ACKNOWLEDGEMENT

I would like to thank all the people who had contributed in some way to the work in this thesis. First and foremost, I would like to thank my supervisor, Mr. Zulfiqar Ali, for accepting my title and guiding me all the way. He supported my work and gave me plenty of guidance when I hit a road block. Additionally, I would like to thank my parents and my sisters for the constant moral support which encouraged me during hard times. Lastly, I would like to thank my fellow friends whom had put in the same amount of effort to complete our dissertations together.

Table of Contents

List of Figures

List of Table

PENAMBAIKAN REKA BENTUK LALUAN DATA PEMANCAR PENGAWAL JALUR TAPAK BLUETOOTH

ABSTRAK

Pengawal Bluetooth diletakkan dalam lapisan fizikal timbunan Protokol Bluetooth untuk menguruskan semua saluran fizikal dan pautan seperti pembetulan ralat, pemilihan hop, keselamatan dan pemutihan data. Jalur asas menguruskan pautan segerak dan tak segerak, mengendalikan paket dan melakukan siasatan bagi peranti Bluetooth di kawasan itu. Salah satu cabaran yang dihadapi oleh peranti Bluetooth adalah antara operasi peranti yang disepadukan ke dalam mana-mana peranti lain. Pengoptimuman prestasi diperlukan tetapi ia adalah pengimbangan dengan penggunaan kawasan dan kuasa peranti. Lebih besar reka bentuk, lebih kuasa perlu digunakan. Kesesakan litar pautan perlu dipertimbangkan juga untuk kawasan yang lebih kecil. Dalam tesis ini, objektif adalah untuk mereka bentuk penghantar di laluan data bagi pengawal jalur asas Bluetooth. Kemudian, proses sintesis akan disimulasikan oleh Synopsys dalam usaha untuk menjana netlist. Netlist ini kemudiannya akan diterjemahkan ke dalam pelaksanaan fizikal logik dan susun atur yang terbentuk. Proses pengoptimuman bermula sekali lagi dari kod VHDL untuk proses susun atur. Keputusan disintesis yang terdahulu dibandingkan dengan keputusan daripada IC Pengkompil. Ia menunjukkan bahawa reka bentuk dioptimumkan mempunyai ruang dan penggunaan kuasa yang lebih besar iaitu 75023.627147 micron persegi dan 18.2595 mW tetapi pemasaan telah ditambah baik dari 4 ps kepada 390 ps. Transmitter ini dapat beroperasi pada 200 MHz dan 1.62 V. Oleh itu, kawasan dan kuasa akan bertambah jikalau pengoptimuman prestasi masa dilakukan. Fokus projek ini adalah mengenai prestasi reka bentuk.

DESIGN OPTIMIZATION OF DATAPATH TRANSMITTER IN BLUETOOTH BASEBAND CONTROLLER

ABSTRACT

A Bluetooth baseband controller is placed in the physical layer of the Bluetooth Protocol stack to manage all the physical channels and links like error correction, hop selection, security and data whitening. The baseband handles the packets and does the inquiry for the Bluetooth devices in the area. The optimization of the performance is needed but it is of a trade off with the area and power consumption of the device. The bigger the design, the more the power being consumed. In this thesis, the objective is to optimize the design of the transmitter in the datapath of the Bluetooth baseband controller. It is also part of the objective to improve the RC delay of the worst path timing. The inherited codes need to be verified with a test bench on Model Sim first. Then, a synthesis process is being done using the Synopsys tool in order to generate a netlist. The netlist is then being translated into physical implementation of the logic and the layout is formed. Then, the optimization process starts again from the VHDL code to the layout process. The synthesized results are first being compared with the results from the IC Compiler. The results of the synthesized results before and after optimization is being compared as well. It is shown that the optimized design has a larger area and power consumption of 75023.627147 square micron and 18.2595 mW but the timing in the worst path is significantly improved from 4 ps to 390 ps. The transmitter is able to operate at 200 MHz from the constraint set and the operating voltage is at 1.62 V. Thus, a tradeoff with the area and power consumption is in place if optimization on the timing performance is done. The focus of this project is on the performance of the design.

CHAPTER 1

INTRODUCTION

1.1 Background

A single router can be used to connect several computers at once to form a small wired network. Large networks would involve numerous routers which allows them to switch among one another and transmit the data further. A wireless network denotes to the usage of radio frequency or infrared signals to transmit data and communicate with one another. Among the newer standards today for short range data transmission is the Bluetooth, Hiperlan, 802.11 and infrared [1]. These standards are creating an extensive range of enabling the connectivity of devices from the enterprises to home networking. Various characteristics had been shown in [1] and there are many pros and cons of the wired and wireless networks. The wireless network that is being focused on in this thesis is the Bluetooth network.

In the modern world, Bluetooth is a commonly found function in the smart devices sold on the markets. For commercial users, it is a useful function in the device as it allows for transfer of data without any external wires, cables or connectors. It is an inexpensive and a low power consumption chip which is miniscule enough to fit in any electronic device. It is a radio interface which is universal in the frequency band of 2.45 GHz that allows the mobile devices to connect to one another through short-range networks [2]. Not only that, each device is able to communicate simultaneously with up to seven other devices and the devices may belong to other piconets as well. A piconet is a network formed by using the Bluetooth technology. Piconets may be combined together to form a scatternet [3]. This technology wandered into the world when Ericsson Mobile

Communications started a research to investigate the possibility of having a low-powered and low-cost radio interface in order for the mobile phones to interact with one another. The clear objective was to eradicate the need of connectors between mobile phones, wireless headsets and so on [2].

There are existing devices which utilizes infrared links to connect to one another. Infrared chips are inexpensive but they still have very limited range, sensitive to direction and can only be used among two devices. As compared to the Bluetooth chip, which not only has a greater range, but can transmit through many types of material and can be connected to multiple devices at the same time. The major encounters faced by the Bluetooth technology would be security solutions which needs to be robust, quality of service, vendor independence, and the interoperability of the application.

