International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 11 (2015) pp. 29621-29630
© Research India Publications
http://www.ripublication.com

Design of Programmable JTAG Controller In FPGA

[1]Ramesh Bhakthavatchalu, [2]Saranya K Kannan
[1,2]Dept. of Electronics and Communication Engineering,
Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Kerala, India
[1]rameshb@am.amrita.edu
[2]saranyakannan20@gmail.com

Abstract

JTAG is one of the most powerful structured DFT technique used in VLSI industry that provides testing and debugging at chip, board level and system level. Besides testing, it can be utilized for some other non-testing purposes such as providing access to the internal components of the device. JTAG controller becomes the integral part of every VLSI IC chip. Here a programmable JTAG controller is proposed and implemented in FPGA. This scheme is employed in all the benchmark circuits of ISCAS_89 design and their performances were evaluated. All the designs are coded in verilog and simulated in Model sim DE 6.5e and synthesized in Xilinx Virtex-5 FPGA technology.

Keyword: Programmable JTAG, TAP controller, SOCs, Xilinx, FPGA

Introduction

Testing plays a very important role in VLSI (Very Large Scale Integration) industry. Every device must be tested before introducing it into the market. Different testing techniques based on different procedures and algorithms are used for this purpose. Structured DFT (Design For Test) techniques such as scan chain, BIST (Built In Self Test) and boundary scan were commonly used. It reduces the dependency on external ATE (Automatic Test Equipment) requirements and thereby reducing the test length and cost to a large extent. Testing of single integrated circuitry is done using the conventional bed-of-nails methods. But with emerging of SOCs (System on Chip), testing difficulties using this conventional methods also rises. SOC is an integrated circuit that contains a number of modules such as IPs (Intellectual Property), memory modules, ASIC cores, DSP modules etc manufactured by different vendors. Different testing techniques are used for testing these modules. Traditional bed-of nails methods were no longer able to control and observe all the I/O pins. The problem was solved by introducing boundary scan or JTAG (Joint Test Action Group) controller.

Boundary scan is a test methodology that used for solving a set of testing problems in the devices ranging from chip to system level, from cores to interconnect between cores, etc [1]. JTAG controller becomes one of the inevitable parts of most of the current day IC chips and it is compatible with IEEE 1149.1 standard. In addition to testing, JTAG can be extended to perform some other non-testing purposes. For example JTAG port is used for loading configuration bit streams of reconfigurable device onto FPGA (Field Programmable Gate Array) devices. In SOCs, JTAG controller becomes a powerful test mechanism for initiating and controlling the internal BIST features [2]. JTAG provides excellent testing features without increasing system complexity.

Presence of Boundary scan registers and TAP (Test Access Port) controller make a chip JTAG enabled. JTAG contains a number boundary scan cells connected between its I/O pins and the logic to be tested. These boundary scan cells are connected in a shift register form called boundary scan register with a serial input TDI and serial output TDO. JTAG can have daisy chain topology in which several chips in a board shares same TAP: TMS, TCK and TRST.

In this paper a hardware programmable JTAG controller in FPGA is designed and implemented. The proposed scheme can be applied to number of designs. Here we can control the size of the TAP controller according to the design in which it is used. It adds only small area overhead to the design and reduces the complexity of the entire device.

Section 2 discuss about the existing works and modification on the JTAG controller. Section 3 gives an overview of the existing JTAG architecture. Section 4 gives details about the proposed programmable JTAG architecture. Section 5 shows the result obtained after implementing the proposed method in all ISCAS_89 benchmark circuit designs. In section 6 concludes the paper by evaluating the performance of proposed architecture on various designs.

Literature Survey

Rapid advancement in the VLSI technology leads to the replacement of IC chips with the SOC designs. Hence JTAG controller associated with these type of designs were also been expanded in their complexity. JTAG controller becomes one of the inevitable parts of every IC in VLSI industry. So one must ensure that JTAG controller functions properly. JTAG should be tested before using it for testing and debugging purpose. Many different implementation schemes and features of the boundary scan architectures for digital VLSI designs were explored. Various debug JTAG port architectures used in the modern surface mount technology and SOC designs is illustrated in [3]. A JTAG testing technique based on CED (Concurrent Error Detection) method that simplifies the testing procedure is discussed in [4]. In [5] JTAG interface controller is implemented as a RISC microprocessor has been discussed. A JTAG controller with enhanced features such as on-chip real time debug, test management and boundary scan test were discussed in [6]. In [7], discussed about the reuse capability of the TAP controller. A high speed reduced pin count JTAG interface between the JTAG controller and target design is described in [8]. Interface achieves high speed by using of SBT (Simultaneously Bi-directional Transceiver) technology. A scheme that make uses of boundary scan technology for FPGA based experiments has been introduced in [9].

