Digital Radio Baseband and Testbed for Next Generation Wireless System

Naing Ye Aung

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2016-87
http://www.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-87.html

May 13, 2016

Table of Contents

List of Figures and Tables

Chapter 1: Technical Contribution (Individual)

Section I: Project Introduction

In this project, "Digital Radio Baseband and Testbed for Next Generation Wireless System", our goal is to design hardware generators for a part of a software defined radio. Unlike the traditional design approach, which entails designing a new hardware for each set of specifications, we design our radio with the vision of "instance-on-demand". In other words, our approach allows the user to use only one framework to get different instances of hardware based on their specifications, thus saving their time and cost of developing new hardware. The whole project is designed in Chisel, an open-source hardware construction language developed at UC Berkeley. Its big benefit for our project is the ability to design parameterized hardware generators.

This project is a collaboration effort with several graduate students in Berkeley Wireless Research Center (BWRC). Together with our advisors, they provided mentorship to our team throughout the duration of the project. Figure 1 illustrates a system diagram of the receiver of software defined radio, where the subsystems marked in red are the ones our capstone team were responsible for: automatic gain control (AGC), channel estimation, and demodulator. AGC is a subsystem that will adjust the amplitude of the received signal to a desired value. To model and compensate for the effects of the environment, a combination of subsystems known as channel estimation and equalizer is used. Lastly, after the necessary compensation, demodulator will transform complex numbers to binary numbers, which can then be decoded to more meaningful information such as music, videos, pictures, etc. This report will focus on the technical work completed for the demodulator, the subsystem I was responsible for.

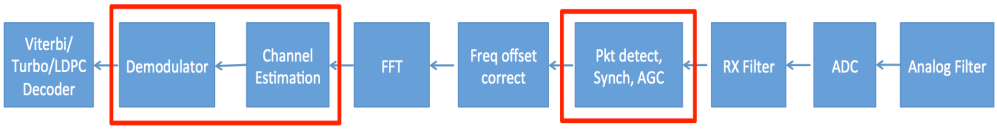
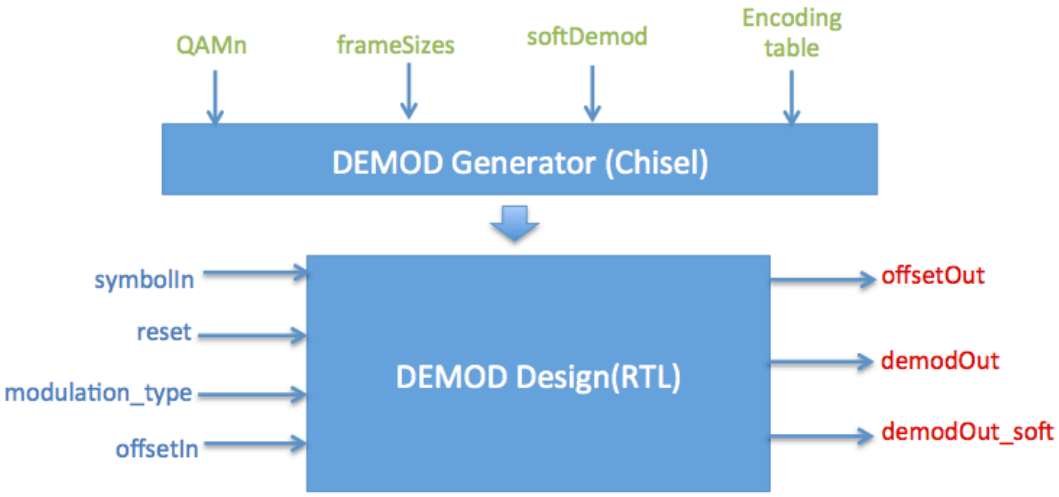


Figure 1: System Diagram of Receiver of a Software Defined Radio

In this project, being parameterizable means that the users, most likely the hardware designers, can specify the hardware capability that they want before generating the hardware design. Being run-time reconfigurable refers to the ability of the end-user of the physical hardware to modify the functionality. The high-level interface of the final design of the parameterizable and run-time reconfigurable demodulator and a brief description of the interface is illustrated in figure 2.

In demodulator, the parameterizable features include selecting a subset of the modulation schemes that users want the hardware to support (BPSK, QPSK, 16-QAM, 64-QAM, 256-QAM), specifying the encoding table for each modulation scheme, and whether the demodulator needs to support soft decision. Each modulation scheme is specified by a constellation, a graph of points that maps from the complex number to the binary numbers. The higher the number of points, the larger the number of binary numbers the constellation can represent, and thus the higher the resolution. Different modulation schemes specify different bit resolutions. For each modulation scheme, an encoding or a demapping table is a look up table to convert between symbols and bits. While hard demodulator returns only the guess of each bit, a soft demodulator returns its guess and the associated confidence level, represented in the form of a log likelihood ratio.



- QAMn: List of supported n-QAM
- frameSizes: Supported frame sizes
- softDemod: If true, should do LLR calc, otherwise hard demod
- encoding table: Mapping table from symbol to bits.
- symbolIn: Input symbol
- reset: reset when high
- modulation_type: constellation type to demodulateBPSK:2, QPSK:4, 16-QAM: 16, 64-QAM: 64, 256-QAM: 256
- offsetIn: offset of the input sample relative to frame size
- offsetOut: offsetIn delayed by the number of cycles to compute demodOut/demodOut_soft.
- demodOut: demapped bits, result is available when parameterized to hard demodulator
- demodOut_soft: Log likelihood ratios(LLRs), result is available when parameterized to soft demodulator

Figure 2: System level diagram of demodulator and the brief description of associated I/O and parameters

Section II: Problem Definition

As discussed earlier, a demodulator is responsible for transforming input symbols, represented in complex numbers, to binary numbers. The output can then be decoded to more meaningful information such as music, videos, pictures, etc.

