BLUETOOTH ENHANCED DATA RATE BASEBAND MODELING

AND IMPLEMENTATION

Master thesis in Electrical Engineering Department
at Linköping Institute of Technology

by

Lei Zou

LiTH-ISY-EX--06/3870--SE

Supervisor: Prof. Dake Liu

Examiner: Prof. Dake Liu

Linköping, May 2006

Presentation Date	**Department and Division**	
2006-06-02		
Publishing Date (Electronic version)	**ISY**	Linköpings universitet

Language	**Type of Publication**	
× English	___ Licentiate thesis	**ISBN Master thesis**
___ Other (specify below)	___ Degree thesis	**ISRN: LiTH-ISY-EX--06/3870--SE**
	___ Thesis C-level	**Title of series**
	___ Thesis D-level	
	___ Report	
Number of Pages	___ Other (specify below)	**Series number/ISSN**
48		

URL, Electronic Version
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-3870

Publication Title
Bluetooth Enhanced Data Rate Baseband Modeling and Implementation

Author(s)

Lei Zou

Abstract
The main issue of this thesis is making the behaviour model of Bluetooth EDR (enhanced data rate) baseband signal processing. This Bluetooth baseband project is part of the soft defined radio project at electrical engineering department, Linköping University.
In this project, both the basic rate and EDR model were built and simulated. The GFSK and π/4 DQPSK digital modulation and demodulation were implemented in C code. The BER was tested to evaluate the demodulation results. Furthermore, the error correction (FEC) and the error checking (HEC,CRC) were also implemented according to the Bluetooth standards. The CRC flag was detected to test the payload demodulation results. Especially, GFSK and π/4 DQPSK specifications have to be combined with each other at sample rate of ADC. Finally, the basic rate and EDR model were simulated to measure the BER and CRC performance.
From the simulation results, the receiver filter, synchronization and channel condition were three key points in this Bluetooth EDR system implementation.
So we get further understanding about the Bluetooth system specification and DSP implementation methods.

Keywords
Bluetooth, Baseband, GFSK, π/4 DQPSK, BER, CRC, Filters, Synchronization, Channel, DSP

ABSTRACT

The main issue of this thesis is making the behaviour model of Bluetooth EDR (enhanced data rate) baseband signal processing. This Bluetooth baseband project is part of the soft defined radio project
at electrical engineering department, Linköping University.

In this project, both the basic rate and EDR model were built and simulated. The GFSK and π/4 DQPSK digital modulation and demodulation were implemented in C code. The BER was tested to evaluate the demodulation results. Furthermore, the error correction (FEC) and the error checking (HEC, CRC) were also implemented according to the Bluetooth standards. The CRC flag was detected to test the payload demodulation results. Especially, GFSK and π/4 DQPSK specifications have to be combined with each other at sample rate of ADC.

Finally, the basic rate and EDR model were simulated to measure the BER and CRC performance.

From the simulation results, the receiver filter, synchronization and channel condition were three key points in this Bluetooth EDR system implementation.
So we get further understanding about the Bluetooth system specification and DSP implementation methods.

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor and examiner Professor Dake Liu for giving me the opportunity to do my master thesis at this Bluetooth system project. I got many amazing ideas about the wireless communication system from him.
It is my pleasure to have a perfect tutor as Prof Dake Liu.

Secondly, I would like to thank Associate Professor Jerzy Dabrowski. He taught me radio frequency electronics knowledge and gave me the chance to do the RF front end project with him.

I also thank many PhD students in electrical engineering department. Di Wu, his kindly suggestion helps me a lot with this project. Kenny Johansson, our discussion with filters design is really useful.

I also thank all my friends in Sweden.

And most importantly to my beautiful mother, Din Din Jin, she is a great mother and supports me all the time.

CONTENTS

List of Figures

List of tables

Chapter 1 Background

1.1 Project Aim

Bluetooth is a universal radio standard recently developed for short range wireless communication applications. With the developing of the DSP algorithms, the baseband signal at receiver can have more complex signal processing.

In computer engineering division of Electrical Engineering Department, Linköping University, a project aimed at multi standard soft defined radio process is oriented and Bluetooth is one of that.

The project is aimed at Bluetooth enhanced data rate standards baseband modeling.

1.2 Methods and Tools

The project is divided into three parts.

Firstly, study the Bluetooth radio and baseband specifications.
Secondly, understand and rebuild the original Bluetooth GFSK baseband model.
Thirdly, design the new EDR (enhanced data rate) π/4 DQPSK baseband model.

The CAD tools are mainly in C and partly in Matlab.

1.3 Report Structure

At chapter 1, the project background is introduced here. At chapte2, a description of the Bluetooth original and EDR specification which includes the topology and packets. Chapter 3 is the implementation of the basic rate specification with GFSK modulation. Chapter 4 is the mean part of this thesis. The new EDR Bluetooth specification was modeling with π/4 DQPSK in detail and whole receiver architecture was introduced. Chapter 5 described the implementation methods and simulation results of previous chapter. The project conclusion and future work list at the end.

All the C codes are attached for future Soft Define Radio project.

Chapter 2 Introduction to Bluetooth

Bluetooth is a short range wireless networking technology which allows easy interconnection of mobile computers, mobile phones, headsets, PDAs and computer peripherals without cables. It is a low cost and low power technology. At specification version 1.2 of Bluetooth the data rate is 1Mbps. However, the data rate is increased to 2M/3M bps at enhanced specification version 2.0 at November 2004.

2.1 Radio Specification

Bluetooth device operate at 2.4 GHz in the license-free ISM band. The used band, between 2.4000-2.4835 GHz is divided into 1 MHz spaced channels. Totally, 79 channels are available. Bluetooth is a time division multiplexing (TDM), each time slot is 625 us. The frequency hopping spread spectrum (FHSS) technique is used here, after each packet, the radio frequency hops from one channel to one of the 79 channels according pseudorandom hopping sequence. So the ISI will not affect adjacent channels.

The max output power is 1mw under 10 meters range. The power summary was shown at table 2.1.

Power Class	Maximum Output Power (Pmax)	Nominal Output Power	Minimum Output Power[1)]	Power Control
1	100 mW (20 dBm)	N/A	1 mW (0 dBm)	Pmin<+4 dBm to Pmax Optional: Pmin[2)] to Pmax
2	2.5 mW (4 dBm)	1 mW (0 dBm)	0.25 mW (-6 dBm)	Optional: Pmin[2)] to Pmax
3	1 mW (0 dBm)	N/A	N/A	Optional: Pmin[2)] to Pmax

Table 2.1 RF power summary

2.2 Topology

Bluetooth devices work as a salve or a master which communicate with each other in a piconet or scatternet the piconet is some device shared a common channel and all salves must follow the frequency hopping and timing of the master. Several piconets can be established and linked tighter in a topology called “scatternet”. In scatternet, one device can be a master in one piconet and salve in another piconet, or salves in two piconets. Figure 2.1 showed the piconet and scatter modes.

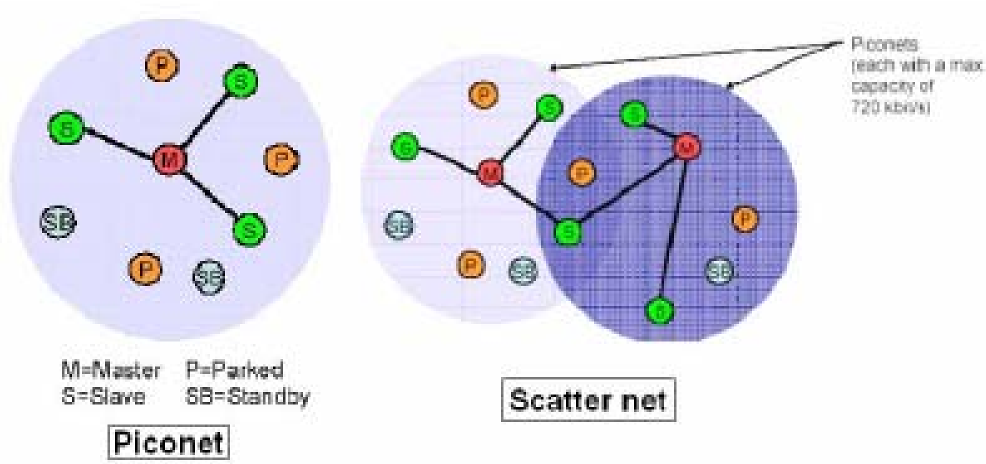

Figure2.1 Bluetooth Topology

Physical links
There are two different communication types (SCO and ACL) according to the different data types.

Synchronous Connection Oriented (SCO)
The SCO link is a symmetric point to point link between the master and slave. The SCO link reserves the channel bandwidth and timeslots and can be considered as a circuit-switched connection. The SCO link typically supports time-bounded information as voice or audio information.

Asynchronous ConnectionLess (ACL)
In the slots not reserved for SCO links, the master can exchange packets with any slave on single slot. The ACL link provides a packet-switched connection between the master and all active slaves in the piconet. For most ACL packets, packet retransmission is applied to assure data integrity. The ACL link is used for data communication.

2.3 Packets

Basic rate packet
The basic rate packet format has shown in Figure 2.2.
Each packet is made of three parts: Access code, Header and Payload. The access code (68 or 72 bits) is used to detect the presence of a packet and to address the packet to a specific device. The header (54 bits) contains all the control information involved with packet types and links. The payload (0-2744 bits) contains the actual message.

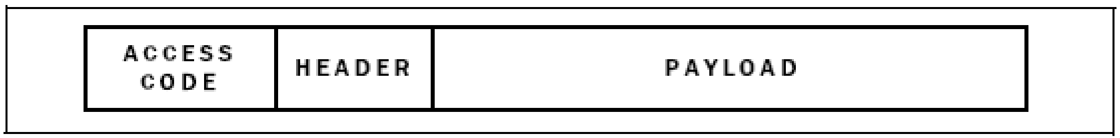

Figure 2.2 Basic rate packet

EDR packet
At EDR mode, the packet structure (Figure 2.3) is similar to the basic rate packets for combined the basic rate mode. It has 6 entities: the access code (68 or 72 bits), the header (54 bits), the guard period, the synchronization sequence (22bits), the enhanced data rate payload (0-8200bits) and the trailer.

The access code and header use the same modulation scheme as for the basic rate packets .The synchronization sequence, enhanced data rate payload and trailer are unique of EDR modulation scheme.

The synchronization sequence has the length of 11 symbols, 10 DPSK symbols followed one reference symbol. The guard time starts at the end of the GFSK symbol of the header and ends at the beginning of the synchronization sequence. It is used to switch in the hardware during the transition from one modulation scheme to another one. The guard time is between 4.75us and 5.25us.

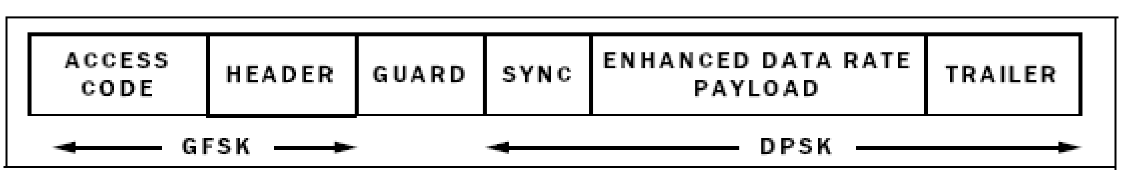

Figure 2.3 EDR packet

2.4 Access Code

Access code is used for identification, synchronization and DC offset compensation. There are three different types of access code: Channel Access Code (CAC), Device Access Code (DAC) and Inquiry Access code (IAC).

Code type	LAP	Code length
CAC	Master	72
DAC	Paged device	68/72[1]
GIAC	Reserved	68/72*
DIAC	Dedicated	68/72*

Table 2.2 Access code type

.

The completed structure of the CAC access code is shown at Figure 2.4.It concludes three parts as Preamble, synchronization word and trailer.

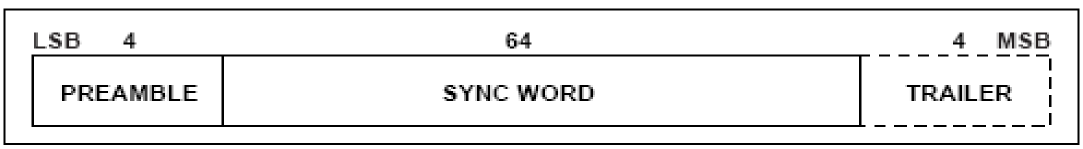

Figure 2.4 Access code format

The preamble is a fixed zero-one pattern of 4 symbols used for DC compensation. The sequence is either 1010 or 0101, depending on whether the LSB of the following sync word is 1 or 0.

The sync word is a 64 bits word derived from a 24 bit address. The synchronization word is made of BCH parity word, LAP and Barker sequence. In the receiver of the device, a sliding correlator correlates against the access code and triggers once a threshold peak exceeded. This auto correlation indicates the coming of a packet.

The trailer is fixed zero-one pattern of four symbols. The sequence is 1010 (MSB of sync word is 0)or 0101 (MSB of sync word is 1).The trailer together with three MSBs of the sync word forms a 7 bits pattern of alternating ones and zeros which may be used for extended DC compensation.

