Design and Implementation of Bluetooth Microcontroller in System-on-Chip (SoC)

Hang Suan Wang[1, a)], Asral Bahari Jambek[1, 2, b)], Zulfiqar Ali bin Abd Aziz[3)], Mohd Nazrin Md Isa[1)], Azizi Harun[1)], Shaiful Nizam Mohyar[1)]

[1]*Faculty of Electronic Engineering Technology, Universiti Malaysia Perlis (UniMAP), 02600 Arau, Perlis, Malaysia.*
[2]*Micro System Technology, Centre of Excellence (CoE), Universiti Malaysia Perlis (UniMAP), 02600 Arau, Perlis, Malaysia.*
[3]*Collaborative Microelectronic Design Excellence Center, Universiti Sains Malaysia, Sains@USM Level 1, Block C, No 10, Persiaran Bukit Jambul, 11900 Bayan Lepas, Penang.*

Author Emails
[a)]*wanghs92@gmail.com*
[b)]*Corresponding author: asral@unimap.edu.my*

Abstract. Wireless microcontrollers have become widely used in domestic and industrial applications, where Bluetooth is one of the most popular wireless communication mediums. This paper discusses the design and implementation of a wireless Bluetooth microcontroller System-on-Chip (SoC) using Silterra 180nm CMOS technology. It incorporates Cortex-M0 as the main processor and other essential peripherals for a microcontroller, such as a timer, watchdog, UART, and RTC. This paper demonstrates the gate-level simulation result of the integrated system where several firmware tests are loaded into the RAM and operate those peripherals to verify the overall system functionality. The simulation results show that the system is able to perform the data transmit and receive successfully.

INTRODUCTION

Wireless microcontrollers are widely used in our daily life nowadays. Bluetooth technology makes wireless microcontrollers more convenient and secure than other wireless communication approaches. Bluetooth microcontrollers can be applied in many applications, such as sensor nodes in a weather monitoring system that avoids using a long wiring network.

System-on-Chip (SoC) design methodology can integrate many parts of the Bluetooth peripherals on the same chip plane [1] and has proven to reduce the design's chip size. Thus, a Bluetooth microcontroller design using the SoC method has more advantages than separately assembled IC designs since it makes this wireless device smaller and more power efficient. In SoC design, Advanced Microcontroller Bus Architecture (AMBA) is one of the popular on-chip buses that interconnect all peripherals. Using the bus standard in SoC design reduces time to market since any peripherals that follow the standard can then be easily ported into the system.

In this paper, a wireless Bluetooth microcontroller design will be discussed and tested at the register transfer level (RTL) using the AMBA bus. The design will then be synthesized using the Synopsys EDA tool, and the simulation result at the gate-level simulation will be done. Software and hardware co-verification will be performed to verify the integration and functionality of the system. To verify the operation, a program will be loaded into the system to perform data transmission and receive using Bluetooth. The gate-level simulation shows that the system is able to perform the required operation successfully.

Authors' Final Manuscript.
Published paper link: https://doi.org/10.1063/5.0192104, AIP Conference Proceeding, 2898, 030008 (2024)

LITERATURE REVIEW

In papers [2,3], the authors present a methodology to integrate an intellectual property design (IP) into a system using IP Execution Requirement Model (IPERM) and IP delay models. These methods manage to secure the IP since the internal features remain obfuscated during IP validation. A set of appropriate information regarding each IP block is given to the system for effective IP integration. Information such as functional and timing constraints are extracted from the IP core. This method provides an essential element for IP integration and synthesis while the internal IP is protected. Timing constraints, system integration constraints, and data transfer delay are incorporated during the synthesis process. A co-simulation using Bus Functional Model and the synthesizable output design is carried out as shown in Fig. 1.

In the paper [4], some effective integration methodologies were discussed. According to the paper, an enhancement of IP by improving the test coverage and implementing Memory Build-In-Self-Test (MBIST) is important for IP reuse. One method is using a clock multiplexer to test multiple clock domains. Furthermore, the asynchronous reset must be disabled for correct scan shift and capture. Managing IP databases is another essential factor that can improve the integration process. As the internal and external sources of the IP may vary from the design requirement, modification in a structured and systematic IP database becomes crucial. Latency reduction and low power operation are required for some designs that can be modified easily in the database.