As a hardware-generator for demodulator, the design has to be generic so that it can suit a wide range of software defined radio designers and yet optimized so that the hardware generated can be efficient. This seeming contradiction is overcome by parameterizing the design. If the parameters are specified in such a way that an optimized design is feasible, such a design is generated. If not, a less optimal but functional design is generated. This approach results in optimized designs for a specific set of parameters while giving the flexibility to generate a working design if the users decide to select a different set to fit their design purpose.

By default, the optimized designs conform to the encoding table of IEEE 802.11a, one of the Wi-Fi wireless network communication standards. Additional assumptions are also made in the demodulator design. The hardware generator assumes that the QAM symbols can be decomposed into two independent PAM signals. The importance of the need for this assumption will become clearer in the future sections. Furthermore, the constellations supported in this work are square constellations, where all the constellation points are arranged in a square. Currently, the square constellations supported are BPSK, QPSK, 16-QAM, 64-QAM, and 256-QAM.

Section III: Literature Review

This section begins with exploring the standard of interest, IEEE 802.11a. In particular, the bit encoding for different constellations in this standard will be discussed.

The primary focus is on some of the existing approaches and designs of hard and soft demodulator for the constellations of interest in this project. Under different assumptions, conventional designs can be optimized for lower complexity and faster run-time while suffering little to no performance degradation.

IEEE 802.11a

This section goes into the constellation bit encoding for BPSK, QPSK, 16-QAM, and 64-QAM. Figure 3 illustrates the encoding for the constellations. (IEEE Standards Association 1601) The corresponding encoding tables(Table 1-4) are also included. (IEEE Standards Association 1602). In all of these illustrations, the order from b_0, b_1, … , b_k goes in the order from most significant bit(MSB) to least significant bit(LSB), where $k = \log_2 M$ and $M = 2,4,16,64$ for BPSK, QPSK, 16-QAM, and 64-QAM respectively. Though not specified in the standard, the extension of encoding can be made to 256-QAM and the signal constellation is included in figure 4. (Yeh & Seo 2007:6)

It is worth noting that the input bits in all the encoding tables are Gray coded, where each adjacent entry in the column only differ by one bit. One major benefit is that a single out-of-position-by-one error results in only a single bit error in the received data.

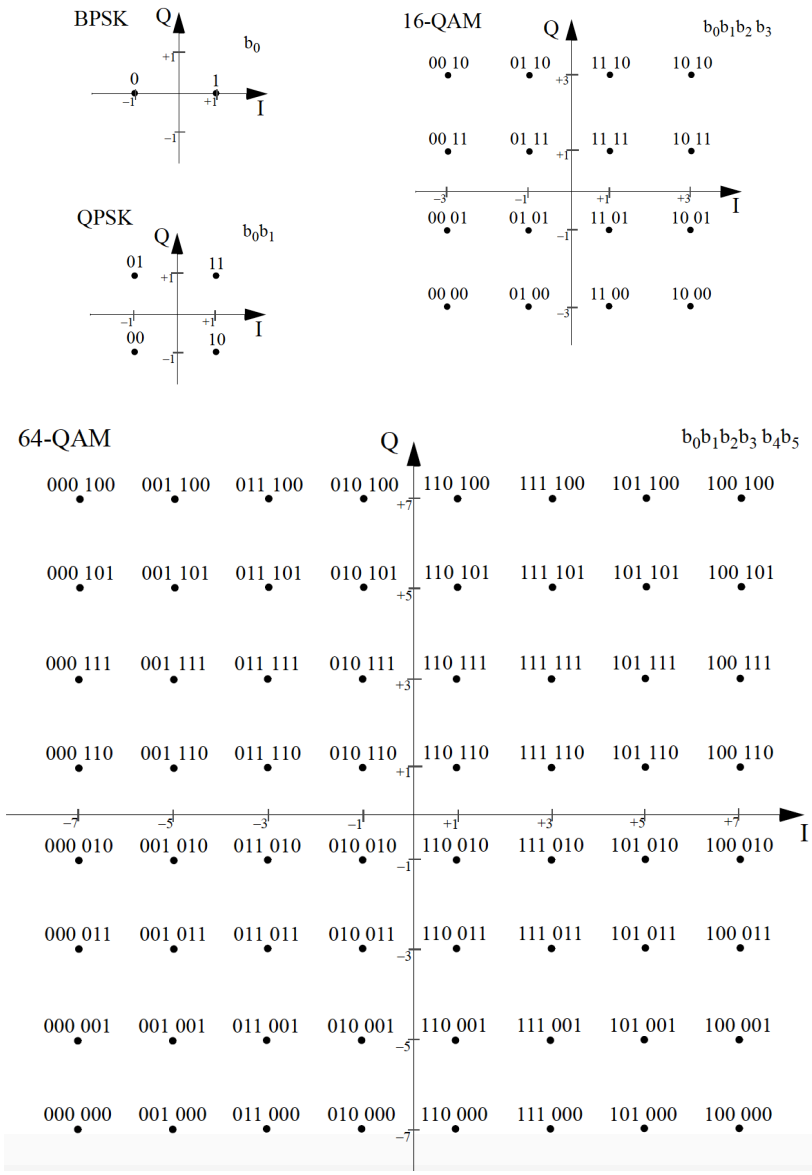


Figure 3: bit encoding for BPSK, QPSK, 16-QAM, 64-QAM for IEEE 802.11a (IEEE Standards Association 1601)