2.5 Header

Header contains link control information associated with the packet. See figure 2.5.

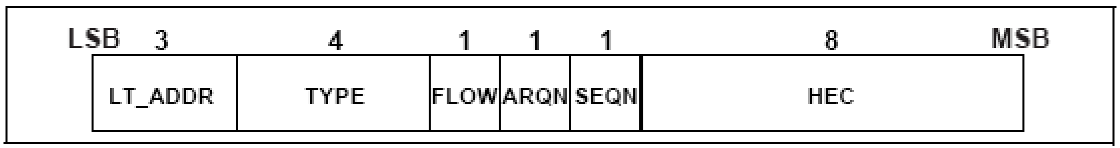

Figure 2.5 Header format

LT_ADDR
The three bits LT_ADDR field contains the logical transport address for packet. This field indicates the destination slave for a packet in a master to slave transmission slot and indicates the source slave foe a slave to master transmission slot.

TYPE
There are max16 different types of packets can be distinguished with 4 bits type code. The interpretation of the TYPE code depends on the logical transport address in the packet. Firstly, it will determine the packet is sent on SCO, eSCO or ACL logical transport. Secondly, it will determine whether the enhanced data rate has been enabled for the logical transport.

FLOW
This one bit is used for flow control of packets over the ACL link. When the RX buffer for the ACL link in the recipient is full, a STOP indication (FLOW=0) is returned to stop the transmission of data. When the RX buffer is empty, a GO indication (FLOW=1) is returned.

ARQN
The 1 bit acknowledgment indication ARQN is used to inform the source of a successful transmission of payload data with CRC. If the reception was successful, a positive acknowledge ACK is returned, otherwise a negative NAK is returned.

SEQN
The SEQN bit provides a sequential numbering scheme to order the data packet stream. For each new transmitted packet that contains data with CRC, the SEQN bit is inverted. By comparing the SEQN of consecutive packets, correctly received retransmissions can be discards.

HEC
Each header has a header-error-check to check the header integrity. The HEC is an 8 bit word generated by certain polynomial. Before checking the HEC, the receiver must initialize the HEC check circuitry with the proper 8 bit UAP (or DCI).

2.6 Payload

The basic rate payload structure shown at Figure 2.6 includes three parts: Payload Header, Payload Data and CRC.

The ACL Payload is formed by 16 bits four parts information. The logical Channel (L_CH) used to recognize the message from L2CAP and LMP. The Flow indicator is used to control the flow at the L2CAP level. Length is the bytes of payload. The undefined is not used at basic rate mode. The payload is 0-2712 bits. And 16 bit CRC at end of the payload.

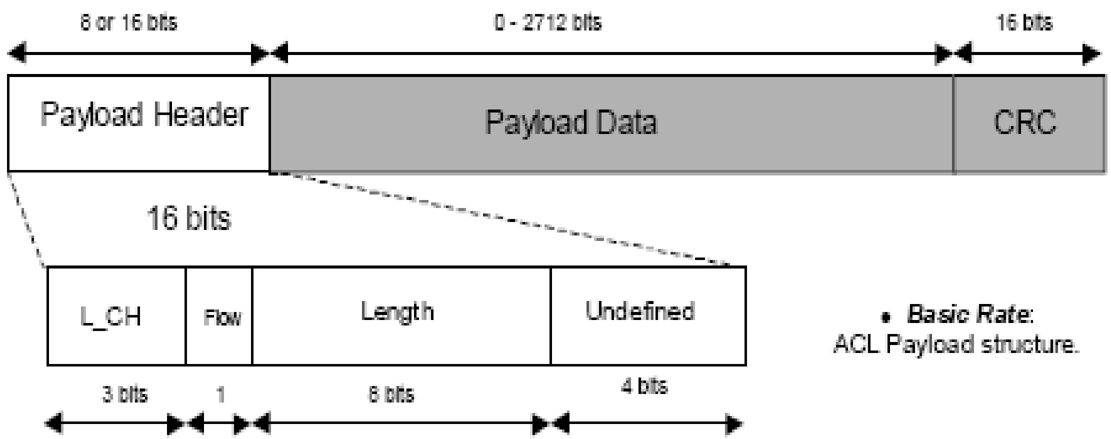

Figure 2.6 Basic rate payload format

The enhanced date rate payload is shown at Figure 2.7.
First, for payload header only reserved 3 bits needed. Second, at end of the payload, a 2 DPSK symbols trailer added. The trailer bits shall be all zero. The trailer is used to avoid the interruption between data packets and modulation.

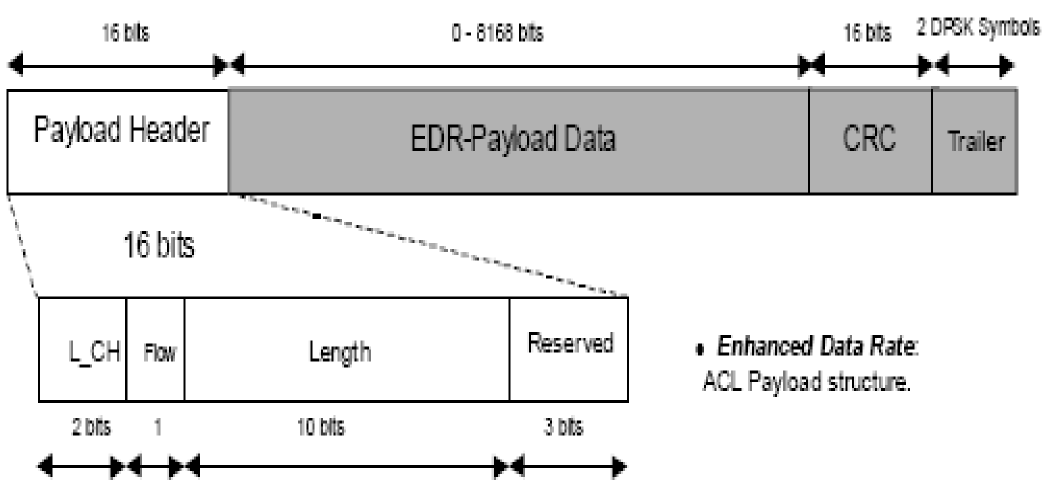

Figure 2.7 EDR payload format

2.7 Packet Types

For Bluetooth, there are three logical transports with distinct packet types are defined: The SCO logical transport, the eSCO logical transport and the ACL logical transport. For each of these logical transports, 15 different packet types can de defined by the 4-bit type code.

Seg-ment	TYPE code $b_3b_2b_1b_0$	Slot occu-pancy	SCO logical trans-port (1 Mbps)	eSCO logical trans-port (1 Mbps)	eSCO logical trans-port (2-3 Mbps)	ACL logical trans-port (1 Mbps) ptt=0	ACL logical trans-port (2-3 Mbps) ptt=1
1	0000	1	NULL	NULL	NULL	NULL	NULL
	0001	1	POLL	POLL	POLL	POLL	POLL
	0010	1	FHS	reserved	reserved	FHS	FHS
	0011	1	DM1	reserved	reserved	DM1	DM1
2	0100	1	undefined	undefined	undefined	DH1	2-DH1
	0101	1	HV1	undefined	undefined	undefined	undefined
	0110	1	HV2	undefined	2-EV3	undefined	undefined
	0111	1	HV3	EV3	3-EV3	undefined	undefined
	1000	1	DV	undefined	undefined	undefined	3-DH1
	1001	1	undefined	undefined	undefined	AUX1	AUX1
3	1010	3	undefined	undefined	undefined	DM3	2-DH3
	1011	3	undefined	undefined	undefined	DH3	3-DH3
	1100	3	undefined	EV4	2-EV5	undefined	undefined
	1101	3	undefined	EV5	3-EV5	undefined	undefined
4	1110	5	undefined	undefined	undefined	DM5	2-DH5
	1111	5	undefined	undefined	undefined	DH5	3-DH5

Table 2.3 Packet types for sync and async logical transport

Chapter 3 Basic Rate Mode of Bluetooth

3.1 GFSK Modulation

In a GFSK modulator, before the baseband signal pulses go into the FSK modulator, it is passed through a gaussian filter to make the pulse smoother so to limit its spectral width. The gaussian filter will do the pulse shaping with (+1,-1) pulse signal.GFSK is very popular in radio mobile communication systems such as GSM, DECT or Bluetooth. There are two obvious benefits for GFSK. First, it is insensitive to signal fading and interference with a narrow signal spectrum. Second, it has small non-linearity effect since constant envelop make a simple linear Amp at transmitter.
The modulation implementation methods were shown at Figure 3.1.

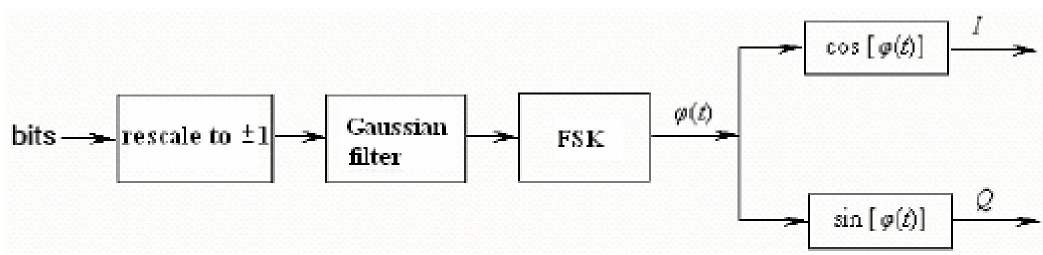

Figure 3.1 Modulation of GFSK

The equation of CPFSK

$$S(t)=\sqrt{\frac{2Es}{T}}\cos(\omega_0 t+\frac{a_n h\pi(t-n)}{T}+\psi_n) \qquad nT\le t\le(n+1)T$$
$$\psi_n=\pi h\sum_{i<n}a_i \tag{3.1}$$

We transfer this equation to discrete time representation as In-phase and Quadrature-phase component, k is the samples of the signal.

$$|A|=\sqrt{I^2(k)+Q^2(k)} \qquad I(k)=|A|\cos(\theta(k))$$
$$\theta(k)=arctag(I(k)/Q(k)) \qquad Q(k)=|A|\sin(\theta(k)) \tag{3.2}$$

Gaussian filter

The filter transfer function $\mathrm{g}\ (\mathrm{t})=\dfrac{1}{\sqrt{2\pi}\sigma T}e^{\frac{-t^2}{2\sigma T^2}}$ (3.3)

$$\sigma=\frac{\sqrt{\ln 2}}{2\pi BT}\ (\mathrm{BT}=0.5)$$

3.2 GFSK Demodulation

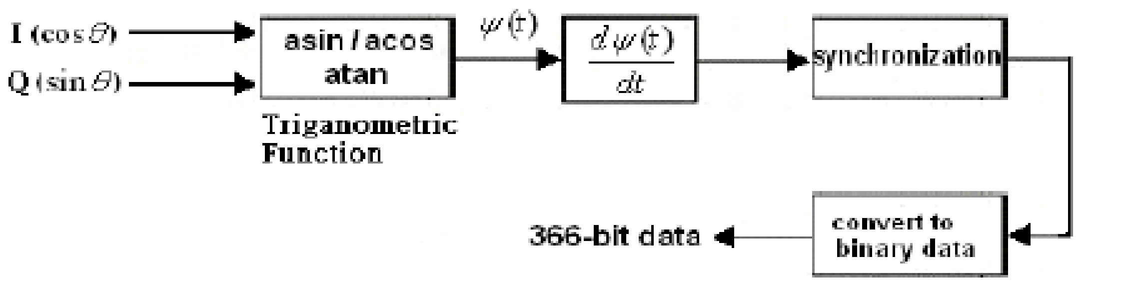

Figure 3.2 Demodulator of GFSK

The demodulator combined four blocks as shown at Figure 3.2.
The arctangent function does the phase extract as described at Equation 3.4.

$$\psi(t) = a\tan(\sin\theta / \cos\theta)$$
$$\Delta\psi = \psi(t) - \psi(t - T_{sample}) \qquad (3.4)$$

From GFSK modulator, we know '1' results in positive slope and '0' results in negative slope.In this method, binary bit value is demodulated by detection frequency in each bit period. If accumulate frequency is positive, the bit is '1'. Otherwise, the data bit is '0'.

Synchronization
Synchronization after GFSK demodulator is needed for frequency offset cancellation.

There is always a steepest slope at some fixed position in each bit period. Try to track every steepest point at each bit period so that we can find the frequency offset and compensate the offset by samples shift.

We assume every T the steepest slope occurs periodically. After deviation, the steepest point should be the largest value between the two adjacent sampled points. Once the value is not larger than both nearby ones, we shift the received date one sample either left or right depending on the biggest value position among these three sample points.

Here we consider 4 samples instead of 1 sample point to track and compensate the offset since the slopes in a couple of samples away can be more suitable.

Chapter 4 EDR Mode of Bluetooth

4.1 Introduction of π /4 DQPSK

The low data rate is the bottleneck of GFSK. We have to use new modulation methods nowadays even the noise comes a critical problem due to PSK constellation.
The Bluetooth uses the new DPSK modulation to increase date rate to 2Mbps or 3 Mbps. Here we focus on 2 Mbps π/4 DQPSK since the 8DPSK can be implemented with the same methods easily.

