A Survey on System-On-Chip Bus Protocols

Nidhi K, Shanta Rangaswamy

Department of Computer Science and Engineering, RV College of Engineering Bengaluru, India.

Abstract

Today's mobile or wireless devices use System-on-Chips that are incorporated with reusable IP Cores so that they are easily adaptable to new technologies and specifications. This paper gives an overview of various on-chip bus protocols such as Inter-Integrated Circuit (I2C), Improved Inter-Integrated Circuit (I3C), Peripheral Component Interconnect Express (PCIe), Serial Peripheral Interface (SPI) and Universal Asynchronous Receiver/Transmitter (UART), which are used by System-on-a-Chips (SoCs) to communicate with the devices. This paper provides a brief overview and a survey of functioning or implementation of bus protocols for various applications in a System-on-Chip.

Keywords

AMBA, ASIC, FPGA, I2C, I3C, PCI, RTL, SoC, SPI, TDMA, UART, USB, VHDL

1. Introduction

The process of decreasing the size of integrated circuits has become increasingly popular and important in the modern day. Because of user/customer demands, chip manufacturers are integrating more components to chipsets. These system-on-a-chips or chipsets consist of one or more programmable components, such as application-specific Intellectual Property (IP) cores, processor cores, or digital signal processors, on-chip memory, front-end and back-end designs, input/ output devices, and other circuits specifically designed for a given application. And buses are a shared communication medium in Communication Architecture (CA) of an SoC that the devices utilise to communicate with one another. This paper focuses on various on-chip communication bus protocols providing a survey of various protocols such as UART, I2C, SPI, PCIe, and I3C.

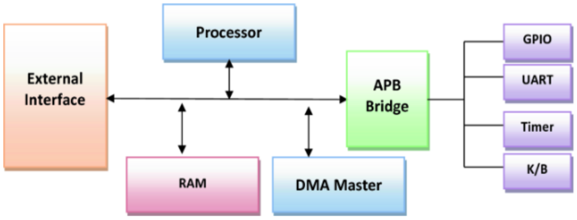

Fig. 1.1 Different bus protocols connected to processor

The above figure shows how different bus protocols can communicate with the SoC or processor using AHB-APB bus or AMBA bridge.

2. State of the Art Developments

Currently, the processor and several sensors, like cameras and accelerometers, communicate using the I2C protocol. I2C is a serial bus interface communication protocol that is built into devices. Despite I2C's wider adoption over the years, it is still lacking key essential functionalities, which is a problem given how many sensors and other components mobile as well as mobile-influenced systems are integrating. The I3C is a new serial communication interface specification that enhances the capabilities, efficiency, and power consumption of I2C while keeping backward compatibility with the majority of devices. I3C offers the simplicity, low pin count, flexible PCB design, and multi-drop benefits of I2C while also offering the faster data rates of 12.5 MHz and reduced power of SPI. In-Band Interrupts, efficient power management, Dynamic Addressing, , and Hot-Join are then added by I3C, along with better throughput for a specific frequency.

3. Literature Survey

This section of the paper contains a survey of present technologies and research available related to the on-chip bus protocols in an attempt to better understand the efforts that have gone into this field of study and also understand the implementation and applications of these bus protocols in SoC. The papers discussed in this section include work related to implementation of I2C using FPGA, VHDL, comparative studies of protocols such as SPI, UART and review papers on protocols like UART, I2C etc.

By automating the connections between components, plug-and-play designs can minimise the construction time for complicated systems. Although plug-and-play technologies have been successfully used in aerospace systems, their general adoption, especially in smaller satellites, is limited by the overhead of the interface electronics. Bryan Hansen discusses a straightforward plug-and-play interface in [1] that is based on the popular I2C standard and results in considerable optimizations of the interface circuits necessary to be plug-and-play compatible. At its most fundamental, Mini-PnP/SPA-1 is an I2C protocol overlay that can be simply implemented using hardware that already complies with this widely accepted standard. Networks of mini-PnP/SPA-1 devices can link to more traditional plug-and-play architectures via gateways. The SPA-1 hardware is similar to other legacy interfaces in that it is simple to integrate into plug-and-play software and that it supports essential plug-and-play functionalities including electronic datasheets and automated enumeration. The development and demonstration of COTS and rad-tolerant SPA-1 interface module versions, as well as the present state of the global programme, are covered in this paper.