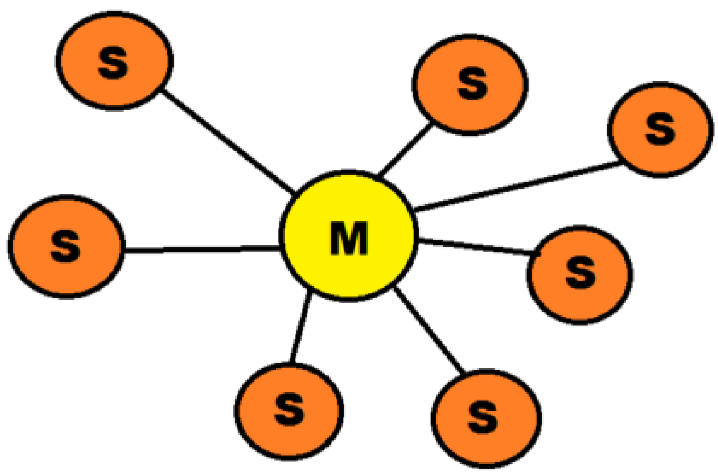

Figure 1.1: Example of a master-slave relationship of the Bluetooth devices

Therefore, this thesis looks to research into producing an optimized design for the transmitter of the Bluetooth baseband controller. There are not many papers specifically on Bluetooth baseband controller design as most of the circuits and layout designs are confidential and cannot be disclosed. The layout design will be generated from digital codes provided from a third party source. The design consist of two parts which are the transmitter and receiver. Both IPs are not linked or connected together. In this thesis, the focus would be on the optimization of the layout design of the transmitter. The main point to prove from this project is to optimize the design as much as possible and solve the challenges encountered.

1.2 Problem Statement

In IC design, it is critical to have a compact, feasible and self-sustained design. The design need to be able to be integrated into various products without the restriction of the product brand and architecture. A plug and play design is desired in any layout design for an application-specific integrated circuits (ASICs). In the Bluetooth technology, it is important for the baseband controller design to be as robust and efficient as possible because it is the layer that enables the primary and secondary to communicate with one another utilizing time slots.

Another challenge faced in the Bluetooth technology is the quality of the service. The design of the circuits in the Bluetooth device need to be able to transmit and receive the data coming in and out of the device. The data integrity needs to be ensured at all times. Security of the data transmitted and received need to be guaranteed as well. As mentioned by T. Panse, there are three main security services which is confidentiality, authentication and authorization [4]. These services are important to prevent eavesdroppers from reading certain acute information, verifying the identity of the communicating devices and to regulate admission to resources [4].

In the inherited design, there are many improvements that can be done. There were no model test bench to test the functionality of the baseband controller. Other than that, there were no input ports to enable the transmitter to communicate with the controller. That makes it impossible for the transmitter to receive any control signals on the packet processing and also impossible to transmit the data to the receiver. Also, the codes were found not to be synthesizable and realized into a hardware form. There are many improvements that can be done before realizing the transmitter into hardware.

1.3 Objectives

The objectives of the research project are as follows:

i. To optimize the layout design of the transmitter designed so as to realize the design in hardware.

ii. To achieve optimum performance of the transmitter by reducing the total RC delay in the worst timing path.

1.4 Scope

This thesis covers the layout design of the transmitter from the Bluetooth baseband controller. The design of the receiver will not be included. The test bench created to enable the simulation of the transmitter will be covered. The synthesis results will be published in the thesis. Optimization works done on the digital design will also be covered while the design of the Bluetooth baseband controller in VHDL will be explained in details. Focus of this project is to optimize the design as much as possible and document the challenges faced. This thesis will not include the development of the VHDL codes for the transmitter of the baseband controller and any optimization works done on the digital codes will be discussed as well. The scope of this thesis will not include any Design Rule Checks on the layout design after the layout is being generated on IC Compiler as it is not practical to fix the DRC errors because this is just one module in the datapath of the Bluetooth baseband controller.

1.5 Thesis Outline

Chapter 1 wraps up the introduction, problem statements, objectives and scope of the thesis. In Chapter 2, the in depth on Bluetooth Protocol, Bluetooth baseband controller and the works done pre-cursor to this work will be looked into. The VHDL code obtained by a third party source would be discussed and criticized as well. In Chapter 3, the methodology of the project would be discussed and any tools used would be specified as well. As in Chapter 4, the results and discussions will be published, explained and analyzed. With the contents of this chapter, a conclusion will be formed and that will be summed up in Chapter 5. Future works that can be done and improved on would be discussed here as well. All of the reports generated would be attached in the appendix of this thesis.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses the Bluetooth Protocol Stack and the various types of Bluetooth baseband controller in the market. It reviews the specifications of the baseband controller and looks into the gaps that the product might be facing. Proposed designs will be reviewed as well. Bluetooth is a technology specifically targeted to be a short-range links between any portable devices, low cost and low powered. It operates in 2.4 GHz ISM band (2400 MHz – 2483.5 MHz). This radio bandwidth is free to be used by any radio transmitter if it complies with the regulations. The focus of this application is mainly on travelers. Portable devices that are able to be connected without any external wires are desired. Therefore, there is no hassle in having to carry around the extra weight. This solution has become so popular that it is now in most of the mobile phones in the market.

However, when designing such an intricate device, there are several problems that needs to be solved. The built system must be able to operate on any other device. This connection must be able to support any voice or data such as pictures and videos. That is to be able to transfer within the air. This section also reviews the functions of the base band layer and understanding the structure. The radio transmitter and receiver must be small in size and able to operate at a low power. Low power alternatives will be researched upon to make sure the design is optimized and able to function on any other portable devices that it will be connected to. It is also important that the designed device is small enough to fit into any portable device such as headphones, mobile phones, speakers and more. All of these specifications will be reviewed and looked upon on.