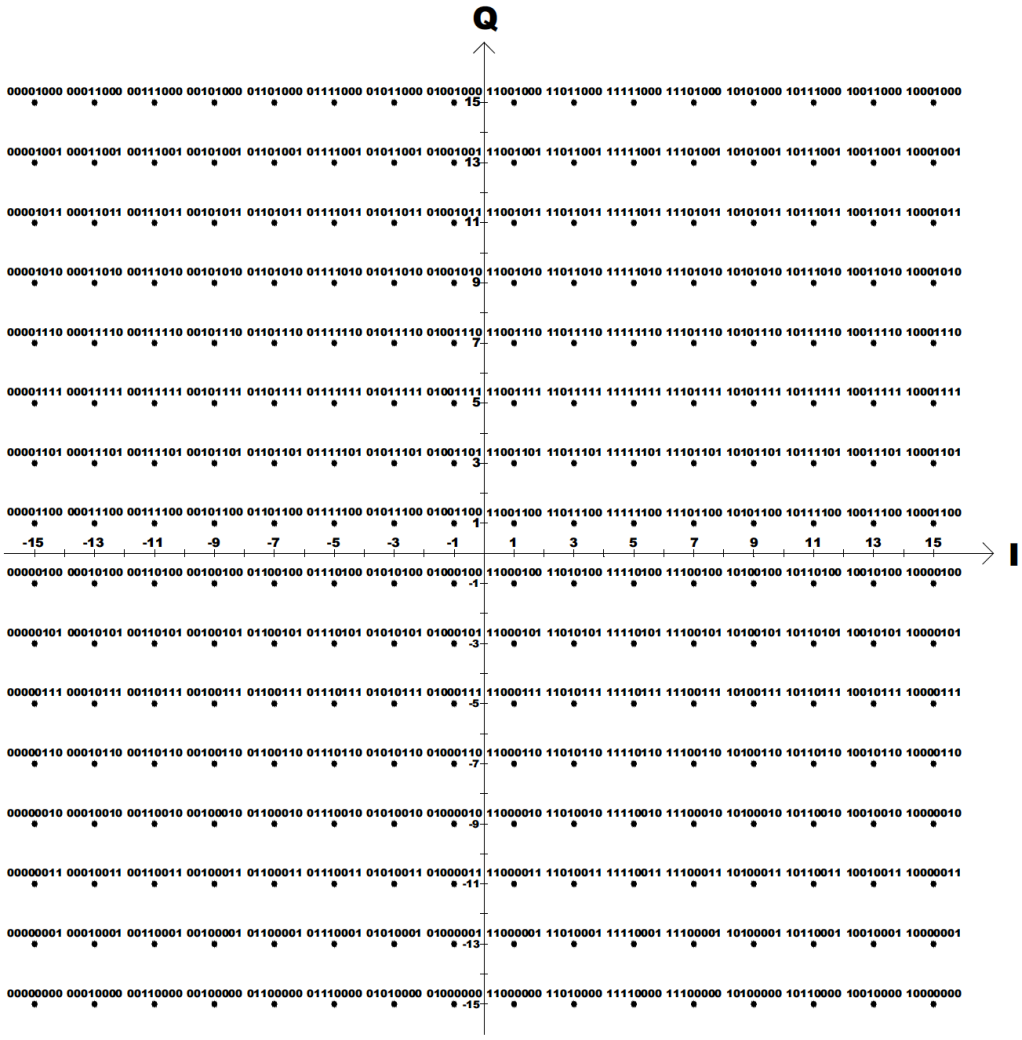


Figure 4: bit encoding for 256-QAM as an extension to IEEE 802.11a. (Yeh & Seo 2007:6)

Input bit (b_0)	I-out	Q-out
0	–1	0
1	1	0

Table 1: bit encoding table of BPSK for IEEE 802.11a (IEEE Standards Association 1602)

Input bit (b_0)	I-out
0	–1
1	1

Input bit (b_1)	Q-out
0	–1
1	1

Table 2: bit encoding table of QPSK for IEEE 802.11a (IEEE Standards Association 1602)

Input bits (b_0 b_1)	I-out
00	–3
01	–1
11	1
10	3

Input bits (b_2 b_3)	Q-out
00	–3
01	–1
11	1
10	3

Table 3: bit encoding table of 16-QAM for IEEE 802.11a (IEEE Standards Association 1602)

Input bits (b_0 b_1 b_2)	I-out
000	–7
001	–5
011	–3
010	–1
110	1
111	3
101	5
100	7

Input bits (b_3 b_4 b_5)	Q-out
000	–7
001	–5
011	–3
010	–1
110	1
111	3
101	5
100	7

Table 4: bit encoding table of 64-QAM for IEEE 802.11a (IEEE Standards Association 1602)

Hard Demodulator

Hard decision demodulator returns the best guess for the bits transmitted without any additional information.

Low Complexity Demodulator for M-ary QAM

Under the assumptions that the constellation of interest is a square, and the symbols are represented by binary-reflected Gray coding, the low-complexity demodulator perform only scaling and comparison to generate a guess for each bit. (Yeh & Seo 2007:2)

For input symbol $\alpha+j\beta$, the algorithm for demodulating each bit b_k for M-ary QAM, k = 0 to n – 1 and $M=2^{2n}$ conforming to the encoding table of IEEE 802.11a is illustrated below:

Step 0: $b_0 = 1$ if $\alpha \geq 0$, 0 otherwise.

Step 1: $b_1 = 1$ if $|\alpha| \leq 2^{n-1}$, 0 otherwise

Step 2: $b_k = 1$, if $(4i-3)2^{n-k} \leq |\alpha| \leq (4i-1)2^{n-k}$, for (i =1 or 2 or 3 up to 2^{k-2})

0 otherwise.

