DESIGN OF IEEE 1149.1 TAP CONTROLLER IP CORE

Shelja A S[1], Nandakumar R[2] and Muruganantham C[3]

[1]Department of Electronics and Communication Engineering, NCERC.
sheljaas@gmail.com
[2]Assistant scientist/engineer, N.I.E.L.I.T, Calicut
nanda@nielit.go.in
[3]Department of Electronics and Communication Engineering, NCERC
murugananthamc994@ncerc.ac.in

ABSTRACT

An implementation of IEEE 1149.1 TAP controller is presented in this paper. JTAG is an established technology and industry standard for on-chip boundary scan testing of SoCs. JTAG TAP controllers are becoming a delivery and control mechanism for Design For Test. The objective of this work is to design and implement a TAP controller IP core compatible with IEEE 1149.1-2013 revision of the standard. The test logic architecture also includes the Test Mode Persistence controller and its associated logic. This work is expected to serve as a ready to use module that can be directly inserted in to a new digital IC designs with little modifications.

KEYWORDS

IP core; IEEE 1149.1, TAP; TMP controller; JTAG; boundary scan; DFT

1. INTRODUCTION

A TAPC is probably the most common part used in support of on-chip testing. With the emergence of IJTAG many SOC design will be using multiple TAPC for accessing internal logic during test. TAPC is a part of the IEEE 1149.1 standard. IEEE 1149.1, the standard for test access port and boundary scan architecture is a common platform for device, board and system level testing. The standard, popularly known as JTAG was originally introduced in the year 1990 and it is now a well established technology in the industry. A TAPC is probably the most common part used in support of on-chip testing. The original motivation for JTAG was boundary scan testing. Boundary scan is a method for gaining direct control of IO pins of a circuit at boundary of chip during test. This enables efficient testing on interconnection between devices that are mounted on a circuit board. The JTAG control is not limited at just the boundary of device it can also be used to gain access to internal structures during the test of device itself. It provides low cost technique for functional, and in- circuit testing that does not requires the test system to have direct access to each node.

The architecture of IEEE 1149.1 boundary scan includes a Test Access Port (TAP) interface, TAP controller (TAPC) logic, Boundary Scan Registers (BSR), Instruction Register (IR) and Test Data Registers (TDR). The IR and TDR form separate scan paths arranged between the Test Data Input (TDI) pin and Test Data Output (TDO) pin. This architecture allows the TAP to select and shift data through one of the path without accessing the other. When the test logic is active only one register is connected between the TDI and TDO interface depending on the value at Test Mode Select (TMS) signal. TCK is the test clock and is not synchronized with the system clock. A significant advantage of JTAG is that it requires only a minimum set of test access pins as it uses a serial interface. It facilitates design reuse and provides a standard protocol on-chip testing.

But when this standard is used for boundary-Scan chained device is put into test mode where their I/Os are completely under control of the Boundary register content. But when the chained TAPs pass through the TLR state; these instructions are replaced with not test mode instructions. The I/O pins then revert to being connected to the internal device logic which will be in an unknown state. The results of this reconnection are unpredictable [3].

The 2013 revision of the standard consider this issue and suggest an optional controller for avoiding disruptions caused to the device. An attempt is made to study the effect of the new controller to the test logic of the standard.

2. TAP CONTROLLER

TAP controller is a synchronous machine which provides access to the device under test and controls the behaviour of test logic using its 4-wIRed interface. It is a 16 state FSM which generates clocks and control signals to the associated test logic. Figure 1 shows the top level architecture of TAPC. Test clock TCK and mode select signal TMS controls the operation of TAPC. An optional TRSTN pin may be used to asynchronously reset the test logic if required and it is active low. A reset of the test logic can also be achieved within five TCKs or less by setting the TMS input high.

The TAPC will be initialized to test logic reset state at the power up. State transitions occur on the rising edge of TCK based on the value of TMS. The FSM has two scan paths for data transmission: one for instruction scan and other for data scan. The state diagram includes six steady states: Test-Logic-Reset, Run-Test/Idle, Shift-DR, Pause-DR, Shift-IR, and Pause-IR. To load and execute a new instruction FSM control is moved to the Select IR-Scan state, from where, it moves through the various states, Capture-IR, Shift-IR, and Update-IR, as required. The last operation is the Update-IR operation and the instruction loaded into the shift section of the Instruction register is latched to the Instruction register to become the new current instruction. This causes the Instruction register to be deselected as the register connected between TDI and TDO and the Data register identified by the new current instruction to be selected as the Data register between TDI and TDO. From now, one can manipulate the data register with the generic signals; Capture-DR, Shift-DR, and Update-DR control signals. TCK can be stopped in either a high or low state without loss of data.