2.2 Bluetooth Protocol Stack

The Bluetooth technology is specifically designed to be the solution for a short-range connectivity such as PDAs, mobile phones and any other electronic devices. This invention has revolutionize the way we perceive data now. This technology differs from the WAN or LAN technology which enable the devices to connect via an infrastructure-based services which is either through a corporate backbone or a provider for the wireless carrier. These services are used to cater for long-ranged connectivity which is different as compared to the Bluetooth technology. There are no formal standard documentations but rather just an implementation manual is being used to communicate the Bluetooth specifications. The documentations can be easier to read since it is written based on the experience of a group of engineers during implementations but the downside would be that there are conflicts between the interpretations of the engineers.

The Bluetooth Special Interest Group developed the Bluetooth protocol stack to govern the developing interactive services over the interoperable radio modules. This protocol must be obeyed tightly by the companies which would like to manufacture both the software and hardware of the Bluetooth devices. This protocol is in place to ensure that all of the devices manufactured is interoperable within various other devices from other manufacturers. The protocol stack is made up of multiple layers and there are some layers which will pass through several layers of the stack. There are two parts of the Bluetooth devices which is a host enforcing the upper layers of the stack while a module is implementing the bottom layers of the stack. In some cases, they can be denoted as the transport and middleware protocols. An application may run a single or all of the vertical slices from the stack [5]. A vertical slice means that the application runs from top of the protocol to the bottom layer of the protocol. This sequencing is necessary for the protocol stack. In the transport protocols, it comprises of protocols which are exclusively developed for Bluetooth technology. These are the protocols involved in any data communication between the devices [6].

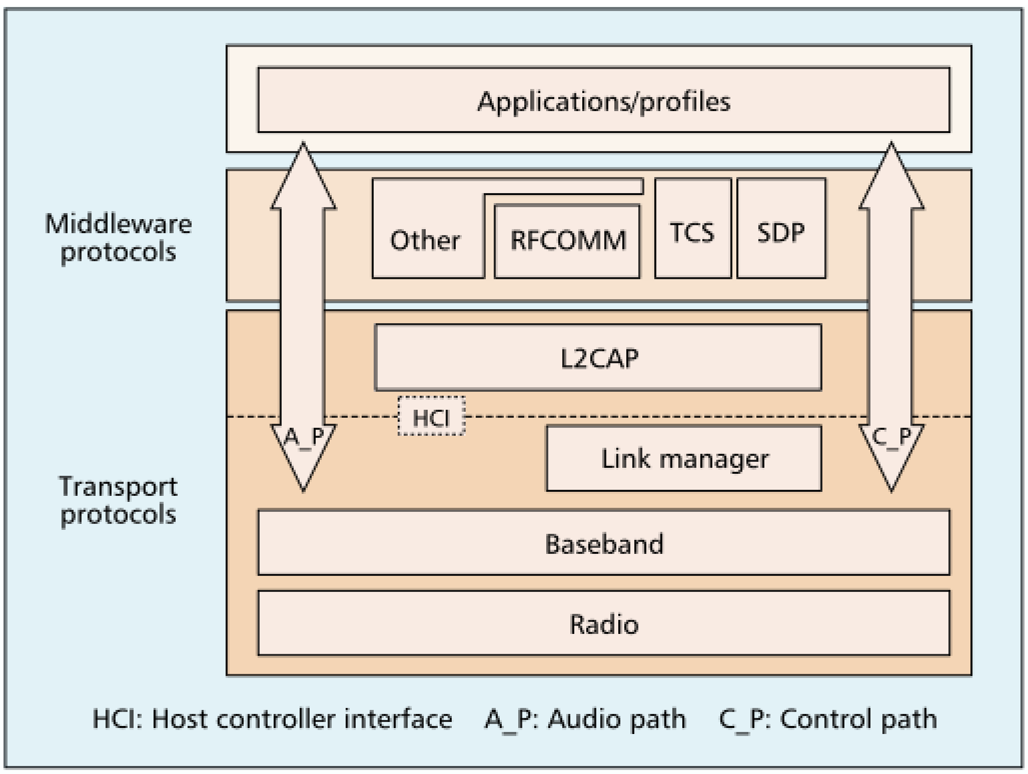

Figure 2.1: Bluetooth Protocol Stack [6].

From Figure 2.1, the middleware protocols are comprising of both the Bluetooth specific protocols and other protocols as well. These protocols are selectively used based on the applications. The separation of these layers have an advantage to it which it allows for hosts with spare capacities to store the higher layers while permitting the Bluetooth device to have a smaller processor and memory which works on a low power mode for cost reduction in manufacturing the chip. The interface which allows the upper and lower layers of the protocol to communicate is the Host Controller Interface (HCI). The radio in the transport layer defines technical features of the radios. The Bluetooth radio runs on the 2.4 GHz ISM band which is compliant to the 15 regulations for intentional radiators in the band [6]. A binary Gaussian frequency shift-keying (GFSK) is the modulation technique used. There are three power of classes for the Bluetooth radios. It is depended on the transmit power of the radio. The classes that are widely used in mobile devices are from class 3 and class 2 with the transmit power of 1 mW and 2.5 mW respectively due to the constraints of cost and power [6]. The baseband layer will be discussed in the subsequent sub chapter.

The main principle in designing the whole protocol stack is to maximize the usage of existing protocols for multiple purposes in the higher layers without having to reinvent the layer. The Link 2 Manager Protocol (LMP) utilizes the links set by the baseband layer between the connecting devices to develop a logical connection. This layer holds the security information and device authentication. Next, is the Logical Link Control and Adaptation Protocol (L2CAP) which is responsible to receive data from the upper layers and having it translated to the Bluetooth format. This allows the data to be communicated to the higher layer of the protocol. Radio frequency Communication Protocol (RFCOMM) allows serial connections emulation over the baseband layer to allow transport abilities for the upper layers and avoids direct interface with L2CAP [4]. The Service Discovery Protocol (SDP) is used to explore services and provides the basis for every available usage models. Telephony Control and Signaling (TCS) protocol layer sets the call control signals for the formation of speech and data calls. The TCS signals are carried over the L2CAP [4]. The application layer contains the applications from the users. These applications communicate with RFCOMM protocol layer to form a serial connection. The Bluetooth protocol stack can be introduced on any device which has a programmable device such as FPGA and microcontroller. Implementations of the protocol had successfully been done by Rocher and Hancke on a low cost microcontroller [7]. An open source software was used to reduce the development costs.

