Chapter-4

Hardware & Software
details of
CPLD based design
Hardware & Software details of

CPLD Based Temperature Measurement and Control System

4.1 Introduction

A Complex Programmable Logic Device (CPLD) is a combination of a fully programmable AND/OR array and a bank of macrocells [1]. The AND/OR array is reprogrammable and can perform a multitude of logic functions. Macrocells are functional blocks that perform combinatorial or sequential logic, and also have the added flexibility for true or complement, along with varied feedback paths. CPLD is a programmable logic device with complexity between that of PALs and FPGAs, and architectural features of both. The building block of a CPLD is the macrocell, which contains logic implementing disjunctive normal form expressions and more specialized logic operations.

Most CPLDs contain macrocells with a sum-of-product combinatorial logic function and an optional flip-flop. Depending on the CPLD, the combinatorial logic function supports from four to sixteen product terms with wide fan-in. CPLDs also vary in terms of logic gates and shift registers. For this reason, CPLDs with a large number of logic gates may be used in place of FPGAs. Another CPLD specification denotes the number of product terms that a macrocell can manage. Product terms are the product of digital signals that perform a specific logic function.

CPLDs are available in many IC package types and logic families. CPLDs also vary in terms of supply voltage, operating current, standby current and power dissipation. In addition, CPLDs are available with different amounts of memory and different types of
memory support. Typically, memory is expressed in bits or megabits. Memory support includes read-only memory (ROM), random access memory (RAM), and dual-port RAM. It also includes content addressable memory (CAM) as well as first-in, first-out (FIFO) memory and last-in, last-out (LIFO) memory.

There are several performance specifications for CPLDs. Internal frequency is the speed at which CPLDs can perform operations or transfer data internally. The propagation delay is the time interval between the application of an input signal and the occurrence of the corresponding output in a logic circuit. Speed grade indicates the delay in nanoseconds (ns) through a macrocell in the device. For example, a device with a speed grade of 10 has a delay of 10 ns through a macrocell. Devices with low speed grade numbers run faster than devices with high-speed grade numbers.

Some CPLDs include integrated phase-locked loops (PLLs) and delay-locked loops (DLLs) with clock-frequency-synthesis capabilities so that designers can use CPLDs for system-on-chip (SoC) applications. PLL and DLL clock multiplication also allow designers to generate a high-speed internal clock for sampling data in digital signal processing (DSP) applications. In addition, PLLs and DLLs provide greater control over the clock frequencies used in integrated designs. This is critical for system integration because different parts of a system operate at different clock frequencies [2].

CPLD's predictable timing characteristics make them ideal for critical, high-performance control applications [3]. Typically, CPLDs have a shorter and more predictable delay than FPGAs and other programmable logic devices. Because they are inexpensive and require relatively small amounts of power, CPLDs are often used in cost-
effective, battery-operated portable applications. CPLDs are also used in simple applications such as address decoding.

Features in common with PALs

(i) Non-volatile configuration memory. Unlike many FPGAs, an external configuration ROM isn't required, and the CPLD can function immediately on system start-up.

(ii) For many legacy CPLD devices, routing constrains most logic blocks to have input and output signals connected to external pins, reducing opportunities for internal state storage and deeply layered logic. This is usually not a factor for larger CPLDs and newer CPLD product families.

Features in common with FPGAs

Large number of gates available. CPLDs typically have the equivalent of thousands to tens of thousands of logic gates, allowing implementation of moderately complicated data processing devices. PALs typically have a few hundred gate equivalents at most, while FPGAs typically range from tens of thousands to several million.

4.2 Hardware Details

The block diagram of CPLD based temperature measurement and control system is illustrated in Fig.4.1. The block diagram consists of Pt100, Water bath with motorized stirrer, Signal Conditioning Circuit, Analog to Digital Converter (AD574), Variac of range 0-260V, Electromechanical Relay, CPLD, JTAG and LED array.
The target CPLD device used in the present work is XC9572 manufactured by Xilinx. Design development and debugging is carried on a low-cost, full-featured kit provided by Applied Digital Microsystems (ADM) Pvt., Ltd., Mumbai.

This board provides all the tools required to quickly begin design and verifying CPLD platform designs. Designs are based on 10 MHz clock. The following sections describe the complete design and development of hardware and software aspects of CPLD based temperature measurement and control system.

Fig.4.1 Block Diagram of CPLD based Temperature Measurement and Control
The complete details of Pt100, Signal conditioning circuit, AD574, Water bath with Motorized Stirrer, JTAG and Electromechanical Relay are discussed in the previous chapter. The Schematic of CPLD based temperature measurement and control system is shown in Fig.4.2.

