



1-1-2020

A 88 \$\mu W\$ digital phase-domain GFSK demodulator compatible with low-IF and zero-IF receiver with preamble detection for BLE

SHEN JIE UNG

AB AL-HADI AB RAHMAN

Follow this and additional works at: <https://journals.tubitak.gov.tr/elektrik>

Part of the Computer Engineering Commons, Computer Sciences Commons, and the Electrical and Computer Engineering Commons

Recommended Citation

UNG, S. J., & RAHMAN, A. A (2020). A 88 \$\mu W\$ digital phase-domain GFSK demodulator compatible with low-IF and zero-IF receiver with preamble detection for BLE. *Turkish Journal of Electrical Engineering and Computer Sciences* 28 (4): 2183-2199. <https://doi.org/10.3906/elk-1912-155>



A 88 μ W digital phase-domain GFSK demodulator compatible with low-IF and zero-IF receiver with preamble detection for BLE

Shen Jie UNG^{*}, Ab Al-Hadi AB RAHMAN

School of Electrical Engineering, Universiti Teknologi Malaysia, Johor Bahru, Malaysia

Received: 25.12.2019

Accepted/Published Online: 03.04.2020

Final Version: 29.07.2020

Abstract: Unlike conventional analog-to-digital converter (ADC), phase-domain ADC (Ph-ADC) is more power efficient for the implementation of fully digital Gaussian frequency shift keying (GFSK) demodulator in bluetooth low energy (BLE). Besides, Ph-ADC based demodulator is flexible to pair with low-IF and zero-IF receiver, opposed to limiter based demodulator that work with low-IF receiver only. Yet, currently reported Ph-ADC based demodulator lack of preamble detection for BLE which will be used as symbol clock synchronization. In this work, a Ph-ADC based demodulator is proposed with the feature of preamble detection on BLE's packet. The detected preamble is used for symbol clock recovery and compensation of carrier frequency offset in a BLE packet. Besides, the proposed demodulator is flexible to demodulate IF or baseband signal by simply configuring a parameter value. Using MATLAB, minimum signal-to-noise Ratio (SNR) needed to demodulate BLE packet is estimated using Monte Carlo simulation with 99% confidence level. For hardware implementation, the proposed demodulator is implemented at RTL in Synopsys and its layout is generated using 0.18 μ m CMOS technology. To understand the trade-off between power consumption, layout size and minimum SNR needed, the proposed Ph-ADC based demodulator is scaled to a different combination of 4-bit to 6-bit resolution and 2 MHz to 16 MHz sampling rate. Configuration with the best trade-off for the proposed Ph-ADC demodulator can achieve bit error rate (*BER*) of 0.1% at SNR of 12.5 dB and able to tolerate carrier frequency offset of \pm 200 kHz while using only half the power needed by state of the art limiter based demodulator.

Key words: Bluetooth, BLE, GFSK, demodulator

1. Introduction

Bluetooth low energy (BLE) is a low power wireless standard aimed to be powered by small battery for months while exchanging data at short distance [1]. Operating at 2.4 GHz ISM radio band, BLE uses Gaussian frequency shift keying (GFSK) modulation with a data rate of 1 Mbit/s [2]. A general building block in a BLE receiver is presented in Figure 1a. Firstly, Bluetooth RF signal received at the antenna is downconverted to IF or baseband signal based on receiver radio architecture. The downconverted signal will be demodulated into binary bits for Link Layer, the controller of BLE that ensures data transmission according to BLE's protocol and communicates with the host.

To achieve low power, RF downconversion is commonly implemented using low-IF and zero-IF architecture as they can be integrated on-chip [3]. Zero-IF architecture can achieve lower power than the low-IF receiver by using a lesser number of components while its downconverted baseband signal can be processed at the lowest possible frequency [4]. However, low-IF receiver does not have the problem of DC offset and flicker noise in

*Correspondence: ungsj93@gmail.com

zero-IF receiver which makes it one of the choices for low power application as well [5]. However, both low-IF and zero-IF receiver require different topologies of demodulator as shown in Figures 1b–1d.

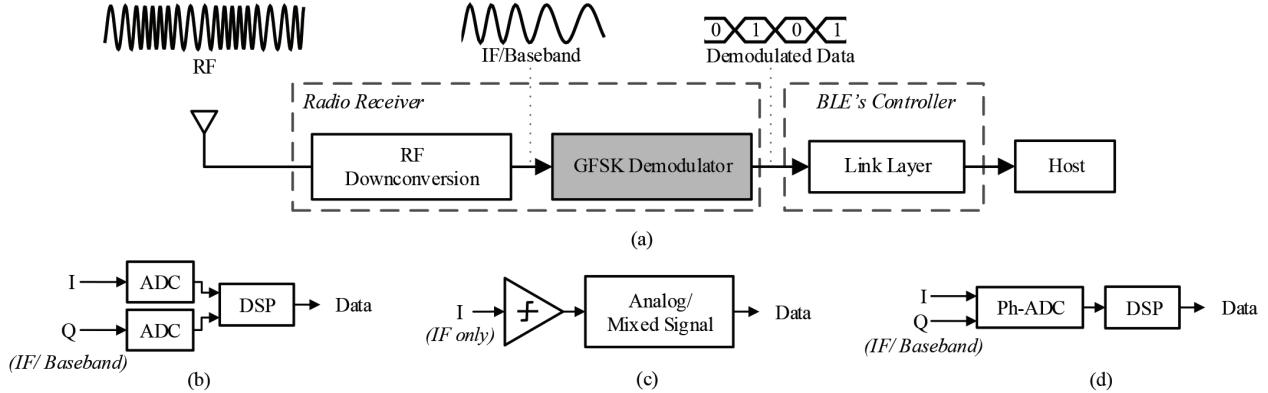


Figure 1. (a) System level overview of bluetooth receiver. (b) Conventional ADC with DSP demodulator. (c) Limiter based demodulator. (d) Ph-ADC based demodulator.

Conventionally, two analog-to-digital converter (ADC) are needed to digitize quadrature signal (I/Q) followed by digital signal processing (DSP) as shown in Figure 1 (b). ADC based demodulator can process both IF or baseband signal but the requirements of two multi-bit ADC and inherent automatic gain control (AGC) take up a considerable amount of layout size and power [6]. To counter that, limiter based GFSK demodulator in Figure 1c uses a hard limiter which transforms analog signal into a single line pulse while the demodulator is implemented through analog/mixed-signal circuitry.

Limiter based demodulator utilizes multiple zero-crossing pulses throughout a symbol period for demodulation. For example, zero crossing detector (ZCD) generates a pulse when the clipped IF signal crosses zero level [7, 8]. The generated pulse is followed by low pass filtering that results in demodulated data. Other limiter technique such as delay-locked loop (DLL) delays clipped IF signal to use as sampling clock in a closed-loop system [9–11]. Other technique such as time-to-digital converter (TDC) uses series of coarse and fine delay line to generate sampling clock to track signal's period difference [12–14]. On the other side, quadrature frequency discriminator (QFD) mixes clipped IF signal with its 90 degrees delayed signal followed by low pass filter to remove high-frequency component [6, 15]. To summarize, limiter based demodulator uses a lesser component than ADC based demodulator but can only work with low-IF receiver.

Alternative to ADC and limiter based demodulator, phase-domain ADC (Ph-ADC) in Figure 1d converts analog I/Q signal into digital phase and its signal is demodulated using DSP. Comparing Ph-ADC implementation and conventional ADC, Ph-ADC requires only single quantization instead of two. At the same time, Ph-ADC is less sensitive towards vector-magnitude variation where the design requirement of AGC can be relaxed [16]. Although Ph-ADC also requires multi-bit quantization, recently reported Ph-ADC can achieve very low power, as low as $12.9 \mu W$ [17]. Comparing to limiter based demodulator, Ph-ADC based demodulator offers more flexibility as it can pair with zero-IF and low-IF receiver. Besides, Ph-ADC based demodulator can be implemented in a fully digital manner which is less sensitive to process voltage and temperature (PVT) variation compared to analog/mixed-signal circuitry [18].

However, currently reported Ph-ADC based demodulator is implemented using simple integrate and dump method only. Symbol clock for bit slicing is assumed to be provided and carrier frequency offset compensation

is not included in reported Ph-ADC based demodulator [19, 20]. Utilization of preamble in BLE packet for symbol clock recovery after Ph-ADC conversion yet to be discussed in the literature. On the other hand, recently reported Ph-ADC of different topologies is optimized for a resolution of 4 to 5 bit with a wide range of sampling rate [19, 20]. More understanding between resolution, sampling rate, power, layout size and minimum SNR performance needed for Ph-ADC based demodulator when preamble detection included can be studied, hence able to select optimum Ph-ADC configuration and topologies that suit BLE requirement.