2.3 Network Topology of Bluetooth

2.3.1 Piconets

Bluetooth is a short range with low-powered abilities in wireless communication. The attractiveness of this device is that is it able to form ad-hoc networks on typical mobile devices. With that being said, technical hurdles that need to be overcomed by the Bluetooth technology are many. The most basic one faced by the Bluetooth industry is how the nodes are being organized in a completely operational network while fulfilling all the constraints or protocols being presented by Bluetooth. In this sub chapter, the basic Bluetooth technology is discussed. Small groups of Bluetooth nodes are called the piconets. Within the piconet, there is a master unit which is the unit that establishes the connection and the multiple slaves which are the remaining units being connected to. Bear in mind that a node can belong to numerous piconets and these nodes are known as the "bridge". A piconet can only have 8 active members at one time [8]. The slaves are not able to communicate with each other directly but they are only able to communicate through the master node which acts as a transit node.

Meanwhile, communication with other piconets are strictly relying on the bridge nodes. The bridge node is not able to connect to several piconets simultaneously. There are different activity states allowable to the nodes such as active, idle, sniffing and parked [8]. The data exchange can only happen when both the nodes are in an active state. The higher the number of piconets that the node belongs to, the worst the connectivity that it can provide to them. This is because the bridge node can only be active in a piconet at one time. A master unit will not be able to take up to role of the bridge unit because in the time that it is active in another piconet, acting as a slave, it will be unable to sustain a connection to its own slaves belonging to its piconet. Also, it is desirable for the bridge node to be connected with smaller number of piconets to preserve the connectivity established. Figure 2.2 shows the illustration of the bridge slave between two piconets. Slaves can be locked in parked states too. They are not active but remains synchronized to the master [9].

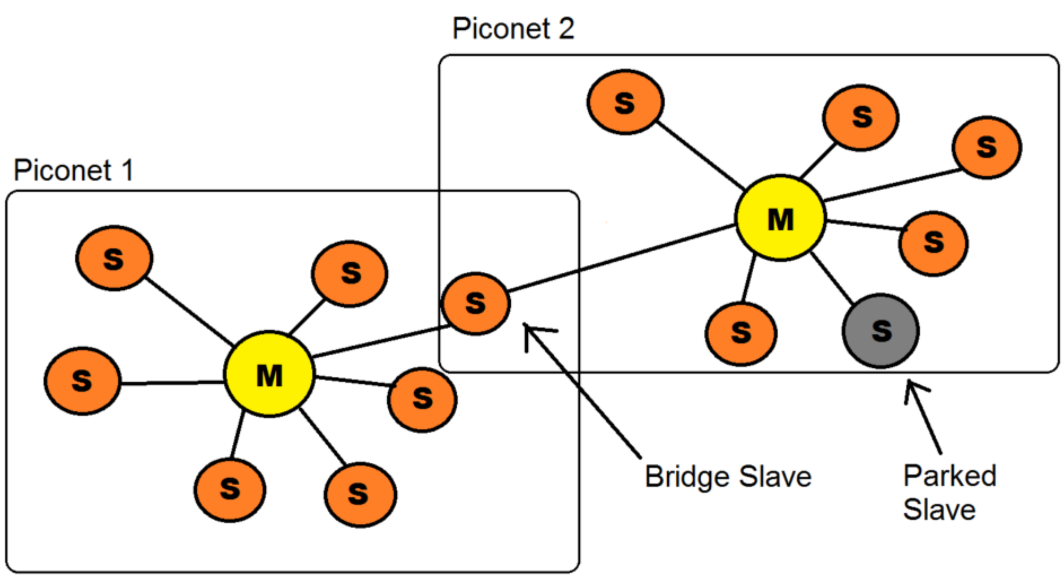

Figure 2.2: Illustration of a bridge slave between two piconets or a scatternet

2.3.2 Scatternets

A group of piconets overlapping each other's coverage area will form a scatternet. The hopping sequence and phase of each piconet is independent and determined by the master [9]. A unit can be in dissimilar piconets but in the same scatternet. The unit can act as a master or a slave in each piconet that it is in but not as a master in both of them. The access of the channel is independent in respective piconet. There will be a limitation of utilization of the time slots when a unit participates in multiple piconets. This is due to the device being unable to transmit data in similar space in two different piconets [9]. A collision occurs when two piconets hop at the equivalent frequency. The probability of collision increases when there are more piconets existing in the same place. The routing in scatternet and formation of the scatternet is not being specified in the Bluetooth specifications. Therefore, there is no proper documentation for the protocol of traffic coordination of a scatternet as well.

There are many algorithms being presented in forming the scatternets. Salonidis et al. presented a scheme of symmetric formation of the link where configuration of master and slave can be ignored [10]. Every node can establish links with any other nodes with alternating between INQUIRY and INQUIRY SCAN continuously. An election process is being used to choose a leader which arranges the scatternet topology. A paper submitted by Godfrey et al proposed a formation algorithm called Tree Scatternet Formation [11]. This formation assigns the roles of master and slaves to the nodes while ensuring the connection of a tree between all the nodes [11]. The scheme simplifies the routing of the packets and scheduling. There are many more up and coming works that is being looked upon in this area.

2.4 Bluetooth Baseband Controller

The baseband layer in the Bluetooth stack protocol is the layer responsible to establish the connections within the piconet, timing, addressing packet format and the power control [12]. In other words, it defines the critical procedures to allow the devices to interact with one another. When two or more Bluetooth devices that share a common channel, a piconet is formed. In order to regulate the traffic in the piconet, one of the units becomes the master. By definition, the unit that creates the connection becomes the master of the piconet. Only one master at a time in a piconet and any participating units can take over the master at any time of the connection. When a connection is formed, an offset is added to allow the slave clock to be in synch with the master clock as shown in Figure 2.3. The two basic categories of physical links that can be formed between a master and a slave is Synchronous Connection Oriented (SCO) and Asynchronous Connection-Less (ACL).

