BRNO UNIVERSITY OF TECHNOLOGY

PULSE AND DIGITAL TECHNIQUES

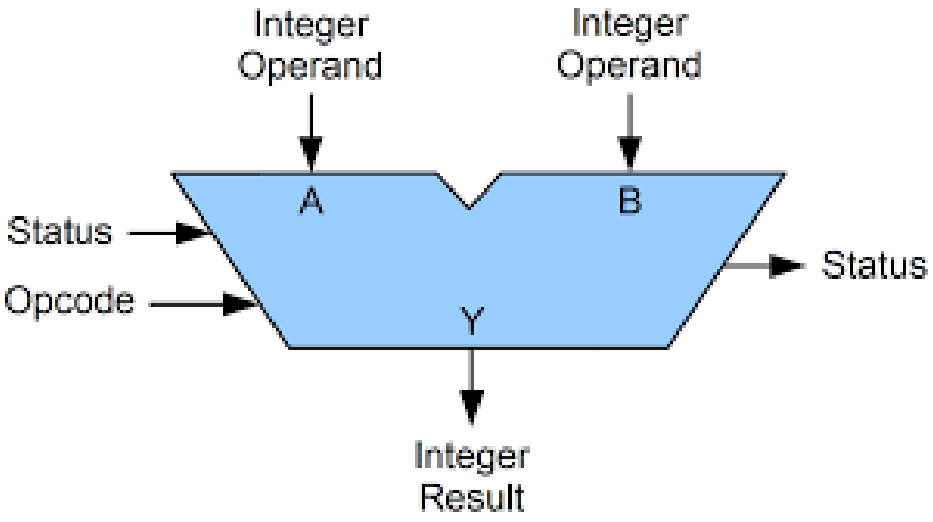
ALU Unit

Enrique J. Castellano Martín

Index

Selected Project

- ALU unit with its own set of arithmetic/logic/binary operations.
- An ALU (Arithmetic Logic Unit) is combinational digital electronic circuit that performs arithmetic and bitwise operations on integer binary numbers.