This paper presents a flexible GFSK demodulator after Ph-ADC conversion. It can demodulate the signal from zero-IF and low-IF receiver by changing a controlled variable if the incoming IF frequency is known. Besides, the proposed Ph-ADC based GFSK demodulator includes preamble detection of BLE that is used to recover symbol clock and calculates carrier frequency offset in a data packet. At MATLAB, the proposed demodulator is tested for its minimum SNR needed to meet BLE's *BERs* specification of 0.1% when paired with Ph-ADC of different sampling rate and resolution combination using Monte Carlo simulation. From a hardware perspective, the proposed GFSK demodulator is simulated using $0.18 \mu\text{m}$ CMOS technology at Synopsys. Trade-off between demodulation performance and hardware resource consumption across 12 combinations of sampling frequency and resolution of Ph-ADC is studied. Lastly, state of the art limiter based GFSK demodulator is compared against this work's Ph-ADC based demodulator.

2. Proposed Ph-ADC based demodulator

Figure 2 presents the top-level view of proposed Ph-ADC based GFSK demodulator. The input of proposed Ph-ADC based demodulator is digital phase sample, $\phi[n]$ with a resolution of n_{ADC} bit at a sampling rate of f_s converted from analog $I(t)/Q(t)$ signal through Ph-ADC. Ph-ADC based GFSK demodulator is also interfacing with link layer, the host of BLE's physical layer which instructs to start/stop demodulator operation through the control signal. The first step of proposed Ph-ADC based GFSK demodulator is noise filtering and compensation of IF signal (for IF input). Then, noise filtered signal, $\phi_{MA}[n]$ is constantly checked for preamble. Once the preamble is detected, the symbol clock is recovered and is used at bit slicing module. At the same time, computation of carrier frequency offset is done concurrently using previously cached peak and valley amplitude $PV_{0...3}$. Finally, the sliced bit is synchronized in frame and demodulated data is sent to Link Layer together with the "data ready" signal for data strobing.

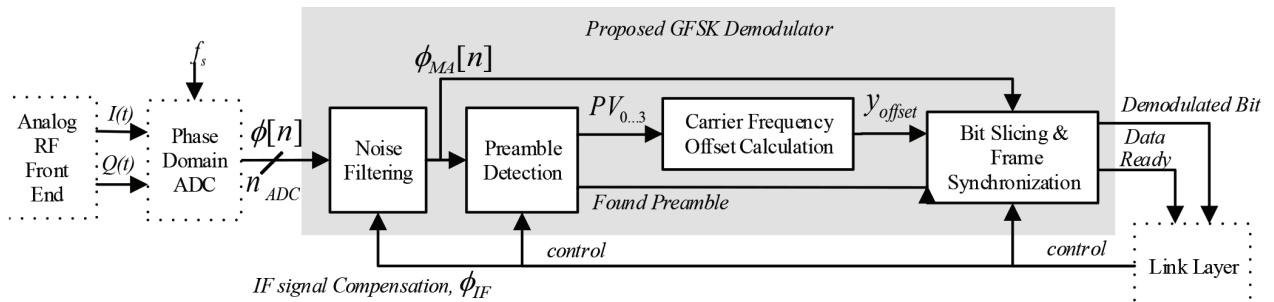


Figure 2. Overview of proposed Ph-ADC based GFSK demodulator interfacing with Ph-ADC and Link Layer of BLE.

2.1. Phase domain input

This section presents phase domain input signal used by the proposed Ph-ADC based GFSK demodulator. Firstly, Ph-ADC converts analog $I(t)$ and $Q(t)$ signal from RF front end into analog phase signal of $\phi(t)$ by

$\tan^{-1}[Q(t)/I(t)]$. Next, sample-and-hold circuit converts analog phase $\phi(t)$ into digital sample, $\phi[n]$ where n is the n^{th} digital sample with resolution of n_{ADC} bit at sampling rate of f_s . For example, 4-bit Ph-ADC divides the unit circle into 16 steps with a minimum phase step of 22.5° as shown in Figure 3 (a). The minimum phase step in degree where Ph-ADC is capable to convert, ϕ_{min} is presented in Equation (1) whereby n_{ADC} is the resolution of Ph-ADC.

$$\phi_{min} = \frac{360^\circ}{2^{n_{ADC}}} \quad (1)$$

Figure 3b depicts phasor rotation in a unit circle with the modulation index of 0.5 and bit rate of 1 Mbit/s. At $t = 0 \mu s$, assuming the phasor begins at positive x-axis of a unit circle. If downconverted signal used for phase conversion is baseband and modulated data is “1”, the phasor rotates by $+90^\circ$ across a bit period of $1 \mu s$ as shown in case (1) of Figure 3b, vice versa for modulated data of “0” as shown in case (2) of Figure 3b. Phasor rotates by $+90^\circ$ because carrier frequency used to represent “0” and “1” is orthogonal to each other when the modulation index is 0.5 [21]. If the modulated signal is IF, phasor rotates further by a constant value on top of rotation due to modulated data. For example, when IF signal is 1 MHz and modulated data is “1”, the phasor rotates by $+450^\circ$ instead of $+90^\circ$ across a bit period. The extra amount of rotation due to carrier signal in degree, θ_{IF} can be calculated as below:

$$\frac{\theta_{IF}}{f_{IF}} = \frac{90^\circ}{\Delta f}, \quad (2)$$

where f_{IF} is the IF frequency. The $+90^\circ$ is the amount of phasor rotation across a bit period with the modulation index of 0.5 while the Δf is peak frequency deviation which is related to the modulation index, $m = (2 \cdot \Delta f)/f_m$, where f_m represents the bit rate [21].

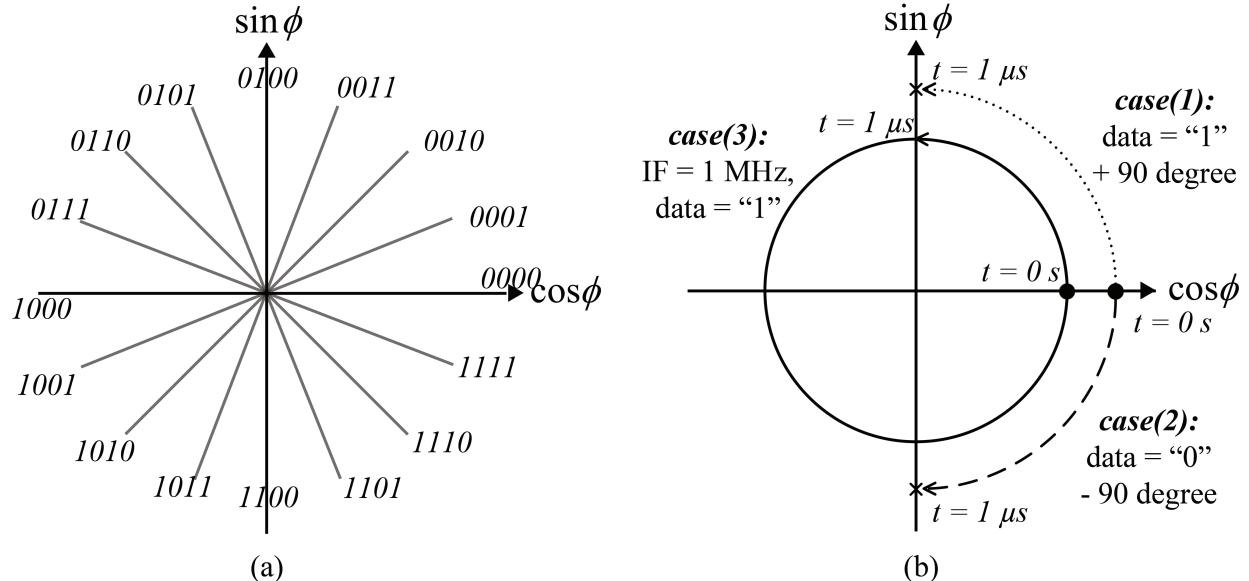


Figure 3. (a) Example of encoded phase signal by 4-bit Ph-ADC. (b) Example of phasor rotation in a unit circle.

