JURNAL INFOTEL
Informatics - Telecommunication - Electronics
Website Jurnal : http://ejournal.st3telkom.ac.id/index.php/infotel
ISSN : 2085-3688; e-ISSN : 2460-0997

Design of AXI4-Stream based Modulator IP Core for Visible Light Communication System-on-Chip

Erwin Setiawan[1]*, Trio Adiono[2]
[1,2]University Center of Excellence on Microelectronics, Institut Teknologi Bandung
[1,2]Jl. Tamansari No. 126, Bandung, 40132, West Java, Indonesia
*Corresponding email: erwin.ouyang@gmail.com

Received 22 May 2018, Revised 26 June 2018, Accepted 05 July 2018

Abstract — In this paper, the design of AXI4-Stream based modulator IP core for Visible Light Communication is reported. The modulator IP core conforms to the AXI4-Stream protocol standard, which is widely used in System-on-Chip (SoC) design. There are three modulation types in this IP core namely, Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), and Quadrature Amplitude Modulation-16 (QAM-16). These modulation types are commonly used in the DCO-OFDM system. The modulation types can be selected programmatically from software that runs in the main processor by accessing the control register. The output of the modulator is designed for DCO-OFDM modulation using 64-point IFFT. According to the simulation results, this modulator IP core can achieve a throughput of 95.36 Mb/s, 184.77 Mb/s, and 347.81 Mb/s for BPSK, QPSK, and QAM-16, respectively. This modulator IP core is reusable in the DCO-OFDM system, so it increases productivity in DCO-OFDM system design.

Keywords – AXI4-Stream, Modulator, IP Core, Visible Light Communication, System-on-Chip

I. INTRODUCTION

Visible Light Communication (VLC) is a type of wireless communication that uses visible light as its medium. The transmitter uses an LED for transmitting data and the receiver uses a photodiode for receiving data. In VLC technology, the functions of lightning and communication are combined into one device, i.e the LED, so it offers the potential for electricity and cost savings [1][2].

VLC can use either digital modulated-signal such as PWM [3-4] or analog signal like audio [5-6] and analog modulated-signal as well such as DCO-OFDM, as in [7]. To use DCO-OFDM modulation, the data must be modulated either with BPSK, QPSK, or QAM. The implementation of DCO-OFDM can be done in a microcontroller, but the throughput is limited [7]. The QPSK modulation can be implemented in a programmable SoC, as in [8]. The implementation of a VLC system in a programmable SoC can be done rapidly [9] because we don't have to design all of the components in the system from scratch. In SoC methodology, we can use reusable IP cores that have been designed and verified by the third party, so this can increase productivity [10-11].

The IP cores for SoC are designed with a standard interface, so it can easily be integrated with other IP cores. Advanced Extensible Interface (AXI) is a bus that commonly used in SoC design [12]. AXI4-Stream is a stream-based version of AXI version 4. AXI4-Stream is suitable for Digital Signal Processing (DSP) applications because the data must be transferred at high speed between IP cores, continuously, as in [13]

Some research has designed SoC for VLC. In reference [14], a VLC receiver SoC is implemented in a 0.18 µm CMOS process. The receiver consumes 2.2 mW of power at 24 Mb/s from a 1.8 V power supply. The modulation type is digital that conforms to IEEE 802.15.7 PHY-II standard. In reference [15], a VLC transceiver SoC is implemented in a 0.18 µm BCDMOS technology. The modulation type is digital using Manchester encoding. The signal waveform is 12 MHz.

In this research, an AXI4-Stream based modulator IP core is designed. This paper consists of four parts.

The first part is an introduction. The second part is the research method. The third part is the result and discussion. The last part is the conclusion. Our work focuses on the development of an IP core for a modulator that can be used in the DCO-OFDM system. The difference between this paper and reference [7] is in the implementation by using SoC instead of a microcontroller. The difference between this paper and reference [8] is the modulator is packed into IP core with AXI-Stream interface instead of just RTL, so it can be reused for future projects. The difference between this paper and references [9, 14, 15] is the modulation type, which is BPSK, QPSK, and QAM-16 instead of digital modulation.

