Full-chip design of a unidirectional USB transmitter using serial and parallel data communication modes. (USB controller)

Fahim Mahmud
ID:20200105212
Department of Electrical and Electronic Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
fahim.eee.20010512@aust.edu

Md. Mamunar Rahoman Roton
ID:20200105196
Department of Electrical and Electronic Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
mamunar.rahoman28@gmail.com

Noor-E-Sadik Rupam
ID:20200105216
Department of Electrical and Electronic Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
noor.eee.200105216@aust.edu

Humayra Tahassum
ID:20200105221
Department of Electrical and Electronic Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
humayra.eee.2001051221@aust.edu

Saji Pal
ID:20200105235
Department of Electrical and Electronic Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
saji.eee.200105235@aust.edu

***Abstract*—This project focuses on the full-chip design of a unidirectional USB transmitter using both serial and parallel data communication modes. The chip's functionality is specifically tailored to handle input and output tasks via distinct communication methods. Input signals include a clock, reset, and control bits, which are processed to generate corresponding parallel or serial output data streams. The design process involved detailed I/O pin planning, finite state machine (FSM) design, and flow optimization. Verification was conducted using Verilog testbench code on ModelSim, and the final chip schematic was synthesized and optimized through Cadence Genus and Innovus tools. The design underwent several stages of optimization, including clock tree synthesis (CTS) and physical verification, with the final results showing minimal violations. This work demonstrates a robust approach to USB transmitter design, balancing functional requirements with hardware constraints to achieve an efficient implementation.**

Keywords— USB Transmitter, Serial Data Communication, Parallel Data Communication, Full-Chip Design, I/O Pin Planning, Finite State Machine (FSM), ModelSim, Verilog Testbench, Cadence Genus, Cadence Innovus, Clock Tree Synthesis (CTS), Physical Verification, STA.

I. INTRODUCTION

The rapid advancement of digital communication systems has necessitated the development of efficient data transfer protocols and hardware components. USB (Universal Serial Bus) technology, being one of the most widely adopted standards, plays a crucial role in facilitating seamless data exchange between devices. Among the key components of a USB system is the USB transmitter, responsible for encoding and transmitting data from one device to another. This project focuses on the design and implementation of a unidirectional USB transmitter that supports both serial and parallel data communication modes. The objective is to create a high-performance, reliable transmitter that efficiently handles data conversion and transmission while minimizing hardware complexity. In this project, we begin with an overview of the design requirements and constraints, followed by a detailed discussion of the design methodology. The implementation process includes I/O pin planning, finite state machine (FSM) design, and various state of optimization using industry-standard tools such as Cadence Genus and Innovus. Verification of the design is conducted using Verilog testbench simulations in ModelSim, ensuring the correctness and efficiency of the final chip layout. The significance of this project lies in its contribution to the ongoing evolution of USB technology, providing a blueprint for future designs that require a balance between functionality and hardware efficiency. Through this work, we aim to demonstrate the practical challenges and solutions in designing a robust USB transmitter, paving the way for further advancements in digital communication hardware.

II. METHODOLOGY

The design and implementation of the unidirectional USB transmitter involved a systematic approach, encompassing various stages from conceptualization to physical verification. The methodology is structured as follows:

A. Design Specification and Requirement Analysis

The initial step involved defining the design specifications, including the transmitter's operational modes (serial and parallel), input/output requirements, and performance targets. Key parameters such as clock frequency, input/output delay, and fanout were identified to ensure the design meets the functional and timing requirements.

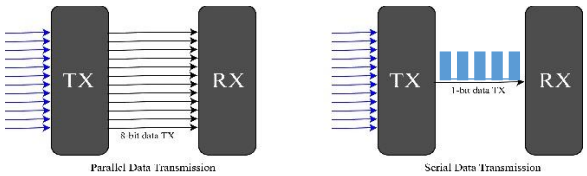

Figure 1.1: Serial and parallel data communication system.

From practical device there comes some design specification as follows-Basically, we can consider two types of operation from this designed usb controller. First one is Navigation operation and second one is Control operation. And these operations can be controlled by two working modes, first one

is serial data communication and second one is parallel data communication mode. Serial data communication is all available bits will transmit through a single wire or there will be only one dedicated output pin for data communication. This serial data communication method works based on data compression technology. And another one parallel data communication transmits all available bits at a same time through the separately dedicated port.