The paper in [5] presents the strategies to develop an IP interface for the AMBA bus. The author demonstrates the methodology by using an open-access IP and incorporating it with the AMBA interface. This interface can be integrated into an embedded system with minimal effort. Every interface employed a structure of two processes, which contain combinational logic and sequential logic. A modified GRLIB AMBA test framework was used to validate the interface with a debug interface that can be controlled using external stimuli.

Paper [6] presents the design of a Bluetooth low-energy controller, which provides an overview of the Bluetooth architecture and its protocol. The design consists of a Bluetooth host, Host Controller Interface, and Bluetooth Controller. The Bluetooth host stores the protocol stack, while the controller act as the physical layer, as shown in Fig. 4. The design was verified using sub-module-based testing, direct back-to-back testing, hardware and software co-verification, and random input verification. A PHY model was used to emulate two connected devices.

The author of the paper [7] describes the design of a Bluetooth low-energy device. It is designed as part of the IoT connection where CC2541 and a software stack from Texas Instruments were used in the experiment. The Bluetooth architecture has an attribute table that can be accessed using Attribute Protocol (ATT).

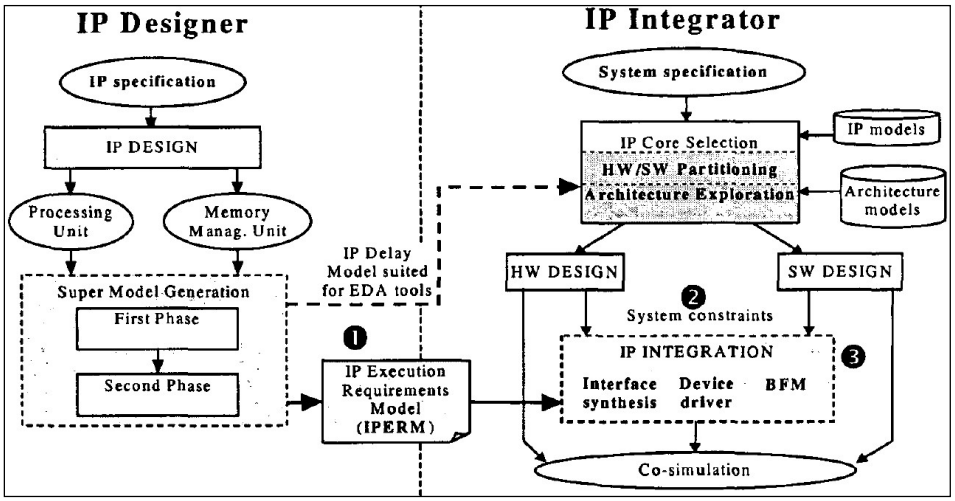

FIGURE 1. IPERM and IP delay model integration flow [2].