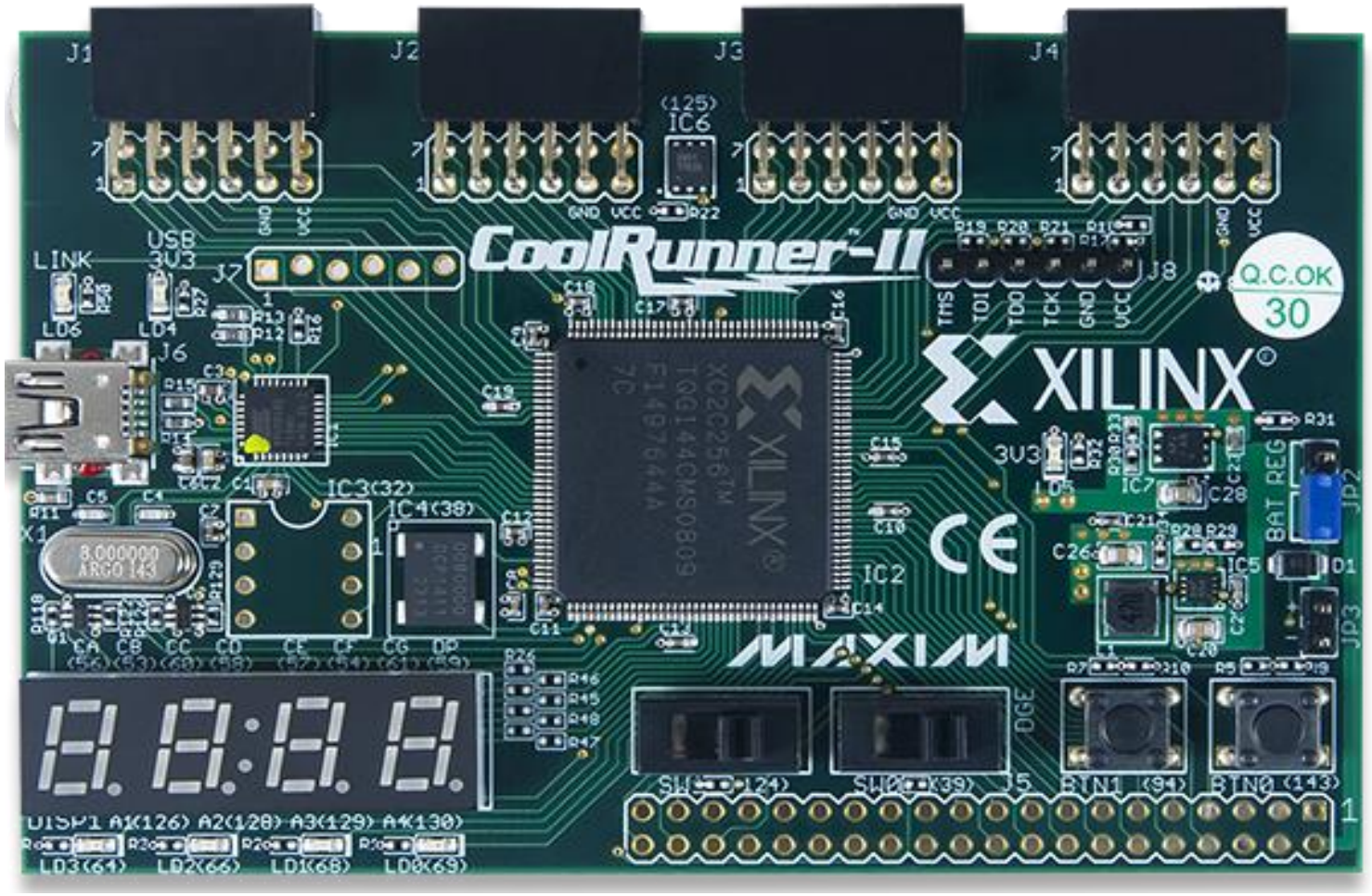


Selected Project

- I inspired in the device 74181:

  - First complete ALU on a single chip
  - It’s a 4-bit wide ALU
  - It can perform all the traditional add / subtract / decrement operations with or without carry, as well as AND / NAND, OR / NOR, XOR, and shift.

Materials


CoolRunner-II
XILINX
MAXIM
Q.C.OK
30
LINK
USB
3V3
BAT REG
J1
J2
J3
J4
J5
J6
J7
J8
IC2
IC3(32)
IC4(38)
IC6
IC7
LD0(69)
LD1(68)
LD2(66)
LD3(64)

VHDL Programming

- Implementation Constraints File

```
# CoolRunner-II CPLD Starter Board

# Inputs
# Pushbuttons
NET BTN0        LOC = P143;
NET BTN1        LOC = P94;
# Slide Switches
NET SW0         LOC = P39;
NET SW1         LOC = P124;

# Outputs
# LEDs
#NET LED<0>      LOC = P69;
#NET LED<1>      LOC = P68;
#NET LED<2>      LOC = P66;
#NET LED<3>      LOC = P64;

# Seven-Segment Display(s)
NET D_POS<0>    LOC = P126;
NET D_POS<1>    LOC = P128;
NET D_POS<2>    LOC = P129;
NET D_POS<3>    LOC = P130;
NET D_SEG<0>    LOC = P56;     # a
NET D_SEG<1>    LOC = P53;     # b
NET D_SEG<2>    LOC = P60;     # c
NET D_SEG<3>    LOC = P58;     # d
NET D_SEG<4>    LOC = P57;     # e
NET D_SEG<5>    LOC = P54;     # f
NET D_SEG<6>    LOC = P61;     # g
#NET D_SEG<7>    LOC = P59;     # dp

# Clock
#NET clk         LOC = P38;
```

VHDL Programming

- VHDL module

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3   use ieee.std_logic_arith.all;
4   use ieee.std_logic_unsigned.all;     -- needed for +/- operations

6   -- Input/output description
7   entity top is
8   port (
9                           BTN0: in std_logic;      -- button 0
10                          BTN1: in std_logic;      -- button 1
11                      SW0: in std_logic;       -- switch 0
12                          SW1: in std_logic;       -- switch 1
13                          D_POS: out std_logic_vector(3 downto 0);     -- display positions
14                          D_SEG: out std_logic_vector(6 downto 0)      -- display segments
15          );
16  end top;

