Hustle and frequency analysis based High speed and energy efficient art Design on spartan-6 fugal

Abhishek Kumar, Bishwajeet Pandey, D M Akbar Hussain, Mohammad Kamrul Hasan, Pervesh Kumar Shabeer Ahmad

Abstract- ***In this paper our aim is to design an energy efficient UART using different IO Standard. UART known as Universal Asynchronous Receiver Transmitter. It is one of the crucial element in communication system to communicate two michro controller based system. For short distance and low cost data exchange, UART is widely used. The implementation of UART's with VHDL can be unified into FPGA for the achievement of reliable, and compact data transmission. In this paper, we aregoing to implementan HSTL(High-Speed Transceiver Logic) IOSTANDARD based energy efficient Asynchronous Receiver Transmitter (UART). To achieve speed and high performance, we are using HSTL (High-Speed Transceiver Logic) IOSTANDARD.The HSTL family which we used in this paper are HSTL_I, HSTL_II, HSTL-I_18, and HSTL_II_18 IO Standards. Frequency Scaling technique is one of the best energy efficient technologies for FPGA, which is used in this paper In this paper, it has been analysed the demand for total power dissipation of different IO Standard at different frequency level. We have analysed that the increasing of frequency leads to increase in the clock and IO powerfor different IO standards.***

Keywords-FPGA, HSTL, UART, Energy efficient, IO Standards.

I. INTRODUCTION

In this paper, our main aim is to implement and design the UART (Universal Asynchronous Receiver Transmitter) on FPGA. It is just because as we know the UART (Universal Asynchronous Receiver Transmitter) is directly available in the case of Microcontroller. If we want serial data communication, then we can choose or use the UART. But when we are talking about the FPGA, serial data communication is not possible.So, we are implementing HSTL and Frequency Analysis BasedHigh Speed and Energy Efficient UART in this paper. We can used the UART for both slow and fast peripheral device i.e. between Computer and Printer, Controller and LCD and also many other place...That's why, UART is very popular and mostly used for little distance, low-speed serial communication at low cost. Figure 1 is showing some important block of UART. As we discussed above, we can used the UART for the slow and fast peripheral device i.e. Between Computer and Printer, Controller and LCD and also many other place... That's why, UART is very popular and mostly used for little distance, low-speed serial communication at low cost. In this paper, we have used Verilog Language to implement the core function of UART. The UART plays an essential role in Modern Complex Control Systems, Motion Control IC design, Reliable High-Speed Data Acquisition System, AES algorithm to communicate quickly.[4]To make the UART more power and energy efficient, various energy efficient techniques can be implemented for, e.g. Voltage Scaling, Thermal Scaling, Capacitance Scaling, change in IO scaling, clock, etc. We have tried to make our design more efficient in terms of energy and power. In this article, we used different HSTL I/O standards to measure the total power of the system. We have also used frequency changing techniques to measure the demand for power when we are changing the operating frequency of different HSTL I/O standards based UART. This will make the UART more efficient, and this design will be a significant contribution towards Green Communication [1].

Fig.1. Simple UART transmitter

II. UART

Transmitter, Receiver, and Baud Rate Generator are mainly three parts of the UART. We have discussed these parts in the next section. The transmitted and received data block diagram

Revised Manuscript Received on December 22, 2018.
Abhishek Kumar, SET, Sharda University,India
Bishwajeet Pande , Gyancity Research Lab, India
D M Akbar Hussain, Aalborg University, Denmark
Dr Mohammad Kamrul Hasan ,Universiti Malaysia Sarawak (UNiMAS)
Pervesh Kumar ,Sungkyunkwan University, South Korea
Shabeer Ahmad, Gran Sasso Science Institute, Italy