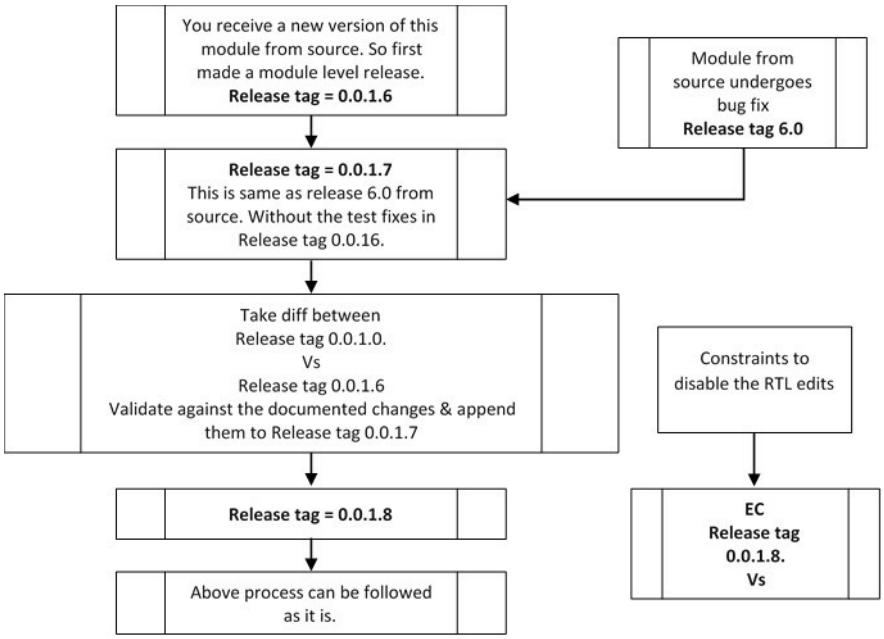

FIGURE 2. Block diagram for the Bluetooth microcontroller system [4].

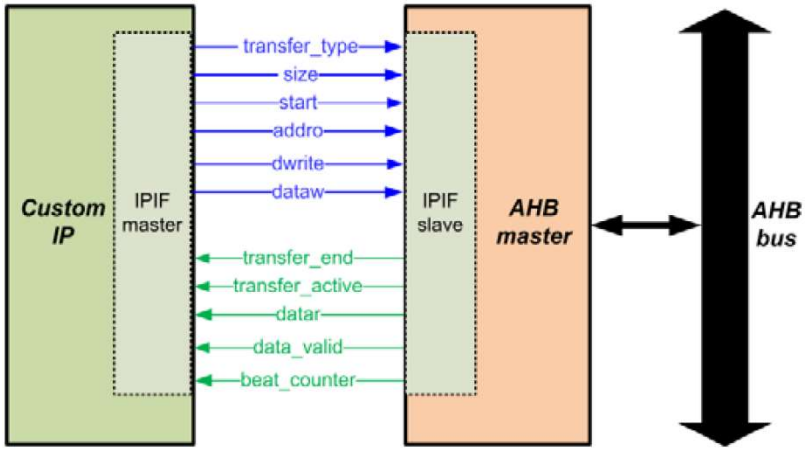

FIGURE 3. IP Interface incorporates AMBA [5].

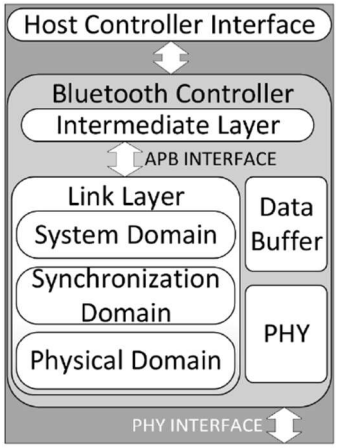

FIGURE 4. Structure of the Bluetooth controller [6].

51		0x2800	GATT Primary Service Declaration	A0:FF	**Accelerometer Service**
52		0x2803	GATT Characteristic Declaration	0A:35:00:A1...	Accererometer enable
	53	0xFFA1	Accererometer enable	01	Write "01" to enable, "00" to d
	54	0x2901	Characteristic User Description	41:63:63:65:...	Accel Enable
55		0x2803	GATT Characteristic Declaration	02:38:00:A2...	Accelerometer range
	56	0xFFA2	Accelerometer range	14:00	Range: 20=2G, 80=8G
	57	0x2901	Characteristic User Description	41:63:63:65:...	Accel Range
58		0x2803	GATT Characteristic Declaration	10:3B:00:A3...	Accelerometer X-coordinate
	59	0xFFA3	Accelerometer X-coordinate	01	
	60	0x2902	Client Characteristic Configuration	01:00	Write "01:00" to enable notific
	61	0x2901	Characteristic User Description	41:63:63:65:...	Accel X-Coordinate
62		0x2803	GATT Characteristic Declaration	10:3F:00:A4...	Accelerometer Y-coordinate
66		0x2803	GATT Characteristic Declaration	10:43:00:A5...	Accelerometer Z-coordinate

FIGURE 5. Bluetooth Device Monitor [7].

