



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

# IJRSET

International Journal of Innovative Research in  
**SCIENCE | ENGINEERING | TECHNOLOGY**

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 4, April 2023

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

[ijirset@gmail.com](mailto:ijirset@gmail.com)

[www.ijirset.com](http://www.ijirset.com)

# Implementation of FSM for Read and Write Operation for OCP Master and Slave

Dr. K, Madhava Rao, Endirala Raja Shekar, Golla Sanjay Kumar, Gollanapally Sri Kasyap

Department of ECE., B V Raju Institute of Technology, Narsapur, India

Department of ECE, B V Raju Institute of Technology, Narsapur, India

Department of ECE, B V Raju Institute of Technology, Narsapur, India

Department of ECE, B V Raju Institute of Technology, Narsapur, India

**ABSTRACT-** The primary goal is to facilitate effective and error-free communication between IP cores that utilize identical protocols in the System on Chip framework. SOC is essentially a collection of interconnected components that rely on the uninterrupted transmission of data to accomplish their designated tasks. As a result, the presence of multiple interfaces is critical for the smooth transfer of data, and any malfunction of these interfaces could result in the failure of the entire system. Depending on the application and designer's choice, protocols may be employed as interfaces in the SOC system, with each interface possessing distinct characteristics that are most appropriate for the particular application.

**KEYWORDS** - MASTER, SLAVE, Internal memory design, FSM's, IPcores.

## I. INTRODUCTION

The protocol known as the Open Core Protocol is an advanced and core-centric communication system. that facilitates the establishment of a high-performance interface between IP cores. This protocol endeavors to mitigate the amount of time, risk, and cost associated with the design and manufacturing processes of System on Chip (SOC) designs.

A salient attribute of OCP is its configurability, which is a direct function of the specific application requirements. The protocol was selected over alternative options due to its comprehensive suite of supporting features. These features distinguish this protocol from other core protocols and provide a significant advantage.

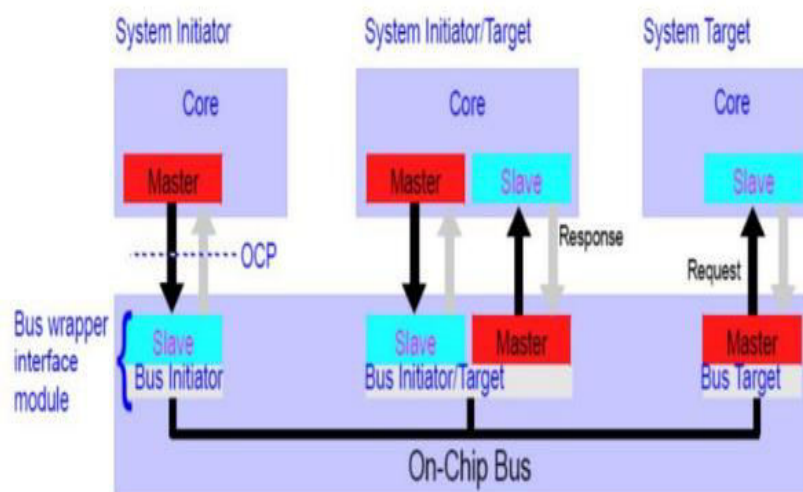


Figure 1.1 : Block Diagram of OCP

OCP stands out from traditional bus and component interfaces that focus only on the data flow aspects of inter-core



communications. It takes a more comprehensive approach, covering all aspects of core communication, such as sideband control and test harness signals. By adopting a synchronous unidirectional signaling structure, this protocol simplifies core implementation, integration, and timing analysis. Furthermore, it has adaptable nature allows for seamless incorporation of new core capabilities, thus minimizing modifications to the test suite during core upgrades.