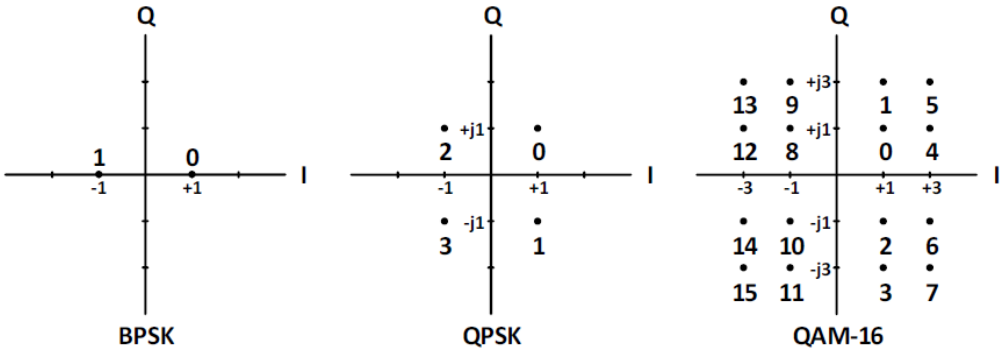

Fig. 1. BPSK, QPSK, and QAM-16 Constellation Diagram [16]

II. RESEARCH METHOD

A. BPSK, QPSK, and 16-QAM Modulation

Modulation is needed for grouping the bitstream data into a symbol. There are three modulation types that are used in this research namely, Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), and Quadrature Amplitude Modulation-16 (16-QAM). Each symbol in BPSK, QPSK, and QAM-16 modulation consists of 1-bit, 2-bit, and 4-bit of data. The data bits are mapped into a complex number in a constellation diagram. The constellation diagram for this research is shown in Fig. 1.

B. Block Diagram

The bitstream input for this modulator can come from data register that can be accessed by the main processor or from another IP core such as Viterbi or Reed-Solomon IP core [16]. In this research, the bitstream input comes from data registers. The output symbol of the modulator is a complex number that will be sent to the Inverse Fourier Transform (IFFT). The IFFT IP core is Xilinx IP core that uses AXI4-Stream interface. The block diagram for this research is shown in Fig. 2.

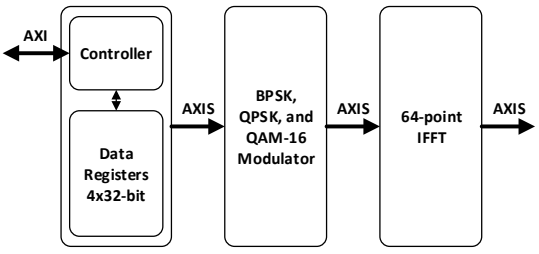

Fig. 2. Block Diagram

The controller uses AXI4-Memory Mapped (AXI) based interface because the main processor works by accessing memory or I/O address. The controller is an AXI slave. The AXI master is the main processor. The main processor can write and read data to and from the data registers. The size of data registers is 4x32-bit. After the bitstream data are written to the data registers, then the controller will send this bitstream to the modulator through the AXI4-Stream interface (AXIS). The output of the modulator is connected to the IFFT IP core by using AXI4-Stream interface.

C. Data Format

The IFFT length is 64-point, so we have 64 subcarriers in the frequency domain. The subcarriers are complex numbers that will be filled with the symbols from BPSK, OPSK, or QAM-16 modulator. DCO-OFDM differs from OFDM that commonly used in RF technology. In DCO-OFDM, the output of the IFFT is real and positive numbers. This happens because in VLC there is no upconversion process that combines the real and imaginary part [7]. In DCO-OFDM, the real and positive output can be realized by imposing Hermitian symmetry in the frequency domain [17]. The output of the modulator will be filled to the IFFT subcarriers as in Fig. 3.

