IMPLEMENT WITH PROGRAMMABLE LOGIC DEVICES OF THE CONTROL UNIT OF A PULSE GENERATOR FOR EXPERIMENTAL STUDIES IN EDM

Constantin DIACONESCU¹, Dan NANU², Mihai TITU³

ABSTRACT

This paper presents the architecture and implementation on programmable logic devices of the control unit for an experimental pulse generator, used for studying the EDM basic processes. The specific design techniques using advanced programmable logic devices and the possibility of realizing flexible architectures are particularly emphasized.

KEYWORDS: generator, EDM, PLD, VHDL

1. PROGRAMMABLE LOGIC DEVICES

Programmable logic devices (PLD) are on large or very large scale integrated circuits containing a large number of elementary logic structures, none or partly interconnected. The user realizes the final connection, in concordance with a specific application. The interconnections are realized in an electric array (matrix), containing programmable connectors, placed in the crossing points of the rows and columns. The producer sales the circuit with all the connectors in the same state: "connected" or "non connected". The user builds his own architecture, by changing the state of several connectors, action named circuit programming. Depending on the physical principle of the connectors, its programming can be reversible (reprogrammable connectors) or irreversible (connectors with definitive programming) [3,4].

The fundamental logic structure in PLDs is the AND-OR group (Fig. 1), able to implement any logic function represented in disjunctive form:

\[ f(a_1, a_2, a_3, \ldots, a_n) = a_1 \cdot a_2 \cdot a_3 + a_4 \cdot a_5 + a_6 \cdot a_7 + a_8 \cdot a_9 + \ldots \]  

Fig. 1 Fundamental logic structure in PLDs

Modern programmable logic devices are structured in logic macrocells [3, 4]. Each macrocell contains a large AND-OR unit, a configurable output block able to implement a memory cell with preset/reset facility and synchronization with a system clock. Macrocells are combined in logic blocks, having a common programmable array. The programmable chip communicates with the environment by specific in, out or in/out cells, connected to the pins. The signal transmission between different blocks occurs by programmable interconnect structures. Figure 2 presents the simplified structure and some important characteristics for three
classes of programmable logic devices: PAL, CPLD and FPGA [3, 4].

**PAL**  
(Programmable Array Logic)
- one only logic block  
- 8…16 logic macrocells (LMC)  
- about 1000 logic gates

**CPLD**  
(Complex Programmable Logic Device)
- 2…56 logic blocks (LB)  
- 32…3000 macrocells  
- 200000 logic gates  
- logic blocks are related by an interconnect array

**FPGA**  
(Field Programmable Gate Array)
- 64…5000 logic blocks (LB)  
- 100000 macrocells  
- 200000 logic gates  
- logic blocks are related by a complex bus system

Fig. 2 Simplified structure and some important characteristics for three classes of programmable logic devices
2. THE CONTROL UNIT FOR A PULSE GENERATOR

Figure 3 presents the architecture of the control unit for a pulse generator for experimental studies regarding the effect of successive discharges in EDM. Previous researches in this field [2] have proved cumulative effects of successive discharge pulses. The control unit pilots a 20 an experimental static power switch [7].

The control unit has to generate equidistant pulse salvos. The lapse of time between successive pulses is controlled by the frequency of a pilot oscillator \( f_0 \). The number of pulses in a salvo can be fixed in the interval for 1 to 127. This number consists in a 7-bit logic vector (a). A reference unit supplies it. The command pulses (clk) are allowed to pass through the \( P_2 \) gate, towards the static power switch (SPS). A digital counter (Count.) counts the current pulses in the discharge circuit, taken over a current sensor (CS). A comparator (Comp.) perceives the equality of the number of effective realized current pulses \( n \) and the reference value \( a \) and blocks the \( P_1 \) gate. The command pulses are synchronized with the pilot oscillator by a d-flip-flop acting on falling edge. A start controller (SC) brings the unit in initial state, ready to generate a new salvo of pulses. Figure 4 shows the state diagram of the start controller.

3. DESIGN AND SIMULATION OF THE CONTROL UNIT

The VHDL program describing the control unit is the following:

```vhdl
library ieee; use ieee.std_logic_1164.all;
use work.std_arith.all;
entity salve is
port (clk,clk1,start,ini:in std_logic;
a:in std_logic_vector(6 downto 0);
ys:out std_logic);
end salve;
architecture a salve is
begin
  s : process (ini,start,a)
  variable d: std_logic;
  begin
    if ini='1' then
      d := "00000000";
    elsif start='1' then
      d := a;
    else
      d := "00000000";
    end if;
    y <= d;
  end process;
end a salve;
```

Fig. 3 architecture of the control unit for a pulse generator for experimental studies

Fig. 4 state diagram of the start controller
end salve;
architecture arch_salve of salve is

type stare is (s0,s1,s2);
signal s:stare;
signal d,q,nstop:std_logic;
signal n:std_logic_vector(6 downto 0);
signal w:std_logic_vector(1 downto 0);

begin
  -- gate P1
  d<=w(0) and nstop;
  -- d_flip_flop
  dff:process(w(1),clk)
  begin
    if w(1)="1" then q<='0';
    elsif falling_edge(clk)then q<=d;
    end if; end process dff;

  -- gate P2
  y<=clk and q;
  -- counter
  num:process(w(1),clk1)
  begin
    if w(1)="1" then n<="0000000";
    elsif rising_edge(clk1) then n<=n+1;
    end if; end process num;

  -- comparator
  nstop<='0' when a=n else '1';
  -- reset controller
  begin
    if ini='1' then s<=s0;
    automat.process(clk,ini)
    automat.process(clk,ini)
    elsif rising_edge(clk) then
      case s is
        when s0=>if start='1' then s<=s;
        else s<=s0;
        end if;
        when s1=>s<=s2;
        when s2=>if nstop='0' then s<=s0;
        when s2=>if nstop='0' then s<=s0;
        end if; end process automat;

  w< = "00" when s=s0 else
  "10" when s=s1 else
  "01";
end arch_salve;

------------------------------------------------------
The VHDL source has been processed using the GALAXY program in WARP 5.1 development package [5]. The CPLD CY7C372i-66JC produced by CYPRESS Semiconductor has been chosen to implement the project.

The project has been simulated using the ACTIVE-HDL Sim program, in the same WARP 5.1 package [5]. The simulation results show the accurate work of the control unit in all significant situations.

4. CONCLUSIONS

Modern programmable logic devices permit to implement very complex and high performance digital structures. Programmable logic device implement is very attractive for designers especially because it offers a high level of liberty in choosing architectural solutions. Using hardware description languages, VHDL particularly, offers the possibility to control the performances by repetitive simulation and recurrent design.

The design of control units for pulse generators in EDM equipments can fully benefit of all these advantages.


AUTHORS

1 PhD. Constantin Diaconescu
2 Prof. Phd., Dan NANU
3 Associate Prof. Phd., Mihai Titu
Lucian Blaga University of Sibiu Romania