## II. OCP DATAFLOW SIGNALS

The Open Core Protocol interface uses a variety of signal groups, including basic signals, burst extensions, tag extensions, and thread extensions, to facilitate communication between two cores. While the basic signals are mandatory for all OCP configurations, the optional signals can be tailored to meet the specific communication requirements of a core. The basic signals, such as Request, Response, and Data Handshake signals, are essential for communication between a master and a slave core. The Request phase employs signals to request a specific operation from the slave, which is terminated by the 'SCmdAccept' signal. The Response phase uses signals to provide an appropriate response to the corresponding request, which is concluded by the 'SResp' signal. The transfer of data between the master and slave is facilitated by the Data Handshake signals.

The fundamental signals facilitate uncomplicated read and write operations, as well as pipelining, of the master and slave. The burst extensions support burst operations, which involve multiple requests and data transfers. These extensions enable the grouping of multiple transfers with a defined address relationship and are only functional if the interface includes the MBurstLength signal.

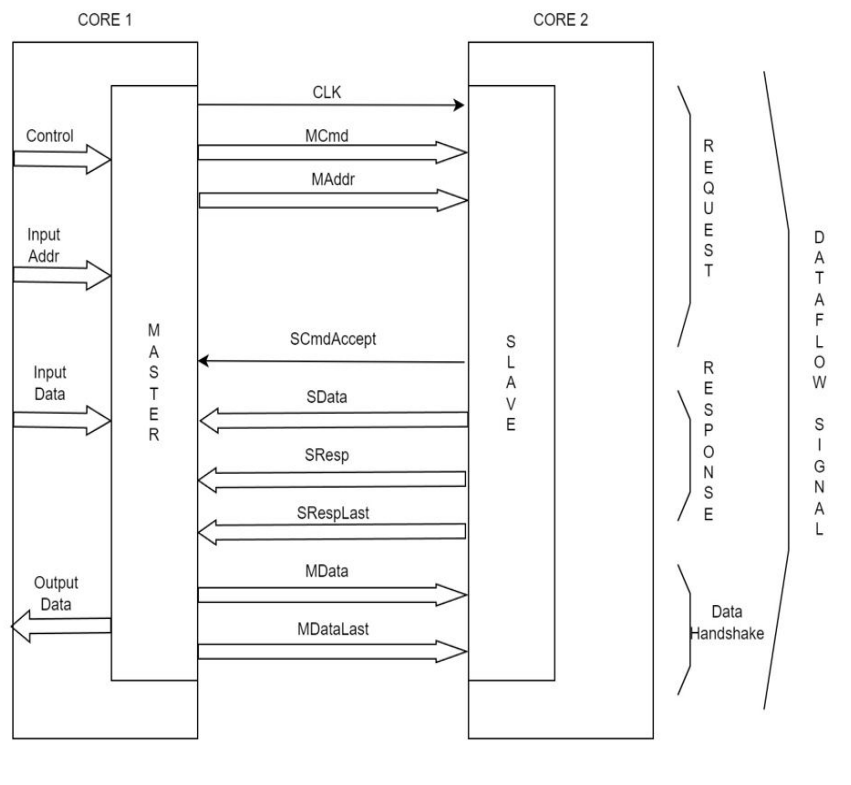


Figure 2.1 : OCP Dataflow Signals

The burst length, defined by the MBurstLength signal, represents the number of read or write operations to be executed in a single burst. The system provides the burst length to the master, which in turn transfers it to the slave. The burst length is only a necessary input for the master in burst mode and is not required in simple read and write operations.

### III. THEORY OF OPERATION

For the working of Open Core Protocol, the following are the operations involved

- i. Point to point interface:  
In OCP, all the signals are unidirectional. No signal is bidirectional. The main reason to specify all the signals unidirectional is to simplify the timing analysis. This will also make the physical design simple. All the signals are OCP are synchronous (other than reset). Hence, this enables all the signals to be point to point with exception of the clock signal.
- ii. Bus Independence:  
Another main advantage of using the open core protocol over the others is that – OCP is bus independent. This means that, OCP can be interfaced into any existing protocol. Usually, other protocols that are bus dependent use address decoders or generate select signals. In OCP, masters are used to pass commands.
- iii. Response:  
In OCP, the flow of data between master and slave occurs in two different phases which is Request and corresponding Response. Request – Response are considered as the phases of a cycle. During the request phase, the master's request is accepted by the slave. The slave will perform the desired operation as required by the master and will send the result in the response phase. The slave will accept the request from the master in one cycle, then it will respond to the master in the next clock cycle. The speciality of OCP protocol is that it provides two options – one is to have the response in a different cycle after receiving the request. Other option is that, it will immediately complete the request in the same clock cycle without sending it in another response.