For example, for 256-QAM, where M = 256, n =4, following the algorithm, $b_0 = 1$ if $\alpha \geq 0$, 0 otherwise. $b_1 = 1$ if $|\alpha| \leq 2^{4-1}$, 0 otherwise. $b_2 = 1$ if $4 \leq |\alpha| \leq 12$, 0 otherwise. $b_3 = 1$ if $2 \leq |\alpha| \leq 6$ or $10 \leq |\alpha| \leq 14$, 0 otherwise. The second half bits (b_4 to b_7) follows the same algorithm as b_0 to b_3, except now β instead of α is used in comparison. (Yeh & Seo 2007:4)

Table 5 provides a comparison on the computational requirements between the design proposed and the regular M-QAM demodulator. This design illustrates the lack of need for multiplication, a computationally intensive operation, and real addition and subtraction. In addition, the number of comparisons necessary to compute a result is far lower. Figure 5 shows the bit error rate (BER) performance of the low-complexity demodulator for BPSK, QPSK, 16-QAM, 64-QAM, and 256-QAM in the presence of Additive White Gaussian Noise (AWGN). The plot illustrates an exact match between theory and actual performance. (Yeh & Seo 2007: 5)

M-QAM Detector	Real Multiplication		Real Addition & Subtraction		Comparison	
	Regular ($2M$)	New	Regular ($3M$)	New	Regular (M-1)	New $2\sum_{i=0}^{n-1} 2^i$
M=4	8	0	12	0	3	2
M=16	32	0	48	0	15	6
M=64	128	0	192	0	63	14
M=256	512	0	768	0	255	30
M=1024	2048	0	3072	0	1023	62

Table 5: Comparison of Computation requirement between regular and low-complexity design(Yeh & Seo 2007: 5)

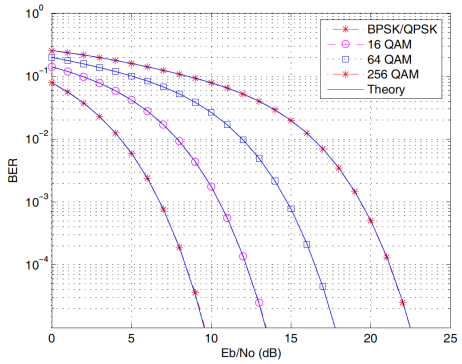


Figure 5: Comparison of BER for multiple modulation schemes in theory and actual performance of the low complexity demodulator (Yeh & Seo 2007 : 5)

Soft Demodulator

A soft demodulator or a soft demapper is a more complex demodulator that provides confidence level of its guess in terms of log-likelihood-ratio (LLR), which is the log of the ratio of the probability of the bit being a 0 to the probability of being a 1. The theory will be explained more below.

Despite being more complex than a hard demodulator, a soft demodulator has a smaller bit error rate for the same signal to noise ratio. In other words, under the same environment conditions, a soft demodulator can result in a much more accurate result. Figure 6 shows a comparison of the bit error performance of hard and soft decoding. (Tosato & Bisaglia 2002: 668) The figure also compares the results with multiple modes of soft decoding, which are irrelevant for understanding the usefulness of soft decision. For the same mode, we can see that the bit error rate of soft decoding is lower than hard decoding.

Most soft demodulator designs provide soft information by calculating the log-likelihood ratio for each bit in the group of bits a symbol maps to. For received signal *y*, the log –likelihood ratio for each *ith* bit is

$$L(b_i) = ln\frac{\mathrm{P(b_i=0|}y)}{P(b_i=1|y)} = \ln\frac{\sum_{\boldsymbol{b}:b_i=0}\exp\left(-\frac{|y-x_{\boldsymbol{b}}|^2}{2\sigma^2}\right)}{\sum_{\boldsymbol{b}:b_i=1}\exp\left(-\frac{|y-x_{\boldsymbol{b}}|^2}{2\sigma^2}\right)} \quad (1)$$

where σ denotes noise variance assuming additive white Gaussian noise, $x_{\mathbf{b}}$ is a symbol on the constellation and $\mathbf{b} = [b_0, b_1, \ldots, b_{m-1}]^T$ that the symbol maps to. The numerator term groups the sum of the exponentials of Euclidean distances where the bit at *ith* location is 0 and the denominator groups the sum of the exponentials of Euclidean distances where the bit at *ith* location is 1. However, due to high number of computations necessary to implement the expression in hardware, an alternative expression is proposed.

Known as the Max-Log-MAP, the simplified expression is

$$L(b_i) \approx \frac{\left(min_{\boldsymbol{b}:b_i=1}|y-x_{\boldsymbol{b}}|\right)^2 - \left(min_{\boldsymbol{b}:b_i=0}|y-x_{\boldsymbol{b}}|\right)^2}{2\sigma^2}, \qquad (2)$$

The first term is the squared of the Euclidean distance of the received symbol and the nearest constellation point where the signal at *i*th bit in the constellation is 1 and the second term is identical except the nearest constellation point is the point where the signal at *i*th bit in the constellation is 0. (Liu & Kosakowski 2015: 50, 51)

Max-Log-MAP Soft Demapper with Logarithmic Complexity for M-PAM Signals

Assuming that the M^2-QAM constellation can be decomposed into two independent M-PAM signals, implementing equation 2 using exhaustive search requires order of (MlogM) subtractions.

If the mapping scheme is 3GPP-like Gray coding, the proposed demapping scheme claims that it can reduce the complexity to the order of O(logM) without any performance loss.