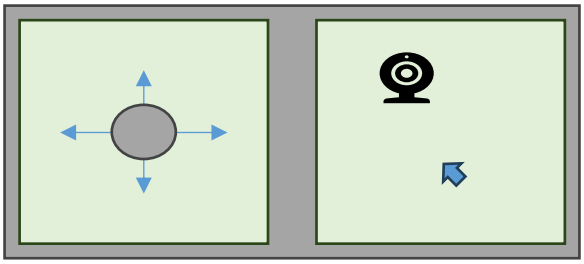

Figure 2.1: Working method of usb controller.

The above picture (figure 2.1) describes that left corner box is the representation of movement of a specific object which we named Navigation and right corner box represents that a mouse cursor available for controlling specific operation for a specific icon.

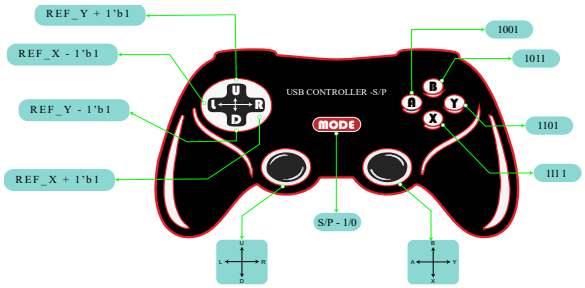

Figure 2.2: Practical implementation of usb controller.

The above picture (figure 2.2) is the representation of pin planning and no. of dedicated buttons for a specific operation.

As Navigation operation is one of the major parts of this project so there will be a reference coordinate for initial pointing of mouse cursor. Here we introduce REF_X for X axis, REF_Y for Y axis and dedicated buttons for navigation is L, R, U, D. For controlling - A, B, X, Y are the dedicated buttons.

Operations of dedicated buttons	
Input Buttons	Output Operation
mode	Serial / Parallel
L	REF_X – 1'b1 [REF_X = 1010 binary for 10]
R	REF_X + 1'b1 [REF_X = 1010 binary for 10]
U	REF_Y+ 1'b1 [REF_Y = 1000 binary for 8]
D	REF_Y– 1'b1 [REF_Y = 1000 binary for 8]
A	Enter to the Application [1001]
B	Exit from the Application [1011]
X	Available options for the application [1101]
Y	Properties of the application [1111]

Table 2.1: Table for dedicated button operations.

The pin planning of the chip which will be use inside the controller-

I/O pin planning	
INPUT	OPUTPUT
mode	8-bit Coordinate out in parallel mode
rst	4-bit Control out in parallel mode
clk	1-bit Coordinate out in serial mode
4-bit move	1-bit Control out in serial mode
4-bit operation	

Table 2.2: Pin planning for usb controller chip.

Practical Pin Implementation of designed usb controller chip (draft copy) –

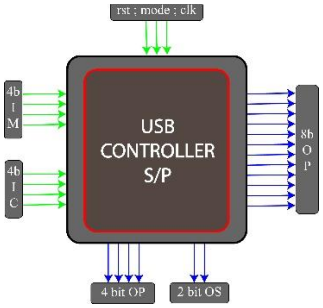

Figure 2.3: Draft copy of pin implementation of designed usb controller.

B. **I/O Pin Planning**

A detailed I/O pin configuration was developed, specifying the assignment of input and output signals. This included Inputs: Mode selection, clock (clk), reset (rst), move, and control signals. Outputs: 8-bit coordinate output and 4-bit control output in parallel mode, and 1-bit coordinate and control outputs in serial mode. This planning was critical to ensuring proper signal routing and avoiding potential conflicts.

C. **Finite State Machine (FSM) Design**

An FSM was designed to manage the operation of the USB transmitter. The FSM controlled the data flow based on the input mode, ensuring correct data transmission in either serial or parallel mode. The FSM was modeled and verified using state diagrams, ensuring all states and transitions were accounted for.