2.2. Noise filtering

Implementation of noise filtering module consists of phase difference conversion, phase unwrap and cascaded moving average filter as shown in Figure 4. Phase difference conversion is needed to convert the extra rotation of IF signal into constant DC offset. The DC offset in quantized phase domain can be presented as in Equation (3) where ϕ_{min} is the minimum phase step of Ph-ADC in degree while n_{samp} is the number of sample across a bit period. θ_{IF} is the extra amount of rotation due to IF signal in degree which can be calculated from Equation (2). To compensate DC offset due to IF signal, ϕ_{IF} is deducted at phase difference domain represented as $\phi_d[n]$, where the phase difference between consecutive phase sample is calculated as shown in Equation (4). Hardware implementation of phase difference and IF signal compensation is implemented by 2 shift registers and adder as shown in phase difference submodule of Figure 4.

$$\phi_{IF} = \frac{\theta_{IF}}{\phi_{min} \cdot n_{samp}} \quad (3)$$

$$\phi_d[n] = \phi[n] - \phi[n - 1] - \phi_{IF} \quad (4)$$

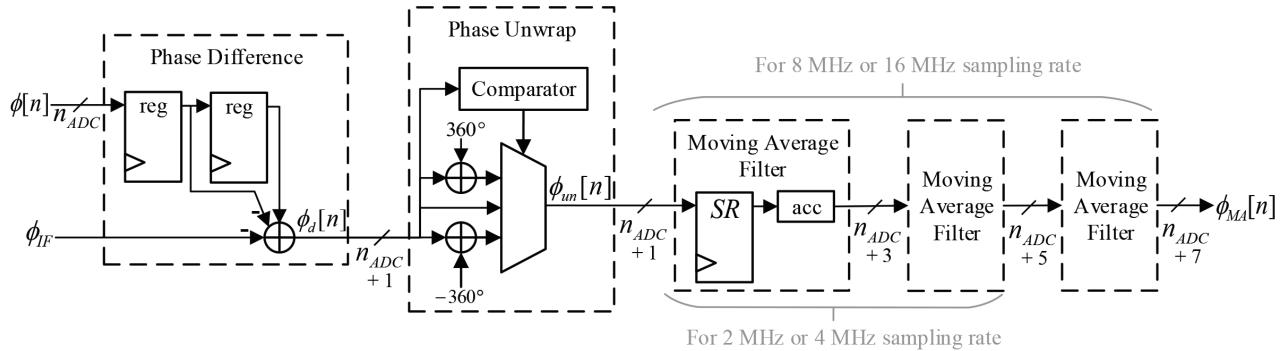


Figure 4. Hardware architecture of noise filtering module which consists of phase difference conversion, phase unwrapping and cascaded moving average filters.

After phase difference conversion, phase has to be unwrapped due to the sharp transition of phase value when phasor rotates across the positive x-axis in a unit circle. This is because encoded phase value by Ph-ADC is largest at 4th quadrant while smallest in the 1st quadrant. If a sharp transition of phase is not unwrapped and filtered directly, actual information of the modulated signal is lost as moving average filter smoothed out the sharp transition. In this work, the phase is unwrapped by deducting phase difference at n_{th} sample, $\phi_d[n]$ by 360° if the phase difference sample is larger than 180° , vice versa for the opposite direction of phase rotation. The implementation is as shown in Equation (5) where $(2^{n_{ADC}} - 1)$ is 360° representation in quantized form of $2^{n_{ADC}}$ bit while $\phi_d[n] \cdot \phi_{min}$ is the degree representation of phase at n_{th} sample. Hardware implementation is made up of a comparator and multiplexor that check for conditions in Equation (5) as shown in phase unwrap submodule of Figure 4.

$$\phi_{un}[n] = \begin{cases} \phi_d[n] - (2^{n_{ADC}} - 1) & , \phi_d[n] \cdot \phi_{min} > 180^\circ \\ \phi_d[n] + (2^{n_{ADC}} - 1) & , \phi_d[n] \cdot \phi_{min} < -180^\circ \\ \phi_d[n] & , others \end{cases} \quad (5)$$

After phase unwrapping, moving average filter is selected to filter noise in this work because it can be implemented efficiently in hardware by using simple adder while maintaining sharp step response [22]. The moving average filter can be presented as in Equation (6) where k is the number of taps in moving average filter. In other words, k number of consecutive samples are averaged to produce the filtered signal $\phi_{MA}[n]$.

$$\phi_{MA}[n] = \frac{1}{k} \cdot \sum_{i=0}^{k-1} \phi[n - i] \quad (6)$$

For better stopband attenuation, this work cascades each k -tap of moving average into multiple passes of moving average filter where multiple pass of moving average filter changes the impulse response from square to a triangle and gaussian shape [22]. In this work, the number of tap in moving average filter, k and the number of cascaded moving average are selected to best fit the performance of preamble detection algorithm where the priority is to keep the phase signal smooth without any spike throughout the preamble signal. In this work, simulation result shows that the optimum number of moving average pass for 2 and 4 MHz sampling rate is 2-pass moving average filter while 8 and 16 MHz sampling rate is 3-pass moving average filter.

2.3. Preamble detection

After noise filtering, the preamble detection module searches the existence of preamble in a signal before bit slicing. The preamble detection module is divided into peak detection and preamble check logic submodule as shown in Figure 5. Peak detection is constantly detecting peak and valley in the filtered signal while preamble check logic identifies if the detected peak/valley resembles preamble. In this work, the peak/valley is detected by comparing the change in sign of the signal's gradient between the consecutive sample of $\phi_{MA}[n]$ and $\phi_{MA}[n+1]$. Peak/valley detection can be represented using Equation (7), where value of F_{pk} can be decoded as “1” = peak, “-1” = valley and “0” = nothing while a and b are the differences in magnitude between 3 consecutive samples as shown in Equation (9). Hardware implementation of peak detection only requires 2 shift registers, 2 adders, comparator and multiplexor implemented in combinational.

$$F_{pk} = \begin{cases} 1 & , a > 0, b \leq 0 \\ -1 & , a < 0, b \geq 0 \\ 0 & , others \end{cases} \quad (7)$$

$$\begin{aligned} a &= \phi_{MA}[n] - \phi_{MA}[n - 1] \\ b &= \phi_{MA}[n + 1] - \phi_{MA}[n] \end{aligned} \quad (8)$$

Whenever a new peak/valley is found at peak detection submodule, F_{pk} triggers the controller of preamble check logic to identify if preamble exists. Figure 5 shows the criteria and checking sequence to identify the existence of preamble. Alternating sign between detected peak/valley signal is used as the first rule of checking because alternating “1” and “0” in the preamble would result in alternating peak/valley. As BLE's specification requires the tolerance of $\pm 1/8$ bit period of symbol timing error, this work specifies the time spacing between detected peak/valley with more error margin by setting it to be $\pm 1/4$ of a bit period. To further differentiate between noise and preamble, the peak-to-valley between consecutive peak/valley has to be larger than a threshold to be qualified as a preamble. In this work, the threshold is selected as 90° which is equivalent to half of the phasor rotation angle in a unit circle due to alternating “1” and “0”. Representation of

90° can be converted to binary representation by dividing ϕ_{min} . Lastly, aforementioned rules/sequence has to be fulfilled consecutively for 4-bit period to confirm preamble is detected. Although preamble consists of 8-bits, consecutive 4-bit period is chosen to confirm the existence of preamble to provide extra error margin in case the first two-bit of preamble is corrupted. Since there will be frame synchronization using the access address of data packet, early detection of the preamble will not cause false detection.

