Design of Control Module for Serial DAC Based on FPGA

Cai-hong Liu[a],Jin-shui Ji[b],Ai-qin Qi[c]

Northwest University for nationalities

Lanzhou, Gansu, China

[a]cindyliu888@qq.com , [b]jsjjs@xbmu.edu.cn, [c]27657032@qq.com

Keywords:FPGA,Serial, DAC,VHDL

Abstract: In order to increase the flexibility of control for serial DAC, a new control method for DAC based on FPGA is proposed in this paper. A state transition diagram can be drawn according to the timing diagram of DAC, Which can be realized in FPGA using Very High-speed Integrated Circuit Hardware Description Language. The simulate results show that logic in FPGA is consistent with the requirements. The module based on FPGA can be modified just by modifying software, not the hardware.

Introduction

The traditional control method for serial DAC is completed by single chip machine. Advantages of this traditional method are simple program and flexible control. Shortages of this traditional method are long period and slow. The TLC5615 is a 10-bit voltage output digital-to-analog converter with a buffered reference input. The DAC has an output voltage range that is two times the reference voltage, and the DAC ismonotonic. The device is simple to use, running from a single supply of 5 V. A power-on-reset function isincorporated to ensure repeatable start-up conditions. The paper is introduced a new method for serial DAC based on FPGA.

Features of Fpga

A. structure of FPGA

FPGAs are programmable digital logic chips. What that means is that you can program them to do almost any digital function. FPGAs are built from one basic "logic-cell", duplicated hundreds or thousands of time. A logic-cell is basically a small lookup table ("LUT"), a D-flip flop and a 2-to-1 multiplexer. The LUT is like a small RAM that can implement any logic function. Each logic-cell can be connected to other logic-cells through interconnect resources (wires/ multiplexers placed around the logic-cells). Each cell can do little, but with lots of them connected together, complex logic functions can be created. The interconnect wires also go to the boundary of the device where I/O cells are implemented and connected to the pins of the FPGAs[1].

B. The general workflow when working with FPGA

- You use a computer to describe a "logic function" that you want. You might draw a schematic, or create a text file describing the function, doesn't matter.
- You compile the "logic function" on your computer, using software provided by the FPGA vendor. That creates a binary file that can be downloaded into the FPGA.
- You connect a cable from your computer to the FPGA, and download the binary file to the FPGA. FPGAs lose their functionality when the power goes away (like RAM in a computer that loses its content). You have to re-download them when power goes back up to restore

This work was supported by Key Program of Research Foundation For Young Scholars under Grant No. zyz2012080.

Published by Atlantis Press, Paris, France.

the functionality[2].

Analysis the Timing of Dac

The converter is partitioned into 4 major sections: the control logic, digital -to- analog converter ,DAC register, and the shift register.

