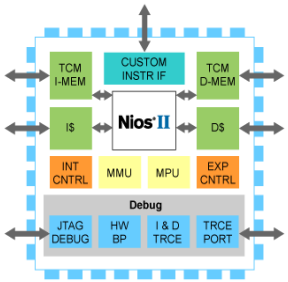
Soft-processor
(e.g. Nios II)

Advantages :

- user-configurable architecture;
- possibility of implementation your own peripherals;
- the ability to divide tasks into hardware and software parts;
- possibility of implementation several S-P units in one integrated circuit.

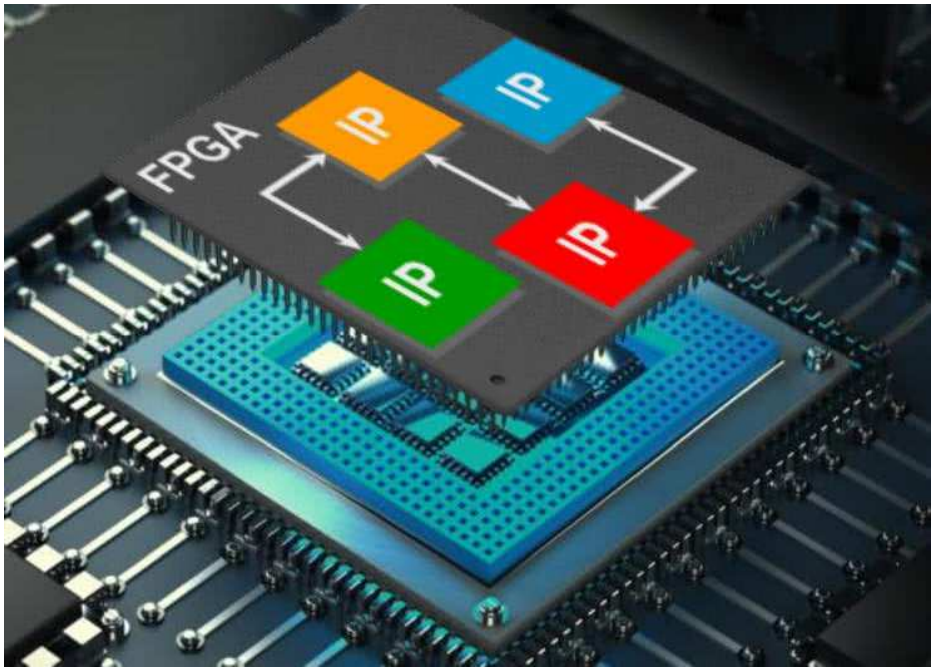
Disadvantages :

- lower operating frequency (clock speed);
- usually this platform is more expensive.



FPGA Application

- Embedded systems
- Control systems of mechatronic devices
- Implementation of numerical computations
- Digital Signal Processing (DSP)
- Measurement cards
- Image processing systems
- Military applications
- Network applications
- Power electronics
- ASIC prototyping



DE10-Lite
Development and Education Board

FPGA

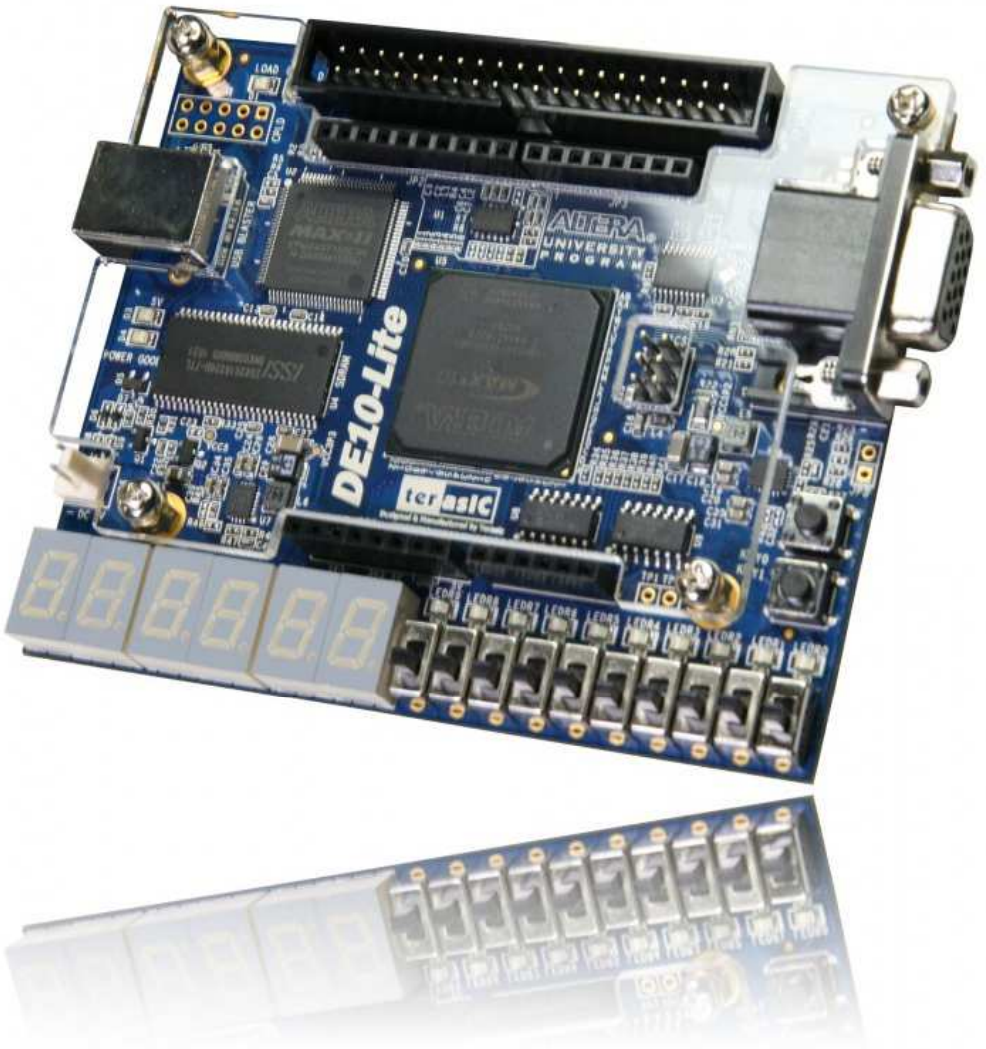
- MAX 10 10M50DAF484C7G
- 50 000 Logic Elements (LEs);
- 360 GPIO

