Science Journal of Circuits, Systems and Signal Processing
2015; 4(4): 23-29
Published online August 12, 2015 (http://www.sciencepublishinggroup.com/j/cssp)
doi: 10.11648/j.cssp.20150404.11
ISSN: 2326-9065 (Print); ISSN: 2326-9073 (Online)

Algorithm Development of a Sampled Data Frequency Modulation Demodulator for the Implementation of Software Defined Radios

Thomas Kokumo Yesufu[1], **Abiodun Alani Ogunseye**[2]

[1]Department of Electronic and Electrical Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria
[2]Department of Electrical /Electronics and Computer Engineering, Bells University of Technology, Ota, Nigeria

Email address:
thomas_yesufu@yahoo.com (T. K. Yesufu), bigabbey4god@yahoo.com (A. A. Ogunseye)

To cite this article:
Thomas Kokumo Yesufu, Abiodun Alani Ogunseye. Algorithm Development of A Sampled Data Frequency Modulation Demodulator for the Implementation of Software Defined Radios. *Science Journal of Circuits, Systems and Signal Processing*. Vol. 4, No. 4, 2015, pp. 23-29. doi: 10.11648/j.cssp.20150404.11

Abstract: The study developed and evaluated the performance of an improved algorithm to demodulate sampled data frequency modulation (FM) signals in typical field programmable gate array (FPGA) - based software defined radios (SDR). An algorithm, based on the modification of a standard differentiate-divide FM demodulator was developed. The hardware resources requirement and the input noise suppression ability of the developed algorithm were investigated. The demodulator developed requires a quarter of the hardware resources needed by conventional differentiate-divide FM demodulators. The output signal to noise ratio (SNR) of the developed demodulator is lower than that of the standard differentiate divide-demodulator when the input carrier to noise ratio (CNR) is less than 45 decibels. The SNR plot of the developed demodulator is more linear, compared to that of the original differentiate-divide demodulator. This implies that the developed demodulator can be implemented using smaller sized FPGAs, thus reducing cost and the power dissipated. The small silicon area occupied by the demodulator gives room for instantiation of more demodulators and other signal processing units on the same FPGA chip.

Keywords: Software Defined Radio, FM Demodulator, Differentiate-Divide Demodulator

1. Introduction

The advent of digital signal processing technology and software technique has created new possibilities in radio communication. One of such possibilities is the emergence of software defined-radio receivers. Unlike traditional receivers, whose functionality is defined by the arrangement of hardware components within the receiver; the functionality of a software defined radio is software defined (Tuttlebee, 2002). Conventional analogue receivers require hardware building blocks like intermediate frequency (IF) amplifiers, mixers and local oscillators. These are normally implemented in hardware, using electronic components such as diodes and transistors. In software-defined radio receivers, these basic functions are implemented in software (Di Stefano *et al*, 2006).

The use of software adds a new dimension of flexibility to radio receivers. For example, it is now possible to receive broadcasts from a number of stations simultaneously, using a digital computer running appropriate software (Szlachetko and Lewandowski, 2013; Rudra, 2003). The received signals can then be stored as digital files on the hard disk of the computer for analysis and other uses. This feature is of use to journalists and broadcast regulatory bodies, that need to monitor a number of broadcasting stations at a time. Without software radio receivers, a number of conventional hardware based receiver will need to be tuned to different stations for proper monitoring. This is inconvenient as well as expensive.

Software defined radio receivers can also be designed to adapt to different modulation formats, without any change in hardware. For example, an FM software-defined radio can easily be reconfigured as an AM receiver by a simple change of software. This does not require any change of hardware component. In contrast, if we want to receive an FM broadcast using an analogue receiver, an FM receiver is required. We cannot use the same FM receiver as an AM receiver; an AM receiver is needed, in spite of the fact that both receivers have a number of hardware components (mixers, local oscillators, intermediate- frequency amplifiers, etc) in common (Le, 2004).

In spite of such an amazing flexibility offered by software defined radios, they are currently not as popular as conventional receivers. This is because they are much more expensive than conventional radio receivers (Cloninger, 2003). Software defined radio receivers can be designed to run on a number of hardware platforms. Such platforms include the following (Hentati *et al*, 2012): Field Programmable Gate Arrays (FPGAs), Digital Signal Processors, Application specific integrated circuits (ASICs), and General Purpose Processors (GPP). The choice of hardware platform depends on a lot of factors, such as cost and the application.