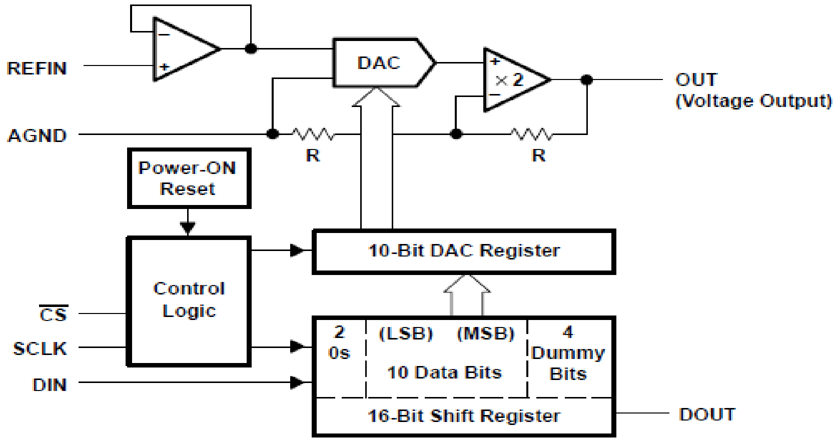

Fig 1. Functional Block Diagram

The TLC5615 uses a resistor string network buffered with an op amp in a fixed gain of 2 to convert 10-bit digital data to analog voltage levels[3]. When chip select is low, the input data is read into a 16-bit shift register with the input data clocked in mostsignificant bit first. The rising edge of the SLCK input shifts the data into the input register. The rising edge of CS then transfers the data to the DAC register. When CS is high, input data cannot be clockedinto the input register. All CS transitions should occur when the SCLK input is low. If the daisy chain function is not used, a 12-bit input data sequence with the MSB first can be used as shown in Figure 2:

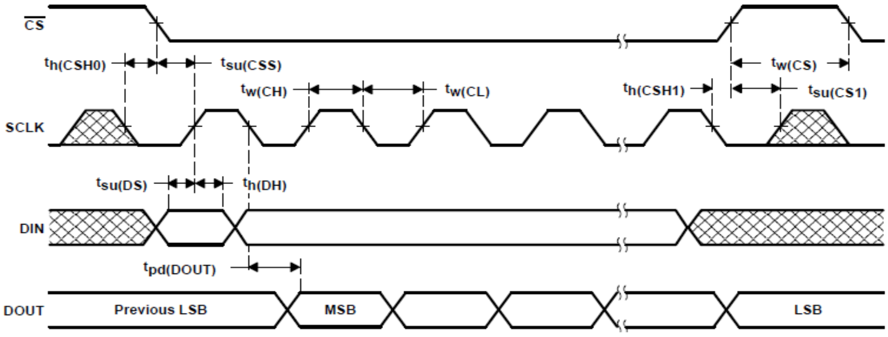

Fig 2. Timing Diagram

Realization of Control Module Based on FPGA

A schematic diagram based entry for the desired circuit becomes very difficult to use this method for a large design with hundreds of primitive gates. Hardware description languages provide standard text based expressions of the structure and behavior of digital circuits. Therefore HDL languages are nowadays the preferred way to create FPGA designs.

VHDL is a computer programming language designed to illustrate the behavior of field-programmable gate arrays and application-specific integrated circuits of digital systems in electronic design. VHDL describes the performance of electronic components in many areas. VHDL is used to describe precise aspects of electrical circuit behavior in order to create a VHDL simulation model. Incorporated with schematics, block diagrams and system-level VHDL descriptions, the VHDL simulation model can be used as the foundation for building larger circuits[4].

A model consisting of a set of states, a start state, an input alphabet, and a transition function that maps input symbols and current states to a next state. Computation begins in the start state with an input string. It changes to new states depending on the transition function. Nine states are defined in this VHDL program. As shown in Fig 3.