DESIGN ARCHITECTURE

In this paper, a wireless Bluetooth microcontroller is designed and implemented using the SoC design methodology. The design consists of an ARM Cortex-M0 processor [8], a Bluetooth processing unit, and other peripherals for the microcontroller, such as a timer, watchdog, and RTC. Figure 6 shows the block diagram of the microcontroller system. The Bluetooth module and other peripherals are integrated with the ARM Cortex-M0 processor using AMBA [9]. The microcontroller is designed as a general-purpose microcontroller with wireless communication capability.

The design flow commences with the system specification, which defines the system architecture features [10] [11], such as the purpose of the microcontroller, the type of processor, and the type of bus architecture to be used. At this stage, all specifications are still in an initial state and can be modified by the designer. Some of the specifications, such as the frequency of the clock, are still not finalized at this stage. A system bus protocol is being implemented on each IP to allow them to communicate with the processor. The System on Chip platform is then built and integrated with different IPs. This process is also known as system integration. An integration test is carried out to check for the correctness of the integration process, such as the IP port connection. A hardware-software co-verification is then performed on the design. If there are any errors, the IP must be redesigned and passed through integration and testing.

The design will proceed to the synthesis stage if it is error-free. In this work, the synthesis mapped the design to Silterra 180nm CMOS technology before re-simulated using gate-level simulation. At the gate-level simulation, a similar test is performed. If the gate simulation does not produce a result similar to the RTL test, a debug step has to be performed to investigate where the fault comes from. In this paper, the process ends after the synthesized netlist has been verified. Figure 7 shows the integration flow of the design.

The system architecture has a memory-mapped as defined in Table 1. The processor will use the memory map to call the peripherals. An application program in the hex format is stored in the ROM and is then used to simulate the system. The application program files are generated using Keil µVision software.

The design uses a single port 16kB SRAM model during the simulation and synthesis stage. Both model and library are produced using the Silterra 180nm CMOS SRAM generator. The generated memory includes Memory Build-In-Self-Test (MBIST) feature for future debugging purposes. The Bluetooth module has a link layer that acts as a controller to begin and end the transmission process. It is designed to modulate and demodulate the signal received by the antenna, as discussed in the paper [7].

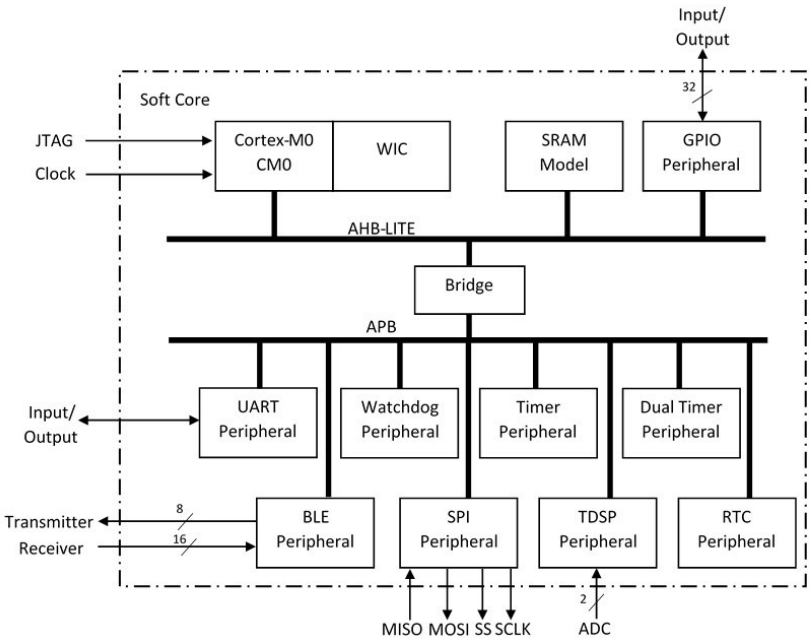

FIGURE 6. Block diagram for the Bluetooth microcontroller system.