18  -- Internal structure description
19  architecture Behavioral of top is
20      -- internal signal definitions
21      signal variable1: std_logic_vector(3 downto 0) :="0000";
22      signal inp_a: std_logic_vector(3 downto 0) :="0000";
23      signal inp_b: std_logic_vector(3 downto 0) :="0000";
24      signal inp_alu: std_logic_vector(3 downto 0) :="0000";
```

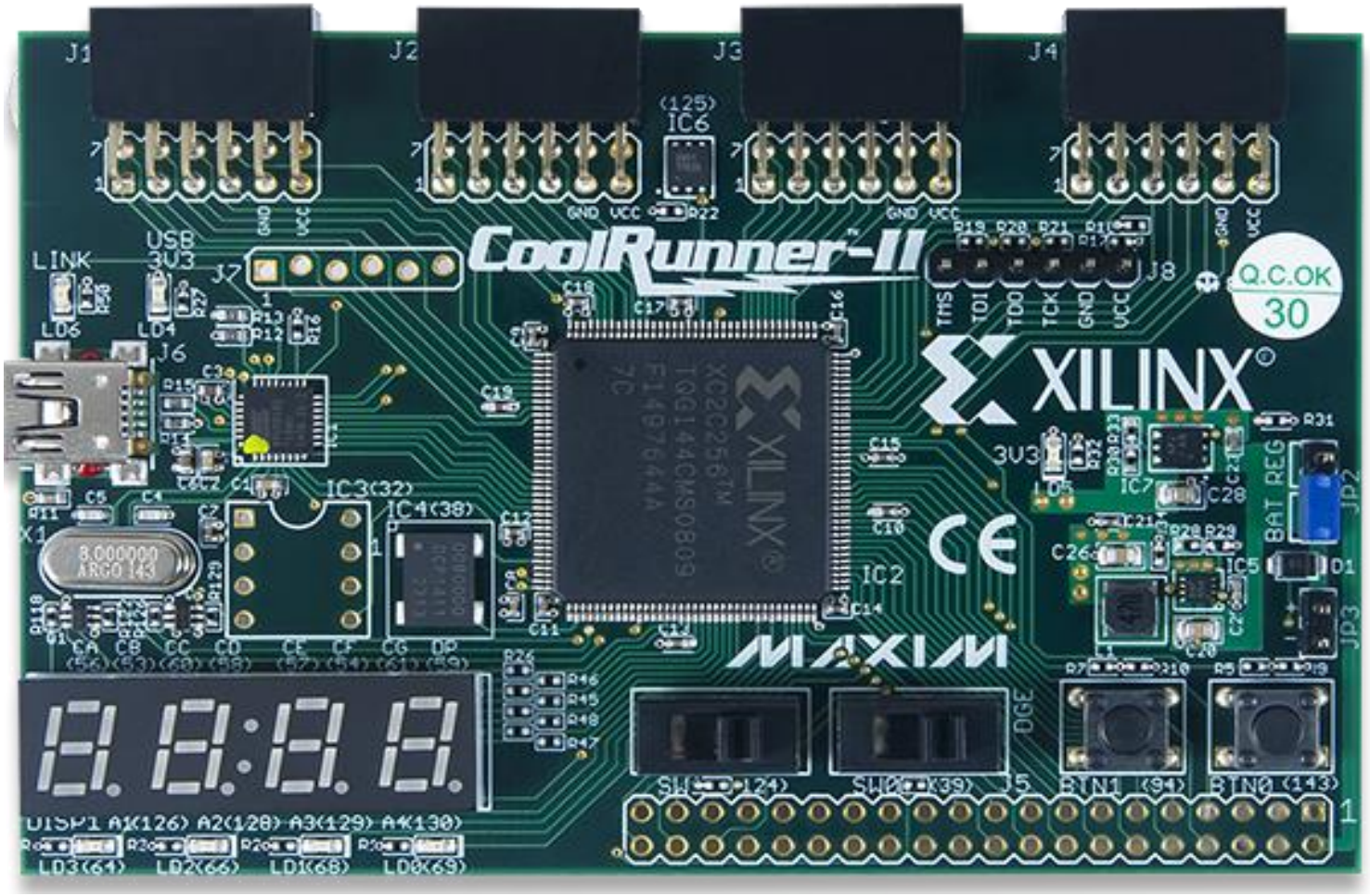
VHDL Programming

- VHDL module

```
begin
    process(SW1,SW0,BTN0,BTN1)
    begin
        if SW1='0' then
            if SW0='0' then
                if BTN0='0' then
                    inp_a <= "0000";
                end if;
                if BTN1='0' then
                    inp_a <= inp_a + 1;
                end if;
            else
                if BTN0 = '0' then
                    inp_b <= "0000";
                end if;
                if BTN1 = '0' then
                    inp_b <= inp_b + 1;
                end if;
            end if;
        else
            if SW0='1' then
                if BTN0 = '0' then
                    inp_alu <= inp_a - inp_b;
                else
                    inp_alu <= inp_a  + inp_b;
                end if;
            else
                if BTN0='0' then
                    inp_alu <= inp_a or inp_b;
                    if BTN1='0' then
                        inp_alu <= inp_a nand inp_b;
                    end if;
                end if;
                if BTN1='0' then
                    inp_alu <= inp_a and inp_b;
                else
                    inp_alu <= inp_a xor inp_b;
                end if;
            end if;
        end if;
    end process;
```

VHDL Programming

- VHDL module



Video

Thank you for your attention