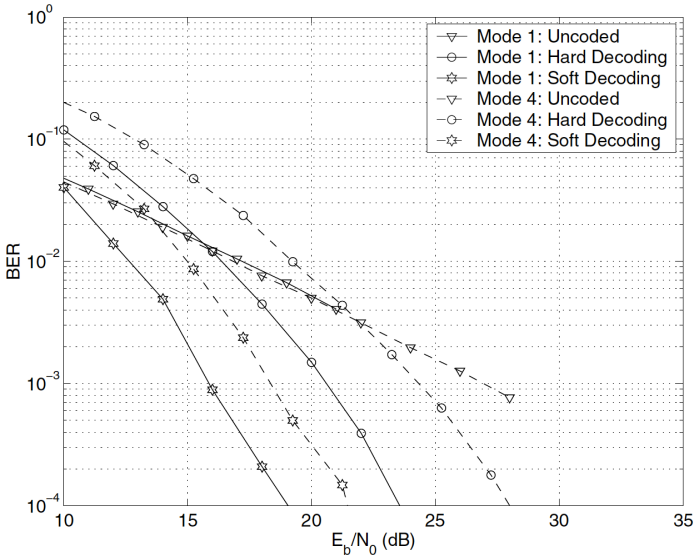


Figure 6: Bit Error Rate performance of soft vs hard decoding (Tosato & Bisaglia 2002: 668)

Under these constraints, the nearest constellation point can be calculated by the following algorithm as directly cited from the paper (Liu & Kosakowski 2015: 52):

1. Calculate the soft sliced bits

$$y_0 = y,$$

$$y_i = 2^{m-i}d - |y_i - 1|, \;\; i = 1, \dots, m-1,$$

$$where\ d\ is\ half\ the\ distance\ between\ two\ constellation\ points$$

2. Calculate the minimum distances for i = 0, …, m-1

$$min_{\boldsymbol{b}:b_i \neq \widehat{b_i}} |y - x_{\boldsymbol{b}}|^2 = (d + |y_i|)^2,$$

$$min_{\boldsymbol{b}:b_i = \widehat{b_i}} |y - x_{\boldsymbol{b}}|^2 = (|y_{m-1}| - d)^2$$

3. Calculate the LLR values from i =0,…, m-1

$$L(b_i) = \frac{\left(min_{\boldsymbol{b}:b_i=1}|y - x_{\boldsymbol{b}}|\right)^2 - \left(min_{\boldsymbol{b}:b_i=0}|y - x_{\boldsymbol{b}}|\right)^2}{2\sigma^2}$$

$$= \begin{cases} \dfrac{min_{\boldsymbol{b}:b_i \neq \widehat{b_i}}|y - x_{\boldsymbol{b}}|^2 - min_{\boldsymbol{b}:b_i = \widehat{b_i}}|y - x_{\boldsymbol{b}}|^2}{2\sigma^2}, y_i \geq 0, \\ \dfrac{min_{\boldsymbol{b}:b_i = \widehat{b_i}}|y - x_{\boldsymbol{b}}|^2 - min_{\boldsymbol{b}:b_i \neq \widehat{b_i}}|y - x_{\boldsymbol{b}}|^2}{2\sigma^2}, y_i < 0, \end{cases}$$

Section IV: Approach

This section aims to provide a high-level design approach taken to get to the final design. The exact design decisions taken will be discussed in the next section. The design flowchart is illustrated in figure 7. At every step of the flow chart, a more refined minimal viable product is created after thorough testing. The input and output ports were refined and parameters were added and removed as appropriate.

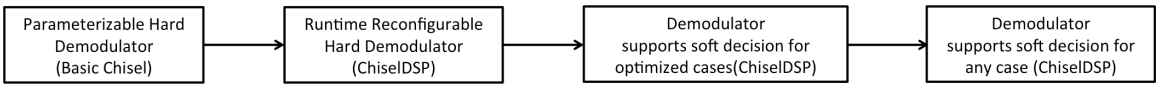


Figure 7: Design Flowchart for a fully parameterizable and runtime reconfigurable demodulator

After agreeing upon a preliminary set of parameters and input and output interfaces with our graduate student mentors, a hard demodulator that supported the constellations was written using the default Chisel environment. At this point, the constellations were

parameterized. A different set of hardware would be generated based on the modulation scheme specified. Moreover, only one type of modulation scheme would be supported for each hardware design being generated. Furthermore, the encoding table for the modulation scheme was restricted such that only the largest constellation supported, 256-QAM could be parameterized.

In the second step, our hardware generator design was refined so that users could specify the list of constellations they would want the hardware generated to support at run-time. They could also specify the encoding table they want for each constellation their hardware design would be supporting. The usefulness of our hardware generator also improved by using the ChiselDSP library, an API developed by one of our graduate advisers. The major difference is the support for decimal points in the form of Fixed representation and double type, which did not exist before.

The next step was to begin implementing the soft decision feature. The end result of third step was a hardware generator design that could support soft decision for binary-reflected Gray coding. At this point, we had a hardware generator that could fully support IEEE 802.11a. The final step was the capability to generate a design that can return soft decisions for any encoding table.

Section V: Design

Different algorithms are executed to create hardware of varying optimality depending on user's specified parameters. The major hardware blocks that can be generated can be categorized into optimized hard demodulator, a generic hard demodulator, an optimized soft demodulator, a generic soft demodulator. The philosophy behind implementing different algorithms is to generate an optimized design when the parameters specified by the users provide an opportunity but still be able to generate a functional design otherwise. The major

factor in deciding whether to create an optimal or a conventional hardware instance comes from the way the encoding table is specified. By default, the encoding table is specified such that the hardware generated will be more optimal than the conventional design.

Optimized Hard Demodulator

The algorithm implemented in this case generates a topology that follows the implementation mentioned in Section III: Low Complexity Demodulator for M-ary QAM. The optimized hardware instance is optimized in the sense that hardware is simple and the computation necessary to return an output is small. This provides the benefit of small physical area, low power consumption, and possibility of running the hardware at faster clock speed. This is the default hardware generated if the users do not modify the encoding table.

Generic Hard Demodulator

If the users' specified parameters do not match the optimization check, a different algorithm is executed to generate a different architecture. The system diagram of the algorithm to support one constellation is illustrated in figure 8. If n constellations are supported in the same hardware instance, the actual system will be n identical copies of the system.