Subcarriers X_0 and X_{32} are filled with 0 and subcarriers X_{33} to X_{63} are complex conjugate of subcarriers X_1 to X_{31} in order to achieve the Hermitian symmetry. The subcarriers X_1 to X_{31} is filled with the BPSK, QPSK, or QAM-16 symbols. For BPSK, QPSK, and QAM-16 modulations, every IFFT process can carry up to 31-bit, 62-bits, and 124-bits. The modulation type can be configured from software by accessing the control register in the controller. When the modulation is BPSK, then whenever there is a data written to the first data register, that data will be sent to the modulator. For QPSK and QAM-16, the data will be sent to the modulator whenever there is a data

written to the second and fourth data register, respectively.

D. AXI4-Stream Protocol

AXI4-Stream protocol is a unidirectional channel between IP cores. One AXI4-Stream channel consists of one master port and slave port. The data always flow from master to slave.

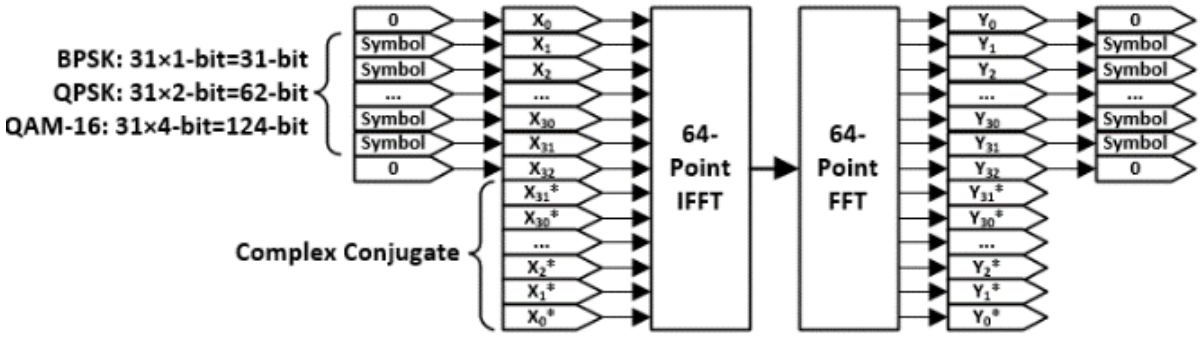

Fig. 3. Modulator Output Format for IFFT Input [16]

In modulator IP core there are two ports, one is for master and another port is for a slave. Slave port is the input port (form data register) because a slave can only receive data. Master port is the output port (to IFFT), because the master can only send data. The AXIS modulator IP core is shown in Fig. 4. The s_axis port is the slave port, and the m_axis port is the master port. The mod_type port is used for selecting the modulation type.

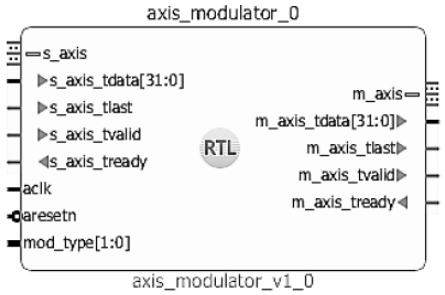

Fig. 4. AXIS Modulator IP Core [16]

AXI4-Stream channel for modulator IP core has four signals, namely TREADY, TDATA, TVALID, and TLAST. TREADY is used for indicating that the slave port is ready to receive data. TREADY is sent by the slave. TDATA is for the data signal, that consists of 32-bit signals. TVALID is used for indicating that the data on TDATA is valid. TVALID is sent by the master. The data is transferred when a handshake occurs. The handshake occurs when TREADY and TVALID have logic 1, i.e. the slave is ready to receive data and the data from the master is valid [18, 19]. TLAST is used for indicating the last data word in a data frame. One data frame can consist of several data word (32-bits).