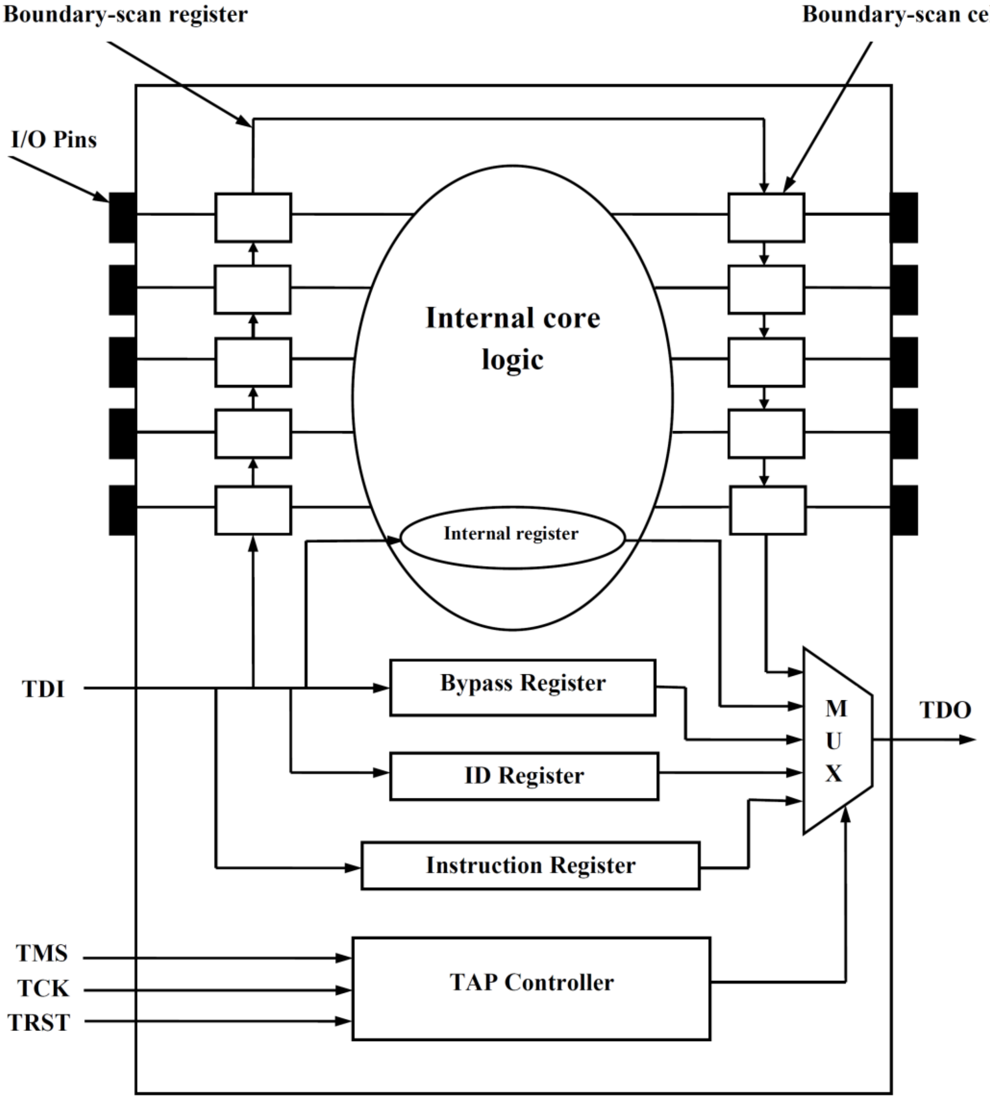

Figure 1: Standard JTAG Architecture [2, 10]

Standard JTAG Architecture

JTAG is powerful test interface and boundary scan architecture for the digital ICs in boards or systems. Structure of a typical JTAG architecture is illustrated in figure 1. It consists of mainly four hardware components: a Test Access Port (TAP), an instruction register and an associated decoder, a TAP controller, various test data registers (TDRs).