One look up table is created for every modulation scheme that users desire to support. The look up table maps from symbol to bits. Each received symbol is split to real and imaginary counterpart and decimal parts are truncated to fit to an existing symbol on the constellation. The addresses of the LUT for real and imaginary symbol are then calculated. The two calculated addresses are then fed into the LUT. The two outputs from the two different addresses are then concatenated with the output from address of real number in the MSB and that of the imaginary number in the LSB. The separation of concern for real and imaginary input is possible only because we assumed that the QAM symbols can be decomposed into two independent PAM signals.

This design is less complex than a common alternative, which is to use successive quaternary search. The computational complexity for this algorithm stays constant while the other increases in the order of O(log M), as constellation size M^2 increases.

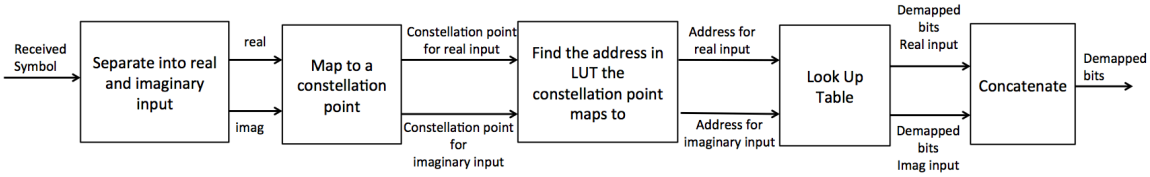


Figure 8: System Diagram of a Generic Hard Demodulator to support one constellation

Optimized Soft Demodulator

The algorithm implemented in this case generates a topology that follows the implementation mentioned in Section III: Max-Log-MAP Soft Demapper with Logarithmic Complexity for M-PAM Signals. It is worth noting that though the algorithm is specified for 3GPP-like Gray-coded PAM, since the encoding table of IEEE 802.11a is the exact inversion, the same set of equations can be applied. The only difference is that in step 3, the order of the two terms are inverted.

The benefits are similar with the case of optimized hard demodulator. This is the default hardware generated if the user does not modify the encoding table and desires a soft decision output.

Generic Soft Demodulator

If an optimized soft demodulator design is not possible, implementing approximate LLR in hardware becomes more complex due to the necessity to find the minimum terms in the equation. Figure 9 illustrates the system diagram of the design to calculate all the LLRs for a given symbol in one clock cycle to support one constellation. The hard demodulator outputs the mapped bits and the mapped symbol, which is the constellation point the symbol maps to for a given constellation. The distance between the mapped symbol and the input symbol is

calculated to obtain the term $min_{\boldsymbol{b}:b_i=\widehat{b_i}}|y - x_{\boldsymbol{b}}|^2$ where $\widehat{b_i}$ = 0 if the mapped bit at *ith* position is 0 and 1 otherwise.

To obtain the term $min_{\boldsymbol{b}:b_i\neq\widehat{b_i}}|y - x_{\boldsymbol{b}}|^2$, the following information were first determined: the *bit to find*, and the *list to find* the bit in. For *ith* position, the *bit to find* is the inverted version of the demapped bit at the same location. The *list to find* is the list of the demapped bits of the entire encoding table at the *ith* location for the particular constellation. This is extracted by putting all the possible addresses into a LUT, which then outputs all the possible demapped bits. By transposing the outputs, each row then become the *list to find* for each *ith* bit. Using input symbol, *bit to find* (Inverted_*bits* in figure 9), *list to find* (list_to_find in figure 9), linear search through *list to find* can be executed to find the minimum distance between the symbol and the constellation point that demaps to *bit to find* for the *ith* location. Therefore, $min_{\boldsymbol{b}:b_i\neq\widehat{b_i}}|y - x_{\boldsymbol{b}}|^2$ can be calculated. Using this and $min_{\boldsymbol{b}:b_i=\widehat{b_i}}|y - x_{\boldsymbol{b}}|^2$, LLR can be calculated for all the bits.

The computational complexity of this algorithm for one received symbol is in the order of O(MlogM), where M^2 is the number of constellation points in any given constellation. It takes O(M) to calculate LLR of each bit and since there are $\log_2 M$ bits, it takes M * log M amount of complexity. This is the same order of complexity as mentioned in Section III.

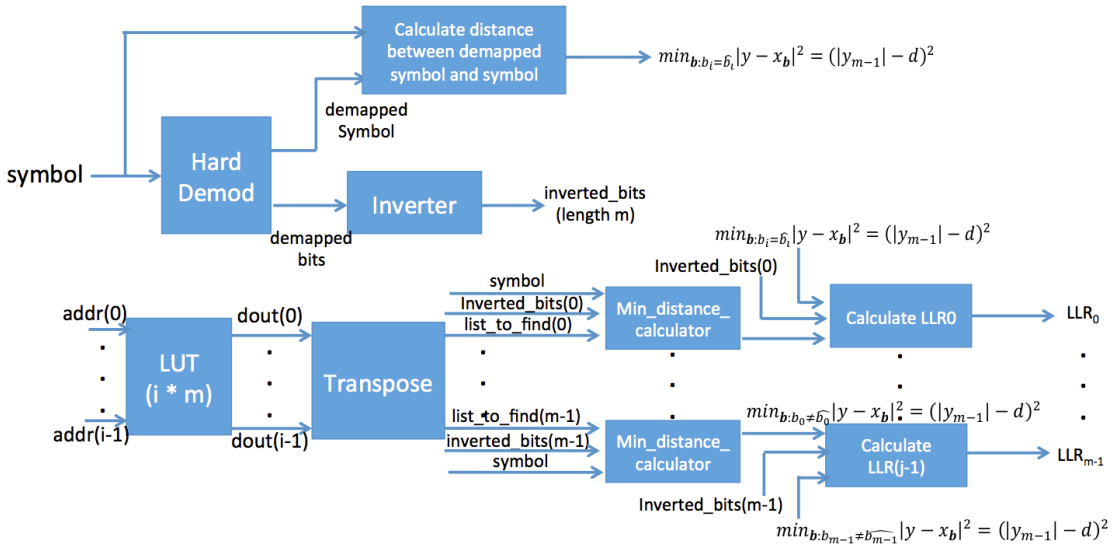


Figure 9: Generic Soft Demodulator architecture supporting one constellation that can calculate all LLRs in 1 clock cycle

Section VI: Validation

Test code was written in Chisel to valid the functionality of the design. i.e The expected outputs were obtained when testing with different inputs. Separate sets of test suites were written for hard and soft demodulator. For hard demodulators, every point in the constellation for every constellation was used as an input to both the optimized and generic hard demodulator and the corresponding output was verified to ensure it matched with the expected output. Since the functionality of hard demodulators should be identical regardless of topology, the same set of tests could be used.