E. IP Core Design

The internal block diagram of the AXIS modulator IP core is shown in Fig. 5. There are BPSK, QPSK, and QAM-16 modulator blocks that connected to the output (m_axis) through an AXI multiplexer. The selector of the multiplexer is used for selecting the modulation type.

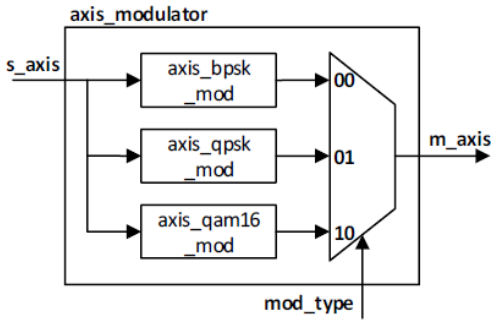

Fig. 5. Internal Block Diagram of Modulator IP [16]

The internal block diagram of BPSK, QPSK, and the QAM-16 modulator is shown in Fig. 6. The figure is a generalized block diagram for BPSK, QPSK, and QAM-16 modulator. There is a memory called data_mem that store the BPSK, QPSK, or QAM-16 symbols. The memory size 31-bits, 62-bits, and 124-bits for BPSK, QPSK, or QAM-16 modulation, respectively. The bpsk_mod, qpsk_mod, and qam16_mod are LUTs that map the bitstream data to a complex number in the constellation diagram. The subcar_mem is used for storing the modulation result before it is sent to the IFFT. The size of subcar_mem is 64x32-bit because the IFFT size is 64-point. The Hermitian subcarrier is filled whenever one BPSK, QPSK, or QAM-16 symbol is created. The axis_controller is a Finite State Machine (FSM) that control the reception of data from the data register and the transmission of data to the IFFT IP core. This controller generate TREADY signal on the slave port, TVALID signal on the master port. The TLAST signal is asserted to logic 1, when the data is the last data.

There are 1, 2, and 4 transactions for BPSK, QPSK, and QAM-16 modulation input, respectively. The

modulator output has 64 transactions of subcarrier data.

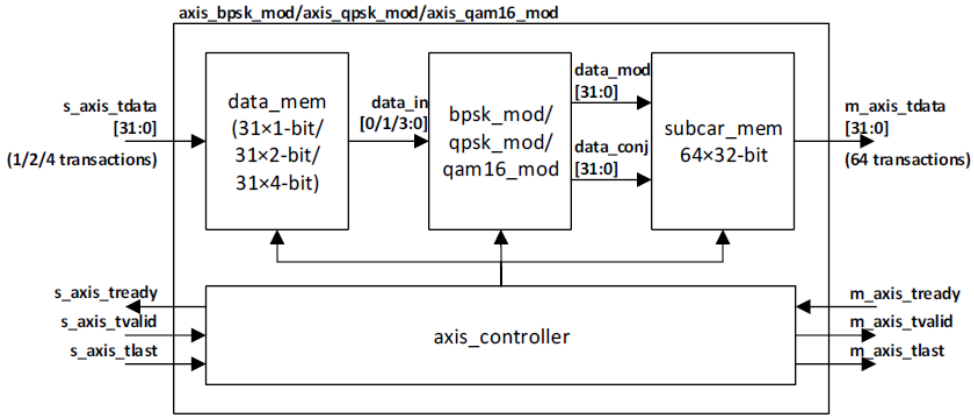

Fig. 6. Internal Block Diagram of BPSK, QPSK, and QAM-16 Modulator [16]

III. RESULT

A. *Input Timing Diagram*

The input timing diagram for BPSK, QPSK, and QAM-16 are shown in Fig. 7, Fig. 8, and Fig. 9, respectively. For BPSK modulation, there is one data word transaction (D0). The data bitstream size is 31-bit. For QPSK modulation, there are two data word transactions (D0 and D1). The data bitstream size is 62-bit. For QAM-16 modulation, there are four data word transactions (D0 to D3). The data bitstream size is 124-bit. The TREADY signal is always 1, so the slave (modulator) always ready to receive data. The TVALID signal is asserted to logic 1 when the bitstream data are being received. The TLAST signal is asserted to logic 1 in the last data word.