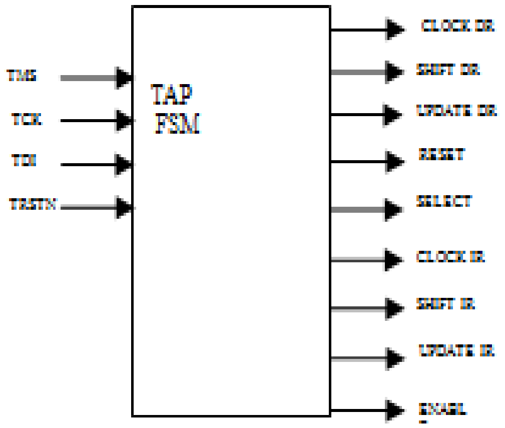


Fig 1. TAP top level

The functional table for the implemented example is given in table 1 with the encoding. The encoding used here is just an example. Different encoding schemes may be used.

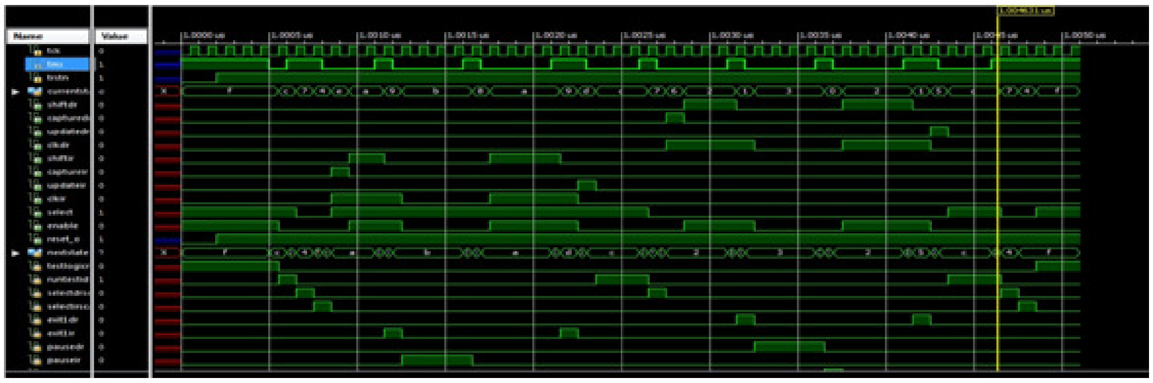
Table 1. Functional table of TAP

State of TAP	State encoding	Present state	Net state		Output								
			X=0	X=1	CLK DR	Shift DR	Update DR	CLK IR	Shift IR	Update IR	Reset	Select	Enable
Test logic reset	F	1111	1100	1111							0	1	0
Run test idle	C	1100	1100	0111							1	1	0
Select DR scan	7	0111	0110	0100							1	0	0
Capture DR	6	0110	0010	0001	1						1	0	0
Shift DR	2	0010	0010	0001	1	1					1	0	1
Exit1 DR	1	0001	0011	0101							1	0	0
Pause DR	3	0011	0011	0000							1	0	0
Exit2 DR	0	0000	0010	0110							1	0	0
Update DR	5	0101	1100	0111			1				1	0	0
Select IR scan	4	0100	1110	1111							1	0	0
Capture IR	E	1110	1010	1001				1			1	1	0
Shift IR	A	1010	1010	1001				1	1		1	1	1
Exit1 IR	9	1001	1011	1110							1	1	0
Pause IR	B	1011	1011	1000							1	1	0
Exit2 IR	8	1000	1010	1101							1	1	0
Update IR	D	1101	1100	0111						1	1	1	0

2.1. Simulation Results

The design is simulated using Modelsim simulator and the waveform obtained is shown in figure 2. The state transition and output signals of TAPC can be verified by applying the following exhaustive test pattern to the TMS input:

1011000100010000110001000010001000110011



An assumption is made that the signals applied to TMS and TDI change state on the rising edge of TCK. The time at which these signals change state is not defined by this standard. It is further assumed that the design does not include the optional device identification register. Therefore, the figures show the BYPASS instruction being set onto the output of the instruction register in the Test-Logic-Reset controller state. When TRSTN is asserted FSM is at TLR state(current state),encoded as F in table 1 and at each rising edge of TCK signal travel through subsequent states as per the IEEE specification with its corresponding outputs (shift IR through reset_o) asserted

Fig 2. Simulation waveform of tap controller

3. IEEE 1149.1-2013 REVISION

Since the last revision of the standard in 2001, the industry witnessed a drastic change in the IC technology. Many of these changes have been driven by design complexity and there are many devices available with programmable features including programmable IO behaviour [4]. Boundary scan testing put the device IOs in to test mode were their IOs are controlled by boundary register contents. The standard uses instructions like EXTEST and CLAMP for this. When the non-test instructions like BYPASS or IDCODE is encountered between the test mode instruction the TAP pass through TLR state and IO pins are revert back to functional mode [4]. These switching events are completely unsynchronised with current activities in the board, so that the internal logic of each IC may see completely illogical states [3].