Test Access Port is a general purpose port that provides four basic I/O test pins: TDI (Test Data Input), TDO (Test Data Out), TMS (Test Mode Select), TCK (Test Clock) and one optional pin TRST (Test Reset Logic) for performing test operations in the devices. TDI is the serial input pin through which instruction and data are loaded into the IR (Instruction Register). TDO pin are serial output pin that shifts the contents of instruction or data register out. TCK is the universal test clock and is independent of the system clock. TMS is the control input to the TAP controller. TRST pin will asynchronously reset the TAP controller into the Test Logic Reset state and the system operates in normal mode.

Test Access Port is controlled by a FSM (Finite State Machine) called TAP controller. TAP controller coordinates and controls all the test operation in the JTAG. It consists of an instruction register that holds current instruction to be executed. The decoder associated with the IR will generate sufficient signals to select the appropriate data register. Various test data registers associated with the JTAG structure include two mandatory (Boundary scan registers and Bypass register) and several optional data registers (ID register). Boundary scan register is a collection of boundary scan cells connected in a shift register form. Bypass register is a 1 bit register that bypasses a chip in a device that is not included in the current test operation and thereby reducing test length and hence reduces the test time. It also contains an optional device ID register that is used to store some device related information such as its part number, version number, etc. JTAG controller executes different types of instructions such as

Mandatory-EXTEST, BYPASS, SAMPLE AND PRELOAD

Optional – INTEST, USRCODE, IDCODE, RUNBIST, CLAMP and HIGHZ

Based on the instruction loaded into the instruction register, the data register is selected and placed between the TDI and TDO pin.

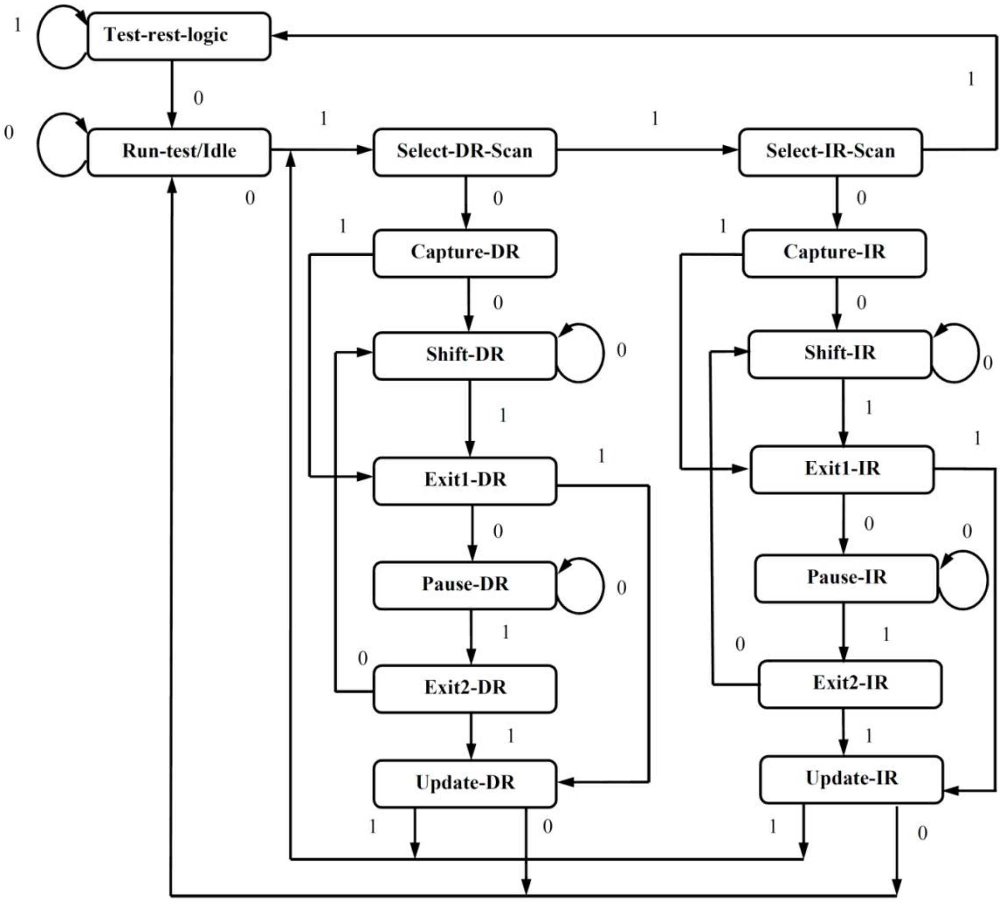

Figure 2: State diagram of TAP Controller [10]