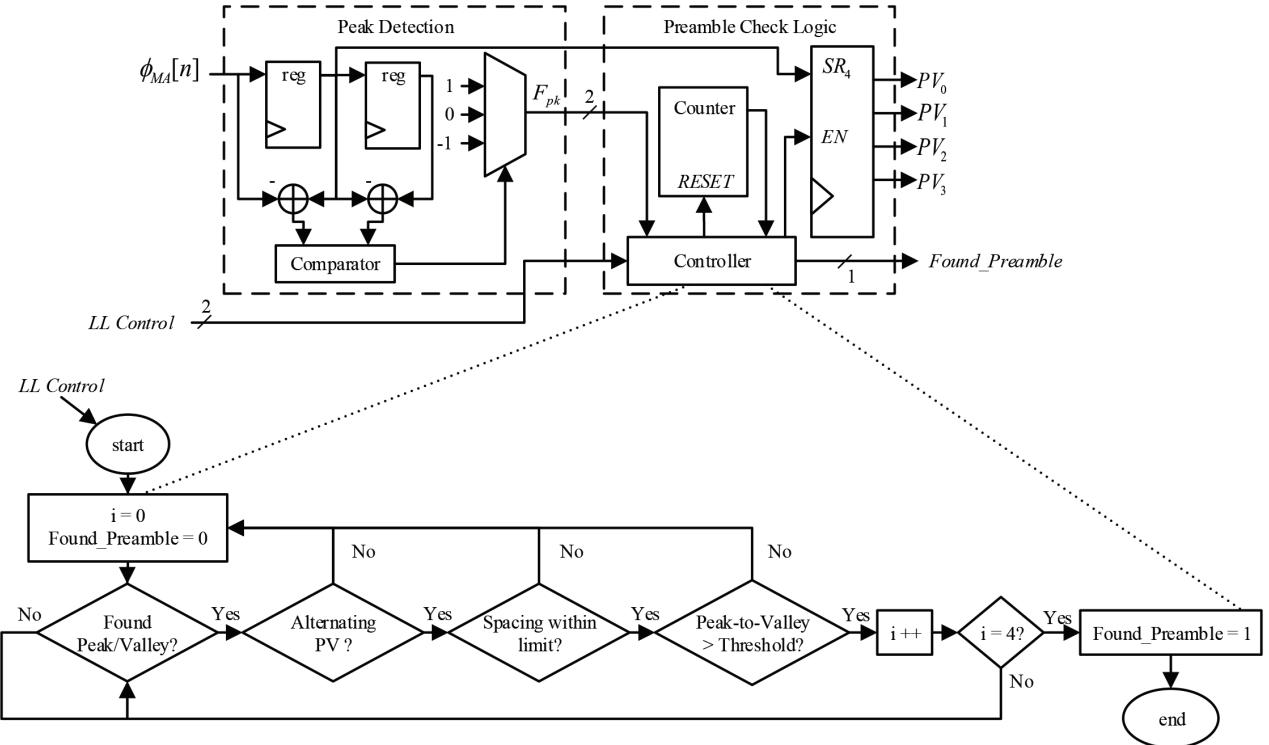


Figure 5. Hardware architecture of preamble detection that consists of peak/valley detection and preamble check logic.

Hardware implementation of preamble check logic is as shown in Figure 5. The controller is implemented using mealy state machine where it accepts an instruction from link layer regarding when to start/stop looking for preamble and notify the next module when preamble is found. The counter is used for tracking the time spacing between peak and valley as well as the total number of times valid peak/valley had occurred. On the other hand, for every valid peak/valley from PV_0 to PV_3 , they are stored in 4 shift registers for carrier frequency offset calculation at the following module.

2.4. Frequency offset compensation, bit slicing and frame synchronization

After preamble detection is confirmed, PV_0 to PV_3 will be used to calculate DC offset caused by carrier frequency offset. Figure 6a illustrates the calculation of estimated DC offset, y_{off} which is defined as the average of PV_0 to PV_3 as shown in Equation (9):

$$y_{off} = \frac{PV_0 + PV_1 + PV_2 + PV_3}{4} \quad (9)$$

Since the last peak/valley detected during preamble detection indicates the end or beginning of a symbol period, a counter will be started to keep track of symbol clock timing and reset every $1 \mu s$. At every $1 \mu s$, bit slicing decides the demodulated bit as “1” if the amplitude of signal is larger than zero, vice versa for demodulated bit of “0”. To synchronize the frame, the expected access address is constantly checked against with new demodulated data at every symbol clock Figure 6b. Once the expected access address matches, the first bit of payload will be on the next symbol clock and the frame is considered synchronized.

Hardware implementation of frequency offset compensation, bit slicing and frame synchronization is as shown in Figure 6c. Calculation of DC offset due to carrier frequency offset, y_{off} is implemented using adder tree while averaging by 4 is simply shifting of 2 bits to the right. The DC offset y_{off} is subtracted from the filtered phase sample before bit slicing decision on $\phi_{MA}[n]$. The controller in this module is a state machine that facilitates start, stop operation which is controlled by link layer. “Found preamble” signal also gives instruction to state machine for triggering bit slicing and look for access address. Once the access address is found, the state machine moves to another state to output demodulated bit to link layer.

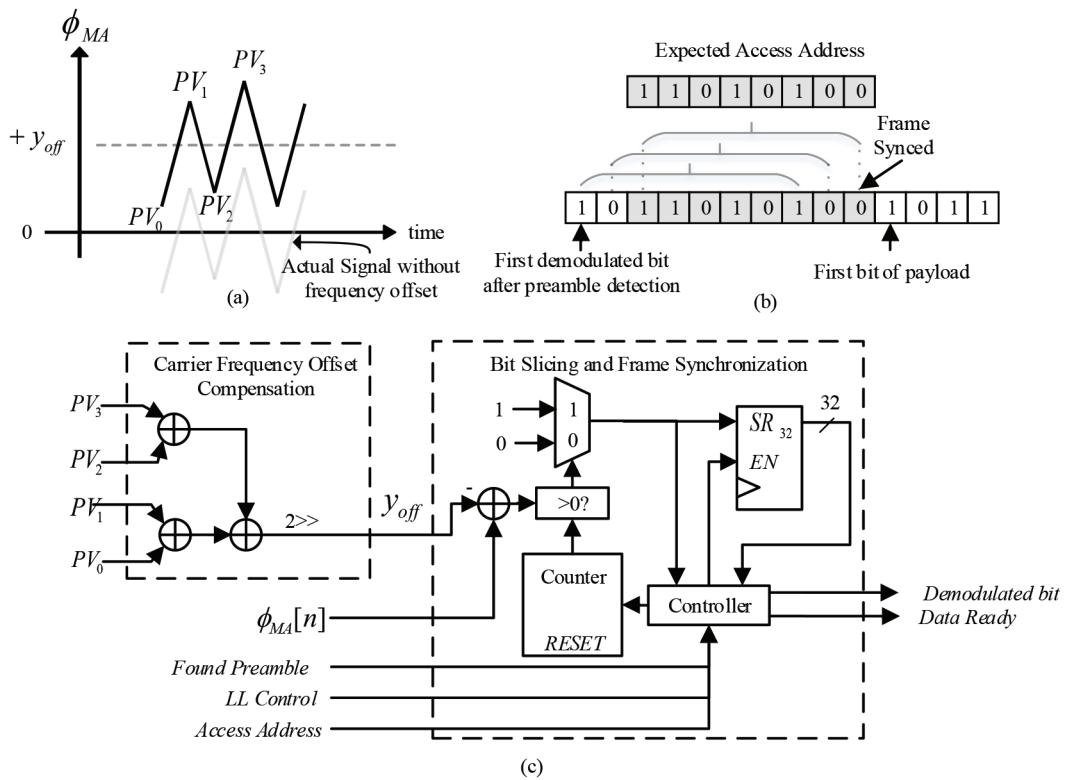


Figure 6. (a) Illustration of carrier frequency offset compensation. (b) Illustration of frame synchronization. (c) Hardware implementation of frequency offset compensation, bit slicing and frame synchronization.

3. Results and discussion

The first part of this section presents an example of a demodulation process using proposed Ph-ADC demodulator. Input signal starting from quantized phase to signal processing and finally, the output of demodulated bit is shown in MATLAB and Synopsys. Next, the performance and resource usage of demodulator are presented. Methodology used for evaluation is explained and the trade-off between a few performance metrics is presented.

Lastly, this section compares this work's Ph-ADC based demodulator with currently reported limiter based demodulator.

3.1. Example of demodulation

For illustration purpose, Figure 7 depicts noise filtering using 4-bit 4 MHz demodulator with an input signal of 25 dB and +200 kHz carrier frequency offset. From $t = 0 \mu s$ to $7 \mu s$ signal consists of noise while $t > 7 \mu s$ onward consists of modulated BLE data packet. It can be seen that quantized phase of baseband in Figure 7a changes slower compared to quantized phase of IF signal in Figure 7d. However, after IF compensation, phase unwrapping and moving average filtered, both signal looks almost identical in Figures 7c and 7f.

The demodulation process after noise filtering is as shown in Figure 7g. From $t = 0 \mu s$ to $10.5 \mu s$, peak/valley detection is constantly tracked until preamble is confirmed to exists at $t = 10.5 \mu s$. The entire waveform had been shifted downwards after $t = 10.5 \mu s$ due to compensation of DC offset caused by +200 kHz carrier frequency offset. At the same time, bit slicing had begun at every $1 \mu s$ interval where sliced data is first stored in a 32-bit shift register to compare with expected access address which also functions as frame synchronization from $t = 10.5 \mu s$ to $44.5 \mu s$. Once the data frame is synchronized, bit slicing submodule will keep on bit slicing at payload (PDU) region until it receives instruction to stop which is given by the link layer.

Figure 8 shows the postlayout simulation of 4-bit 4 MHz demodulator in Synopsys where timing delay due to parasitic of layout is included in the simulation. Internal control signal of "RESET" and "ENABLE" had been toggling to reset and latch value for internal register and counter. These internal control signal are controlled by local controller as seen in Figure 5 and Figure 6c. The local controller is implemented in mealy state machine which issues a different set of control signal at respective event such as preamble detection, frame synchronizing, data output or a hard reset by link layer to return idle state.