The concept of “ready_to_test” was introduced in [3]. The IEEE 1149.1-2013 revision introduces such a concept to the standard where the device under test is place and hold in test mode till the testing process is over. And the optional initialisation instruction added keeps the device in safe mode when the test is over. The standard also includes other recommendations, most of them being optional and is summarized below.

Test-Mode Persistence (TMP) Controller (optional): New, optional, synchronous finite state machine which assert test mode regardless of active instruction

Electronic Chip Identification (ECIDCODE) (optional): The ECIDCODE instruction and associated ECID register instruction permits tracking the history of the component through its lifetime.

Initialization (optional): The problem of initializing a device for test has been addressed by providing a new, optional INIT_SETUP, INIT_SETUP_CLAMP, and INIT_RUN instructions paired with their associated initialization data and initialization status test data registers.

IC Reset (optional): Provide test control of system reset and related inputs through TAP.

Power domain control (optional): to support multiple power domains in a system having a single TDR, an optional standard TAP to TDR interface is recommended that allows for segmentation of test data registers. The concept of register segments allows for segments that may be excluded or include.

Procedural Description Language (PDL) (optional): a new executable description language to document test procedures unique to a component.

4. IP CORE ARCHITECTURE

The demand for more powerful products and the increasing capacity of today's silicon technology have moved the design methodology to the system abstraction level. The integration technology supports the integration of a complete system in silicon (System-on-chip) and design methodologies are more and more based on pre-defined and pre-designed Intellectual Property blocks (IP-core). The reusing of IP-cores has been an alternative to reduce the increasing gap between design productivity and chip complexity of emerging SoC designs [7]. [7] suggests a structured method for IP core design.

Figure 3 shows the revised test logic architecture. It includes the optional TMP controller (TMPC) and the associated TMP status register as per the specification in IEEE 1149.1-2013 revision. The IR used here is a four bit shift register. Only bypass register and TMP status register is the implemented data registers. Boundary scan registers and other optional registers are not implemented in this architecture.

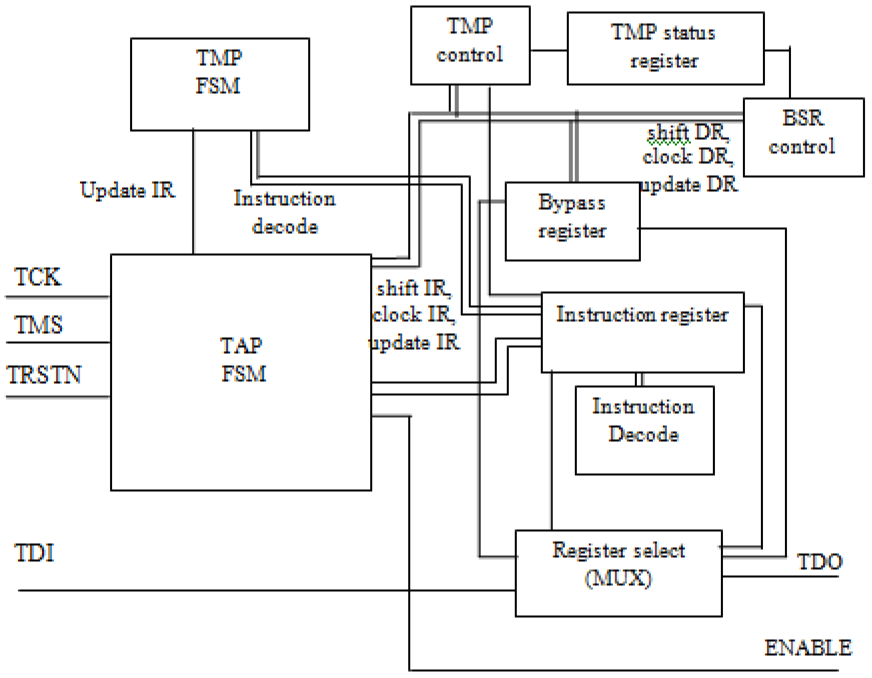


Fig .3 Conceptual schematic of the IP core

4.1. TMP Controller

TMPC is a synchronous state machine which keeps a device in test mode persistently Irrespective of the state of the remaining test logic. During board or system test TMP controller provides control over which components are in test mode and which are not, independent of the active instruction [1]