TAP Controller

TAP controller is a Moore type Finite State Machine (FSM) with 16 states. The state diagram of the TAP controller is shown in figure 2. It consists of an input control signal that controls the test operations called TMS. The next state is determined by the value on the TMS input. State transition occurs only at the rising edge of the TCK signal. TAP controlled produces nine primary output signals. They are TCK, reset, enable, clock-DR, shift-DR, update-DR, clock-IR, shift-IR and update-IR. Since it is a Moore type FSM, its primary output depends only on its present state. TAP controller consists of two separate parts each with seven states. One section is used for controlling the data register operations and other for controlling instruction register operations. TAP controller enters test-logic-reset state synchronously by applying logic value 1 to the TMS for five consecutive TCK cycles. Various states included in data register operations are Select-DR, Capture-DR, Shift-DR, Exit1-DR, Pause-DR, Exit2-DR and Update-DR.

Proposed Programmable JTAG Architecture

Most of the VLSI IC contains independent JTAG module embedded in it for testing and debugging purpose. Usually JTAG module is designed for each digital VLSI circuit as part of the design. Since the boundary scan structure is an IEEE standard and all features are implemented based on the standard, a programmable JTAG controller is proposed here. Here a Single JTAG controller is designed and made it programmable at the Verilog HDL (Hardware Description Language) level to be used in wide varieties of designs. In addition to mandatory instruction, programmable JTAG controller supports additional instructions called PRIVATE-X instructions. This proposed programmability scheme is compatible with the IEEE 1149.1 standard. In comparison to the typical JTAG controller designed separately for each IC, proposed programmable JTAG controller helps in reducing the area overhead on the device and hence reducing the testing time and cost to a large extend. Also it helps in maintaining the uniformity of the structure of the JTAG module synthesized using the CAD (Computer Aided Design) tool for every new design in both ASIC and FPGA technologies.

Powerful feature in the Verilog HDL is the “parameter” feature. Whenever hardware size of circuit elements changes their entire code has to be modified whereas this feature helps to change the parameterized feature to modify just by modifying the parameter value. In verilog HDL this feature is defined in a module by using the keyword “**parameter**”. Parameter is a constant that is local within the module structure and it cannot be used as a variable. Parameter value can be changed or modified in two ways either by using **defparam** statement or by module instantiation. By defining parameter in a module makes it more flexible.

FPGA is basically a programmable or configurable IC (Integrated Circuit) that contains many logical components such as slices to implement various combinational and sequential logical functions. Each and every block in the FPGA can be programmed or configured by the user. An FPGA architecture consist of mainly three components: CLBs (Configurable Logic Blocks), IOBs (Input-Output Blocks) and interconnects. CLB forms the basic building block of an FPGA. Its features and number vary from device to device. Slices are grouped to form CLBs. An FPGA slices consists of no of LUTs (Look Up Table), flip-flops and multiplexers. Different FPGA families implement slices and LUTs in different manners. Signals between CLBs and to/from I/Os are routed through interconnects. FPGA grouped its I/O into banks with each bank supports different I/O standards independently. Virtex 5 FPGA family meets our design specifications. Virtex 5 that contains four LUTs with either 6-input - 1 output or 5input - 2 output, 3 user controlled multiplexer for combinational logics, two 1 bit adder with carry logic and four 1 bit registers that can be configured either as a flip-flops or as a latch. A LUT-FF pair consists of one LUT paired with one Flip-Flop in the slice. This feature helps in the optimized synthesis of the design.