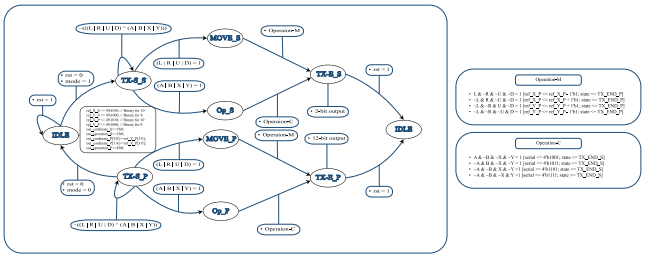

Figure 2.4: FSM diagram and design flow of designed usb controller.

D. **RTL Design and Synthesis**

The design was implemented in Verilog, creating a Register Transfer Level (RTL) model of the USB transmitter.

Synthesis was performed using the Cadence Genus tool, translating the RTL model into a gate-level netlist. This step involved optimizing the design for area, speed, and power.

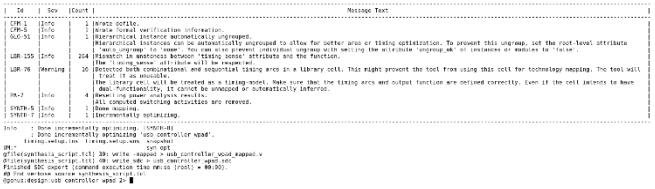

Figure 3.1: Synthesis summary message | Generated by Cadence Genus Synthesis Tool.

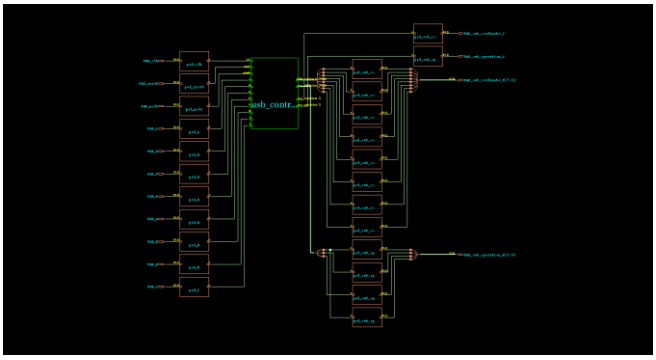

Figure 3.2: Schematic diagram with IO Pad cell | Generated by Cadence Genus Synthesis Tool.

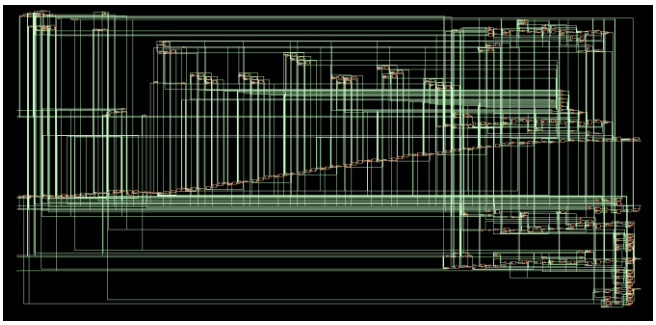

Figure 3.3: Full Schematic diagram | Generated by Cadence Genus Synthesis Tool.

E. **Verification using Verilog Testbench (ModelSim)**

A comprehensive testbench was developed in Verilog to simulate the USB transmitter's behavior. The ModelSim tool was used to run these simulations, generating output waveforms to verify the correctness of the design against the expected functionality. Any discrepancies found during simulation were addressed through iterative design modifications.

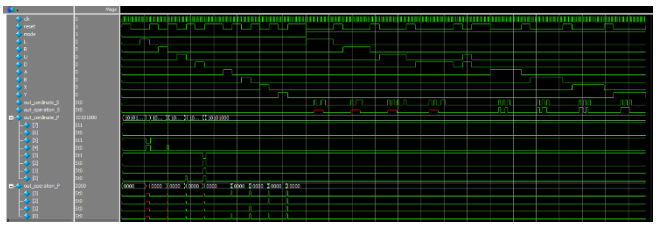

Figure 3.3: Logic state verification using Verilog Testbench code.

F. **Physical Design and Optimization**

The gate-level netlist was imported into Cadence Innovus for physical design. This stage involved several critical steps: Placement: Arranging the cells on the silicon to optimize area usage. Clock Tree Synthesis (CTS): Ensuring that the clock distribution meets timing requirements across the chip. Routing: Connecting all the cells according to the netlist while minimizing delays and avoiding design rule violations. Design optimization was carried out to improve performance metrics such as timing, area, and power consumption.