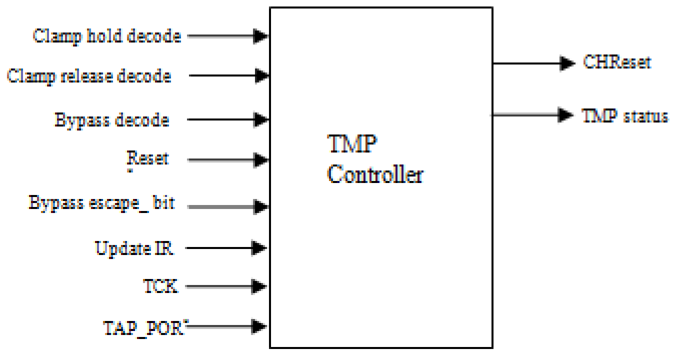


Fig 4. TMP controller top module

The TMP controller is an FSM, as shown in figure 5, with two states: persistence on and persistence off. State of the controller is determined by decode signals from instruction register and TAP controller output signals. Asynchronous reset options, either TRSTN or POR*, must be provided to reset the controller.

Table 2. TMP controller signals

Pin name	**Type**	**Signal origin**	**Function**
Clamp hold decode	In	From IR	To set TMP controller to persistence on mode
Clamp release decode	In	From IR	To set TMP controller to persistence off mode
Bypass decode	In	From IR	Sets the bypass escape bit to 1 together with the update signal
Update IR	In	From TAP	State of TAP which sets the bypass escape bit
Reset*	In	From TAP	Reset signal from TAP. Generate chreset
Bypass_escape	In	From TMP status register	Allows a component to escape test mode
TAP_por*	In	Input pin	Asynchronous on-chip reset at Power up
Tck	In	Input pin	Clock
Chreset*	Out	To boundary register	Reset signal to boundary registers
TMP_status	Out	To boundary register	Indicates State of TMP controller

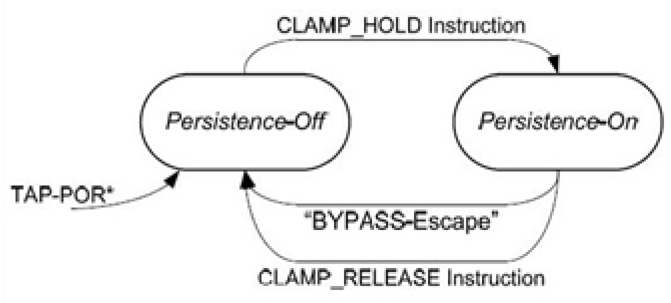


Fig 5. TMP controller state diagram

A possible implementation of TMPC is shown in figure 6. TMPC generates reset signals to control the boundary registers and other design specific registers. The TMPC together with the initialisation instruction allows safe switching between mission mode and test mode of system being tested.