Results and Analysis

Proposed Programmable JTAG controller is coded in verilog and simulated in Modelsim DE 6.5e. This programmable JTAG controller is applied to all the sequential benchmark circuits of ISCAS_89 design. All the design have been simulated in Modelsim DE 6.5e and implemented in Xilinx ISE Design suite 14.2, virtex5, XC5VLX110T and verified. Here we also analysed the power using Xilinx Xpower analyzer. No of PI, PO, slice registers, slices LUTs, power and speed of all the ISCAS_89 designs with associated pads are shown in table I. No of PI, PO, slice registers, slices LUTs, power and speed of all the ISCAS_89 designs after incorporating the JTAG controller and private instruction is shown in table II. No of slice LUTs utilizes by some of the benchmark circuits of ISCAS_89 designs without and with the addition of boundary scan and private instruction is depicted as a line graphs are shown in figure 3. A graph that shows the speed variation of some benchmark circuits of ISCAS_89 designs before and after incorporating the boundary scan and private instruction is plotted in figure 4.

Table 1: Details of FPGA Synthesis Result of All ISCAS_89 Designs With Pads

Design	**#PI**	**#PO**	**Slice registers**	**Slice LUT pairs**	**Fully used LUT-FF pairs**	**No of bonded IOBs**	**No of BUFG/BUFG controls**	**Power (W)**			**Speed (MHz)**
								Leakage	**Dynamic**	**Total power**	
s27	11	2	3	3	2	6	1	1.182	0.008	1.190	920.938
s298	10	7	14	16	14	10	1	1.183	0.038	1.221	376.400
s344	16	12	15	22	15	21	1	1.183	0.072	1.255	525.445
s349	16	12	15	22	15	21	1	1.183	0.072	1.255	525.445
s382	10	7	21	32	21	10	1	1.183	0.029	1.212	493.133
s386	14	8	6	34	6	15	1	1.183	0.021	1.203	580.080
s420	25	2	16	36	16	20	1	1.182	0.011	1.193	543.907
s444	10	7	21	28	21	10	1	1.183	0.029	1.212	526.621
s510	26	8	5	16	5	16	1	1.183	0.027	1.210	475.014
s526	10	7	21	26	21	10	1	1.183	0.024	1.206	376.400
s641	42	25	16	69	16	59	1	1.183	0.052	1.236	434.367
s713	42	24	16	65	16	58	1	1.183	0.050	1.233	431.658
s820	25	20	7	93	5	38	1	1.183	0.036	1.219	420.821
s832	25	20	6	89	6	38	1	1.183	0.039	1.222	420.932
s838	41	2	32	85	32	36	1	1.183	0.015	1.198	403.706
s953	23	24	26	139	26	40	1	1.183	0.047	1.230	315.498
s1196	21	15	18	136	18	29	1	1.183	0.020	1.203	459.517
s1238	21	15	18	149	18	29	1	1.183	0.016	1.198	343.637
s1423	24	5	74	148	72	23	1	1.183	0.020	1.202	226.560
s1488	15	20	6	136	6	27	1	1.183	0.074	1.258	382.914
s5378	42	50	160	294	113	85	1	1.184	0.099	1.282	373.232
s9234	43	40	126	240	97	68	1	1.184	0.082	1.266	249.635
s13207	70	153	399	527	286	214	1	1.185	0.169	1.354	186.555
s15850	84	151	433	547	315	227	1	1.186	0.247	1.433	195.284
s38417	35	107	1360	1644	750	135	1	1.189	0.406	1.594	213.881
s35932	42	321	1728	2426	1721	356	1	1.200	1.132	2.332	473.328
s38584	45	306	1230	1960	1094	342	1	1.191	0.543	1.734	207.009

Table 2: Details of FPGA Synthesis Result of All ISCAS_89 Designs With Pads, Boundary Scan and Private Instruction