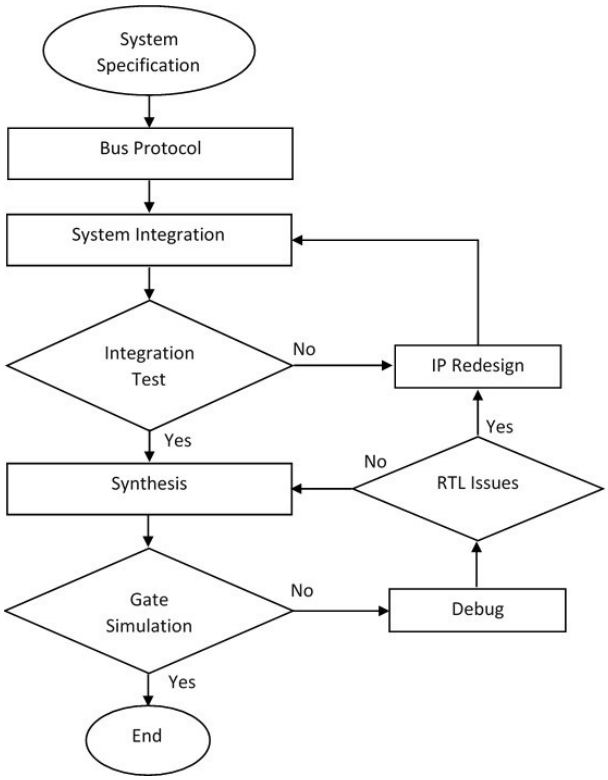

FIGURE 7. The flow of the Bluetooth Microcontroller design.

TABLE 1. System Memory map.

Module	Base Address	End Address
ROM	0x0000_0000	0x0000_FFFF
SRAM	0x2000_0000	0x2000_0FFF
GPIO0	0x4001_0000	0x4001_0FFF
GPIO1	0x4001_1000	0x4001_1FFF
Timer0	0x5000_0000	0x5000_0FFF
Timer1	0x5001_0000	0x5001_0FFF
Dual Timer	0x5002_0000	0x5002_0FFF
UART0	0x5004_0000	0x5004_0FFF
UART1	0x5005_0000	0x5005_0FFF
UART2	0x5006_0000	0x5006_0FFF
RTC	0x5007_0000	0x5007_0FFF
Watchdog	0x5008_0000	0x5008_0FFF
TDSP	0x5009_0000	0x5009_0FFF
Test Slave	0x500B_0000	0x500B_0FFF
SPI	0x500C_0000	0x500C_FFFF
Bluetooth	0x500D_0000	0x500D_FFFF

RESULT AND DISCUSSION

The microcontroller is tested by checking the SRAM storage content starting from the first to the end address. A random address check is also carried out in the simulation. Figure 8 shows the SRAM checking result using memory validation software from ARM. A false address is added to the software code to verify the software output matches the SRAM content. For this test, as expected, the result gives a hard fault interrupt by the processor, as shown in Fig. 9.

Next, the digital modulator for Bluetooth is then verified. The testing part was divided into transmitting and receiving processes to verify the wireless microcontroller. In the transmitting process, the microcontroller transmits the pre-stored data in the memory to the modulator. Figure 10 shows the modulated waveform.

The receive process was emulated using the Analogue Digital Converter (ADC) signal. The data set was generated using Matlab to simulate the data received from ADC input. The binary code is emulated as packet data transmitted by another Bluetooth device. Figure 11 shows the result of the received data, 0x0000_0006, which is the same as the injected data (i.e., generated data set). After receiving data by the Bluetooth module, the data was read in AMBA and PRDATA. The status of Bluetooth becomes idle once the data is received (the state status is 0). This operation is simulated using gate-level simulation, and the result is shown in Figure 12.

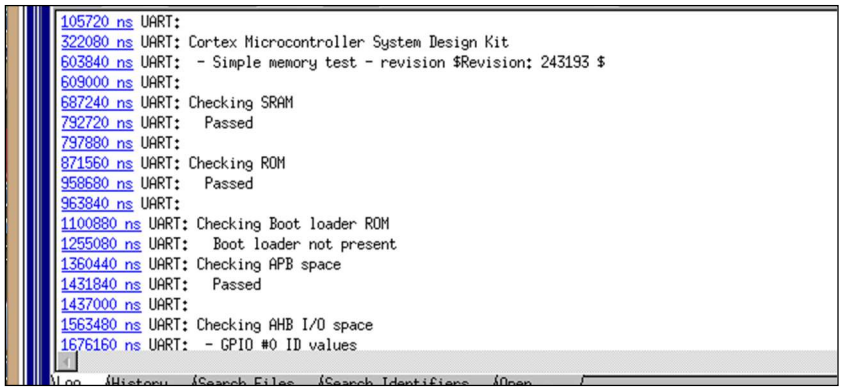

FIGURE 8. SRAM memory checking.