Dmitry Levshun, Andrey Chechulin and Igor Kotenko et. al [2] have proposed a methodology for ensuring the security and dependability of the data transfer environment in microcontroller-based systems. The paper's main contribution is a new method for designing data transfer protocols that enables creating suggestions for enhancing the functionality of the data transfer protocols to satisfy the stated functional and non-functional requirements. This approach provided the possibility to combine the main concepts of traditional methodologies and modify them to create a dependable and secure data transfer environment. The suggested method enables designers to create a secure and dependable data transmission environment in microcontroller-based devices without enlisting security experts. This is made feasible by the automatic analysis of data transmission protocols and the formulation of suggestions for enhancing their functionality to satisfy the needs. The calibre of the used expert knowledge base directly affects the range of alternative options. This implies that the accuracy and applicability of this knowledge base directly influence the quality of the solutions that the technique offers. As a result, this strategy cannot fully replace expert knowledge. By incorporating dynamic addressing, integrity control, message transmission with a size limit of 160 bytes, encryption, and mutual device authentication into the suggested approach, the intended I2C protocol's capability was enhanced above its simple implementation for Arduino devices. These modifications result in a 20 percent and a 60 percent drop in the planned I2C protocol's data transmission speed and message payload, respectively. An specialist in the security of microcontroller-based systems, who is familiar with the best and most highly specialised solutions now available, will typically come up with more effective methods to create a dependable and secure data transfer environment than the technique indicated. The recommended method, on the other hand, might be helpful to the professional as a tool for automating a portion of normal chores as well as a source of answers that are different from their subjective opinion. Additionally, the created solution is a component of the DLSEDS design approach for secure microcontroller-based systems.

In the paper [3], author Ekta Patnaik, has focussed on the design of the I2C transport controller and the interface between the two coordinated devices, a microcontroller and an EEPROM, acting as a master controller and a slave for serial communication in a built-in system. Just two bidirectional wires and a common protocol are used as the I2C transport controller's component elements to transmit data between two coordinated circuits and devices. Serial clock line comes in second after the serial information line. Philips Semiconductors created the I2C standard to enable faster devices to communicate with slower devices and with one another without information loss. The entire module is described in VHDL, and Model-SIM simulates it.

The main element of a computer's serial communications subsystem is the UART controller. Bytes of data are taken and transmitted by the UART as sequential bits. SPI is a popular technology used nowadays for peripheral device communication when we need to transport data quickly and under real-time limitations. For counters triggered by surface detectors, a subterranean data gathering system has been developed. It communicates with a dedicated processor using the UART and SPI protocols. M.Poorani et. al [4], in their study, have implemented the entire system using Xilinx 12.4i and SPARTAN 3E FPGA. The SPI protocol is checked using a 12 bit DAC MCP4922, while the UART protocol is checked using a hyper terminal. The suggested approach can efficiently supply both protocols for wireless serial communication. These serial protocols are mostly utilised by wireless zigbee technology.

The commonly used serial data transmission technology UART supports full duplex connection. Depending on the application a designer needs UART for, there are a variety of methods it can be implemented. Some UARTs have FIFOs for the receiver or transmitter to use as a data buffer, while others operate in 9-bit mode, and so on. In this paper[5], Ashwini Dhanadravye and Samrat S Thorat have reviewed many methods that were used in conjunction with UART for secure data delivery. The receiver module, transmitter module, and baud rate generator are the three primary kernel modules in the UART design, and they all play a significant part in the serial communication between the UART and host CPU. The many techniques researchers have used to achieve serial communication through UART modules are presented in this study. Since Verilog Hardware Description Language makes the design implementation easier to read and comprehend and can be integrated into the FPGA to deliver compact, stable, and reliable data transfer, the majority of researchers have utilised it to create the UART modules. Thus, the use of UART can be investigated utilising various algorithms and techniques to better and more effectively meet the communication demands. Particularly in the area of embedded systems, where SOC technology has recently gained prominence, this architecture has considerable significance.