Differential modulations encode the transmitted information to a phase. This encoding introduced the memory to the signal since transmitted symbols depend on previous symbols.

The advantage of differential encoding is significantly simplified receiver structure. Phase tracking and channel estimation are not needed since absolute knowledge of carrier phase and the channel effect is not needed

The equation of π /4 DQPSK

$$s(t) = \cos\left[w_c t + \Delta\Phi\right] \qquad (4.1)$$

$$\Delta\Phi_k = \Phi_k - \Phi_{k-1} \qquad (4.2)$$

$\Delta\Phi$: the phase difference of two PSK modulated signals.
Φ_k : the next phase Φ_{k-1} : the current phase

Signal constellation
The signal constellation of π /4 DQPSK is a rotated version of 8 QPSK. It use two identical constellations which are rotated by π/4 with respect to one another, the π/4 DQPSK is a differential modulation which is more robust to noise.

Here we can see Figure 4.1, the π/4 DQPSK signal constellation with two 4 QPSK signal constellation as '+' for odd and '×' for even data bits.

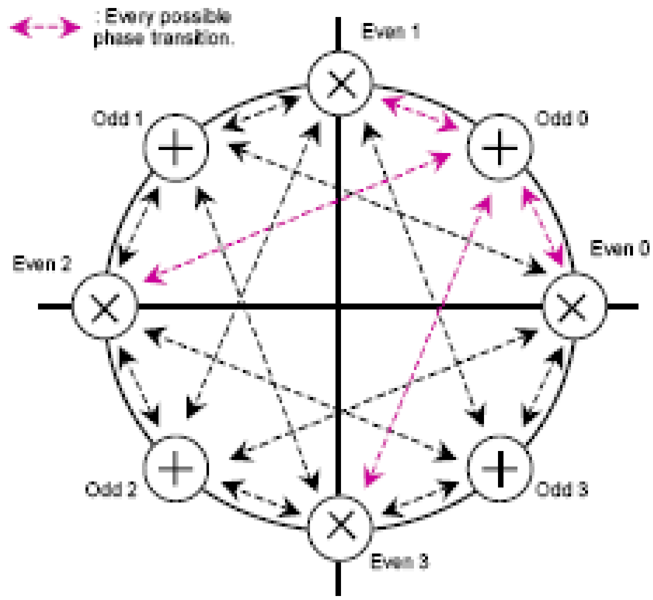

Figure 4.1 Signal constellation of π/4 DQPSK

From the signal constellation, we find only four possible phase transferred as [+π/4, - π /4, +3π/4 , -3π/4] between neighbor symbols. This reduces the phase shift from a maximum of 180° and the amplitude fluctuations are improved in this way.
In every symbol period T, there are four possible changes of phase which can be expressed with gray coding as shown at Table 4.1.

b_{2i+1}	b_{2i}	$\Delta\Phi$
0	0	π/4
0	1	3π/4
1	1	-3π/4
1	0	-π/4

Table 4.1 Gray coding map

4.2 Transmitter Modulation

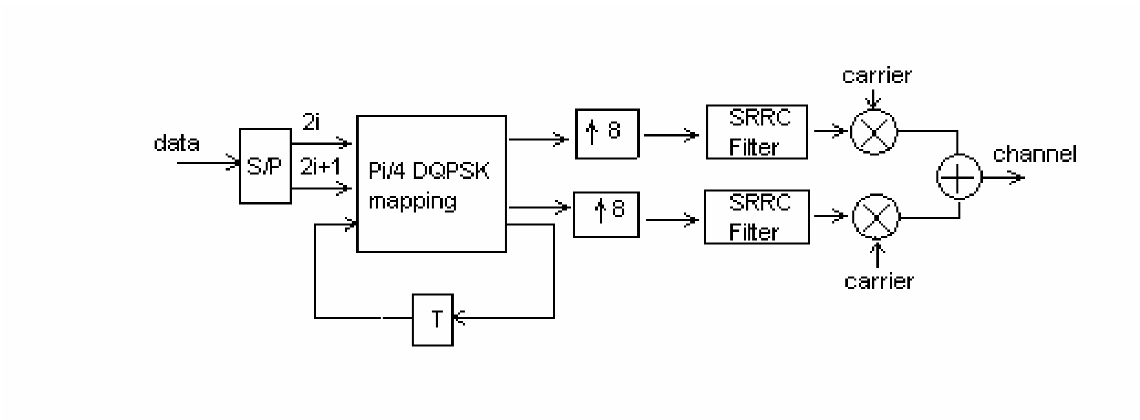

Figure 4.2 Transmitter of π/4 DQPSK

The transmitter modulator had been implemented as figure 4.2. Firstly, S/P block split input data to 2 bits per symbol. Followed, completed the differentially encoding using Look Up Table (LUT). For π/4 DQPSK, only four possible phase transitions, the mapping LUT for encode list here.

Current phase	Next phase 00	01	11	10
Φ0=π/8	Φ1	Φ3	Φ5	Φ7
Φ1=3π/8	Φ2	Φ4	Φ6	Φ0
Φ2=5π/8	Φ3	Φ5	Φ7	Φ1
Φ3=7π/8	Φ4	Φ6	Φ0	Φ2
Φ4=-7π/8	Φ5	Φ7	Φ1	Φ3
Φ5=-5π/8	Φ6	Φ0	Φ2	Φ4
Φ6=-3π/8	Φ7	Φ1	Φ3	Φ5
Φ7=-π/8	Φ0	Φ2	Φ4	Φ6

Table 4.2 Look up table of π/4 DQPSK

Note: If the $\Delta\Phi_k < -\pi$, then at demodulator mapping the $\Delta\Phi_k + 2\pi$,

If the $\Delta\Phi_k > \pi$, then at demodulator mapping the $\Delta\Phi_k - 2\pi$;

After that the signal was up sampled 8 times. The purpose of 8 over samples can gain the noise folding with reasonable hardware cost. After that a square root raised cosine (SRRC) filter followed for pulse shaping. At the end of this transmitter, the carrier frequency will up convert the base band signal to the 2.4 G radio frequency. Here, we only focus on the baseband signal.

4.3 SRRC Filter

The SRRC filter used separately both in transmitter and receiver which works as pulse shaping and Rx out band noise filter. It is used to limit the required bandwidth of the transmitted symbols by shaping them with a finite pulse form. So the inter symbol interference could be avoided.

The raise cosine filter transfer function H (f) =

$$\begin{cases} 1 & 0 \le |f| \le \dfrac{1-\beta}{2T} \\ \sqrt{(1-\sin\left(\dfrac{\pi(2fT-1)}{2\beta}\right))/2} & \dfrac{1-\beta}{2T} \le |f| \le \dfrac{1+\beta}{2T} \\ 0 & elsewhere \end{cases} \tag{4.3}$$

T is the input signal sampling period and r is the oversample rate, β is the rolloff factor. A specification comes from the specification of Bluetooth given at table 4.3

Specification of SRRC filter	
Oversample rate	8
Orders	16
Rolloff factor	0.4

Table 4.3 Specification of SRRC filter

Implement this filter is using FIR filter structure.

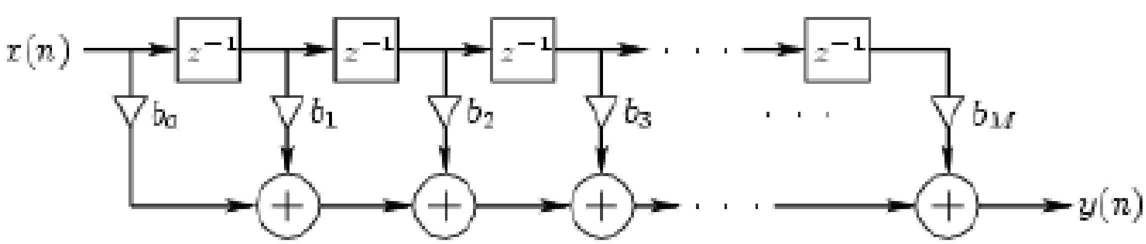

Figure 4.3.1 Implementation of filter

The figures below showed the time domain and spectrum simulation result of the SRRC filter.

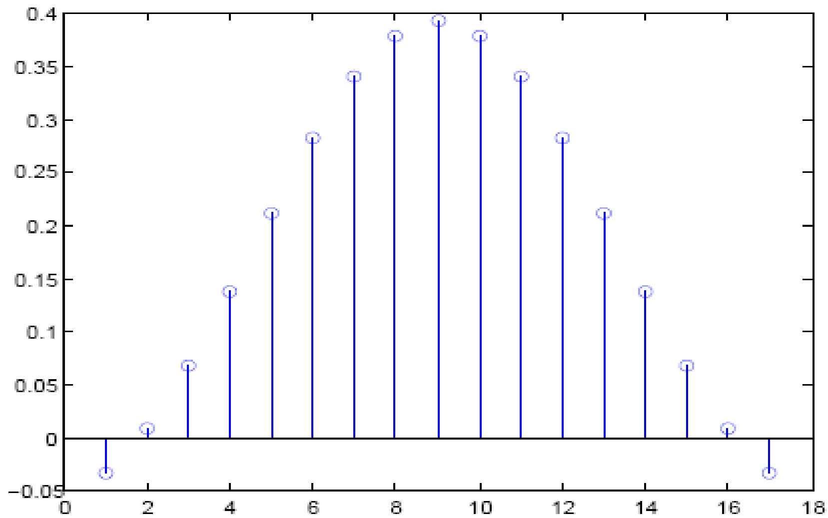

Figure 4.3.2 Stem plot of SRRC filter

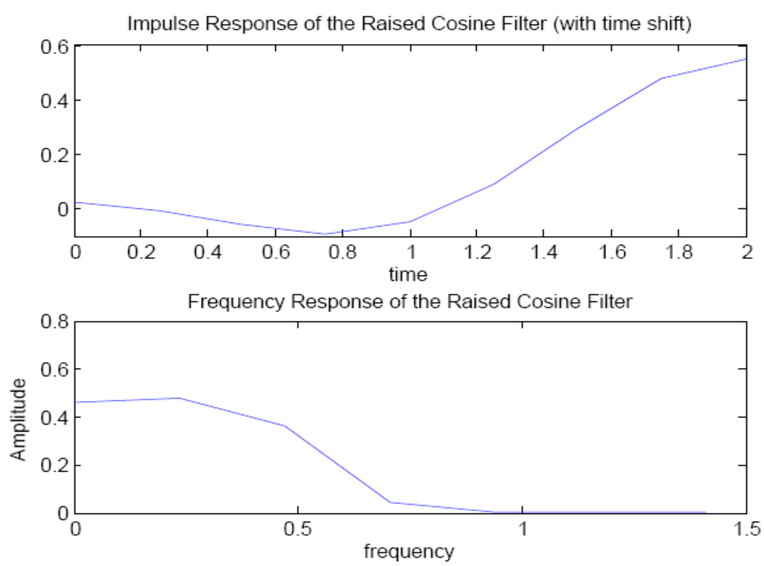

Figure 4.3.3 Impulse and frequency response of filter

4.4 Channel

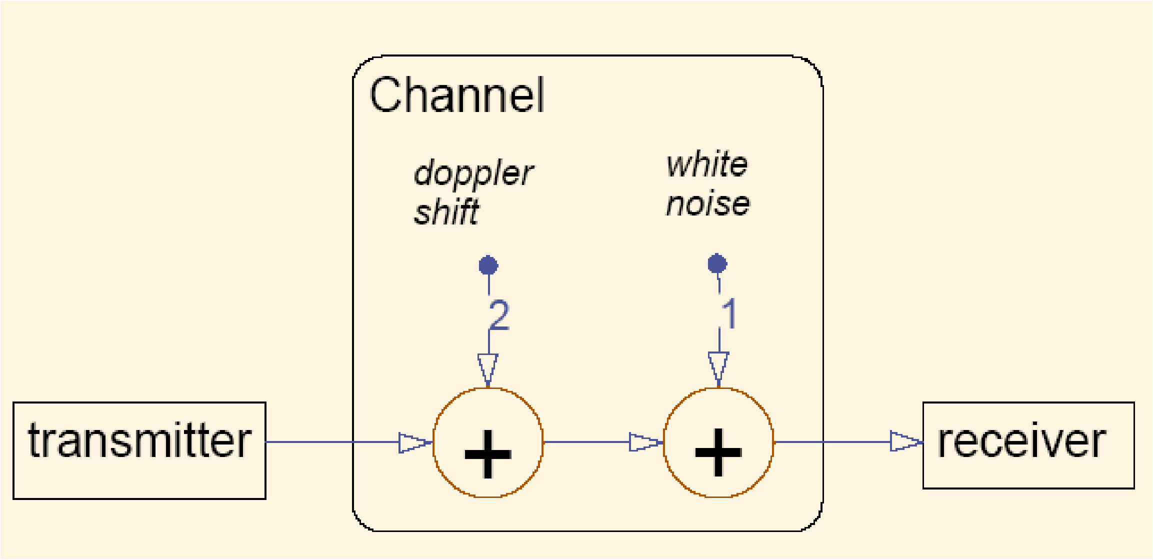

Figure 4.4.1 Channel model

Channels are where most signal distortion occurs. Radio channel is the most common case we used at wireless communication system.
In Bluetooth communication environment, the radio channel model has considered both Doppler shift [2] and Awgn [1] noise shown at Figure 4.4.1.