The basic structure of all SDRs is illustrated in Figure 1. The radio frequency (RF) signals picked up by the antenna are conditioned prior to sampling. Ideally, this conditioning is little more than amplification by a low-noise amplifier (LNA). Given the current state of technology, the conditioning usually consists of additional tasks such as filtering and frequency translation into an intermediate frequency (IF). After conversion to the discrete-time domain, the desired frequency band is isolated using a channelizer. The desired frequency band is translated to complex (or I/Q) baseband and re-sampled to a lower, more manageable sample rate (Rice *et al*, 2009). Multirate techniques are commonly used to simultaneously carry out the channelization, re-sampling and frequency down conversion.

An essential part of a software-defined FM radio receiver is the sampled-data FM demodulator. Examples of sampled data FM demodulators are the differentiate-divide demodulator, the arctangent-derivative demodulator and the phase locked loop. These demodulators have been compared elsewhere, in terms of their silicon area requirement on an FPGA and processing time (Rice *et al*, 2009). In cost sensitive applications, it is desirable to implement a demodulator that requires very little hardware. This work seeks to develop an FM demodulator that is competitive in terms of silicon resources requirement on an FPGA chip.

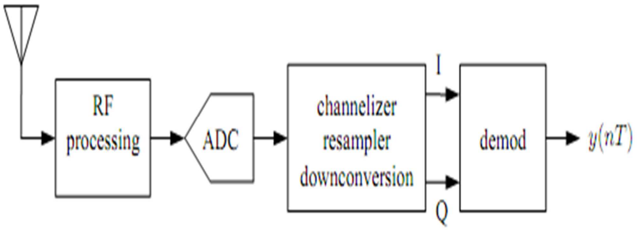


Figure 1. *Block diagram of a typical Software defined radio (Rice et al, 2009).*

2. Materials and Methods

An FM wave can be expressed as

$$X_c(t) = A_c\cos\left[\omega_c t + 2\pi f_d \int_0^t X(\lambda)d\,\lambda\right] \qquad (1)$$

Where $X_c(t)$ = the frequency modulated wave

$\omega_c = 2\pi f_c$ = angular rotational frequency of the unmodulated carrier wave of frequency f_c

f_d = maximum frequency deviation of the carrier wave from the nominal frequency, f_c

$X(\lambda)$ = the modulating waveform

where λ is a dummy variable.

Equation (1) can be written in the form

$$X_c = [I(t)\cos(\omega_c t) - Q(t)\sin(\omega_c t)\,] \qquad (2)$$

Where

$$I(t) = A_c\cos\left(2\pi f_d \int_0^t X(\lambda)d\,\lambda\right)$$

and

$$Q(t) = A_c\sin\left(2\pi f_d \int_0^t X(\lambda)d\,\lambda\right)$$

$I(t)$ is known as the in-phase component of the FM wave while Q(t) is known as the quadrature component.

$$\frac{Q(t)}{I(t)} = \frac{\sin\left(2\pi f_d \int_0^t X(\lambda)d\lambda\right)}{\cos\left(2\pi f_d \int_0^t X(\lambda)d\lambda\right)} = \tan\left(2\pi f_d \int_0^t X(\lambda)d\,\lambda\right) \qquad (3)$$

Or

$$\left(2\pi f_d \int_0^t X(\lambda)d\,\lambda\right) = \tan^{-1}\left(\frac{Q(t)}{I(t)}\right) \qquad (4)$$

$$X(t) = \left(\frac{1}{2\pi f_d}\right)\frac{d}{dt}\left[\tan^{-1}\left(\frac{Q(t)}{I(t)}\right)\right] \qquad (5)$$

$$= \left(\frac{1}{2\pi f_d}\right)\left[\frac{I(t)\frac{dQ(t)}{dt} - Q(t)\frac{dI(t)}{dt}}{(I(t))^2 + (Q(t))^2}\right] \qquad (6)$$

If $X(\lambda) = A_m\cos 2\pi f_m t$, a sinusoidal input with frequency f_m and amplitude A_m, then

$$X_c(t) = A_c\cos\left[\omega_c t + 2\pi f_d A_m \int_0^t \cos 2\pi f_m \lambda\, d\,\lambda\right] \qquad (7)$$

$$X_c(t) = A_c\cos\left[\omega_c t + \frac{f_d A_m}{f_m}\sin 2\pi f_m t\right] \qquad (8)$$

$$X_c(t) = A_c\cos[\omega_c t + \beta\sin 2\pi f_m t] \qquad (9)$$

Where $\beta = \frac{f_d A_m}{f_m}$

The quantity β is known as the modulation index.

The FM signal requires an infinite bandwidth, but most (90%) of its energy occupies a bandwidth, given by

$$B_{fm} = 2nf_{m.} = 2(\beta + 1)f_{m.}$$

Equation (1) suggests that the FM signal can be demodulated by differentiating it and then performing peak detection.