3.2. Demodulation performance and resource usage

To measure the performance of demodulator, BLE stated *BER* shall be less than or equal to 0.1% while able to tolerate ± 150 kHz frequency offset [2]. Since BLE does not has forward error correction code, any single-bit error at preamble, access address or payload during transmission results in entire packet error. In this work, packet error rate (*PER*) is also used to measure the performance of demodulator because the proposed demodulator included preamble detection. The *PER* of BLE is estimated to be 25.63% when *BER* is 0.1% and 296 bits of payload in each packet [2, 23]. Monte Carlo simulation of 99% confidence level with a sample size of 100 BLE packet is used to estimate the *PER* performance of this work's demodulator. As less than 10 BLE packets are transferred per connection [2], a sample size of 100 BLE packet is sufficient to emulate the actual use case. In each BLE packet, 1000 bits of payload is transferred instead of 296 bit to give an extra margin of accuracy during performance evaluation.

Monte Carlo simulation begins with the generation of input test vector which consists of 1000-bit payload that is generated randomly, padded with preamble and access address to emulate actual data packet of BLE. Then, the BLE data packet is GFSK modulated and added with additive white Gaussian noise (AWGN) to emulate noise during transmission. The noisy GFSK modulated signal is used as the input of proposed GFSK demodulator. If the packet is not detected, whereby error happens at preamble or access address, *PE* counter is incremented. On the other hand, if a packet is detected but the error happens during PDU, *PE* counter is also incremented. Once 100 BLE packet had been tested at i_{th} iteration, the *PER* at i_{th} iteration is calculated,

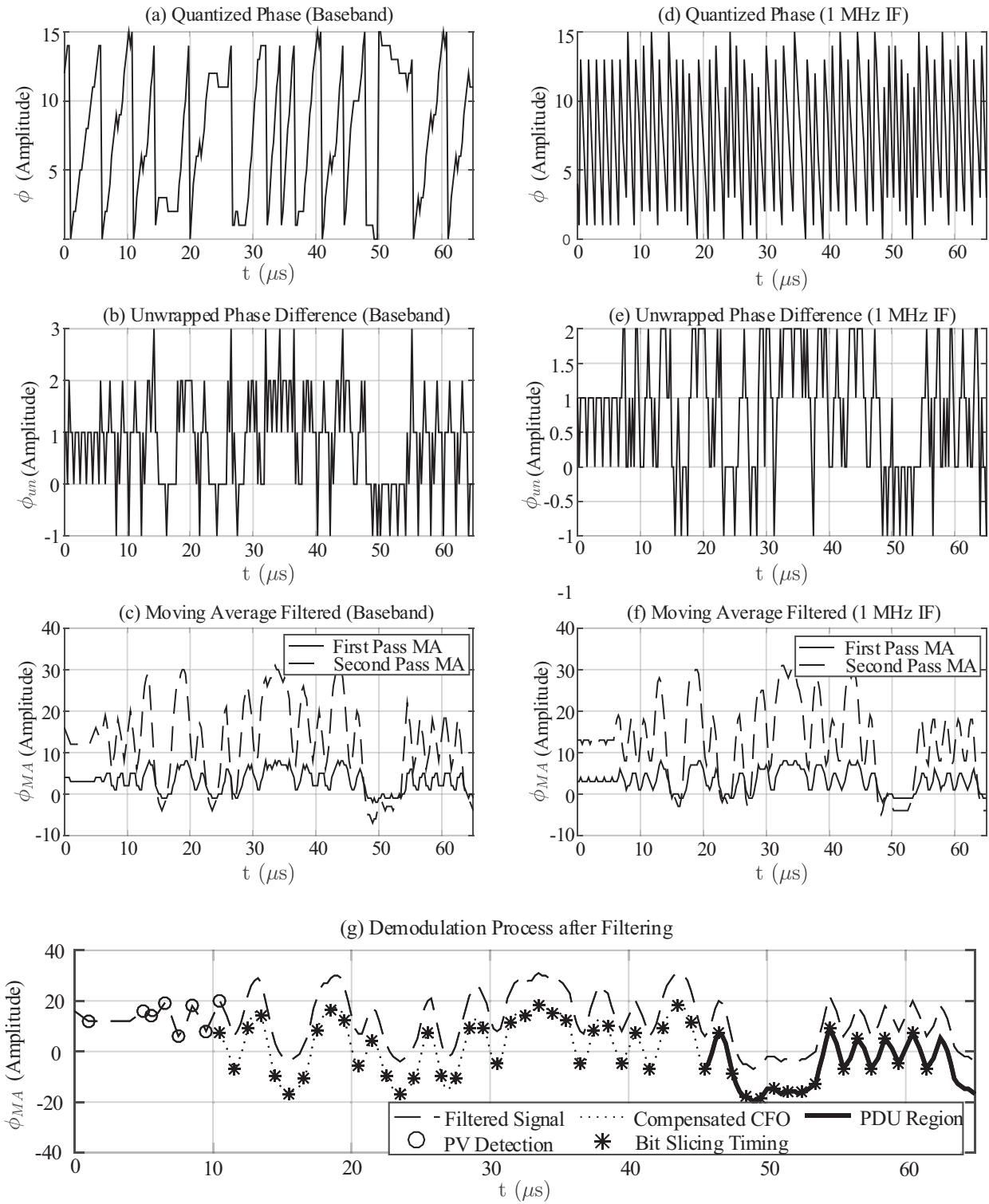


Figure 7. Example of demodulation process from quantized phase to demodulation process of 4-bit 4 MHz demodulator.

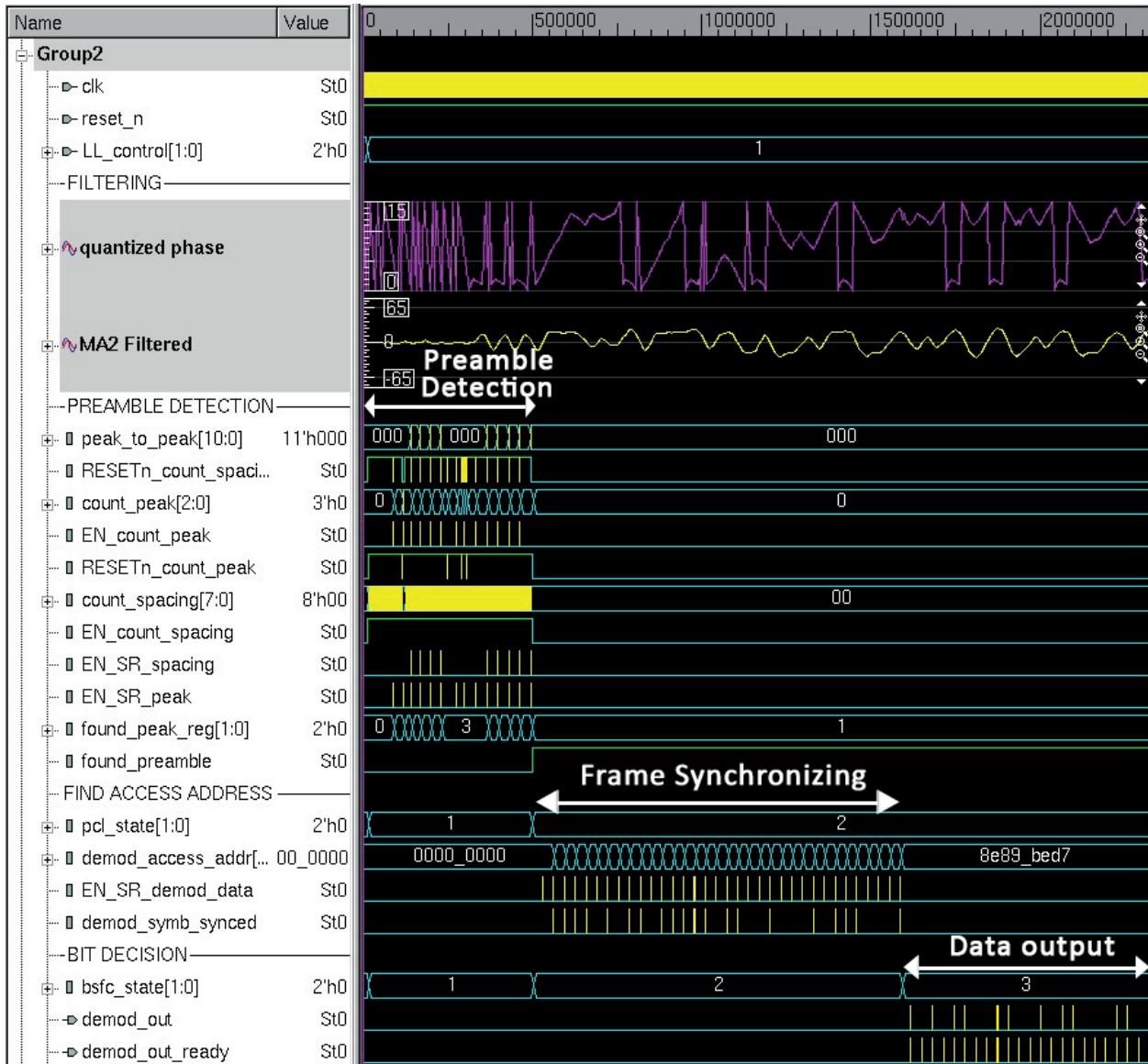


Figure 8. Postlayout simulation on 4-bit 4 MHz demodulator using Synopsys where input test vector is generated in MATLAB.

stored and the entire simulation is restarted again. Until the 100th iteration, the upper, lower bound and mean of estimated *PER* is then calculated by using *PER* of 100_{th} iteration.