In the paper [6], authors Poonam R Kedia and N. N. Mandaogade have demonstrated the controller's structure as well as an approach to designing an asynchronous FIFO. This controller is built with a FIFO and UART circuit block on an FPGA to quickly and efficiently integrate communication in contemporary complicated systems. Here, the UART basic operations are implemented using VHDL and integrated into an FPGA chip to achieve stable, dependable, and compact data transfer. They have concentrated on baud rate generation at various frequencies and check the received data for errors in the result and simulation section. The UART architecture includes the Baud Rate Generator. According to the needs, this frequency divider will automatically adapt. A multi-channel UART controller can also be created based on FIFO technology, FPGA, and it is also significant for the design of SOC to meet the communication requirements of contemporary complicated control systems.

Regu et al. [7] has focused on the interface between the master bus protocol and the slave using the I2C (inter-integrated circuit) protocol. Here, the microcontroller is connected to a DS1307. The DS1307 receives 8 bit data via the I2C bus protocol from the microcontroller. I2C is sized and powered for efficiency. Any high speed or low speed device, as well as any low speed or high speed device, can use this principle. This module serves as a master for the microcontroller device, which can be viewed as a slave, while serving as a slave for the DS1307. It can connect low-speed peripherals such as motherboards, embedded systems, mobile phones, set-top boxes, DVD players, PDAs, and other electronic devices.

Amandeep Kaur et. al [8] described a design of eUART (enhanced UART). The advantages of the eUART when used for real-time communication are demonstrated in this paper, along with the approach's capabilities and limitations. In order to communicate with a TTP/A (Time-Triggered Protocol Class A) or LIN (Local Interconnect Network) network, this SPEAR processor's expansion module was developed. It is economically advantageous to construct even low-cost devices with a distributed microcontroller-based control system because of technical advances made by the silicon industry. A field bus network connects multiple sensors and actuators in order to obtain the information from the environment. Real-time functions are frequently required since such field-bus-based systems are employed for controlling purposes. Two field-bus protocols that meet the requirements are TTP/A and LIN. Both protocols strive to establish reliable communication while also reducing costs by utilising commercial-off-the-shelf (COTS) hardware, such as common UARTs. It is preferable to combine all components onto a single silicon die in order to reduce the size and cost of a network node. Since quartz crystals can't yet be incorporated on silicon dies, the clock source must either be used or put next to the chip, which increases size and cost. The low clock frequency prevents the UART's transmission rate from being adjusted, which causes an arithmetic error. Another issue is caused by the RC-high oscillator's frequency drift, which has a direct impact on the UART's baud rate. In reality, it was determined that typical components cannot function under these circumstances by analysing the effects of an imprecise oscillator on a TDMA-based network. The analysis' findings led to the development of the eUART prototype. The eUART can function even in the worst-case scenario by lowering the arithmetic error in baud rate setting and eliminating the transmit jitter. Thus, the message sender's synchronisation pattern enables the eUART to automatically synchronise its baud rate. The eUART module also has a filter state machine and offers I6 fold oversampling in order to increase the communication's reliability. It is possible to configure the sample interpretation to work best for defect detection or availability.

A way to construct UART communications based on programmable logic devices is suggested in this study [9] by Yongcheng Wang and Kefei Song, in order to link devices using asynchronous communications protocol to DSPs with synchronous serial ports. The suggested method uses VHDL to incorporate the basic UART capabilities into the CPLD. After analysing a few ways to actualize UART, the UART data frame format and operational principle were first introduced. In-depth illustrations of the use of VHDL to implement UART transmitter, UART receiver, and baud rate generator were provided. Then the VHDL programme was pre-simulated and synthesised. Finally, the physical system underwent the bit error rate test. According to experimental findings, UART utilises 75% of the GLB, and the bit error rate is less than 10-9. The RS-422 protocol was used to carry out the experiment, and the baud rate is 62.5 kb/s. The suggested approach can meet the system needs for high integration, stabilisation, low bit error rate, potent anti-jamming, and low cost.