A discrete-time FM signal, derived from sampling the continuous-time FM signal at a given sampling period T, is described by the expression

$$X_c(nT) = A_c\cos[\omega_c nT + 2\pi f_d T \textstyle\sum_{i=0}^{n-1} X(i)] \qquad (10)$$

where $\mathrm{n} = 0, 1, 2 \ldots$

For a discrete-time FM signal, (5) becomes

$$\mathrm{X(n)} = \left(\frac{1}{2\pi \mathrm{f_d}}\right)\frac{\mathrm{d}}{\mathrm{dt}}\left[\tan^{-1}\left(\frac{\mathrm{Q(n)}}{\mathrm{I(n)}}\right)\right] \quad (11)$$

And (6) becomes

$$\mathrm{X(n)} = \left(\frac{1}{2\pi \mathrm{f_d}}\right)\left[\frac{\mathrm{I(n)}\frac{\mathrm{dQ(n)}}{\mathrm{dt}} - \mathrm{Q(n)}\frac{\mathrm{dI(n)}}{\mathrm{dt}}}{(\mathrm{I(n)})^2 + (\mathrm{Q(n)})^2}\right] \quad (12)$$

The arctangent-differentiate demodulator is based on the direct implementation of (11). The block diagram is given in Figure 2. The differentiate-divide demodulator is based on (12). The block diagram is given in Figure 3.

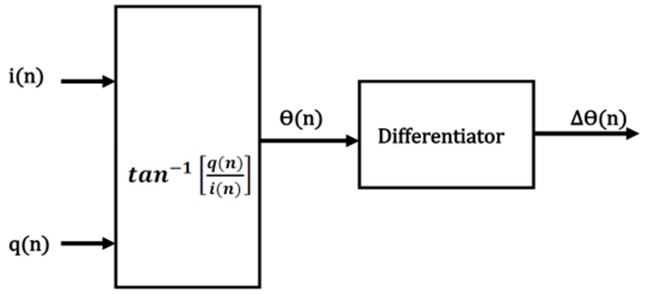


Figure 2. *Block diagram of the Arctangent-divide FM demodulator (Rice et al, 2009).*