Figure 9 shows the performance of demodulator with different resolution and sampling rate over signal with a variable amount of noise represented by E_b/N_o and carrier frequency offset of ± 200 kHz tested using Monte Carlo simulation. As the performance of demodulator is comparable when the input signal is baseband or IF, only results from baseband input is presented in Figure 9. The horizontal dotted line in Figure 9 indicates the minimum *PER* needed to meet *BER* of 0.1%.

The design is implemented using Silterra 1.8 V 0.18 μm CMOS technology and the layout size of demodulator is as shown in Figure 10. No I/O pad is included in the generated layout as the demodulator

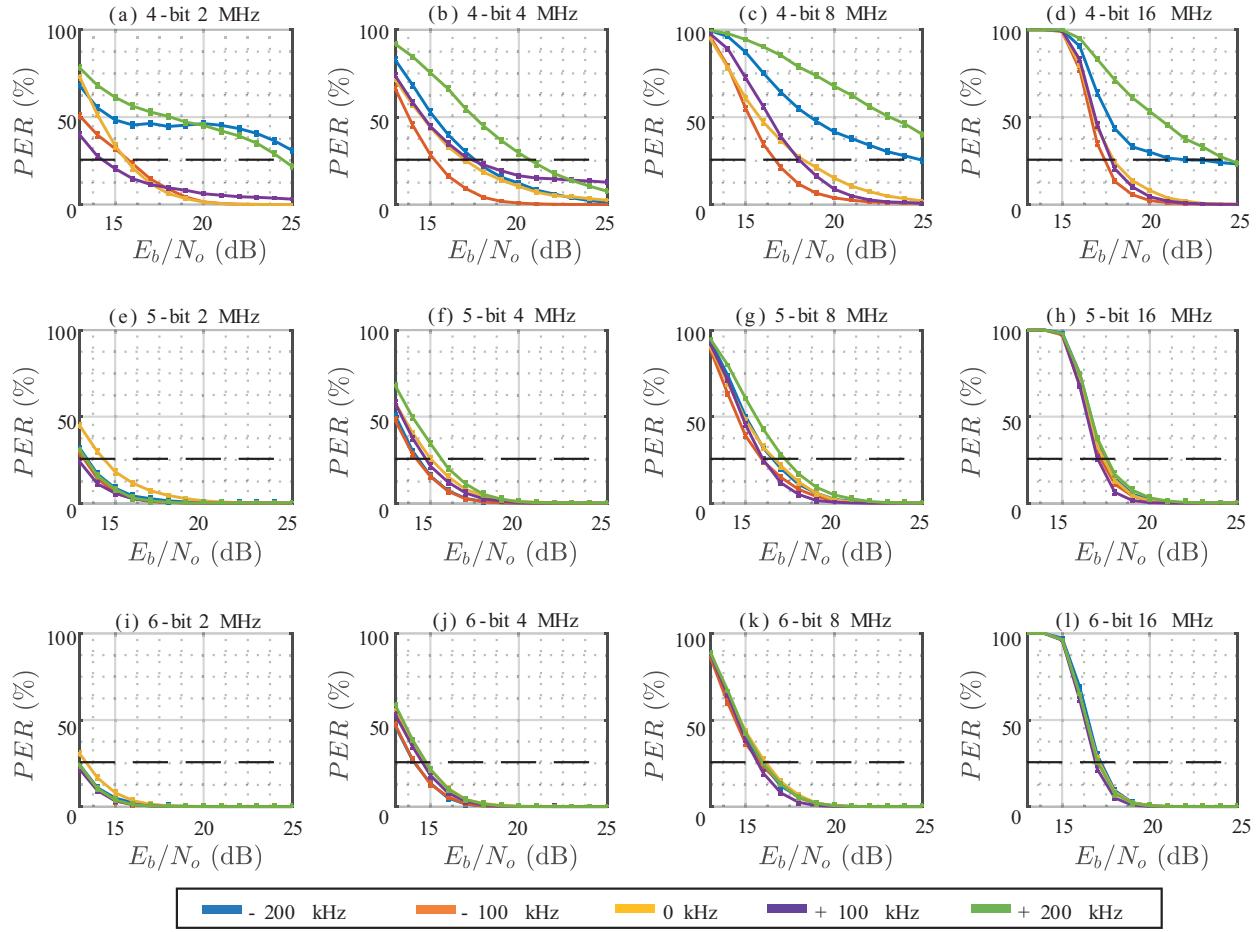


Figure 9. Estimated PER performance using Monte Carlo simulation of 99% confidence level over input signal with variable noise of E_b/N_o and ± 200 kHz.

does not interface with external signal directly, in fact, it will be integrated with link layer instead. As seen in Figure 10, each demodulator is routed with 6×6 power grid to minimize the IR drop at the centre of the chip. All of the layouts are routed with core optimization of 80 % to prevent global route congestion that could possibly result in signal integrity issues.

The minimum E_b/N_o needed to meet BLE requirement is plotted against the power consumption and layout size in Figure 11. Based on the performance and resource trade-off, it can be seen that doubling of sampling rate increases both power consumption and layout size exponentially. This is because a higher sampling rate used by demodulator requires a higher number of cascading moving average stages which increases the resource usage. Besides higher resource consumption, higher clock rate increases dynamic switching power of registers, the basic component used by moving average filter to store samples. With these 2 factors combined, the power consumption and layout size increase exponentially when sampling rate double.

On the other hand, increasing the resolution from 4-bit to 5-bit improves the minimum E_b/N_o needed to demodulate BLE data packet effectively without significant increase of power consumption and layout size. Further increase of resolution from 5-bit to 6-bit offers little improvement to the minimum E_b/N_o . To summarize, the best performance trade-off demodulator is 5-bit 2 MHz sampling rate for baseband input signal.