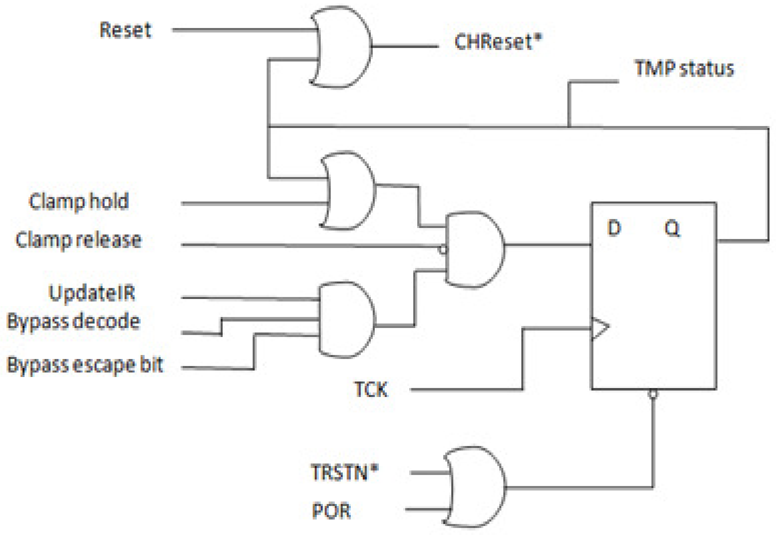
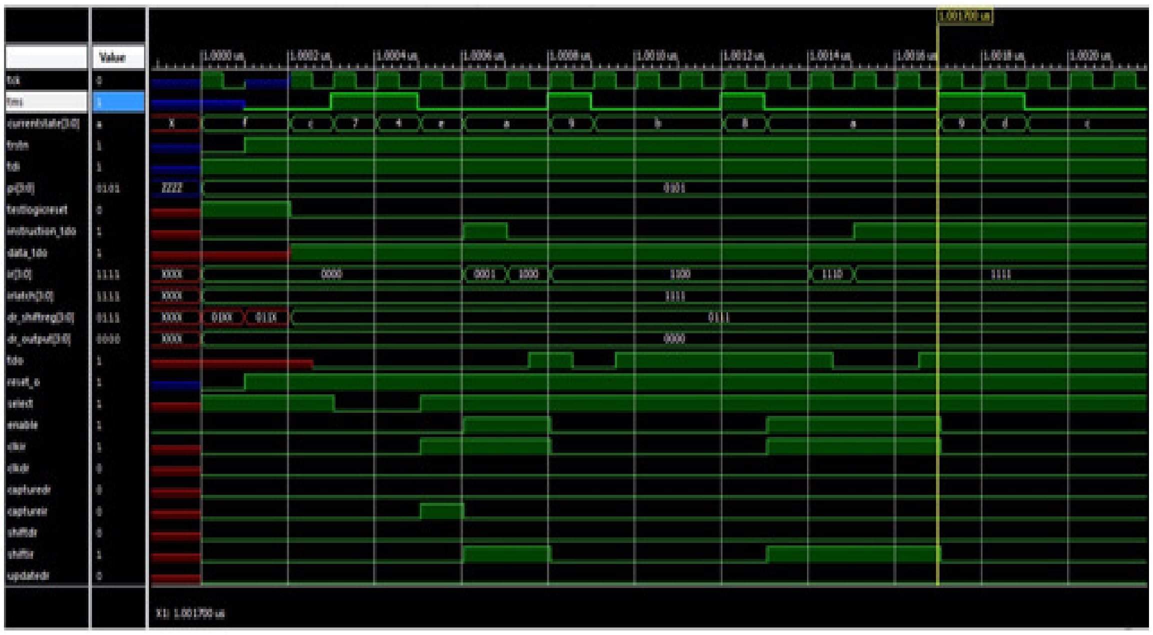


Fig 6. TMP controller example implementation

4.2. TAP Instruction Scan

Instructions maybe loaded in to TAP controller's IR by traversing the TAP state machine down to Update-IR state. At this state, the current contents of the register are shifted out (by TDO) as new value is shifted in. However, the value shifted out will always be fixed with 01 as last to bits, as mandated by the IEEE 1149.1standard for use in testing the functionality of the JTAG interface. The procedure for shifting a bit in and out of either shift register (IR or DR) is identical; the TAP state machine must be in the Shift-IR or Shift-DR state, respectively. The input bit (at TDI) is sampled by the TAP controller on the rising edge of each TCK cycle, while the output bit is driven out on the falling edge of the TCK cycle.

For each bit in the transfer, except the final bit, TMS must remain low. This is important so that, as each bit is shifted in and shifted out, the state machine remains in the Shift-IR



Instruction shift operation at IR shift register (ir [3:0]) currentstate [3:0] is at shift_ir state (b)

FIG 7. TAP IR scan

4.3. TAP Data Scan

Shifting values into and out of the DR of the TAP controller is performed in a similar manner to that of the IR. The simulation result of data scan is given in figure 8.