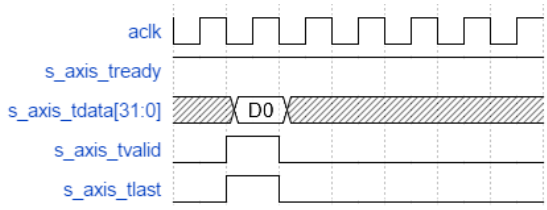

Fig. 7. BPSK Input Timing Diagram

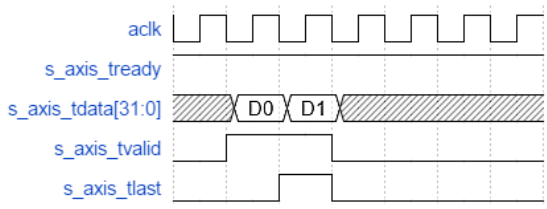

Fig. 8. QPSK Input Timing Diagram

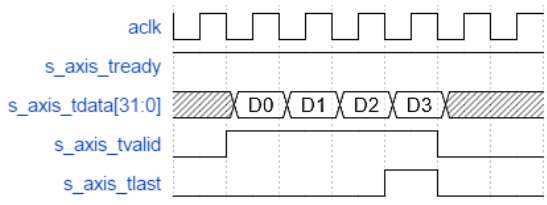

Fig. 9. QAM-16 Input Timing Diagram

B. *Output Timing Diagram*

The output timing diagram is shown in Fig. 10. There are 64 data transactions (D0 to D63). The TREADY signal is always 1, so the slave (IFFT) is always ready to receive data. The TVALID signal is asserted to logic 1 when the data are being transmitted. The TLAST signal is asserted to logic 1 in the last data word.

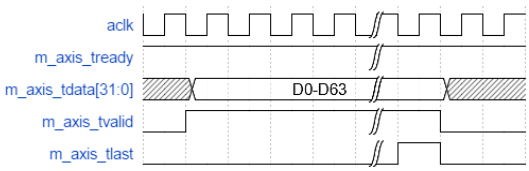

Fig. 10. Modulator Output Timing Diagram

C. *Latency*

Latency is the time between the first input and the first output. The clock speed is 100 MHz. The latency for BPSK, QPSK, and QAM-16 modulation is shown in Table 1. This latency is obtained by simulating the IP core in Vivado® simulator. The latency timing diagram for BPSK, QPSK, and QAM-16 are shown in Fig. 10., Fig. 11, and Fig. 12.

Table 1. Latency for BPSK, QPSK, and QAM-16

Modulation Type	Latency	Latency (ns)
BPSK	31 clock cycles	310 ns
QPSK	32 clock cycles	320 ns
QAM-16	34 clock cycles	340 ns

D. *Throughput*

The throughput is calculated from latency using equation (1). The total bits is 31-bits, 62-bits, and 124-bits for BPSK, QPSK, and QAM-16, respectively. The throughput for BPSK, QPSK, and QAM-16 modulation is shown in Table 2. The sample data timing diagram is shown in Fig. 13. This sample data are randomly generated data.

$$\text{throughput} = \frac{1}{\text{latency}} \times \text{total bits} \qquad (1)$$