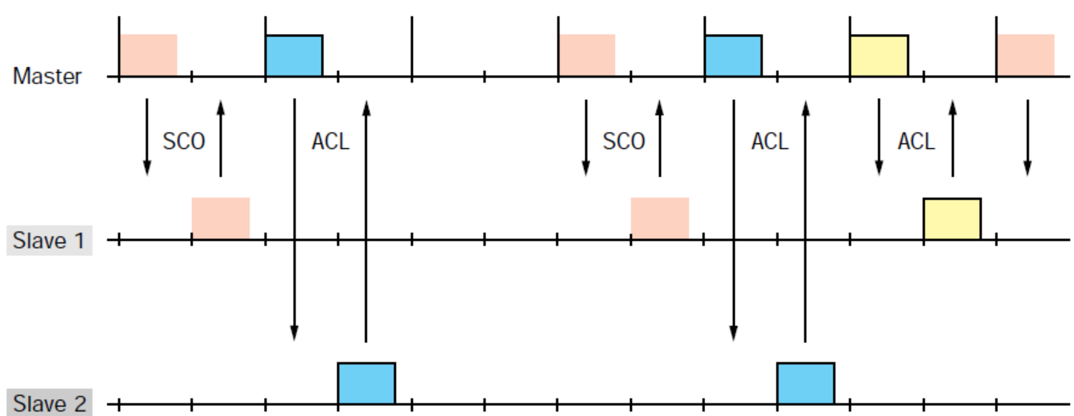

Figure 2.3: Synchronous Connection Oriented and Asynchronous Connection-Less links with one master and two slaves depicted from Ericsson Review[2]

The SCO link provides a regular periodic interchange of data in the arrangement of reserved slots which forms a symmetric link between the master and slaves. Therefore, a circuit-switched connection is being provided by the SCO link where data are being exchanged regularly and is intended for the use with time-bounded information such as audio [13]. A master of the piconet can form up to three different SCO links to the identical or distinguished slaves. This applies the same to the slave which is able to support up to three SCO links from one master. SCO links does not support retransmission of data but when a transmission error occurs, it is able to use FEC mechanisms to recuperate [6].

Unlike the SCO link, the ACL link provides a point-to-multipoint link between the master and all of its slaves in the piconet. Remaining slots that are no used on the channel by the SCO links will be occupied by the ACL links. A packet-switched connection is being provided by the ACL link whereby the data are being exchanged periodically [13]. It also ensures the integrity by retransmissions and sequencing of members. Forward Error Correction (FEC) is used when necessary [6]. The traffic in ACL links are being controlled by the master. A summary table for both the links are as shown in Table 2.1. In the work of Saif et al, the proposed architecture of the baseband consists of master and slave module connected to a 4x2 multiplexer and the transmission is controlled by a transceiver module [14]. The baseband layer function is dependent to the mode that it is in.

Table 2.1: Summary table of SCO and ACL [6], [13]

Synchronous Connection Oriented (SCO)	Asynchronous Connection-Less (ACL)
Symmetric link	Point-to-multipoint link
Circuit-switched connection	Packet-switched connection
Periodic exchange of data in reserved slots	Periodic exchange of data in remaining slots not used by SCO links
Utilized for audio only	Utilized for combination of audio and data
No retransmissions of audio	Have retransmissions of data

2.5 Bluetooth Packet

Each Bluetooth device would need to have a 48-bit address which is used for the formation of the access code. This allows the device to verify whether it is sending the data to the right destination. In the access code containing pseudo-random specifications, the identity of the piconet's master is included [13]. The packets that are exchanged will be identified by this master distinctiveness. This helps to disable the possibilities of the packets being received wrongly by devices in a different piconet occupying the same hopping frequency. Bluetooth packets have the similar format which starts with an access code, follows by a packet header and ends with the user payload as shown in Figure 2.4.

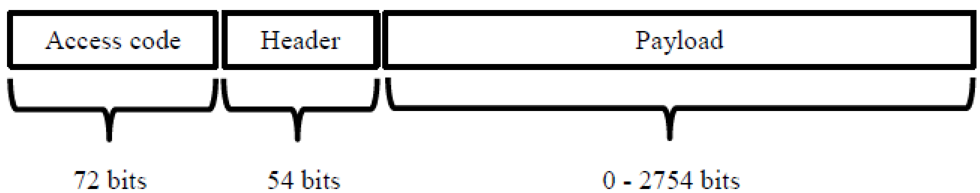

Figure 2.4: An example of the Bluetooth Packet Structure

Address of the packet to the exact device is embedded in the access code. Control information about the packet and links are being stored in the header. The actual message is in payload. When the packet is lost, only a small part of the information is lost because of the frequency hopping techniques applied which allows the packets to have high hopping rates and short packet lengths. All of these packets can be protected ahead by forward error control. Each of the data packets is secured by ARQ scheme whereby the lost data packets are retransmitted automatically [6]. The receiving end would check the received packets for any errors during the transmission and if errors are found in the message received, the indication would be in the header of the return packets. However, voice messages are never retransmitted. A robust voice-encoding scheme is implemented instead. Based on CVSD (continuous variable slope delta) modulation as shown in Figure 2.5, the scheme allows the audio waveform to be followed precisely and is very resilient to bit errors. Errors are observed as background noise which increases when the bit error increases.

