WATCHDOG TIMER USING VHDL FOR ATM SYSTEM

Gore S.S.[1], Lokhande A.A.[2], Mahajan S.B.[3]

[1] Professor , Elecronics & Telecommunication, MCOERC, Maharashtra, India.
[2] Student , Elecronics & Telecommunication, MCOERC, Maharashtra, India.
[3] Student , Elecronics & Telecommunication, MCOERC, Maharashtra, India.

ABSTRACT

A watchdog timer is a computer hardware timing device that triggers a system reset if the main program, due to some fault condition, such as a hang, neglects to regularly service the watchdog. The intention is to bring the system back from the hung state into normal operation. Such a timer has got various important applications, one of them being in ATMs which we have studied in our paper. We can implement watchdog timer by using hardware as well as software. The advantage of implement it using software rather than hardware is that it will required less power consumption, less cost and we obtain high speed compare to hardware. The compatible or good known language for Xilinx is VHDL. The key advantage of VHDL when used for systems design is that it allows the behavior of the required system to be described (modeled) and verified (simulated) before synthesis tools translate the design into real hardware (gates and wires) and information theory. The simulation tool that we have used is Xilinx 6.2i . Xilinx provide platform for VHDL.

First the required code for timer circuit was written in VHDL and simulated so as to obtain the required output waveforms. After the coding was completed, VHDL model is translated into the "gates and wires" that are mapped onto a programmable logic device. The programmable logic device used here is Spartan-II. The above coding and burning methods were completed and the output was observed on FPGA kit. The timer code was implemented using VHDL while burning was done using Spartan-II kit.

Keywords: *ATM (Automated Teller Machine), CLB(Configurable Logic Blocks), DLL (Delay Locked Loops), DRC (Design Rule Checker), EMI (Electromagnetic Interference), FPGA(Field Programmable Gate Array),VHDL (Very High Speed Integrated Circuits Hardware Description Language), VLSI (Very Large Scale Integration).*

1.INTRODUCTION

Today, microcontrollers are being used in tough environments where electrical noise and EMI are plentiful. In environments like this, it is beneficial if the system contains resources to help ensure proper operation. In many systems, a commonly used technique for verifying proper operation is the combination of a watchdog timer. A watchdog timer is fundamentally a time measuring device that is used in conjunction with or as part of a microprocessor and is capable of causing the microprocessor to be reset. A system using a watchdog timer is particularly well suited to detecting bit errors. Momentary bit errors can be caused by such things as soft memory failures and electromagnetic discharges into memory devices and their interfaces. These can cause temporary bit polarity flipping of data into and out of the processor. When this occurs while fetching program information, the microprocessor will begin executing invalid code. The most common use of the High-Speed Micro's watchdog timer is as a system supervisor [5].

VHDL is a hardware description that can be used to model a digital system. The hardware abstraction of this digital system is known as ENTITY. To describe an entity, VHDL provides five different types of primary construct, called as design units. They are: Entity declaration, Architecture body, Configuration declaration, Package

declaration and Package body. Architecture body consist of three types of modeling which are structural, dataflow and behavioral. Behavioral style of modeling specifies the behavior of an entity as a set of statement that are executed sequentially. All the flow of program is executed sequentially, which is the function of behavioral modeling. As VHDL provides an extensive range of modeling capabilities, it is possible to quickly assimilate a core subset of the language that is both easy and simple to understand without learning the complex features [3-4].

FPGA is a semiconductor device containing programmable logic components and programmable interconnects .The programmable logic components can be programmed to duplicate the functionality of basic logic gates such as AND, OR, XOR, NOT or more complex combinational functions such as decoders or simple mathematical functions. The advantages of FPGA are that it is shorter time to market,ability to re-program in the field to fix bugs and lower non recurring engineering costs. The typical basic architecture consists of an array of CLBs and routing channels. Multiple I/O Pads may fit into the height of one row or the width of one column in the array. Generally, all the routing channels have the same width (number of wires). An application circuit must be mapped into an FPGA with sufficient resources [5].