of the UART has shown in figure 1. As shown in Fig. 1, the UART has mainly three blocks which are- Clock, Transmitter, and Receiver. From the name of Transmitter, the purpose of this module is to convert the eight bit serial data into just single bit serial data. To enhance the reliability of UART, it can also be implemented with embedded Built-In-Self-Test (BIST) architecture incorporating FPGA technology. The main aim of this research paper is to achieve Compacted, Steady, and reliable data Transmission, so to achieve these we implement the UART with VHDL, which can be unified into FPGA for the achievement of reliable and compact data transmission. To make the UART more power and energy efficient, various energy efficient techniques can be implemented for, e.g. Voltage Scaling, Thermal Scaling, Capacitance Scaling, change in IO scaling, clock, etc. [2]. The HSTL family which we used in this paper are consists of HSTL_I, HSTL_II, HSTL-I_18, and HSTL_II_18 IO Standards. Frequency Scaling techniques are one of the best energy efficient technologies for FPGA, which is used in this paper[3].

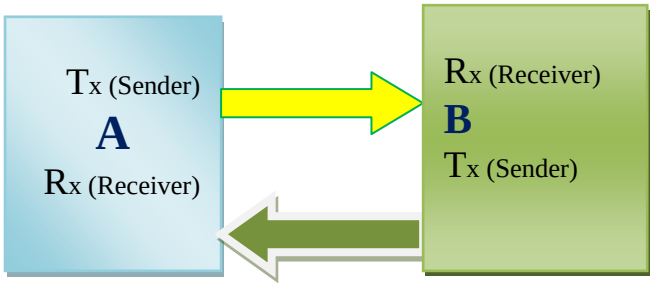


Fig.2. Structure of Communication

A.TRANSMITTER

From the name the main aim of this module is to convert the eight bit sequential data into single bit serial data so then Data can be transmitted. Transmitter operation is very simple, from line state, the timing cannot be determined, and it is not bound to any fixed timing intervals. Here is an important thing is that the DATA_IN is stored in holding register. Now after that DATA_IN stored in the Holding register then that data is send to the intermediate register. After that data which is transferred to the intermediate register, the start bit of that data will be zero. Now as we know the intermediate register has 9 bits. Whenever the shift signal is high, the least significant bit of intermediate register, e.g., the start bit comes first at an output of the transmitter and acts as input to the receiver. When the whole data has been sent, now the transmitter provides a parity bit to the receiver. Now here we have to check the CRC error, for that we give the divisor as the user input, and once the entire data has been transmitted, the transmitter produces remainder to the receiver, so now CRC_OUT provide by the receiver.

B.RECEIVER

From the name the main aim of this module is tostore the transmitter outputs into the intermediate register. It will save TX_OUT. Whenever the receiver get the high signal of the load, a start bit is to be sent by a transmitter to a receiver so then the receiver can starts. And, wheneverthe receiver get the high signal at no load,the transmitted data from the transmitter get shifted to the intermediate transistor at receiver side and then it gives the eight bit serial data which we have provided at inputside of the transmitter. Now entire data has been sent the parity error, so then the CRC error has been checked out and served as the I/P to the transmitter. If there is parity error and CRC error occur, it means the transmitting data have some errors. The Baud rate generator is nothing it is a type of frequency divider.

III. TECHNOLOGY USED

We have used frequency scaling techniques where we are varying the frequency of different I/O standards so then we can analyse the demand for power at the different frequency. It makes our design more efficient. Frequency scaling technique is the best way of energy efficient techniques for FPGA based VLSI designs [4].

A) LOGIC FAMILY USED:

As per above discussion**,** we have going to implement an HSTL (High-Speed Transceiver Logic) IOSTANDARD based energy efficient Asynchronous Receiver Transmitter UART. It's because, as we know, to achieve speed and high performance, we can use HSTL (High-Speed Transceiver Logic) IOSTANDARD. The HSTL family which we used in this paper are consists of HSTL_I, HSTL_II, HSTL-I_18, HSTL_II_18 IO Standards. Frequency Scaling techniques are one of the best energy efficient technologies for FPGA, which is used in this paper. In this paper, it has been analysed the demand for total power of different IO Standard at different frequency level. We have analysed that the increasing of frequency, the clock and IO power is increasing the power consumption in different IO standards. The operational voltage of classI in case of HSTL, is 1.2 volt and for class II HSTL is 1.5-1.8 volt. In our paper, we have used HSTL_I (Operating Voltage-1.2 Volt), HSTL_II (Operating Voltage-1.5 volt), and HSTL_I_18 (Operating Voltage- 1.8 Volt), HSTL_II_18 (Operating Voltage- 1.8 Volt). The use of HSTL IO Standard helps us to achieve speed and high performance in addition to energy efficiency. Researcher have used Xilinx ISE Design Suite and results are tested on 28 nm FPGA [5-10].Figure3 is showing the range of frequency which we have used in this paper.