1 White noise

The white noise is the random noise according to the Gaussian distribution. The AWGN adds white Gaussian noise to a complex input signal.When the input signal is complex, this block adds complex Gaussian noise and produces a complex output signal.

Typical characteristics of white gaussian noise are a statistically independence of any two noise samples and a constant power spectral density equal to:

$$s(f) = N_0 / 2 \tag{4.4}$$

2 Doppler shift channel estimation

Since the transmitter and receiver are not stable all the time. The signal will suffered from the frequency shift as doppler shift.

The relation of the received frequency to the transmitted frequency is given

$$\nu' = \nu \frac{(c - \nu_r)}{(c - \nu_t)} \tag{4.5}$$

ν' : Frequency of receiver

ν : Frequency of transmitter

ν_r : Speed of receiver

ν_t : Speed of transmitter

C: Speed of light

Assume the transmitter is stationary and only the difference speed considered.

$$\nu' = \nu(1 - \frac{\nu_r}{c}) \tag{4.6}$$

The difference frequency according to the different speed can be presented as

$$\nu_\Delta = \nu * \nu_r / c \tag{4.7}$$

Use the equation at this Bluetooth application.

Vr: Max speed of receiver 2m/s

Vt: Frequency of transmitter 2.4G C: The speed of light 3×10^8 m/s

ν_Δ =16Hz

Angle frequency shift per packet =2 π ft=360° ×16Hz×625us= 3.6° (4.8)

The doppler estimation is 3.6° per packet. This result is quite small compared to the min decision region 45° of π/4DQPSK.

So we can simplified think the channel do not suffered from the Doppler shift in Bluetooth application.

Here, we take a look at the baseband signal spectrum simulation results at transmitter, channel and receiver at this Bluetooth system.

Figure4.4.2 Transmitter pulse shaping baseband signal
The transmitter used the Tx SRRC filter to define the wanted signal spectrum

Figure4.4.3 Channel baseband signal with Awgn noise
The channel model adds the constant power density noise at whole frequency range.

Figure4.4.4 Receiver baseband signal after Rx SRRC filter
The receiver use the filter to remove the outband (bandwidth 700 kHz) AWGN noise.

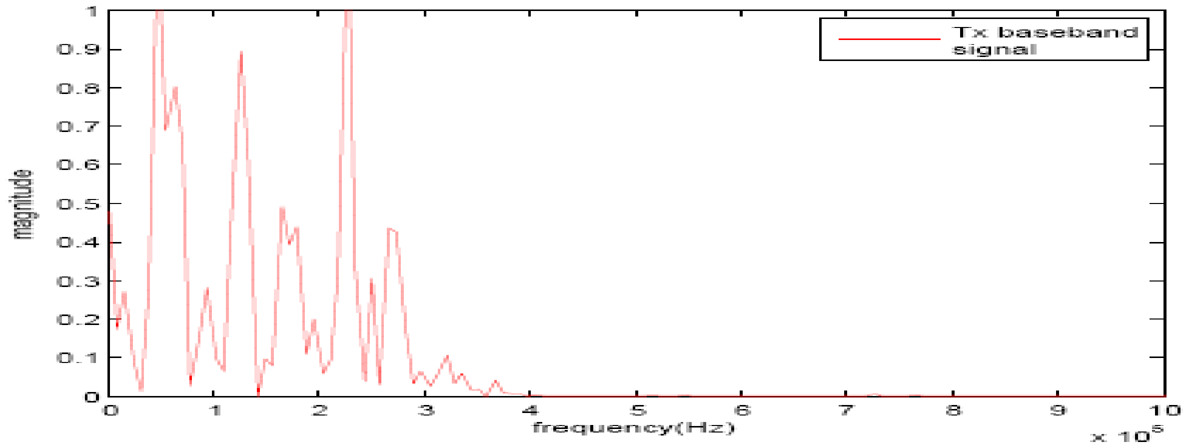

Figure 4.4.2 Transmitter baseband spectrum

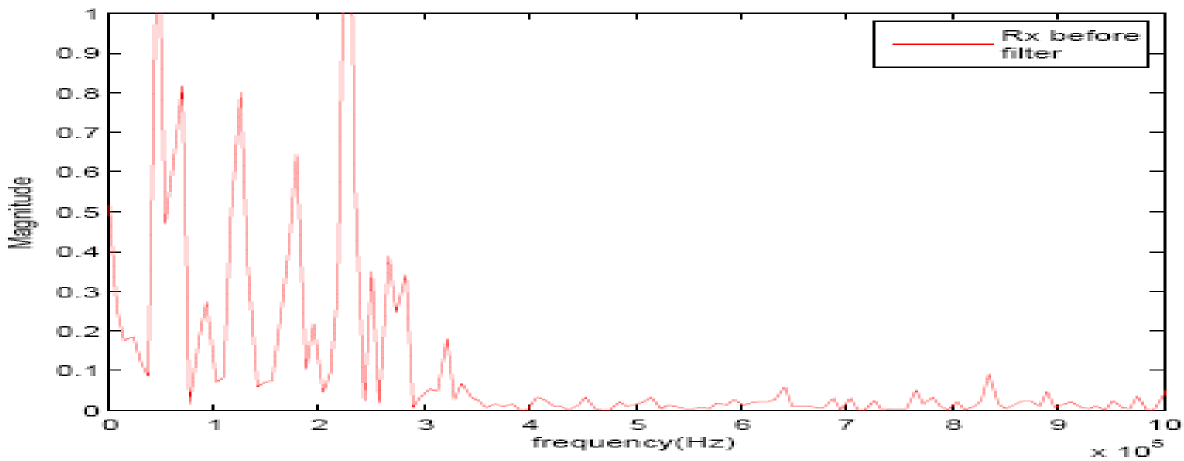

Figure4.4.3 Channel baseband spectrum

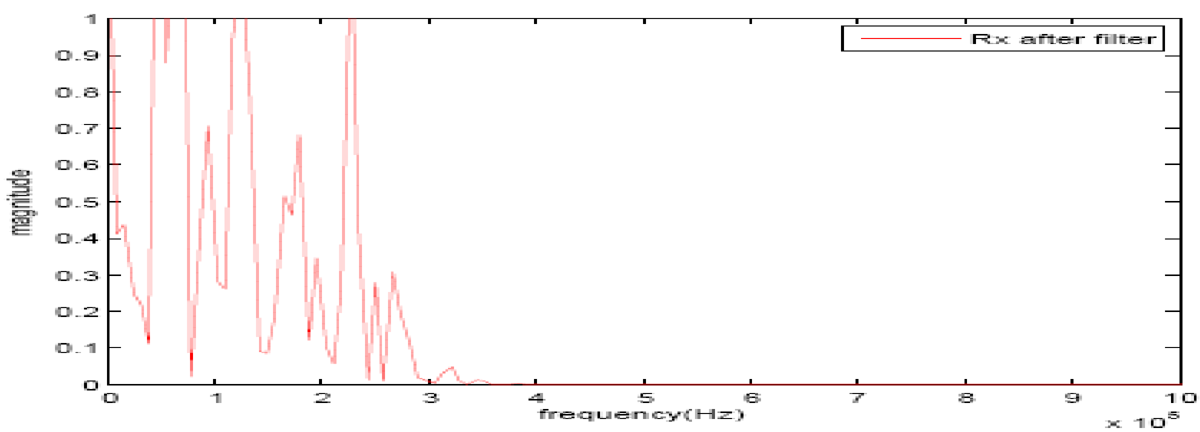

Figure4.4.4 Receiver baseband spectrum

4.5 Receiver Architecture

Bluetooth uses the low IF architecture which concludes the RF front end, interface circuit and digital baseband processing. Figure 4.5.1 showed this receiver architecture.

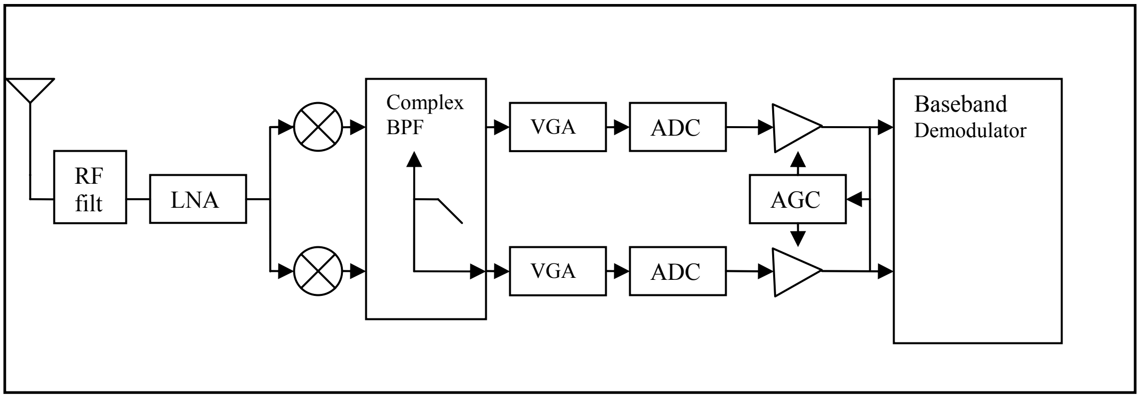

Figure 4.5.1 Bluetooth receiver architecture

RF front end

LNA is the first gain stage in the receive path after the RF filter. Noise figure of LNA is the critical part of the whole system.
Mixers down convert the received radio frequency signal into the baseband signal.
Complex band pass filter (BPF) will extracted the wanted 1MHz band signal.
Analog VGA is used to adapt the input signal level to ADC.

ADC circuit

The unique ADC sample frequency will determine the sample frequency of the GFSK and π/4 DQPSK modulation blocks.

The ADC sample frequency drift will cause the series problem which needed the carrier synchronization. However, Bluetooth is low cost standards which use the π/4 DQPSK to modulation and demodulation. As describe in chapter 4.1, no carrier frequency synchronization needed in this case.

Fsample of ADC is choosed as 16M Hz. Let's define the sample rates in this system.
GFSK data rate is 1M bps. Sample rate is 16 samples per bit.
π/4 DQPSK is 2M bps. Oversample rate is 8 samples per bit.

$$\text{Fsample} = 1\text{M} * 16 = 2\text{M} * 8 \qquad (4.9)$$

AGC circuit

The automatic gain control (AGC) immediately follows the ADC to scale the signal since the timing recovery circuit in the baseband signal modeling need unity incoming signal power.

The AGC has two parameters as reference level B and loop time constant A. The AGC will set the amplitude of the signal to a reference level B. The feedback loop was determined by the critical loop with the delay cell. The value of A is chosen so that the response is fast enough to track changes in signal amplitude.

With suitable values, the loop can track changes in signal amplitude correctly.

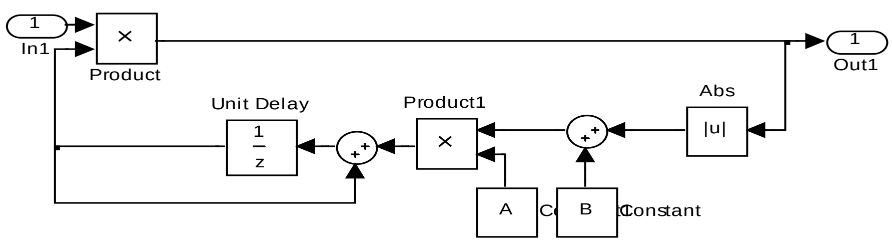

Figure 4.5.2 Implementation of AGC

4.6 Demodulator and Synchronization

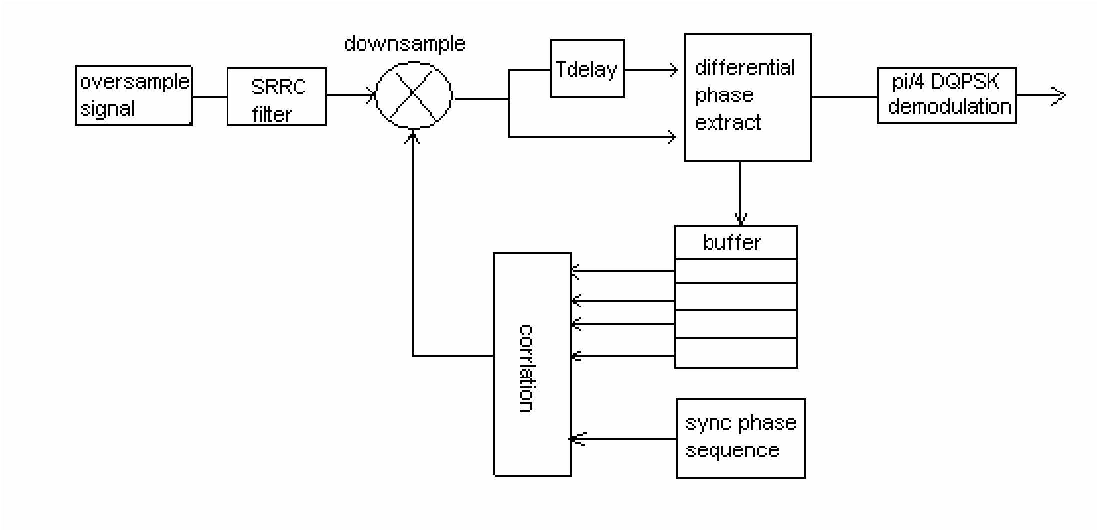

Figure 4.6.1 Baseband demodulator structure

Thanks for the ADC, we can handle the digital baseband signal directly. The SRRC filter used in order to respect the matching with the SRRC filter at transmitter so the outband noise was removed. We down sample the signal to the original data rate 2Mbps.