The Spartan-II 2.5V FPGA family gives users high performance, rich logic resources, and a rich feature set, all at an exceptionally low price. The six-member family offers densities ranging from 15,000 to 200,000 system gates. System performance is supported up to 200 MHz Spartan-II devices deliver more gates, I/Os, and features per dollar than other FPGAs by combining advanced process technology with a streamlined Virtex - based architecture . Features include block RAM (to 56K bits), distributed RAM (to 75,264 bits), 16 selectable I/O standards, and four DLLs. Fast, predictable interconnect means that successive design iterations continue to meet timing requirements. The Spartan-II family is a superior alternative to mask-programmed ASICs. The FPGA avoids the initial cost, lengthy development cycles, and inherent risk of conventional ASICs. Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary [6].

2. LITERATURE SURVEY

VLSI system can be implemented with the help of back-end as well as front-end tools. Approach of good designer towards in front-end is because of number of advantages over back-end tools like: Logical approach is more in front-end as compare to back- end, syntax rules are present in front-end whereas in back-end DRC rules are present , front-end is more flexible that means we can edit it fast but back-end is not that much flexible [1-2]. Hence we are implementing watchdog timer with the help of front end-tool that is Xilinx To implement the watchdog timer we can use different languages like assembly, Verilog… etc, but VHDL is the most powerful language as well as different modeling techniques can use so as to improve flexibility and performance of the system. Different modeling techniques of VHDL are Dataflow, Structural and Behavioral. In our implementation we are using behavioral modeling due to it's sequential execution and program syntax is easy as compare to dataflow and structural modeling [3-4]. We can use Altera, Atmel and Cortex-M are front end tools but that are used in industry level applications but Xilinx is used at academic level which is easy to understand at student level.

3. SOURCE CODING AND RESULT WITH DISCUSSION

3.1 SOURCE CODE

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity testclk is
Port (
        clkin: in STD_LOGIC;  ---------the clk of FPG
        reset : in STD_LOGIC;----------the reset
        exin : in std_logic;----------the input to the system
        msg : out std_logic;----------the output signalling lack of input
        clkout : out STD_LOGIC );----------divided clk
end testclk;
architecture Behavioral of testclk is
```

```
signal toggle : std_logic := '0';
signal xu  : std_logic_vector(22 downto 0);   ----------user  define signal
begin
process(clkin,reset,exin)     ----------process  is a sequential statement containing input signals