I2C and SPI are the most widely used serial protocols for both intra- and inter-chip low/medium bandwidth data transfers. Using numerous new Xilinx FPGA families, A.K. Oudjida et al [10] has analysed and compared the logical and physical aspects of the two protocols, emphasising which elements of the protocols are considerably increasing area overhead. The utilisation of this knowledge helps architects make well-considered and precise decisions. I2C and SPI are both implemented as general-purpose IP alternatives for a comprehensive comparison research, according to a recent market analysis of a considerable variety of commercial I2C and SPI devices. These systems offer every feature needed by contemporary ASIC/SoC applications. From the study, the authors found out that the I2C over SPI area overhead caused by the RTL code is about 25%, and the latency for the two systems are nearly the same. Although this comparison is only applicable to the slave side of the protocol, it is simple to make reasonable predictions for the master side: roughly the same latency with a larger area overhead to accommodate the I2C protocol's multi-master capability. If a straightforward counter-based baud rate is implemented, there won't be any appreciable space overhead for SPI-Master compared to SPI-Slave. If a digital-frequency synthesiser is utilised instead, this won't be the case. The results of a modern FPGA implementation of the slave side of the two common protocols I2C/SPI are also presented in the paper.

Fieldbus applications that are still commercially viable increasingly call for real-time communication. The fieldbuses LIN and TTP/A provide a temporally deterministic communication protocol that uses a standard UART (Universal Asynchronous Receiver/Transmitter) as a communication interface in order to take this requirement into consideration. Because UARTs were not initially intended for this kind of application, issues can occur that increase the amount of software and processing required or lower the bandwidth that can be achieved. Martin Delvai and Ulrike Eisenmann et. al[11], in their study, have shown a UART module that has been used for real-time applications. The described real-time communication protocols call for periodic synchronisation, which is made possible by a synchronisation mechanism that also lessens the complexity of the software. The UART can also initiate activities in response to happening events. In this approach, the UART module may handle a sizable portion of the communication process on its own. Both send jitter and arithmetic error associated with baud rate setting have been totally eliminated. As a result, the UART module can operate with erratic clock sources that have a high rate of drift, such as low-cost RC oscillators.

Junwei Zhou, Andrew Mason, et al. [12] have developed a new sensor bus for low power micro software systems, evaluated sensor network architectures, and analysed current digital communication buses that are often used in sensor networks. In order to minimise hardware overhead at the sensor node, the new intra-module multi-element microsystem (IM^2) bus is a nine-line interface with 8b serial data that offers a number of cutting-edge features like plug-and-play and power management. The discussion has ended with a few wireless sensor networking-related topics. The discussion of these topics offers recommendations for selecting the best bus for various sensor network applications. IM^2 preserves the benefits of both IEEE 1451.2 TII and I2C bus, whereas IM^2 is designed with short wiring distances between the microsystem module controller and the sensor nodes for low power micro system applications.

4. Applications of Bus Protocols

A high speed protocol such as I3C is used in OIS or Optical Image Stabilisation System. OIS is used in today's smartphones to improve mobile photography by physically adjusting the camera sensor or lens module to counterbalance any little camera movement to prevent blurry shots. This system requires communication with the processor in very high frequency to send the data completely and thus I3C is used in this system. NFC is a collection of short-range wireless technologies that normally require a connection to be established at a distance of 4 cm or less. Between an Android-powered smartphone and an NFC tag or between two Android-powered devices, NFC enables the sharing of tiny data payloads. These NFC applications use SPI or I2C to communicate with processors based on the specific application of the NFC.