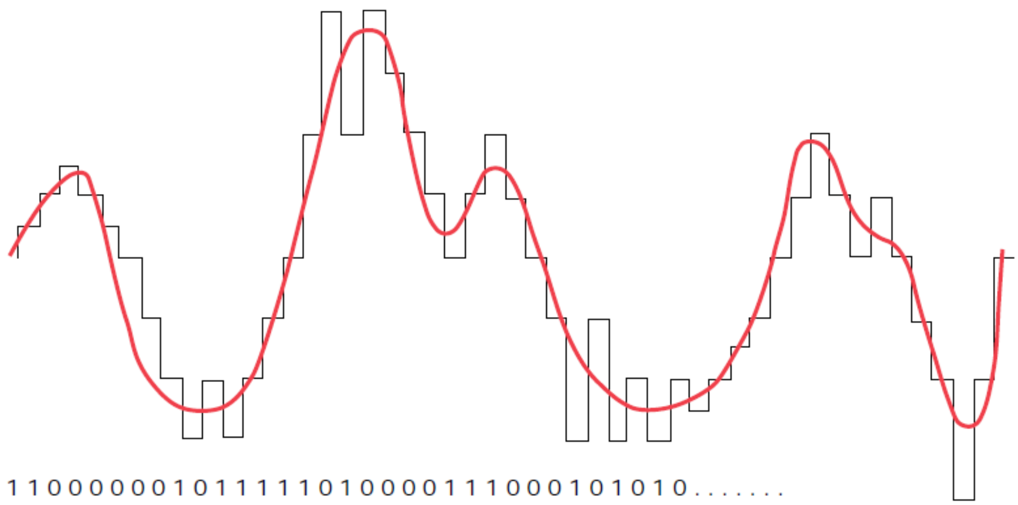

Figure 2.5: CSVD waveform.

Looking into the access code structure, it can be made up of 72 bits which comprises of 4 bits preamble, 64 bits of sync word and 4 remaining bits are the trailer. The preamble bits are dependent on the least significant bit of the sync word while the trailer bits are dependent on the most significant bit of the sync word [15]. There are three different types of sync word which is Channel Access Code, Device Access Code and Inquiry Access Code. The usage of which sync word is dependent on the type of access code. It could be different for the different types of devices that is sold on the market. The specific sync word is used to match the form of pattern in the correlator when the receiving device receives the packets. The incoming message is only accepted when the sync word matches the pattern in the correlator. Then, the process of recovering the desired message signal is started. Figure 2.6 shows the structure of the access code.

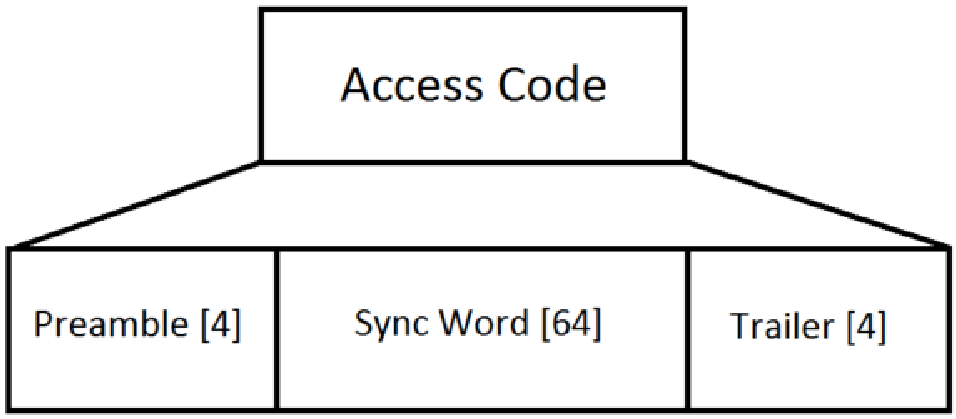

Figure 2.6: Structure of Access Code

The next structure in the Bluetooth packet is the header. The header contains 54 bits but it is only made up of 18 bits in actual [15]. The 18 bits are being duplicated thrice due to the FEC encoding process. It consists of 3 bits of address, 4 bits of packet type definition, 1 bit for flow control, 1 bit for acknowledgement indication, 1 bit for sequencing and 8 bits of Header Error Check (HEC). Functions of each bit are listed in Table 2.2 below.

Table 2.2: Functions of bits in header[15]

Bit Type	Num. of Bits	Function
Addressing	3	Active Member Address. Address of the slave to where the packet is directed to or received from.
Packet Types	4	ACL or SCO links
Flow Control	1	Flow control for ACL link: 0 is to stop while 1 is to go
Acknowledge Indication	1	Acknowledgement: 0 is NAK and 1 is ACK
Sequence Number	1	Bit is toggled for the following packets
Header Error Check	8	Integrity check value

Last but not least, the payload contains the actual message information. It does not have a fixed amount of bits. The number of bits can be any number from 0 to 2754. This depends on the device used and the type of data sent. 4 bytes (32 bits) of data is commonly used in the computing world. A Cyclic Redundancy Check (CRC) is being used as an approach to accompany the actual message information. The desired message would be constructed by unknown bits number and 16 bits of CRC. This is then followed by FEC encoding which is either $\frac{1}{3}$ or $\frac{2}{3}$ FEC encoding. In the $\frac{1}{3}$ FEC encoding, 1 bit of information is being duplicated three times as what was seen in the header portion. As for the $\frac{2}{3}$ FEC encoding, 5 parity bits are being calculated using the Hamming code to accompany every sequence of 10 bits. Trailing zeroes are important in this case to ensure that the multiple lengths of 10 bits is being fulfilled to carry out the $\frac{2}{3}$ FEC encoding. An example of an ACL link packets is showed in Figure 2.7.