### IV. OCP SPECIFICATION

For supporting the read and write operation, the OCP defines a set of dataflow signals. Each dataflow signal has its own role for the successful transmission of data. The width of the signals vary and the driver (master or slave) also varies for the signals. The following are the signals that play a major role while performing read and write operations.

**Clk** – Clk stands for clock. Clock is used to generate the input signal. Every rising edge of the clock can be corresponded to the rising edge of the Clk. Similarly, every falling edge of the OCP clock can be corresponded to the falling edge of the Clk.

**MAddr** – MAddr stands for Master address. This signal is used to specify the address that is going to get affected because of the of the current transaction. This address is a dependent on the slave. This means that the master can send the address where the data has to written or can specify the address from where the data has to be fetched (read).

**MCmd** – MCmd stands for Master Command. The master uses this signal to specify the type of operation that needs to be performed. This signal can be idle or non-idle. When non-idle, the master uses this signal to specify whether read operation is being performed or write operation.

**MData** – Mdata stands for Master Data. The purpose of this signal is to transfer data from the master to the slave. The data that is being transferred in this signal is the write data. We use the mdata parameter to transmit the data.

**SCmdAccept** – It stands for Slave Command Accept. As the name suggests, this signal is used for acknowledgement of the slave to the master that the slave has received the request and is accepting the request. It is a single bit signal. When the value of SCmdAccept is high, i.e., 1, it signals that the slave has accepted the request. Whereas, when the value is low, it means that the slave did not accept the request.

**SData** – Sdata stands for Slave Data. This is the opposite of MData. Mdata is used to carry the data which is to write. Whereas, Sdata is used to carry the read data. If the MCmd signal corresponds to a read request, then the slave will transfer the data from specific address at MAddr to the master as a part of read operation. This field can be composed into the Open Core Protocol by using sdata as a parameter.

**SResp** – SResp stands for Slave Response. It is given as a part of the transfer request by the slave to its master.





V. IMPLEMENTATION

I. FSM FOR OCP MASTER

A Finite State Machine has been designed to handle the simple write and read operations of the OCP Master. The FSM ensures that control returns to the IDLE state after each operation. The development of the FSM for OCP Master - Simple Write and Read is shown in the figure, which consists of four states: IDLE, WRITE, READ, and WAIT.

The OCP operation is carried out in two phases: the Request Phase and the Response Phase. At the start, the control is in IDLE mode where control is "000", and all outputs are set to "don't care."

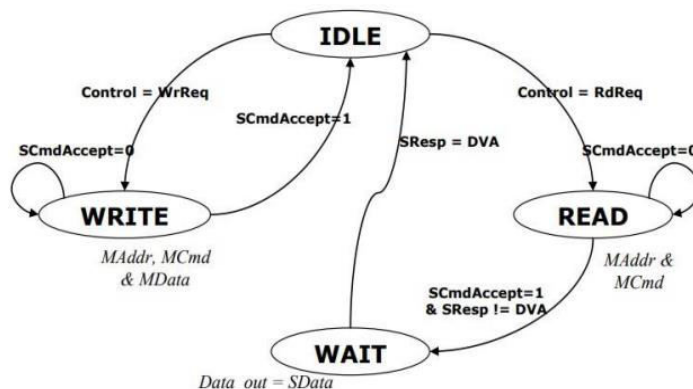


Figure 5.1 : FSM FOR OCP MASTER

The initial state of the FSM is the IDLE state, where the address and data are sent to the slave to be written, and the SCmdAccept signal is checked to determine if the write operation has been completed. If it is not asserted to high, it indicates that the writing operation is still ongoing, and as a result, the control remains in the WRITE state. Once the write operation is finished, the control returns to the IDLE state and waits for the next phase. When the Master receives a read request, it switches to the READ state and forwards the address to the slave. The slave then responds with the SCmdAccept signal, ending the request phase. If SResp is not set to Data Valid (DVA), the control enters the WAIT state and waits for the SResp signal. After the read operation is finished, the SResp signal is set to DVA, and the data stored at the designated memory address is retrieved. The SResp signal terminates the response phase, and the control returns to the IDLE state to wait for the next request.