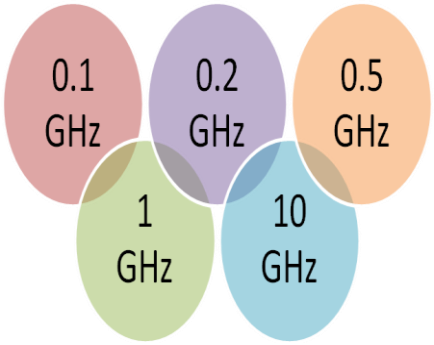


Fig.3. Range of Frequency

As we can see in Table 2, clock power is reduced by 89.25%, 94.21%, 97.52%, and 98.34%, when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. Also, when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, 0.1GHz, the Logic power is also reduced by 80%, and 100% respectively. There is a reduction in signal power by 89.47%, 94.73% and 100% when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. We can illustrate these results in Fig.4.

Table 2. Values of Clock, logic, & Signals at Different Frequencies with HSTL_I

Frequency (GHz)	0.1 GHz	0.2 GHz	0.5 GHz	1 GHz	10 GHz
Clock Power (Watts)	0.002	0.003	0.007	0.013	0.121
Logic Power (Watts)	0	0	0	0.001	0.005
Signal Power (Watts)	0	0	0.001	0.002	0.019

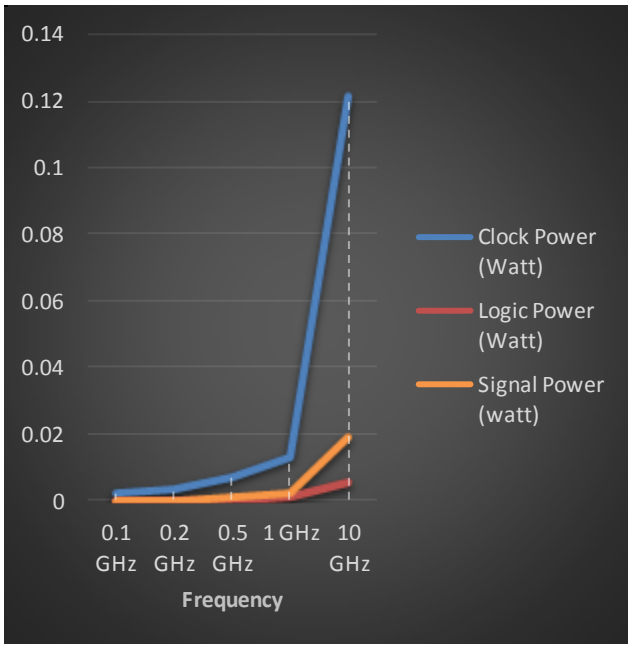


Fig 4. Values of Clock, logic, & Signals at different Frequencies with HSTL_I

As we can see in Table 3, IOs power is decreasing by 45%, 47.32%, 48.85%, and 49.61% whenwe reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. Also, when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, 0.1GHz, the leakage power drop is reduced by 13.33%, and this drop remains constant. The demand for Total power is reduced by 65.29%, 68.72%, 71.13%, and 71.82% when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. We can illustrate these result in Fig.5.