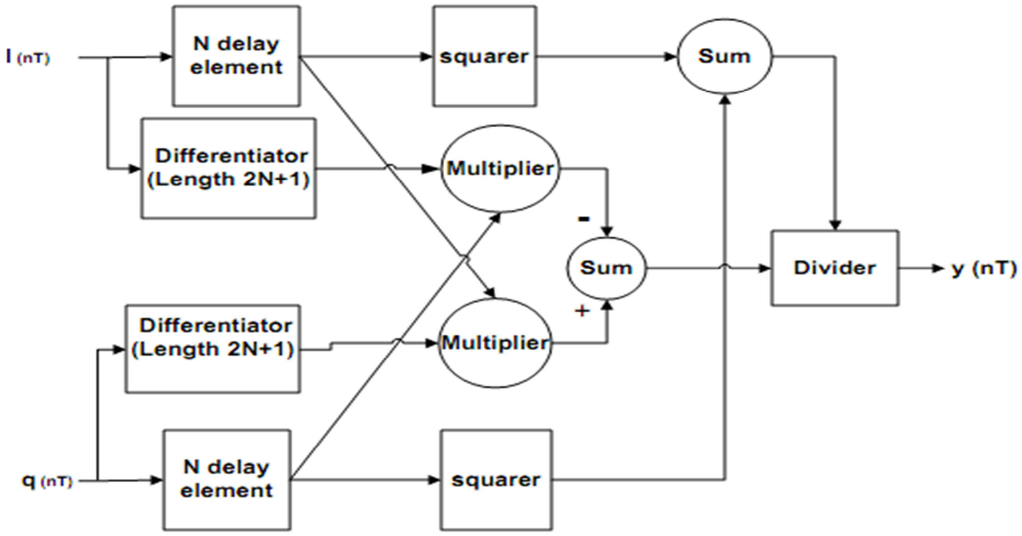


Figure 3. *Block diagram of the differentiate-divide demodulator.*

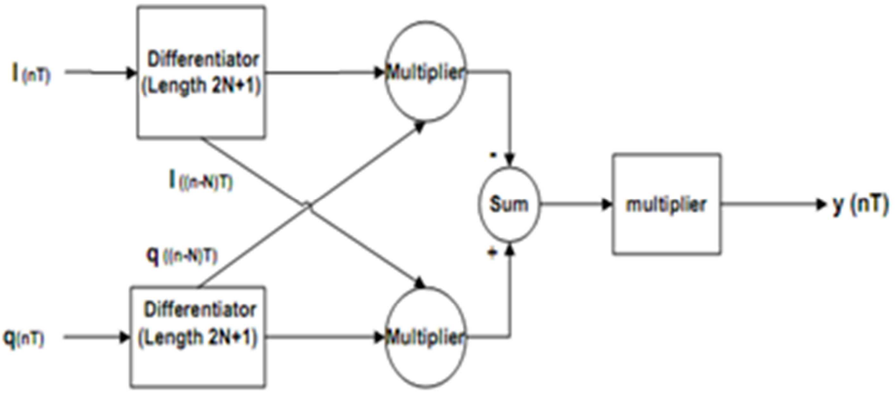


Figure 4. *The modified differentiate- divide demodulator.*

The modified differentiate-divide demodulator, investigated in this work, is based on the direct modification of equation (12). It is observed that the term $(I(n))^2 + (Q(n))^2$ *is simply* $(A_c)^2$. Equation (12) becomes

$$y(n) = \left(\frac{1}{{A_c}^2 2\pi f_d}\right) \times \left[I(n)\frac{dQ(n)}{dt} - Q(n)\frac{dI(n)}{dt}\right] \quad (13)$$

The block diagram of the modified differentiate-divide demodulator is given in Figure 4.

An important part of the demodulators is the discrete-time differentiators, which are normally FIR differentiators, to avoid distortion. A discrete-time differentiator structure suggested by Lyons (2007) was used in the implementation of the demodulator. This differentiator is described by the difference equation

$$y(n) = \frac{-x(n)}{16} + x(n-2) - x(n-4) + \frac{x(n-6)}{16} \quad (14)$$

where y (n) and x (n) are the output and input samples respectively. This differentiator does not require the use of a multiplier, a component which is area intensive to implement on an FPGA chip. The divide by 16 operations can easily be carried out by bit-shifting. The models of the two demodulators developed using Simulink, are shown in Figures 5 and 6.

Equation (14) is a little bit more complex than that of the forward and central difference equations. However, the implementation of this differentiator does not require the use of a multiplier, because the division (by 16) can easily be carried out by bit shifting. Moreover, the amplitude response of this differentiator is given as

$$\left|H(e^{j2\pi r})\right| = 2\left[\sin(2\pi r) - \frac{1}{16}\sin(6\pi r)\right] \qquad (15)$$

The phase response is given as

$$ø(r) = \frac{\pi}{2} - 6\pi r \qquad (16)$$

This filter has a group delay of exactly three sampling periods.

The digital differentiator used in the standard differentiate-divide demodulator was designed using Simulink Filter Design Toolbox. It has a tap length of 31. The silicon area requirements of the standard differentiate-divide demodulator and the modified differentiate-divide demodulator are compared. Because it will be resource intensive on an FPGA chip to implement a floating point algorithm, all the parameters in the models were converted to fixed point numbers. The VHDL codes of the demodulators were generated using HDL Coder, a tool available in MATLAB. The VHDL codes were then simulated on Quartus 2 software from Altera Inc. The FPGA device chosen is EP2S15F484C3, manufactured by Altera Inc.