![Schematic of CPLD based Temperature Measurement and Control System](image)

**Fig.4.2 Schematic of CPLD based Temperature Measurement and Control System**

### 4.2.1 Details of XC9572 (CPLD)

The XC9572 is a high-performance CPLD providing advanced in-system programming and test capabilities for general purpose logic integration [4]. It is comprised of eight 36V18 Function blocks, providing 1,600 usable gates with propagation delays of 7.5ns. Power dissipation can be reduced in the XC9572 by configuring macrocells to standard...
or low-power modes of operation. Unused macrocells are turned off to minimize power dissipation.

The Specifications of XC9572 are as follows

⇒ 7.5ns pin-to-pin logic delays on all pins
⇒ $f_{\text{CNT}}$ to 125MHz
⇒ 72 macrocells with 1,600 usable gates
⇒ Up to 72 user I/O pins
⇒ 5V in-system programmable
  • Endurance of 10,000 program/erase cycles
  • Program/erase over full commercial voltage and temperature range
⇒ Enhanced pin-locking architecture
⇒ Flexible 36V18 Function Block
  • 90 product terms drive any or all of 18 macrocells within Function Block
  • Global and product term clocks, output enables, set and reset signals
⇒ Extensive IEEE STD 1149.1 boundary-scan (JTAG) support
⇒ Programmable power reduction mode in each macrocell
⇒ Slew rate control on individual outputs
⇒ User programmable ground pin capability
⇒ Extended pattern security features for design protection
⇒ High-drive 24 mA outputs
⇒ 3.3V or 5V I/O capability
⇒ Advanced CMOS 5V FastFLASH technology
Chapter 4

Hardware & Software details of CPLD Based Design

- Supports parallel programming of more than one XC9500 concurrently
- Available in 44-pin PLCC, 84-pin PLCC, 100-pin PQFP, and 100-pin TQFP packages

4.2.2 Daughter Board Details

In the present application, the same base board which is discussed in the previous chapter is used. The Daughter board contains XC9572 (CPLD) from Xilinx, which can be plugged on to the base board [5]. XC9572 device supports serial configurations, using the master/slave serial and JTAG modes, as well as byte-wide configuration employing the slave parallel mode. Fig.4.3 shows the photograph of the XC9572 (CPLD) Daughter board used in the present work.

![Photograph of the XC9572 (CPLD) Daughter Board](image-url)

Fig.4.3 Photograph of the XC9572 (CPLD) Daughter Board
Daughter Board includes the following items

- Xilinx CPLD (XC9572)
- Seven 10-pin Headers, intended for use as a GPIO
- Four sets of 20x2 female berg connector for plugging on to the baseboard
- JTAG pins for in circuit programming
- Power supply 5V, 3.3V and 2.5V are provided from the baseboard

4.3 Software Features

The following section describes the software development for CPLD based temperature monitoring system using Pt100 in VHDL. The necessary functional modules, state flow diagrams are presented and thoroughly discussed. The software tools used for building and testing these modules in the present work is Xilinx ISE 9.1i.

Flow chart of the CPLD based temperature measurement and control system is shown in the Fig.4.4.
Flow Chart

Start

Initialize Clk, Relay
ADC & LCD

Turn ON the heater and
read the temperature
from sensor

Start ADC
conversion

Call Delay

NO

Is conversion
Completed?

YES

Send converted digital
data to CPLD

Compare digital data with temperature
values in look up table

Display temperature on LED Array

NO

Is set point
reached?

YES

C

A
Chapter 4

Hardware & Software details of CPLD Based Design

Turn OFF the heater
Call Delay & wait for some time
Go back to check for steady state of temperature

Fig. 4.4 Flow chart for Temperature Measurement and Control System

VHDL code for temperature measurement and control

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ad574_relay_lcd_Pt100 is
port (adc_data : inout std_logic_vector(11 downto 0);
      -- 12-bit ADC data
);
clk100k : in std_logic;
-- 100 KHz Clock to the module

relay : out std_logic;
-- Electromechanical relay to control the temperature

adc_RC : inout std_logic;
-- Read/Convert pin of the ADC

LED : out std_logic_vector (11 downto 0));
-- 12 LEDs to display the digital output

dend ad574_relay_lcd_Pt100;

Architecture Behavioral of ad574_relay_lcd_Pt100 is

signal clkdiv : std_logic_vector(14 downto 0);
signal clckey : std_logic;
signal clkrelay : std_logic;

begin

Signal assignment
-- The below statements are used for signal assignment of data read from AD574, --
-- Temperature display on LED Array and control its value using Relay--