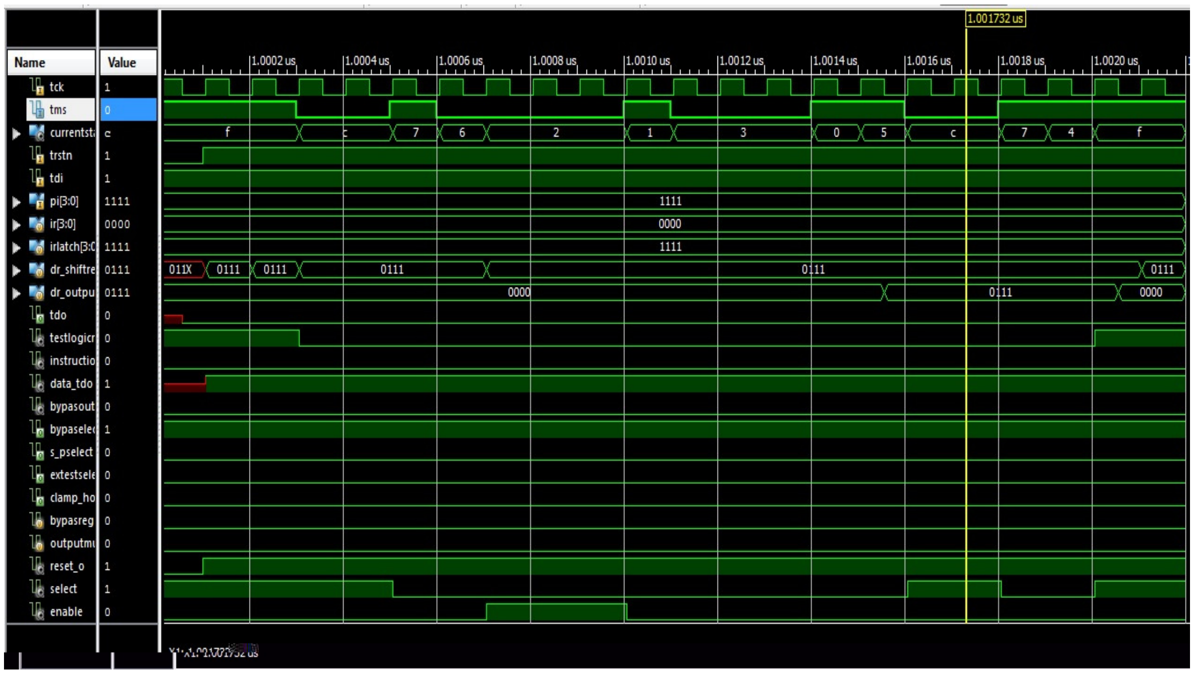


Fig 8. TAP DR scan

4.4. Simulation Results of Test Logic

The test logic is simulated using Xilinx ISE design suite and the result is shown in figure 9.

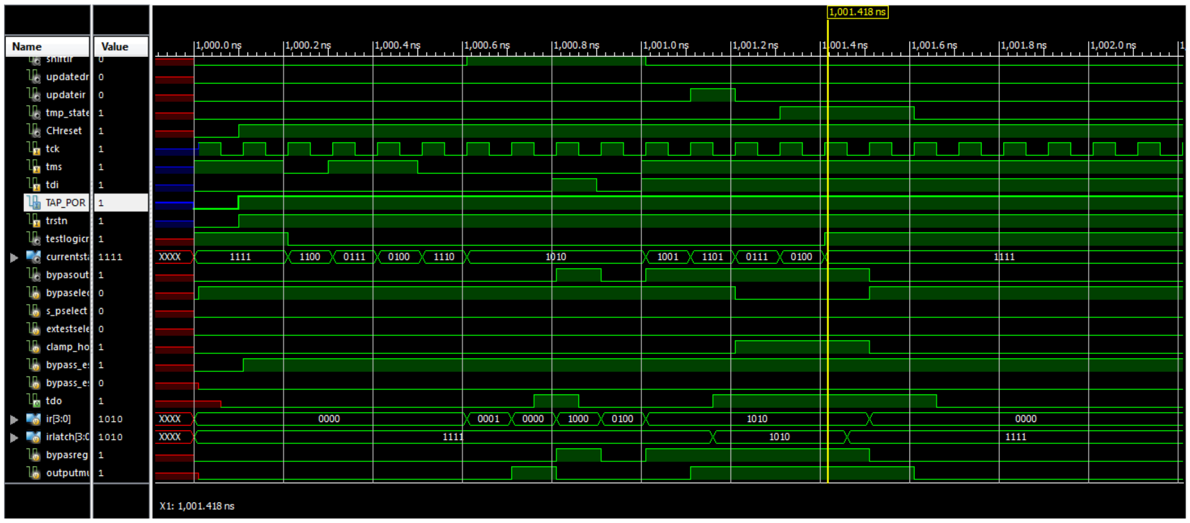


Fig 9. Test logic simulation waveform

5. IMPLEMENTATION RESULTS

Design is implemented in Xilinx XC6LX16-CS324 evaluation board and results are analyzed. The Xilinx nexys evaluation board is used here for physical design automation (floor planning, placement and rooting). The RTL design is also analyzed using Cadence Encounter™ RTL Compiler. Cadence RTL Compiler is a powerful tool for logic synthesis and analysis for digital designs. The FPGA utilization of TAPC is shown in table 3.