Table 2. Throughput for BPSK, QPSK, and QAM-16

Modulation Type	Throughput
BPSK	95.36 Mb/s
QPSK	184.77 Mb/s
QAM-16	347.81 Mb/s

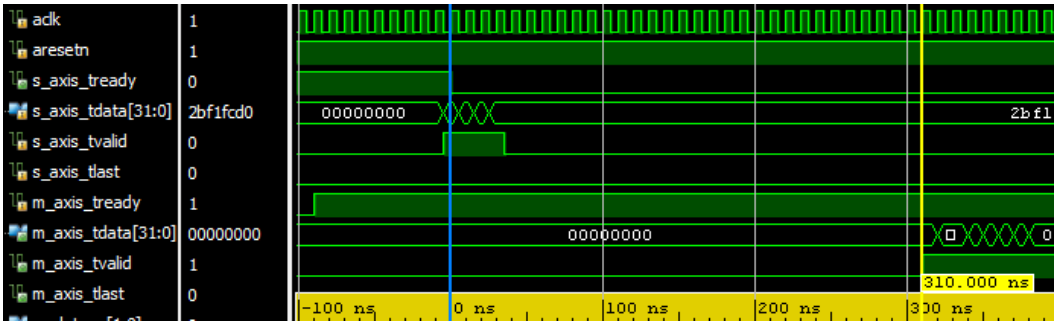

Fig. 10. BPSK Latency Timing Diagram

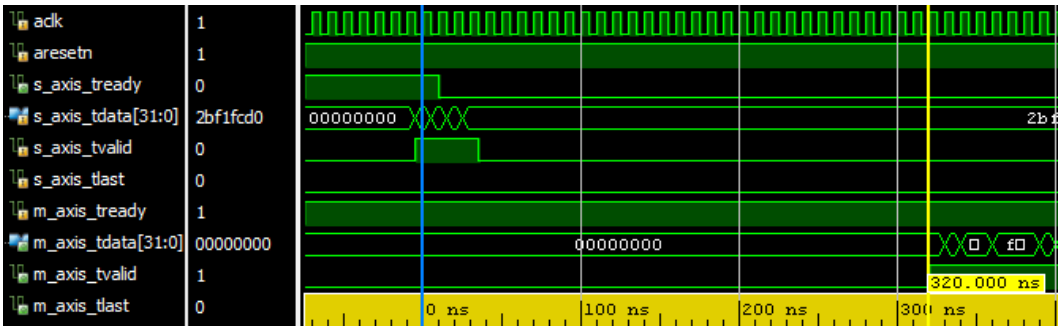

Fig. 11. QPSK Latency Timing Diagram

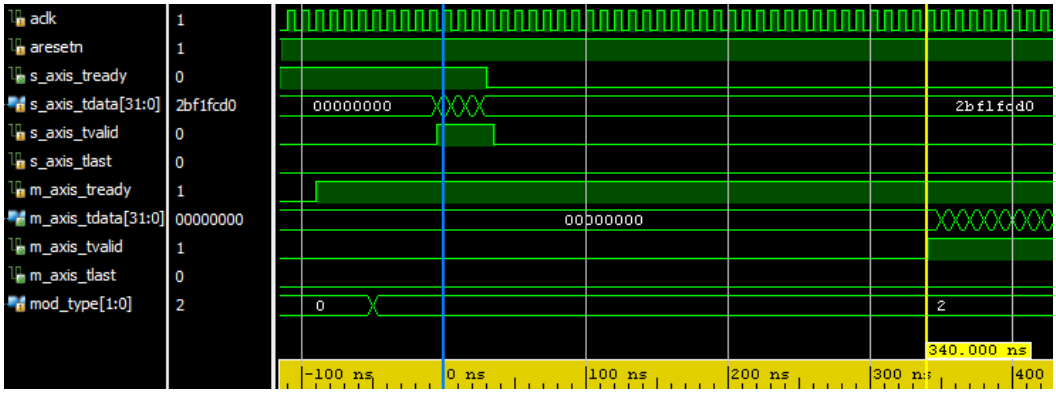

Fig. 12. QAM-16 Latency Timing Diagram

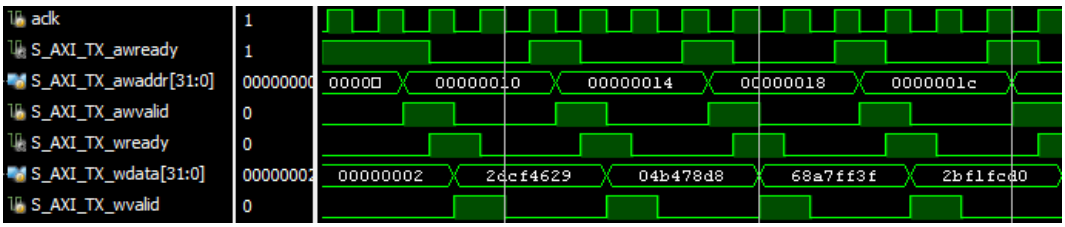

Fig. 13. Sample Data Timing Diagram

E. Functional Verification

Functional verification is done by plotting the modulator output data in MATLAB. The input data for the modulator are random generated data. The modulator constellation diagram for BPSK, QPSK, and QAM-16 are shown in Fig. 14, Fig. 15, and Fig. 16, respectively. Based on the constellation diagram, the modulator output can modulate bitstream data correctly.

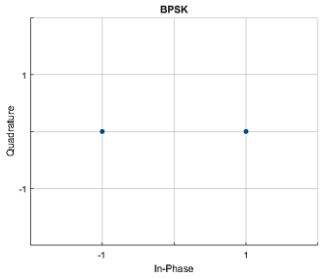

Fig. 14. BPSK Modulator Constellation Diagram

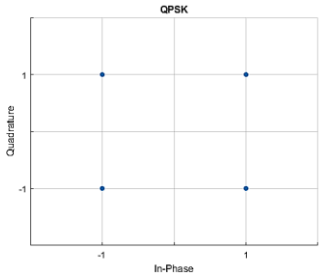

Fig. 15. QPSK Modulator Constellation Diagram

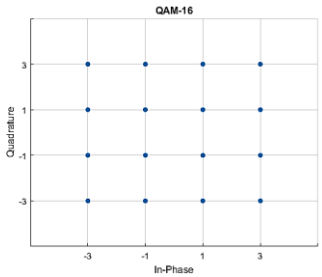

Fig. 16. QAM-16 Modulator Constellation Diagram

IV. DISCUSSION