II. FSM FOR OCP SLAVE

The slave's state and output are determined based on the MCmd issued by the master. If the command is a write request, the control begins in the IDLE state, and the data is written to the memory address specified by the master.

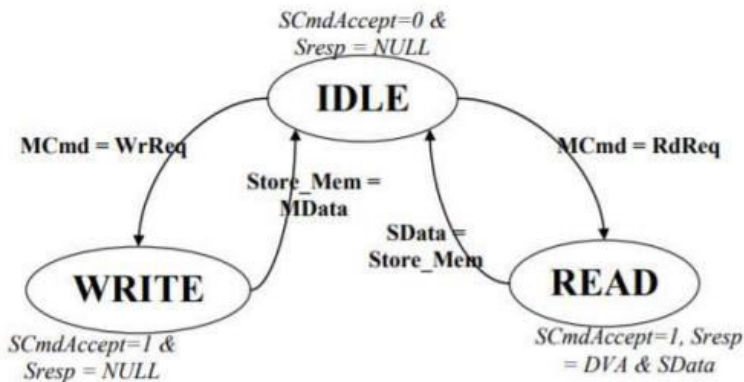


Figure 5.2 : FSM FOR OCP SLAVE



After the completion of the write operation, the SCmdAccept signal is set to high and transmitted to the master. In the case of a read request using MCmd, the control shifts to the READ state, and data is read from the memory address specified by the master. Upon completion of the read operation, the SCmdAccept value is set to high, and the SResp value is set to DVA, indicating that the operation is finished, and control has returned to the IDLE state.

## VI. RESULTS

### I. EXPECTED WAVEFORMS

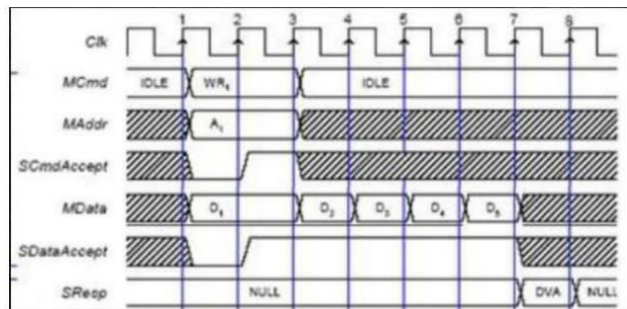


Figure 6.1 Expected waveform

The above figure displays the expected waveforms that are to be obtained for the successful read and write operations of the OCP master.

### II. OBTAINED WAVEFORM

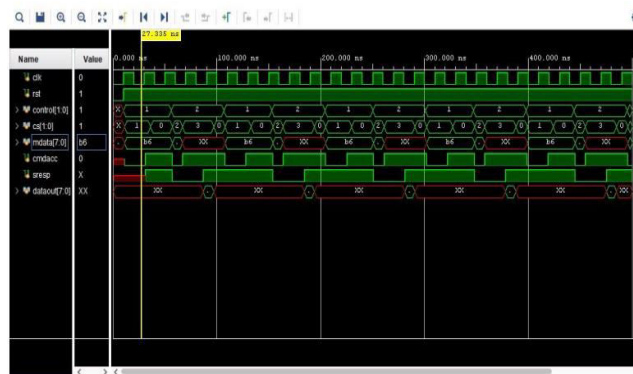


Figure 6.2 Obtained Waveform

Once the Verilog code is executed without any syntactical or logical errors, the read and write operations get performed between the master and slave. The command initiated by the system will be the basis on which the master will perform a read or a write operation. After the command has been successfully initiated, the read or write operation has been performed successfully. The Fig.7 shows the waveforms that are obtained from the Verilog code that has been previously executed. As we can see, the obtained waveforms are similar to the expected waveforms. This means that the program was written correctly the read and write operations have been successfully performed.