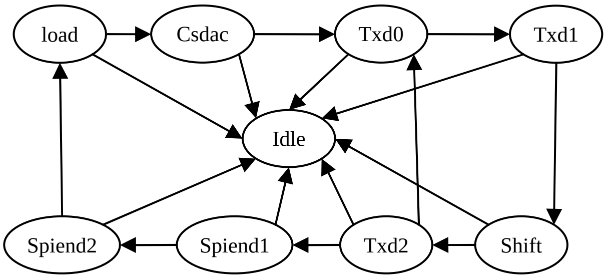

Fig 3. State transition diagram

A part of program is process for state convertion:

```
process( Reset,clkdiv4)
begin
  if(Reset='0')then
     bit_cnt<="0000";
     state<=idle;
     loadH<='1';
     wait_on<='0';
  elsif(clkdiv4'event and clkdiv4='1')then
  case state is
  when Idle =>State<=load;
  when Load =>State<=Csdac;
              bit_cnt<="0000";
                   if(loadH='1')then
                          shift_reg<=DAC_HIGH;
                   else
                          shift_reg<=DAC_LOW;
                        end if;
  when Csdac =>State<=Txd0;
  when Txd0  =>State<=Txd1;
               bit_cnt<=bit_cnt+1;
  when Txd1  =>State<=Shift;
  when SHIFT =>State<=Txd2;
       for i in 11 downto 1 loop
        shift_reg(i)<=shift_reg(i-1);
        end loop;
        shift_reg(0)<='0';
  when Txd2  =>if(bit_cnt="1100")then
                                State<=Spiend1;
                       else
                                State<=Txd0;
                       end if;
  when Spiend1 => State<=Spiend2;
                  wait_on<='1';
  when Spiend2 =>if(wait_cnt="111")then
                                       State<=Load;
                                     wait_on<='0';
                                     loadH<= not loadH;
                            end if;
  when others => State<=Idle;
```

```
    end case;
    end if;
  end process;
```

Conclusion

When chip select is low, the input data is read into a 16-bit shift register with the input data clocked in most significant bit first. The rising edge of the SLCK input shifts the data into the input register. The rising edge of CS then transfers the data to the DAC register. When CS is high, input data cannot be clocked into the input register. All CS transitions should occur when the SCLK input is low. When reset (Reset) is high and chip select (CS) is low, The most significant bit of shift register moves left one bit on the rising edge of clkdiv4. And the most significant bit of shift register outputs one bit on the rising edge of dac_sclk.when parallel data is latched in shift register,the controller will convert parallel data into serial data. As shown in Fig 4.

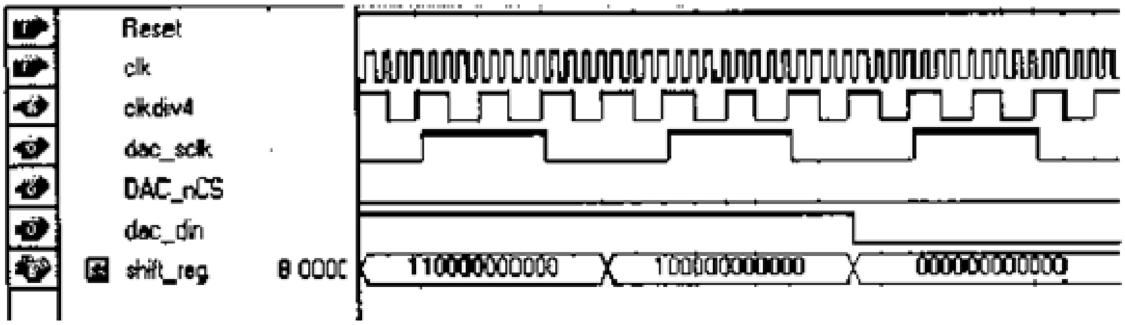

Fig 4. Simulation results

A new control method for DAC is achieved in FPGA. Advantages of this new method are simple programming method for FPGA. Modules based on FPGA can be modified by just modify software, don’t modify the hardware. Therefore, a new control method for DAC based on FPGA is an effective method.

References

[1] X.Zhang,H.Li,Q.Wang, “Researches on 1553 based semi-product-simulation technology,” in Computer Science & Education (ICCSE), 2012 7th , Melbourne, VIC,2012, pp. 382 - 385 .

[2] A.S.Molahosseini, S.Sorouri, A.A.E.Zarandi, “Research challenges in next-generation residue number system architectures,” in Computer Science & Education (ICCSE), 2012 7th , Melbourne, VIC,2012, pp. 1658 - 1661 .

[3] C.Wang,H.Xu,X.Mao,“Construction of hardware curriculum group for transition from network to Internet of Things engineering major,” in Computer Science & Education (ICCSE), 2012 7th , Melbourne, VIC,2012, pp. 1575 - 1579 .

[4] M.S.Bascil, I.Yazici, F.Temurtas, “A FPGA based digital design training platform” in Information Science and Technology (ICIST), Singapore,2011, pp. 1384 - 1389.

[5] M. Yusufu, “Comparative study of formal specifications through a case study” in Information Science and Technology (ICIST), Hubei,2012, pp. 318–321.