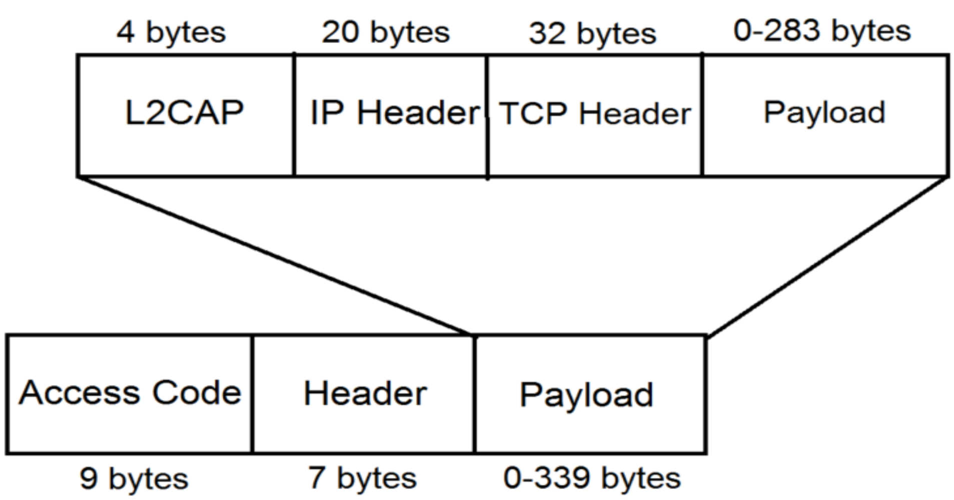

Figure 2.7: Example of the structure of the payload in an ACL link packet

Another format available for the Bluetooth packet is the Enhanced Data Rate (EDR) packets. The functions of the access code and header remains the same as the conventional Bluetooth packets. However, EDR protocol has additional packet types added to modulate a new scheme for the payload data. The change is that the EDR packet needs to insert a small guard band and a synch word which synchronize the sequence of the header and payload [16]. Figure 2.8 depicts the EDR Bluetooth Packet.

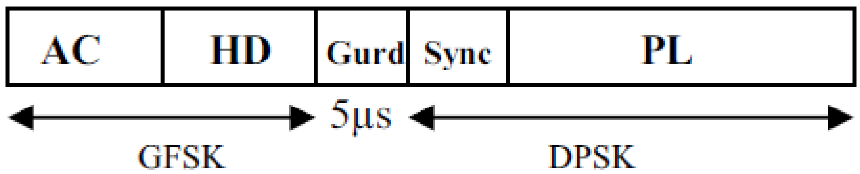

Figure 2.8: EDR Bluetooth Packet

In a paper by Moron et al, a model for serial port profile with the retransmissions of packets is being discussed. The mean of the Bit Error Rate is being calculated from the Bluetooth packet delay and the background noise [17]. The delay is being determined as a function for the chances of having to retransmit a package [17]. In another paper by Mohsen et al, the throughput was improved by using adaptive packets [18]. The proposed design of the packet is to employ Channel Quality Driven Data Rate (CQDDR) in selecting the transmitted packet size through Received Signal Strength Indicator (RSSI) depending on the channel conditions. New packets are added to grow the number of CQDDR choice[18]. Figure 2.9 shows the proposed design of adaptive Bluetooth packets by Mohsen et al.

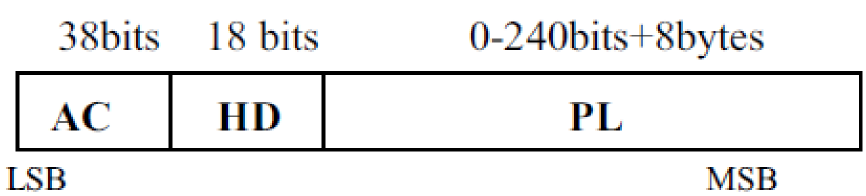

Figure 2.9: Proposed adaptive Bluetooth classic packets format by Mohsen et al [18]

2.6 Design of Bluetooth Baseband Controller

2.6.1 Basic Design of Bluetooth Baseband Controller

The Bluetooth baseband controller is designed in the digital language of Verilog Hardware Description Language (VHDL). It can then be integrated onto a Field Programmable Gate Array (FPGA). The baseband controller manages the baseband link layer in the Bluetooth protocol stack. Functions of the baseband link layer are related to the timing, framing of the packets, error detection with correction and flow control [23]. There are several blocks in the controller design such as the register files, a controller, a modem, a data path, a clock generator, a hop selector and interface block. The register file enables communication between the link manager processor and controller. Information such as local device information, current status information, remote device information, interrupt flags and the transmitted and received packet information will be stored in the register file [23]. Next is the controller which is subdivided into three parts, a timing controller, a link controller and a state controller. The channel is divided into slots of 625 microseconds in length and numbered according to the clock of the master. It takes care of the timing and changing of the states. The data path is the unit responsible to compose the packets to be sent out and decompose any packets it receives. The data path will be discussed in detail in the following sub chapter.

The modem consists of a modulator and a demodulator. A modulator is made up of a Gaussian Low Pass Filter (GLPF) and a symbol mapper. The access code is usually passed directly to GLPF so that timing synchronization can be done. A demodulator is made up of the correlator, symbol demapper and clock recovery [23]. The hop selector does the generation of hopping sequences. There are various hopping sequences such as page hopping, page response, inquiry sequence, inquiry response and channel hopping. The last block is the clock generator which is important for the clocks to be aligned to the functionality of the baseband controller. Table 2.5 sums the functions of each of the blocks in the Bluetooth baseband controller.

Table 2.5: Functions of blocks in the baseband controller

Block	Function
Register File	Information transmitter between the baseband and link manager.
Controller	Manages the timing and change of states.
Modem	Smoothing of the packet in the path of transmission and recovering the data in the path of receiving. Recovers the clock pulses as well.
Data Path	Compose packets to be transmitted and decompose packets received.
Clock Generator	Supplies the clock signals to all the blocks in the baseband
Hop Selector	Selects the hopping frequency for the transmission of the packets

2.6.2 Data Path

The data path consists of the packet composer and packet decomposer. In the packet composer, header error check (HEC) is being added to the header information being received from the register file and the controller. Then, the header is scrambled with a whitening word and encoded with 1/3 forward error correction (FEC)[23]. The rest of the payload information is then added with the cyclic redundancy check (CRC) will be scrambled and encrypted with 2/3 FEC. As for the packet decomposer, the header and payload information will be extracted from the received packets. The received packets may be inclusive of CRC and FEC or either one, based solely on the packet type. Figure 2.10 shows the components in the transmitter and receiver data path.