Then the delay block get the previous phase so that the differential phase can be extracted and be mapped in which region of the constellation.

After the synchronization we get the correct payload beginning to demodulate the message.

Lastly, we do the π/4 DQPSK demapping to recover the data.

Synchronization

Synchronization is one of the most important parts at Bluetooth EDR version.
The Bluetooth specification suggests a known sequence consists of a reference symbol followed by ten DPSK symbols. The length of the synchronization sequence is 11us.
The phase changes and format
{3π/4,-3π/4,3π/4,-3π/4,3π/4,-3π/4,-3π/4,3π/4,3π/4,3π/4}

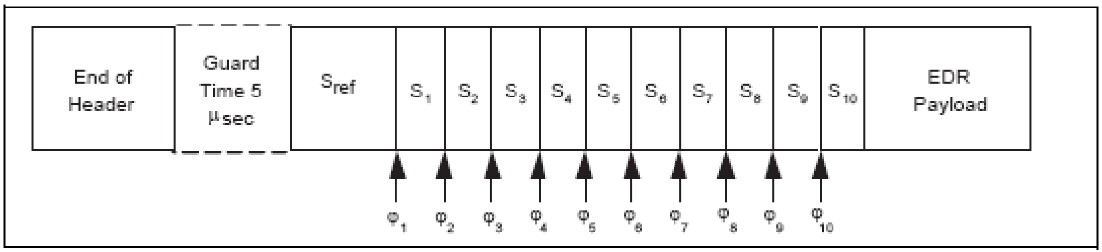

Figure 4.6.2 Synchronization sequence

The payload had pre pending the synchronization sequence during transmission.

We do correlation with different start points of the downsampled signals to extract the start point.

$$C_{corre} = \sum r(n) * s(n) \quad (5.0)$$

r (n) : received signal with noise. s(n) : reference synchronization sequence.

In this case, we take cross correlation as shown at figure 4.6.3 and find the highest peak with four different correlation procedures.

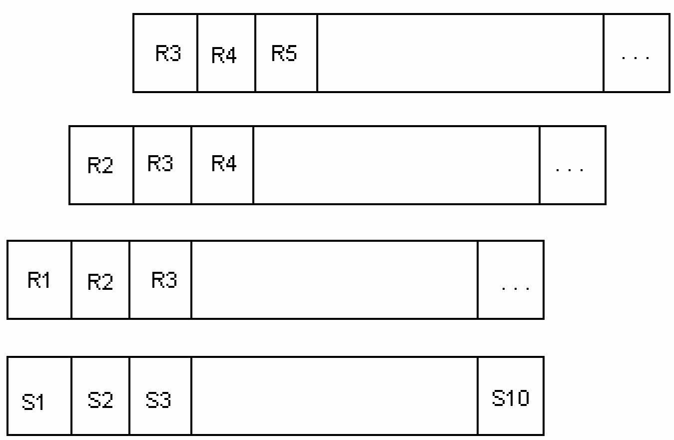

Figure4.6.3 Correlation procedures

The highest peak pointed the beginning used for payload demodulator.

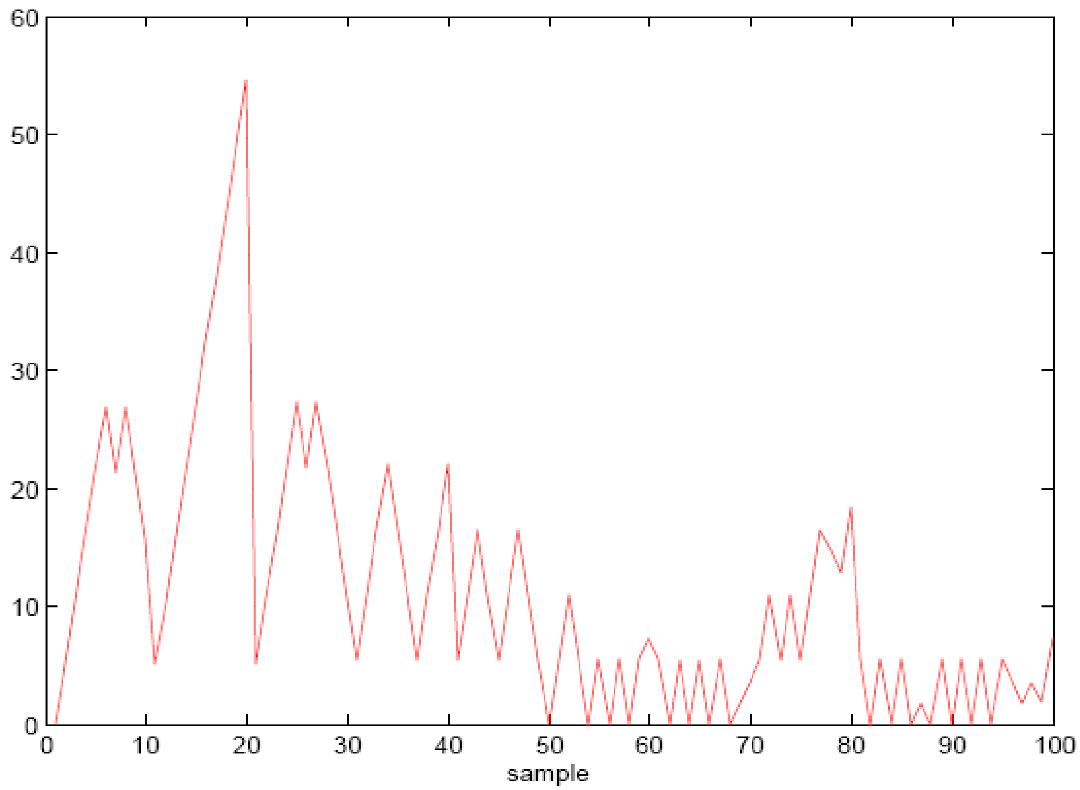

Figure 4.6.4 Correlation result

To check the complex baseband signal with synchronization, we plot the signal at In phase and In quadrature components.

The Rx complex baseband signal before downsampling is shown at Figure 4.6.5. The In phase and In quadrature signals have plot separately.

After downsampling and synchronization, the wanted I and Q baseband signal shown at Figure 4.6.6.

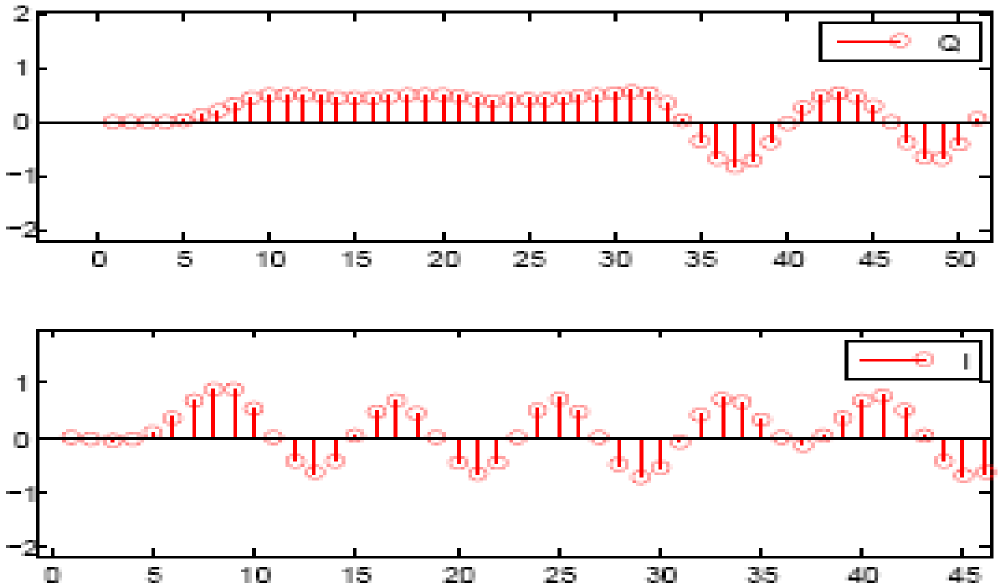

Figure 4.6.5 Rx I and Q signal before downsampling

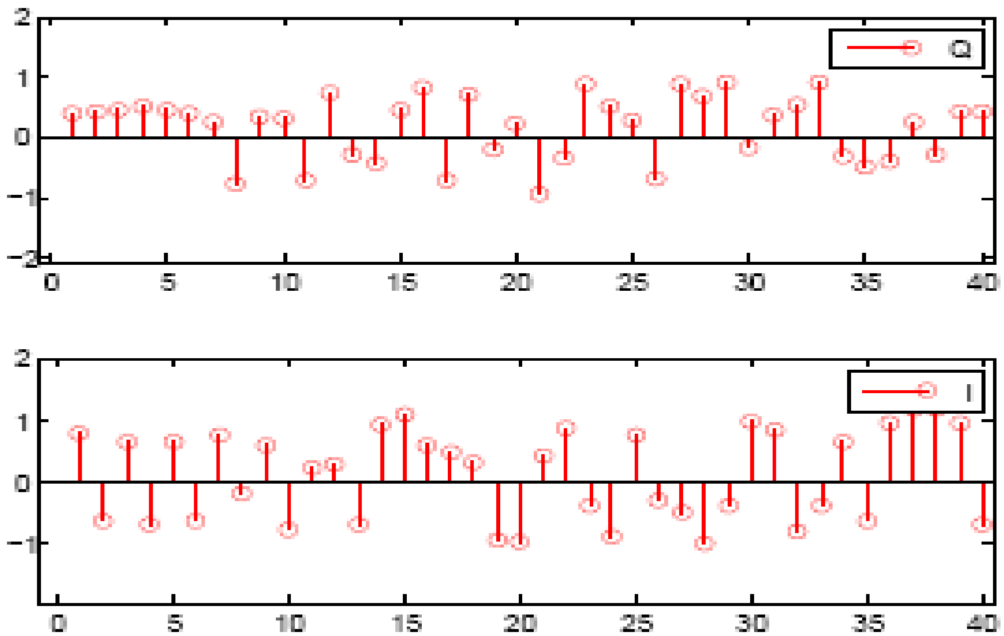

Figure 4.6.6 Rx I and Q signal after synchronization

4.7 Decoding

To demodulate the signal, we can use the signal constellation figure.

As we had explained at modulation chapter, the possible transmitted data phase is one of the value at {+π/4, -π/4, +3π/4, -3π/4}. So we can detect the phase and find the exacted data which had received.