(General Input/Output Ports);

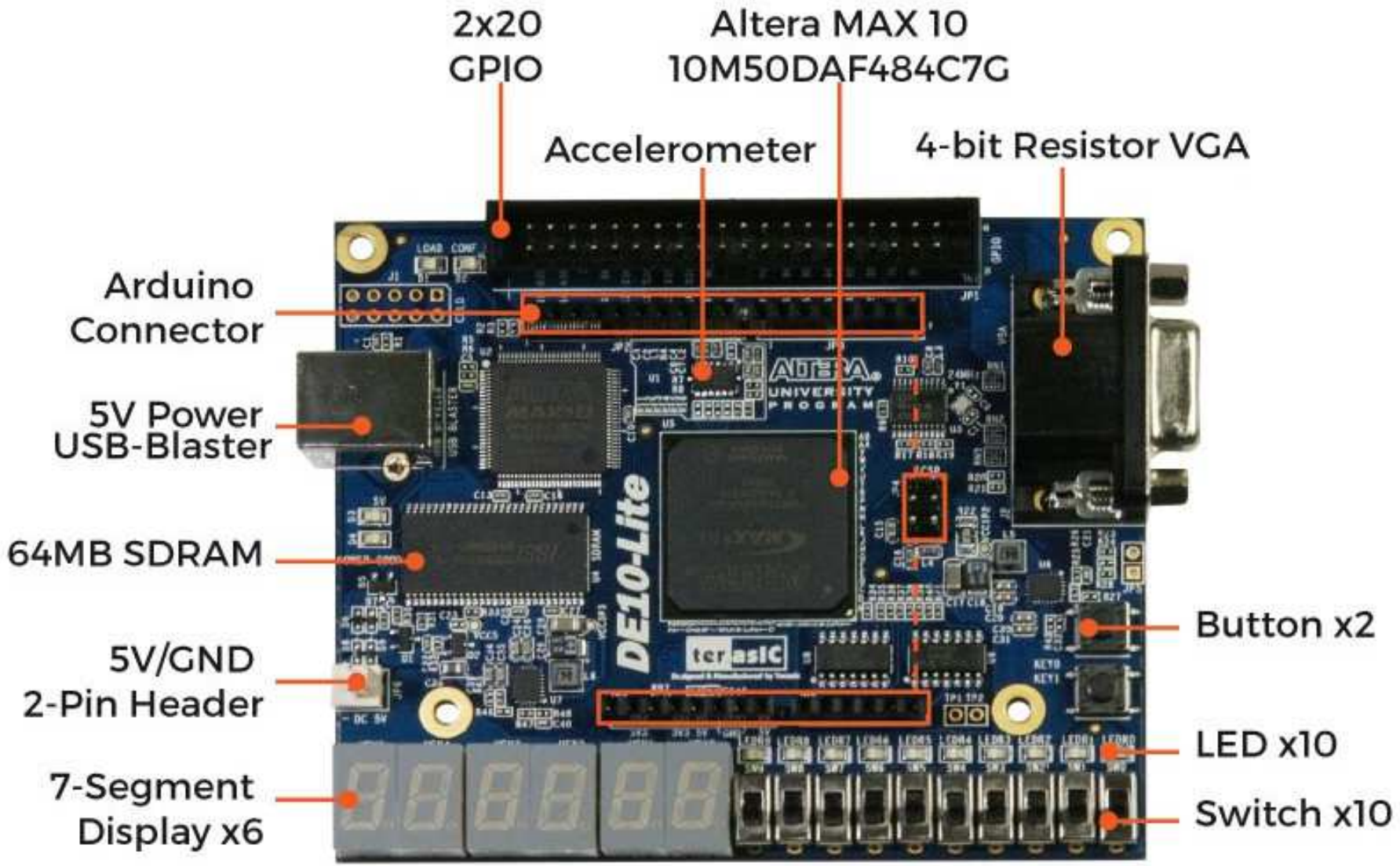
- 288 embedded multiplier (9-bit elements);
- 7 ns propagation time through LEs;

(sequential logic circuits ~ 100-120 MHz);

- 1.638 Kbit memory bits;
- 4 x PLL;
- embedded 2 x ADC.



DE10-Lite
Development and Education Board



BIBLIOGRAPHY

[1] https://www.intel.com/
[2] https://www.terasic.com.tw/en/
[3] https://kamami.pl/
[4] https://www.arm.com/
[5] https://allaboutfpga.com/

Video
[A] https://www.youtube.com/watch?v=G-dKPe4Zz_M
[B] https://www.youtube.com/watch?v=CnSE-gF4jng
[C] https://www.youtube.com/watch?v=lG2I56NKLQk
[D] https://www.youtube.com/watch?v=k_7IV0U2JhQ
[E] https://www.youtube.com/watch?v=CXb_m8p3Yt4
[F] https://www.youtube.com/watch?v=ANxHyCAYGp0
[G] https://www.youtube.com/watch?v=NBtGxF4UZRs