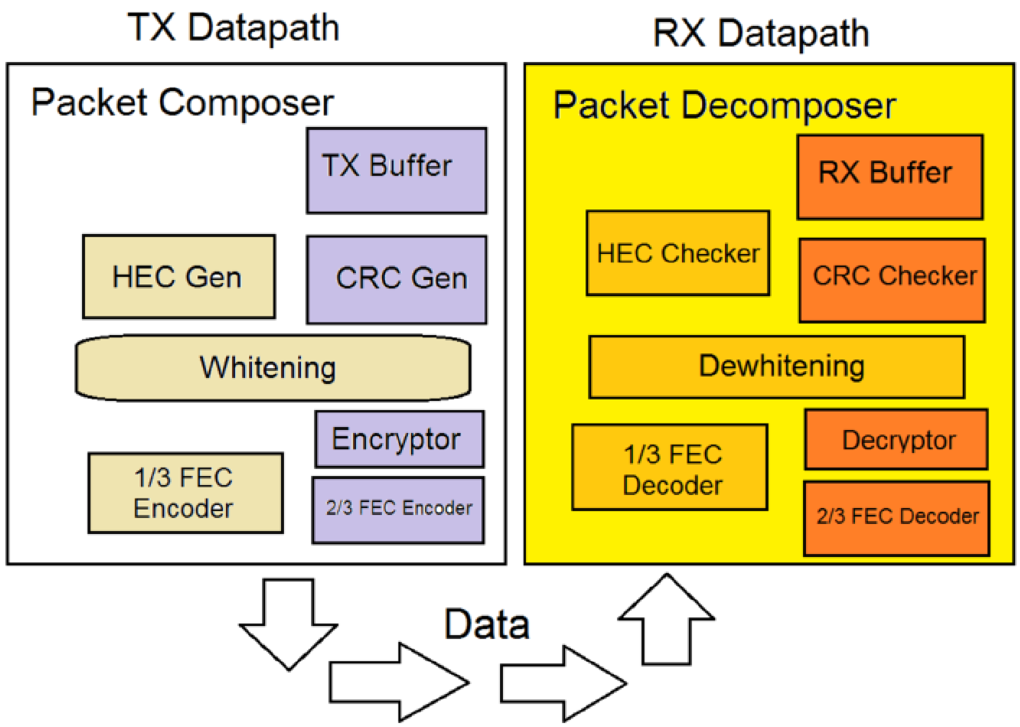

Figure 2.10: Components in Packet Composer and Decomposer

2.6.3 Products in the Market

This subchapter would be reviewing all of the products available in the market currently. Firstly, a review on MC71000, a Bluetooth baseband controller by Motorola. This baseband controller is targeted to be low power, high throughput and reduction of size. It is able to abide to the BLE protocol and functions on any device. The baseband controller is suggested to be used with another set of radio frequency transceiver. Any timing related transmissions are being taken care by this controller. Also, encryption such as CRC and HEC generations are generated from this controller. It handles all serial communications as well. The supply voltage for the core is about 1.8V. The supply going into the controller is between 1.8V to 3.3V. The whole structure is about 7mm x 7mm. It does not specify the size of the controller itself.

Then there is the LMX5452 by Texas Instruments which is able to support a data transfer rate of 921.6kbps. This baseband controller is significantly small of about 6.1mm x 9.1mm. It is only compliant to the classic Bluetooth protocol and the input or output ports voltages range between 1.6V to 3.6V. It supports both the ACL and SCO links. Another subsequent product by Texas Instruments to replace the LMX5452 is LMX5453. The size remains the same but there are several added functions with enhancements. The differences are summarized in Table 2.6. Looking into another baseband controller by Silicon Cores, the SI23BTB11, it is specifically developed to be marketed as a low-power Bluetooth applications. The data rate supported is 1Mbps. However, other specifications such as the physical outlook and power supply needed is not being disclosed in the factsheet. It is only able to support ACL links.

Table 2.6: Comparison of LMX5452 and LMX5453

Type	LMX5453SM/NOPB	LMX5452SMX/NOPB
Manufacturer	Texas Instruments	Texas Instruments
Frequency Range:	2.4 GHz	2.4 GHz
Operating Temperature	- 40 C to + 85 C	- 40 C to + 85 C
Package/Case	nFBGA-60	nFBGA-60
Packaging	Tray	Reel
Sensitivity	- 80 dBm	- 80 dBm
Series	LMX5453	LMX5452
Type	Bluetooth	Bluetooth
Interface Type	HCI, SPI, UART, USB	HCI, SPI, UART, USB
Mounting Style	SMD/SMT	SMD/SMT
Maximum Data Rate	723 kbps	1000 kb/s
Operating Supply Voltage:	2.5 V to 3.6 V	2.5 V to 3.6 V
Standard Pack Qty:	320	2500
Output Power:	-	0 dBm
Number of Receivers:	-	1
Number of Transmitters:	-	1
Supply Voltage - Max:	-	3.6 V
Supply Voltage - Min:	-	2.5 V

Table 2.7: Specifications for Classic Bluetooth

Specifications	Classic Bluetooth
Network/Topology	Scatternet
Speed	700 Kbps
Power consumption	Low (less than 30 mA)
Range	<30 m
RF band	2400 MHz
Frequency Channels	79 channels from 2.400 GHz to 2.4835 GHz
Modulation	GFSK(modulation index 0.35)
Latency	Approximately 100ms
Spreading	FHSS (1MHz Channel)
Link Layer	TDMA
Message size	358 bytes
Error detection	8 bit CRC(header), 16 bit CRC, 2/3 FEC(payload), ACKs
Security	64b/128b, user defined application layer
Throughput	0.7 to 2.1 Mbps
Nodes/Active Slaves	7

The Bluetooth baseband controller in this thesis will be designed and optimized based on the design specifications as listed in Table 2.7. It is not an apple to apple comparison but the speed and throughput will be taken into consideration. As this thesis only focuses on the transmitter of the datapath in the Bluetooth baseband controller, it is not easy for a good comparison as the speed and performance of the transmitter may differ once it is connected to the rest of the modules in the controller. Therefore, only the speed can be compared. The inherited design is designated for the classic Bluetooth settings. This is of different setting as compared to the BLE. The link layer would need to be modified if the existing settings are to be used on a BLE product.