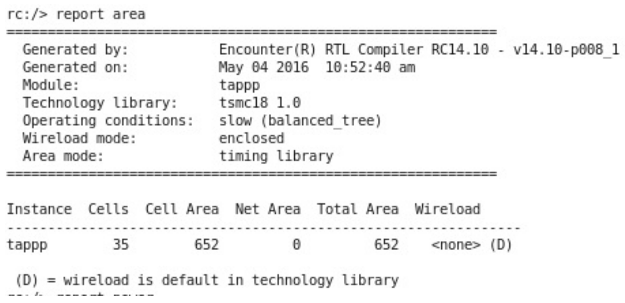
Table 3. Resource utilization of TAP controller

Logic Utilization	Used	available	utilization
Number of slice flip flops	4	1536	1%
Number of 4 input LUTs	13	11536	1%
Number of occupied slices	7	768	1%
Number of bonded IOBs	18	63	28%
IOB latches	7		
Number of BUFGMUXs	1	8	12%
Average fan-out of Non-clock nets	7.00		

Table 4. Resource utilization of IP core

Logic Utilization	Used	available	utilization
Number of slice flip flops	17	8224	1%
Number of 4 input LUTs	19	9112	1%
Number of occupied slices	10	2278	1%
Number of bonded IOBs	13	232	5%
Number of BUFGMUXs	1	16	6%

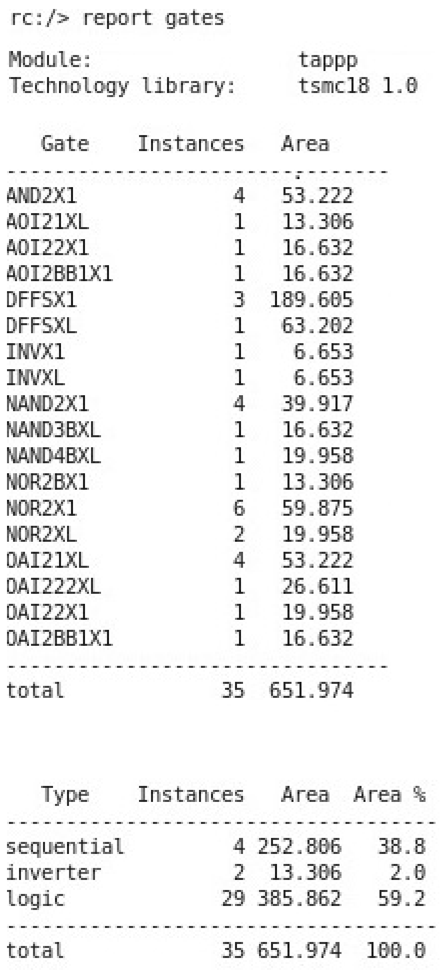
The area in terms of the cells used is shown in figure 10.



```
rc:/> report area
============================================================
  Generated by:          Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:          May 04 2016  10:52:40 am
  Module:                tappp
  Technology library:    tsmc18 1.0
  Operating conditions:  slow (balanced_tree)
  Wireload mode:         enclosed
  Area mode:             timing library
============================================================

Instance  Cells  Cell Area  Net Area  Total Area  Wireload
---------------------------------------------------------------
tappp       35       652         0         652    <none> (D)

 (D) = wireload is default in technology library
```

Fig 10 Area consumption of TAPC

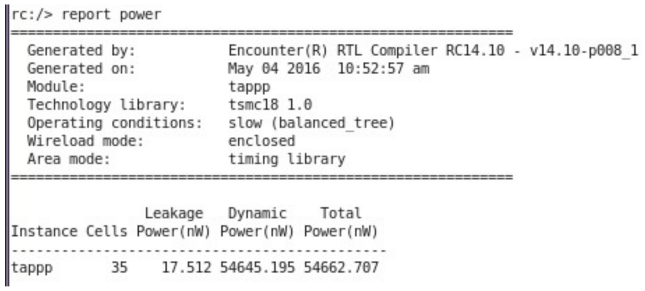
Alternatively the area may also be represented using number of gate equivalent (GE). The GE is an estimation of hardware design complexity independent from circuit realisation and fabrication technology. One approach to calculate the GE is by dividing design area over area of one GE. Using the values from fig.5 the approximate GE of TAPC is obtained to be 66



```
rc:/> report gates

Module:                tappp
Technology library:    tsmc18 1.0

   Gate    Instances   Area
--------------------------------
AND2X1             4   53.222
AOI21XL            1   13.306
AOI22X1            1   16.632
AOI2BB1X1          1   16.632
DFFSX1             3  189.605
DFFSXL             1   63.202
INVX1              1    6.653
INVXL              1    6.653
NAND2X1            4   39.917
NAND3BXL           1   16.632
NAND4BXL           1   19.958
NOR2BX1            1   13.306
NOR2X1             6   59.875
NOR2XL             2   19.958
OAI21XL            4   53.222
OAI222XL           1   26.611
OAI22X1            1   19.958
OAI2BB1X1          1   16.632
--------------------------------
total             35  651.974


   Type    Instances   Area  Area %
------------------------------------
sequential         4 252.806   38.8
inverter           2  13.306    2.0
logic             29 385.862   59.2
------------------------------------
total             35 651.974  100.0
```