To test the soft demodulator designs, every point in the constellation for every constellation was used as inputs except with a difference: a random number is added on top of the constellation point. This provided the capability to test that the outputs are correct for varying distances from the ideal constellation point. Exhaustive testing was not possible because the inputs were real numbers and the number of possible real numbers in a range is infinite. The

boundary cases when the received symbol were exactly midway between two constellation points were also tested and validated.

The design was also pushed through FPGA flow with a variety of different parameters. Table 6 illustrates the area utilization in terms of Slice LUTs and SLICE Registers for four different sets of parameters supporting all constellations. They are the optimized hard demodulator, the generic hard demodulator, the optimized soft demodulator, and the generic soft demodulator.

Type	Slice LUTs	SLICE Registers
Full Hard Demodulator (Optimal)	0.08%	< 0.01%
Full Hard Demodulator (Generic)	0.02%	< 0.01%
Full Soft Demodulator (Optimal)	1.32%	0.03%
Full Soft Demodulator(Generic)	5.01%	0.03%
Total Available Resources	303600	607200

Table 6: Resource Utilization for four different hardware instances generated.

As expected, the resource utilization of soft demodulator is significantly larger than that of the hard demodulator. Additionally, the resource utilization of soft demodulator supporting any constellation is larger than that of the optimal soft demodulator. However, it is interesting to note that the resource utilization for the optimal hard demodulator is larger than that of the generic demodulator.

Section VII: Conclusion

The current work supports the selection of modulation schemes, corresponding encoding tables, and choice to output soft or hard decisions. Choices of modulation schemes include BPSK, QPSK, 16-QAM, 64-QAM, and 256-QAM. After the parameters are specified and the design is generated, end users of the design will be able to select among the modulation schemes that the designers specify they wanted this hardware to demodulate the received symbols.

More improvements and extensions can be made upon the current work. For the sets of parameters where generic demodulator algorithms are executed, there might exist some cases where more optimal topologies exist. It may be worthwhile to conduct more literature search to identify these cases and implement more optimal topologies. Additionally, the LLR equation consists of noise variance term, which was assumed to be one in this work. Instead, mechanism that estimates the term can be implemented. Lastly, supporting non-square constellations such as 128-QAM can increase the usefulness of this hardware generator to support more communication standards.

References:

IEEE Standards Association. (2007). IEEE 802.11 standard-Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

Liu, X., & Kosakowski, M. (2015). Max-Log-MAP Soft Demapper with Logarithmic Complexity for-PAM Signals. *Signal Processing Letters, IEEE*,*22*(1), 50-53.

Tosato, F., & Bisaglia, P. (2002). Simplified soft-output demapper for binary interleaved COFDM with application to HIPERLAN/2. In *Communications, 2002. ICC 2002. IEEE International Conference on* (Vol. 2, pp. 664-668). IEEE.

Yeh, H. G., & Seo, H. (2007, April). Low complexity demodulator for M-ary QAM. In *Wireless Telecommunications Symposium, 2007. WTS 2007* (pp. 1-6). IEEE.

Chapter 2: Engineering Leadership(Team-Written)

Section 1: Industry Analysis

To understand the prospect of profitability in the industry our product fits in the best, the Software Defined Radio Industry, we adopt Porter's Five Force Analysis, where the five forces are substitutes, buyers, new entrant, rivalry, and suppliers. We will analyze how these five forces influence profitability in this industry (Porter, 2008).

The force of substitutes for our product is weak. Our primary substitute is the traditional hardware based radio. SDR poses very attractive advantages that substitutes cannot provide. Its unique ability to implement multiple current communication standards on a radio is non-existent in traditional hardware based radio devices. Secondly, SDR customers will have a sustainable system with less recurring cost every time a new communication technology update emerges. As a result, SDR can create a reconfigurable radio system that has multimode and multiband to support different communications standards with lower overall cost.

Buyers have a lot of power in our industry. The two largest customer sectors in the current SDR industry are the military and the telecommunication infrastructure equipment companies. SDR is being deployed in tactical radios because it enables joint operations between separate troops from national and international operations even though the network communications in each country are different. For the telecommunication infrastructure, they avoid creating new infrastructure for each update. According to Mobile Experts and the Wireless Innovation Forum, tactical radio manufacturers sold approximately 200,000 SDR embedded tactical radios in 2012 and the amount has been increasing annually (Pucker and Renaudeau, 2012). The buyers possess significant power because of low switching costs between different SDR vendors. Additionally, there are a limited number of buyers in a huge market, which strengthens their negotiation power. Since SDRs support a variety of standard communication

standards, the buyers can just switch to another SDR vendor that support the standards if the product of current vendor, for example, becomes too expensive.

The force of new entrants is also strong because of the low barrier to entry. As there are several small SDR design/manufacturing companies, very few of which are public, it indicates that the SDR industry does not require large financial investment. From a technical perspective, the core technology dates back to a few decades ago so the information is widely available.