Table 3. Values of I/Os, Leakage and Power at different Frequencies with HSTL_I

Frequency (GHz)	0.1 GHz	0.2 GHz	0.5 GHz	1 GHz	10 GHz
IO-Power (Watts)	0.066	0.067	0.069	0.072	0.131
Leakage Power (Watts)	0.013	0.013	0.013	0.013	0.015
Total Power (Watts)	0.082	0.084	0.091	0.101	0.291

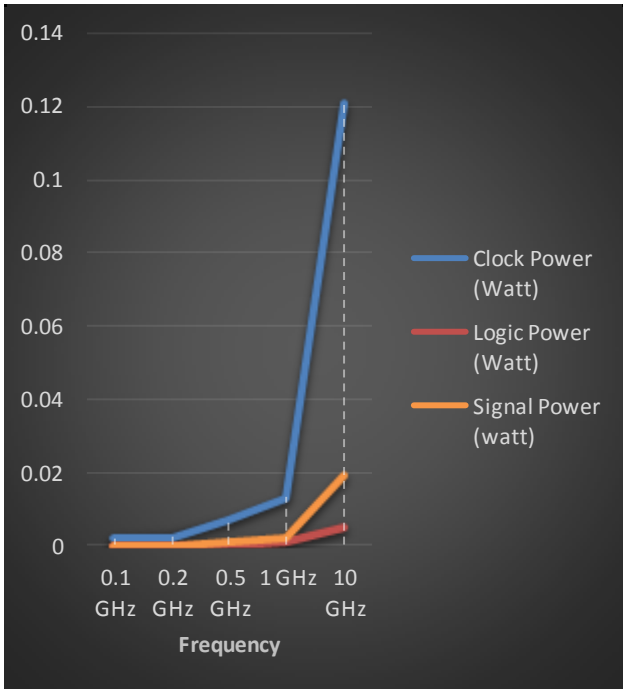


Fig 5. Values of I/Os, Leakage and Power at different Frequencies with HSTL_I

As we can see in Table 4, Clock power is reduced by 89.25%, 94.21%, 97.52%, 98.34%, when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, 0.1GHz respectively. Also, When we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, 0.1GHz, the logic power is also reduced by 80%, and 100% respectively. There is a reduction in signal power by 89.47%, 94.73% and 100% when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, 0.1GHz respectively. We can illustrate these results in Fig.6.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table 4. Values of Clock, logic, & Signals at Different Frequencieswith HSTL_I_18

Frequency (GHz)	0.1 GHz	0.2 GHz	0.5 GHz	1 GHz	10 GHz
Clock Power (Watts)	0.002	0.003	0.007	0.013	0.121
Logic Power (Watts)	0	0	0	0.001	0.005
Signal Power (Watts)	0	0	0.001	0.002	0.019

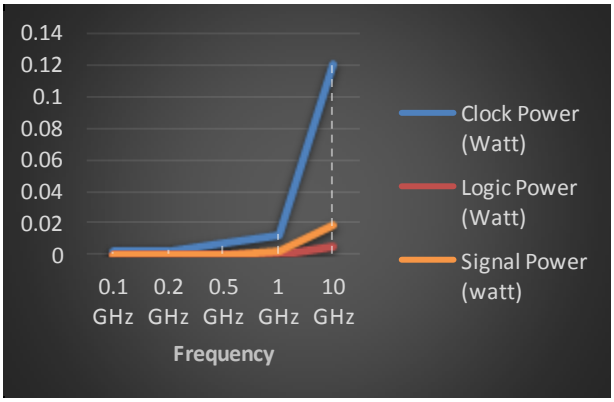
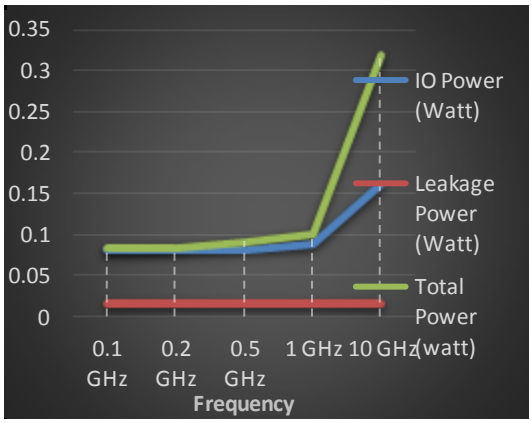


Fig 6. Values of Clock, logic, & Signals at different Frequencies with HSTL_I_18

As we can see in Table 5, IOs power is decreasing by 45.65%, 48.12%, 49.37%, and 50% when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. Also, when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, 0.1GHz, the leakage power drop is getting cut by 12.5%, and then it remains constant. The demand for total power is reduced by 63.75%, 67.18%, 69.37%, and 70% when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. We can illustrate these result in Fig.7.