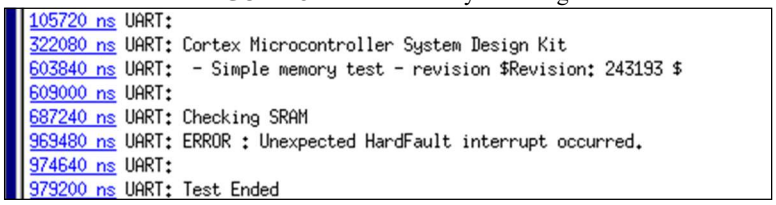

FIGURE 9. For example, write data beyond the SRAM address range.

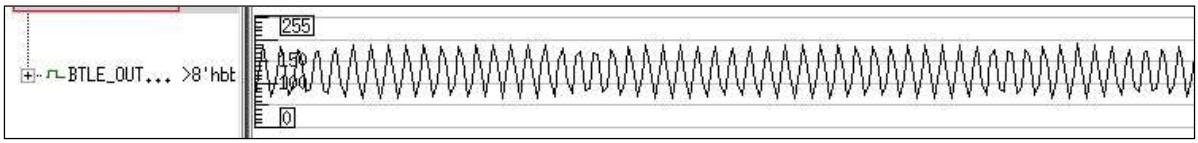

FIGURE 10. Modulation before transmission.

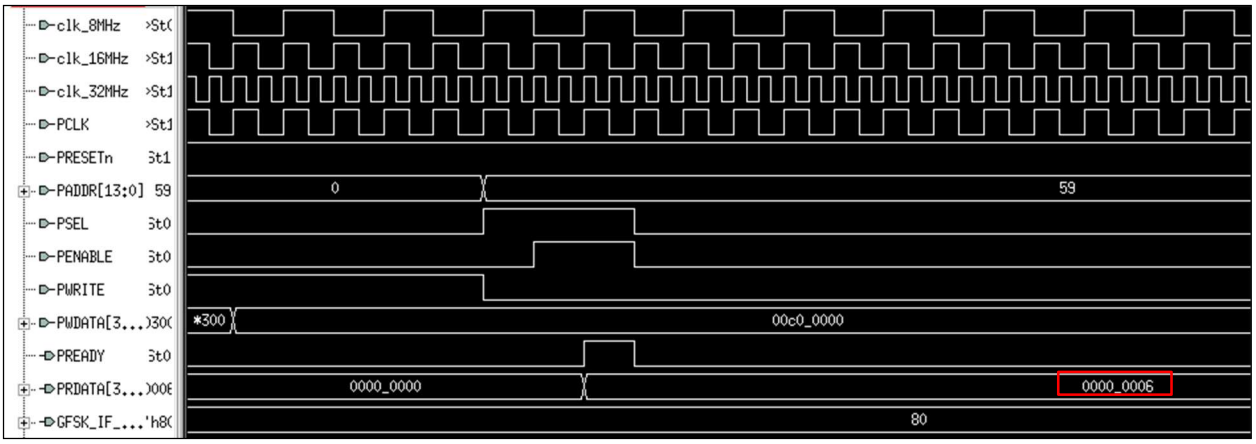

FIGURE 11. Data was successfully retrieved from the Bluetooth module.