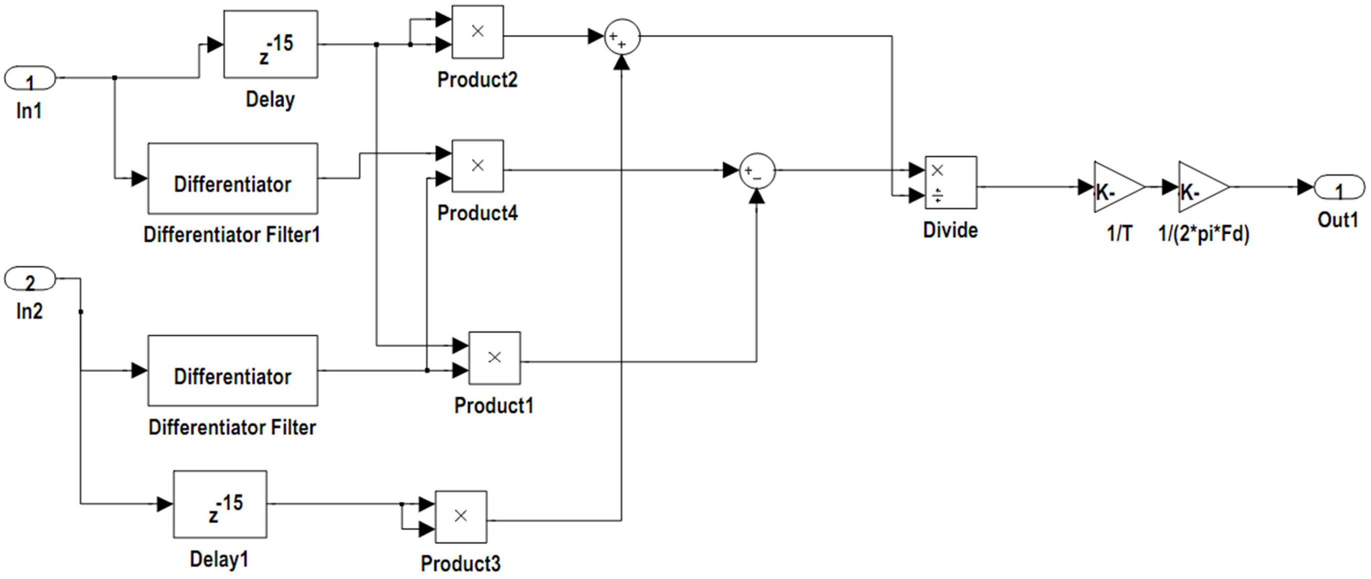


Figure 5. *Model of the differentiate divide demodulator.*

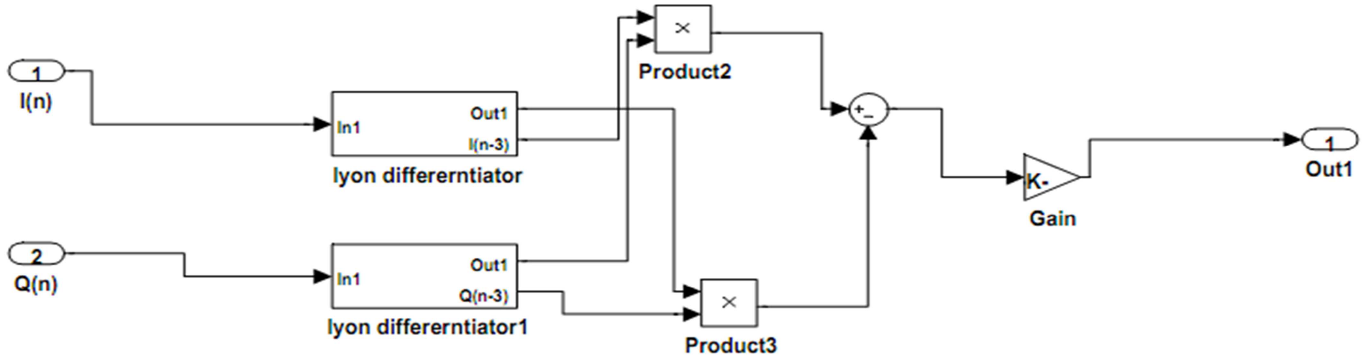


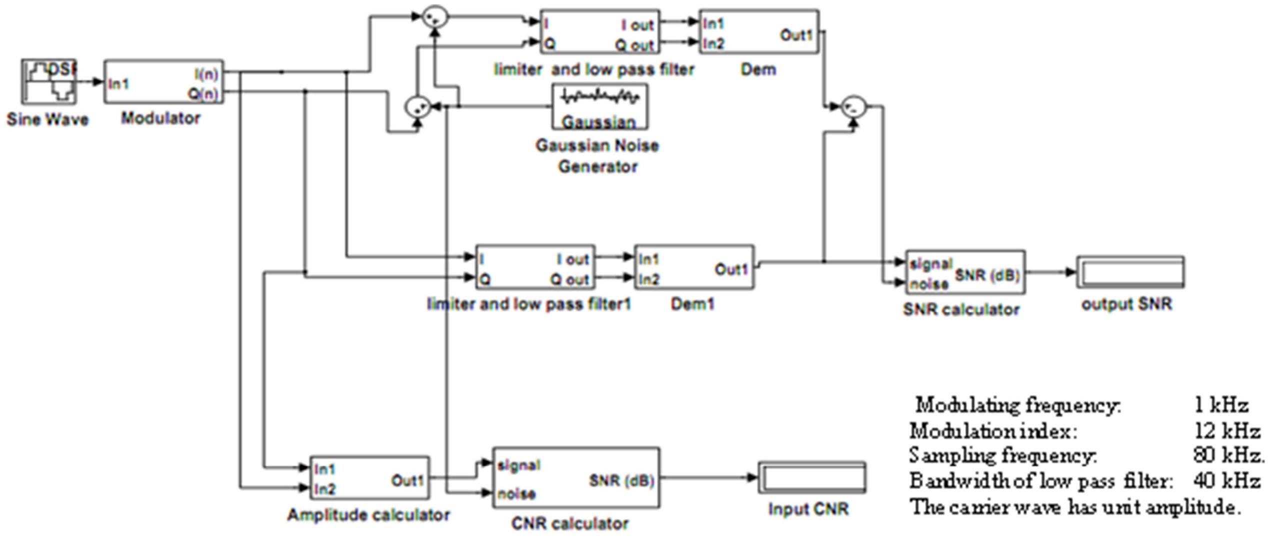
Figure 6. Model of the modified differentiate-divide demodulator.