## VII. FUTURE SCOPE

One can expand the design by building a complete system around it. For instance, the protocol can serve as an interface between an ARM processor and any OCP compliant device (like SRAM). Additionally, this protocol can be implemented in Burst mode, which facilitates the execution of multiple read or write operations in a single go. This mode can be further extended to accommodate various types of supported bursts, and the protocol can also be



augmented with Thread and Tag extensions supported by corresponding signals. The current project work presents an optimal foundation for enhancing or further advancing the Open core protocol.

## VIII. CONCLUSION

OCP's interconnected systems and cores allow for easy integration of modules, allowing system integrators to choose the best cores and application interconnect systems. This reduces the design time as designers of cores and systems can work simultaneously. Furthermore, the ability to reuse cores without the need for core recreation saves time. These capabilities are useful for real-time applications. Our project involves modeling the OCP master and slave, as well as the internal memory design using VHDL. Our simulation results demonstrate successful OCP communication between multiple IP cores without any loss of data or control information. The OCP protocol supports simple read and write operations, as well as burst extension, which appears to automate address generation based on the simulation results. The OCP design components are effectively verified in terms of their behavior during many scenarios in the simulation.

## REFERENCES

- [1] Weber, Wolf-Dietrich. "Enabling reuse via an IP core-centric communications protocol: Open core protocol (tm)." Proceedings of the IP (2000): 2019.
- [2] Pamarthy, Gopika Rani Alekhya, M. Durga Prakash, and AvinashYadlapati. "Extensions of open core protocol and their high level verification using system verilog and UVM." 2017. [3] International Conference on Inventive Communication and Computational Technologies (ICICCT). IEEE, 2017. SRIKANTH, N., and K. SURESH KUMAR. "Design of FSM Oriented Open Core Protocol for Multi Stage Interconnection Network." (2017).
- [4] Elina Rajan Varughese and Rony Antony, "Implementation of extended open core protocol interface memory system using Verilog HDL", P. 978-1-4673-6126/2/13/\$31.00 c 2013, IEEE.
- [5] P. Wielage, J. Marinissen, M. Altheimer, and C. Wouters, "Design and DFT of a high-speed area-efficient embedded asynchronous FIFO," in Proc. Design, Automation and Test in Europe (DATE), 2007, pp. 853–858.
- [6] V. S. Vij, R. P. Gudla, and K. S. Stevens, "Interfacing synchronous and asynchronous domains for open core protocol," in 27th International Conference on VLSI Design and 13th International Conference on Embedded Systems, 2014, pp. 282–287.
- [7] P. H. Wu, C. G. Lin, C. S. Wen, and S. F. Hsiao, "Asynchronous AHB bus interface designs in a multiple-clock-domain graphics system," IEEE Asia-Pacific Conference on Circuits and Systems, Proceedings, APCCAS, pp. 408–411, 2012.
- [8] A. H. M. Soliman, E. M. Saad, M. El-Bably, and H. M. A. M. Keshk, "Designing a WISHBONE Protocol Network Adapter for an Asynchronous Network-on-Chip," International Journal of Computer Science Issues, vol. 8, no. 2, pp. 261–267, 2012.
- [9] E. Kasapaki and J. Sparsø, "Argo: A Time-Elastic Time-DivisionMultiplexed NOC using Asynchronous Routers," in Proc. IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC). IEEE Computer Society Press, 2014, pp. 45–52.
- [10] Accellera Systems Initiative, "Open Core Protocol specification, release 3.0," 2013. [Online]. Available: [http://www.accellera.org/downloads/standards/ocp/ocp\\_3](http://www.accellera.org/downloads/standards/ocp/ocp_3)



**INNO SPACE**  
SJIF Scientific Journal Impact Factor  
**Impact Factor: 8.423**



**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details