An FPGA Configuration Circuit Based on JTAG

Huiguo Zhang, Yulan Tang
School of Information Engineering
Jiangnan University
Wuxi, Jiangsu 214000, P.R. China
hweigo@yahoo.com.cn

Zongguang Yu
No. 58 Research Institute
China Electronics Technology Group Corporation
Wuxi, Jiangsu 214000, P.R.China

***Abstract*—An FPGA configuration circuit based on JTAG is designed. The configuration circuit has the JTAG architecture which is compatible with the IEEE standard 1149.1. Under the shift function of JTAG, a data chain for configuration is provided. The process of the configuration is controlled by three simple counters. Implemented with the CSMC 0.5um technology, the configuration circuit has the area of 1.404mm^2, occupies the 3 percent of total FPGA area. Compared with the control scheme of complex state machine, this design is simple and has been used in an FPGA.**

***Index Terms*—FPGA，Configuration circuit，JTAG，CRC**

I. INTRODUCTION

FPGAs are general purpose programmable devices that are customized by the end users. The basic device architecture of an FPGA consists of an array of configurable logic blocks(CLBs) embedded in configurable interconnect structure and surrounded by configurable I/O blocks (IOBs). The function of the CLB and IOB is depending on the contents of the corresponding configuration memory cell. And the interconnect structure includes programmable interconnect points that control the connection of wiring segments in the FPGA. The interconnect point can be a transistor controlled by the configuration memory cell.

Configuration is the process of loading a stream of bits containing the program data into the configuration memory cells which control the configurable logic blocks, I/O blocks and interconnect points of the FPGA. There are many modes for FPGA configuration, among which JTAG mode is the basic method and is provided in all commercial FPGAs. FPGA vendors such as Xilinx, Altera have already designed their configuration circuit. However, the commercial FPGAs use the complex state machines to control the configuration process[1-4]. In this paper, we introduced a new control scheme for FPGA configuration and designed a simple configuration circuit for an actual FPGA.

The remainder of the paper is organized as follows: Sec. 2 presents the JTAG architecture .Sec. 2 and Sec. 3 deal with the configuration control structure and CRC check module. The verification and physical implementation of configuration circuit is given in part 5. Finally Sec. 6 gives the conclusions.

II. PROCESS OF JTAG CONFIGURATION

JTAG circuits for FPGA are compatible with the IEEE standard 1149.1 which was derived from a proposal by the Joint Test Action Group. JTAG has a finite state machine which has sixteen states. Figure.1 is the state transition diagram of Test Access Port (TAP) controller[5,6].

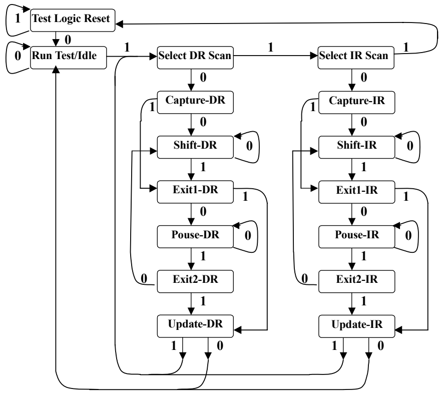

Figure 1. TAP controller state diagram

The state machine consists of three state cycles. The master state cycle determines the TAP state of Reset, Idle, Data operation or Command operation. Two slave state cycles indicate the steps of the command operation and data operation respectively. By utilizing the Test Access Port(TAP), FPGAs containing boundary-scan have the capability of driving or observing the logic levels on I/O pins. TAP has four dedicated pins: Test Clock(TCK), Test Mode Select(TMS), Test Data Input(TDI) and Test Data Output(TDO). TCK is the clock input of JTAG, the input data are valid at positive edge of TCK. TMS controls the state transition of TAP, the value shown adjacent to each state transition in the figure represents the signal of TMS at the rising edge of TCK. Whenever TMS keeps 1 in five continuous TCK cycles, TAP will go to the reset state. TDI is the data input terminal, and provides data for Instruction Register (IR) and Data Register (DR).TDO output the data at the negative edge of TCK.

In configuration mode, TMS first keeps 1 for five continuous TCK cycles, and reset the TAP. TAP select the command operation in master state cycle, configuration word is then fed into instruction register(IR) through TDI. Afterwards, TAP return to the idle state of master cycle, forward to the data operation cycle and dealing with the data register according to the command word of instruction register. Finally, TAP return to the reset or idle state for the new procedure.

978-1-4244-2800-7/09/$25.00 ©2009 IEEE

Figure.2 is the simplified diagram of JTAG architecture for configuration. Here we designed the JTAG TAP controller, which has the 16 states and can generate the command control signals. When the command load into the instruction register (IR), the IR DEC module decodes the command to control the corresponding data register and select the data chain respectively.