As a signal processing system, the output signal to noise ratio (SNR) relative to the input signal carrier to noise ratio (CNR) is a reflection of the noise suppression qualities of the demodulator. The signal to noise ratio (SNR) of a sampled data signal can be calculated using the formula.

$$SNR = \frac{wanted\ signal\ power}{noise\ signal\ power} = \frac{\frac{1}{N}\sum_{n=0}^{N-1}[x_s(n)]^2}{\frac{1}{N}\sum_{n=0}^{N-1}[x_n(n)]^2} \qquad (15a)$$

$$SNR_{db} = 10log_{10}SNR \qquad (15b)$$

The Simulink model used to estimate the noise suppression abilities of the modified differentiate-divide demodulators is shown in Figure 7.



Modulating frequency: 1 kHz
Modulation index: 12 kHz
Sampling frequency: 80 kHz.
Bandwidth of low pass filter: 40 kHz
The carrier wave has unit amplitude.

Figure 7. *SNR estimation of the modified demodulator.*

Gaussian noise samples were added to the baseband FM signal generated by the modulator. This noisy signal is then passed through a low pass filter and an amplitude limiter. The bandwidth of the low pass filter was chosen to be a little bit higher than the Carson bandwidth of the input FM signals. This noisy FM signal is then demodulated by the demodulators.

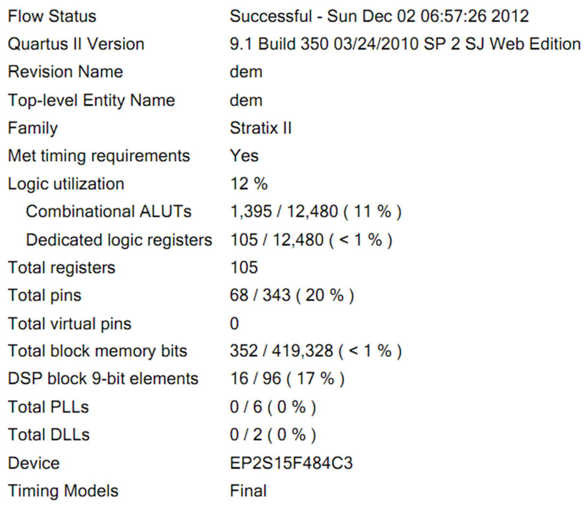
A noiseless output sequence is generated by demodulating the baseband FM signal without any noise samples. The noise sequence is now obtained by subtracting the noiseless output signal from the noisy signal. The input carrier to noise ratio, as well as the output signal to noise ratio is calculated using a model based on (15).

3. Results and Discussions

The hardware resources requirement of the demodulators were also investigated using Quartus software. The compilation results generated by Quartus 2 for the standard differentiate-divide demodulator and the simplified demodulator are given in Figures 8 and 9 respectively.

The original demodulator required 1,395 combinational ALUT while the simplified demodulator required just 187 (Figures 8 and 9). The modified demodulator requires 3% of logic utilization of the resources of the chosen FPGA chip, while the original demodulator requires 12%. The maximum clocking frequency of the original differentiate divide demodulator, as reported by Quartus 2 software on simulation is 451.57MHz. For the modified differentiate divide demodulator, the maximum clocking frequency is 500MHz. Accordingly, Figures 8 and 9 show that less than 1% and 2% of the dedicated logic registers were respectively used by the original and modified demodulators.

The noise suppression ability of the modified demodulator is investigated by using this demodulator to process a noisy baseband FM signal. A zero mean Gaussian noise generator model is used to generate the noise samples. The power (variance) level of the noise generator is varied to obtain various values of input carrier to noise ratio of the modulated FM signal.