Table 5. Values of I/Os, Leakage and Power at different Frequencies with HSTL_I_18

Frequency (GHz)	0.1 GHz	0.2 GHz	0.5 GHz	1 GHz	10 GHz
IO-Power (Watts)	0.080	0.081	0.083	0.087	0.160
Leakage Power (Watts)	0.014	0.014	0.014	0.014	0.016
Total Power (Watts)	0.096	0.098	0.105	0.116	0.320



7. Values of I/Os, Leakage and Power at different Frequencies with HSTL-I-18

As we can see in Table 6, Clock power is reduced by 89.25%, 94.21%, 97.52%, 98.34%, when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. Also, when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, 0.1GHz, the logic power is also reduced by 80%, and 100% respectively. There is a reduction in signal power by 89.47%, 94.73% and 100% when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. We can illustrate these results in Fig.8.

Table 6. Values of Clock, logic, & Signals at Different Frequencies with HSTL_II

Frequency (GHz)	0.1 GHz	0.2 GHz	0.5 GHz	1 GHz	10 GHz
Clock Power (Watts)	0.002	0.003	0.007	0.013	0.121
Logic Power (Watts)	0	0	0	0.001	0.006
Signal Power (Watts)	0	0	0.001	0.002	0.020

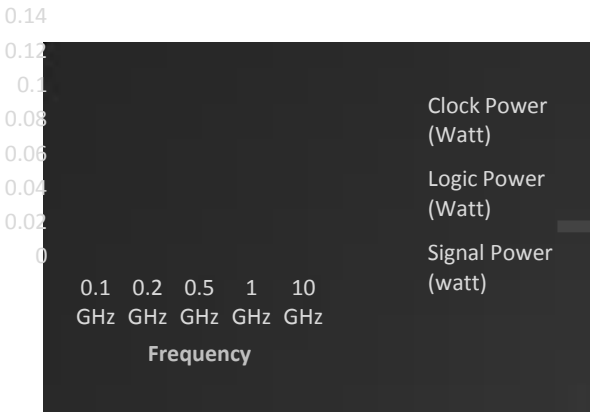


Fig 8. Values of Clock, logic, & Signals at different Frequencies with HSTL-II

As we can see in Table 7, The IOs power is decreasing by 36.37%, 38.19%, 39.10%, and 40% when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. Also, when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, 0.1GHz, the Leakage power drop is getting cut by 13.34%, and then it remains constant. The demand for total power is reduced by 63.46%, 66.78%, 69%, and 69.74% when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. We can illustrate these result in Fig.9.

Table 7. Values of I/Os, Leakage and Power at different Frequencies with HSTL_II

Frequency (GHz)	0.1 GHz	0.2 GHz	0.5 GHz	1 GHz	10 GHz
I/O Power (Watts)	0.066	0.067	0.068	0.070	0.110
Leakage Power (Watts)	0.013	0.013	0.013	0.013	0.015
Total Power (Watts)	0.082	0.084	0.090	0.099	0.271