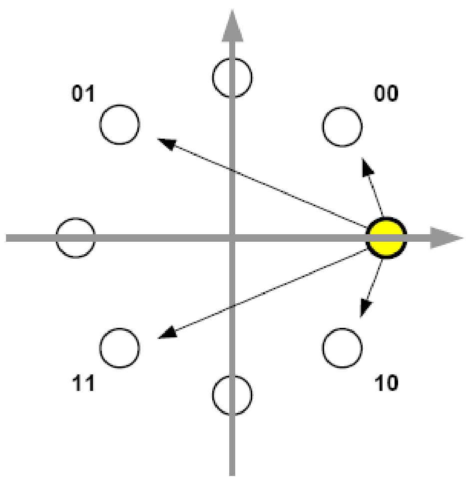

Figure4.7.1 Transferred phase

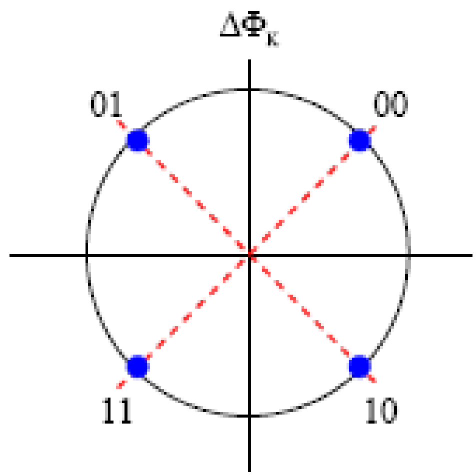

Figure 4.7.2 Data decision region

Chapter 5 System Simulations and Results

5.1Whole System Block Overview

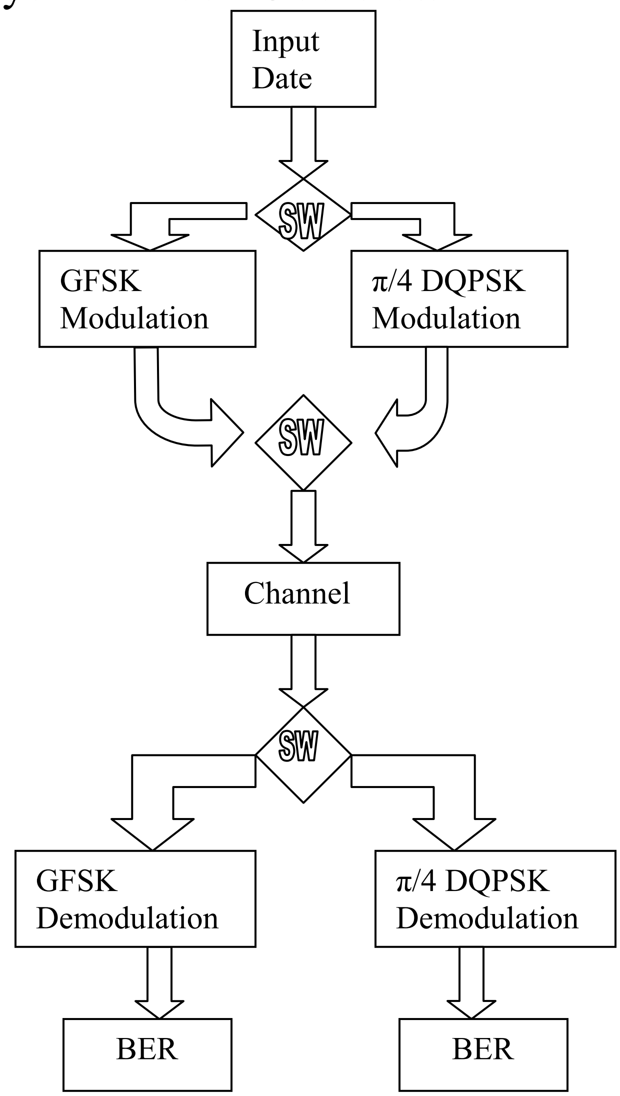

Figure 5.1 Bluetooth whole system flowcharts

5.2 Bluetooth Basic Rate Model

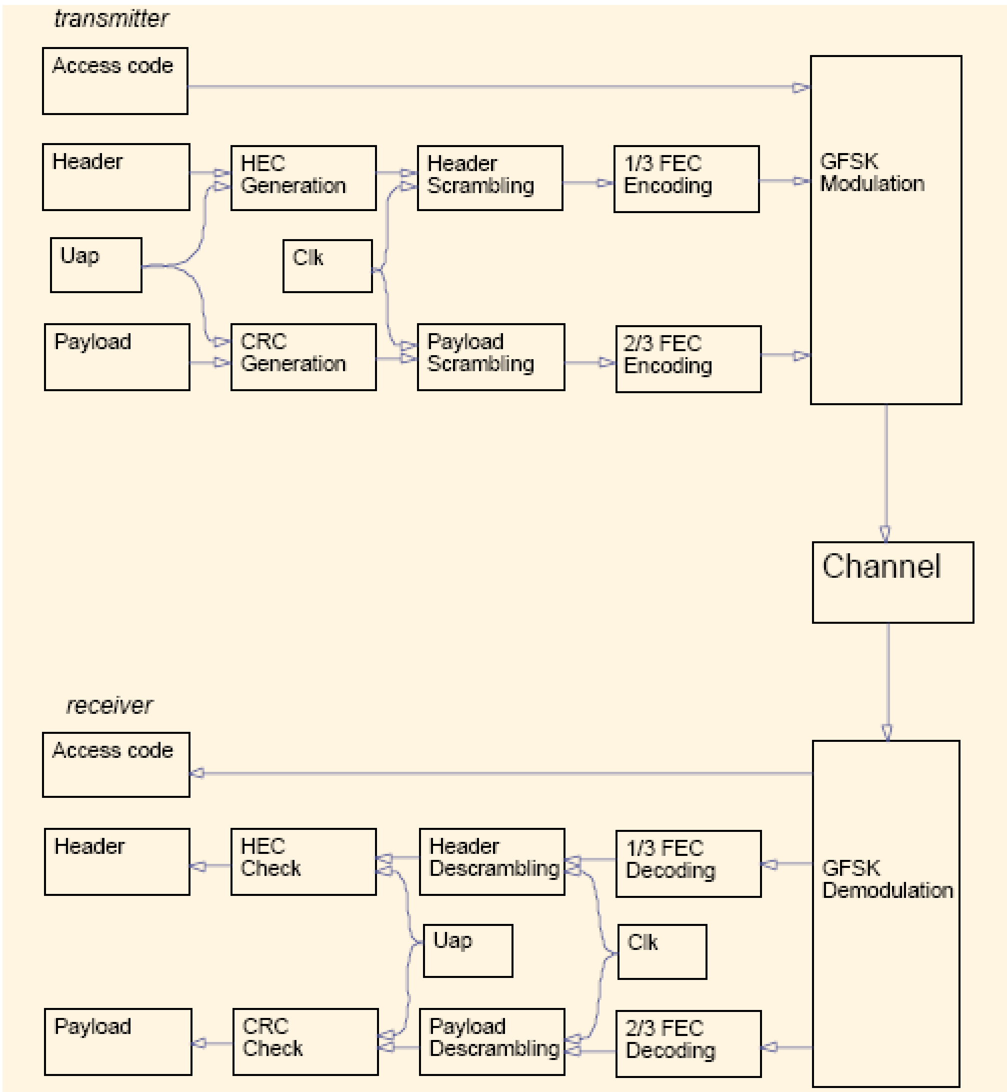

Figure 5.2 Bluetooth basic rate model

Blocks	Introduction	C Code implementation	
		Header	Main
access code	Chapter 2.4	accesscode.h	accesscode.c
Header	Chapter 2.5	header.h	header.c
Payload	Chapter 2.6	payload.h	payload.c
HEC generation	Appendix A	hec_gen.h	hec_gen.c
CRC generation	Appendix A	crc_gen.h	crc_gen.c
Header scrambling	N/A	hscram_gen.h	hscram_gen.c
Payload scrambling	N/A	pscram_gen.h	pscram_gen.c
1/3 FEC	Appendix B	fec13_gen.h	fec13_gen.c
2/3 FEC	Appendix B	fec23_gen.h	fec23_gen.h
GFSK modulator	Chapter 3.1	gfsk.h	gfsk.c
Channel	Chpater 4.4	channel.h	channel.c
GFSK demodulator	Chapter 3.2	gfsk_de.h	gfsk_de.c
1/3 FEC decode	Appendix B	fec13_de.h	fec13_de.c
2/3 FEC decode	Appendix B	fec23_de.h	fec23_de.c
Header descramb	N/A	hscram_de.h	hscram_de.c
Payload descramb	N/A	pscram_de.h	pscram_de.c
HEC check	Appendix A	hec_de.h	hec_de.c
CRC check	Appendix A	crc_de.h	crc_de.c
Clk	N/A	clk.h	clk.c
uap	N/A	uap.h	uap.c

Table 5.2 Implementation of basic rate system blocks

5.3 Bluetooth EDR Model

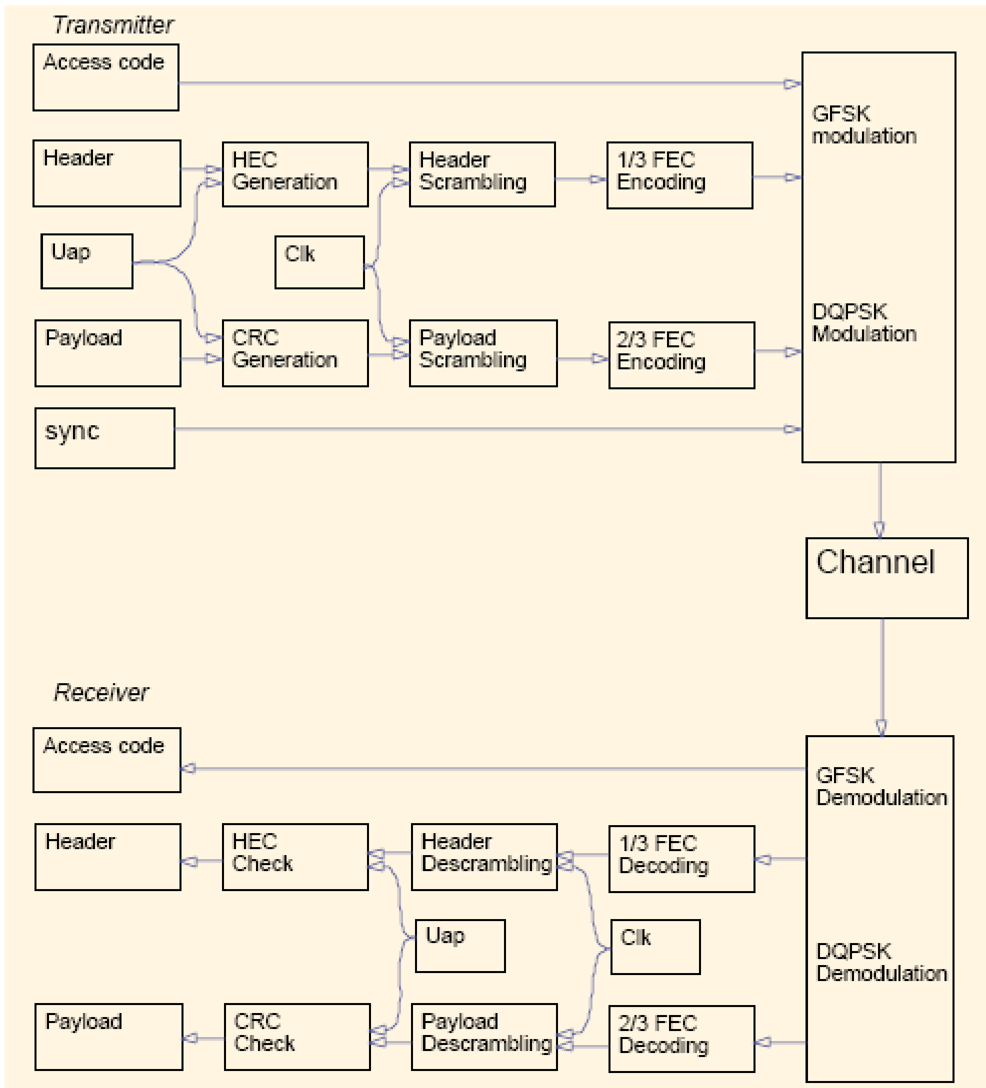

Figure 5.3 Bluetooth EDR model

Blocks	Introduction	C Code implementation Header	 Main
access code	Chapter 2.4	accesscode.h	accesscode.c
Header	Chapter 2.5	header.h	header.c
sync	Chapter 4.6	sync.h	sync.c
Payload	Chapter 2.6	payload.h	payload.c
HEC generation	Appendix A	hec_gen.h	hec_gen.c
CRC generation	Appendix A	crc_gen.h	crc_gen.c
Header scrambling	N/A	hscram_gen.h	hscram_gen.c
Payload scrambling	N/A	pscram_gen.h	pscram_gen.c
1/3 FEC	Appendix B	fec13_gen.h	fec13_gen.c
2/3 FEC	Appendix B	fec23_gen.h	fec23_gen.h
DQPSK modulator	Chpater 4.1	qpsk.h	qpsk.c
Channel	Chapter 4.4	channel.h	channel.c
DQPSK synch & demodulator	Chapter 4.6 Chapter 4.7	qpsk_de.h	qpsk_de.c
1/3 FEC decode	Appendix B	fec13_de.h	fec13_de.c
2/3 FEC decode	Appendix B	fec23_de.h	fec23_de.c
Header descramb	N/A	hscram_de.h	hscram_de.c
Payload descramb	N/A	pscram_de.h	pscram_de.c
HEC check	Appendix A	hec_de.h	hec_de.c
CRC check	Appendix A	crc_de.h	crc_de.c
Clk	N/A	clk.h	clk.c
uap	N/A	uap.h	uap.c