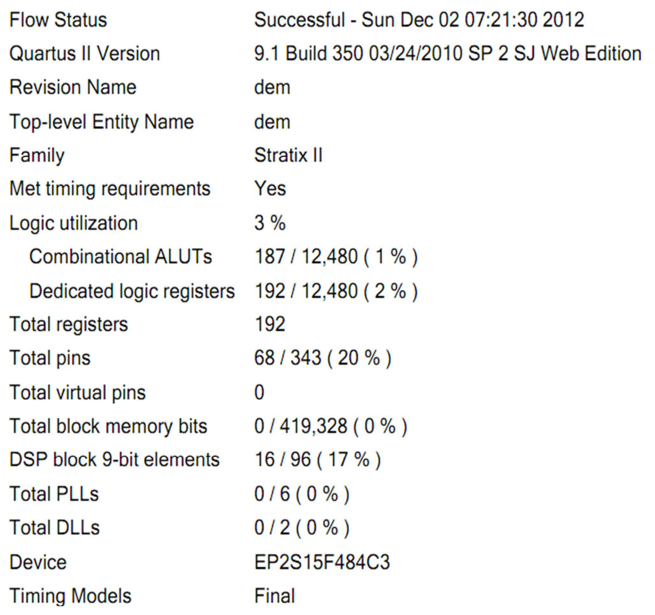


Flow Status	Successful - Sun Dec 02 06:57:26 2012
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Web Edition
Revision Name	dem
Top-level Entity Name	dem
Family	Stratix II
Met timing requirements	Yes
Logic utilization	12 %
Combinational ALUTs	1,395 / 12,480 (11 %)
Dedicated logic registers	105 / 12,480 (< 1 %)
Total registers	105
Total pins	68 / 343 (20 %)
Total virtual pins	0
Total block memory bits	352 / 419,328 (< 1 %)
DSP block 9-bit elements	16 / 96 (17 %)
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 2 (0 %)
Device	EP2S15F484C3
Timing Models	Final

Figure 8. *Quartus 2 compilation report for the original differentiate-divide demodulator.*

An example of the noisy demodulated signal obtained from the output of the demodulator is shown in the Figure 10.

A plot of the input CNR versus the output SNR for the modified differentiate-divide demodulator is given in Figure 4.7. The SNR plot of the original differentiate-divide demodulator, obtained from (Rice et al, 2009), is shown for comparison. The modified demodulator has an output SNR that increases linearly with the input CNR. From Figure 11, it can be deduced that the standard differentiate divide demodulator has a higher output SNR than the modified demodulator, for input CNR less than 45dB. The SNR plot of the modified differentiate-demodulator has a better linearity, compared to that of the standard differentiate-divide demodulator.



Flow Status	Successful - Sun Dec 02 07:21:30 2012
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Web Edition
Revision Name	dem
Top-level Entity Name	dem
Family	Stratix II
Met timing requirements	Yes
Logic utilization	3 %
Combinational ALUTs	187 / 12,480 (1 %)
Dedicated logic registers	192 / 12,480 (2 %)
Total registers	192
Total pins	68 / 343 (20 %)
Total virtual pins	0
Total block memory bits	0 / 419,328 (0 %)
DSP block 9-bit elements	16 / 96 (17 %)
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 2 (0 %)
Device	EP2S15F484C3
Timing Models	Final

Figure 9. *Quartus 2 compilation report for the simplified differentiate-divide demodulator.*

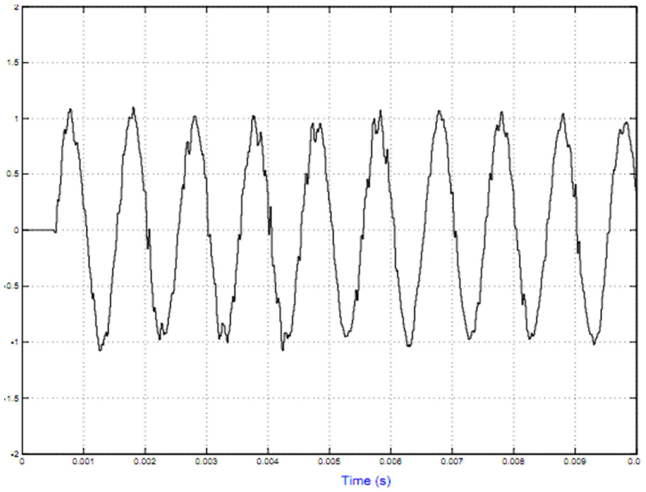


Figure 10. *Demodulated noisy FM signal.*

4. Conclusion

This paper explored the possibility of reducing the hardware resources requirement of an FPGA-based differentiate-divide FM demodulator. A modified algorithm was developed, and then compared with the standard differentiate divide demodulator in terms of hardware requirements and noise suppression abilities. The sampled-data FM demodulator developed is based on the modification of the structure of the standard differentiate-divide demodulator. The hardware resources requirement and the maximum clocking speed of the developed demodulator were investigated using Quartus 2 software. The target FPGA chip is a Virtex 4 FPGA produced by Altera Inc. Also, the signal processing property of the demodulator, quantified in terms of the output SNR versus input CNR, was investigated using MATLAB/SIMULINK.