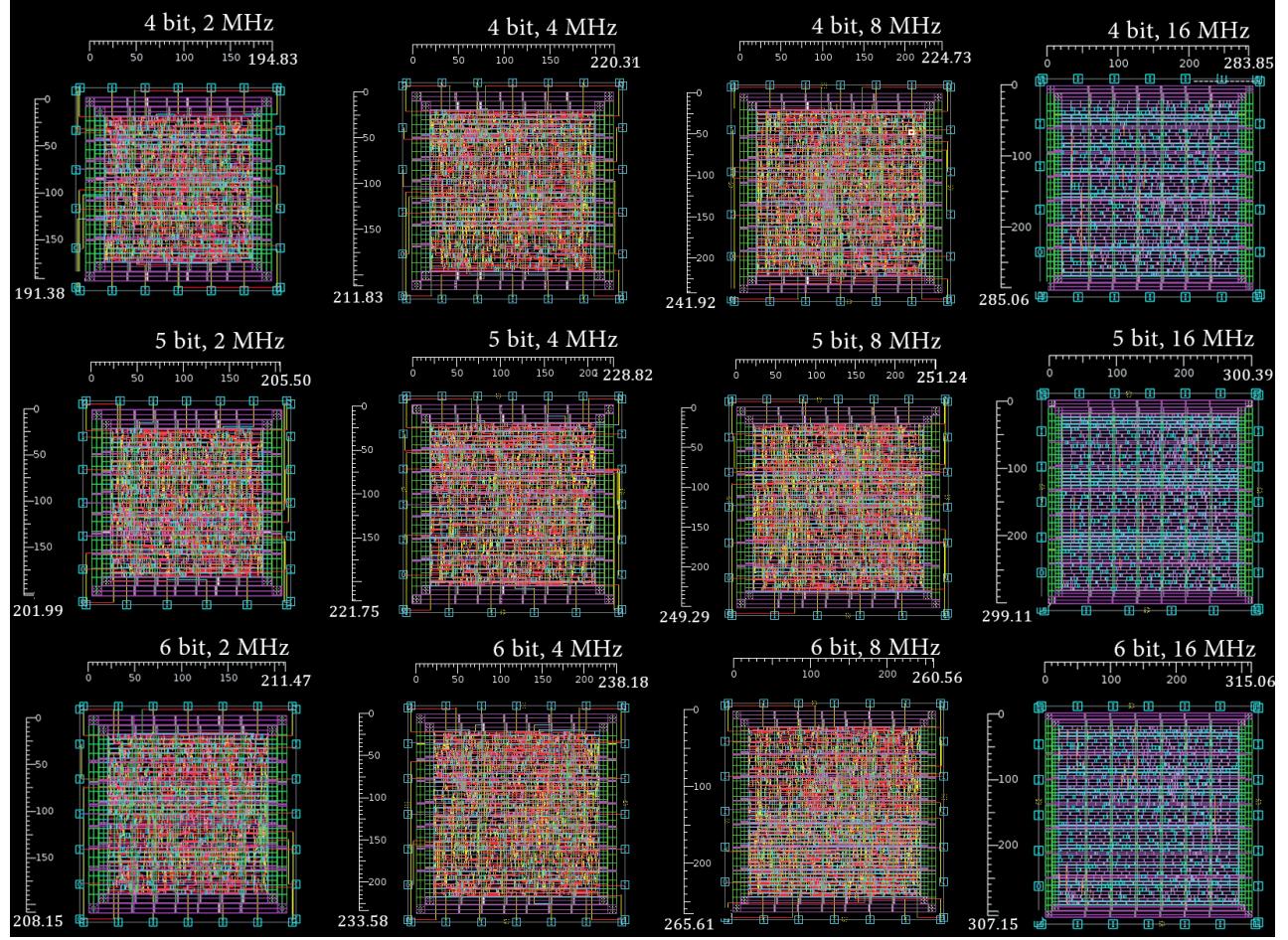


Figure 10. Layout size of each demodulator studied in this work. The image of each demodulator layout is not scaled similarly to each other and the ruler measurement unit is in μm .

If the input signal is 1 MHz IF signal (commonly used for limiter based demodulator), 5-bit 4 MHz sampling rate is needed to prevent aliasing.

3.3. Comparison with limiter based demodulator

Table summarizes a comparison of limiter based demodulator with the proposed Ph-ADC based demodulator in this work. Two configurations of demodulators from this work are selected for comparison where 5-bit 2 MHz sampling rate demodulator is chosen for baseband input signal while 5-bit 4 MHz sampling rate is chosen for input IF signal of 1 MHz. As most of the limiter based demodulator use SNR as the performance metrics, the minimum E_b/N_o needed to meet BLE requirement is converted to SNR using $SNR = E_b/N_o - 10 \log_{10}[0.5 \cdot f_s/f_m]$, where f_s is the sampling rate and f_m is the bit rate.

One of the performance metrics is minimum SNR needed by demodulator to achieve BER of 0.1% whereby the lower the minimum SNR, the better the performance of demodulator. Ph-ADC based demodulator from [19] can demodulate at the SNR of 11.0 dB which is the lowest among all demodulator. This is because [19] uses a sampling rate of 20 MHz which lowers the noise floor within the bandwidth of wanted signal. However,

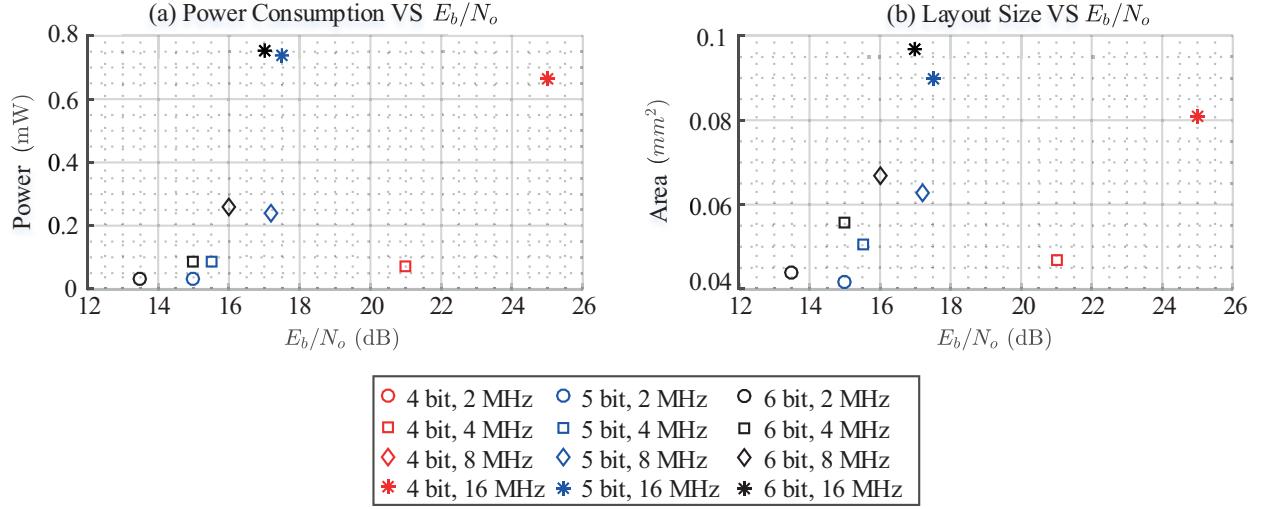


Figure 11. Performance and resource trade off across demodulator with different resolution and sampling rate (Missing demodulator configuration in the graph requires more than 26 dB E_b/N_o).

Table . Summary of performance comparison between limiter based demodulator with Ph-ADC based demodulator proposed in this work.

Work (Year)	Type	Tech (μm)	Voltage (V)	Power (mW)	Area (mm^2)	SNR (dB)	Sampling (MHz)	IF (MHz)	CFO tolerance
[6] (2016)	QFD	0.13	1.2	*0.204	0.048	14.4	32	1	-150 kHz, + 200 kHz
[9] (2015)	DLL	0.18	1.8	0.468	0.140	18.5	N/A	1.5	\pm 180 kHz
[7] (2014)	ZCD	0.18	1.8	0.918	0.080	16.8	24	3	-1500 kHz, +700 kHz
[10] (2012)	DLL	0.18	0.5	0.200	0.360	18.7	N/A	3	\pm 160 kHz
[12] (2009)	TDC	0.18	1.8	4.590	0.260	13.9	N/A	6	\pm 160 kHz
[19] (2012)	Ph-ADC (4-bit)	0.13	1.0	0.190	0.140	11.0	20	0	\pm 170 kHz
This work (Baseband)	Ph-ADC (5-bit)	0.18	1.8	0.030	0.042	15.0	2	0	\pm 200 kHz
This work (IF)	Ph-ADC (5-bit)	0.18	1.8	0.088	0.051	12.5	4	1	\pm 200 kHz

symbol clock synchronization is not included in the design and it is assumed to be provided. In terms of worst SNR performance, DLL demodulator from [9, 10] requires SNR of 18.5 dB and 18.7 dB, respectively. As for this work, the 5-bit 2 MHz sampling rate requires SNR of 15 dB while the 5-bit 4 MHz sampling rate demodulator requires SNR of 12.49 dB. Both configurations of Ph-ADC demodulator proposed are not the best in SNR performance but being the second place for 5-bit 4 MHz sampling rate demodulator among the other demodulator. In terms of CFO tolerance, this work can tolerate \pm 200 kHz which has wider tolerance range than BLE's requirement of \pm 150 kHz. While limiter based demodulator from [7] can tolerate much higher CFO, it is overkill when compared to BLE's requirement. This work can also tolerate more than \pm 200 kHz if the shifted IF frequency is constant and known before transmission.

For power consumption, this work achieves the lowest power compared to limiter based demodulator. Demodulator of this work uses 0.030 mW for 5-bit 2 MHz sampling rate demodulator while 5-bit 4 MHz sampling rate demodulator uses 0.088 mW . For lowest power demodulator among limiter category, DLL technique from [10] still requires 0.2 mW even implemented through low voltage of 0.5 V. This is because the DLL is implemented mainly by analog circuits which is less efficient than digital implementation. For QFD demodulator, it consumes 0.204 mW although implemented fully in digital [6] and smaller CMOS technology node compared to this work. This is because the QFD demodulator has to run at 32 MHz sampling rate for 1 MHz IF signal input while this work only requires 4 MHz sampling rate. The high sampling rate increases dynamic power consumption due to high switching speed of transistor.