Table 5.3 Implementation of EDR system blocks

5.4 Bluetooth π/4 DQPSK Digital Modulation

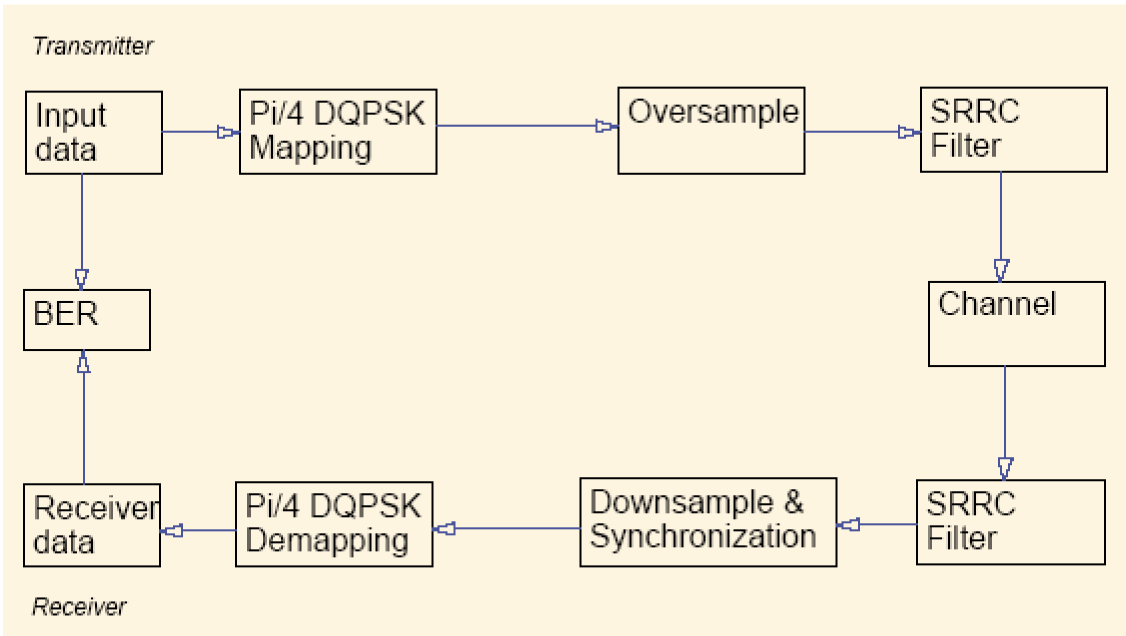

Figure 5.4 Bluetooth π/4 DQPSK model

Blocks	Introduction	C code implementation	
π/4 DQPSK mapping	Chapter 4.2	Attached Page 118-119	qpsk.c qpsk.h
Oversample	Chapter 4.3	Attached Page 120	
Tx SRRC filter	Chapter 4.3	Attached Page 121-122	
Channel	Chapter 4.4	Attached Page 123-124	channel.c channel.h
Rx SRRC filter	Chapter 4.6	Attached Page 127	qpsk_de.c qpsk_de.h
Downsample & synchronization	Chapter 4.6	Attached Page 128-133	
π/4 DQPSK demapping	Chapter 4.7	Attached Page 134-135	

Table 5.4 Implementation of π/4 DQPSK modulation and demodulation

5.5 Simulation Results

Here the C code of each block was used to simulation the system performance according to the pervious system flow charts.
1 Simulated the GFSK model to verify the basic rate system implementation method.

2 Simulated the π/4 DQPSK model to find the performance difference between with synchronization and without synchronization.

3 Simulated the π/4 DQPSK model to find the performance difference between with Rx filter and without Rx filter both on synchronization condition.

The BER measure block was implemented to measure the performance.
BER is the probability of making an error in detector when transmitting a symbol due to AWGN in this case.

GFSK model simulation result

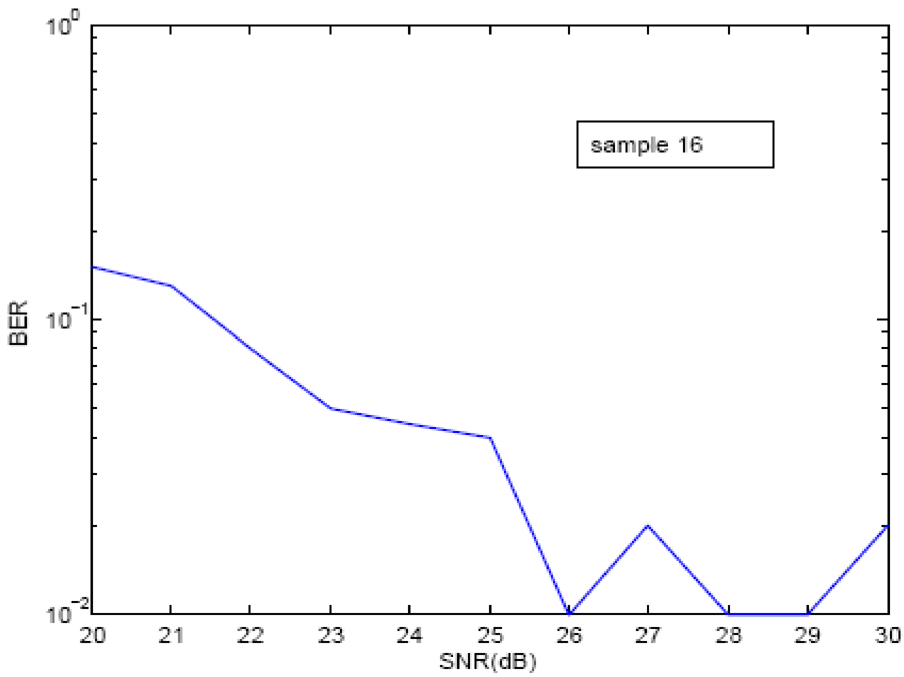

Figure 5.5 GFSK simulation result

Based on the Bluetooth basic rate system at figure 5.2 simulated different BER under channel of different SNR. The sample rate per bit is 16. Simulation was done without receiver synchronization.

The BER is decreased with SNR increased and the BER decreased dramatically at the range form 18 dB to 25 dB.

Note: The BER is still too high in this case. More input dates should be test in the future to evaluate the accuracy of this GFSK model. What’s more, the offset compensation block should be built in a better way.

π/4 DQPSK model simulation results

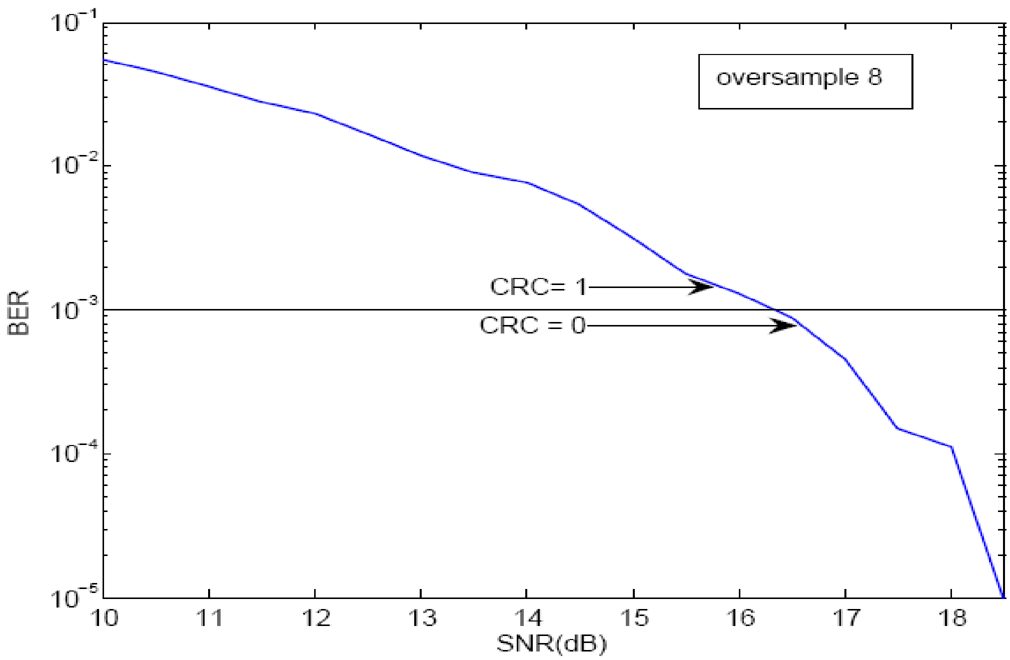

Figure 5.6 π/4 DQPSK simulation with synchronization

This simulation was done with the system at Figure 5.3.

The oversample rate is 8 and the receiver filter is 16 orders. Synchronization was done before desmapping the data. The test channel is awgn channel with different SNR from 10 dB to 20 dB.

From the simulation result, the BER is decreased dramatically as the channel SNR increased. Once the BER lower than 10^{-3}, the CRC flag will become 0 which means accepted payload received at this system.

π/4 DQPSK model simulation results

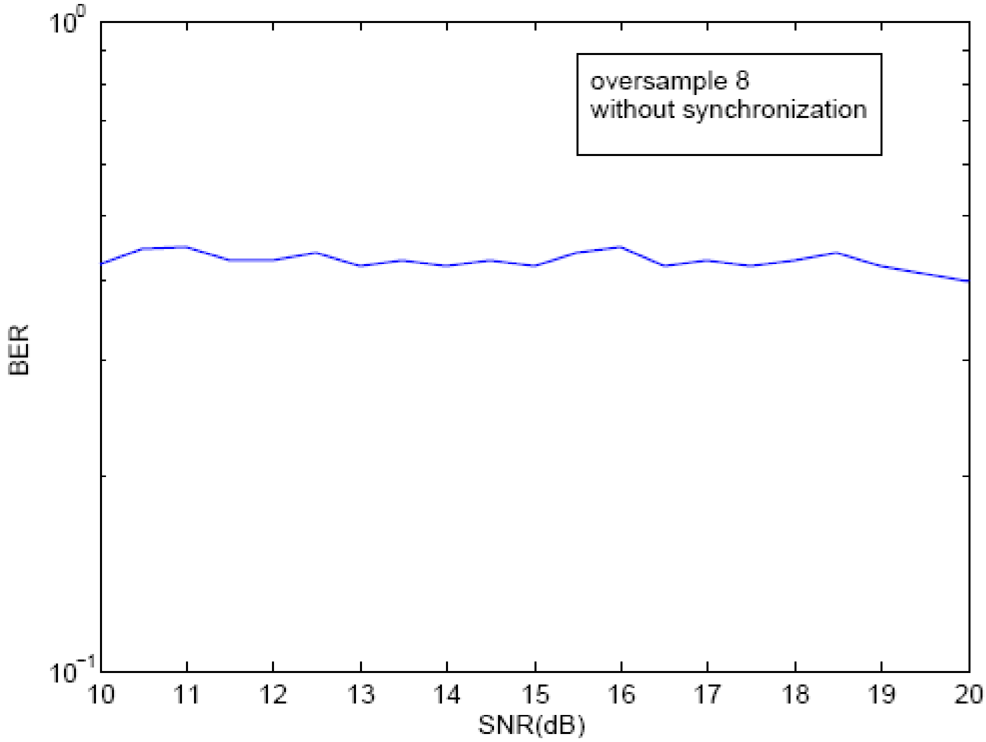

Figure 5.7 π/4 DQPSK simulation without synchronization

This simulation was done with the system at Figure 5.3

The oversample rate is 8 and the receiver filter is 16 orders. Synchronization was not done before desmapping the data. The test channel is awgn channel with different SNR from 10 dB to 20 dB.

From the simulation result, we found the very high BER is not related to the channel SNR and remain at certain level all the time. We must do the synchronization at receiver otherwise the whole receiver will not work as it supposed to do.

π/4 DQPSK model simulation results

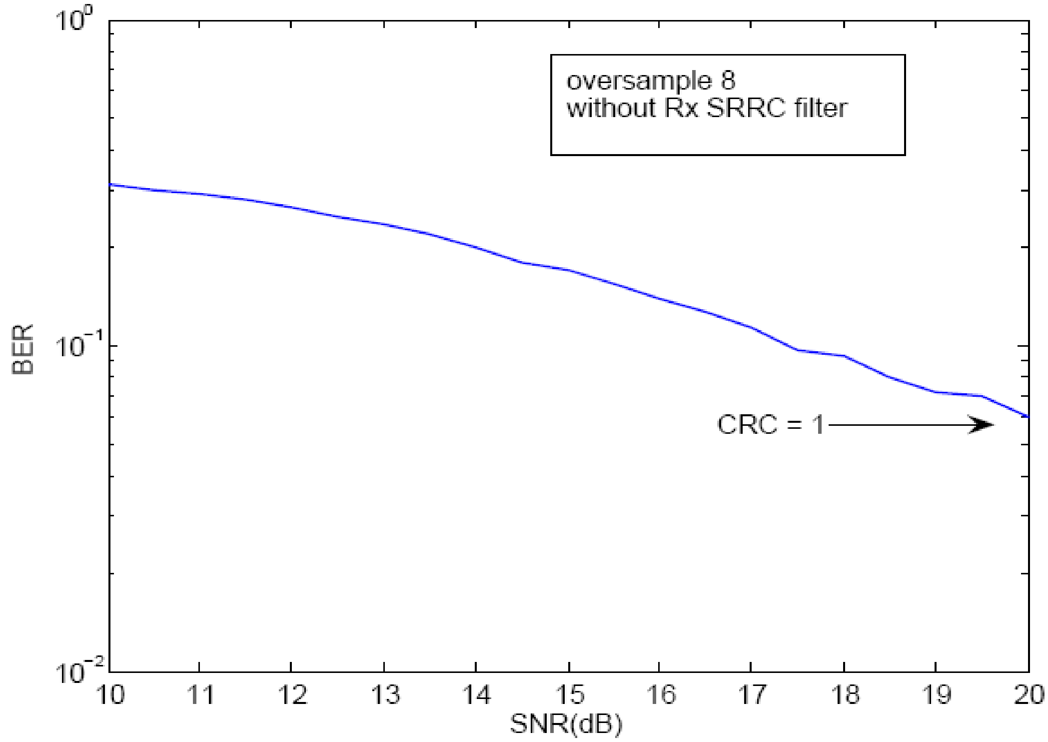

Figure 5.8 π/4 DQPSK simulation without filter

This simulation was done with the system at Figure 5.3.
The oversample rate is 8. The receiver SRRC filter was not used in this simulation. Synchronization was done before desmapping the data. The test channel is awgn channel with different SNR from 10 dB to 20 dB.