Based on the result, the input and output timing diagram of the modulator IP conforms to AXI4-Stream standard. The modulator IP can send data to the Xilinx's IFFT IP without any error. The latency result shows that QAM-16 modulation has the longest latency, but it has the highest throughput. The functional verification shows that the data are modulated correctly without any error.

V. CONCLUSION

This paper presents the design of AXI4-Stream based modulator IP core for VLC SoC. According to the simulation results, the modulator IP core achieves a throughput of 95.36 Mb/s, 184.77 Mb/s, and 347.81 Mb/s for BPSK, QPSK, and QAM-16, respectively. This modulator IP core is used in VLC PHY layer [16].

ACKNOWLEDGMENT

We would like to grateful to Mr. Syifaul Fuada (Coordinator of VLC research in PME ITB) for his comments that greatly improved this manuscript.

REFERENCES

[1] S. Dimitrov and H. Haas, "Introduction," in *Principles of LED light communications towards networked LiFi*, 1st ed. Cambridge.

[2] S. Fuada, "Kajian Aspek *Security* pada Jaringan Informasi dan Komunikasi Berbasis *Visible Light Communication*," *J. INFOTEL*, Vol. 9(1), pp. 108-121, August 2017. DOI: https://doi.org/10.20895/infotel.v9i1.163.

[3] T. Adiono, S. Fuada, and A. Pradana, "Desain dan Realisasi Sistem Komunikasi Cahaya Tampak untuk *Streaming* Teks berbasis PWM," *J. Setrum*, Vol. 6(2), pp. 270-279, Desember 2017.