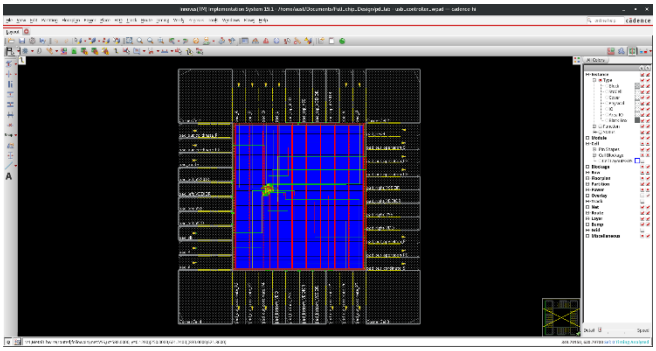

Figure 4.1: Design Placement

G. **Static Timing Analysis (STA)**

Pre-CTS, CTS optimization, Post-routing timing report, Post-routing optimization are parts of Static Timing Analysis (STA), was performed to verify that the design met all timing constraints. This analysis was done before and after the Clock Tree Synthesis (CTS) to identify and resolve any timing violations.

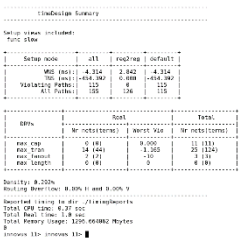

Figure 3.4: STA report before CTS (preCTS)

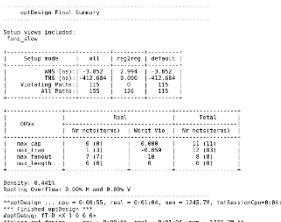

Figure 3.5: STA report after preCTS optimixation

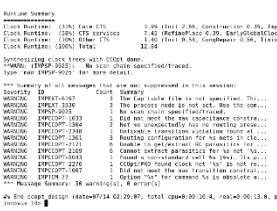

Figure 3.6: Optimized timing report of CTS.

Figure 3.7: Clock Tree Formation after CTS.

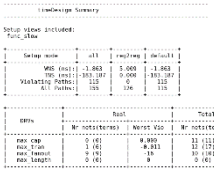

Figure 3.8: Timing report after post-route.

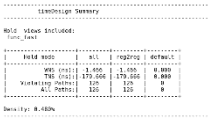

Figure 3.9: Timing report after post-route -hold.

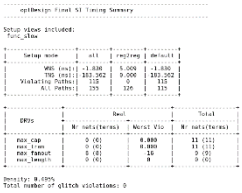

Figure 3.10: Design optimization after post-route.

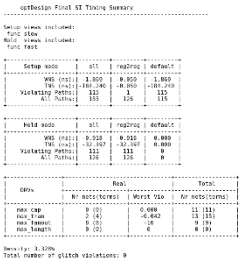

Figure 3.11: Design optimization after post-route -hold.

H. **Physical Verification**

The final step involved running physical verification using Cadence Innovus, focusing on: Design Rule Check (DRC): Ensuring the design adheres to the manufacturing process constraints. Layout Versus Schematic (LVS) Check: Verifying that the physical layout matches the schematic design. Antenna Effects and Power/Ground Short Verification: Ensuring reliability against manufacturing-induced defects.

DRC : Design Rule Check is the term by which a physical designer can identify a design rule violation. In our project, there were several types of DRC violations because of PAD cells. The VDD and VSS of the PAD cell and power mesh were connected manually. In the process of connecting PAD cell VDD and VSS to power mesh VDD and VSS, the connecting via mismatched the DRC rule.

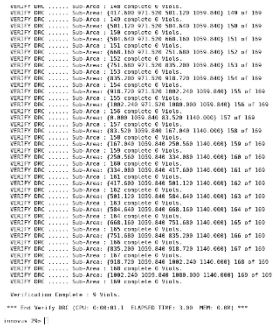

Figure 4.2: DRC Verification.

LVS : Layout Vs. Schematic is the term by which a physical designer can identify or check the layout and schematic violations. In our project, there were several types of LVS errors because of PAD cells. The reason was that the VDD and VSS connections of the PAD cell to the power mesh were not generated after synthesis, but we connected them manually.