P1: process (clk100k,clkdiv)
begin
    if (rising_edge(clk100k)) then
        clkdiv <= clkdiv + 1;
        end if;
        clkkey <= clkdiv(13);
        clkrelay <= clkdiv(14);
    end process P1;

-------------------------------------------------------------------Initialization of AD574 & LED display------------------------
--The below statements are used to send a signal for ADC from CPLD and to display the
digital data on 12- LEDs-------------------------------------------------------------------

P2: process (clkkey, adc_RC)
begin
    if rising_edge(clkkey) then
        adc_RC <= '1';
        if (adc_RC = '1') then
            LED <= adc_data;
            adc_RC <= '0';
        end if;
    end if;
end process P2;
Controlling the temperature

---The below statements are sending a signal to relay from CPLD and controls the temperature at desired value---

P3: process (clkrelay)
begin
if rising_edge(clkrelay) then
if (adc_data >= "001100101110") then --80.0 degrees
relay <= 'O';
else
relay <= '1';
end if;
end if;
end process P3;
end behavioral;

4.4 Xilinx ISE9.1i (EDA Tool)
The Integrated Software Environment (ISE 9.1i) is the Xilinx design software suite that allows to take the design from design entry through Xilinx device programming. The ISE
Project Navigator manages and processes the design through the following steps in the ISE design flow. The snapshot of the Xilinx ISE 9.1i environment opening window is shown in Fig. 4.5.

### 4.4.1 Design Entry

Design entry is the first step in the ISE design flow [6]. During design entry, source files can be designed based on the design objectives. Top-level design file can be created using a Hardware Description Language (HDL), such as VHDL, Verilog, or ABEL, or using a schematic. In the present work VHDL is used as a programming language.
4.4.2 Assigning Package Pins

Input and output signals to package pins in the design are assigned using the Assign Package Pins process. This process launches the Floorplan Editor View within ISE and displays pin constraint information for the design. This process launches Pinout and Area Constraints Editor (PACE) for all other device-targeted designs. Using the editor, one can assign I/O locations, specify I/O banks, specify I/O standards, prohibit I/O locations, and create legal pin assignments using the built-in design rule checks (DRC). Package pin assignments are saved as a user constraints file (UCF). The snapshot of the package pins assignment window is shown in Fig.4.6.

Fig.4.6 The package pin assignment window for XC9572 (CPLD)
4.4.3 Implement Design

For CPLD, the implement design process can be run in one step from Project Navigator, or each of the implementation processes like Synthesis, Translate, Fit and Generate programming file can be run separately [7]. During Synthesis step, VHDL designs become netlist files that are accepted as input to the implementation step. Design implementation converts the logical design into a physical file format that can be downloaded to the selected target device. The snapshot of the Fitter Report display in a browser window is shown in Fig.4.7.

Fig.4.7 The fitter report display for XC9572 (CPLD)
4.4.4 Generating Programming file

JEDEC Files

JEDEC files are XC9500/XL/XV CPLD programming files generated by the Xilinx fitter [8]. They are ASCII text files containing programming information and, optionally, functional test vectors that can be used to verify the correct functional behavior of the programmed device. One JEDEC file is required for each XC9500/XL/XV device in the JTAG programming chain as shown in Fig.4.8. The name of the JEDEC file is called as <design name>.jed

Device configuration

The device is configured after generating a programming file, during configuration, the generated configuration files and programming files are downloaded to Xilinx device from host computer.

Fig.4.8 JTAG chain which shows the configured device
4.5 Temperature measurement by interfacing Pt100 to CPLD using AD574

Pt100 is inserted in the water bath; when the temperature of the water bath changes, then the resistance of Pt100 is also changes. The output produced by the Pt100 needs to be converted into quantized voltage levels. This can be achieved through a process known as “signal conditioning”. The signal conditioning circuit gives the analog voltage, which is proportional to the temperature. The output of the signal conditioning circuit is fed to the ADC (AD574), since FPGA can process only digital data. AD574 converts the input analog voltage into 12-bit digital data.

In order to initiate AD574, CPLD sends a high signal to R/C pin of AD574. The conversion starts when signal to R/C goes low. When the end of conversion signal of the AD574 is found to be high, it is an indication to the CPLD that 12-bit data which represents the temperature can be read in. CPLD compares the corresponding temperature value from look up table and finally the calibrated temperature value is displayed on LED array.
REFERENCES


2. Bob Zeidman, Designing with FPGAs and CPLDs, CMP Books (USA), 2002.