[4] T. Adiono, A. Pradana, and S. Fuada "Rancang Bangun Sistem Komunikasi Cahaya Tampak dengan Modulasi 2-PWM berbasis Mikrokontroller," *Unpublished.*

[5] S. Fuada and T. Adiono, "Short-range Audio Transfer through 3 Watt White LED based on LOS Channels," *Proc. of Int. Conf. on Intelligent Signal Processing and Communication Systems* (ISPACS), pp. 398-403, November 2017. DOI: 10.1109/ISPACS.2017.8266511.

[6] S. Fuada, and T. Adiono, "Rancang Bangun Layer Fisik Visible Light Communication Pada Sistem Transmisi Audio," *J. INFOTEL*, Vol. 9(3), pp. 352-360, August 2017. DOI: https://doi.org/10.20895/infotel.v9i3.288.

[7] T. Adiono, A. Pradana, R. V. W. Putra, W. A. Cahyadi, and Y. H. Chung, "Physical layer design with analog front end for bidirectional DCO-OFDM visible light communications," *Optik*. Vol. 138, pp. 103-118, 2016.

[8] E. Setiawan, M. M. Latin, V. A. Mardiana, and T. Adiono, "Implementation of baseband transmitter design based on QPSK modulation on Zynq-7000 all-programmable System-on-Chip," *Proc. of the Int. Symposium on Electronics and Smart Devices* (ISESD), 2017.

[9] T. Adiono, S. Fuada, and R. A. Saputro, "Rapid Development of System-on-Chip (SoC) for Network-Enabled Visible Light Communications," *Int. J. of Recent Contributions from Engineering, Science & IT. 6 (1)*, pp. 107-119, 2018.

[10] M. Keating and P. Bricaud, "Introduction," in *Reuse methodology manual for system-on-chips designs*, 3rd ed. Dordrecht.

[11] R. Dorothy and T. Sasilatha, "System on Chip Based RTC in Power Electronics," *Bulletin of Electrical Engineering and Informatics*, Vol. 6(4), pp. 358-363, Dec 2017.

[12] S. S. Math, R. B. Manjula, S. S. Manvi, and P. Kaunds, "Data Transactions on System-on-Chip Bus Using AXI4 Protocol," *Proc. of Int. Conf. on Recent Advancements in Electrical, Electronics and Control Engineering*, 2011.

[13] A. P. Putra, S. Fuada, Y. Aska, and T. Adiono, "System-on-Chip Architecture for High-Speed Data Acquisition in Visible Light Communication System," *Proc. of the IEEE Int. Symposium on Electronics and Smart Devices* (ISESD), pp. 63-67, October 2016. DOI: 10.1109/ISESD.2016.7886693.

[14] X. Li, B. Hussain, L. Wang, J. Jiang, and C. P. Yue," Design of a 2.2-mW 24-Mb/s CMOS VLC Receiver SoC With Ambient Light Rejection and Post-Equalization for Li-Fi Applications," *J. of Lightwave Technology*, Vol. 36, pp. 2366 – 2375, 2018.

[15] T. Lang, Z. Li, G. Chen, and A. Wang, A., "LED-Based Visible Light Communication and Positioning Technology and SoCs," [Online]. Available: http://ieee. tpu.ru/proceedings/papers/0ed299.pdf

[16] E. Setiawan, "System-on-Chip Architecture for Visible Light Communication," M.S. thesis, School of Electrical and Informatics Engineering, Institut Teknologi Bandung, Bandung, Indonesia, 2018.

[17] J. Dang, L. Wu, and Z. Zhang, "*OFDM Systems for Optical Communication with Intensity Modulation and Direct Detection*," [Online]. Available: https://www.intechopen.com/books/optical-fiber-and-wireless-communications/ofdm-systems-for-optical-communication-with-intensity-modulation-and-direct-detection, 2017.

[18] *AMBA® 4 AXI4-stream protocol specification*, version 1.0, Mar. 2010.

[19] *AMBA® AXI™ and ACE™ protocol specification*, issue d, Oct. 2011.