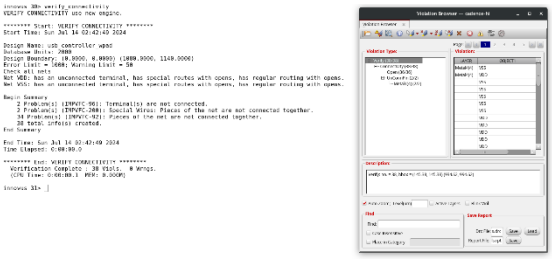

Figure 4.3: LVS Verification.

ARC : Antenna Rule check is the term by which a large area of metals that might affect manufacturing. Ensure that the transistors of the chip are not destroyed during fabrication.

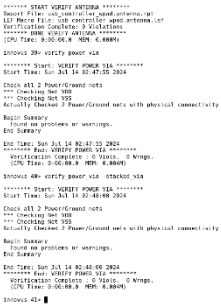

Figure 4.5: Power via process Antenna verification

PG Short : Power Ground short checks if any Power and Ground net shorts together. In our Project the special route PG was connected to power mesh that was legal but Innovus could not identify that was legal.

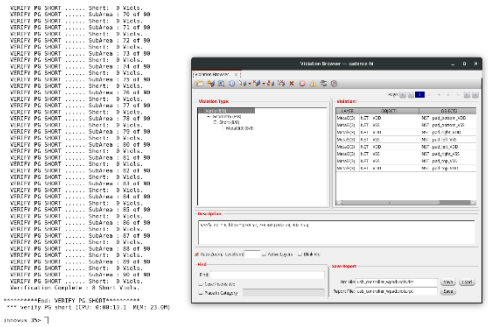

Figure 4.6: PG Short Verification.

Standard cell and Gate count :

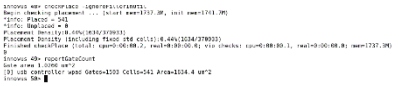

Figure 4.7: Placement Check and Gate Count

I. **Final Design Sign-off**

Upon successful completion of physical verification, the design was finalized, with detailed reports generated for clock frequency, delay metrics, and violation counts. The design was signed off after meeting all required specifications, ready for potential fabrication. This structured approach ensured that the unidirectional USB transmitter was designed, verified, and optimized to meet the desired specifications with high reliability and efficiency.

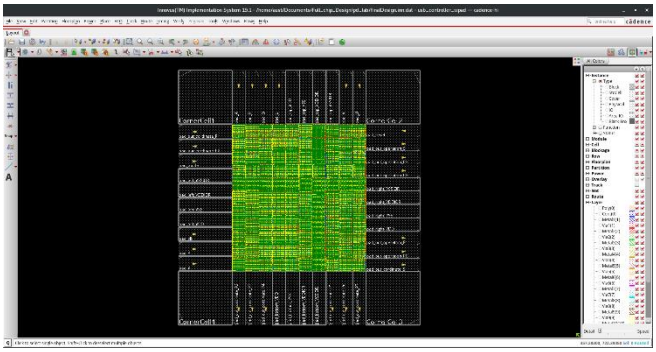

Figure 4.7: Final Design of Full Chip.

III. RESULT ANALYSIS

The result analysis of the unidirectional USB transmitter design focuses on evaluating the performance, optimization effectiveness, and overall success of the project against the predefined specifications. The key aspects analyzed include timing, area, power consumption, and the number of violations encountered during physical verification.

Design Criteria					
Initial Clock Freq. (MHZ)	Max Tran. (ns)	Driving cell	Input Delay (ns)	Output Delay (ns)	Max Fanout
205	0.9	BUFX8	1.86	2.38	8

Physical Verification						
Total No. of cells	Total No. of gates	Gate Area (Um^2)	No. Of DRC Viols.	No. Of LVS Viols.	No. Of ARC Viols.	No. Of PG short Viols.
541	1593	1634.4	9	38	0	8

Here, all physical verification techniques have been applied, and we found some violations. All the violations are due to power and ground wire specification mismatches. Cadence Innovus is not a verification tool; it is just a Place and Route tool. However, we applied some verification operations through this tool. The main reason for these violations is that if any power rail goes over any ground rail, it shows some DRC, LVS, and PG short errors.