Nowadays, smartphones include a magnetometer that allows us to sense the direction/orientation in space. Some simple applications such as the Compass App can be used to locate oneself in relation to Magnetic North. Magnetometer uses I3C or I2C to communicate with the processor. Sensors such as IMU accelerometers, pressure sensors, humidity sensors and some touch sensors use I2C as the communication protocol in the current smartphones. 2-wire and 4-wire UARTs are widely used by the low power bluetooth module. GPS receivers and GPRS modems also use UART for communication purposes. In-display fingerprint sensors, mounted under the display of the smartphones, use SPI as their communication protocol. SPI is also used by some 3rd party camera applications, Laser Range Finders and flicker detection modules. These are some of the important applications of various bus protocols in current smartphone applications.

5. Conclusion

Various on-chip protocols are reviewed, along with their features, architectures or applications, in this paper. There is a descriptive comparison of different on-chip protocols. UART is an effective hardware communication protocol that uses asynchronous serial communication with configurable speed. I2C Protocol is a serial communication method that allows various devices to exchange data with one another. For data transmission and reception between masters and slaves, it is a half-duplex, bi-directional and two-wire bus system. Serial peripheral interfaces are widely used in microcontrollers and other devices such as sensors, ADCs, shift registers and others. It is a synchronous mode of communications where the devices are synchronised on the rising and falling edge of the clock. These bus protocols are widely used in SoC design, Sensor networking, and all these protocols can be implemented in FPGA using VHDL and C language.

REFERENCES

[1] Lyke, J., Mee, J.K., Bruhn, F., Chosson, G., Lindegren, R., Lofgren, H., Schulte, J., Cannon, S.R., Christensen, J., Hansen, B., Vick, R., Vera, A., & Calixte-Rosengren, J. “A Plug-and-play Approach Based on the I2C Standard”. 24 th Annual AIAA/USU Conference on Small Satellites, SSC10-VII-7, 2019

[2] D. Levshun, A. Chechulin and I. Kotenko, "A technique for design of secure data transfer environment: Application for I2C protocol," 2018 IEEE Industrial Cyber-Physical Systems (ICPS), 2018, pp. 789-794,

[3] Ekta Patnaik, Sachin Bandewar, “Design and Realization for Interfacing Two Integrated Devices Using I2C Bus- a Review'', International Research Journal of Technology And Applied Science, 05 May 2017, pp. 186-195, doi: 10.17005./ IRJTAS.2017. 0809001

[4] M. Poorani and R. Kurunjimalar, "Design implementation of UART and SPI in single FGPA," 2016 10th International Conference on Intelligent Systems and Control (ISCO), 03 November 2016, pp. 1-5, doi: 10.1109/ISCO.2016.7726983.

[5] Ashwini D Dhanadravye and Samrat S Thorat. “A Review on Implementation of UART using Different Techniques”. International Journal for Computer Science and Information Technology, 10 June 2014, pp. 394–396.

[6] B. Eswari, N. Ponmagal, K. Preethi and S. G. Sreejesh, "Implementation of I2C master bus controller on FPGA," International Conference on Communication and Signal Processing, 2013, pp. 1113-1116, doi: 10.1109/iccsp.2013.6577229.

[7] Amanpreet Kaur and Amandeep Kaur. “An approach for designing a universal asynchronous receiver transmitter (UART)”. International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol. 2, Issue 3, May-Jun 2012, pp.2305-2311

[8] Y. Wang and K. Song, "A new approach to realize UART," Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, 2011, pp. 2749-2752, doi: 10.1109/EMEIT.2011.6023602.

[10] A.K. Oudjida, M.L. Berrandjia, R. Tiar, A. Liacha, K. Tahraoui, “FPGA implementation of I2C & SPI protocols: A Comparative Study”, 16th IEEE International Conference on Electronics, Circuits and Systems - (ICECS 2009), IEEE. 2009, pp. 507–510.

[11] Martin Delvai, Ulrike Eisenmann, and Wilfried Elmenreich. “Intelligent UART module for real-time applications”. Proceedings of the First Workshop on Intelligent Solutions in Embedded Systems,27 June 2009, pp. 177–185.

[12] Junwei Zhou and Andrew Mason. “Communication buses and protocols for sensor networks”. Sensors 2.7 (2002), doi:10.3390/s20700244, pp. 244–257.