Comparing the size of layout, this work's 5-bit 2 MHz sampling rate demodulator has the smallest area which is 0.042 mm^2 . For the 1 MHz IF demodulator using 5-bit 4 MHz sampling rate, the area is comparable to limiter based demodulator of QFD, where QFD is smaller by only 6% to be precise [6]. However, the layout size of this comparison favors the QFD as it is implemented using $0.13 \mu\text{m}$ while this work uses $0.18 \mu\text{m}$.

To conclude, this study can achieve the smallest size and lowest power consumption due to 2 main factors. Firstly, the implementation of this study mainly consists of digital circuits unlike study in [9, 10, 12] that mainly constructed by analog circuits. Secondly, the algorithm of this study can demodulate at a low sampling rate of 2 MHz or 4 MHz which keep the dynamic power consumption low unlike the high sampling rate case from study in [6, 7].

4. Conclusion

This work presented a flexible Ph-ADC based demodulator that can work with zero-IF and low-IF receiver by changing a controlled variable. The proposed Ph-ADC based demodulator features preamble detection that detect peak and valley of preamble signal at phase domain, followed by a series of rule check to identify existence of preamble, to the best of our knowledge this method yet to be reported in Ph-ADC based demodulator. Detected preamble is used for synchronization and its peak and valley is averaged to estimate required compensation for carrier frequency offset. The proposed demodulator is scaled to different combination of resolution and sampling rate from 4-bit to 6-bit and 2 MHz to 16 MHz respectively. With the aid of MATLAB, minimum SNR needed to meet BLE's requirement of each demodulator is estimated using Monte Carlo simulation with confidence level of 99%. At hardware level, the demodulator is implemented using Silterra $0.18 \mu\text{m}$ CMOS technology in Synopsys where layout size and power consumption are measured.

Results shown power consumption and size increase exponentially when sampling rate of demodulator is doubled while improving minimum SNR by approximately 1 dB only. This is due to higher number of cascaded moving average filter requirement of design and increase of dynamic switching power in registers at higher clock rate. On the contrary, increasing resolution from 4-bit to 5-bit improves minimum SNR effectively by approximately 5 dB, while 5-bit to 6-bit increase offers minimal improvement. In short, the proposed demodulator works best at 5-bit and at lowest sampling rate that would not cause aliasing. For input IF signal of 1 MHz, the proposed Ph-ADC demodulator requires minimum SNR of 12.5 dB at 0.088 mW while occupying 0.042 mm^2 area to demodulate BLE packet. Compared to the lowest power limiter based demodulator reported, this result is twice more power efficient while having comparable size. For baseband input signal, the demodulator can achieve lower power at 0.03 mW and smaller size of 0.042 mm^2 with input SNR of 15 dB, making it a low power solution to pair with zero-IF receiver for BLE.

References

- [1] Heydon R. Bluetooth Low Energy: The Developer's Handbook. Upper Saddle River, NJ, USA: Prentice Hall, 2013.
- [2] Bluetooth Special Interest Group. Bluetooth Core 4.0 Specification (Vol. 0), 2010.
- [3] Salamin Y, Pan J, Wang Z, Tang S, Wang J et al. Eliminating the impacts of flicker noise and DC offset in zero-IF architecture pulse compression radars. *IEEE Transactions on Microwave Theory and Techniques* 2014; 62 (4): 879-888. doi: 10.1109/TMTT2014.2307832
- [4] Razavi B. Design considerations for direct-conversion receivers. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 1997; 44 (6): 428-435. doi: 10.1109/82.592569
- [5] Spiridon S. Toward 5G Software Defined Radio Receiver Front-Ends. Tustin, CA, USA: Springer, 2016.
- [6] Pereira MS, Vaz JC, Leme CA, Sousa JT, Freire JC. A $170 \mu\text{A}$ all-digital GFSK demodulator with rejection of low SNR packets for Bluetooth-LE. *IEEE Microwave and Wireless Components Letters* 2016; 26 (6): 452-454. doi: 10.1109/LMWC.2016.2562639
- [7] Yin Y, Yan Y, Wei C, Yang S. A low-power low-cost GFSK demodulator with a robust frequency offset tolerance. *IEEE Transactions on Circuits and Systems II: Express Briefs* 2014; 61 (9): 696-700. doi: 10.1109/TCII.2014.2335429
- [8] Xia B, Xin C, Sheng W, Valero-Lopez AY, Sánchez-Sinencio E. A GFSK demodulator for low-IF Bluetooth receiver. *IEEE Journal of Solid-State Circuits* 2003; 38 (8): 1397-1400. doi: 10.1109/JSSC.2003.814424
- [9] Yang T, Jiang Y, Liu S, Guo G, Yan Y. A low-power CMOS WIA-PA transceiver with a high sensitivity GFSK demodulator. *Journal of Semiconductors* 2015; 36 (6): 1-8. doi: 10.1088/1674-4926/36/6/065005
- [10] Lai CM, Shen MH, Wu YS, Huang PC. A 0.5V GFSK $200 \mu\text{W}$ limiter/demodulator with bulk-driven technique for low-IF Bluetooth. In: Proceedings IEEE Asian Solid-State Circuits Conference; Kobe, Japan; 2012. pp. 321-324. doi: 10.1109/IPEC.2012.6522690
- [11] Byun S. Analysis and verification of DLL-based GFSK demodulator using multiple IF-period delay line. *IEEE Transactions on Circuits and Systems II: Express Briefs* 2016; 64 (1): 6-10. doi: 10.1109/TCSII.2016.2543144
- [12] Chen CP, Yang MJ, Huang HH, Chiang TY, Chen JL et al. A low-power 2.4-GHz CMOS GFSK transceiver with a digital demodulator using time-to-digital conversion. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2009; 56 (12): 1-12. doi: 10.1109/TCSI.2009.2016184
- [13] Kao HS, Yang MJ, Lee TC. A delay-line-based GFSK demodulator for low-IF receivers. *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*; San Francisco, CA, USA; 2007. pp. 88-89. doi: 10.1109/ISSCC.2007.373601
- [14] Cheng TY. High-data-rate time-to-digital converter for GFSK demodulator of low-power RF receivers. *Electronics Letters* 2012; 48 (2): 76-77. doi: 10.1049/el.2011.3565
- [15] Chi B, Yao J, Chiang P, Wang Z. A 0.18-um CMOS GFSK analog front end using a Bessel-based quadrature discriminator with on-chip automatic tuning. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2009; 56 (11): 2498-2510. doi: 10.1109/TCSI.2009.2015728
- [16] Liu Y, Lot R, Hu Y, Serdijn WA. A comparative analysis of phase-domain ADC and amplitude-domain IQ ADC. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2015; 62 (3): 671-679. doi: 10.1109/TCSI.2014.2374852
- [17] Rajabi L, Saberi M, Liu Y, Lotfi R, Serdijn WA. A charge-redistribution phase-domain ADC using an IQ-assisted binary-search algorithm. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2017; 64 (7): 1696-1705. doi: 10.1109/TCSI.2017.2681461
- [18] Svensson C. Towards power centric analog design. *IEEE Circuits and Systems Magazine* 2015; 15 (3): 44-51. doi: 10.1109/MCAS.2015.2450671

- [19] Gao S, Jiang H, Weng Z, Guo Y, Dong J et al. A $7.9 \mu\text{A}$ multi-step phase-domain ADC for GFSK demodulators. *Analog Integrated Circuits and Signal Processing* 2018; 94 (1): 49-63. doi: 10.1007/s10470-017-1081-5
- [20] Masuch J, Delgado-Restituto MA. $190\text{-}\mu\text{W}$ zero-IF GFSK demodulator with a 4-b phase-domain ADC. *IEEE Journal of Solid-State Circuits* 2012; 47 (11): 2796-2806. doi: 10.1109/JSSC.2012.2216211.
- [21] Proakis J, Salehi M. *Digital Communications*, 5th Edition. Pennsylvania, NY, USA: McGraw-Hill, 2007.
- [22] Smith SW. *The Scientist and Engineer's Guide to Digital Signal Processing*. San Diego, CA, USA: California Technical Pub, 1997.
- [23] Khalili R, Salamatian K. A new analytic approach to evaluation of packet error rate in wireless. In: Proceedings of the 3rd Annual Communication Networks and Services Research Conference; Halifax, NS, Canada; 2005. pp. 1-6. doi: 10.1109/CNSR.2005.14