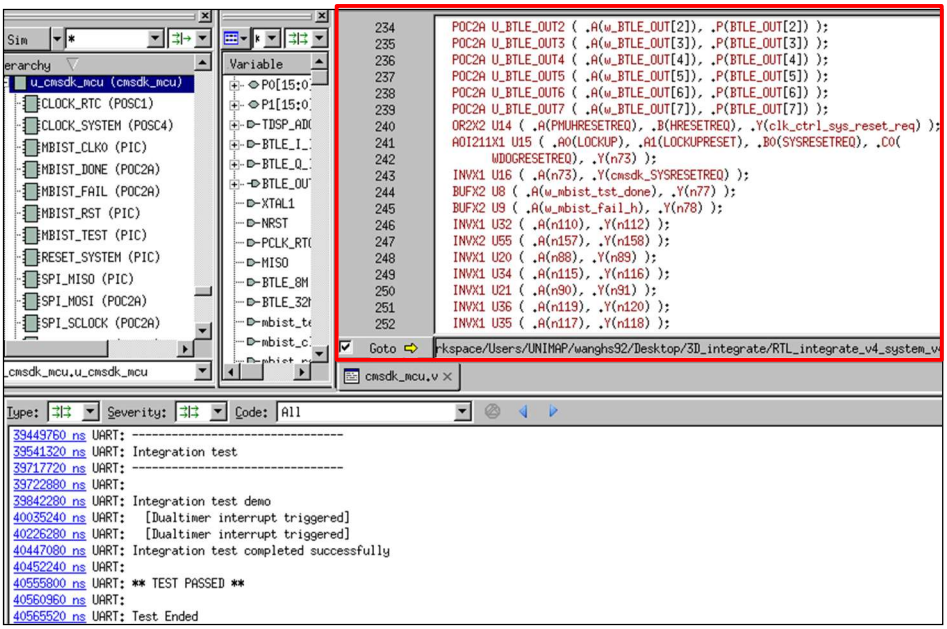

FIGURE 12. Data was successfully retrieved from the Bluetooth module.

CONCLUSION

The wireless device has become an essential tool in today's domestic and industrial system automation. This device could become smaller and more power efficient by using a System-on-Chip design methodology. This paper presents the design and integration of the Bluetooth wireless microcontroller SoC. The system consists of the processor, Bluetooth module, Real Time Clock, and other important peripherals. The verification of the whole system was performed in the Synopsys System-on-Chip EDA tools environment at the gate level. The simulation results show that the Bluetooth system is able to perform the data transmit and receive successfully.

ACKNOWLEDGMENT

This research was part of the Demand, Driven, and Development (3D) program supported by TalentCorp Malaysia.

REFERENCES

1. A. Brás, "Systems on Chip: Evolutionary and revolutionary trends," in 3rd Internal Conference on Computer Architecture (ICCA'02), 2002.
2. P. Coussy, A. Baganne, and E. Martin, "A design methodology for integrating IP into SOC systems," in Custom Integrated Circuits Conference, 2002, pp. 307–310.
3. V. S. Chakravarthi, *A practical approach to VLSI system on chip (SoC) design,* Springer International Publishing, 2020.
4. S. Sarkar, S. Shinde, and S. Chandar G., "An effective IP reuse methodology for quality System-on-Chip design," in International Symposium on System-on-Chip, 2005, pp. 104–107.
5. L. Acasandrei and A. Barriga, "Open library of IP module interfaces for AMBA bus," in Proceedings of the World Congress on Engineering 2015 (WCE), 2016, vol. 1, pp. 281–294.
6. P. Wiecha, M. Cieplucha, P. Kloczko, and W. A. Pleskacz, "Architecture and design of a Bluetooth low energy controller," in 23rd International Conference Mixed Design of Integrated Circuits and Systems (MIXDES) 2016, 2016, pp. 164–167.
7. S. Mischie, "On the development of Bluetooth low energy devices," in 2018 International Conference on Communications (COMM), 2018, pp. 339–344.
8. ARM Limited, *Cortex TM -M System Design Kit Technical Reference Manual,* pp. 1–168, 2013.
9. ARM Limited, *Supporting academia towards tomorrow's technology*, ARM Limited.
10. H. Bhatnagar, *Advanced ASIC Chip Synthesis*, 2nd ed. Kluwer Academic Publisher, 2002.
11. D. Haoming, et al. "Radio frequency system-on-chip design for digital receiver." 2018 IEEE 3rd International Conference on Integrated Circuits and Microsystems (ICICM), 2018.