IV. CONCLUSION

The design and implementation of a unidirectional USB transmitter with both serial and parallel data communication modes have been successfully achieved through a structured and iterative approach. The project began with a clear definition of design specifications, followed by detailed I/O pin planning, FSM design, RTL implementation, and thorough verification using advanced industry tools such as Cadence Genus, Innovus, and ModelSim. Key performance metrics, including timing, area, and power consumption, were optimized to meet stringent design requirements. The physical design was meticulously verified through Design Rule Checks (DRC), Layout Versus Schematic (LVS) checks, and power integrity verifications, ensuring that the final design was free of critical violations and ready for fabrication. The successful resolution of challenges related to timing closure, physical verification, and design optimization underscores the effectiveness of the methodology employed. The final design not only meets the functional requirements but also demonstrates robustness and efficiency, making it a viable solution for real-world USB communication systems. This project contributes to the broader field of digital communication hardware by providing a blueprint for designing efficient USB transmitters. The techniques and tools used in this project can serve as a reference for future designs, potentially leading to further innovations in USB technology and other communication protocols. Overall, the project highlights the importance of a systematic design process, rigorous verification, and iterative optimization in achieving high-performance digital systems.

V. REFERENCES

[1] LIU Wan, HE Dao-Jun., TAN Ming. The Design and Application of FPGA. Beijing: Tsinghua University Press, China, June 2006.

[2] Guo Shu-Tao, JING Yong-Zhi. Serial Communication Based on FPGA. The transaction of Beijing Electronic Science and Technology Institute, China, Vol .14, No.4, December 2006.

[3] JIANG Wei, WANG Kai-Jun., CHEN Han-Mei. A Design of Serial Communication Based on FPGA. Journal of Engineering Design, China, Vol.10, No. 2, April 2003.

[4] ZHANG Yi. MCS51 Microcontroller Application and Device. Harbin: Press of Harbin in Institute of Technology,1992.

[5] HOU Bo-Heng. VHDL & Design of Digital Circuits.Xi'an:Xidian University Press, China, 2009.

[6] LIU Le-shan. The Technology and Application of Microcomputer Interface.Wuhan:Huanzhong University of Science and Technology Press, China,2000.

[7] LIU-An-jie, HOU lv, DUAN Zhi-jiao. VGA Display and Serial Communication Based on FPGA. Indurstry Control Compter,China, Vol. 19, No.6,2006.

[8] F. S. Collins, L. D. Brooks, and A. Chakravarti, “A DNA polymorphism discovery resource for research on human genetic variation,” Genome Research, vol. 8, pp. 1229-1231, Dec 1998.

[9] J. C. Stephens, J. A. Schneider, D. A. Tanguay, J. Choi, and T. Acharya,” Haplotype variation and linkage disequilibrium in 313 human genes,” Science, vol. 293, pp. 489-493, 2001.

[10] R. S. Wang, L. Y. Wu, Z. P. Li, and X. S. Zhang,” Haplotype reconstruction from SNP fragments by minimum error correction,” Bioinformatics, Vol. 21, pp. 2456-2462, 2005.

[11] D. He, A. Choi, K. Pipatsrisawat, A. Darwiche, and E. Eskin,” Optimal algorithms for haplotype assembly from whole-genome sequence data,” Bioinformatics, vol. 26, pp. i183-i190, Jun 15 2010.

[12] W. Qian, Y. Yang, N. Yang, and C. Li,” Particle swarm optimization for SNP haplotype reconstruction problem,” Applied Mathematics and Computation, vol. 196, pp. 266-272, 2008.

[13] S. Levy, G. Sutton, P. C. Ng, L. Feuk, and A. L. Halpern,” The Diploid Genome Sequence of an Individual Human,” PLoS Biol, vol. 5, p. 254, 2007.

[17] Bansal and V. Bafna. “Hap CUT: an efficient and accurate algorithm for the haplotype assembly problem,” Bioinformatics, vol. 24, pp. I153-I159,2008.

[14] Panconesi and M. Sozio,” Fast Hare: A Fast Heuristic for Single Individual SNP Haplotype Reconstruction Algorithms in Bioinformatics.” vol. 3240, ed: Springer Berlin / Heidelberg, 2004, pp. 266-277.

[15] K. Taeho and J. Hyun,” ClustalXeed: a GUI-based grid computation version for high performance and terabyte size multiple sequence alignment,” BMC Bioinformatics, vol. 11, p. 467, 2010.