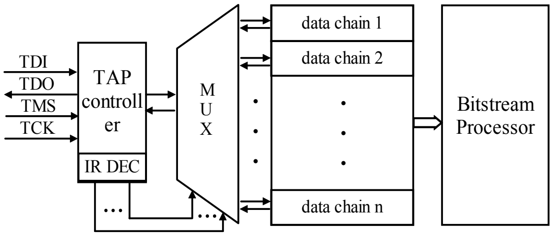

Figure 2. JTAG Architecture for Configuration

We defined the specific command CONFIGURE for the configuration of FPGA. When power is on, the boundary scan is enabled and the CONFIGURE command is shifted into the command register. And the TAP controller goes into the Update-IR state and updates the CONFIGURE instruction. Then TAP controller goes into the Shift-DR state, and the bitstream is shifted into the FPGA datachain through TDI. Then the bitstream is processed and configured into the configuration memory.

III. Configuration Architecture

Though the configuration memory is distributed through the FPGA physically, they are logically connected together as the symmetric standard SRAM arrays. The array of configuration memory cells is shown in Figure.3 with the control logic and configuration cell array. In the actual FPGA, there are more than 100,000 configuration cells. Here only the 4-bit by 4-bit array cells are shown for clarity.

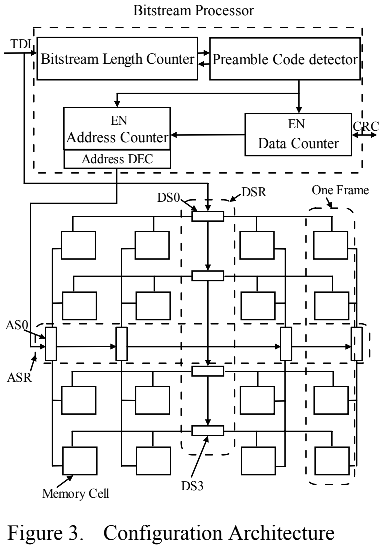

Figure 3. Configuration Architecture

In configuration, the bitstream will be shifted into data shift register(DSR) under the control of JTAG TCK. When a frame of data(4bits for example) has been shifted into bit positions DS0 through DS3 of the data shift register DSR, the control logic will produce a strobe signal to select a column of configuration cell. The data frame is then shifted in parallel into the column of selected configuration memory cell. The strobe signal is produced by the AS0 to AS3, only one strobe signal is valid at a time. When the first frame comes, AS0 has the high valid signal thus strobe the first column; Then this signal is shifted into AS1 to strobe the second column when the second frame is ready. In the same manner, FPGA will finish the configuration process.

The shifting of frame data in DSR and strobe signal in ASR is controlled by the control logic. The specific format of bitstream is corresponding with the design of control logic. The bitstream data format used in our FPGA is shown in TABLE Ⅰ. The data frame has the length of 182 bits which is determined by the length of DSR and can be varied depending on the FPGA array size. Preamble code is used for indicating the start of configuration data. Fill bits are used for synchronous of the data frame. Also, the 16-bit CRC checksum data is produced in the data stream. The total length of bitstream is indicated by 24-bit data length.

TABLE I. Bitstream Data Formats

Data type	Bit Amount	Modes
Preamble code	4	0010b
Length Count	24	COUNT(23:0)
Fill bits	4	1111b
Data Frame	182	DATA(n-1:0)
CRC	16	XXXX

The configuration architecture is mainly consisted of three counters: bitstream length counter, address counter and data counter. The bitstream processor let the bitstream flow through the 28 serial connected flip-flops, the first 24 flip-flops act as a bitstream length counter. They record the total length of the bitstream. The last four flip-flips act as a preamble detector. They watches the input stream, when the 0010b pattern occurs, the detector outputs an active high signal immediately. This signal feed back to the length counter and cause the counter latch the 24 length bits. Then the counter holds on in the next four clock cycle to skip over the next four fill bits. Afterwards, the detector outputs the enable signal to the address counter and data counter. In the next clock, the length counter begins to count down. The address counter and data counter count up from zero. When the data counter starts, it issues a CRC signal to enable the CRC circuit. When the data counter counts to the frame length, the address counter increases by one. When the address counter reaches the max frame number, it checks the error output bit of the CRC. If there has any error, the configuration process will be restarted. If there has no errors, control logic outputs a DONE signal. Then the FPGA starts up and becomes operational.

IV. CRC check module

Because the functions performed by the logic or I/O blocks are determined by the values of the configuration memory cells, any error in the values could affect the functions and render the design inoperative. In configuration process, we design a CRC module for error detector. Cyclic codes possess attractive properties for digital error detection circuit construction. Their inherent algebraic structure enables simple coding operations and decoding algorithms. And their error detection capability is nearly independent on the number of code word information bits[7].