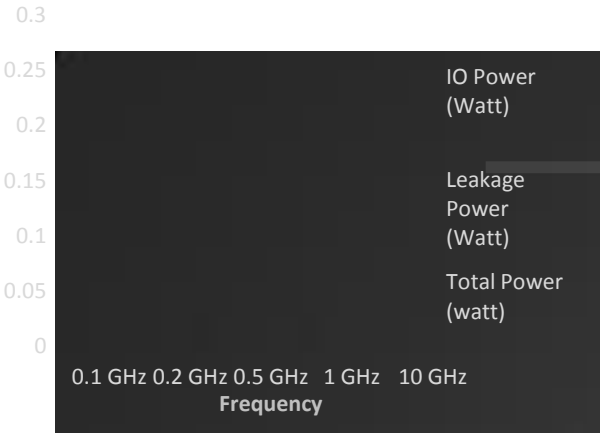


Fig 9. Values of I/Os, Leakage and Power at different Frequencies with HSTL-II

As we can see in Table 8, Clock power is reduced by 89.25%, 94.21%, 97.52%, 98.34%, when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. Also, when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, 0.1GHz, the logic power is also reduced by 80%, and 100% respectively. There is a reduction in Signals Power by 89.47%, 94.73% and 100% when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. We can illustrate these results in Fig.10.

Table 8. Values of Clock, logic, & Signals at Different Frequencies with HSTL_II_18

Frequency (GHz)	0.1 GHz	0.2 GHz	0.5 GHz	1 GHz	10 GHz
Clock Power (Watts)	0.002	0.003	0.007	0.013	0.121
Logic Power (Watts)	0	0	0	0.001	0.006
Signal Power (Watts)	0	0	0.001	0.002	0.020

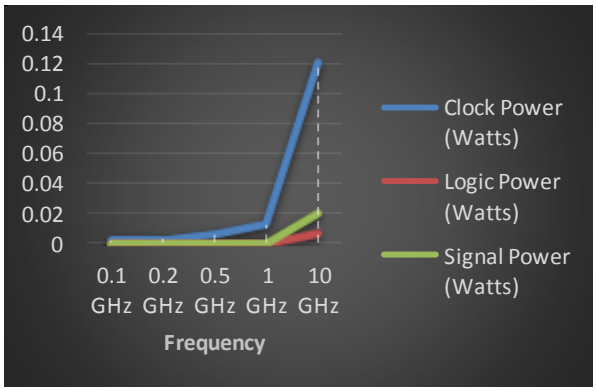


Fig 10. Values of Clock, logic, & Signals at different Frequencies with HSTL-II_18

As we can see in Table 9, IOs power is decreasing by 37.05%, 39.25%, 40%, and 40.75% when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. Also, when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, 0.1GHz, the leakage power drop is getting cut by 12.5%, and then it remains constant. The demand for total power is reduced by 61.62%, 65%, 67%, and 68% when we reduce operating frequency from 10GHz to 1GHz, 0.5 GHz, 0.2 GHz, and 0.1GHz respectively. We can illustrate these result in Fig. 11.

Table 9. Values of I/Os, Leakage and Power at different Frequencies with HSTL_II_18

Frequency (GHz)	0.1 GHz	0.2 GHz	0.5 GHz	1 GHz	10 GHz
I/O Power (Watts)	0.080	0.081	0.082	0.085	0.135
Leakage Power (Watts)	0.014	0.014	0.014	0.014	0.016
Total Power (Watts)	0.096	0.098	0.104	0.114	0.297

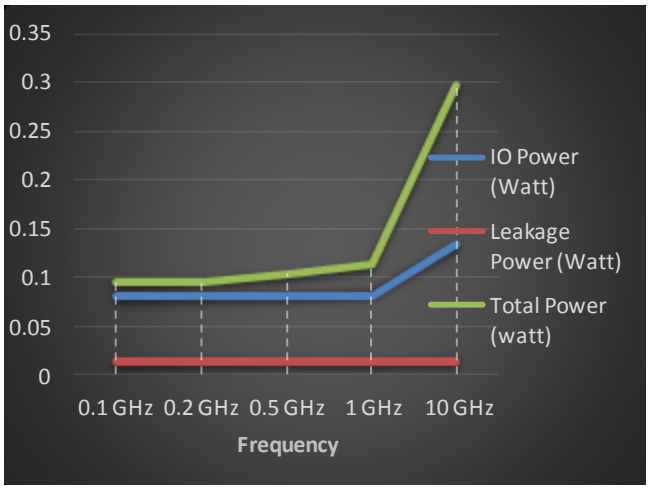


Fig 11. Values of I/Os, Leakage and Power at different Frequencies with HSTL-II_18