Design	#PI	#PO	Slice registers	Slice LUT pairs	Fully used LUT-FF pairs	No of bonded IOBs	No of BUFG/BUFG controls	Power (W)			Speed (MHz)
								Leakage	Dynamic	Total power	
s27	11	2	59	71	58	11	3	1.183	0.077	1.261	421.852
s298	10	7	79	97	73	15	3	1.186	0.216	1.402	421.852
s344	16	12	96	118	85	26	4	1.188	0.359	1.547	421.852
s349	16	12	96	118	85	26	4	1.188	0.359	1.547	421.852
s382	10	7	86	110	81	15	3	1.186	0.218	1.403	421.852
s386	14	8	77	108	71	20	3	1.186	0.244	1.431	421.852
s420	25	2	86	116	85	25	4	1.183	0.063	1.246	421.852
s444	10	7	86	111	82	15	3	1.186	0.261	1.447	421.852
s510	26	8	88	109	81	32	4	1.187	0.307	1.494	421.852
s526	10	7	86	104	80	15	3	1.186	0.261	1.447	421.852
s641	42	25	149	198	125	65	5	1.194	0.741	1.935	374.315
s713	42	24	145	185	124	64	5	1.192	0.615	1.807	421.852
s820	25	20	114	186	98	43	4	1.193	0.703	1.897	414.805
s832	25	20	111	195	96	43	4	1.191	0.587	1.779	416.832
s838	41	2	118	194	117	41	4	1.183	0.071	1.254	394.174
s953	23	24	142	253	124	45	5	1.196	0.849	2.045	408.470
s1196	21	15	110	241	99	34	4	1.187	0.276	1.463	379.492
s1238	21	15	110	238	97	34	4	1.187	0.307	1.494	421.852
s1423	24	5	152	254	145	28	4	1.185	0.143	1.328	201.676
s1488	15	20	102	236	91	33	4	1.190	0.511	1.702	307.552
s5378	42	50	343	516	292	90	5	1.200	1.096	2.296	323.918
s9234	43	40	296	438	244	81	5	1.204	1.385	2.589	312.417
s13207	70	153	855	936	596	221	5	1.267	5.179	6.447	208.066
s15850	84	151	875	1149	693	233	5	1.224	2.584	3.807	204.251
s38417	35	107	1739	1958	1059	140	5	1.225	2.696	3.921	178.985
s35932	42	321	2453	3060	2147	361	5	1.372	10.963	12.334	421.852
s38584	45	306	1930	2656	1518	348	5	1.362	10.448	11.810	206.041

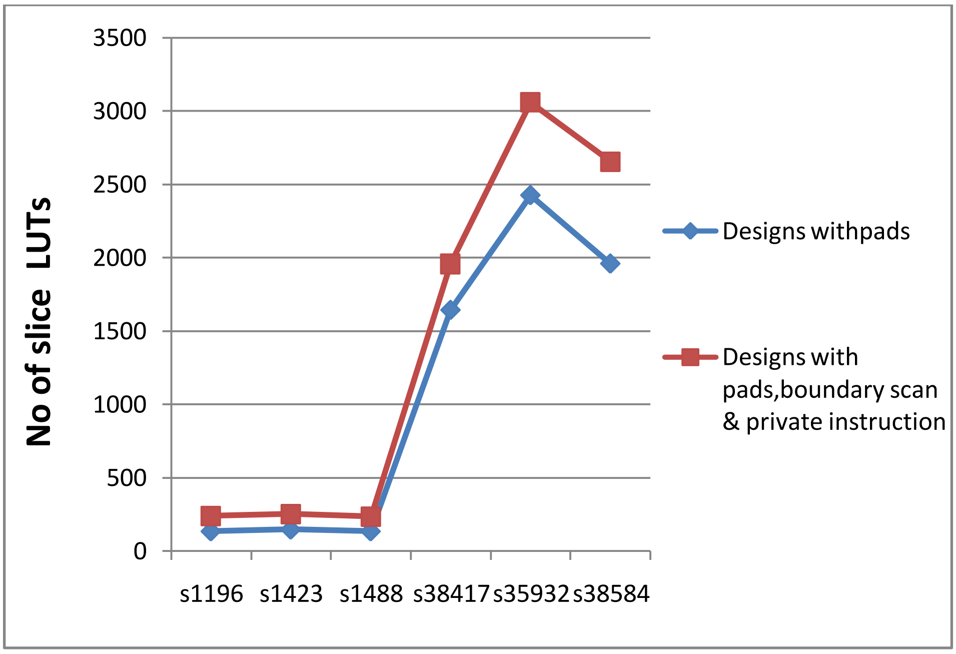

Figure 3: Graph showing the No of slice LUTs for some of the benchmark circuits of ISCAS_89 designs with the addition of boundary scan and private instruction