begin
if reset = '1' then
            xu  <= (others => '0');
            clkout <= '1';
elsif rising_edge(clkin) then
             xu <= xu+1;
             if exin  = '1' then
             xu <= (others => '0');
             msg <= '0';----------resetting  the output
      end if;
             if exin  = '0' and xu = "11111111111111111111111"   then
             msg <= '1';----------setting  the interrupt
     end if;
              if (xu  = "11111111111111111111111")  then
              toggle <= not toggle;----------for  toggling the clk out
              clkout <= toggle;
                 xu  <= (others => '0');
         end if;
       end if;
    end process;
  end Behavioral;
```

3.2 RESULT WITH DISCUSSION

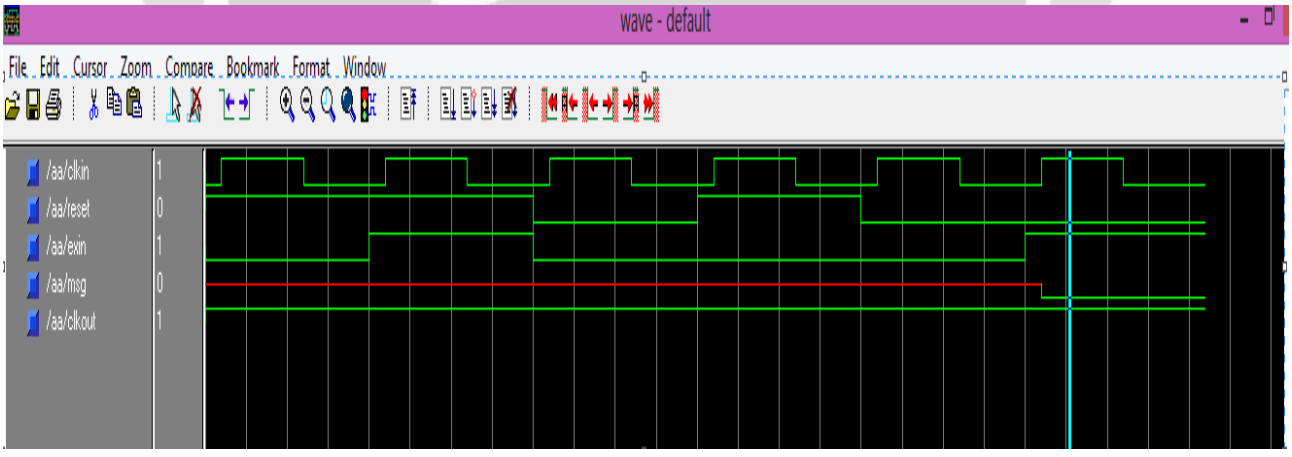


Figure 3.1: Result

We have given 3 input signals clkin, reset and exin. Clkout signal remains 1 as per condition given in program. We have also use a xu as a user defined signal. The xu signal can be incremented at the rising edge of the clkin signal. The message signal which is a output signal which depends on xu signal. The xu signal counts from 22 downto 0, which will then reset our watchdog timer if any fault condition occurs.

4. CONCLUSIONS AND FUTURE SCOPE

As VHDL provides an wide range of modeling capabilities, it is possible to quickly integrate a core subset of the language that is both easy and simple to understand without learning the complex features. We successfully implemented a timer for ATM application and observed its output, both as test bench waveform and on Spartan-II kit. The scope of this paper or project to be simulate the Watchdog timer using different simulation tools like Atmel, Altera and Cortex-M.

5. ACKNOWLEDGEMENT

We would like to take this opportunity to express our respect and deep gratitude to Dr. Pable S.D. & Prof. Ahire D.D. for giving us all necessary guidance required for this paper, apart from being constant source of inspiration and motivation. It was privilege to have worked under him. We are highly obligated to our entire staff of Electronics and Telecommunication Engineering Department, our friends, whose contribution intellectually and materially in words and deeds for preparation of this paper or project. Last but not the least, the backbone of our success and confidence lies solely on blessings of our parents.

6. REFERENCES

[1]. Gore S.S. and G.M. Phade, "Parametric Performance Analysis of 2-GHz Low Noise Amplifier", International Journal of Engineering Research and Application (IJERA), Vol-3, Issue-5, September-October -2013, PP.552-557.

[2]. Gore S.S. and G.M. Phade, "Design Challenges and Performance Parameters of Low Noise Amplifier", International Journal of Innovations in Engineering and Technology (IJIET),Vol-3, Issue-1, ISSN: 2319-1058, October -2013, PP.204-210.

[3]. J.Bhaskar, **"**A VHDL Primer",PHI Publication, Third Edition, ISBN:978-81-203-2366-7, PP.14-23,67-101.

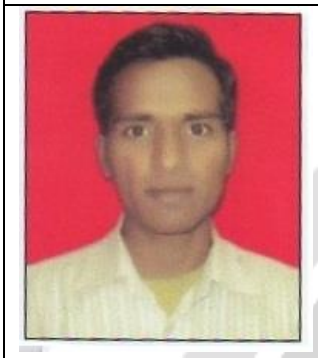
[4]. Volnei A.Pedroni, "Circuit Design with VHDL", ISBN:0-262-16224-5, PP-4-5, 13-15, 41-42, 52-53, 65-66, 91, 317-327.

[5]. Subhrajit Mishra and Ishan Dhar, **"**Design Of Timer For Application In ATM Using VHDL and FPGA", Thesis of Department of Electronics & Communication Engineering National Institute of Technology, Rourkela 2007, PP.2-47.

[6]. Spartan-II and Spartan-IIE Libraries Guide for HDL Designs, PP.1-247.

BIOGRAPHIES:

Mr. Gore Shukracharya Sampatti.
He has completed M.E.(VLSI and Embedded Systems) from SITRC -Nashik under Savitribai Phule Pune University, Pune. His major fields of studies are VLSI, Embedded Systems, Digital Signal Processing, ITCT and ASIC's.
He has presented and published many papers in national and international journals/Conferences.
Currently he is working as a Assistant Professor in E&TC department at MCOE&RC-Nashik.

	**Mr. Lokhande Abhijit Ashok.** Student of third year E&TC of MCOE&RC- Nashik under Savitribai Phule Pune University, Pune. His major fields of studies are VLSI, Embedded Systems, Digital Signal Processing, ITCT, Signals and System.
	**Mr. Mahajan Sachin Bapu.** Student of third year E&TC of MCOE&RC- Nashik under Savitribai Phule Pune University, Pune. His major fields of studies are VLSI, Embedded Systems, Digital Signal Processing, ITCT .