IV. CONCLUSION

We have designed this UART's with Verilog can be unified into FPGA for the achievement of reliable and compact data transmission. From the above analysis, it has been observed that with increasing the device operating frequency, the power demand by different HSTL IO Standard is also growing. Here is one observation that when we are growing the device operating frequency, there is no effect on logic power,and leakage power, but I/O power, signal power and clock power are increasing with frequency.In this paper, we have described that what is the UART working principle and how it works with different HSTL IO Standard family which we used. It is proved the communication can fast and very effective between the transmitter and the receiver.

V. FUTURESCOPE

There is wide scope to replace Spartan-6 FPGA with other FPGA families i.e. Automotive Artix-7, automotive Coolrunner-2, automotive Spartan, automotive Spartan-3A DSP, automotive Spartan-3A, automotive Spartan-3E, automotive Spartan-6, Spartan-3, Spartan-3E in our research work to make an energy efficient UART. We can also achieve energy efficient using other energy efficient techniques by changing O/P loads, clock getting, various design goals, impedance matching, Capacitance scaling technique, Thermal scaling and mapping etc.

REFERENCES

1. Jayesh More, Rushank Suryavanshi, Gaurav Dasarwar, Sivanantham S* and Sivasankaran K " FPGA Implementation of Universal Asynchronous Transmitter and Receiver" Online International Conference on Green Engineering and Technologies (IC-GET) 2015.
2. Gupta, Isha, et al. "28nm FPGA based Power Optimized UART Design using HSTL I/O Standards." Indian Journal of Science and Technology 8.17 (2015)
3. Singh, Sunny, et al. "Thermal-aware low power universal asynchronous receiver transmitter design on FPGA." Computational Intelligence and Communication Networks (CICN), 2014 International Conference on. IEEE, 2014
4. Amanpreet, Vidhoytma,Simranpreet, Surbhi, Divjot, Wamika,"Thermally Aware LVCMOS based Low Power Universal Asynchronous Receiver Transmitter Design on FPGA" Indian Journal of Science and Technology, Vol 8(20), DOI:10.17485/ijst/2015/v8i20/84107 August 2015
5. Shivani, Bishwajeet, Amanpreet Kaur, Md Hashim, DMA Hussain, "HSTL IO Standard Based Energy Efficient Multiplier Design using Nikhilam Navatashcaramam Dashatah on 28nm FPGA", International Journal of Conttrol Automation, Vol.8, No.8, pp.35-44, 2015
6. Isha Gupta, Garima, Swati Singh, Harpreet Kaur, Deepshikha, Aamir, "28nm FPGA based Power Optimized UART Design using HSTL I/O Standards", IJST- Vol 8(17), DOI: 10.17485/ijst/2015/v8i17/76859, August 2015
7. Vandana Thind, Bishwajeet Pandey, and DM Akbar Hussain. "Power Analysis of Energy Efficient DES Algorithm and Implementation on 28nm FPGA." Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES), 2016 IEEE Intl Conference on. IEEE, 2016.
8. Kartik Kalia, Bishwajeet Pandey, and D. M. A. Hussain. "SSTL based thermal and power efficient RAM design on 28nm FPGA for spacecraft." In Smart Grid and Clean Energy Technologies (ICSGCE), 2016 International Conference on, pp. 313-317. IEEE, 2016.
9. Bishwajeet Pandey, Vishal Jain, Rashmi Sharma, and Mragang Yadav. "Scaling of Supply Voltage in Design of Energy Saver FIR Filter on 28nm FPGA." *International Journal of Control and Automation* 10, no. 12 (2017): 77-88.
10. Bishwajeet Pandey, Nisha Pandey, Amanpreet Kaur, DM Akbar Hussain, Bhagwan Das, and Geetam S. Tomar. "Scaling of Output Load in Energy Efficient FIR Filter for Green Communication on Ultra-Scale FPGA." Wireless Personal Communications: 1-14.