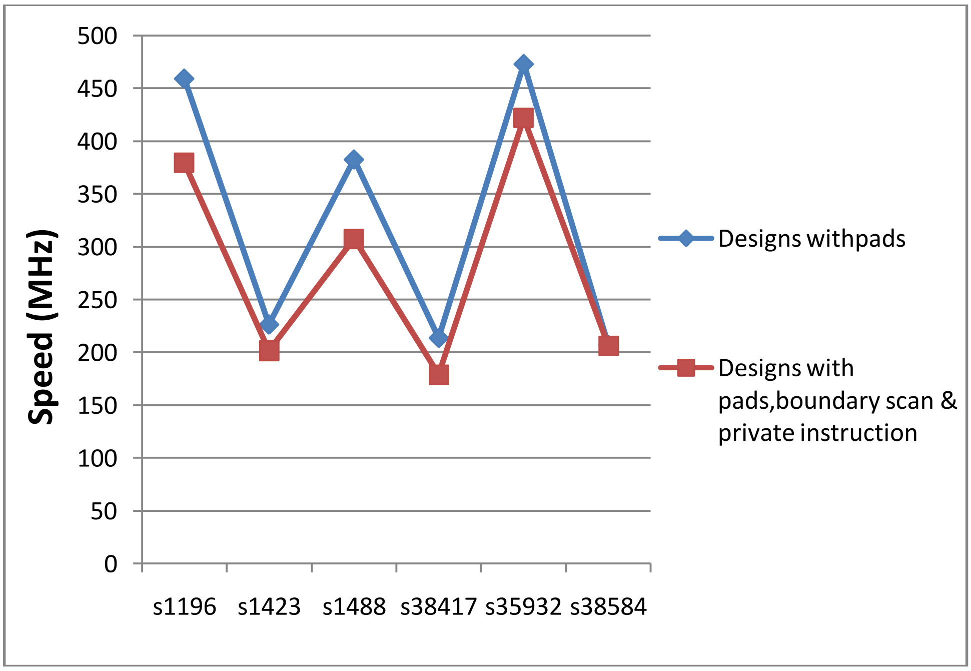

Figure 4: Graph showing the speed variation of some of the benchmark circuits of ISCAS_89 designs before and after incorporating boundary scan and private instruction

Conclusion

JTAG is a standard test interface present in many digital IC chips. Here a programmable JTAG controller is design and implemented in FPGA. It is then employed in number of designs with varying size. In addition to mandatory instruction, programmable JTAG controller supports an additional instruction called PRIVATE-X instructions. This scheme is well-suited with IEEE 1149.1 standard. It adds less area overhead to the entire design. It also reduces the testing complexity involved in SOC designs.

References

[1] Laung-Terng Wang, Cheng-Wen Wu, Xiaoqing Wen, 2006, “VLSI Testing principles and Architectures: Design for Testability,” chapter 10, 1st edition.

[2] Mohammad Tehranipoor and Cliff Wang, 2011, “Introduction to Hardware Security and Trust, Chapter 1,” Springer Publishing Company.

[3] Dimitry Akselrod, Arie Margulis, 2008, “Debug port controller architectures for system-on-chip integrated circuits,” Canadian Conference on Electrical and Computer Engineering (CCECE).

[4] Mitra, S., McCluskey, E J., Makar, S., 2002, “Design for testability and testing of IEEE 1149.1 TAP controller,” Proceedings 20th IEEE VLSI Test Symposium (VTS’02) pp: 247-252.

[5] Shen XuBang, Liang SongHai, 1993, “Design and Implementation of a JTAG Boundary-Scan Interface Controller,” Proceedings of the Second Asian Test Symposium.

[6] Guo Jian-min, Luo De-lin, 2009, “A functional enhancement methodology to JTAG controller in complex SOC,” Proceedings of 4th International Conference on Computer Science & Education (ICCSE).

[7] David B. Lavo, 2000, “A Good Excuse for Reuse: “Open” TAP Controller Design,” Proceedings of International Test Conference.

[8] Lee Whetsel, 2006, “A High Speed Reduced Pin Count JTAG Interface,” IEEE International Test Conference.

[9] Hui Zhao, Tiejun Xiao and Xiaoru Han, 2009, “The Application of Boundary-Scan Technology to FPGA-Based Experiment System,” 9th International Conference on Electronic Measurement & Instruments (ICEMI).

[10] Kenneth Parker, 2002, “The Boundary-Scan Handbook,” Kluwer Academic Publishers, 2nd edition.