Fig 11. Gate count of TAP

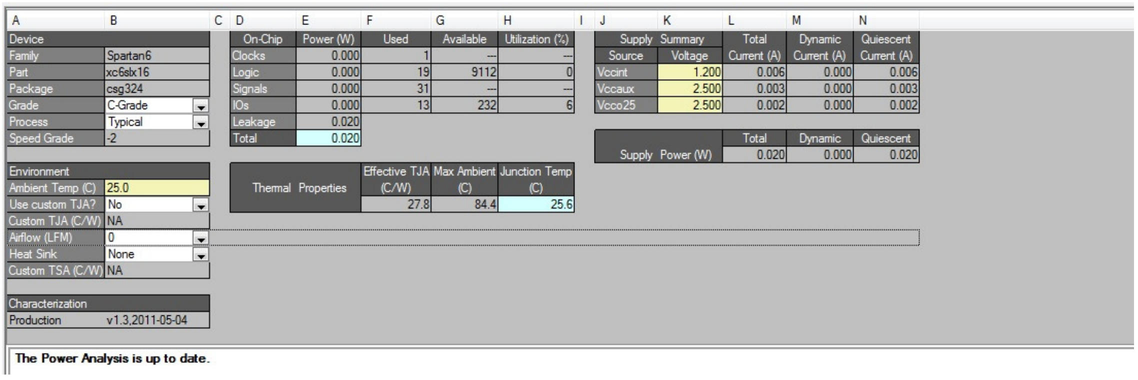
Power usage of TAPC in terms of leakage and dynamic power dissipation is shown in figure 12. The leakage power gives the static power dissipation during quiescent condition and dynamic power is the power usage during its normal operation.



```
rc:/> report power
============================================================
  Generated by:           Encounter(R) RTL Compiler RC14.10 - v14.10-p008_1
  Generated on:           May 04 2016  10:52:57 am
  Module:                 tappp
  Technology library:     tsmc18 1.0
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

                 Leakage   Dynamic    Total
Instance Cells Power(nW) Power(nW) Power(nW)
--------------------------------------------
tappp       35    17.512 54645.195 54662.707
```

Fig 12. Power usage of TAPC

Power usage of IP core using XPower analysis tool in terms of leakage and dynamic power dissipation is shown in figure 13. The leakage power gives the static power dissipation during quiescent condition and dynamic power is the power usage during its normal operation.



Device	
Family	Spartan6
Part	xc6slx16
Package	csg324
Grade	C-Grade
Process	Typical
Speed Grade	-2

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0
Heat Sink	None
Custom TSA (C/W)	NA

Characterization	
Production	v1.3,2011-05-04

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	1	---	---
Logic	0.000	19	9112	0
Signals	0.000	31	---	---
IOs	0.000	13	232	6
Leakage	0.020			
Total	0.020			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	27.8	84.4	25.6

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.006	0.000	0.006
Vccaux	2.500	0.003	0.000	0.003
Vcco25	2.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.020	0.000	0.020

The Power Analysis is up to date.

Fig. 13. Power usage of IP core

6. CONCLUSIONS

The thesis proposes design and implementation of the IEEE 1149.1-2013 TAP controller IP core. The design is synthesised using Xilinx® ISE and implemented in Xilinx nexys 3 evaluation board. The proposed work is fully compatible with the standard and provides a reusable module with a robust and easily testable design. This work is expected to serve as a ready to use module that can be directly inserted in to a new digital IC designs with little modifications.

REFERENCES

[1] IEEE Standard 1149.1-2013, ``Standard Test Access Port and Boundary-Scan Architecture''

[2] IEEE Standard 1149.1-2001, ``Standard Test Access Port and Boundary-Scan Architecture''.

[3] Kenneth P. Parker, "Surviving State Disruptions Caused by Test: the "Lobotomy Problem"," IEEE International Test Conference 2010.

[4] Kenneth P. Parker, David Dubberke, Shuichi Kameyama, “Surviving State Disruptions Caused by Test: A Case Study,” Keysight Technologies, August, 2014.

[5] David B. Lavo, “A Good Excuse for Reuse: “Open” TAP Controller Design,” ITC International Test Conference, 2000, p. 1090-1099.

[6] Dave Stang, and R. Dandapani, “An implementation of IEEE 1149.1-To avoid violation and other practical In Compliance”, IEEE-2002.

[7] Lima, M., F. Santos, J. Bione, T. Lins, and E. Barros, “ ipPROCESS: A Development Process for Soft IP-core with Prototyping in FPGA “, Forum on Design Languages (FDL 2005), Swiss, Sept. 2005