Figure.4 is the structure of the CRC check module. It is a simple bit-serial architecture which is a kind of linear feedback shift register (LFSR). When the data counter begins to count, the CRC check module is cleared with the rst signal. Then the bitstream is fed into the CRC check module as well as fed into DSR. After the address counter reaches max frame number, the CRC check module compares the produced check bits with the pre-calculated bits in bit stream. While the last check bit is read and the output of CRC check module is active high, it means the frame has CRC error. Otherwise there’s no CRC error.

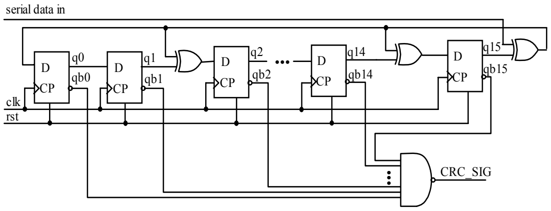

Figure 4. CRC Check Module

V. Design Verification and Physical Implementation

In order to verify the design, the Verolog-XL is used in the simulation. We use the C language generate the random configuration data with the 16 bits CRC checksum. The data chain length (frame size) is 182 bits, address chain is 576, with 4-bit preamble code and 4-bit fill bits, and total bitstream is 182×576+48 = 104880 bits. The configuration circuit reads the generated bitstream file and produces the expected result.

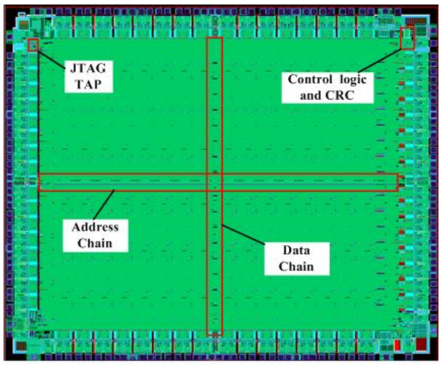

Figure 5. Circuit Layout location

The layout of configuration circuit consists of four parts: JTAG TAP circuit, data chain, address chain, control logic and CRC. Their location is shown in Figure 5. JTAG TAP controller is on the top-left. The configuration logic and CRC is on the top-right. In the middle of FPGA, data chain lies vertically across the FPGA and address chain lies horizontally.

The FPGA is designed with the CSMC 0.5um single-poly-3-metal process. The area of each part of the configuration circuit is shown in the TABLE Ⅱ. The total area of configuration circuit is 1.404 mm^2 which occupies about 3 percent of the whole FPGA area of 45.5 mm^2.

TABLE II. Area of Layout

Circuit	Area(mm^2)
JTAG TAP circuit	0.010×0.015
Control logic and CRC	0.015×0.030
Data chain	0.095×6.0
Address chain	6.845×0.110
FPGA(whole)	7.195×6.325

VI. Conclusions

We have designed and implemented a new JTAG configuration circuit for FPGA. When the frame data is downloaded into the configuration cell, a typical CRC check module is implemented for reliability. The main configuration process is controlled by three counters. Compared with the commercial FPGAs which use several complex state machines for configuration process, our approach is simple and effective, which occupies only 3 percent of the FPGA area.

Acknowledgments

This work was supported by the Jiangsu Natural Science Foundation of China under the grant number of BK2007026.

References

[1] David P.Schultz, Steven P. Young and Lawrence C. Hung, “Method and Architecture for Reading, Modifying and Writing Selected Configuration Memory Cells of FPGA” US. Patent, Ser. No. 6,255,848 B1 ,Jul 3 ,2001.

[2] Roman Iwanczuk, LosGatos and Steven P. Young. “Architecture and method for loading wide frames of data from a narrow input bus”. US. Patent, Ser No. 6,137,307.Oct 24, 2000.

[3] Yabin Wang, Yuan Wang and Jinmei Lai, “An FPGA Configuration Circuit Used for Fast and Partial Configuration”. The 7th International Conference on ASIC, 2007, pp.157-160.

[4] J. Wang, L. Chen and J. Lai, "FPGA Downloading Circuit Design and Implementation”. The 8th Int. Conf. on Solid-State and Integrated-Circuit Technology, 2006, pp.1950-1953.

[5] IEEE Standard Test Access Port and Boundary-Scan Architecture. IEEE Std 1149. 1- 2001, 2001, pp. 18-32.

[6] Xilinx Corporation .Application Note Xapp139. “Configuration and Readback of Virtex FPGAs Using JTAG Boundary-Scan v1.7”. Feb 14, 2007. http:// www.xilinx.com.

[7] Sprachmann, M. “Automatic generation of parallel CRC circuits”. Design & Test of Computers. 18(3),2001, pp. 108-114.