From the simulation result, the BER is decreased smoothly as the channel SNR increased. The poor BER can not pass the CRC checking (CRC flag = 1).

The reason of poor BER is we do not filter out outband noise of baseband signal at receiver.

π/4 DQPSK model simulation results

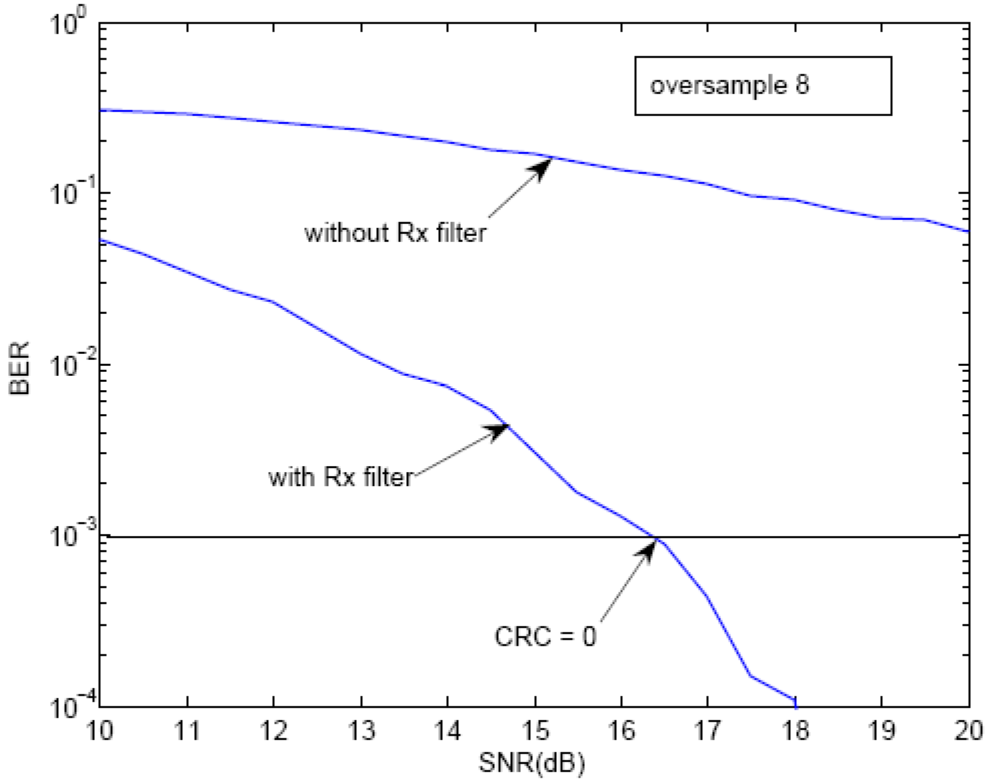

Figure 5.9 π/4 DQPSK results compared with and without filter

Here, to show the BER difference clearly, simulation results at Figure 5.6 and Figure 5.8 was compared in one Figure 5.9.

So we must use the receiver SRRC filter at this Bluetooth receiver system to satisfy the CRC checking algorithm.

Chapter 6 Conclusion and Future Work

This Bluetooth EDR baseband C modeling gave the basic methods of DSP implementation of the reconfigurable receiver.

The basic rate model was rebuild and worked with certain BER.

The EDR baseband modulation and demodulation model were implemented and the CRC error detection also indicated the accepted BER level.

With this project, I studied deeply into the wireless communication field. And get the ability to understand the real industry standards. Furthermore, I also improved my C programming skills during the implementation procedure.

There are some parts of this Bluetooth project should be get further research.

Firstly, at the EDR version, the 8DPSK modulation can be implemented with the same methods as $\pi/4$ DQPSK.

Secondly, the GFSK basic rate BER is still too high for this system. I suggested more test should be done with it.

Last but not least, the receiver can done the differential phase extract before downsampling to get improved BER performance. However, more MIPS will be cost at synchronization since the data will become quite lager compared with current receiver.

Appendix A

Header Error Check (HEC)

HEC performed error detection on the packet's header by checking the remainder in receiver part. The HEC generation and checking has been shown at Figure 6.1.

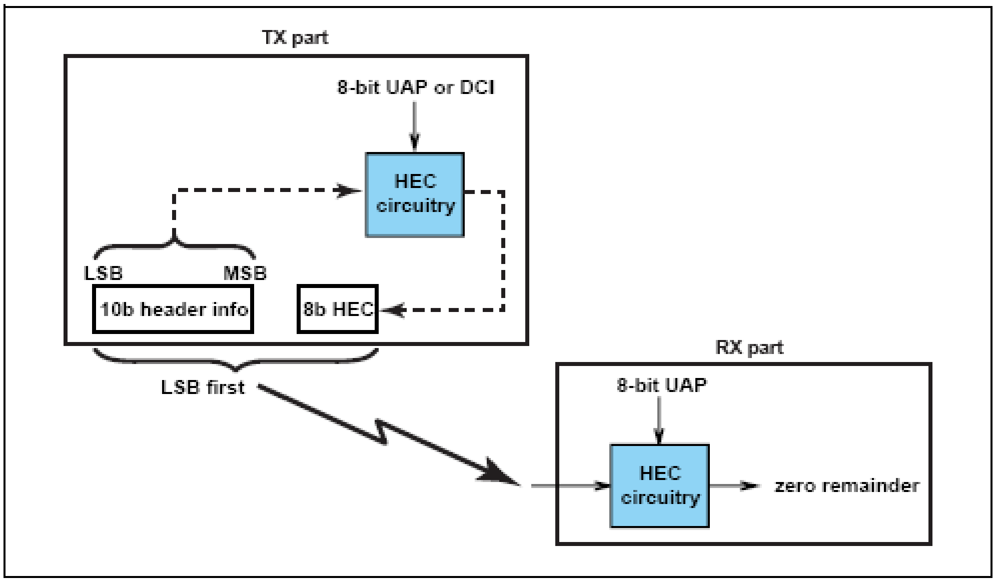

Figure 6.1 HEC generation and checking algorithm

The generator polynomial is

$$g(d) = (D+1)(D^7 + D^4 + D^3 + D^2 + 1) = D^8 + D^7 + D^5 + D^2 + D + 1$$

Initially this circuit shall be pre-loaded with the 8 bits UAP such that the LSB of the UAP goes to the left-most shift register element and MSB goes to the right most elements. See Figure 6.2.

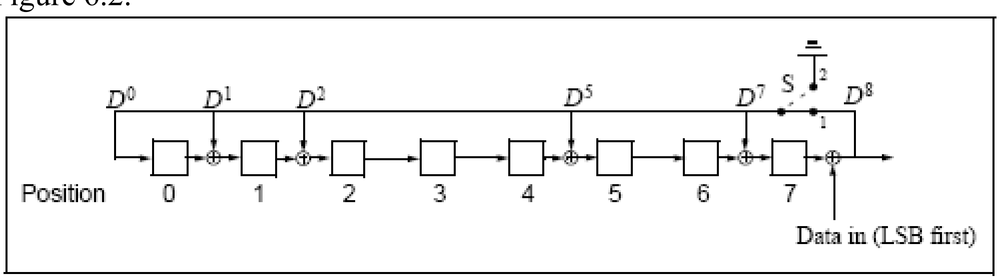

Figure 6.2 LFSR circuit of HEC

Cyclic Redundant Check (CRC)

The 16 bit LFSR for the CRC is constructed similarly to the HEC using the generator polynomial $g(D) = D^{16} + D^{12} + D^{5} + 1$.

The 8 left-most bits shall be initially loaded with the 8-bit UAP while the 8 right-most Bits shall be reset to zero. The switch S shall be set in position 1 while the data is shifted in. After the last bit has entered the LFSR, the switch shall be set in position 2.
The CRC generation was shown at Figure 6.3 and LFSR was shown at Figure 6.4.

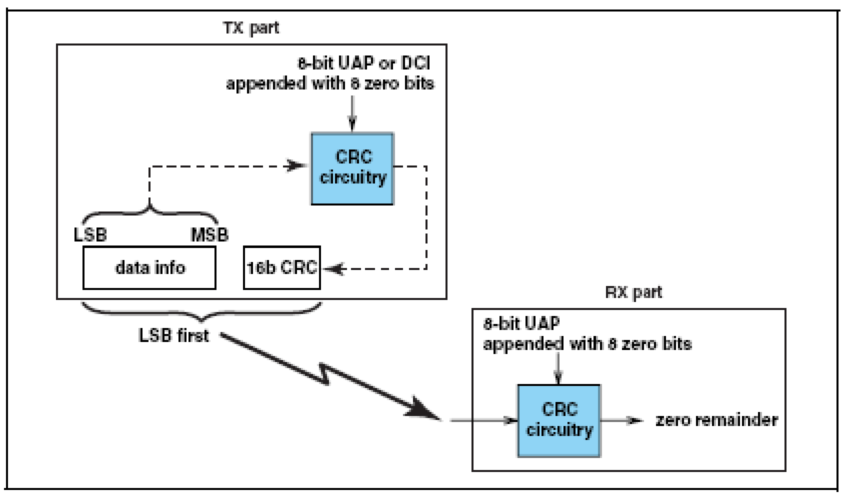

Figure 6.3 CRC generation and checking

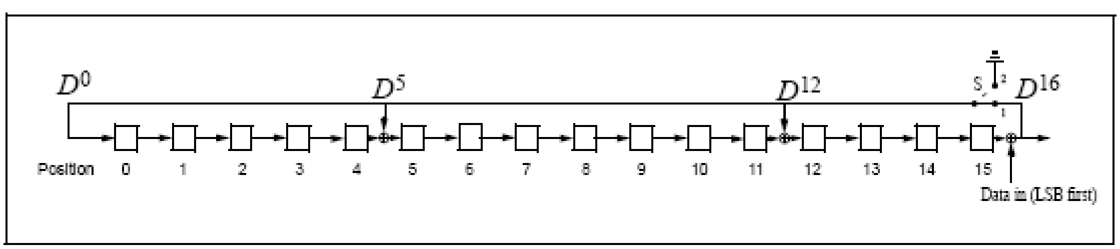

Figure 6.4 LFSR of CRC

Appendix B

Forward Error Control (FEC)

The purpose of the FEC is to reduce the number of retransmissions. However, it will add overhead time during transmission.

1/3 FEC

1/3 FEC used for header.3 times repetition of each bit in header form the 1/3 FEC code. The 1/3 FEC is shown at Figure 6.5.

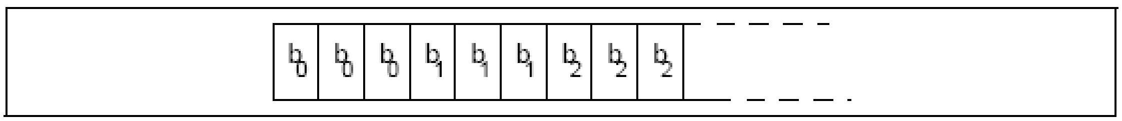

Figure 6.5 Generation of 1/3 FEC

2/3 FEC

2/3 FEC used for payload. Basically, it is a (15, 10) shortened Hamming code. The generator polynomial is $g(D) = (D+1)(D^4 + D + 1)$ .see Figure 6.6.

Bits in payload are divided into blocks. Each block has 10 information bits which can be encoded into a 15 bit code word by appending a 5 bit remainder in tail.

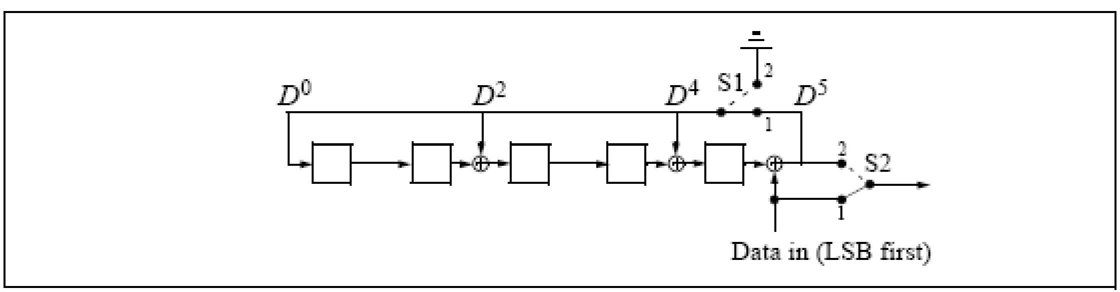

Figure 6.6 LFSR of 2/3 FEC

Reference

[1] Specification of the Bluetooth System V1.0-Core, http://www.bluetooth.com

[2] Specification of the Bluetooth System V2.0+EDR-Core, http://www.bluetooth.com

[3] John B.Anderson, “Digital Transmission Engineering”.1998

[4] Dake liu, “Design of Embedded DSP processors”.2004

[5] Ying-chi Kuo , “Implementation of Bluetooth Baseband Behavioral Model in C language”, Master thesis, linköping 2004.

[6] V.S.lin,A.A Ghoneim and P.A.Dafesh, “Implementation of reconfigurable Software Radio for Multiple Wireless Standards”, 2004 IEEE aerospace conference proceedings.