Although entering the market is relatively easy, there is strong rivalry based on features such as power consumption, bandwidth, and efficient architecture. Since there are already many SDR manufacturers such as Northrop Grumman Corporation, L-3 Communication Holdings Inc., and Raytheon Co. (Marketsandmarkets.com, 2014), most of the features are already covered by one or more companies' product. However, if our product, which is a design tool for SDR, can be used to build a new product that has a unique feature, the rivalry force would be weak.

However, the force of suppliers is weak. The major suppliers for SDR are similar to the suppliers for traditional radio, which includes component manufacturers, for example, antenna and other basic analog devices like analog digital converter (ADC), digital analog converter (DAC). Since these parts are standardized, the suppliers have weak bargaining power.

With weak forces in suppliers and substitute, but strong forces in new entrants, rivalry based on feature, and buyer, SDR may not seem to be an attractive industry to enter at the first glance. However, our marketing strategy explained in the following section will create a more favorable situation for our product.

Section II: Marketing

The SDR market has been constantly growing in the last few years. The SDR market size is predicted to reach $27.29 billion by 2020 with a Compound Annual Growth Rate(CAGR) of 12.5% from 2014 to 2020 (Marketsandmarkets.com, 2014). The market demand for SDR still remains strong and has yet to saturate more than 30 years after its initial inauguration (Clarke and Kreitzer, 2015). The huge growth is understandable because new communication technologies have been emerging year after year. According to IBIS industry reports, the revenues of both the Communication Equipment Manufacturing (Ulama, 2015) and Wireless Telecommunications Carriers (Blau, 2015) industries in the US currently stand at $33.8 billion and $248.7 billions respectively. This positive market trend and huge market size on the communication industry provides worthy reasoning to explore SDR development.

There is another great future potential market for SDR. Part of current SDR users are those who use small consumer electronic devices. Tremendous growth in the Internet of Things(IoT) market has led to various applications, for instance Smart Grid, home automation systems, and intelligent industrial system (Bushehri, 2013). IoT acts as a smart gateway to connect between multiple low-power, low-cost devices and with each other and the internet (Bushehri, 2013). All of these devices have different ranges of communications protocols and interfaces, such as Bluetooth, Wi-Fi, and ZigBee. These multiple wireless standards are implemented more efficiently using SDR technology as SDR can re-configure those devices and create a gateway that can connect everything into a whole system.

Apart from the market size, we will also discuss the “4P’s” of marketing: product, price, place, and promotion. Though our project is currently set for research purposes, for this analysis, we will be assuming the scenario where we apply our technology in a business setting.

Our product is an open-sourced Software Defined Radio(SDR) platform which can be customized to user's needs. It can provide SDR designers with instant hardware designs to expedite their product development by only focusing on developing the value-added components.

As an open-sourced project, our product is available for anyone who is trying to develop SDR from scratch. Users can integrate our product on their systems without paying any licensing fee. On the other hand, as the main developer of the product, we can adapt a service business model similar to that of Linux/Red Hat. We can provide services to help potential customers transition to our framework, implement, and integrate customized features not yet implemented. The major platform for support will be available through online forum, chat, or video-conferencing. Thus, we will be able to create a direct channel without face-to-face encounter (Wenkart, 2014). Field Engineers will also be available if requested.

We recognize the lack of trust potential users may have in adopting our platform. Therefore, we aim to promote our reputation, which will be achieved through journal publications and conferences. Sponsorship deals will also be provided for renowned corporates as their subscription for our support service will accelerate the level of credibility.

Section III: Intellectual Property (IP)

The two options considered for open source licensing are Berkeley Software Distribution(BSD) and General Public License(GPL). The main difference between BSD and GPL is that any developer who modifies source code licensed under GPL is also required to license his/her work under GPL if he/she wants to distribute the product. In contrast, developers who enhance or modify a product licensed using BSD are not under such restriction, opening up business opportunities. As a result, BSD will encourage more people to improve and use this tool.

Open source is chosen due to several reasons. We are currently using publicly available design architectures to create these generators. Thus, we do not have any new algorithm/design for the SDR blocks, making them unsuitable for patents. Instead, our generator aims to help others who want to build new products to use the existing design. As we are using/implementing the existing designs, making this project open source will protect us from lawsuit. It is also beneficial to the design community as donating our idea to the public will encourage innovation by saving time/effort to design the fundamental blocks.

By building a support model around this, we can create win-win situation; by creating a lower barrier to adopt this framework, the number of SDR designers will increase which not only encourages innovation, but also generates more revenue for our business.

References:

Blau, G. (2015). *IBISWorld Industry* Report 51721: Wireless Telecommunications Carriers in the US*.* Retrieved October 18, 2015 from IBISWorld database.

Bushehri, E. (2013). *Future Wireless Networks Will Rely On Programmable SDRs.* Electronic Design.

Clarke, B. & Kreitzer, K. (2015). How to Maximize Your Software-Defined Radio's Dynamic Range. Microwave Product Digest.

Marketsandmarkets.com. (2014). Marketsandmarkets: Software Defined Radio (SDR) Market for Communication by Component (FPGA, DSP, GPP, PSOC, Amplifier, and Software), Application (Military, Telecommunication, Short Range, Positioning, Transportation, and Public Safety), & Geography - Analysis & Forecast to 2014 – 2020. Retrieved from http://www.marketsandmarkets.com/Market-Reports/software-defined-radio-communication-market-11265833.html.

Porter, M (2008). *The Five Competitive Forces That Shape Strategy.* Harvard Business Review.

Pucker, L & Renaudeau, D. (2012). *Conquering SDR tactical radio market challenges.* Military Embedded Systems.

Ulama, D. (2015). *IBISWorld Industry Report 33422:* Communication Equipment Manufacturing in the US*.* Retrieved October 18, 2015 from IBISWorld database.

Wenkart, M (2014). The Marketing Bible. Norderstedt: on Demand.