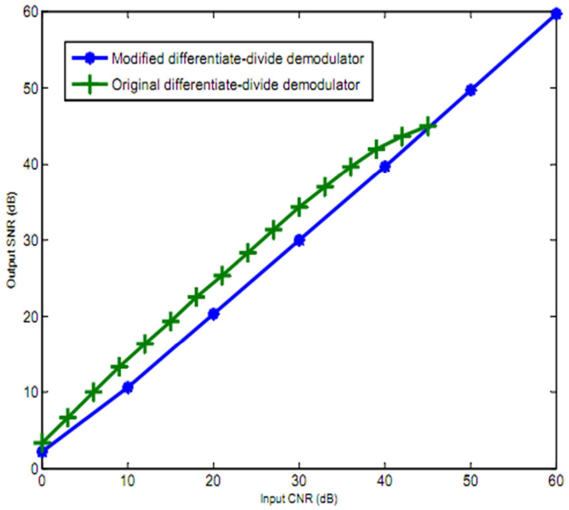


Figure 11. *Input CNR versus output SNR plot.*

The modified differentiate-divide FM demodulator developed requires 187 adaptive look up tables (ALUTs) and 192 dedicated registers on the FPGA chip. The overall hardware resource requirement was 3% of the total resources available on the selected FPGA. In comparison, the original differentiate-divide demodulator required 1,395 ALUTs and 105 dedicated registers; its total hardware resource requirement is 12% of the available FPGA resources.

The developed demodulator achieved a maximum clocking speed of 500 MHz, compared to the 451. 7MHz achieved by the standard differentiate-divide demodulator. The output signal to noise ratio (SNR) of the developed demodulator is lower than that of the standard differentiate divide-demodulator when the input carrier to noise ratio (CNR) is less than 45 decibels. The SNR plot of the developed demodulator is more linear, compared to that of the original differentiate-divide demodulator. The modified demodulator requires a quarter of the hardware resources on an FPGA chip. It can also be clocked at a faster rate, thus exhibiting a higher silicon area/ time ratio. This implies that the four of the modified demodulators can be instantiated on the FPGA chip for every original differentiate divide demodulator implemented using the same FPGA. The additional demodulators can be used to demodulate other available FM broadcast channels, leading to a lower cost multi-channel FM receiver. The standard differentiate-divide demodulator has a higher output SNR for input CNR lower than 45dB. However, the CNR response of the modified algorithm is essentially linear.

References

[1] Cloninger, C., (2003): "Software-Defined Radio comes of Age", Available online: www.eetimes.com, (Accessed date: 17th March, 2012).

[2] Di Stefano, A., Fiscelli, G. and Giaconia, C. G. (2006): "An FPGA- Based Software-Defined Radio Platform for the 2.4GHz ISM Band". Research in Microelectronics and Electronics 2006, Ph.D, pp. 73-76.

[3] Gustafsson, E. M. I., (2008): “Reconfigurable Analog to Digital Converters for Low Power Wireless Applications”, Doctoral Dissertation. KTH-Royal Institute of Technology, Stockholm, Sweden.

[4] Hentati, M., Nafkha, A., Leray, P., Nezan, J. and Abid, M., (2012):” Software Defined Radio Equipment: What’s the Best Design Approach to Reduce Power Consumption and Increase Reconfigurability?”, *International Journal of Computer Applications*, Vol. 45, No. 14, pp. 26-32.

[5] Le, T. H., (2004):”AM/FM Digital Radio Receiver Implementation in FPGA”, M.Sc Thesis University of Adelaide, Adelaide.

[6] Lyons, R. G, (2007):” A differentiator With a Difference”, Available online: http://www.dsprelated.com, (Accessed Date: 10th March, 2012).

[7] Ogunseye, A.A. (2013): Development of a Sampled Data Frequency Modulation (FM) Demodulator for the Implementation of Software Defined Radios, An M.Sc. Thesis submitted to the Department of Electronic and Electrical Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria.

[8] Rice, M., Padilla, M. and Nelson, B., (2009): “On FM Demodulators in Software Defined Radios Using FPGAs”,*IEEE International* Conference on Military Communications (MILCOM), pp. 1-7.

[9] Rudra, A., (2003): “Multichannel, Multiband VHF Software Radio Based Receiver Eliminates R.F Downconversion”, Available online: http://mobiledevdesign.com, (Accessed date: 10th June, 2012).

[10] Szalchetco, B. and Lewandowski, A., (2013): “A Multichannel Receiver of the Experimental FM Based Passive Radar Using Software Defined Radio Technology”, *International Journal of Electronics and Telecommunications,* Volume 58, Issue 4, pp. 301–306.

[11] Tuttlebee, W., (2002): “Software Defined Radio: Enabling Technologies”, West Sussex, Wiley, pp. 6-7.